DEPARTMENT OF ELECTRONIC & ELECTRICAL ENGINEERING

University of Strathclyde Engineering

Wind Farm Simulation Modelling and Control

Saman Poushpas

A thesis presented in fulfilment of the requirements for the degree of Doctor of Philosophy

Centre for Doctoral Training in Wind Energy Systems
Department of Electronic and Electrical Engineering
University of Strathclyde
Glasgow G1 1XW
Scotland, UK

August 2016

This thesis is the result of the author's original research. It has been composed by the author and has not previously been submitted for examination which has led to the award of a degree.

"Necessity is the mother of invention"

Acknowledgments

First and foremost, I would like to express my sincere gratitude to Professor Leithead, for his constant support, guidance and motivation, without which this work would not have been possible.

I would like to extend my gratitude to my second supervisor Dr Hong Yue and other colleagues and friends in the Centre of Doctoral Training in Strathclyde University for their support and friendship who made my PhD an enjoyable experience. Special thanks to Adam Stock and Sung Hur, for their excellent advice and guidance.

I am truly grateful to my family; my mum and dad for their priceless love, and encouragement throughout all level of my studies, my mother-in-law and father-in-law, my brother, sister, and my brother-in-law for their endless support and motivations, you have been outstanding inspiration to me. Finally, a very special thanks to my wife Bahareh for her infinite patience and encouragement at every step of this journey. I could never have completed this PhD without your immeasurable love and support. I dedicate this Thesis to you and my family.

Abstract

The increasing trend towards large-scale deployment of wind energy imposes numerous operational challenges regarding large integration of wind power to the transmission system including the maintenance of system stability due to the uncertain nature of wind power. Thus, the traditional way of operating and controlling wind turbines and wind power plants are becoming less acceptable. Furthermore, wind power plants are progressively being subjected to the Transmission network operators (TSO's) regulations and are required to operate as a single controllable unit, similar to the conventional power plants, to provide active power regulation. To provide such a functionality, wind turbines in a wind farm must provide more flexible output power control in a quick and safe operating manner. Additionally, the operation of the wind turbines must be coordinated so as to operate the whole wind power plant as a single controllable entity.

The main goal of this thesis is to generate a wind farm Simulink model that captures all the essential dynamics for the wind farm controller design and load analysis. To achieve this main aim, a mathematic wind farm model has been developed which offers sufficiently fast simulation for iterative controller design task, and contains a suitable wind-field model that provides a suitable representation of the wind-field and wake propagation through the wind farm. The wind farm controller design with the objectives of primary frequency response and power optimisation has also been investigated.

Contents

Chapter 1: Introduction

IN this thesis various wind turbine models and their controllers are investigated and a suitable model with relevant properties for wind farm modelling is outlined. A wind farm model suitable for fast simulation and controller design is developed in Matlab/ Simulink. This model is suitable for investigating wind turbines' performance operating in wake conditions. The wind turbine models are capable of adjusting their power output according to a power set-point change request from wind farm control algorithms in a quick and safe operating manner. A general wind farm model construction tool is also developed to build a wind farm model containing a large number of turbines. This construction tool accepts user defined parameters and builds the wind farm accordingly. The developed wind farm control algorithm available in the code can be conveniently modified following different requirements.

1.1 Motivation

The motivation behind this work is the need for a wind farm model sufficiently fast and detailed for iterative wind farm controller design and turbine load analysis. Since information about the commercial wind turbine dynamics and their controller structure are not freely available, most of the available wind turbine and wind farm models contain very simplified wind turbine dynamics and particularly neglect the wind turbine controller. However, wind turbine control dynamics have significant influence on the wind turbine and wind farm operation. Hence, a suitable wind farm model with relevant wind turbine dynamics for load analysis and controller design is developed in this thesis.

1.2 Thesis objectives

The main focus of this research is to develop an applicable wind farm Simulink model for adequately fast simulation and controller design purposes. The wind farm model must be suitable for load analysis and controller design. Thus, the initial task is to define the general requirements for the wind farm model. Explicitly, the wind farm model must be capable of fast simulation with large number of wind turbines included. It must contain the necessary structural modes and dynamics of wind turbines. The wind turbine model, incorporated in the modelling, must contain a suitable full envelope controller for load analysis. Since interaction between the rotor and the turbulent wind field induce loads on wind turbine structure, therefore, a suitable representation of turbulent wind-field at the farm level and turbine level needs to be included. The wind-field model also must include appropriate wake effects and their propagation through the wind farm for wind farm controller design. Additionally, a wind farm control algorithm suitable for coordinating the wind turbines operation allowing for flexible control of wind farm power output must be included. Since the wind turbine full envelope controller, is not essentially designed for the power set-point adjustment of the wind turbine, an appropriate interface between the wind farm controller and the wind turbines' full envelope controller is required to provide this functionality in a safe and quick operating manner.

Moreover, wind farm control objectives are defined to investigate the primary frequency response and power maximisation of the wind farm.

1.3 Summary of the thesis

Chapter 2 overviews the history and general background of wind power, investigates the challenges facing wind power concerning high integration of wind power to the power network and underlines available options from the control point of view to resolve these issues.

Chapter 3 provides a summary of the relevant wind energy background theory for wind farm modelling and controller design and provides literature review, focusing on relevant wind turbines and wind farm model available for wind farm controller design.

Chapter 4 details the 5MW Supergen Exemplar wind turbine model which is suitable for wind farm modelling and controller design with the objectives defined in this thesis. The wind turbine dynamics, its full envelope controller, and its augmented power adjusting controller (PAC) functionalities are described.

Chapter 5 describes the wind farm modelling procedure, including wind-field modelling, and wind farm construction procedure. It demonstrates the operation and control performance of the established wind farm model for different operating and control conditions. Power reference tracking control mode is implemented to investigate the wind farm operation performance in curtailed control mode. The power maximisation objective of the wind farm controller is also investigated through a positive reference signal request.

Chapter 6 describes a wind farm controller algorithm developed for primary frequency response and power curtailment control. The wind farm controller communicates with the power adjusting controller of the involved wind turbines to coordinate the wind turbines operation according to their operating status.

Chapter 7 investigates the wind farm model performance for different control scenarios. Power curtailment control of a wind farm with large number of wind turbines is presented. The negative and positive wind farm power set-point adjustment is investigated at different wind conditions.

Chapter 8 presents the primary frequency response of the wind farm. A wind farm model of 10 turbines is built and utilised to evaluate the performance of the wind farm controller in this control mode. The algorithm includes synthetic inertia and droop control loops combined with the power reference tracking objective of the controller.

In chapter 9 the power/load optimization objectives of the developed wind farm controller is investigated. Heat & Flux control approach is utilized to investigate the power maximization objective of the wind farm controller. Additionally, a simple and effective optimisation control approach is proposed for improving the wind turbines operating performance in a wind farm.

Chapter 10 provides concluding remarks for the whole thesis and discusses directions for future research in this area.

1.4 Contribution to knowledge

- Developed a wind farm model with relevant fidelity level for fast simulation and controller design in Matlab\Simulink. The wind farm model includes a dynamic wind-field model with appropriate representation of wake propagation through the wind farm.
- Established and interface controller between the wind turbines' full envelop controller and a wind farm controller for dynamic power set-point adjustment of the turbines.
- Developed a hierarchical wind farm control architecture allowing for dynamic power adjustment of the wind turbines' power set-point in primary frequency response and power curtailment control mode.
- Proposed a simple and effective optimisation control strategy with the aim of improving the wind turbines' performance in a wind farm.

1.5 Publications

- S. Poushpas and W. Leithead, "Wind farm simulation modelling and control for primary frequency support", in 4th International Conference on Renewable Power Generation, Beijing, China 2015
- S. Poushpas and W. Leithead, "Wind farm control through dynamic coordination of wind turbines reference power", in 1st International Conference on Renewable Energy Offshore, Lisbon 2014
- S. Poushpas and W. Leithead, "application of coordinated and power coordinated control of a 5MW Supergen wind turbine model", in 8th PhD Seminar on Wind Energy in Europe, ETH Zurich 2012, pp. 92

Chapter 2: Context

WIND power has been used in various ways throughout recorded history. The use of wind power dates back to the early civilizations in the Middle East. Sail boats are the first known use of wind power by the Egyptians. Since then, humans have been harnessing wind energy in numerous ways for different applications such as pumping water, grinding grains etc. The first known documented practical windmill was built and used in Sistan a region in south east of Iran around the 7th century. These windmills, as shown in Figure 2-1, were vertical axis and were used to grind wheat and pump water. These ancient windmills are still in operation in some rural areas to this date.

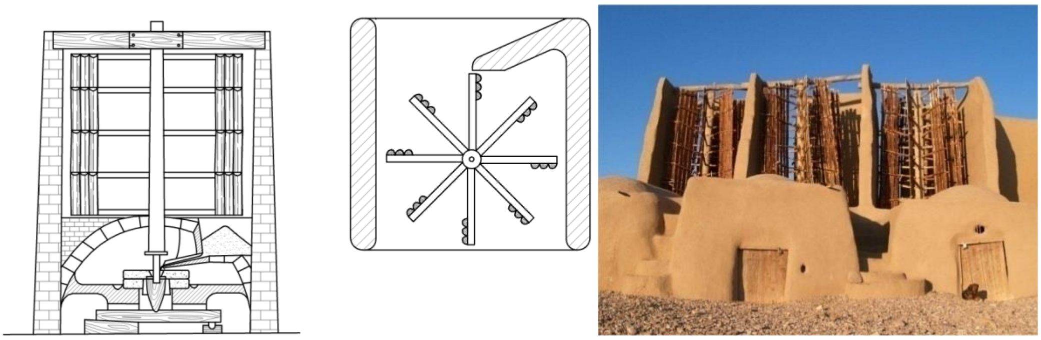


Figure 2-1: Historic windmills[1]

In 1887 Professor James Blyth from Anderson College (University of Strathclyde), Glasgow, built the first wind turbine. He was a pioneer in producing electrical power from wind. His wind turbine, as shown in Figure 2-2, was used to power his holiday home in Marykirk. Around the same time professor Charles F.Brush from Ohio US, built a 12KW wind turbine with 50m rotor diameter and 144 blades.

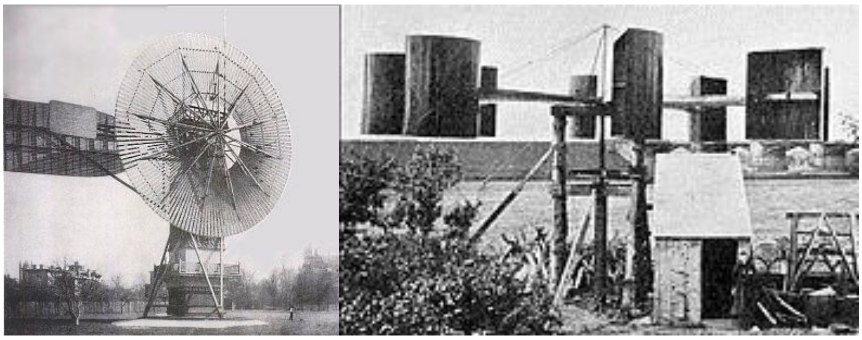


Figure 2-2: Blyth & Brush wind turbines [1]

Around 1903 Poul La Cour from Askov, Denmark discovered that wind turbines with less rotor blades are more efficient in power production. In 1956, Johannes Juul a former student of Poul la Cour designed the first three bladed horizontal axis wind turbine (HAWT), see Figure 2-3. Ever since, the three bladed wind turbine concept has inspired wind turbine engineers and is accepted as the most efficient design for HAWTs in modern wind energy systems.

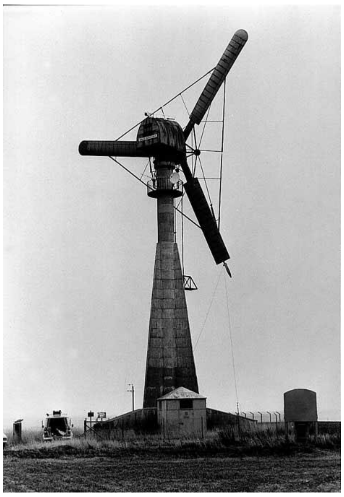


Figure 2-3: The first three bladed wind turbine[2]

2.1 Transition from fossil fuels to renewable resources

From the industrial revolution in the 18th century until now fossil fuels have been the primary source of energy for industries and the major driver for economic growth. Consequently, the level of energy consumption and, thus, demand for fossil fuels have increased exponentially. According to the wind energy council, the global energy demand is expected to double by 2050 [3]. However, the volatility in fossil fuel prices, as a result of political and market uncertainties, can result in unstable economic growth and global energy crisis. The 1970's oil crisis is a clear example of such a shock to the global economy and its consequences.

Moreover, increasing global concerns about the impact of human activities on climate change and increasing level of CO_2 concentration in the atmosphere have been other major drivers to change the picture for the future energy generation and supply. As a result, various obligations are being imposed on developed and developing countries to reduce their greenhouse gas emissions. For instance the Climate Change Act (2008) obliged the UK to reduce its annual greenhouse gas emission by half by 2050 [4].

The necessity for reducing the electricity power generation dependence on fossil fuels and replacing these primary energy resources with sustainable and native (domestic) energy resources has paved the way for significant exploitation of renewable energy resources. Accordingly, the government's support for renewable energy has been increased significantly and renewable energy has driven to the top of the energy policy agenda. Numerous domestic and industrial incentives such as feed-in-tariffs, heat incentives, green certificates etc. have been proposed by governments to promote renewable energy and encourage electrical power generating companies to invest more in renewable energy technologies.

Amongst them, wind energy in particular, has proven to have more potential to become a significant contributor to the future energy mix. Thus, the share of wind power installed capacity in the energy supply market has increased significantly. According to the EWEA 2013 European statistics [5], annual installations of wind power increased from 3.2GW to 11.2GW in the last 13 years since 2000. Figure 2-4 illustrates the percentage share of the wind power in the power mix in the UK.

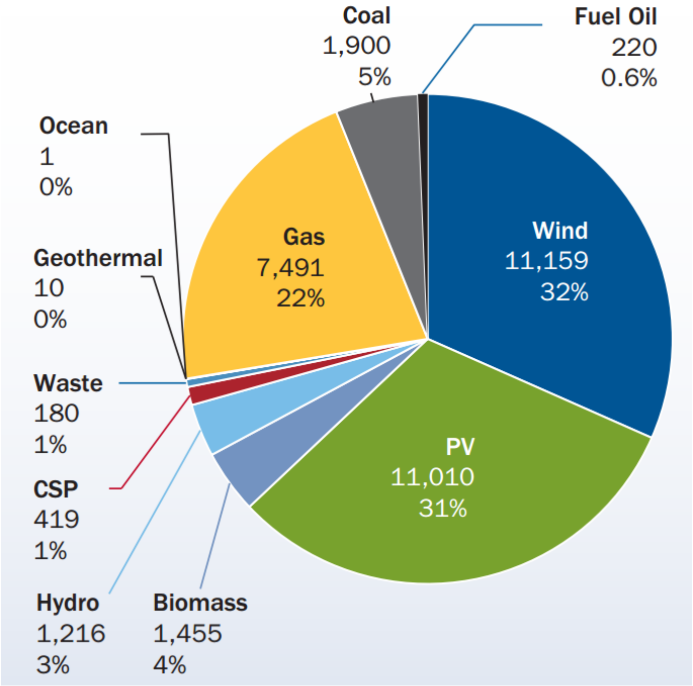


Figure 2-4: Percentage share of wind power in the energy mix for generating electricity in UK[5]

2.2 Challenges facing wind power production

Wind power is one of the most cost effective and promising renewable technologies for replacing fossil fuel based power generation. However, this relatively young technology has to overcome many obstacles in its path towards maturity.

In order for wind power to compete with conventional power generation in the competitive energy supply market, it is essential to make wind power more economically viable. Levelised cost of electricity generation is one of the main financial indicators in wind energy project feasibility studies.

The levelised cost of electricity is defined as the minimum price of unit electricity that breaks even the costs over the lifetime of the project [6], specifically

$$LCOE = \left(\frac{(ICC \times FCR) + LRC + O\&M}{AEP}\right) \tag{2-1}$$

where, ICC is the initial capital cost, FCR is the fixed charge rate, LRC is the levelised replacement cost, $O\&M$ is the operating and maintenance cost and AEP is the annual electricity production or energy yield for a given wind speed distribution plus the mechanical and electrical losses [7].

Therefore, to reduce the LCOE the AEP of the wind turbines must be maximised whilst the capital and operating costs must be minimised. In order to increase the AEP, wind turbines and wind farms must remain operational for most of their life time. In general AEP can be calculated by [7]

$$AEP = 8760 \times \sum_{i=cut-in}^{i=cut-out} P(v_i)f(v_i) \tag{2-2}$$

where $P(v_i)$ is the power curve of the wind turbine and $f(v_i)$ is the frequency distribution of the mean wind speed bins v_i.

However, a number of technical challenges exist that limit the continuous generation and transmission of wind power mainly due to grid integration issues. These problems mostly arise from the variability of the wind and the limited capacity of existing power networks. Particularly, in high wind and

low load periods, transmission network operators would require wind power plant to curtail their power production. Moreover, balancing between power generation and demand must be managed in the most economic and cost effective way.

The task of optimally dispatching power demand between generation units to balance the system at minimum operating cost is known as power system economic dispatch [8]. Solving the economic dispatch problem in a system with high penetration of wind power is an extremely challenging task. In such a system, additional reserve generation capacity is required to compensate for the mismatch between the forecast and the actual level of generated wind power. Normally, reserving generation units are expensive, therefore the levelised cost of wind power can be affected by uncertainties in market values of the reserve generation units.

In addition, as the penetration of wind power into the grid system is increased, the total system inertia can be lowered. Since variable speed wind turbines are decoupled from the AC network by their power converters, they cannot inherently contribute to the system inertia response in a similar manner to conventional power plants. Therefore, maintaining system stability can become a challenging task for wind power generation without compensating for the reduction in the system inertia. Increasingly, wind power has been subjected to greater regulation by TSOs to mitigate issues related to wind power grid integration [9].

Furthermore, aerodynamic interaction between wind turbines can have a significant influence on the performance of a wind power plant. In [10], it was shown that individual control strategy of wind turbines in a row is not the optimal control strategy for operating the wind turbines collectively. It was also suggested that the wind farm array efficiency can be increased by coordinating wind turbines' power set-point to reduce the effect of aerodynamic interaction among them.

Evidently uncertainties regarding wind power along with the limited capacity of the power systems, influence the level of wind power integration into the grid system. Since the commercial wind turbine controllers traditionally do not provide flexible power output control, prediction and control of the wind power level in a grid system is becoming a challenging task. From a control point of view, many of these integration challenges can be addressed by utilising new control systems for wind turbines and wind farms that are capable of providing active power output control similar to conventional power plants.

2.3 Wind farm operation and control as a conventional power plant

Advances in wind turbine technology including increase in wind turbines' size and capacity requires more sophisticated control systems to overcome the technical challenges regarding high exploitation of wind power. Figure 2-5 shows the upscaling trend of wind turbines in the recent years.

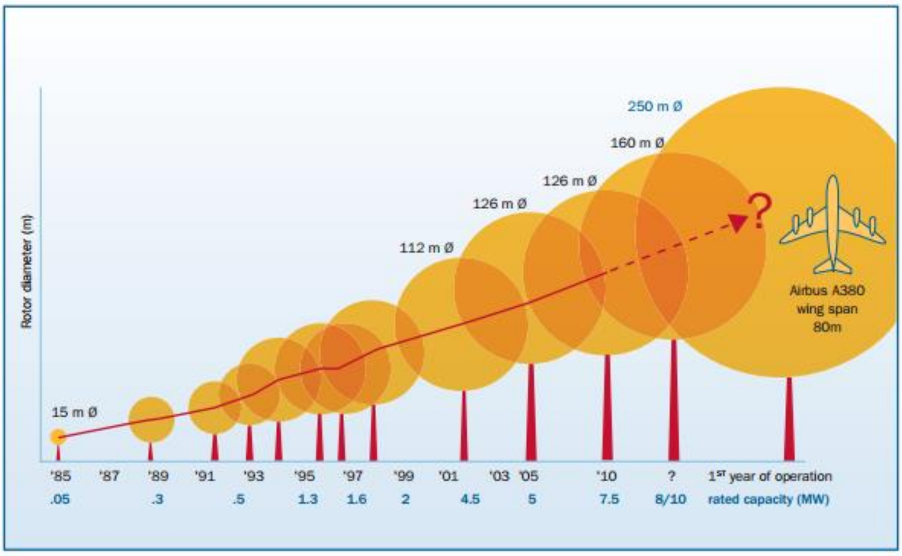


Figure 2-5: Wind turbines up scaling [11]

Improving efficiency and reducing costs are the main drivers for developing more advanced wind turbine and wind farm controllers. It is desirable to control all wind turbines in a wind power plant as a single controllable unit. The independent control and operation of wind turbines in a wind farm for maximising their individual power outputs, regardless of the array losses due to aerodynamic interaction between the turbines, is becoming less acceptable as the installed capacity of wind turbines in the power system is largely increased. The total power output of the grouped wind turbines cannot be controlled flexibly with a traditional control approach, balancing a power system including high level of intermittent wind power, is thus becoming a challenging task. Large wind power plants are required to have more flexible control on their output power. In order to provide such control functionality, wind turbines in a wind farm must be able to adjust their power output in a quick and safe operating manner in all operating conditions similarly to conventional power plants. For this purpose, advanced control strategies are required to coordinate the operation of the wind turbines. In this thesis a wind farm model with flexible power control functionalities is established. A central wind farm controller with various control objectives is designed to address several crucial issues regarding wind power grid integration.

Chapter 3: Background

IN definition, a wind farm is a cluster of aggregated wind turbines which are carefully positioned and installed to convert kinetic energy of wind into electricity. The operating performance of a wind farm evidently depends on the operating performance of individual wind turbines and also the aerodynamic interaction between them. A wind turbine is an electromechanical device with very large moving parts that are prone to wear and tear during their operating life time. Since the structural loads on wind turbines are known as one of the major reasons for their failure, it is essential to alleviate these loads as much as possible. The source of these loads are the external and internal forces acting on wind turbine components. Interaction between the wind-field and rotor induces low frequency loads on to the wind turbine structure due to rotational sampling by the rotor blades.

Wind turbine and wind farm control, provides a promising solution to alleviate the structural loads and increase the performance of wind turbines. To design such a controller, it is essential to have a broad understanding about the wind turbine dynamics, wind turbine aerodynamics, control objectives, and also the aerodynamic interaction between the turbines. In this chapter wind turbine aerodynamics, wind turbine operation and its control objectives are discussed.

3.1 Wind turbine aerodynamics

Wind turbines convert the kinetic energy of the passing wind into mechanical power. The kinetic energy in the wind is the source of aerodynamic forces acting on the wind turbines rotor. The aerodynamics of HAWTs can be described simply by an actuator disc model. However, to closely study the effect of aerodynamic forces acting on the wind turbine structure, blade element theory and blade element momentum theory are more appropriate approaches to describe and investigate the forces produced by passing flow and acting on the wind turbine structure [12].

3.2 Actuator disc theory

Actuator disc theory describes the flow condition at far upstream, in the vicinity, and far downstream of the wind turbine rotor disc based on the conservation of the mass flow rate and momentum theory using a hypothetical stream tube as shown in Figure 3-1.

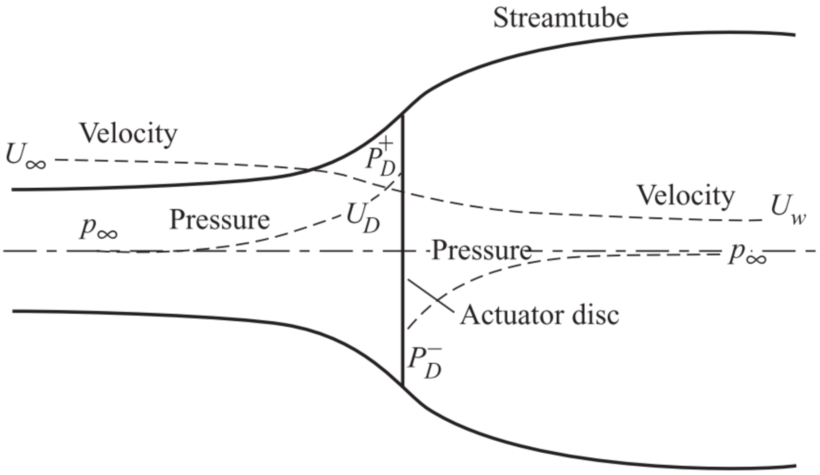


Figure 3-1: Actuator disc model [13]

Mass flow rate conservation theory states that the rate of change of mass must be maintained everywhere in the stream tube, this is,

$$\rho A_\infty U_\infty = \rho A_D U_D = \rho A_W U_W \qquad \textbf{(3-1)}$$

where, ρ is the air density $A_\infty \; and \; U_\infty$ are the area and velocity of the flow in far downstream, respectively; $A_D \; and \; U_D$ are the area and velocity of the flow in the vicinity of the actuator disc, respectively; and $A_W \; and \; U_W$ are the area and velocity of the flow far upstream, respectively.

Since the velocity of the flow field reduces in the vicinity of the disc and far downstream, the stream tube cross-sectional area expands to conserve the mass flow rate throughout the stream tube. For steady flow, the stream-wise flow velocity at the disc is defined as a function of, a_s using:

$$U_D = U_\infty(1 - a_s) \tag{3-2}$$

where, a_s is the steady axial flow induction factor

Momentum theory states that the rate of change of momentum is equal to the mass flow rate multiplied by the overall change in the flow velocity, i.e.

$$\text{rate of change of momentum } = (U_\infty - U_W)\rho A_D U_D \tag{3-3}$$

The rate of change in the momentum arising from the pressure difference across the actuator disc is

$$T = (P_D^+ - P_D^-)A_D = (U_\infty - U_W)\rho A_D U_\infty(1 - a_s) \tag{3-4}$$

where, P_D^+, P_D^- are pressure before and after the actuator disc.

According to Bernoulli's equation, the total energy in the flow must remain constant under steady conditions. Thus, Bernoulli's equation can be applied to the upstream and downstream sections of the stream-tube separately.

$$\frac{1}{2}\rho U^2 + p + \rho g h = constant \tag{3-5}$$

where, U is wind speed, g is the acceleration of gravity, p is pressure and h is height

The pressure difference across the disc can be related to the difference between wind speeds upstream and downstream of the disc in the stream-tube by

$$(P_D^+ - P_D^-) = \frac{1}{2}\rho(U_\infty^2 - U_W^2) \tag{3-6}$$

$$\frac{1}{2}\rho U_\infty^2 + p_\infty = \frac{1}{2}\rho U_D^2 + p_D^+ \tag{3-7}$$

$$\frac{1}{2}\rho U_W^2 + p_\infty = \frac{1}{2}\rho U_D^2 + p_D^- \tag{3-8}$$

where, p_∞ is the pressure at far upstream and downstream of the actuator disc

By substituting (3-7) and (3-8) into (3-6),

$$U_W = (1 - 2a_s)U_\infty \tag{3-9}$$

Therefore, the thrust force acting on the rotor disc is given by

$$T = 2\rho A_D U_\infty^2 a_s(1 - a_s) \tag{3-10}$$

and thus, the power extracted by the actuator disc is given by

$$power = TU_\infty = 2\rho A_D U_\infty^3 a(1 - a_s)^2 \tag{3-11}$$

These relationships are derived by assuming the flow to be incompressible, horizontal, and frictionless and only hold for axial induction factors $a_s < 0.5$.

The power available in the flow can also be computed from its kinetic energy.

The kinetic energy in the moving air is

$$E = \frac{1}{2}mv^2 \tag{3-12}$$

where, m is the mass, and v is the speed of moving air

The instantaneous power of the flow through the disc is

$$P_{wind} = \frac{dE}{dt} = \frac{1}{2}U_\infty^2 \frac{dm}{dt} \tag{3-13}$$

$$\frac{dm}{dt} = \rho A \frac{dx}{dt} \tag{3-14}$$

$$\frac{dx}{dt} = U_\infty \tag{3-15}$$

$$P_{wind} = \frac{1}{2}\rho A_D U_\infty^3 \tag{3-16}$$

The ratio of the extracted power to the available power is defined as the power coefficient

$$C_P = \frac{Power}{P_{wind}} = 4a_s(1 - a_s)^2 \tag{3-17}$$

As a result, the extracted power from the wind is related to the available power in the wind by C_P through

$$Power = \frac{1}{2}\rho U_\infty^3 A_D C_P \tag{3-18}$$

Similar to the power coefficient, the thrust coefficient is also defined as

$$C_T = \frac{Trust}{\frac{1}{2}\rho A_D U_\infty^2} = 4a(1 - a) \tag{3-19}$$

where, a is induction factor.

The rotor power coefficient C_P can be related to the thrust coefficient C_T and the axial induction factor a by:

$$a_s = \frac{1}{2}(1 - \sqrt{1 - C_T}) \tag{3-20}$$

$$C_T = 4a_s(1 - a_s) \tag{3-21}$$

$$C_P = 4a_s(1 - a_s)^2 \tag{3-22}$$

$$C_P = \frac{C_T\left(1 + \sqrt{1 - C_T}\right)}{2} \tag{3-23}$$

According to Frederic Lanchester (1915) and Albert Betz (1919), the theoretical maximum limit to the available power in the wind, which can be extracted by a wind turbine rotor is $C_{Pmax} = \frac{16}{27} \cong 59\%$ [13], that corresponds to an axial induction factor of $a_s = \frac{1}{3}$.

Therefore, wind turbines' power efficiency can be evaluated based on either their C_P curve or power curve.

The power coefficient C_P is a function of the blade pitch angle, β, and the tip speed ratio, λ. A typical $C_P - \lambda$ curve for a 5MW wind turbine is depicted in Figure 3-2.

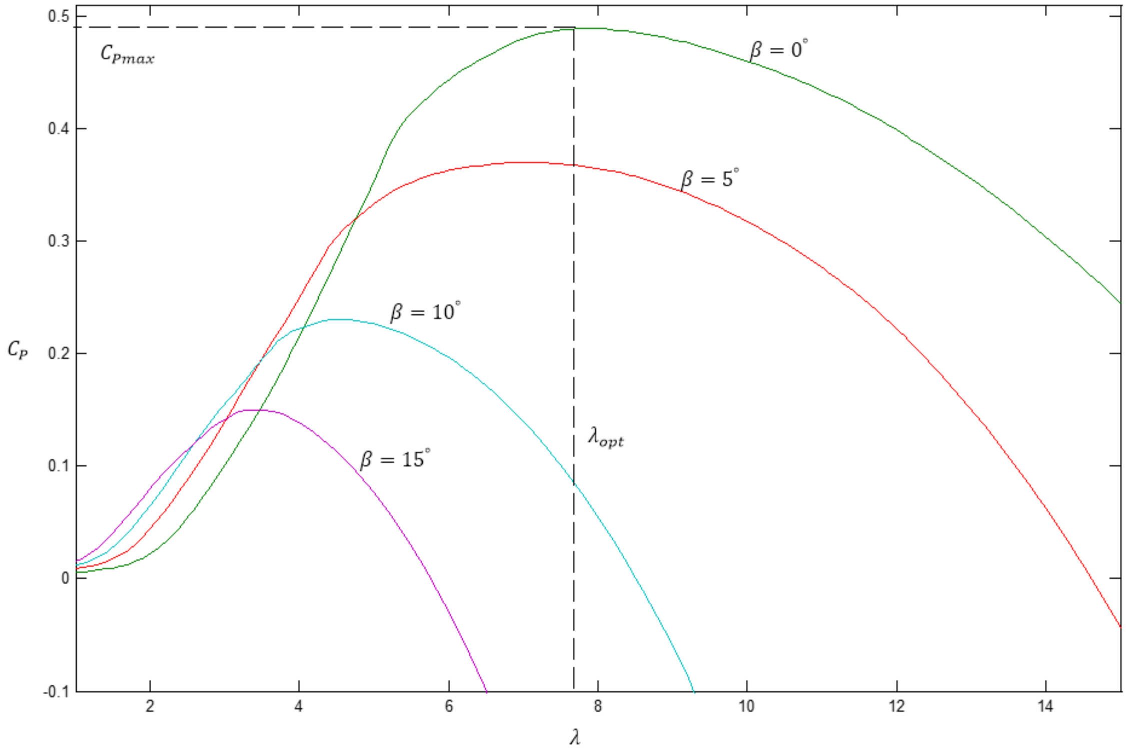


Figure 3-2: A typical $C_P - \lambda$ curve of a 5MW HAWT

For different pitch angles, the performance of the wind turbine rotor can be evaluated based on its $C_P - \lambda$ curve.

The actuator disc model provides significant insight into the flow condition in the wake which is useful for the wind farm modelling and control design. This insight is, however, inadequate for investigating the effect of aerodynamic forces on the rotor blades and the wind turbine structure. As a result, blade element momentum theory (BEM) was developed to describe the aerodynamic forces acting on each blade element based on its geometry characteristics, such as twist and chord length, as well as wind conditions local to the blade element.

3.3 Blade element momentum theory

Aerodynamic forces acting on the rotor are the source of useful power extracted from the air and also the structural loads on the wind turbine. BEM theory describes these forces in details and allows investigation of the effect of these forces on different components of the wind turbine [13]. BEM theory is a combination of classic momentum theory and blade element theory. BEM theory can be described by dividing the blade into a number of sections to calculate the aerodynamic forces for each element individually. It is common practice to assume that there is no aerodynamic interaction between the blade elements and to describe the forces acting on the blade elements merely by the lift and drag coefficients.

Since, each rotor element has different rotational speeds as well as different blade geometry characteristics, each blade element experiences a different flow and thus different forces. By integrating the performance of the blade elements the overall performance characteristic of the blade is determined.

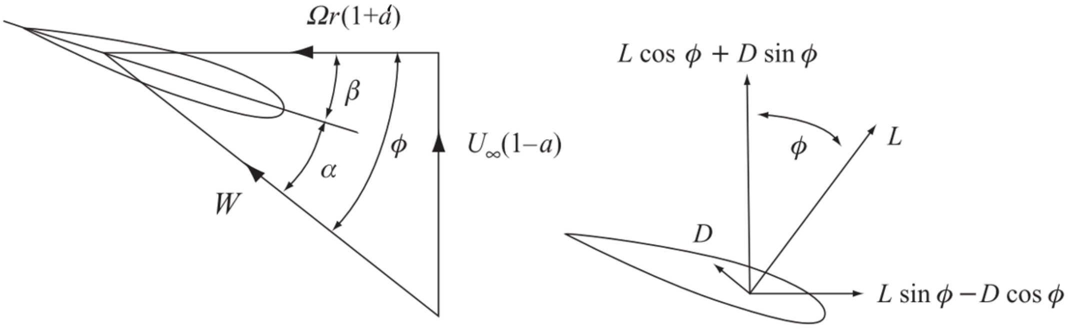


Figure 3-3: Velocities and forces acting on a blade element [13]

Figure 3-3 illustrates the velocities and forces acting on a blade element at radius r relative to the blade element chord line. The angle α between the incident resultant velocity W and the chord line is known as angle of attack. The forces acting on a blade element are determined in terms of the axial induction factor a and the tangential flow induction factor a'.

The resultant flow velocity local to the blade element is

$$W = \sqrt{U_\infty^2(1-a)^2 + r^2\,\Omega^2(1+a')^2} \tag{3-24}$$

where, Ω is the rotor velocity, r is the radius relative to the blade element

The lift and drag forces on a span-wise length δr normal and parallel to the W direction, respectively, are

$$\delta L = \frac{1}{2}\rho W^2 c C_l \delta r \tag{3-25}$$

$$\delta D = \frac{1}{2}\rho W^2 c C_d \delta r \tag{3-26}$$

where, C_l and C_d are lift and drag coefficients which are the characteristic of the aerofoil, and c is the cord length. The angle between the resultant flow to the plane of rotation is ϕ which gives

$$\sin\phi = \frac{U_\infty(1-a_s)}{W}, \quad \cos\phi = \frac{r\Omega(1+a')}{W} \tag{3-27}$$

and therefore, the thrust and torque on a blade element is

$$\delta T = \frac{1}{2}\rho W^2 Bc(C_l\cos\phi + C_d\sin\phi)\delta r \tag{3-28}$$

$$\delta Q = \frac{1}{2}\rho W^2 Bcr(C_l\sin\phi - \mathrm{C_d}\cos\phi)\delta r \tag{3-29}$$

where B is the number of blades.

The overal torque and thrust forces are then computed by numerically integrating the $\delta T, \delta Q$ along the rotor span.

3.4 Wind turbine mechanical loads

Structural loads are known to be a major factor in wind turbines failure. Structural loads lead to significant fatigue damage which shortens the wind turbine operational life. The operation and maintenance (O&M) cost of wind turbines are directly related to the wind turbine components' failure as the result of structural loads. Therefore, structural loads directly affect the levelised cost of production. According to [14] the O&M cost of a wind turbine is 20%-35% of the total levelised cost per MWh production over the lifetime of the turbine.

Wind turbine and wind farm control can be used to alleviate the structural loads and reduce the fatigue damage on wind turbine components. Therefore, it is essential to learn about the source of these loads and include them in the wind turbine and farm models.

Interaction between the rotating rotor and the non-uniform wind field induces unbalanced loads on the wind turbine structure. Variability in the rotor wind flow is the source of fluctuations in the aerodynamic lift and drag forces acting on the rotor blades; consequently, the source of fluctuations in the rotor torque and thrust. These fluctuations induce stress cycles with different amplitudes on the wind turbine structure which accordingly result in fatigue loading and component failure. Wind turbine structural modes are sensitive to these fluctuations and excitation of these modes induce fatigue loads [15].

Moreover, structural loads are, also, related to the wind turbine's controller actions. The wind turbine control objective is to regulate the generator speed in below rated and maintain the generator power in above rated conditions. Perturbations in the wind-field may result in a change in the wind turbine operating point which can cause a change in the generator speed and aerodynamic forces. The change in aerodynamic forces can induce forces and bending moments through the rotor to the drive-train components and the tower.

Generally, the structural loads are divided into fatigue loads and extreme loads. The extreme loads are due to high amplitude loading on the wind turbine structural components, which may occur from sudden shut down of the machine as a result of a faulty condition, loss of grid connection, etc. The extreme loads can cause immediate damage to the structural components and lead to a turbine failure [16].

Fatigue loads, on the other hand, arise from repetitive stress cycles on the wind turbine structural components.

The cyclic loads result from interaction between the rotating rotor and the stochastic wind field, gravity, wind shear, tower shadow, and etc. These loads are known as the $n\,\Omega_0$ loads due to the rotational sampling of the wind-field by the rotor blades at rotational speed Ω_0.

The $n\,\Omega_0$ loads consist of stochastic and deterministic components [17].

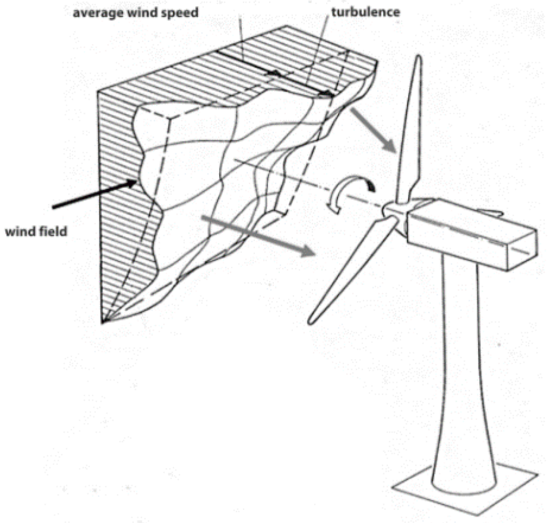


Figure 3-4: Interaction between rotor and stochastic wind field [18]

The stochastic component is due to the turbulence in the wind. As the rotor blades sample the turbulent wind-field, each blade cuts through a complex field of mixed eddies which results in unbalanced loading on the blades and the rotor. The sampling of the stochastic wind-field is shown in Figure 3-4.

The deterministic component is mainly due to the tower shadow and wind shear [19]. Wind shear is defined as the variation of wind speed with respect to altitude, which is caused by the friction between the air flow and the ground surface [13]. Hence, a given point on a rotor blade experiences different wind speeds at different altitudes as the rotor rotates. The wind shear effect is shown in Figure 3-5 and is modelled by

$$\bar{U}(z) \propto \ln\left(\frac{z}{z_0}\right) \tag{3-30}$$

where, $\bar{U}$ is mean wind speed, z_0 is the surface roughness length, and z is height above ground [13].

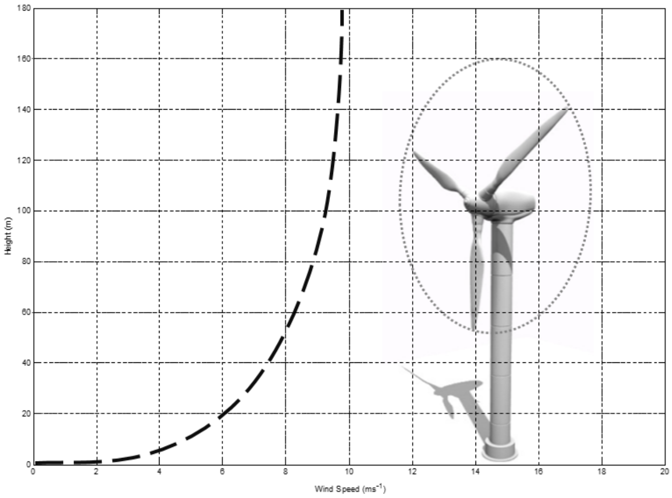


Figure 3-5: Illustration of wind shear

Tower shadow effect is the reduction in flow velocity in the vicinity of the tower due to the blockage of the air flow by the tower as shown in Figure 3-6. Therefore, each rotor blade experiences a lower wind speed in the vicinity of the tower as the rotor rotates. Tower shadow is also a major source of cyclic loads [13].

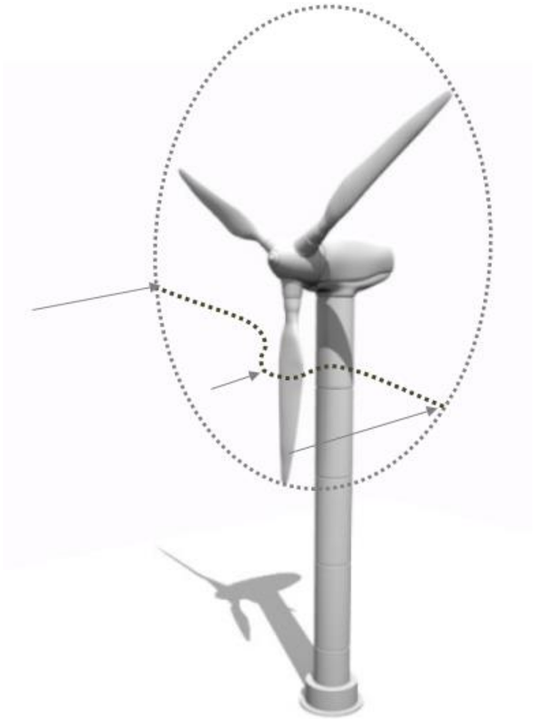


Figure 3-6: Tower shadow effect

3.5 Blade loads

Blade loads are due to the forces and bending moments acting at the blade root. The blade root bending moments are defined in four directions: out-of-plane, in-plane, spanwise, and torsion. For control design purposes mainly the in-plane and the out-of-plane bending moments are considered in the analysis.

One of the sources of the out-of-plane (flap-wise) bending moment is the aerodynamic forces acting on the blades. These forces cause the blades to move out of the plane of rotation.

The in-plane (edge-wise) bending moment arises from tangential wind force, and blade gravity [20].

The out-of-plane bending moment is the main factor of the fatigue loading on the blades.

3.6 Tower loads

Wind turbine tower loads can arise from aerodynamic forces acting on the rotor, and wind turbine controller actions. The tower fatigue loads can be expressed in terms of the tower side-side bending moment and the tower fore-aft bending moment.

The tower fore-aft bending moment is mainly due to rotor thrust loading and is the main cause for the tower fatigue loading. Wind turbine control actions could also cause the for-aft bending moment.

Unlike the fore-aft direction, the rotor provides negligible aerodynamic damping in the side-side direction. One of the main source of the side-side bending moment is the rotor torque acting on the gearbox and generator mounting. These forces at the top of the tower, induce bending moments at the tower base [21]. Generally, the magnitude of the tower fore-aft bending moment is higher than the side-side bending moment.

3.7 Drive train loads

The interaction between the rotor and turbulent wind-field can induce imbalanced forces through the rotor into the drive-train. Drive train modes are lightly damped and excitation of the first drive-train mode can results in torsional vibration and consequently fatigue loads in drive-train components [16].

3.8 Structural load reduction via control design

As mentioned earlier, excitation of the structural modes results in the fatigue loads. Therefore, it is important to consider the most significant structural modes during controller design to prevent their excitation. Normally, the full envelope controller of a variable speed HAWT are augmented with additional control loops to alleviate structural loads. Namely, tower feedback loop acting on the tower top acceleration adjusts the pitch demand signal to effectively reduce the tower fore-aft moment; and a drive train filter acting on the generator speed adjusts the torque demand to reduce the drive train loads.

In [22], a coordinated control approach was adopted to coordinate the pitch and torque control actions at frequencies close to the first tower mode frequency to avoid its excitation.

Individual pitch controllers can also be employed to effectively reduce the rotor blade loads [17]. These control strategies can, mostly, be implemented in the wind turbine full envelop controller to reduce the asymmetric loading, such as loading due to the wind shear.

3.9 Wind turbines aerodynamic interaction

As wind turbines convert the kinetic energy of the wind into mechanical power the wind-field downstream of their rotors undergoes significant changes which can be characterised by reduced velocity and increase turbulence level. These conditions downstream of an operating turbine are known as the wake effects. As mentioned earlier, wind turbines in a wind farm can experience wake effects from their neighbouring turbines. It is known that the performance of the wind turbines operating in the wake is lower than those outside the wake. Therefore, the total performance of the wind turbines can be affected as a result of wake effects.

In order to mitigate the effect of aerodynamic interaction between the wind turbines and reduce the power losses, it is essential to have a good understanding of the wind turbine's wake behaviour. Therefore, a wind-field model that includes an appropriate representation of the wake through the wind farm is required. Various wake modelling techniques for simulations are developed in the literature. An overview of the wake modelling techniques is provided here to underline the most effective wake modelling method for wind farm modelling and controller design.

3.10 Wake modelling for simulation models

Wake interaction between wind turbines can be mitigated by dynamically coordinating the power set-point of wind turbines through a wind farm controller[23]. Therefore, it is desirable to include a suitable wake interaction model in the wind farm model.

The wake properties such as direction, magnitude and diameter are dependent on the local atmospheric conditions such as, surface roughness, wind direction, and the wind farm layout. The wake effects observed at a point in a wind farm can either originate from a single wind turbine or a combination of multiple wakes from a number of turbines[24]. As a result, the wake effects can be modelled as a single wake or multiple wakes.

The wake area downwind of a wind turbine rotor is divided into the near wake and the far wake regions[25]. The near wake is the region just after the rotor disc to approximately one rotor diameter downstream. The flow properties in this region are significantly determined from the rotor properties[26]. The far wake is the region far downstream where the effect of the rotor on the flow field is not dominant and the wake is completely developed. In the far wake region, the velocity deficit is dependent on the turbulent mixing and the level of turbulence in the freestream [27].

Typically for normal atmospheric conditions, the wake far downstream of a wind turbine recovers at around 10 to 15 rotor diameters. The higher the turbulent intensity in the ambient wind-field the faster the wake downstream recovers[28][29]. However higher turbulent wind results in higher fatigue loading on the wind turbines.

The far wake region is particularly of interest for modelling the single and multiple wakes for wind farm simulation and control design. In [30], existing wake models were classified as field models, kinematic models, and roughness element models.

3.11 Field models (implicit)

A very good review on the field models is given in [31],[32]. Field models are generally developed by solving the Rynolds-averaged Navier-Stocks (RANS) equations for incompressible flow with eddy viscosity closure to calculate the magnitude of the flow at every point in the wind field [26][33].

UPMWAKE is one of the most well-known RANS field wake models which is developed in [34] to describe a single wake. Later a modification to UPWAKE was undertaken to consider multiple wakes in the modelling [34].

EVMOD, another example of the numerical single wake models, is developed in [35]. This model is also established on the basis of RANS equations. In this model, the wake profile is assumed to be axisymmetric, and wind shear and ground effect are ignored in the calculations. Similarly, Garrad-Hassan Ltd in UK developed the EVFARM commercial code for multiple wakes based on the EVMOD wake model [36].

Wake field models in general provide a very detailed representation of the flow field at the expense of intensive computational effort and complexity. Therefore, these models are not suitable for wind farm simulation and control design.

3.12 Kinematic models (explicit)

These models are generally developed based on the classic momentum conservation theory and assume axisymmetric, self-similar velocity deficit and turbulent intensity for the far wake profile. The kinematic models are mainly used to describe wake effects at the far wake region. A good review of the well-known kinematic wake models is given in [37].

The most commonly used kinematic models are the Jenson Park model[38], the Larsen wake model[29],[39], and the Frandsen analytical model [40].

Compared with the implicit field models, the kinematic analytical wake models usually require less computational effort, and are also simple enough for wind farm optimisation and control design purposes.

Amongst them, the Frandsen wake model is the most commonly used for analytical wake modelling in the literature. In [41], the Frandsen model is evaluated against real data from Horns Rev wind farm, as shown in Figure 3-7.

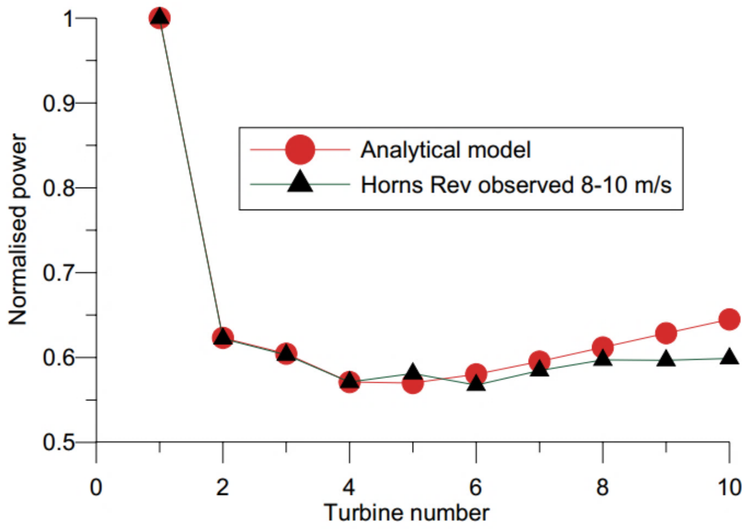


Figure 3-7: Wake velocity deficits [41]

As observed, the model results are in a good agreement with the real data under below rated conditions.

3.13 Roughness element models

Roughness element models are mainly developed for predicting the overall effects of large wind farms on the wind field. In these models wind turbines are considered as distributed roughness elements in the wind-field that modify the ambient flow field [42], [43], [44].

3.14 Available wind farm simulation models and control approaches in the literature

As wind energy is becoming more dominant in the electricity market, greater control flexibility will be required to address the technical and economic issues regarding variability and unpredictability of wind power [45]. Moreover, wind power plants are being subjected to greater regulation to provide more flexible power output control. As mentioned earlier, in order to control wind turbines in a more intelligent manner, the wind farm control system must have access to the individual wind turbines operating status and be able to adjust their power set-points dynamically. Therefore, in order to design a wind farm controller, an appropriate wind farm model is required that offers dynamic power set-point adjustment capability and includes realistic wind farm conditions within the simulations.

In the literature, there are many contributions to wind turbine and wind farm modelling for power system control and stability analysis from the electrical and power electronics point of view [46],[47],[48], [49]. However, not all of these models are relevant or can be directly used for the simulation and wind farm control design with the objectives of power/ load optimisation. Since structural loads on a wind turbine depend on both the wind-field/rotor interaction and the turbine full envelope controller, the simulation models must include the most significant turbine structural modes, the full envelop controller as well as a representation of the wind-field and wind turbines aerodynamic interaction.

A few relevant available simulation models, contain little information about the wind turbine structural modes and the wind turbine controller dynamics [50]. Most of these models were developed for wind farm layout optimisation problems [51]. Therefore, in this section a literature review is presented to underline the relevant wind farm models available for controller design within the objectives specified in this thesis.

The wind farm model complexity varies depending on the model and control objectives. The complexity of simulation models can arise from the fidelity level of the wind turbine dynamics and the wind-field models incorporated in the wind farm modelling. High fidelity wind farm models can become very stiff and difficult to simulate. Generally, stiff systems require an appropriate implicit variable-step solver[52]. On the other hand, low fidelity models offer less information about the system dynamics, which makes them less useful for the purpose of controller design and structural load analysis[53].

SOWFA is a well-known high-fidelity wind farm simulator, which was developed by NREL using computational fluid dynamics and the NREL wind turbine simulator FAST [54]. SOWFA requires extensive computational capacity and power to run simulations.

In the following, the available research literature on the wind farm modelling and control can be divided into model-free and model-based control approaches [55].

3.15 Model-free wind farm control methods

The model-free approaches mostly consider wind farm optimisation problems based on static relationships of wind turbine dynamics and wind-field interactions in the farm. The objectives for these optimisation problems are mainly defined as the wind farm layout optimisation, power maximisation and fatigue load reduction. Normally, these static relationships are defined in terms of axial induction factors or thrust coefficients. The aerodynamic interaction between wind turbines are also included as quasi-static aerodynamic interaction models. The most commonly used quasi-static model is the PARK model developed by Jensen et al which is normally used in wind farm array efficiency studies [38].

Since the wind speed is mapped to the rotor power and thrust as a function of thrust coefficient C_T, the decision variables of the optimisation objective functions are usually defined as either the actual induction factor, a, or thrust coefficient, C_T. Optimisation problems are then solved numerically to obtain a set of joint optimal decision variables in order to achieve the best objective function and satisfy the constraints.

The cost function for model-free optimisation problems are generally defined using the following equations

$$V_i(a_i) = U_\infty\big(1 - \delta(a_i)\big) \tag{3-31}$$

$$P_i(a_i) = \frac{1}{2}\rho\pi A_i C_p(a_i) V_i(a_i)^3 \tag{3-32}$$

$$C_{p_i}(a_i) = 4a_i(1 - a_i)^2 \tag{3-33}$$

$$C_{T_i} = 4a_i(1 - a_i) \tag{3-34}$$

$$C_{P_i} = \frac{C_{T_i}\big(1 + \sqrt{1 - C_{T_i}}\big)}{2} \tag{3-35}$$

$$a_i = \frac{1}{2}(1 - \sqrt{1 - C_{T_i}}) \tag{3-36}$$

where, the subscript i is the index for the i-th turbine, $P_i, C_{pi}, C_{Ti}, \delta(a_i)$, and a_i represent power, power coefficient, thrust coefficient, velocity deficit, and the axial induction factor at turbine i.

The general form of the wind farm optimisation problems used in many of the model-free methods is defined as follows

$$\underset{(CT_1 \dots CT_n)}{max} \quad \sum_{i=1}^{N} P_i(V_i, C_{Ti}) \tag{3-37}$$

$$subject\ to \begin{cases} 0 \le P_i \le P_{rated} \\ \vdots \\ \omega_{min} \le \omega_i \le \omega_{rated} \end{cases} \tag{3-38}$$

where, N is the total number of wind turbines, $\omega_{min}, \omega_{rated}$ represent minimum and rated generator speed, respectively.

Various numerical optimisation solvers can be used to obtain a set of joint axial induction factors or thrust coefficients, satisfying the cost function and constraints of the optimization problem, including game theoretic methods, consensus methods, collaborative extremum seeking, etc.

In [56], a cooperative control approach was adopted to optimise the wind farm power output, in which a local control policy (objective function) was designed to obtain a set of axial induction factors that maximise the total power production. A method called Safe Experimentation Dynamics, originally developed for payoff based learning algorithms in[57], is used to solve the optimisation problem. In the Safe Experiment Dynamics method a discrete set of decision variables can be obtained based on the previous axial induction factor for each turbine and the total power output of the wind farm at the previous iteration. This method is also called model-free optimisation with communication.

In [58], a broad survey of cooperative control and game theoretic approaches for wind farm optimisation is presented. The wind farm is modelled on the basis of a quasi-steady wake model, wake interaction model and wind turbines power model.

Similarly, in [59],[60],[61], and [62], different game theoretic approaches for model-free control problems are proposed to optimise the wind farm objective functions.

In [59], a wind farm with eight wind turbines aligned in a row parallel to the wind direction is considered. Likewise, static equations for the optimisation problems are derived from a combination of wind turbine power model and a wake model developed by Brand [63]. The power model is obtained as a function of rotor speed from the 5MW NREL wind turbine model utilising C_p tables. The optimisation problem is defined to obtain a set of rotor speeds for wind turbines to maximise the total power output subjected to the generator power and rotor speed limits.

An overview of game theory, distributed coordination and cooperative control is given in [64],[65],[66]. Generally speaking, cooperative control problems are defined as a game where the players can coordinate their strategies to accomplish a global objective. Players can agree on how to divide the total payoff (reference power) between them based on their individual status and external conditions (available wind power). Hence, the optimisation method seeks for a set of joint axial induction factors that maximises the total farm power output.

In [67],[68],[69] an Extremum Seeking Control (ESC) approach is utilised to optimise the wind farm power output. ESC is an online search for an optimal control input to maximise or minimise a time varying cost function. In [67] an ESC approach is employed to maximise the total power production with respect to a set of optimal axial induction factors. It is claimed that, the ESC algorithm improves the total power output of the farm. For different simulations at 8m/s wind speed with 0%, 2% and 4% turbulent intensities, the total power output increased levels of 4.1%, 3.8% and 1.1% are reported, respectively, compared with the normal control operation of the wind turbines.

An overview of ESC is given in [70]. This optimisation control approach can be used to obtain a rough estimate of wind turbines' power set-points required for power maximisation in real time simulations.

Since model-free approaches are mostly based on the static wind farm equations while the wind turbine dynamics are not adequately considered in the problems, these control approaches and wind farm models are not sufficient for wind farm simulation and control design in the real practice.

3.16 Model-based wind farm control methods

Generally, wind turbine dynamics can be modelled utilising a number of lumped-parameter models, Newton's Law of motion, and Lagrange equations [71]. The mathematical representation of the system maps the inputs of the system to its states and outputs. A precise mathematical representation of a wind turbine must include various complex dynamics of the physical system. Therefore, the order of ordinary differential equations could be high with the level of dynamics incorporated in the modelling. However, wind turbine models can be simplified for control design purposes, by only considering the most significant dynamics required for controller design.

One of the most commonly used wind turbine model for wind farm modelling and controller design is the NREL 5MW model developed by the National Renewable Energy Laboratory (NREL) [71].

Spudic et.al. in [72], developed a wind farm model based on the NREL 5MW wind turbine model and utilised a hierarchical control approach to optimise the power production and structural loads. The wind farm controller is designed for the above rated operating regions of the wind turbines. The wind farm is operated in a curtailed operating mode to provide enough head room for power adjustments. The wind farm reference power is assumed to be less than the available power in the wind farm.

In [73], a lumped-parameter method is used to obtain a linearised state-space wind turbine model. Likewise, C_P, C_T look-up tables are used in the modelling, and a receding horizon controller is developed to reduce the wind turbines loads. The receding horizon control method adopted in this research requires the time history of the wind data for its estimation process. Therefore, the Taylor's frozen turbulence hypothesis is assumed to obtain the time history of the wind speeds in front of the turbines. However, Taylor's hypothesis assumption is not adequate for wind farm control design. Moreover, wind turbine controller dynamics and aerodynamic interaction between the turbines are not considered. Thus, load analysis cannot be conducted appropriately with this model.

In [74], a wind farm optimisation control strategy is used to minimise the fatigue loads on the turbines whilst tracking the reference power. The proposed control algorithm is then tested on a wind farm model of 6 wind turbines. The wind farm model was developed in Aeolus SimWindFarm toolbox. The Aeolus toolbox utilises the NREL 5MW wind turbine model dynamics. Yet again, the wind turbines power control mode was only assumed to be feasible in above rated operating region of the turbines.

In [75], a distributed control algorithm was used to develop a wind farm controller with the objectives of power reference tracking and fatigue load reduction. Similarly, the Aeolus toolbox is used for the modelling of a 10 NREL 5MW wind turbine for simulations. The wind farm controller is developed in a number of sections and divided between the wind turbines. Therefore, each wind turbine can locally determine its own power set-point based on the measurement data from a number of neighbouring wind turbines. A cost function is then defined to optimise the wind turbines operation.

In [76], a model predictive control approach is proposed for maximising the power output of a wind farm, considering the wake effects in the simulations. The Horns Rev wind farm layout is considered with 500m spacing between the turbines. It is assumed that the wind turbines in a row are not interacting with the neighbouring rows. Therefore, only wake effects from the wind turbines in a single row are considered. A single row of 10 turbines is developed in the Aeolus toolbox to represent one row of the Horns Rev Wind farm. The Horns Rev Vestas V80 2MW turbines are replaced with the 5MW NREL models in Aeolus. A wind farm optimisation problem is formulated to compute the power set-points for each turbine in order to maximise the power production. In addition, a model predictive controller (MPC) is utilised at each wind turbine to track the computed power set-points from the optimisation. A 1% total power increase was reported for 1000s simulation at 10m/s mean wind speed

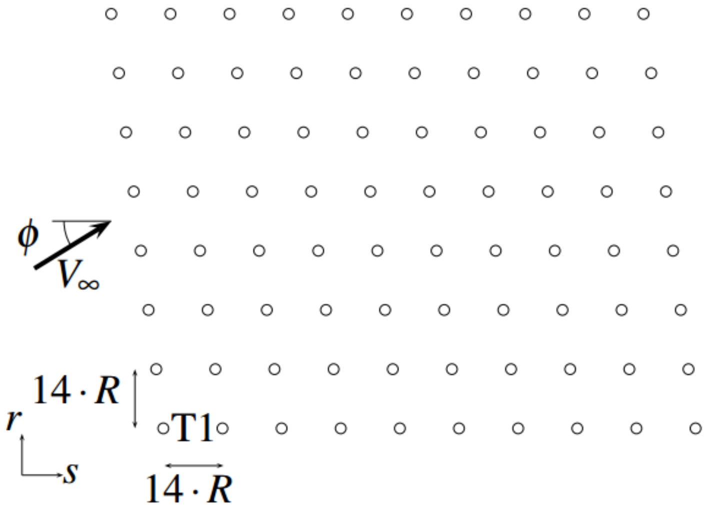


Figure 3-8: Horns Rev layout [76]

Since most of the model-based wind farm control methods were developed based on the NREL 5MW wind turbine model and the Aeolus SimWindFarm toolbox [77], a brief review of these models is provided here.

3.17 Aeolus SimWinFarm toolbox & NREL 5MW wind turbine model review

The SimWindFarm is a Matlab/Simulink based toolbox which was developed on the Aeolus project to provide a fast simulation environment for wind farm control design. The SimWinFarm toolbox creates a wind farm model based on a simplified 5MW NREL wind turbine model. The simplified 5MW NREL model is a aeroelastic model developed based on C_p/C_t lookup tables and consists of a simple 3rd order drive train model, 1st order generator model, 2nd order pitch actuator, and a 2nd order tower dynamics model [78]. Figure 3-9 illustrates the 5MW NREL model subsystems.

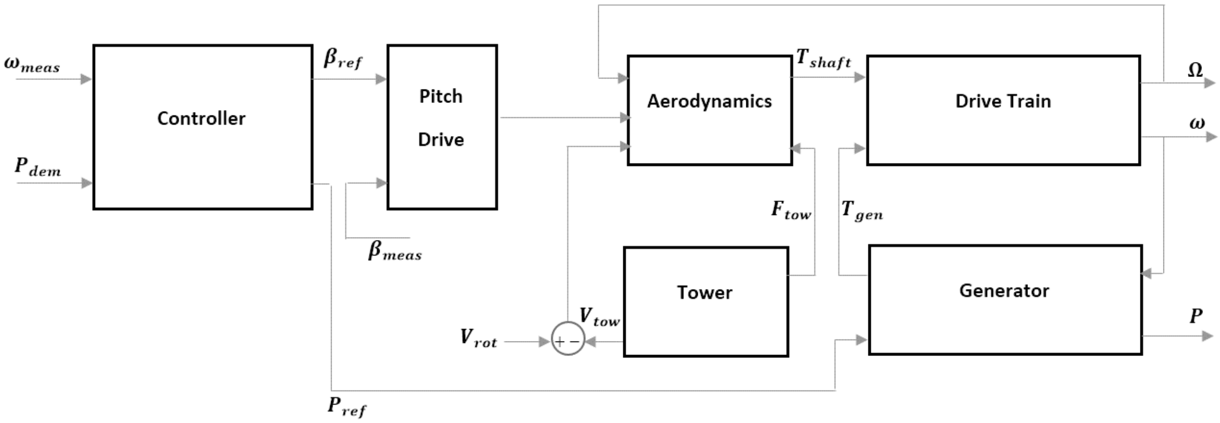


Figure 3-9, NREL 5MW [79]

The pitch drive contains a second order system representing a hydraulic pitch actuator model with a time delay from input to pitch rate. A proportional controller controls the actuator to reduce the error between reference and measured pitch.

The mathematical representation of the pitch drive is [77]:

$$\begin{gathered}\ddot{\beta} = \frac{1}{\tau}(U_\beta - \dot{\beta}) \\ U_\beta = K(\beta_{ref} - \beta_{measured})\end{gathered} \tag{3-39}$$

where, U_β is the output of the proportional controller into the pitch actuator, β_{ref} is pitch demands, and $\beta_{measured}$ is measured pitch.

The aerodynamics are modelled on the basis of the static torque and thrust equations utilising look-up tables[77].

$$\begin{gathered} T_{shaft} = \frac{1}{2}\rho A V_{rot}^3 C_p(\lambda,\beta)/\Omega \\ F_{tow} = \frac{1}{2}\rho A V_{rot}^2 C_T(\lambda,\beta) \end{gathered} \tag{3-40}$$

where, T_{shaft} mechanical torque, F_{tow}, is the thrust force, Ω is rotor speed, A is swept area of the rotor, ρ is the air density, $C_T(\lambda,\beta)$ is the thrust coefficient, λ,β are top speed ratio and the pitch angle respectively

The drive-train model is based on a 3-state model including generator speed, rotor speed, and drive-train torsional spring force state.

$$\begin{gathered} \dot{\Omega} = \frac{1}{I_{rot}}\left(T_{shaft} - B_{shaft}\dot{\phi} - K_{shaft}\phi\right) \\ \dot{\omega} = \tfrac{1}{I_{gen}}\left(-T_{gen} + \tfrac{1}{N}\left(B_{shaft}\dot{\phi} - K_{shaft}\phi\right)\right) \\ \dot{\phi} = \Omega - \frac{1}{N}\omega \end{gathered} \tag{3-41}$$

where, $\omega \; and \; \Omega$ represent generator and rotor speed, $B_{shaft} \; and \; K_{shaft}$ represent viscous friction and torsional spring constant, respectively, $T_{shaft} \; and \; T_{gen}$ represent main shaft and generator torque, respectively, and $I_{rot} \; and \; I_{gen}$ represent rotor and generator inertias, respectively.

The generator model is represented by

$$\dot{T}_{gen} = \frac{1}{\tau_{gen}}\left(\frac{P_{ref}}{\omega} - T_{gen}\right) \tag{3-42}$$

where, P_{ref} is reference power and τ_{gen} is the generator time constant

The tower model is represented by a second order system

$$\ddot{Z} = \frac{1}{m_{tow}}\left(F_{tow} - B_{tow}\dot{Z} - K_{tow}Z\right) \tag{3-43}$$

where, Z represents the tower deflection, m_{tow} is the tower mass, $B_{tow}\ and\ K_{tow}$ represents damping and spring constant in the model

The NREL 5 MW baseline wind turbine model includes a generator torque controller and a simple collective blade-pitch controller. The generator torque controller acts on a filtered generator speed to maximise the power output at below rated wind speeds. The collective pitch controller also acts on the filtered generator speed to maintain the generator speed at it's rated, in above rated operation. The control strategy of the baseline controllers are described by NREL in [71].

The generator torque controller is designed based on the control strategy defined for the wind turbine model. There are 5 operating regions defined for the torque controller. The main objective of the torque controller is to maximise the power production in below rated and to keep the generator torque at its rated in above rated. The generator torque demand is zero before cut-in wind speed allowing rotor to accelerate. At the cut-in wind speed, the generator torque is increasing linearly with the generator speed as shown in Figure 3-10. The slope of the leaner line in this region is defined by

$$slope\ 1^{1/2} = \frac{(K_{opt}\omega_1^2)}{\omega_1 - \omega_{cut-in}} \tag{3-44}$$

where, K_{opt} is the optimal gain for the CP_{max} tracking curve, ω_1 is the first generator constant speed, and ω_{cut-in} is the cut-in generator speed

Once the generator torque is reached to the initial value of the generator torque on the CP_{max} tracking region, the generator torque increases with the generator speed by

$$Torque\ demand = K_{opt}\omega^2 \tag{3-45}$$

Similarly, at the end of the CP_{max} tracking curve a linear transient is applied to move from power maximisation region to the above rated region. The slope of the linear curve is

$$slope\ 2^{1/2}\frac{T_{reted}}{\omega_{rated}-\dfrac{\omega_{reted}}{1+0.01*\%\ generator\ slip}} \tag{3-46}$$

where, T_{reted} is the rated generator torque, ω_{reted} is the rated generator speed, and $\%\ generator\ slip$ is the percentage of the generator slip

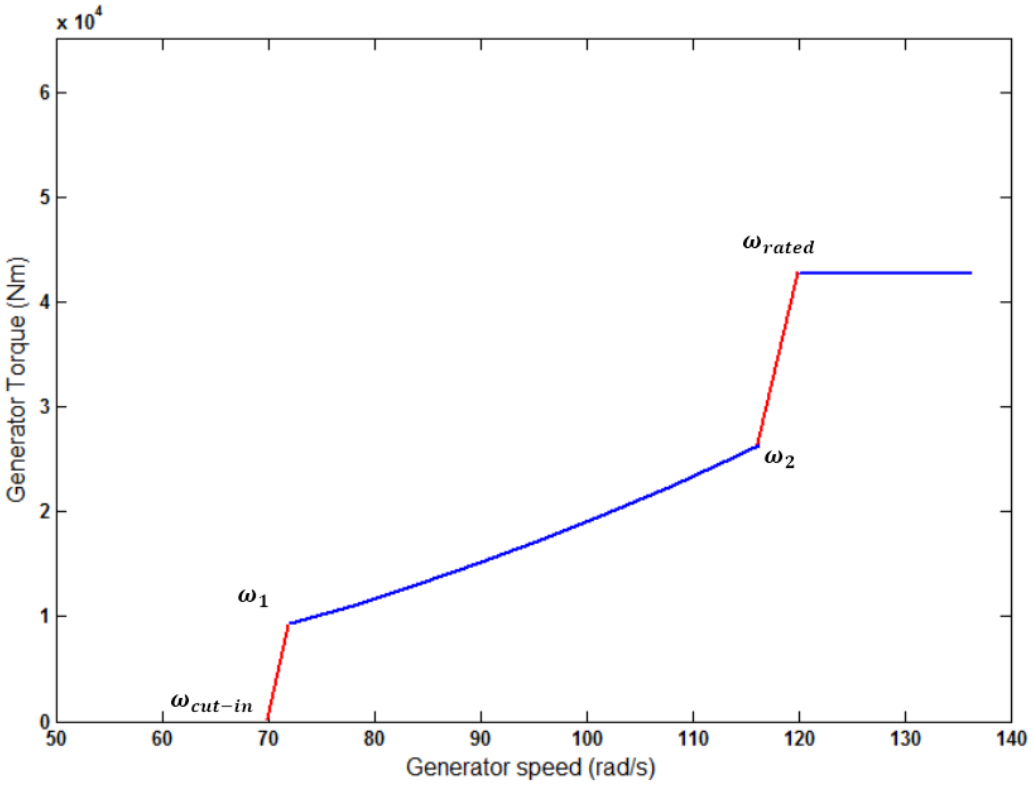


Figure 3-10, NREL 5MW control strategy [71]

The above rated controller is a multi-input multi-output controller and includes generator torque controller and a collective pitch controller. The main goal of the above rated controller is to maintain the generator speed at its rated value. A gain scheduled PI controller is utilised to eliminate the generator speed error from its rated level. The generator torque is inversely proportional to the generator speed in above rated wind speeds.

The controller is also able to adjust the power output of the wind turbine in above rated operations. The controller computes the generator torque demand based on the reference power and the measured generator speed by

$$T_{demand} = \frac{\mathrm{P_{ref}}}{\omega_{gen}^{filtered}} \tag{3-47}$$

where, T_{demand} is torque demand, $\mathrm{P_{ref}}$, is reference power, and $\omega_{gen}^{filtered}$ is the filtered generator speed.

The maximum power is limited to the rated generator power.

The switching between operating modes of the controller is implemented by utilising a switching logic with three conditions

$$\begin{cases} P_{dem} \leq T_{opt} * \omega_{gen} \\ \omega_{gen} \geq \omega_{rated} \\ \beta_{ref}(t-1) > 0 \end{cases} \tag{3-48}$$

where, T_{opt} is the optimal torque, and β_{ref} is the reference pitch angle

If any of the above conditions occur, the wind turbine controller switches to the power reference tracking mode.

The SimWinFarm toolbox utilises a wind-flow model similar to the one developed in [80]. The wake effects are also adopted from the Frandsen analytical wake model [81]. Namely, wake diameter, wake centre, and wake deficit are modelled on the basis of C_T coefficients of the wind turbines as a function of distance downwind.

In order to evaluate the ability of the toolbox for wind farm control design, a simple case scenario of 5 wind turbines positioned in a row aligned with the wind direction with 800m distance between them is considered. A number of simulations are performed at mean wind speeds of $8ms^{-1}, 14ms^{-1}$ for a period of $1000s$.

In the below rated scenario at $8ms^{-1}$ wind speed, the aim is to evaluate the effectiveness of the model for investigating the wake interaction between the wind turbines. The idea is to switch off one of the upstream wind turbines in the farm and investigate the effect of wake interaction between the remaining turbines.

Since the mean wind speed is $8ms^{-1}$ and the distance between the neighbouring turbines is $800m$ it takes $400s$ for the wake to completely evolve through the wind farm. Figure 3-11 to Figure 3-13 illustrate the wind speeds, wake deficits and the power outputs of each individual wind turbine.

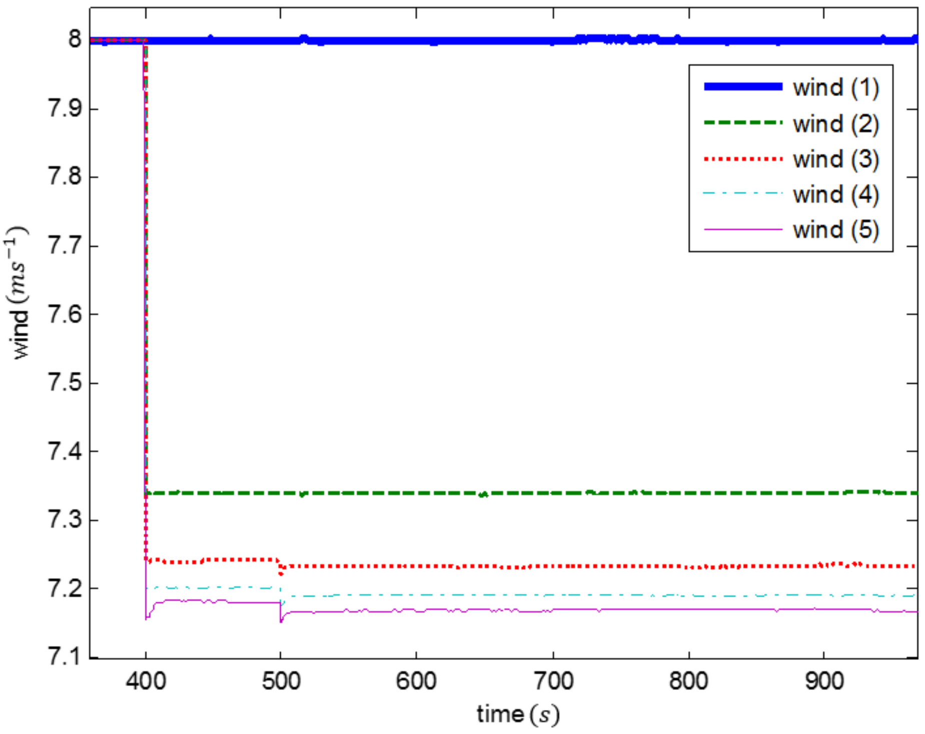


Figure 3-11: Wind speed time profile for each turbine

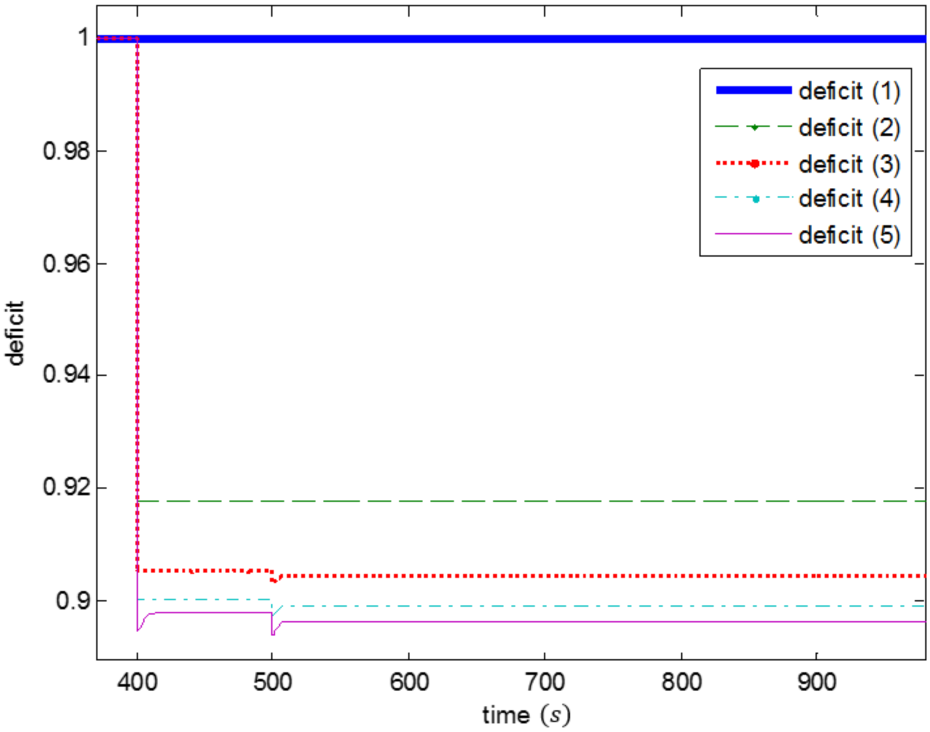


Figure 3-12: Wake deficit time profiles for the 5 turbines

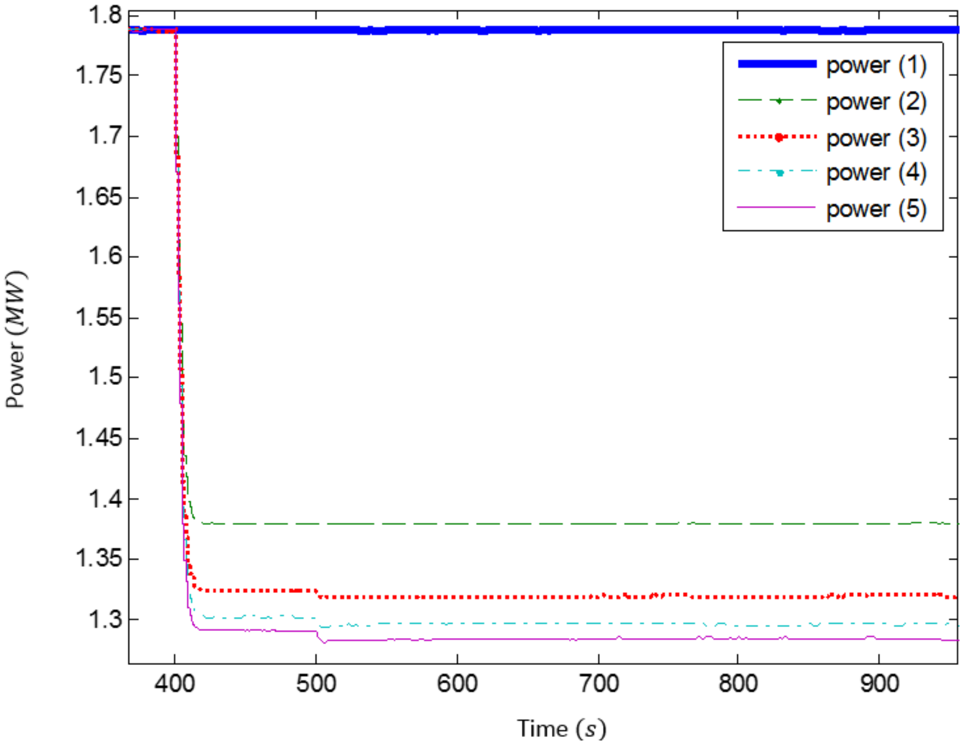


Figure 3-13: Power outputs for the 5 wind turbines

As can be observed, the wake deficits come into effect when the wake of the first turbine has reached the end of the farm at 400 seconds. After this initial transient, the effect of upstream wind turbines wake on the wind speed and power output of the downstream wind turbine in the wake can be observed. Since the wake is fully evolved, at 400 seconds all the turbines must experience a combined wake deficit from the upwind neighbouring turbines. Therefore, WT2 must experience wake deficit from WT1 and WT3 must see a combined wake deficit from WT1 and WT2. A similar pattern must be observed for the rest of the turbines in the row. However, this is not the case in this simulation. At 400 seconds wind turbines in the wake only experience a single wake effect from their upstream turbine. For instance, WT3 only sees the wake effect from WT2, and WT4 only experiences the wake from WT3. After another 100 seconds, at 500seconds WT2 is still experiencing a wake from WT1, and WT3, WT4, WT5 are experiencing a combination of wake effects from two upstream neighbouring turbines. For instance at this time, WT3 experiences a combination of wake effects from WT1, WT2, and WT5 sees a combination of wakes from WT3, WT4. Therefore, it takes another 300 second for the wake from WT1 to reach WT5 at the end of the farm. At this point, the wake effects are fully evolved through the wind farm. Figure 3-14 illustrates the time that takes for the wake from WT1 to reach each wind turbine in its wake after the initial 400seconds transient.

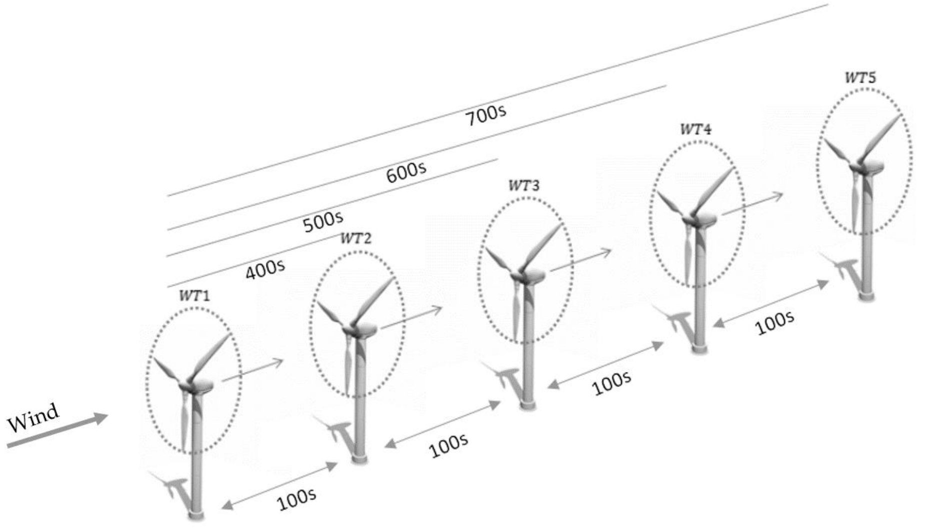


Figure 3-14: Wake effect from WT1 seen by each turbine after initial transient

The initial concern arises from the way that the wake deficit transport delays are considered in the wake modelling. Therefore, another simulation is performed at this wind speed to investigate the effect of wind turbines power set-point change on the wake conditions downstream of the turbine and also on the other turbines performance.

Since in this case WT1 encloses all other turbines in its wake, a simulation is conducted for a shutdown case of WT1 after its wake fully evolved through the wind-field at 750 seconds.

Figure 3-15 to Figure 3-17 illustrate the wind speeds, wake deficits, and the power outputs of each wind turbine.

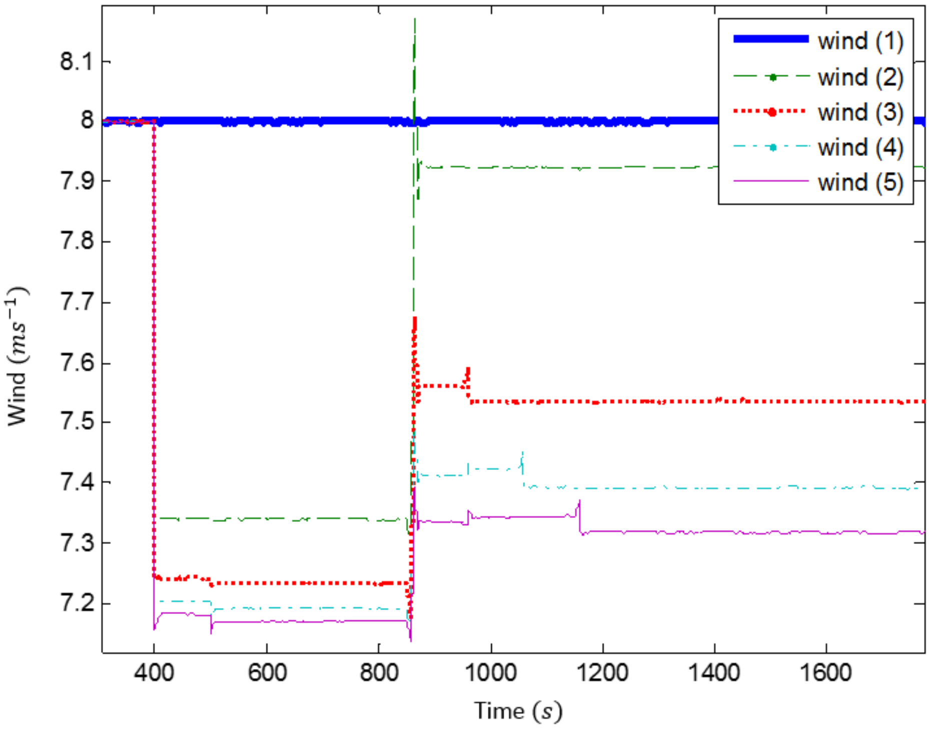


Figure 3-15: Wind speeds for the shutdown case

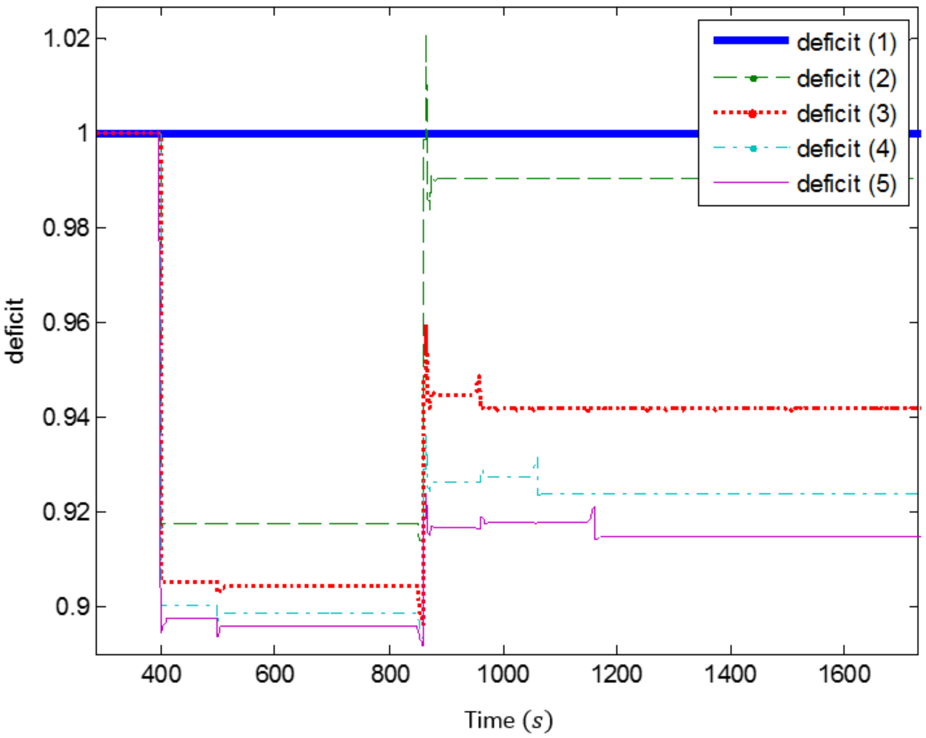


Figure 3-16: Deficits for the shutdown case

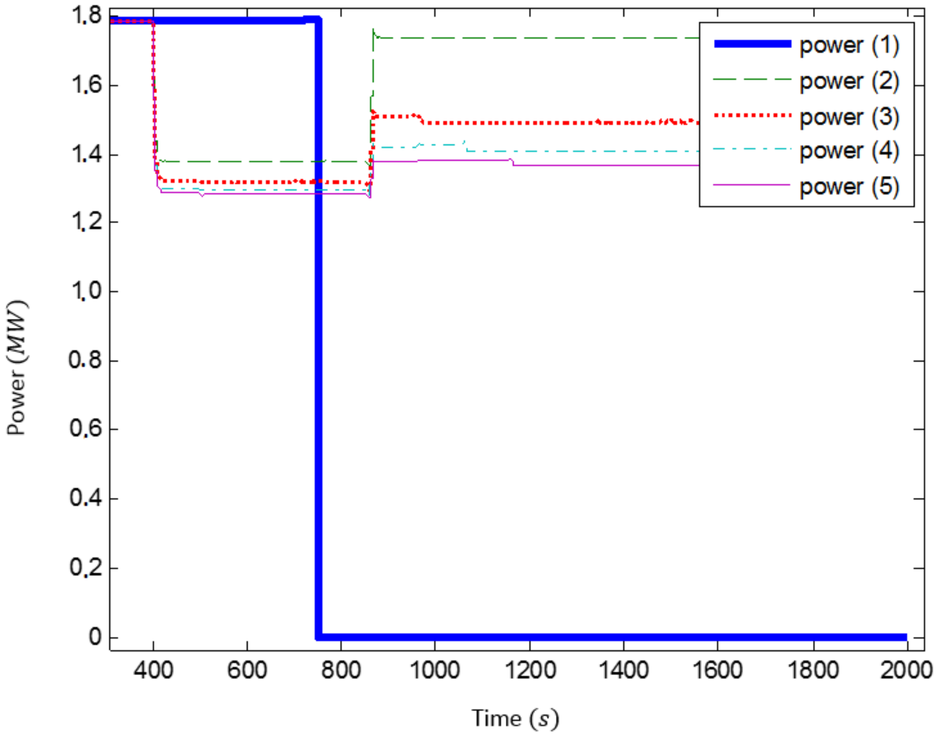


Figure 3-17: Power profiles for the 5 turbines in the shutdown case

Again, the initial wake effects are observed to come into effect at $400sec$, and completely evolved at 700 seconds. At 750 seconds WT1 power set-point is reduced to zero to investigate the effect of this power set-point change on the other wind turbine performances. Since the initial wake transient condition is already passed between the 400-700 seconds period of simulation and the wind turbines are experiencing a fully evolved wake from all neighbouring upwind turbines, the effect of this power set-point change is expected to be observed sequentially on the other wind turbines in a 100 seconds intervals as shown in Figure 3-18. For instance, WT2 must experience an increase in its wind speed and consequently its power output at 850 seconds, and WT3 must experience the effect of this change at $950sec$ and so on. However, the graph displays that this is not the case. Instead, all the wind turbines in the wake experience changes in their wake deficits 100s later, after the power set-point change at WT1, at the same time.

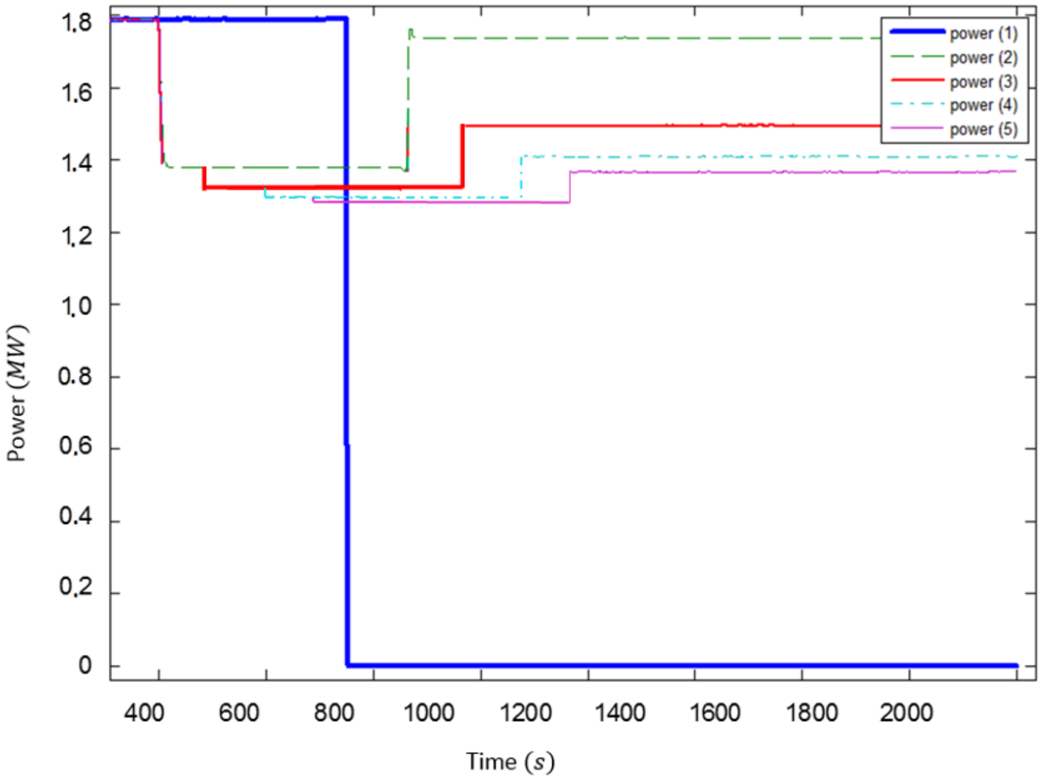


Figure 3-18: Expected power output after power set-point change at WT1

To investigate the behaviour of the wind farm in above rated conditions, a number of simulations are performed at $14ms^{-1}$ for a normal operating condition of the wind farm and a shutdown case of the first wind turbine similar to the below rated scenario. As Figure 3-19 and Figure 3-20 show, similarly to the below rated operation, the wake propagation is considered in a quasi-static manner.

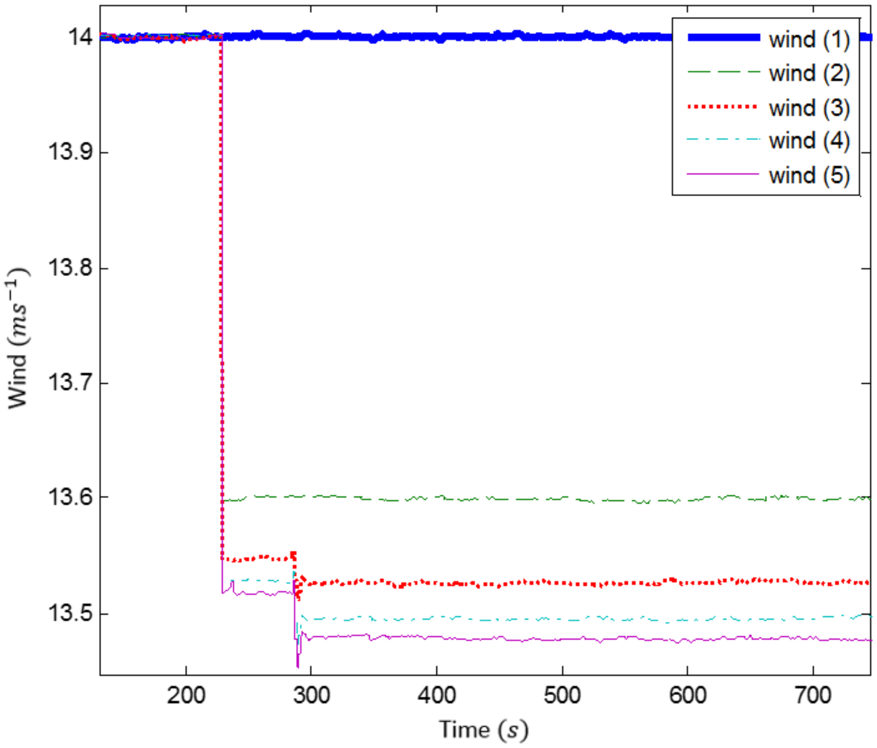


Figure 3-19: Wind speeds plus wake deficits at each turbine

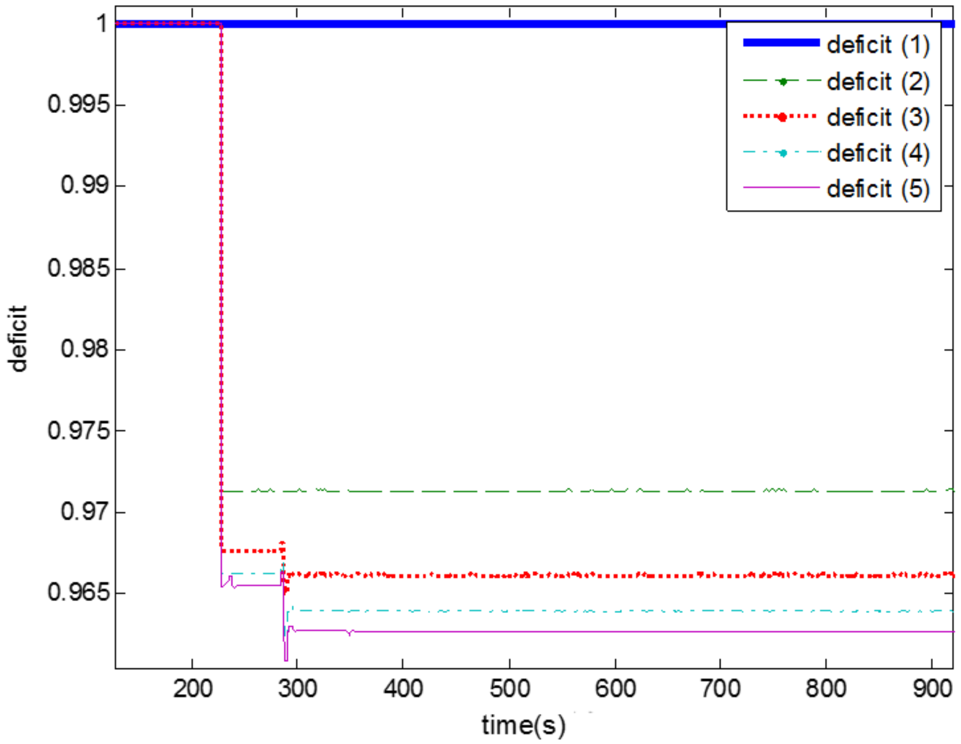


Figure 3-20: Wake deficits at each turbine

Since the effect of power set-point adjustment of each wind turbine on their downstream turbines is the key factor in the wind farm optimisation, another simulation is performed to investigate the effect of a shutdown case of the first wind turbine on the other turbines' performance. Similar to the below rated scenario, Figure 3-21 to Figure 3-23 show that the effect of this power set-point adjustment cannot be observed as expected on the wake deficits.

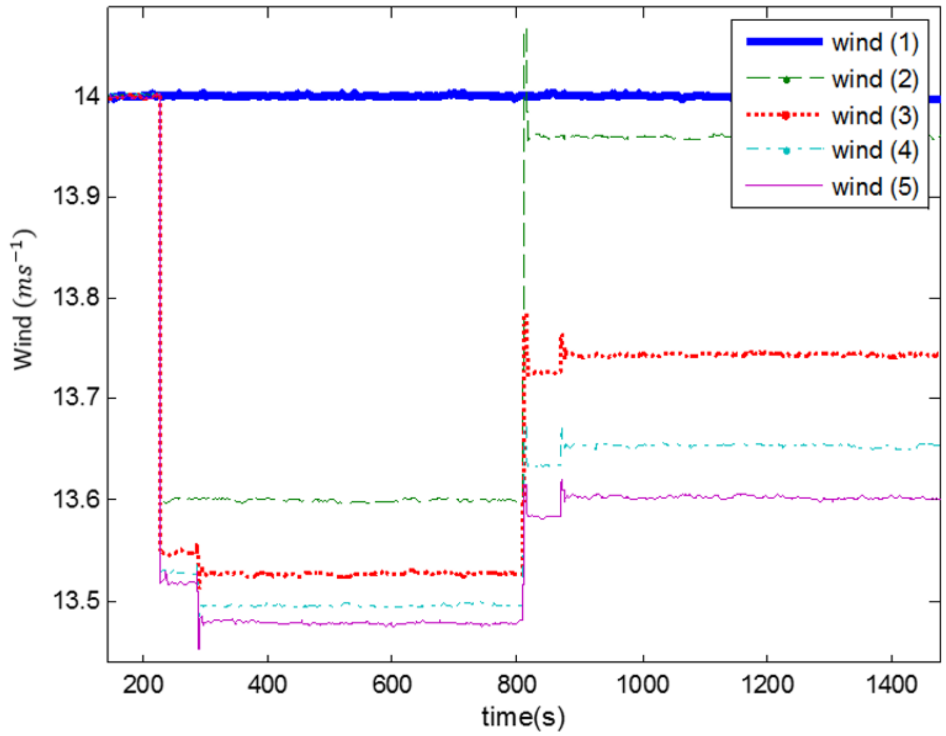


Figure 3-21: Wind speeds at shutdown case

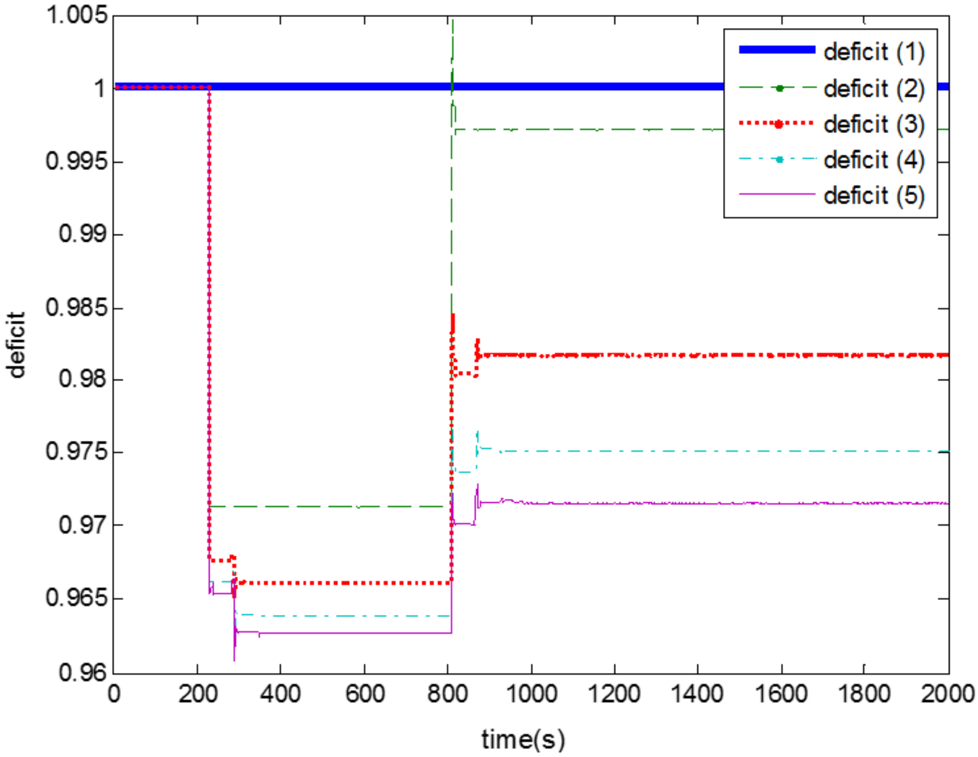


Figure 3-22: Deficits at shutdown case

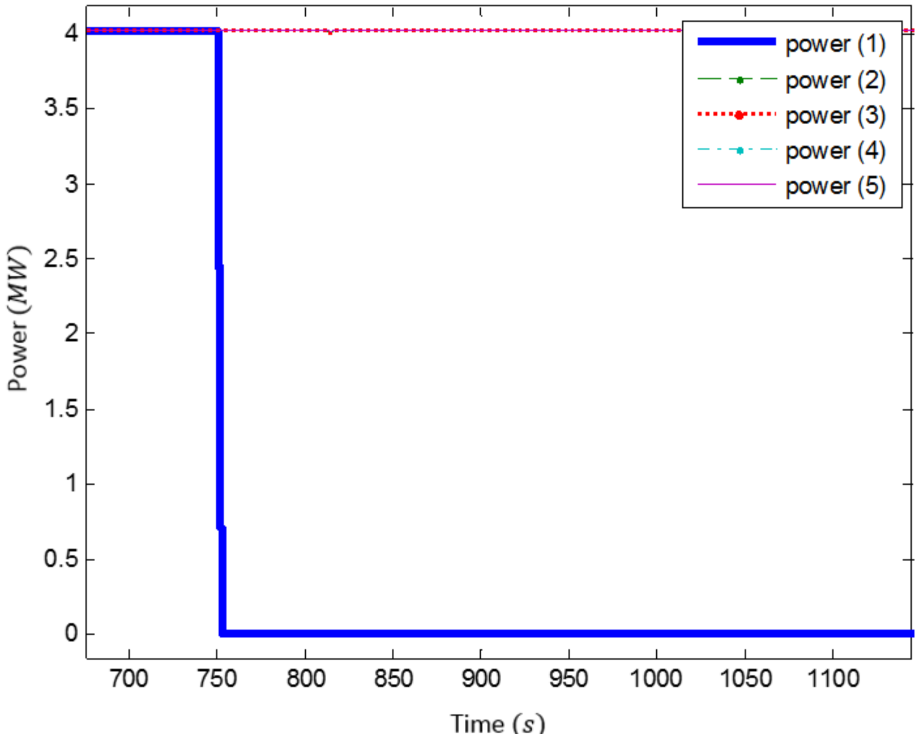


Figure 3-23: Power outputs at shutdown case

The Aeolus toolbox was developed for fast simulation and wind farm controller design purposes. However, simulation results suggest that this model may not be adequate for investigating the wind farm performance under different operating conditions and control scenarios. Since the wake effects are not modelled in a dynamic sense and the wake propagation through the wind farm is not accurately represented, the performance of the wind farm controllers cannot be accurately evaluated.

Moreover, the NREL 5MW wind turbine model utilised in the toolbox is largely simplified. Therefore, investigating the operating performance of the wind turbines under different wind farm control algorithms is limited.

The power control mode of the wind turbine controller, is only applicable for the cases where the available power is higher than the reference power. Therefore, the wind farm controller design with the objective of power reference tracking is only valid for these operating conditions.

Additionally, replacing the NREL 5MW model with a more sophisticated wind turbine model, requires careful implementation of the user defined wind turbine inputs and outputs to match the existing model.

Since most of the models available in the literature are not suitable for this research, a new wind farm model needs to be developed and implemented in Simulink with adequate properties included for wind farm controller design.

3.18 Concluding remarks

In this chapter, an extensive literature review is provided which outlines the relevant wind turbine and wind farm models suitable for wind farm controller design for the control objectives defined in this thesis. The most commonly used wind turbine and wind farm models, the NREL 5MW wind turbine model and the Aeolus SimWindFarm model, are closely examined to evaluate their effectiveness for wind farm controller design.

The Aeolus SimWinFarm model is also investigated by considering different wind farm control and operating scenarios. It is shown that this model has clear limitations for wind farm controller design. Therefore, in Chapter 4 a suitable wind turbine model with relevant dynamics and control functionalities for wind farm model construction is presented.

Chapter 4: 5MW Supergen wind turbine, dynamic model and controllers

IN this chapter the Supergen 5MW Exemplar wind turbine model and its controllers, which have the relevant fidelity level for fast simulation and controller design, are described. The model is a non-linear Simulink model with parameters chosen to correspond to the physical parameters of a 5MW wind turbine [82]. The model only includes the most significant structural dynamics relevant to controller design and thus, it is less complex in comparison to the models in full aero-elastic packages.

The relevant wind turbine dynamic modes included for controller design are two blade modes, two tower modes, actuator dynamics, and a simplified drive-train model. A nonlinear model of the rotor/wind-field dynamic interaction and a full envelope controller are, also, included, so that the wind turbine model is appropriate for gain-scheduling analysis and design. The full envelope controller and its implementation in C++ are fully equivalent to those installed on commercial wind turbines. The full envelope controller is augmented by a Power Adjusting Controller (PAC) designed to dynamically adjust the set-points of the turbine [83].

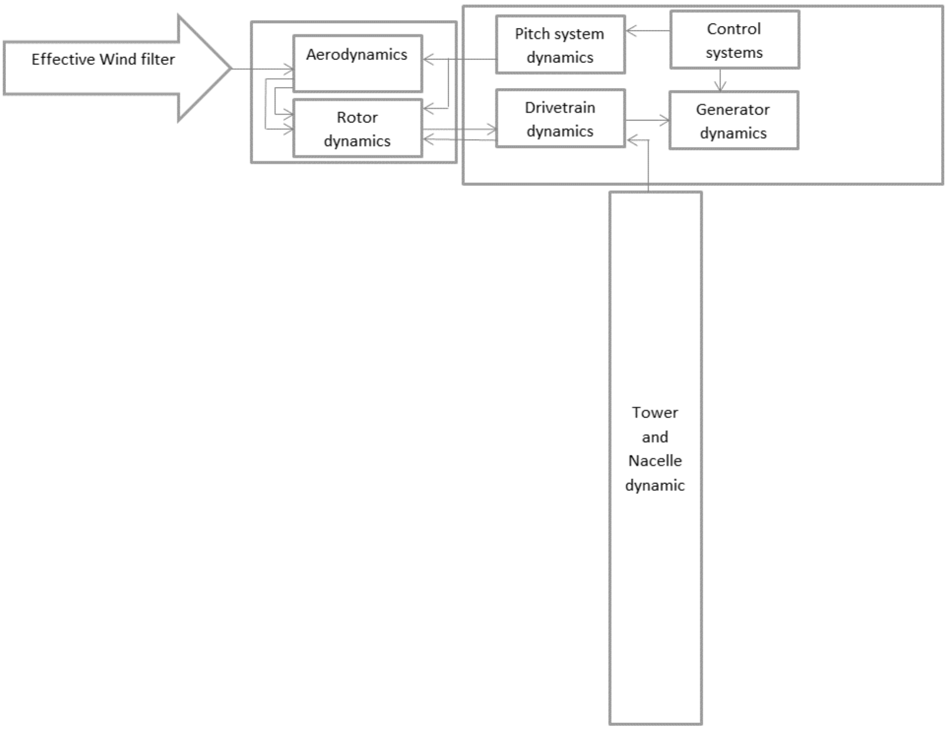


Figure 4-1: Model blocks for the 5MW Suppergen Wind turbine model

Figure 4-1 depicts the different wind turbine structure and dynamic components incorporated in the modelling.

4.1 Rotor wind-field

The wind turbine model utilises a spatial filter to generate the rotor effective wind speed for simulations. The spatial filter was developed in [84], [85] for simulation and control design in Simulink.

The spatial filter transfer function for the effective wind speed is

$$v_{effective} = \frac{\frac{1}{\sigma}\left(s + \frac{\sqrt{2}}{\sigma}\right)}{\left(s + \frac{\sqrt{2}}{\sqrt{a'}\sigma}\right)\left(s + \frac{\sqrt{a'}}{\sigma}\right)} v_{point} \quad \textbf{(4-1)}$$

$$\sigma = \frac{\gamma R}{\hat{v}} \quad \textbf{(4-2)}$$

where, γ is the turbulent decay factor, $a' = 0.55$, R is the rotor radius, and $\hat{v}$ is the point wind speed averaged over time this is

$$\hat{v} = \frac{1}{\tau s + 1} v_{point} \qquad \textbf{(4-3)}$$

Where, τ is the filter time constant

An example of filtered $10ms^{-1}$ wind speed with 10% turbulent intensity is shown in Figure 4-2 and Figure 4-3.

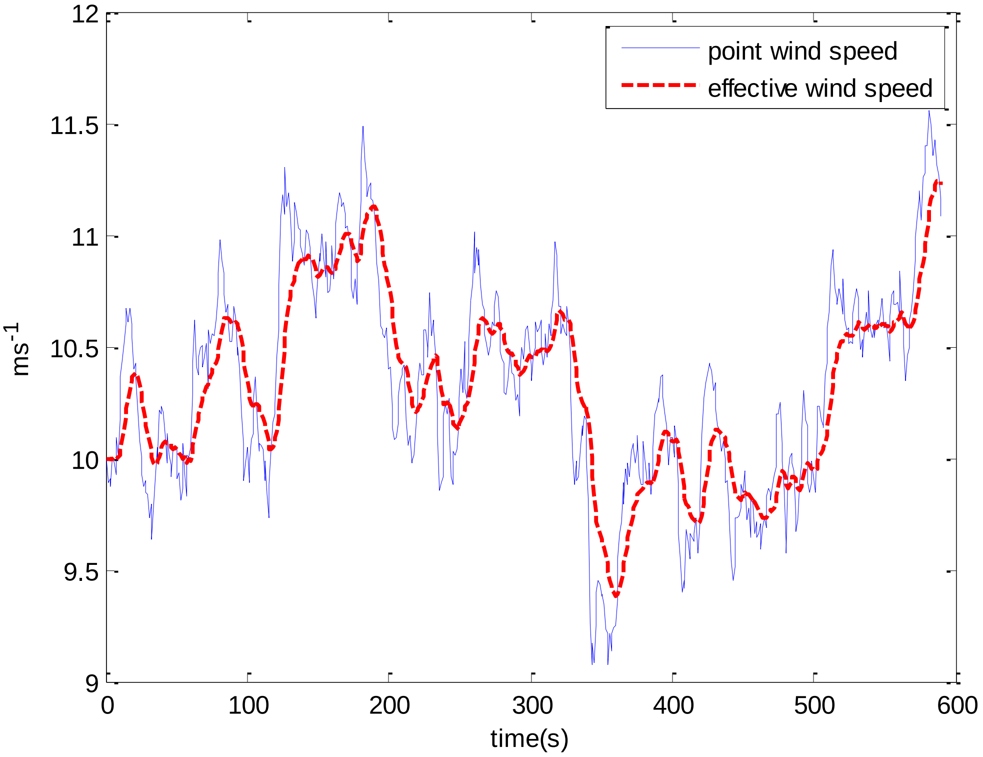


Figure 4-2: Point wind speed and filtered effective wind speed

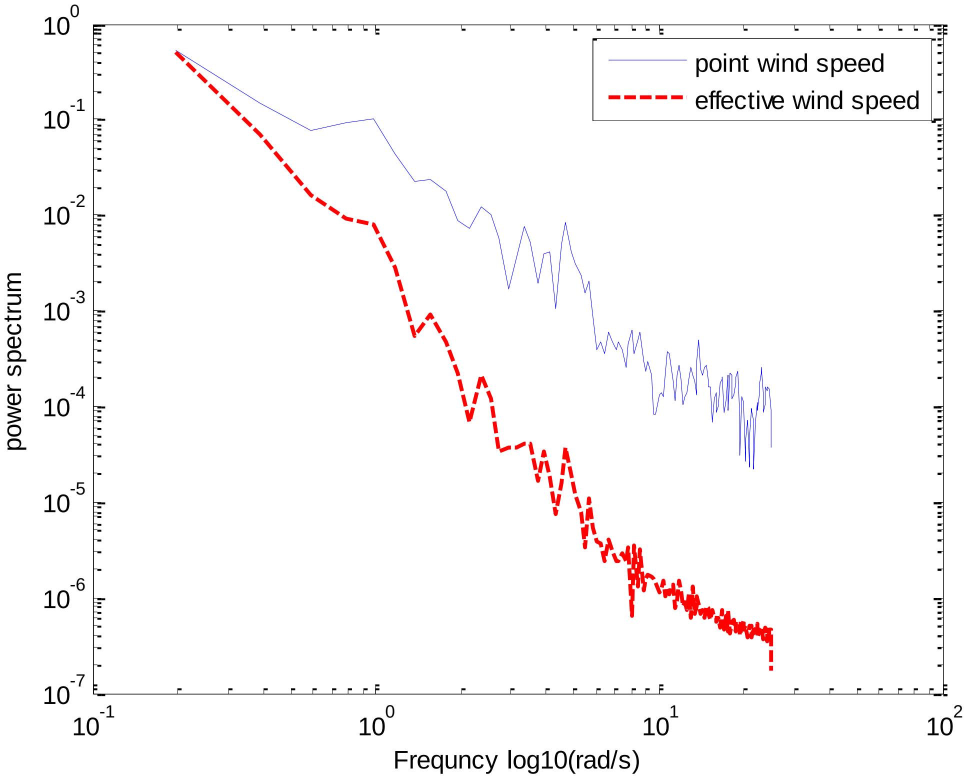


Figure 4-3: Spectrum of point wind speed and filtered effective wind speed

The wind-field, as seen by the rotor blades, induce cyclic $n\Omega$ loads, Ω is the rotor speed per revolution, as the blades sweep through it over each revolution. These $n\Omega$ loads consist of deterministic components, due to tower shadow and wind shear, and stochastic components, due to wind field turbulence. These cyclic loads result in unbalanced rotor loads which affect all wind turbine components from the rotor blades through the drive train to the tower and foundations. It is known that the most significant peaks occur at $n\Omega$, $\mathrm{n} = 1,2,\cdots$[85]. Therefore, to better represent the structural loads due to the rotor/wind-field interaction, for a 3-bladed HAWT, the 3Ω & 6Ω cyclic torque oscillations are also included in the model.

The 3Ω and 6Ω torque oscillations are modelled as explained in [85] by

$$L_{n\Omega_0}(\mathrm{t}) = \varepsilon_1 \cos(n\Omega_0 t) + \varepsilon_2 \sin(n\Omega_0 t) \tag{4-4}$$

where, $L_{n\Omega_0}$ is torque oscillation, Ω_0 is the rotor angular speed, $n = 1,2,3$, t is time, and $\varepsilon_{1,2}$ are independent coloured noise defined by

$$\dot{\varepsilon_1} = -a_{n\Omega}\varepsilon_1 + b_{n\Omega}w_1 \tag{4-5}$$

$$\dot{\varepsilon_2} = -a_{n\Omega}\varepsilon_1 + b_{n\Omega}w_2 \tag{4-6}$$

where, $w_{1,2}$ are White Gaussian noise

The spectral peaks roll-off at high frequencies, is ensured by filtering $L_{n\Omega_0}$ by a first order filter as

$$h(s) = \frac{1.25}{\frac{s}{n\Omega} + 1.25} \tag{4-7}$$

where, $n = 3, 6$ for a 3-bladed HAWT

The 3Ω and 6Ω torque oscillations can be seen in the generator and rotor speed power spectrum plots in Figure 4-4 and Figure 4-5. The PSD plots are generated from simulation results obtained at 14 ms^{-1} wind speed with 10% turbulence intensity in Simulink.

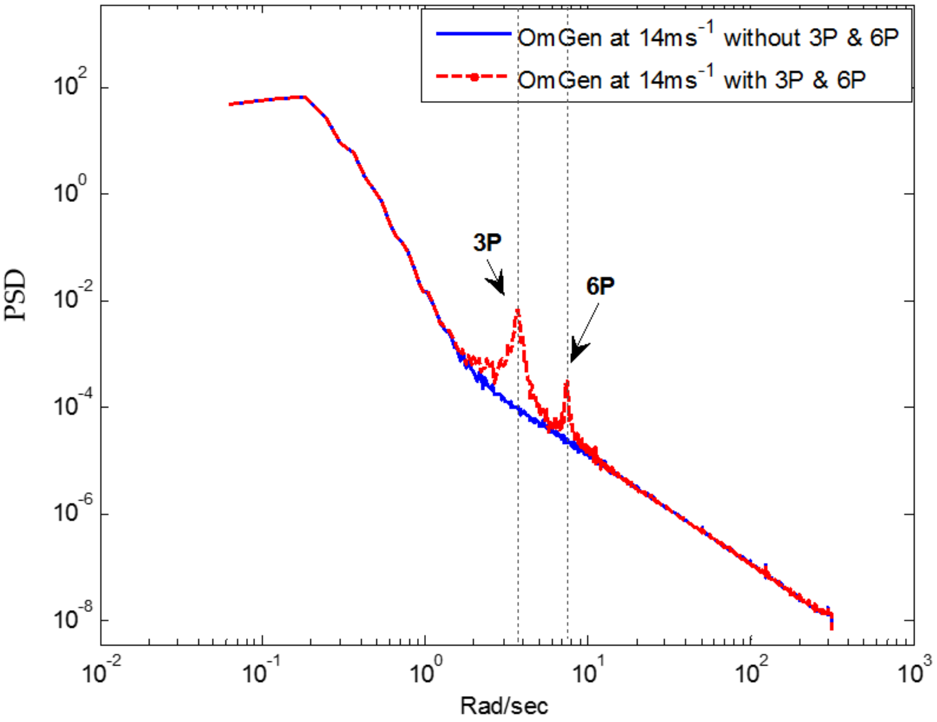


Figure 4-4: Power spectrum of generator speed at 14ms^{-1} with and without 3Ω, 6Ω peaks

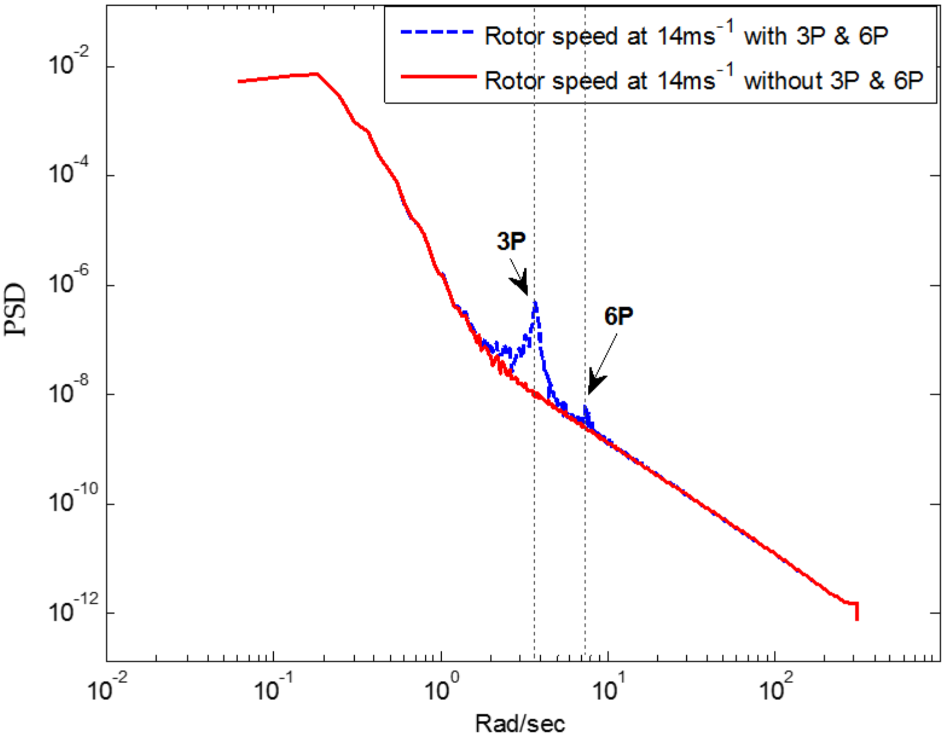


Figure 4-5: Power spectrum of rotor speed at 14ms^{-1} with and without 3Ω, 6Ω peaks

4.2 Rotor, tower, and drive-train dynamics

Rotor dynamics are included in the form of the blade symmetric edge-wise and flap-wise modes, where the blades are oscillating in phase at the same frequency. The edge-wise and flap-wise modes of the rotor are coupled with the first drive train mode and the first tower fore-aft mode respectively.

Two tower modes are included in the wind turbine model, which are of particular interest for controller design. The coupling between the tower mode and the drive-train mode introduces a pair of complex right half plane zeroes (RHPZs) in the dynamics of the turbine. The linkage between the generator speed and the blade pitch angle, as a results of these RHPZs, impose limitations on regulating the generator speed through a blade pitch controller in above rated.

As the wind turbines grow in size, the frequencies of these RHPZs reduce. Therefore, designing speed control loop with acceptable performance becomes a challenging task. The frequency of the tower modes strongly depend on the wind turbine size. As the wind turbine size increases the tower modes' frequency tend to reduce and put further limitations on the generator speed control loop.

The tower fore-aft, tower side-side and the first drive-train modes are evident in the power spectrums in Figure 4-6, Figure 4-7, and Figure 4-8. These spectrums are generated based on the results obtained from simulations at $14ms^{-1}$ wind speed with 10% turbulence intensity in Simulink.

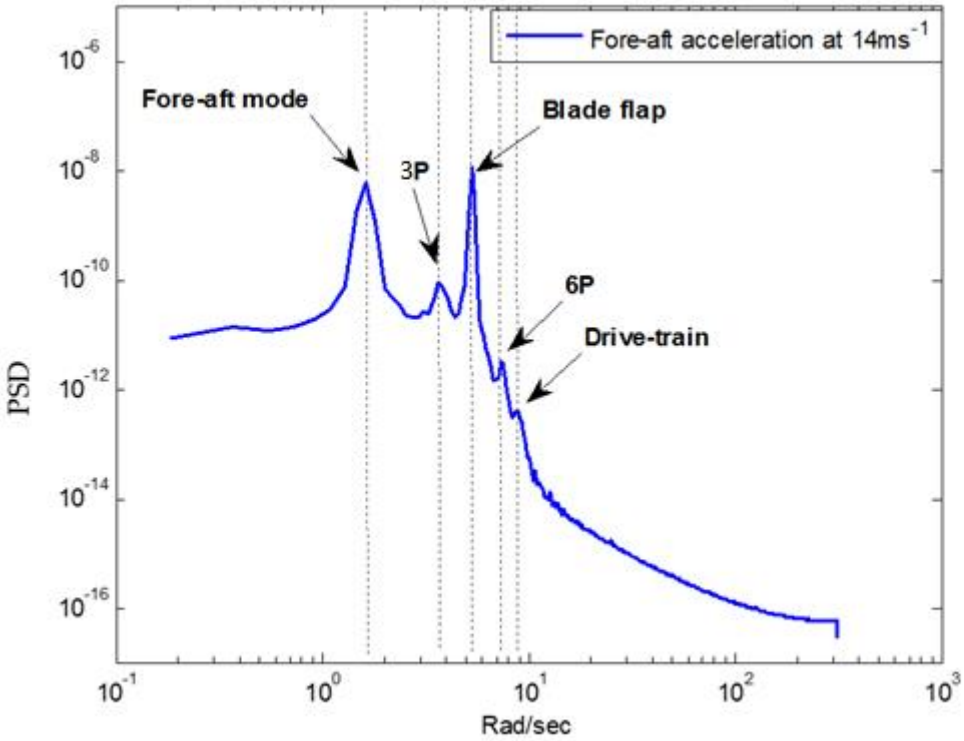


Figure 4-6: Power spectrum of the tower fore-aft acceleration

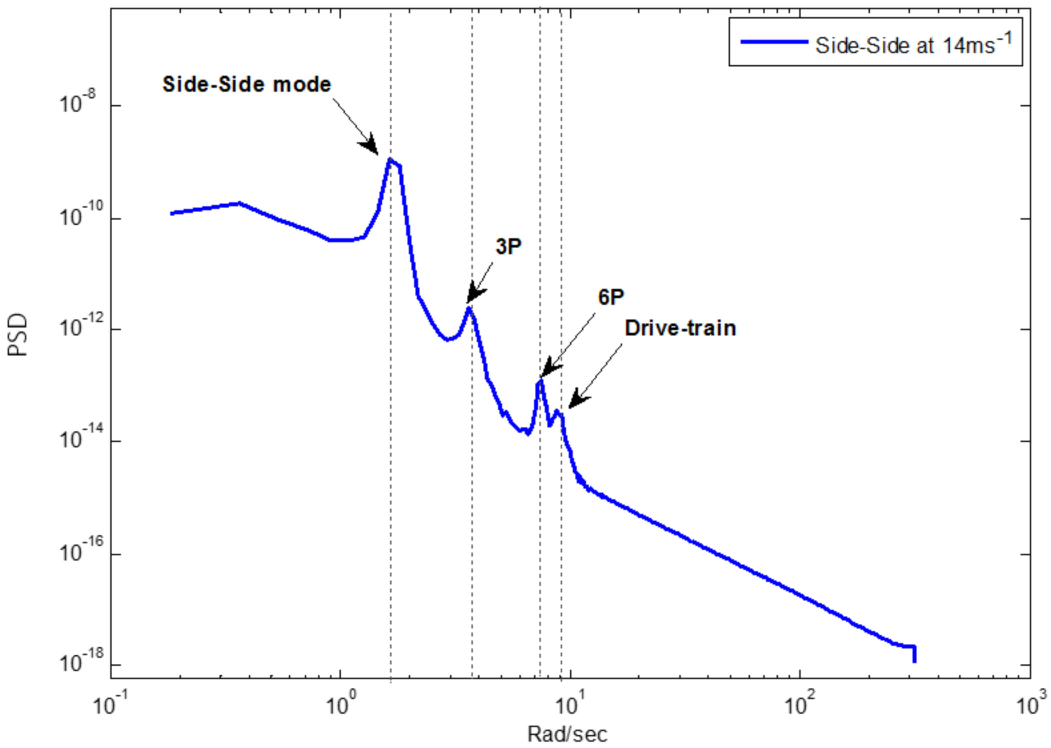


Figure 4-7: Power spectrum of the Tower side-side dynamics

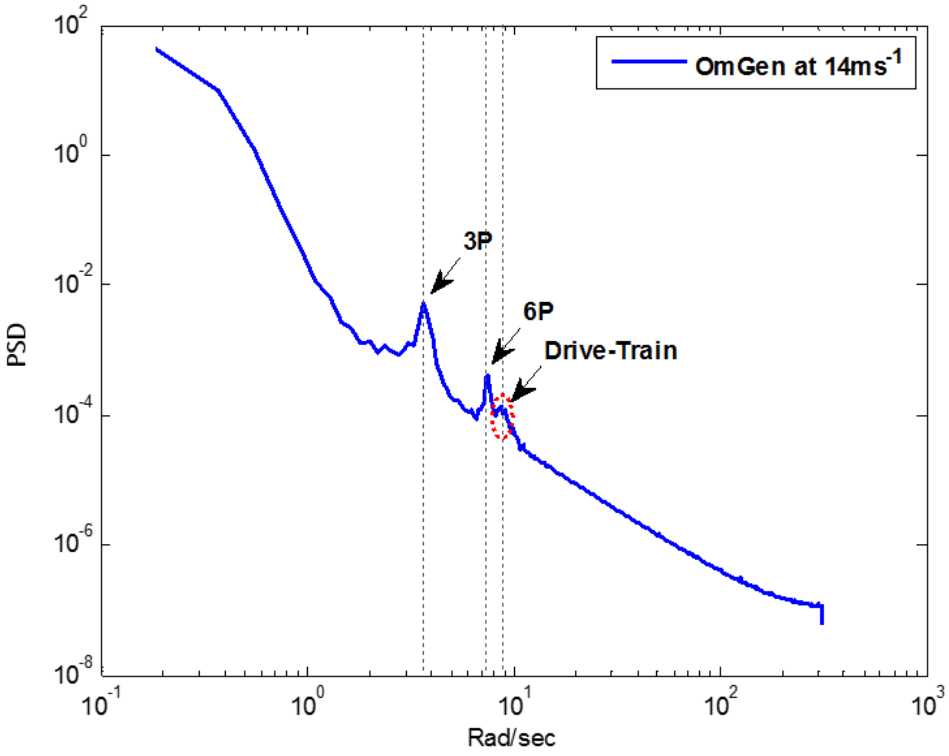


Figure 4-8: power spectrum of generator speed

4.3 Wind turbine full envelope controller

The full envelope controller of the 5MW Supergen wind turbine consists of necessary control loops for optimising the wind turbine operation and alleviating the structural loads and is fully equivalent to the commercial wind turbine controllers [86],[87].

Since the wind turbine dynamics in the generator speed loop are nonlinear, a nonlinear controller design approach is adopted to effectively accommodate the aerodynamic nonlinearities between the aerodynamic torque and wind speed, rotor speed and the rotor blade pitch angle. Furthermore, the presence of the RHPZs impose limitations on the generator speed controller design which must be considered in the linear control design task of the generator speed loop controller.

The aerodynamic torque is a nonlinear function of the blade pitch angle β , rotor speed Ω , and the effective wind speed V_{eff}.

$$T(\beta, \Omega, V_{eff}) \tag{4-8}$$

According to the separability theory described by (Leith and Leithead) in [87] ,[88] the aerodynamic torque nonlinearities are separated by

$$T(\beta, \Omega, V_{eff}) = h(\beta, \Omega) - g(V) \tag{4-9}$$

As a result of this separation the effective wind speed dependent term can be considered as an additive disturbance to the system, and a global cancellation of the first term is possible by including a $h(\cdot)^{-1}$ term to the system.

The above rated and below rated linear controllers, are designed on the basis of lead compensation, including an integrator for eliminating the steady state error and a lead compensator for obtaining the necessary closed-loop stability margin, using loop shaping in frequency domain. Since the generator speed control loop includes integral control action to eliminate the steady state error, the open-loop phase is reduced around the crossover frequency. Thereby, a lead compensator is included to compensate for the phase lag introduced by the integral control action and the system dynamics at the vicinity of crossover frequency.

The requirements for the controllers are defined to achieve good disturbance rejection at low frequencies, 1 rad/sec close-loop bandwidth, high frequency roll-off rate, and acceptable gain and phase margins. The controller must ensure the system stability in all operating conditions.

Furthermore, the full envelope controller includes additional control loops to alleviate the structural loads. Tower and drive train feedback loops are designed to increase aerodynamic damping at the first tower frequency, and drive train damping at the first drive train frequency.

A band pass filter centred at the first drive train frequency is utilised to increase damping at the first drive train mode, as shown in Figure 4-9.

Figure 4-10 illustrates the effect of drive train filter on the first drive train mode, where $Gdtr(s)$ is drive train filter, $C(s)$ is wind turbine controller, and WT is wind turbine dynamics.

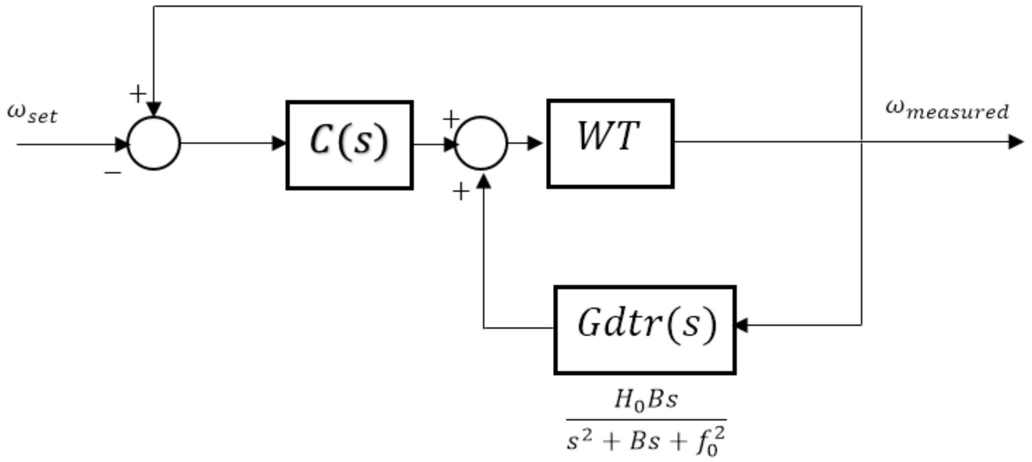


Figure 4-9: Drive train filter

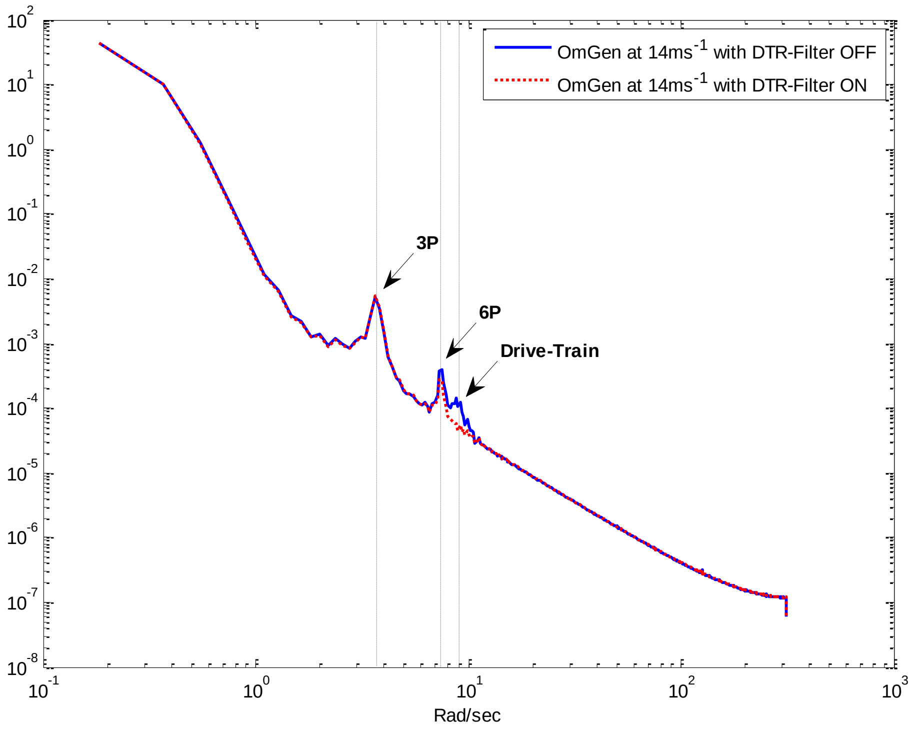


Figure 4-10: Power spectrum of generator speed with and without Drive train filter

Additionally, a tower feedback loop is included to actively alleviate the tower loads. The tower motion is regulated through the blade pitch angle controller in response to the tower-top acceleration. Figure 4-11 illustrates the addition of the tower feedback control loop to the generator speed control loop. Since the generator speed controller bandwidth is around 1 rad/s, it is below the first tower frequency; therefore, the performance of the generator speed loop is not compromised by the addition of the tower feedback loop.

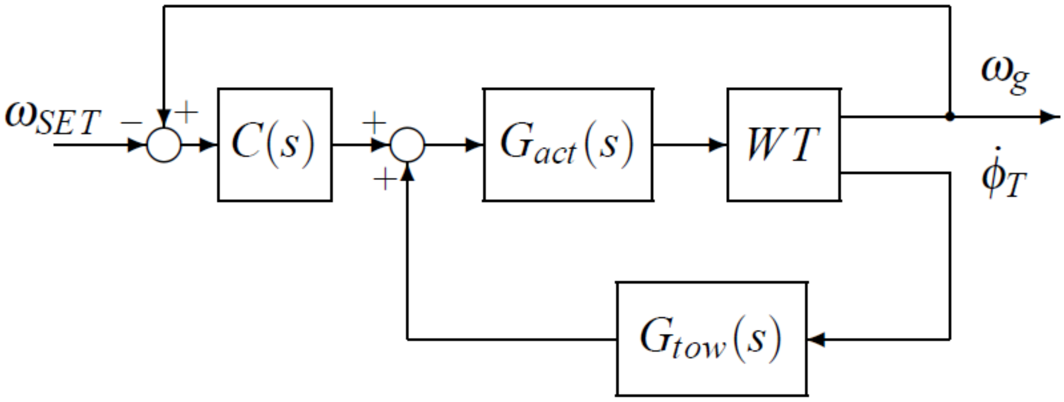


Figure 4-11: Tower feedback loop [22]

The power spectrum of the fore-aft tower moment in Figure 4-12 clearly reveals the effect of the tower feedback loop in the first tower mode. The power spectrums are plotted based on the simulation results obtained from Bladed at $14ms^{-1}$ mean wind speed with 10% turbulence intensity.

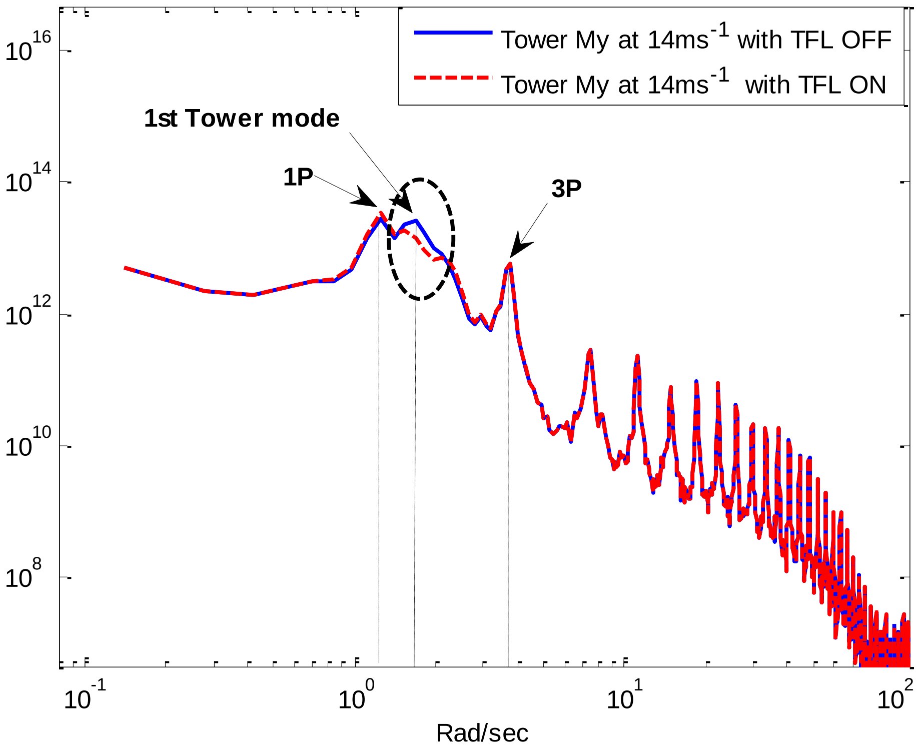


Figure 4-12: Nacelle fore-aft acceleration power spectrums with and without tower feedback loop

In order to avoid interaction between the generator speed control loop and the tower feedback loop, a coordinated controller design is implemented in the full envelope controller.

The coordinated controller is used to reduce the pitch activity in vicinity of the tower frequency. The principle behind this approach is to decouple the generator speed loop from the tower feedback loop in order to achieve more effective tower feedback loop (TFL) control. Figure 4-13 illustrates implementation of the coordinated controller in the full envelope controller.

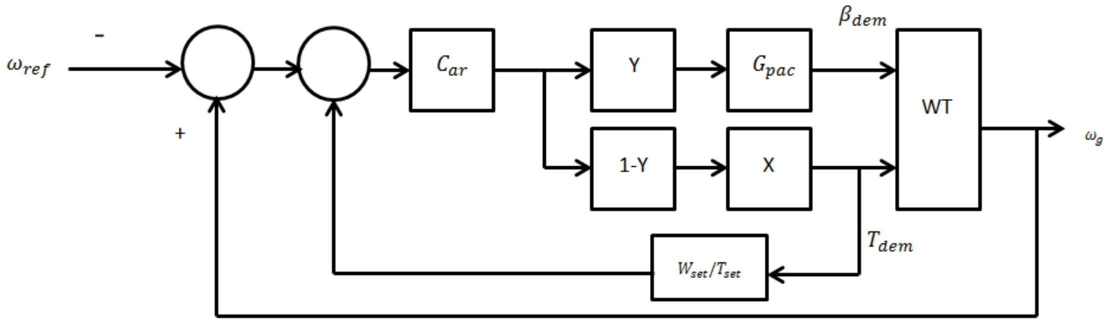


Figure 4-13: Coordinated control design

where, X, Y are the coordinated controller transfer function and the notch filter transfer function designed at the first tower frequency.

The application of the coordinated controller design for the 5MW Supergen Exemplar model is given in [89], [22]. The effect of the augmented coordinated controller is evident in the PSD plot of the fore-aft acceleration in Figure 4-14.

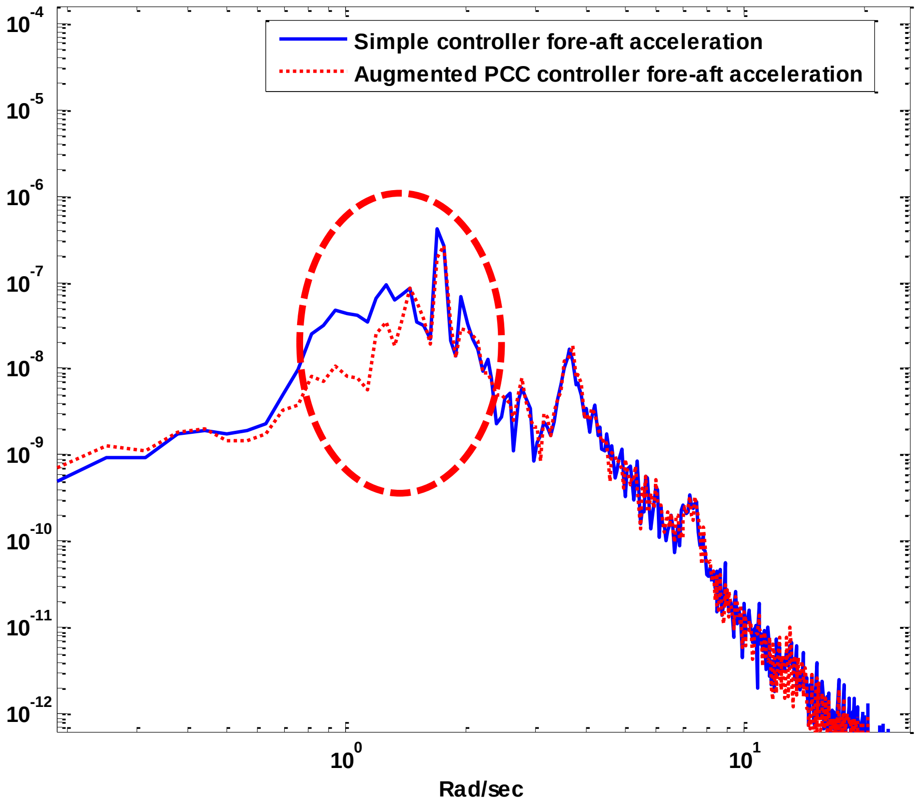


Figure 4-14: Power spectrum of the fore-aft acceleration with simple controller, and augmented coordinated controller

Figure 4-15 illustrates the PSD plot of the tower fore-aft moment (My) with and without augmented power coordinated controller (PCC).

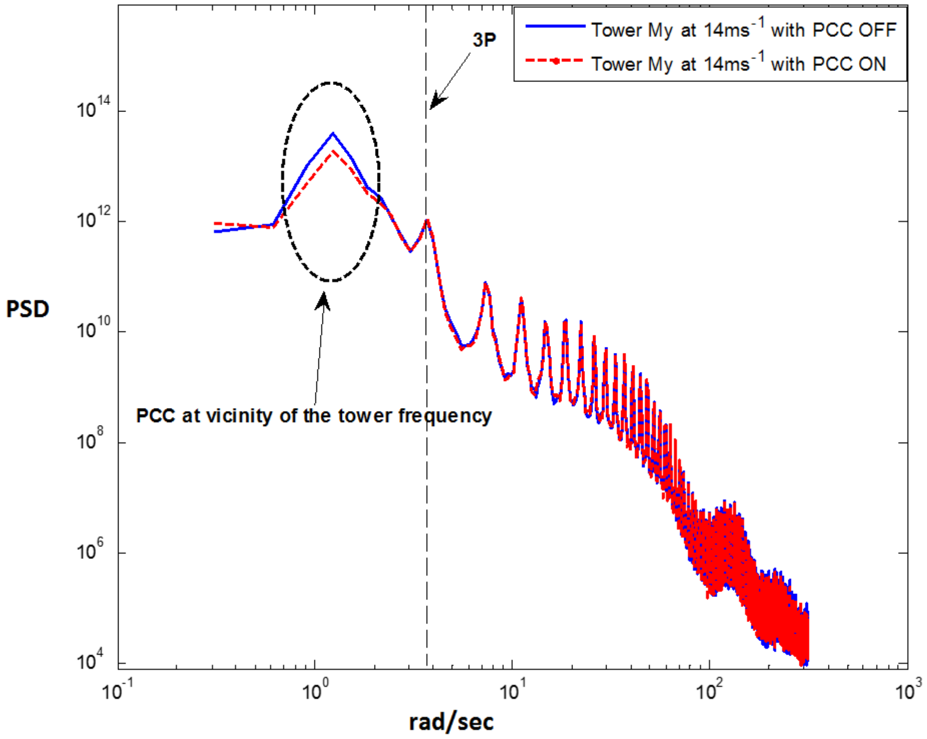


Figure 4-15: Power spectrum of the tower My (fore-aft moment) with simple controller, and augmented coordinated controller

4.4 Wind turbine power adjusting controller

The power adjusting controller (PAC)[83], is a feedforward modification to the full envelope controller that includes flexible adjustment of the power generated by wind turbine. Since dynamic power set-point adjustments are required for wind farm controller design, PAC is included in the wind turbine model.

The PAC acts as a feed-forward modification to the full envelope controller; thus, the full envelope controller operation is not affected by the PAC.

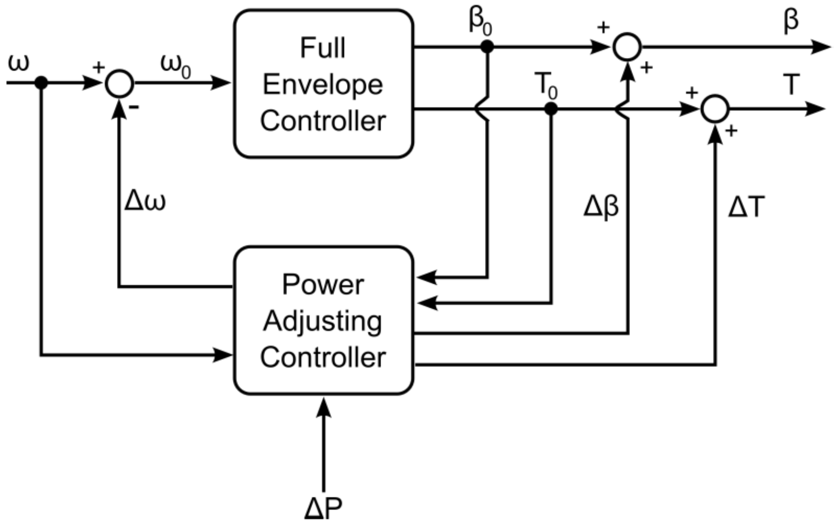


Figure 4-16: Power Adjusting Controller [83]

The PAC alters the power output of the wind turbines by adjusting torque and pitch demand of the full envelope controller, corresponding to a power set-point adjustment request by the wind farm controller, as shown in Figure 4-16. The implementation of the power adjusting controller is also shown in Figure 4-17 and described in [83].

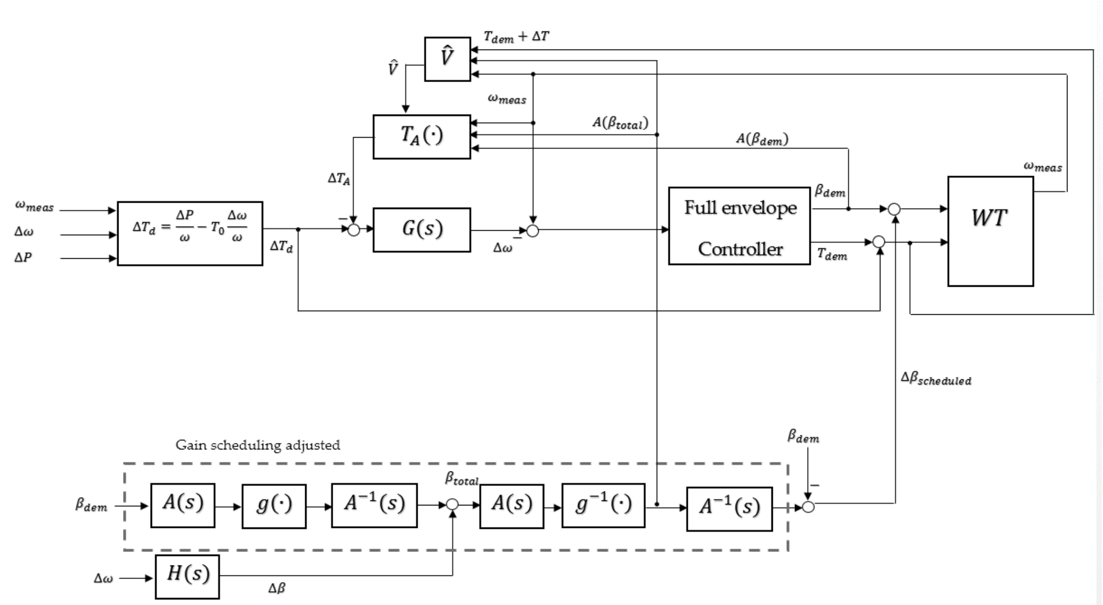


Figure 4-17: Implementation of the gain scheduled pitch control loop in the PAC

In order to avoid the wind turbine operating outwith the safe operating boundaries as the result of PAC, a set of PAC supervisory rules were defined to ensure the safe operation of the wind turbine whilst the PAC adjustments are applied [83].

The PAC generates a set of flags, indicating the operating condition of the turbine that can be used by the wind farm controller to make decisions about each turbine power set-point adjustment regime.

Two sets of PAC supervisory rules are implemented to mark the safe operating boundaries of the wind turbine. The Black (hard) limit and the traffic light (soft) limits.

- Black supervisory rules: boundaries are defined that cannot be crossed under any conditions. These limits are defined in the torque-speed plane based on the physical properties of the turbine.
- Traffic-light supervisory rules: green/amber/red boundaries lying within the black boundaries are defined within which adjustment of the power set-point is possible. The set-point adjustment is subjected to a maximum within the green and amber boundaries, but must be zero outwith the red boundary.

The black limits are defined to mark the most outer edge of the wind turbine operating region. This boundary limit is shown with a black thick line in Figure 4-18 and defined based on the rotor upper/lower torque limits and rotor maximum/minimum speed limits.

The region outside this boundary is defined as the Black region. The wind turbine operating point should never cross this boundary in any condition. The black boundary limits cannot be changed without agreement by the original equipment manufacturer (OEM).

The region inside the black boundary is defined as the red traffic light region. Only in high priority events such as synthetic inertia response, the wind turbine is allowed to operate in this region. The wind farm controller can request the wind turbines' power adjusting controller to set the Priority flag in order to extract more power by entering the red traffic light region. In low priority events, the amount of extra power , ΔP_{Red} , allocated for this region is zero. The red traffic light region is bounded on the outside by the black region boundary limits and bounded on the outside by the red/amber region boundary limits.

The region inside the red region is the amber region which is separated from the red region by the red/amber boundary limits. The amber region boundary can be adjusted by the wind farm control designer and the amount of extra power , ΔP_{Amber} , allocated here for this region is $0.2MW$. The amber traffic light region is bounded by the red/amber region boundary and the amber/green region boundary. The region inside the amber/ green boundary is defined as the green traffic light region and the amount of extra power , ΔP_{Green} , allocated here for this region is $0.5MW$.

Unlike the black limits the traffic light limits can be adjusted by the wind farm control designer. The amount of allocated power for each traffic light region can be defined as required.

The PAC operating regions and boundaries are depicted in Figure 4-18.

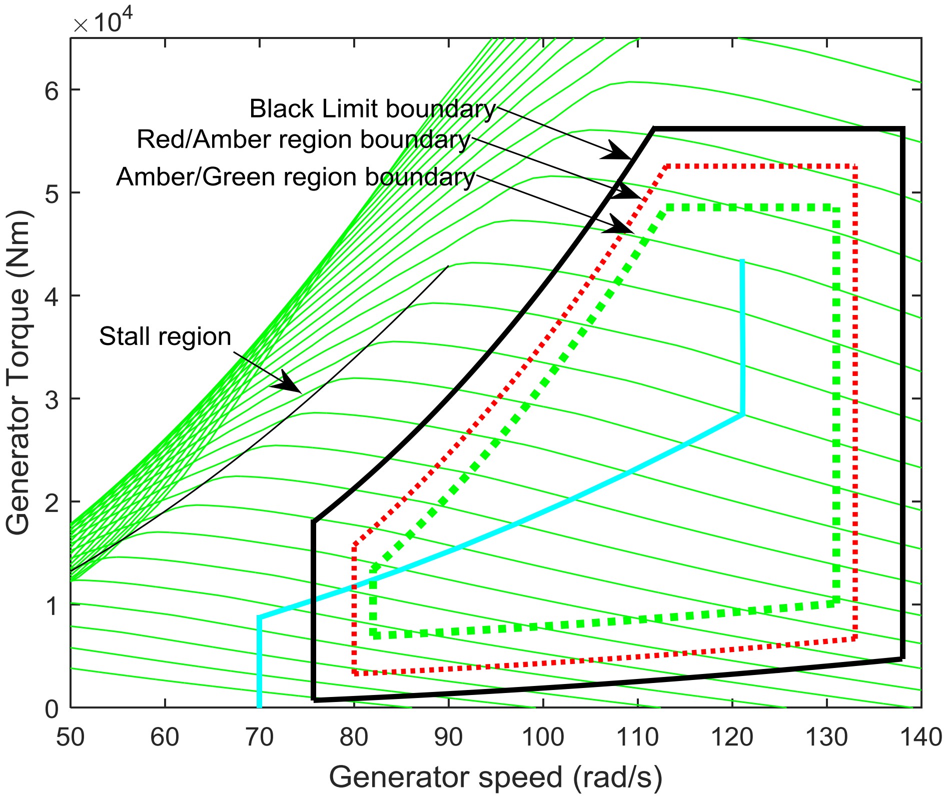


Figure 4-18: PAC safe operating regions and boundary limits

4.5 Implementation of the 5MW Supergen controllers in C code and MATLAB C MEX S-Function

When the wind turbine and its full envelope controller are modelled entirely as a continuous time system, it is a stiff system for which a variable step solver is essential. However, when the full envelope controller is implemented as a discrete section, an alternative solver, that also works efficiently, is ode14x a fixed step solver. It would, therefore, be possible to implement the full wind turbine model in discrete form incorporating the solver algorithm. However, the discretisation of the wind turbine model has not been done to date but is left to future work.

Some of the full envelope controller functionalities, such as the anti-windup loop, are not easy to implement in the continuous time domain. Therefore, the discretised form of the controller is implemented by Simulink S-Functions to include the full functionality of the controller. Moreover, a DLL of the C controller is compiled and used in the DNV-GL Bladed software to evaluate the discrete controller performance.

The full envelope controller is converted from s-domain to z-domain using Tustin's Bilinear transform method.

$$s \cong \frac{2}{T}\frac{z-1}{z+1} \tag{4-10}$$

The Bilinear transformation is done as follow

$$C(s) \xrightarrow{s \cong \frac{2z-1}{Tz+1}} C(z) = \frac{n(z)}{d(z)} = \frac{a_n z^{-n} + a_{n-1} z^{-(n-1)} + \cdots + a_1 z^{-1} + a_0}{b_m z^{-m} + b_{m-1} z^{-(m-1)} + \cdots + b_1 z^{-1} + 1} \tag{4-11}$$

Therefore, the outputs of the transfer functions depend on the previous input plus the current input times a constant. The discretised form of the model in C/C++ code is compiled into DLL using Matlab C-MEX compiler. The DLL of the controller is used with S-Functions to implement the discrete controller in the Simulink model. The MEX-Function refers to Matlab Executable external interface function.

The advantages of using the discrete controller in the Simulink model are summarised as:

- Increase the simulation speed
- Decouple the simulation sampling time from the wind turbine controller, the wind model, and the wind farm controller sampling times
- Utilise Anti-Windup loop in discrete form
- Possibility of using Simulink fixed-step solvers for this stiff model
- Develop wind farm models utilising discretised wind turbine models

Figure 4-19 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. The Controller.c and Controler.h codes contain the discretised form of the wind turbine controllers. The Controller1.c is the wrapper function that is required by the compiler to generate the Controller1.mexw64 MEX file. The Simulink S-Function block calls the discretised wind turbine controller code at each iteration through the Controller1.mexw64 MEX file.

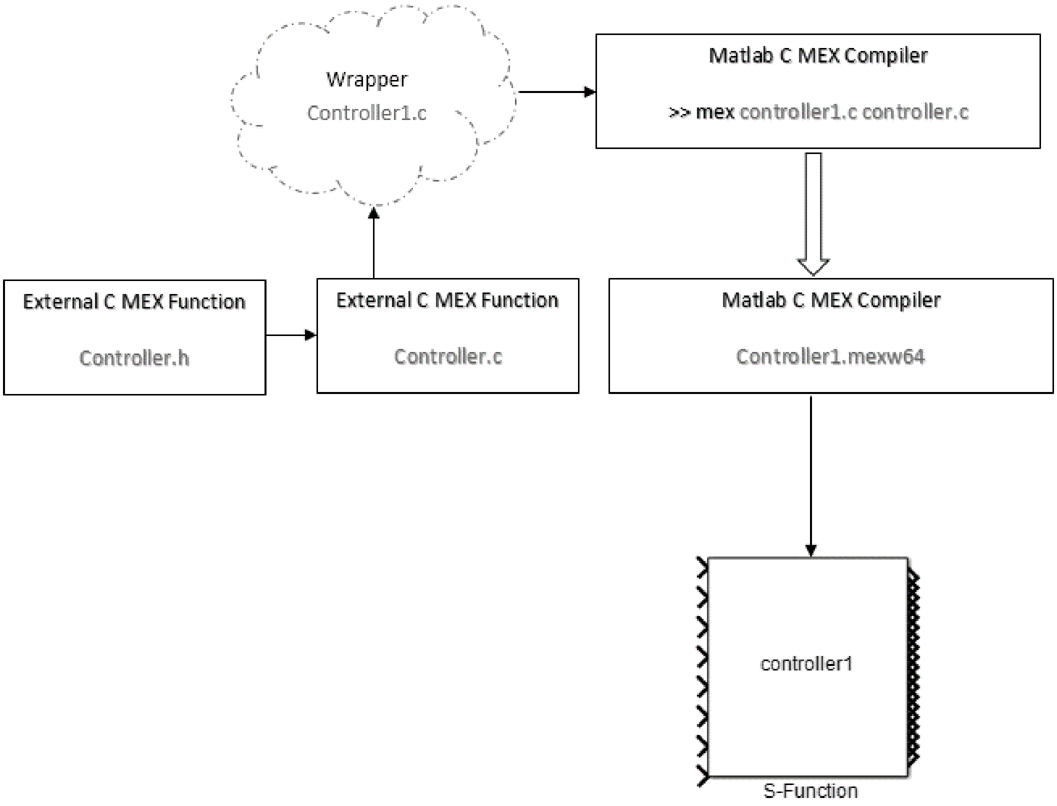


Figure 4-19: Compiling process of C controller

In future work the ode14x fixed step solver will be used to discretise the wind turbine model dynamics and build the whole wind farm model in C code.

4.6 Evaluation of the 5MW Supergen model performance and its discretised controllers in Simulink

The 5MW Supergen wind turbine model and its discrete controller are utilised for wind farm modelling. The PAC capability of adjusting the power set-point of the wind turbines in a safe and quick operating manner is the key to wind farm controller design. In order to evaluate the performance of the wind turbine model and its controllers, a number of simulations in below rated and above rated wind speeds are conducted. The aim is to test the discretised wind turbine controller in different control scenarios.

4.7 Wind turbine power set point reduction in below rated operation

The wind turbine model is simulated at below rated $8ms^{-1}$ mean wind speed with 10% turbulence intensity. The effective wind speed used in this simulation is shown in Figure 4-20. To investigate capability of the wind turbine for power curtailment, de-loaded operation of the wind turbine at different power set point reductions is investigated. The power set-point of the wind turbine is reduced by 0.25MW, and 0.5MWrespectively.

Figure 4-21 illustrates the power output of the turbine before and after the power set-point reductions at 500s. The difference between the power outputs before and after power set-point reductions is also shown in this figure. To investigate the operating condition of the wind turbine in this control mode, the PAC supervisory flags are illustrated in Figure 4-22. The wind turbine operating status is also shown in the generator speed generator torque graph in Figure 4-23.

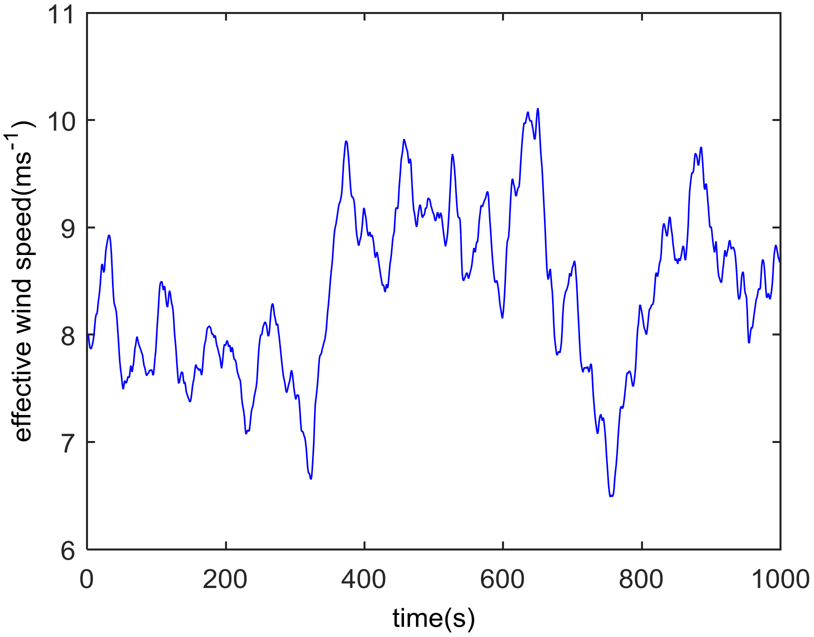


Figure 4-20: Effective wind speed with a mean value of $8ms^{-1}$

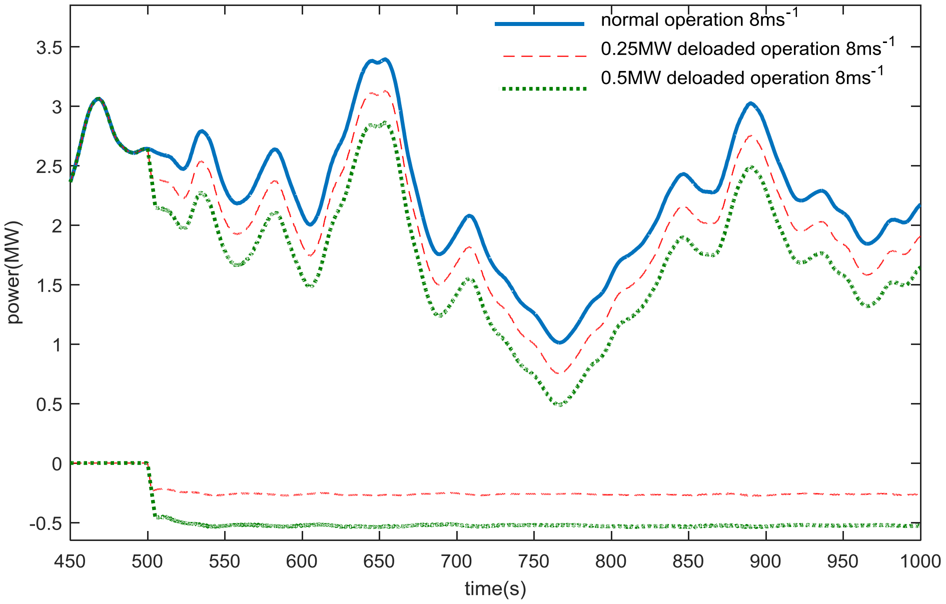


Figure 4-21: Power de-loaded operation at below rated

As can be seen from the wind turbine's generator speed-generator torque graph, the wind turbine operation is kept within a safe operating boundary. The power output of the turbine is reduced by 5% and 10% of its rated power. The supervisory flags indicate that the wind turbine operating point is crossed the amber traffic light zone for some times for 10% power reduction in below rated operation. These flags can be used in the wind farm controller design to adjust the power set-point of the turbines according to their operating condition. The power deloading control of the wind turbines can be used to develop a wind farm control algorithm for the power curtailment objective.

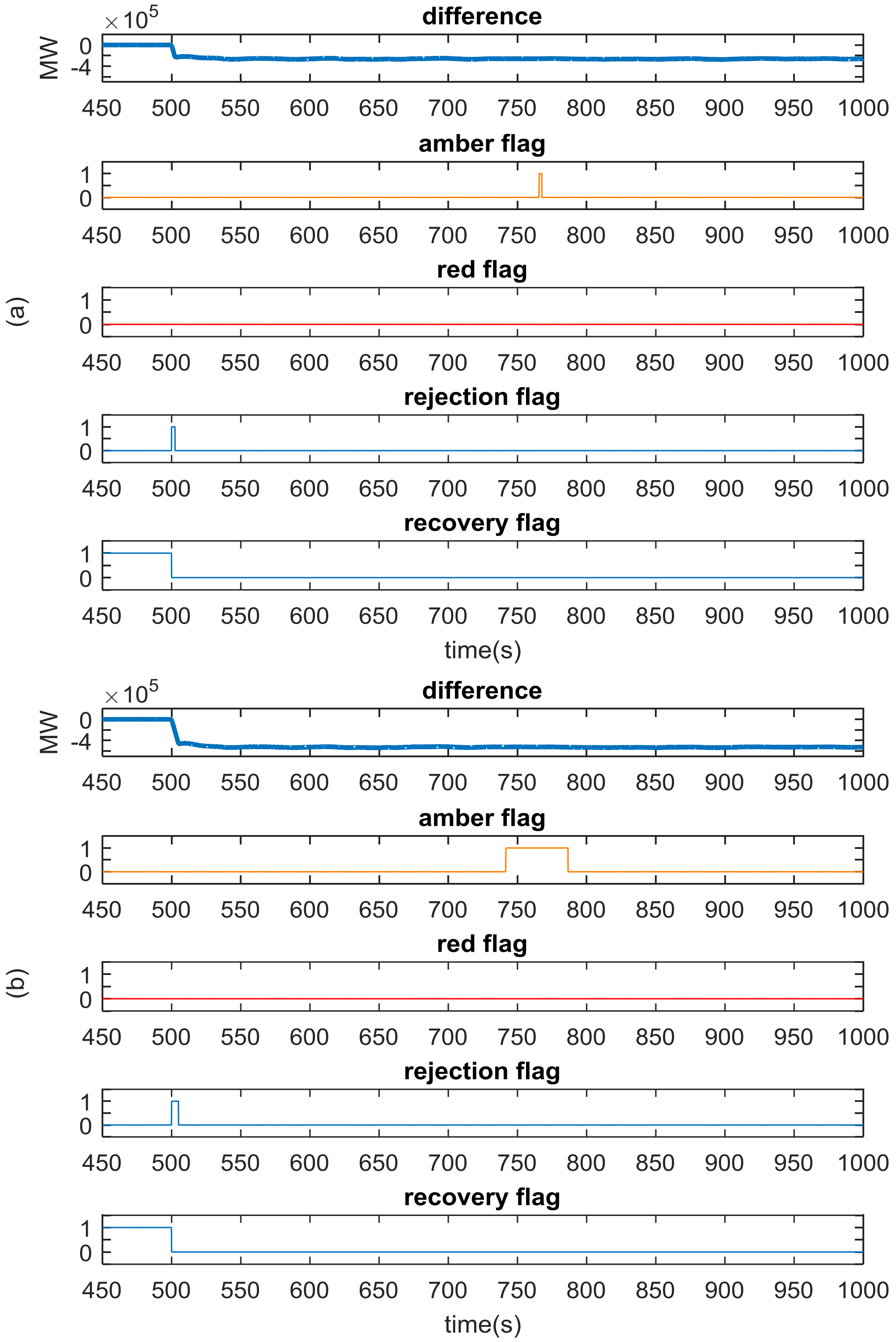


Figure 4-22: Wind turbine operating status at 0.25MW (a) and 0.5MW (b) power de-loaded operation

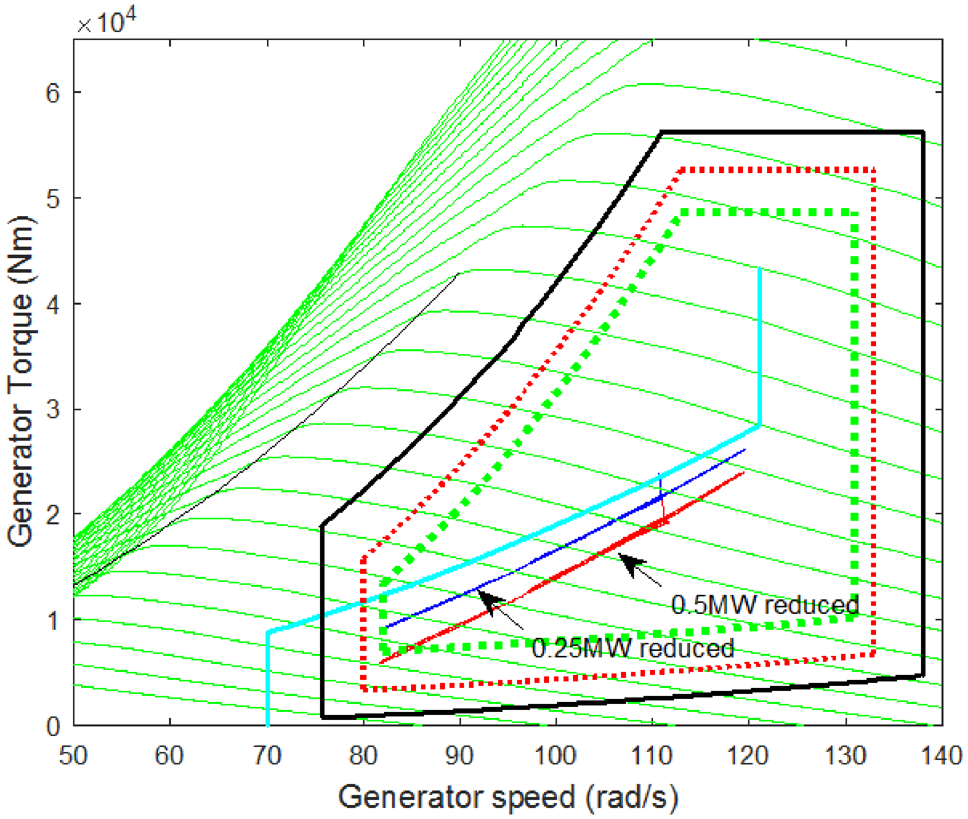


Figure 4-23: Wind turbine operation in the Generator Torque/Speed plane

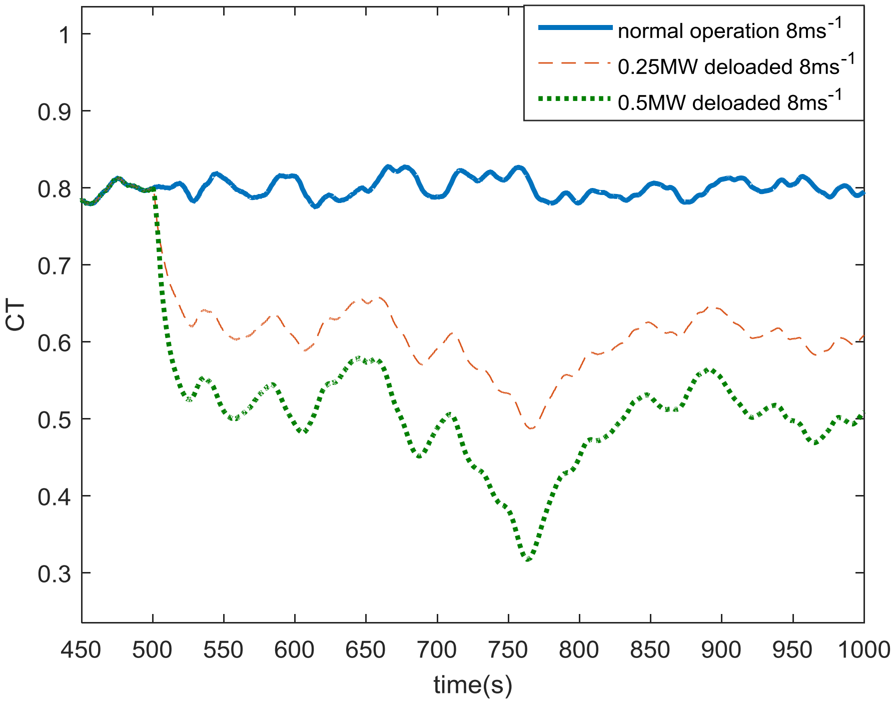


Figure 4-24: Thrust coefficient before and after power adjustment

Figure 4-24 illustrates the thrust coefficient of the wind turbine for normal operation and different level of de-loaded operations. Since the wake effects are directly related to the thrust coefficient of the turbine, reduction in the thrust coefficient as a result of de-loaded operation of the turbine suggests that this wind turbine's wake effects can be alleviated by carefully adjusting the wind turbine operation.

The reduction of energy capture can also reduce the aerodynamic loads on the turbine structure. Figure 4-25 illustrates the cumulative PSD graph of the fore-aft acceleration of the wind turbine tower at this control mode at different level of power reduction. As can be seen the fore-aft acceleration of the tower is reduced as the power capture is reduced. As a result, the root bending moment of the tower is reduced and therefore the tower fore-aft load can be reduced by adjusting the turbine power set-point. This idea may be used to coordinate the wind turbines' operation in a wind farm to reduce the aerodynamic loads.

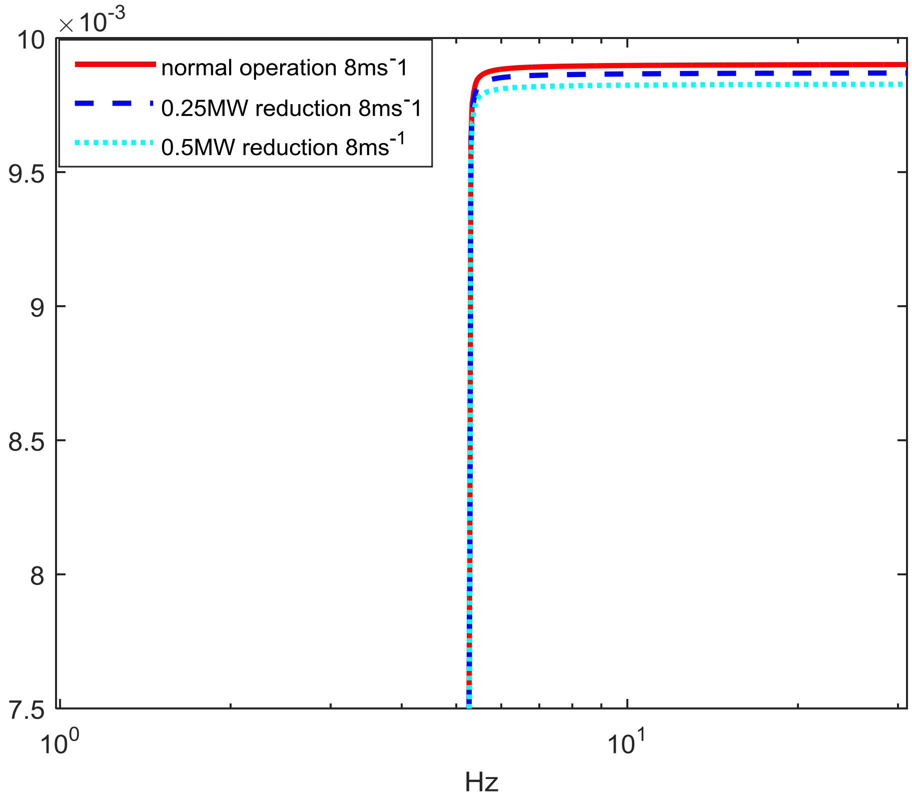


Figure 4-25: Cumulative PSD plot of the wind turbine tower fore-aft acceleration

4.8 Wind turbine power set-point increase in below rated operation

The power set-point increase capability of the wind turbine and its controllers is investigated in this simulation. The power set-point increase of the wind turbines can be utilised in the wind farm control algorithm for implementing the primary frequency response.

The same below rated effective wind speed as shown in Figure 4-20 is used and the power set-point of the wind turbine is also increased by 0.25MW and 0.5MW respectively. Since the power output of the wind turbines must be adjusted in a short time scale in response to a perturbation in the grid frequency, the ability of the wind turbine controller to increase the power set-point of the turbine in a quick and safe operating manner at below rated wind speeds is of most interest in this simulation.

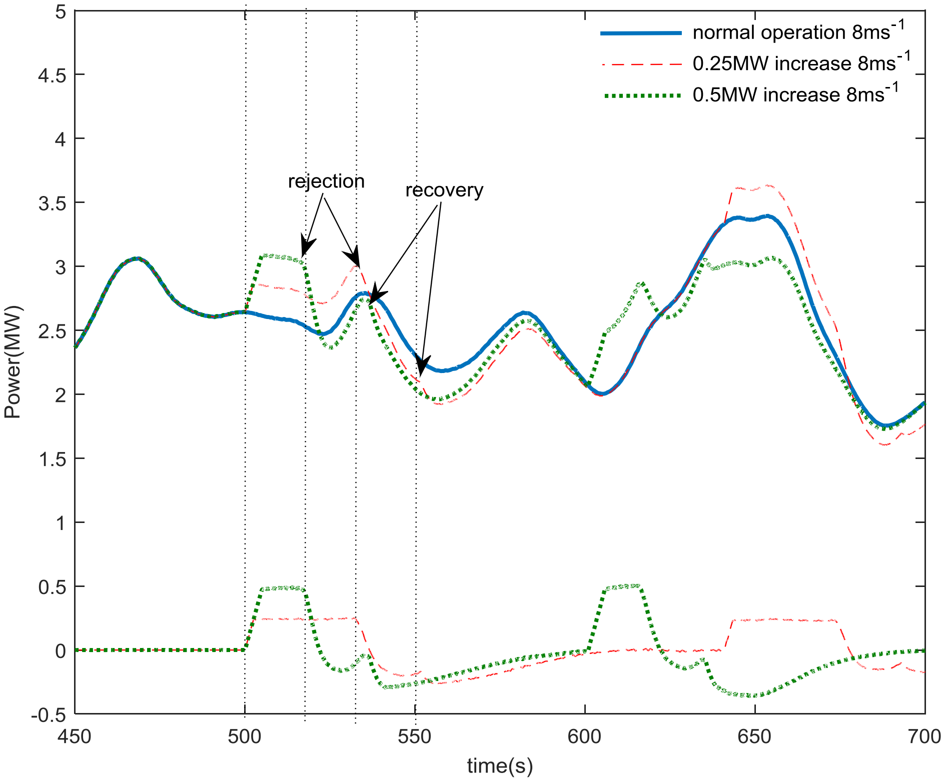


Figure 4-26: Wind turbine power output before and after power set-point increase

Figure 4-26 illustrates the power output of the turbine before and after power set-point increase of 0.25MW and 0.5MW. The difference between the power outputs before and after these power set-point adjustments are also depicted in the graph. As can be seen the period of time that the wind turbine can provide the increase power set-point request varies at different level of power set-point increase. For higher level of power set-point the wind turbine increased power delivery is shorter time in comparison with the lower power set-point increase.

Figure 4-27 depicts status flags generated by the power adjusting controller indicating the operating condition of the wind turbine after this power set-point adjustment. The traffic light rules are not applied in this simulation. The power output of the wind turbine is increased by 0.25MW for almost 40 seconds and by 0.5MW for almost 20 seconds. Once the operating point of the wind turbine hits the black boundary the rejection flag is set and the power set-point delivery request is rejected by the PAC. 20seconds after the rejection of the power set-point delivery the PAC recovery is started. The recovery of the PAC is evident in the wind turbine power output graph in Figure 4-26.

Figure 4-28 illustrates the operating condition of the wind turbine in the generator speed-generator torque graph. As can be seen, the wind turbine power set point is moved towards the black boundary in the power set-point increase control mode. The level of power set-point increase is evident in the plot (a) and (b) in Figure 4-28. Since the traffic light rules are not considered in this simulation, the wind turbine power set-point crosses the traffic light limits and hits the black boundary.

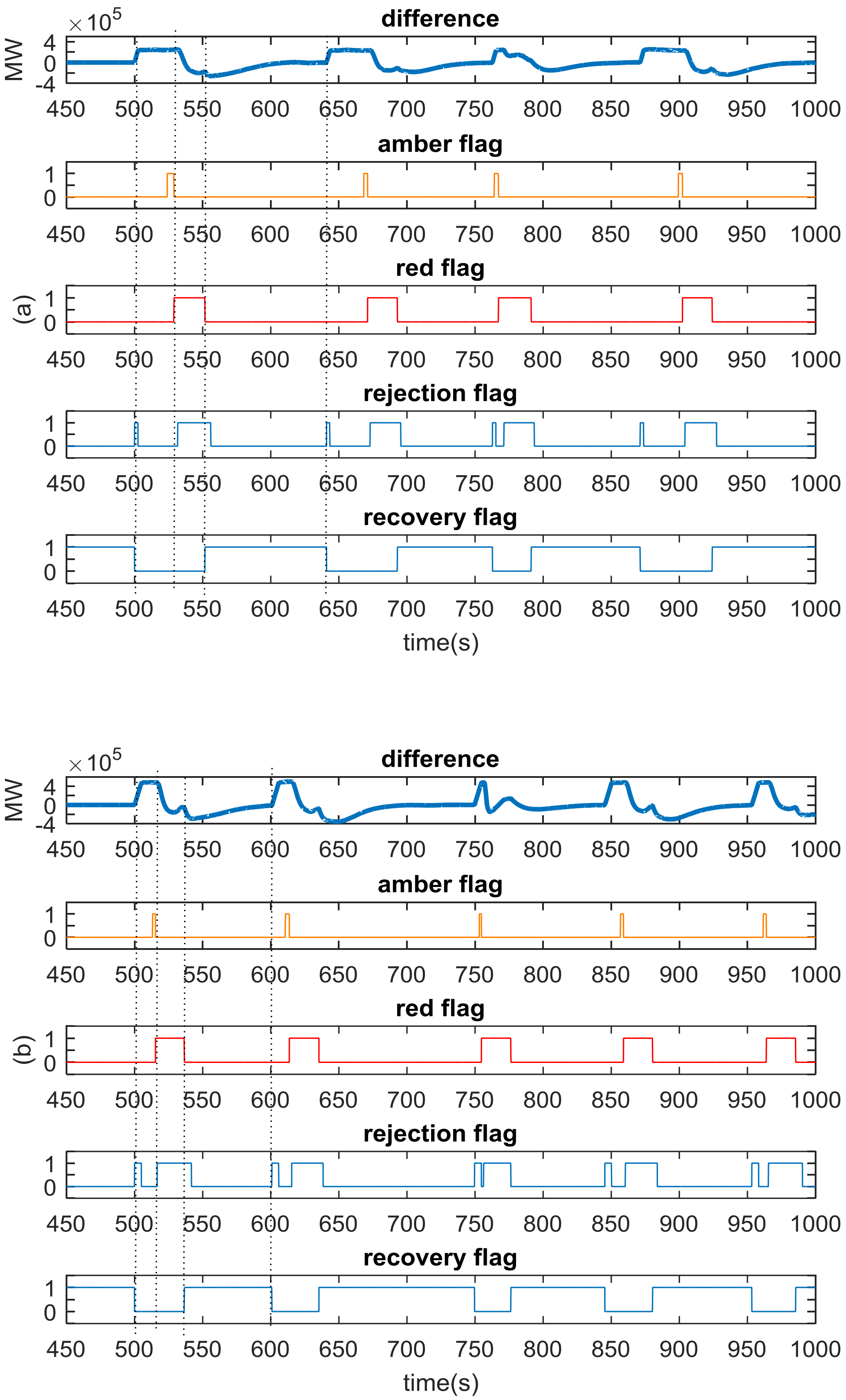


Figure 4-27: Wind turbine operating status at below rated operation with (a) 0.25MW power increase and (b) 0.5MW power increase

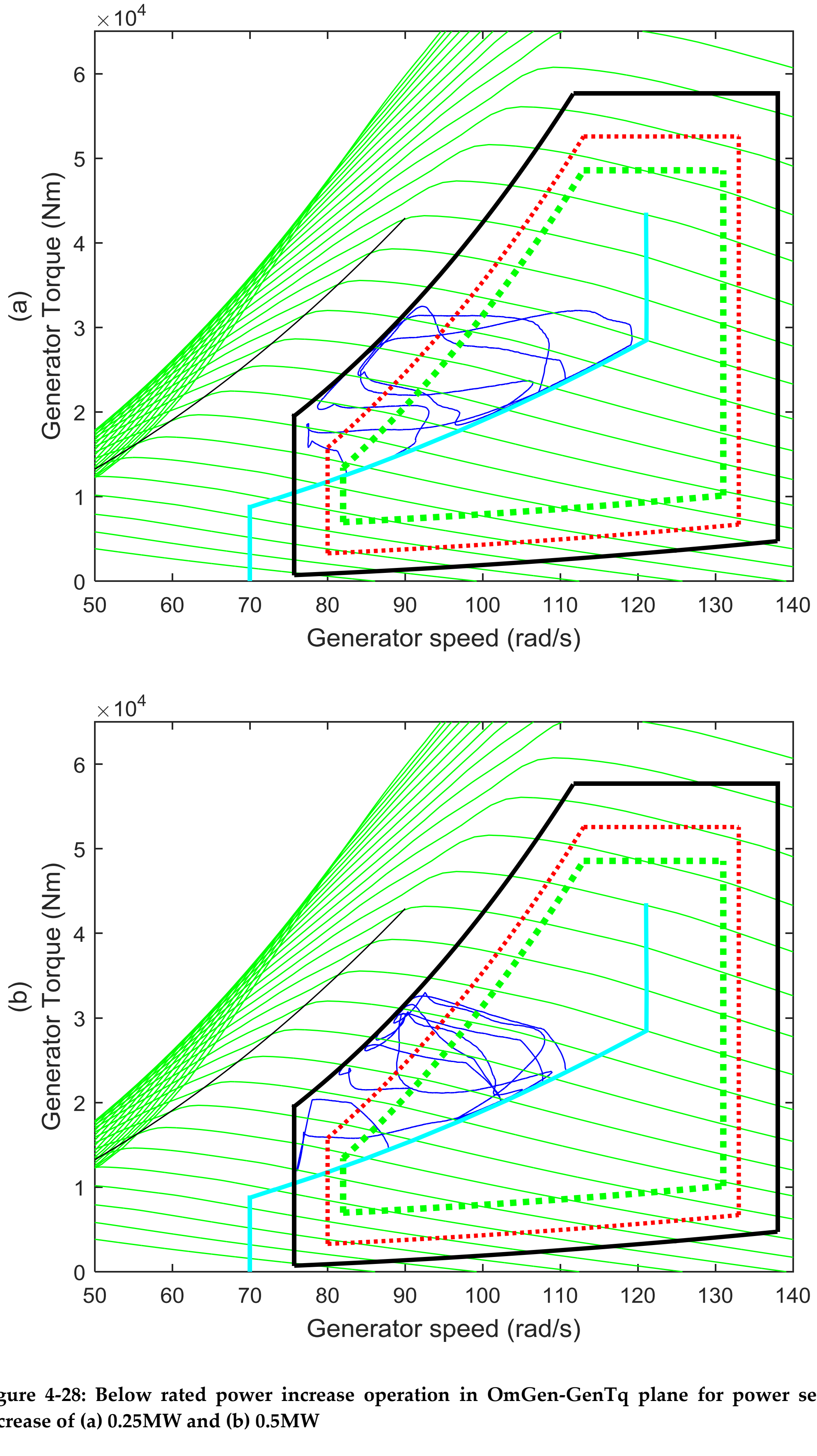


Figure 4-28: Below rated power increase operation in OmGen-GenTq plane for power set-point increase of (a) 0.25MW and (b) 0.5MW

4.9 Wind turbine power set-point reduction in above rated operation

Similar control strategies are adopted in the above rated operation of the wind turbine at $14ms^{-1}$ mean wind speed with 10% turbulence intensity. Figure 4-29 to Figure 4-31 illustrate the operation of the wind turbine in the above rated operation in the curtailed control mode.

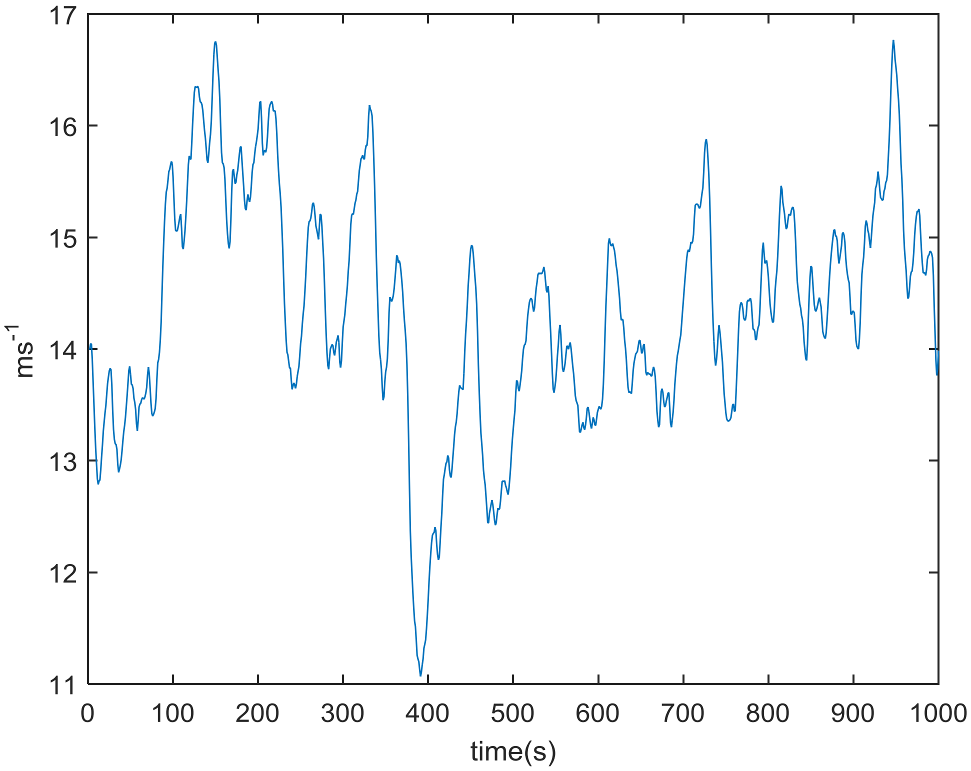


Figure 4-29: Effective wind speed with a mean value of $14ms^{-1}$

In above rated, the full envelope controller of the wind turbine employs a collective pitch control action to maintain the generator speed and power at their rated values. Therefore, the power curtailment task becomes easier since the pitch controller is already in action. Figure 4-30 to Figure 4-32 show an example of the power de-loading operation of the wind turbine in above rated. The operating status flags generated by the PAC are depicted in Figure 4-32. The operation of the wind turbine in the green operating region after this power adjustment is evident in the Generator Torque/Speed plane in Figure 4-31 and the PAC supervisory flags in Figure 4-32.

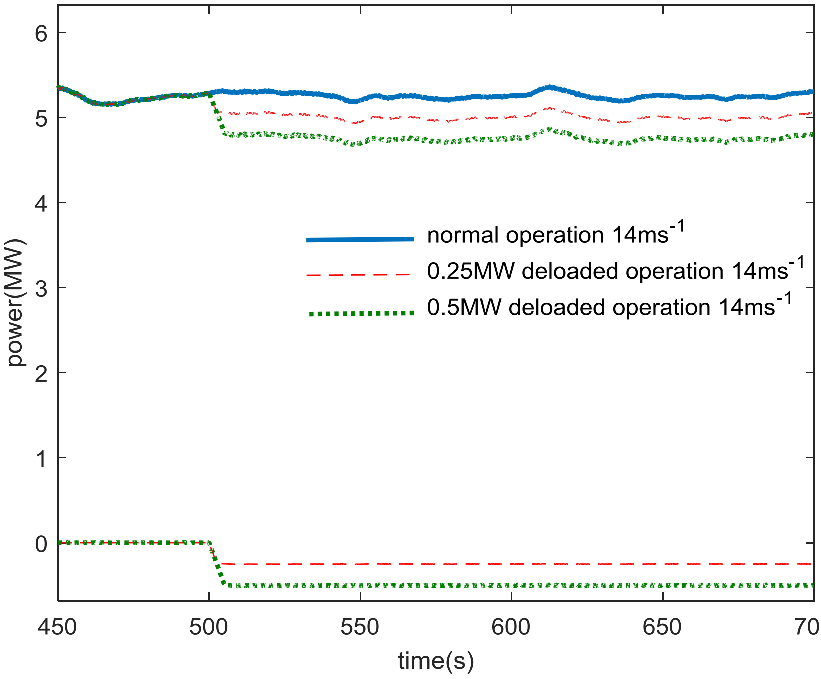


Figure 4-30: Mechanical power output in de-loaded operation in above rated wind speed

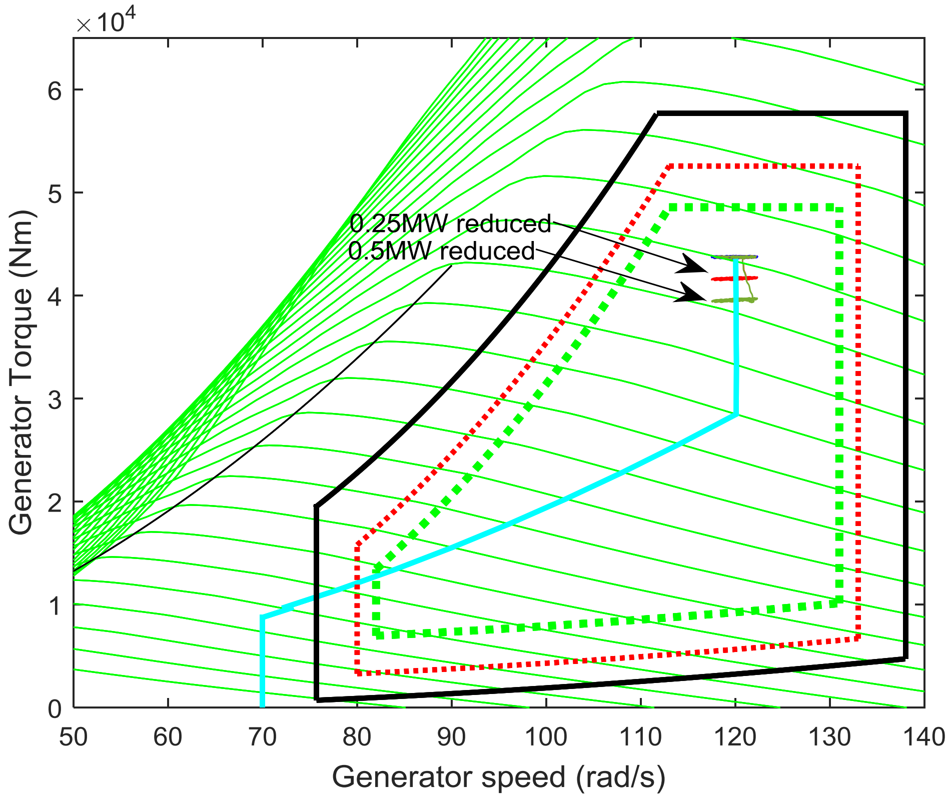


Figure 4-31: Deloaded oeration in generator Torque/Speed plane in above rated

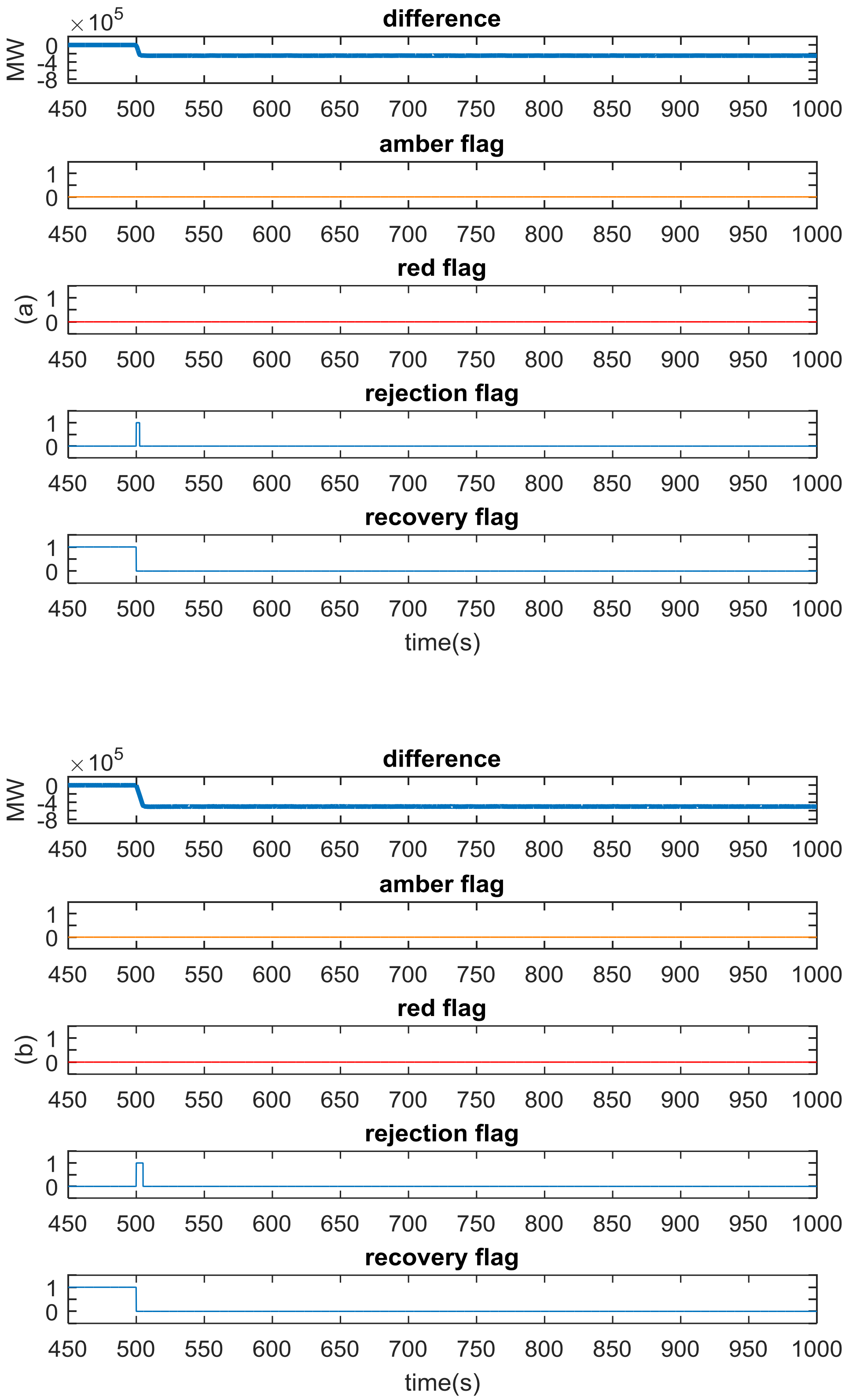


Figure 4-32: Wind turbine status flags in above rated de-loading operation

4.10 Wind turbine power set-point increase in above rated operation

Using the same above rated wind field, the power increase control mode of the turbine is investigated. Figure 4-33 to Figure 4-35 illustrate the results of this simulation.

The power increase control mode is more achievable in above rated where enough wind power is available. Wind turbine power converters are capable of increasing their power output by 10% above the generator rated power for a reasonable period of time before they cross their maximum heating limitation [90].

As can be seen from the generator speed-generator torque graph the wind turbine provides 10% extra power above its rated, within the safe operating boundary. This ability of the wind turbine and its controller can be deployed in the wind farm control design for providing primary frequency response to the grid.

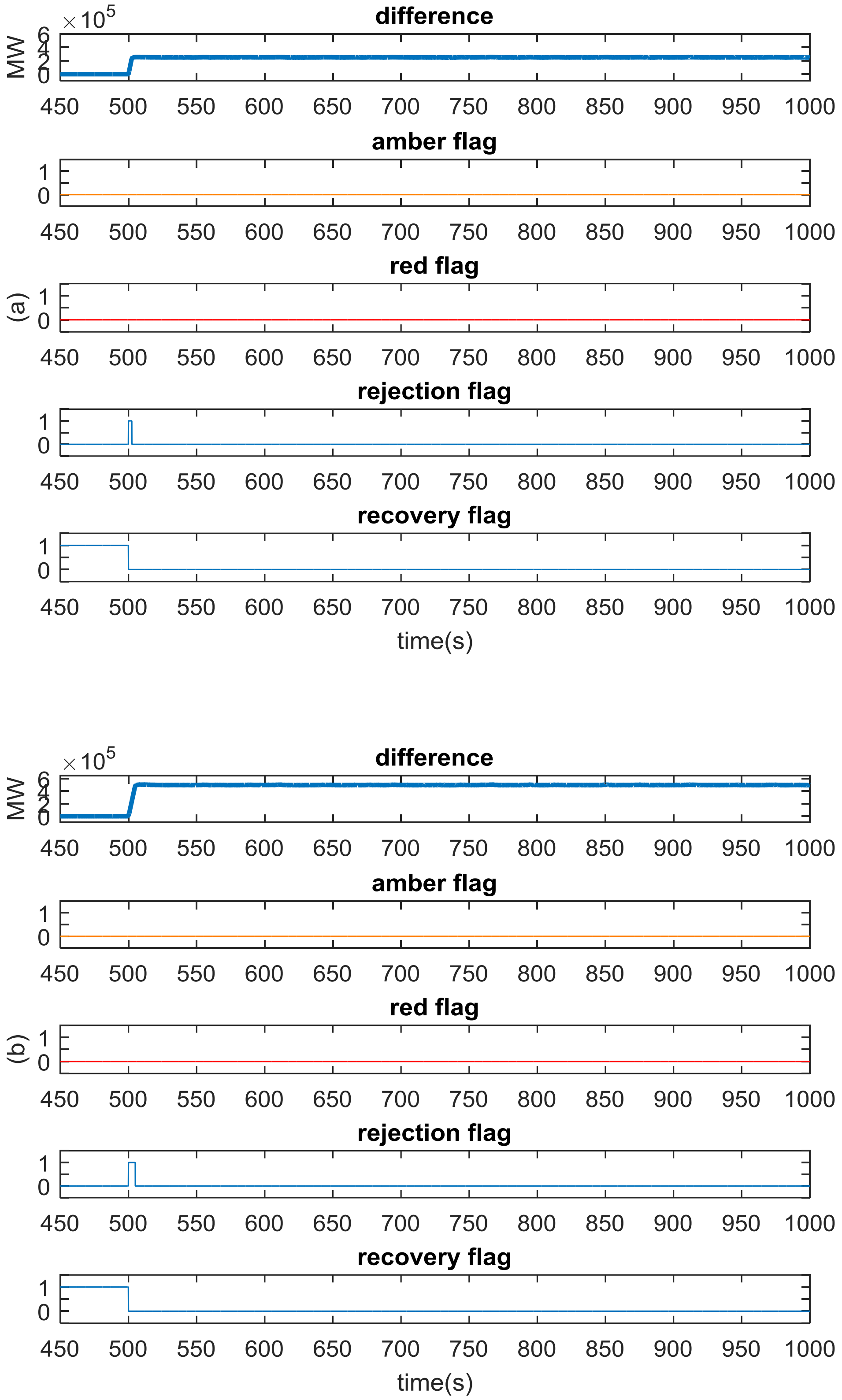


Figure 4-33: operating status flags for above rated power set-point adjustment

The power set-point adjustments are achieved according to the traffic light rules as shown in Figure 4-33. The power set-point adjustment level is reduced once the amber and red limits are realized. This is shown in the power output graph in Figure 4-34. The operation of the wind turbine is kept inside a safe operating boundary, as shown in Figure 4-35. At the black boundary, the PAC is kept ON for 20 seconds before being reset to OFF and the recovery process initiated. Once the recovery is completed, the PAC is ready to adjust the power output of the turbine all over again. The power adjustment after recovery is evident in Figure 4-33 to Figure 4-35.

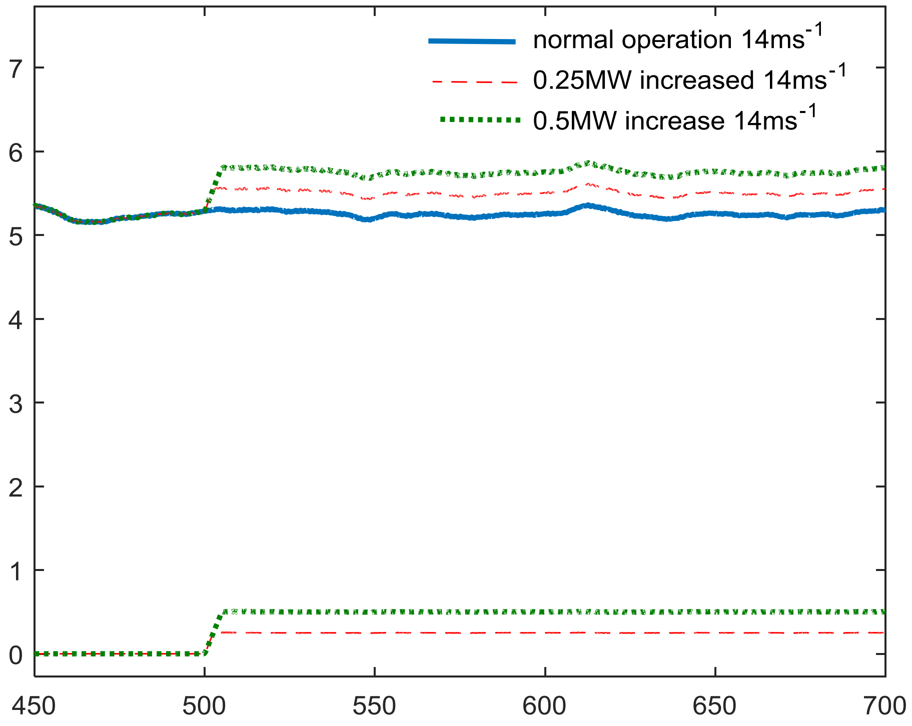


Figure 4-34: Power before and after power adjustment in above rated

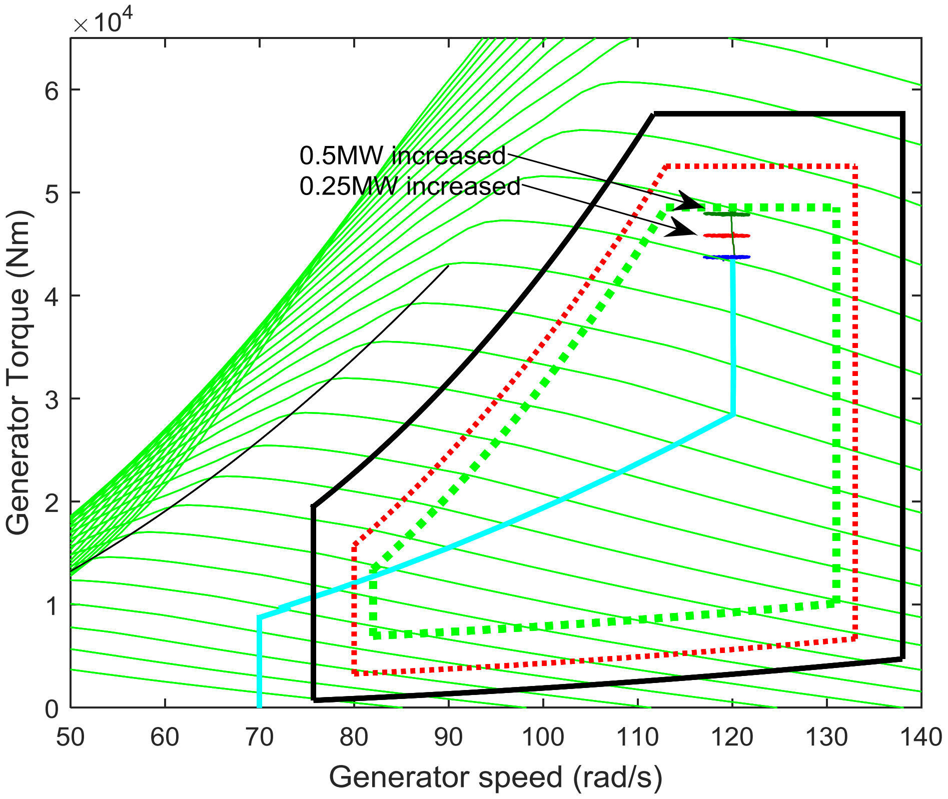


Figure 4-35: Wind turbine operating condition in Generator Torque/Speed plane, above rated

4.11 Concluding remarks

In this chapter, the 5MW Supergen Exemplar wind turbine model and its controllers are presented. The procedure of converting the wind turbine controllers into C code to be used in Simulink is detailed. The wind turbine model is tested for different control scenarios in below and above rated wind speeds. It is shown that the wind turbine model is capable of providing flexible power output adjustment that can be used in wind farm controller design applications.

Since the wind turbine controllers are converted to C code the speed of simulation is increased significantly. Therefore, implementation of a wind farm model with large numbers of wind turbines is possible.

The effect of power set-point reduction on wind turbine operation is investigated. It is shown that by reducing the power capture the aerodynamic loads on the wind turbine structure could also be reduced.

Chapter 5: Development of wind farm Simulation model

IN this chapter, the construction of a wind farm model appropriate for wind farm controller design and analysis is discussed. A wind farm model is required, which is sufficiently detailed and supports sufficiently fast simulation for the iterative task of wind farm controller design. Most of the available models in the literature either contain very simplified dynamics of the wind turbine and its controller or include a very complex wind field model which requires extensive computational capacity [54].

The general requirements specified for the wind farm model developed in this chapter are, thus, summarized as:

- Wind farm model capable of fast simulation with large number of wind turbines (more than 50WTs) for wind farm controller design.
- Wind turbine model dynamics and full envelope controller for load analysis.
- Wind-field model that represents turbulent wind field at farm level and turbine level.
- Wake models to correctly represent wake propagation through the wind farm.
- Wind farm control algorithm that coordinates the wind turbines’ operation.

5.1 Implementation requirements

One of the main challenges in developing a large wind farm model for simulation and control design purposes is computational capacity and simulation time. The simulation time increases as the number of wind turbines increases and the level of complexity in the model increases. Generally, the choice of numerical solver, implicit or explicit, fixed or variable step length, dictates the sampling time and therefore the speed of the simulations. This becomes increasingly an issue, when the stiffness of the system increases, but can be alleviated through appropriate choice of model components.

The wind farm model consists of two main sub-systems which must be modelled according to the defined requirements. The first sub-system consists of the wind turbines. The wind turbine model must be developed in such a way as to increase the simulation speed. Nevertheless, in order to investigate loads on the wind turbine structure, the most significant wind turbine structural modes must be included in the wind turbine model. Since the model complexity directly affects the simulation speed, it is desirable to only represent the structural modes that are absolutely necessary. Specifically, the first blade, tower, and drive train modes are the crucial structural modes that are required to be included in the model. Since the loads are also dependent on the wind turbine controller, a full envelope controller must be included in the model of each turbine.

Moreover, an interface between the wind turbine controllers and wind farm controller is required to establish the required control functionality at the wind farm level. The interface function must be able to adjust the power set-point of a wind turbine controller as required by the wind farm controller.

The second main sub-system is the wind-field model. The wind-field model must include long length-scale aspects that correlate the behaviour 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 are also 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 turbines in the wind farm. However, the loads on each wind turbine, also, depend on higher frequency components in the wind-field. Thereby, the high frequency components of the wind-field must be added locally for each turbine. Similarly, to increase simulation speed, the wind-field model must be developed in such a way as to minimise the calculation effort in the simulation.

Finally in order to coordinate the operation of the wind turbines in a wind farm, a wind farm controller is required to communicate with the wind turbines through the interface controller at each turbine. It is desirable to develop the wind farm control algorithms in C/C++ or MATLAB scripts to enable the implementation of multiple control loops operating with different sampling rates as required.

5.2 Implementation method

The construction of a Simulink wind farm model suitable for wind farm control is discussed here. To meet the requirements in Section 5.1, the structure of the wind farm model is chosen to be that depicted in Figure 5-1.

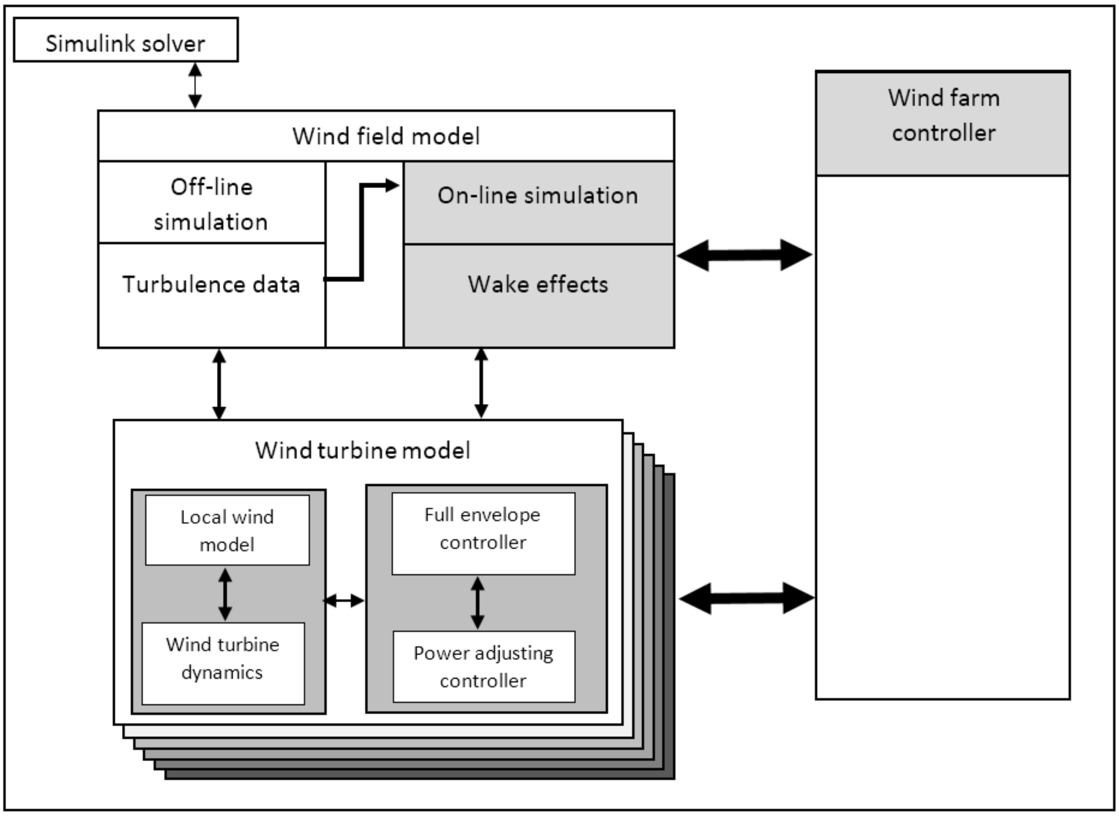


Figure 5-1: Wind farm model block diagram

Overall, the wind farm model is described in continuous time. Nevertheless, in order to increase the simulation speed, the model is divided into discrete and continuous time sections. Simulink provides a useful feature that allows implementation of different part of the system in discrete form in C/C++ code using Simulink S-functions. The sampling time of the MEX_C S-Functions can be defined during the implementation. This approach to the discrete sections, which further increases the simulation speed, is adopted when appropriate.

The correlated components of the wind-field are at low frequency, namely, the low frequency turbulence and wake propagation, whereas the loads on a wind turbine are caused by wind-field components over a far larger frequency range. However, the higher frequency components in the wind-field correlate over a short length-scale and so can be interpreted as local to any particular turbine. This frequency separation is exploited by including only the low frequency turbulence and wakes in the wind-field model in Figure 5-1. The higher frequency turbulence is incorporated into the wind turbine model through a local wind model in the wind turbine model. To minimise the calculation when conducting simulations, a large integration time step can thereby be chosen for the wind-field model e.g. 1 second, whilst a small integration time step is chosen for the wind turbine models e.g. 0.025 second. Furthermore, the data used to represent the low frequency turbulence in the wind-field is pre-calculated and stored in a file from which it is read during simulation. A similar approach is taken to the data used to represent the local wind model at each wind turbine. Adopting a split level simulation in this manner reduces simulation time significantly. To support the split level simulation and further reduce the simulation time, the wind-field model is implemented as a discrete section using a Matlab S-function.

For complete flexibility the wind turbines can be placed at any desired location in the wind farm. The wind-field model in Figure 5-1, is represented by the longitudinal turbulence time series only at each turbine position. However, in order to model the propagation of the wakes through the wind farm, a more detailed representation of the wind-field must be included. Specifically, wake centres, wake diameters and wake deficit models must be included in the model utilising a regular 2D grid, representing the lateral component of the wind-field. The density of the grid points determines 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 and stored in a file to be read during simulation. The wake meandering is represented by calculating the wake centres at any distance behind the wind turbines producing the wake.

The wind turbine model utilized in the wind farm modelling is the Supergen 5MW Exemplar wind turbine model which includes sufficient dynamics and structural modes for relative comparison of the turbine loads arising from the wind farm controller (but not to determine their absolute values sufficiently accurately for turbine component design). The wind turbine model includes a full envelope controller augmented with a PAC acting as an interface between the wind turbine controller and the wind farm controller allowing for dynamic power set-point adjustment of each wind turbine. The PAC alters the power output of the wind turbines by adjusting the torque demand and pitch demand signals of the full envelope controller in response to a power adjustment signal issued by the wind farm controller. The wind turbine model and its controllers are fully explained in Chapter 4.

The wind turbine controller and its associated PAC are discretised, translated to C/C++ code and implemented as a MEX_C S-Function. The implementation of the wind turbine controller is explained in Chapter 4. The default sampling frequency of the controller is set to be 20Hz. The main advantage of implementing the wind turbine controller in C/C++ is the increase in simulation speed, which in turn allows for deployment of more wind turbines in the model.

The final sub-system in the model shown in Figure 5-1 is the wind farm controller. The wind farm control algorithm includes different control objectives and is implemented with a MATLAB S-Function enabling the design of multiple control loops at different sampling rate. The implementation of the wind farm model subsystems are explained in the following subsections.

5.3 Implementation of the Wind-field model

The implementation of the wind-field model is proceeded in two stages as shown in Figure 5-1. A novel approach is adopted to generate longitudinal and lateral turbulence time series with the required characteristics. The wind-field model is developed based on the Shinozuka cross spectral density function algorithm[91] explained by Veers et al in [80]. The wind-field model generates low frequency correlated turbulence time series at the position of each wind turbine in the longitudinal direction to represent a single point wind speed time series at each wind turbine and a series of lateral turbulence time series for wake meandering calculations.

The Veers algorithm was developed [80], for a three-dimensional wind simulator for aerodynamic and structural analysis of wind turbines. The Veers algorithm has been used frequently in the literature and is utilised in many commercial codes such as TurbSim by NREL [92].

The lateral turbulence time series are generated using a 2D grid layout as shown in Figure 5-2. The 2D layout is defined based on the wind-field length, width and grid size. The lateral and longitudinal grid lines represent the columns and rows in the 2D grid layout respectively. The lateral turbulent time series are generated at each grid point defined on the columns in the 2D grid at the hub height whilst the longitudinal turbulence time series are only generated at the position of each wind turbine.

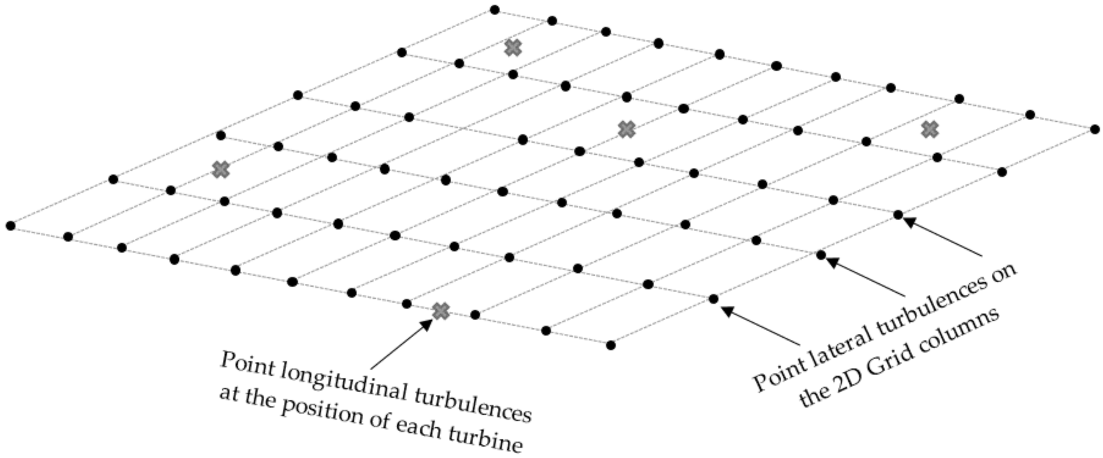

Point longitudinal turbulences at the position of each turbine
Point lateral turbulences on the 2D Grid columns

Figure 5-2: A 2D grid layout illustration for generation of turbulence time series

The lateral turbulence time series are used to calculate wake meandering, by iteratively estimating the centre position of an upwind turbine's wake, at each column in the 2D grid in the wake. Thereby, the wake centre at a distance behind a wind turbine rotor is computed by iteratively calculating and summing up the wake centres at each column between the turbine position and the distance in the wake.

5.4 Procedure of generating the turbulence time series

Veers algorithm generates N correlated turbulence time series according to Shinozuka cross spectral density function method, based on single point PSD functions, and their coherence function for the N wind turbines at their positions in the wind farm.

Following [80], the algorithm generates an $N \times N$ matrix of PSD functions incorporating the coherence between the turbulent wind speeds at the N different points in the wind-field. This $N \times N$ spectral density function matrix,$S(f)$, 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 distance l.

The cross spectral densities between two points A and B separated by distance l is described as a function of the PSD functions at A and B and coherence function between them by

$$Coh_{A,B}(f,l,U_0) = e^{-c_{u,v} f \frac{l}{U_0}} \tag{5-1}$$

where, l is the distance between two points A, B , f is the frequency in rad/sec

The magnitude of the cross spectral density function, $|S_{A,B}(f)|$, is

$$|S_{A,B}(f)| = Coh_{A,B}(f,l,U_0)\sqrt{S_{AA}(f) \cdot S_{BB}(f)} \tag{5-2}$$

The coherence function,$Coh_{A,B}(f,l,U_0)$, describes the correlation between the wind speeds at two points separated by distance l in terms of mean wind speed, frequency and spatial separation [80]. The coherence parameter decay factors for longitudinal and lateral separation, $c_{u,v}$ are defined according to the IEC recommendation by $c_u = 7.1$, $c_v = 4.2$.

Since calculations are conducted numerically, a discrete frequency domain is defined as

$$\Delta f = \frac{2\pi}{n \times Sampling\ time} \tag{5-3}$$

$$f = -n \times \Delta f : \Delta f : (n-1) \times \Delta f \tag{5-4}$$

where $n = length(simulation\ time)$, Δf is sampling frequency, and f is frequency in rad/sec

The PSDs are generated using double sided Kaimal spectrum [93] as

$$s_{u,v}(f) = \frac{1}{3}\sigma_{u,v}^2 \frac{\frac{L_{u,v}}{U_0}}{\left(1 + f\frac{L_{u,v}}{U_0}\right)^{\frac{5}{3}}} \tag{5-5}$$

where, (u, v) represent longitudinal and lateral components, $\sigma_u = T_i\left(\frac{3}{4}U_0 + 5.6\right), \sigma_v = 0.8\sigma_u \ and \ L_u = 340.2, L_v = 113.4$, T_i is turbulence intensity.

The spectral density function matrix,$S(f)$, is then spectrally factorised and a matrix $H(f)$ is determined accordingly. Since $S(f_i)$ is real, the $H(f_i)$ can, also, be real. To do so $H(f_i)$ is determined as follows:

$$H_{11} = S_{11}^{1/2}$$

$$H_{21} = \frac{S_{21}}{H_{11}}$$

$$H_{22} = (S_{22} - H_{21}^2)^{1/2}$$

$$\vdots$$

$$H_{m,k} = \frac{\left(S_{m,k} - \sum_{l=1}^{k-1} H_{m,l}H_{k,l}\right)}{H_{kk}} \tag{5-6}$$

$$H_{kk} = \left(S_{kk} - \sum_{l=1}^{k-1} H_{K,l}^2\right)^{1/2}$$

where, $m = 1,2,..N$, $K = 1,2, \dots, N$, and N is the number of turbines.

The $S_{1,1}, S_{2,1}, \dots$ elements are made periodic; that is,

$$S_{i,j}(f) = S(N+1 : 2N) + S(1 : N) \ i,j = 1,2, \dots number \ of \ turbines \tag{5-7}$$

Using the H matrix as a frequency response function matrix, with input consisting of a vector of N independent white noises as shown in Figure 5-3, yields N independent correlated point wind speeds with the correct spectral properties.

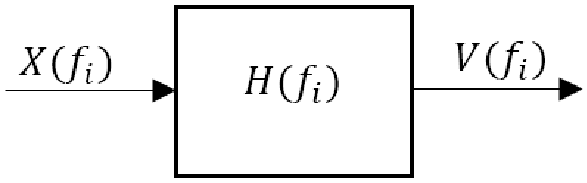


Figure 5-3: Frequency response function with an input vector of unit magnitude white nose and output vector of independent correlated wind speeds

The $X(f)$ vector is obtained by

$$w(t) = \frac{\mathrm{randn(1, simulation\ time)}}{\sqrt{sampling\ time}} \tag{5-8}$$

$$X(f) = \frac{\mathrm{fft(w)}}{N \times sampling\ time} \tag{5-9}$$

where $w(t)$, represents the vector of randomly generated white noises in time domain and $X(f)$ is the transformation of the white noise vector into frequency domain.

$V\,(f)$, the $N \times 1$ vector of Fourier transforms of the correlated turbulent wind speeds is determined using

$$V(f) = H(f) \times X(f) \tag{5-10}$$

Finally, the turbulent wind speed time series are computed by taking the Inverse Fourier transform of the vector $V(f)$ as

$$V_{point}(t) = (N \times sampling\ time) \times ifft\left(V(f)\right) \tag{5-11}$$

Due to the periodicity imposed on $H, V_{point}(t)$ is real as required. The generated point wind speed time series are saved in a data structure in the Matlab work space to be read in during simulation.

5.5 Wake modelling

Wake effects in the wind farms could result in significant array losses [94]. The increased level of turbulent intensity in the wake, also may result in higher loads on wind turbines. It has been shown that coordinated control of wind turbines in a wind farm could reduce the wake effects and improve the array efficiency [10]. Therefore, a suitable representation of the wake effects and its propagation through the wind farm, must be included in the wind farm model for controller design. As explained earlier the most relevant approach for modelling the wake effects for fast wind farm model simulations, is the kinematic method. Among them, the most commonly used explicit wake modelling approach is the Frandsen analytical wake modelling method. With this method wake effects including wake centre, wake diameter, and wake deficits are modelled as a function of distance and thrust coefficient.

Wind turbines could experience either a single wake or multiple wakes of other turbines depending on their position in the wind-field. The Frandesn wake modelling method describes single wake and multiple wakes seen by a wind turbine in a wind farm. The single wake model considers the wake effects generated from only one upwind turbine as seen by a downwind turbine. The multiple wake model combines the wake effects from all upwind turbines experienced by a turbine inside these wakes. In order to compute and apply the wake effects on the wind-field at each turbine position, the first step is to identify upstream wind turbines that produce wakes for each wind turbine in the wind farm. For instance, in a row of 5 wind turbines positioned behind each other and aligned with the wind direction, the second turbine experiences wake effects only from the first upwind turbine in the row, and the fifth turbine may experience a combination of wake effects generated from the turbines in front of it. Since the wake is meandering through the wind-field, to identify the wind turbines affected by the wake of upwind turbines, the wake centre position and the diameter of an upstream turbine at a downstream turbine position, must be estimated. By comparing the wake centre and diameter of an upstream turbine with the position and rotor diameter of downstream turbines, wind turbines in the wake can be identified.

5.6 Wake centre

Wake centre is computed assuming an axisymmetric wake, meandering through the wind farm. Wake meandering is the term used to describe large scale lateral movement of the entire wake downwind. Two main reasons are suggested for this phenomenon in the literature. The first reason, relates the wake movement due to the atmospheric turbulence eddies which are large enough compared to the wake size to transport the wake as a whole. The second idea, relates the wake movement with the instability characteristic in the wake as a result of periodic vortex shedding [95], [27]. Figure 5-4 shows a sketch of the wake meandering phenomenon downwind of a wind turbine, and the position of the wake centre at any distance in the wake, as the wake evolves downstream with the mean wind speed.

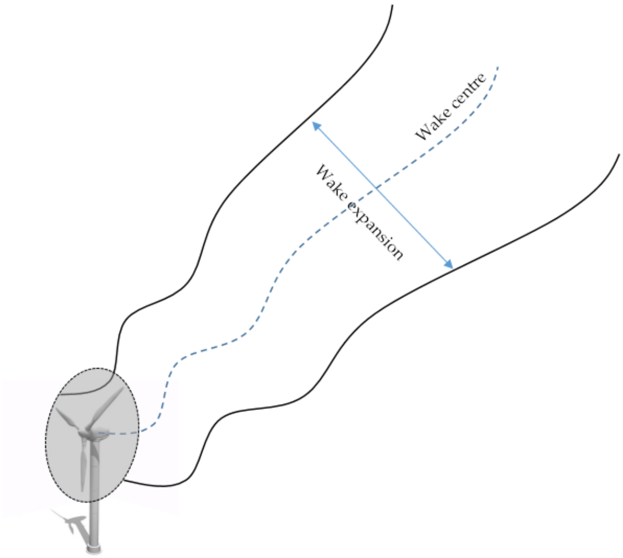


Figure 5-4: Wake meandering, wake centre

In order to understand the wake meandering and wake centre calculations, a two turbine wind farm layout is considered to describe the calculation process.

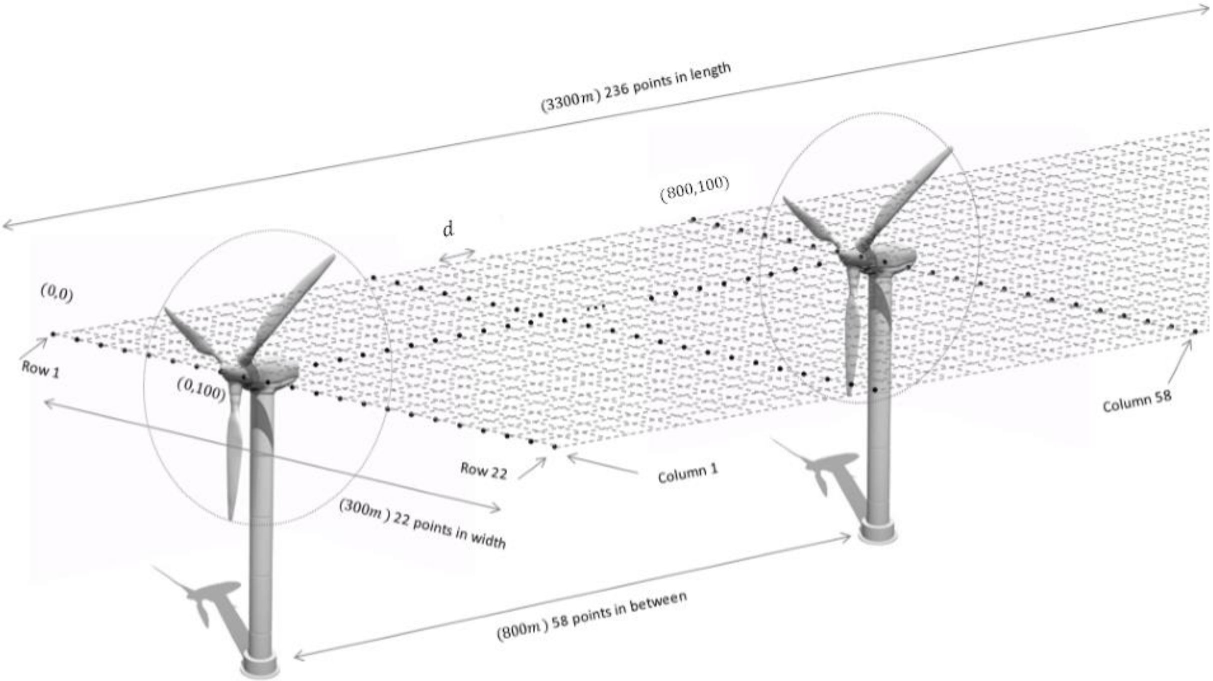

(3300m) 236 points in length
(800,100)
d
(0,0)
Row 1
(0,100)
Row 22
Column 1
Column 58
(300m) 22 points in width
(800m) 58 points in between

Figure 5-5: Wind farm 2D grid layout

Figure 5-5, shows a 2D grid for this wind farm layout. 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 considered to be 3300m and 300m, respectively.

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 in wake centre and meandering calculations. 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 stored in a file 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 are calculated by

$$N_{long} = \frac{wind\ farm\ length}{d} = \frac{3300}{14} \cong 236 \tag{5-12}$$

$$N_{lat} = \frac{wind\ farm\ width}{d} = \frac{300}{14} \cong 22 \tag{5-13}$$

where, $d = V \times T_s$ is the distance between the grid points, V is the mean wind speed ($V = 14ms^{-1}$ is chosen in this case) and T_s is the sampling time ($T_s = 1s$ is chosen in this acse).

The lateral turbulence time series are computed for all the grid points on each columns on the mesh.

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

$$R(x) = \sqrt{R_0^2 + \frac{R_0 x}{2}} \tag{5-14}$$

where R_0 is the rotor radius.

Figure 5-6, illustrates the wake meandering radius at a downstream distance x from a turbine.

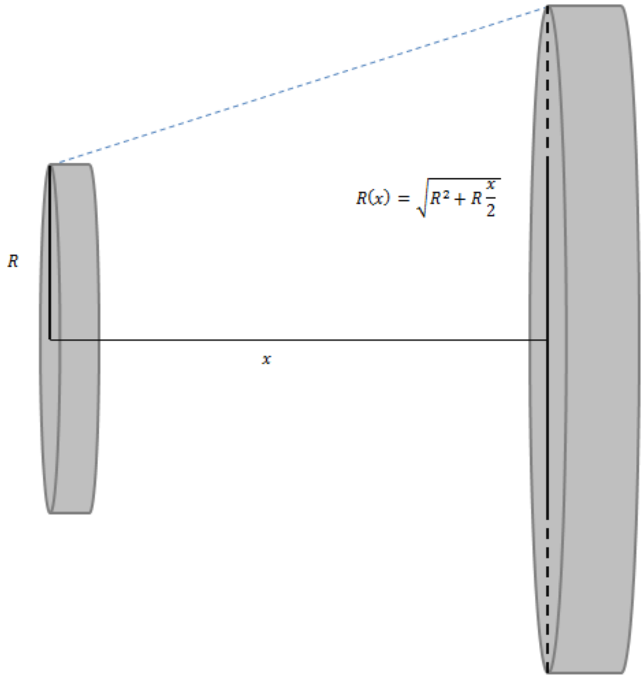


Figure 5-6: Wake meandering distance and wake radius

The turbulence time series corresponding to the lateral grid points inside the wake at a distance x downstream from the rotor are used to compute the wake centre at that distance. The wake centre starts at the centre of the rotor just behind the turbine as shown in Figure 5-7. Therefore, in this case the lateral position of the turbine is the initial wake centre $WC_0 = 100$, and the lateral displacement of the wake on each column between the turbine and distance x behind the turbine is computed by

$$WC = U_{L_{av}} * T_s + WC_0 \tag{5-15}$$

$$U_{L_{av}} = \frac{\sum_{i=1}^{n} u_{L_i}}{n} \tag{5-16}$$

$$WC_0 = WC \tag{5-17}$$

where, $U_{L_{av}}$ is the average of the lateral velocity components inside the wake diameter at each column between the turbine and distance x behind the rotor, n is the number of lateral grid points inside the wake diameter, and T_s is the sampling time. The computed lateral displacement at each column is added to the previous wake centre computed at previous column to determine the wake centre at distance x behind the rotor. This computation is repeated for all columns between the wind turbine and distance x behind thr turbine. The value of the wake centre at previous column updates the initial wake centre value WC_0 for the next iteration of the calculations. Therefore for the wake centre calculation on the next column, the previous computed wake centre is the starting point.

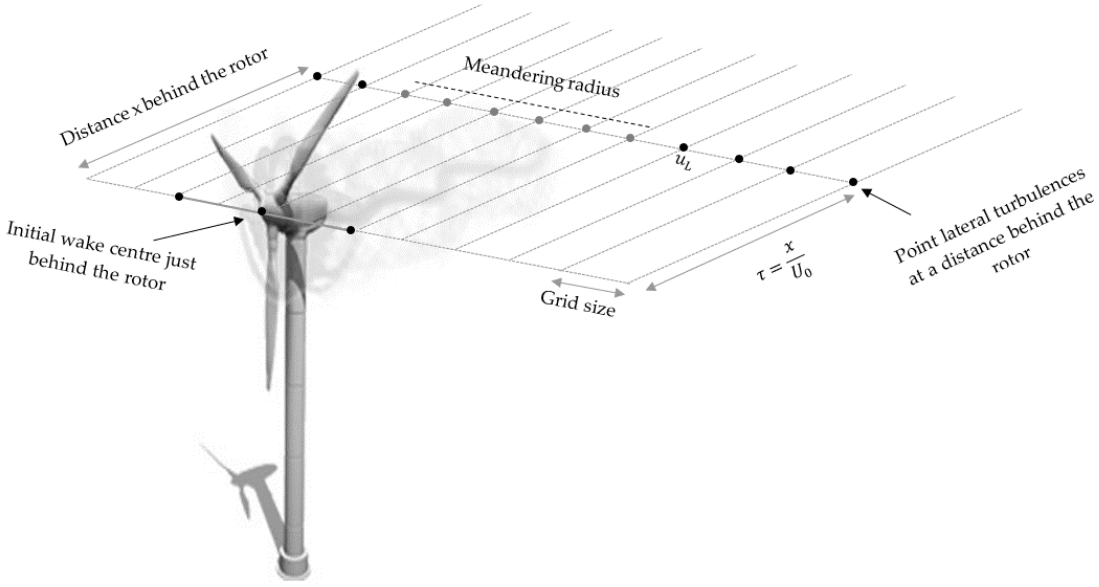

Distance x behind the rotor
Meandering radius
u_L
Initial wake centre just behind the rotor
Grid size
$\tau = \frac{x}{U_0}$
Point lateral turbulences at a distance behind the rotor

Figure 5-7: A 2D grid layout with grid points inside the meandering radius at distance x

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 find the wake centre at the lateral position of the downstream turbine.

5.7 Wake diameter

The wake diameter is also computed based on the Frandsen method described in [81]. The wake is represented as a rectangular distribution of the flow speed. The thrust coefficient is related to the induction factor in the wake by

$$C_T = a(2-a), \qquad a = 1-\sqrt{1-C_T}, \qquad CT < 1 \tag{5-18}$$

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

$$\frac{A_a}{A_0} = \frac{1-\frac{a}{2}}{1-a} \tag{5-19}$$

where, A_0 is the swept area of the rotor and A_a is the cross section area in the wake.

It is assumed that the wake expands immediately after the rotor, therefore for distance $x = 0$ behind the rotor, the wake cross section is assumed to be A_a. Combining (5-18) and (5-19), there is

$$A(x=0) = A_a = \beta A_0 \tag{5-20}$$

$$\beta(t) = \frac{1}{2}\frac{1+\sqrt{1-C_{T_i}(t-\tau)}}{\sqrt{1-C_{T_i}(t-\tau)}} \tag{5-21}$$

where β is a wake factor

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

$$\frac{U}{U_0} \approx 1 - \frac{1}{2} C_T \frac{A_0}{A} \tag{5-22}$$

In the Frandsen method, an expression for the wake expansion as a function of distance is adopted in such a way that equation (5-22) 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 by

$$WD_j\left(x_{i,j}, t\right) = \left(\beta(t)^{\frac{k}{2}} + \alpha \frac{x}{D_0}\right)^{\frac{1}{k}} D_0 \tag{5-23}$$

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

5.8 Wake deficit

Once the wake centre and diameter are computed, 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 wind turbines producing the wakes are identified. The wake deficit for each turbine is computed either, as a single wake or a combination of multiple wakes from upstream turbines, 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 (5-23) for the wake expansion area as a function of distance, the wake deficit stemming from a single turbine is

$$Def_j(t) = 1 - \frac{1}{2}\frac{CT_i(t-\tau)D_0^2}{WD_j(t)^2} \tag{5-24}$$

where, $Def_j(t)$ is the velocity deficit at turbine j due to the wake from turbine i and $WD_j(t)$ is the wake expansion diameter of turbine i seen by turbine j at time t computed from (5-23). Since it takes some time for the wake to propagate to the downstream turbine, the deficit delay is also considered in the calculations. Therefore, deficit is computed based on the thrust coefficient of the upstream turbine at time $t - \tau$; where τ is deficit delay between the turbines. To apply the deficit delays in the calculations, wind turbines' thrust coefficient are delayed appropriately to represent the deficit delay between the turbines. Thus, a suitable propagation of the wake through the wind farm is ensured.

The single wake calculations can be expanded to consider multiple wakes affecting a turbine. In a wind farm with a large number of wind turbines, a combination of the wakes from upstream turbines affect the wind turbines in the wake can be shown in Figure 5-8. Therefore, it is desirable to combine the effect of multiple single wakes at each turbine in the simulations.

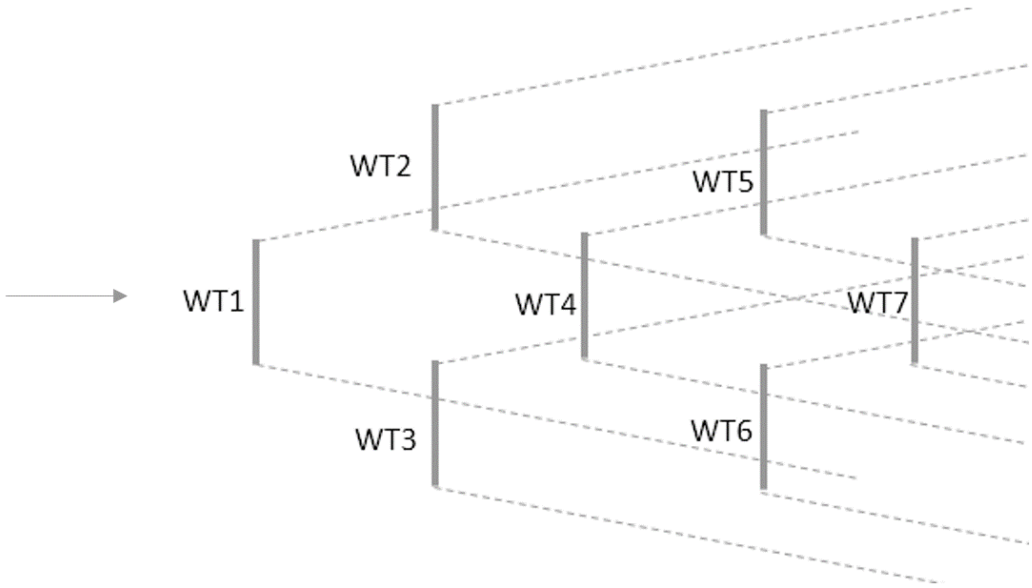


Figure 5-8: Multiple wakes effect

Various methods are available to combine the multiple single wake effects at each turbine. The main methods such as Sum of squares of velocity deficits, Energy balance, Geometric superposition, linear superposition are frequently reported in the literature [96].

The multiple wake deficit combination based on the Frandsen method is computed considering wind turbines in a single row behind each other. 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. In this method the multiple wake deficit is computed considering the momentum conservation in the control volume by

$$\rho A_{n+1}\, U_{n+1}(U_0 - U_{n+1}) = \rho A_n\, U_n(U_0 - U_n) + T \tag{5-25}$$

$$T = \frac{1}{2}\rho A_0 C_T U_0^2 \tag{5-26}$$

Substituting thrust force expression into equation (5-25) yields

$$\frac{U_{n+1}}{U_0} = 1 - \left[\frac{A_n}{A_{n+1}}\left(1 - \frac{U_n}{U_0}\right) + \frac{1}{2}\frac{A_0 C_T}{A_{n+1}}\left(1 - \frac{U_n}{U_0}\right)\right] \tag{5-27}$$

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 $n + 1$.

$$Def_{n+1} = 1 - \left[\underbrace{\frac{WD_n(t-\tau)^2}{WD_{n+1}(t-\tau)^2}\left(1 - Def_n(t-\tau)\right)}_{effect\ of\ other\ turbines} + \underbrace{\frac{1}{2}\frac{C_{T_n}(t-\tau)D_0^2}{WD_{n+1}(t-\tau)^2}\left(1 - Def_n(t-\tau)\right)}_{effect\ of\ closest\ turbine}\right] \tag{5-28}$$

As shown in Figure 5-9 the wake effects at turbine WT_{n+1} stems from the wake effects from WT_n and all other turbines in the row. Equation (5-28) 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} .

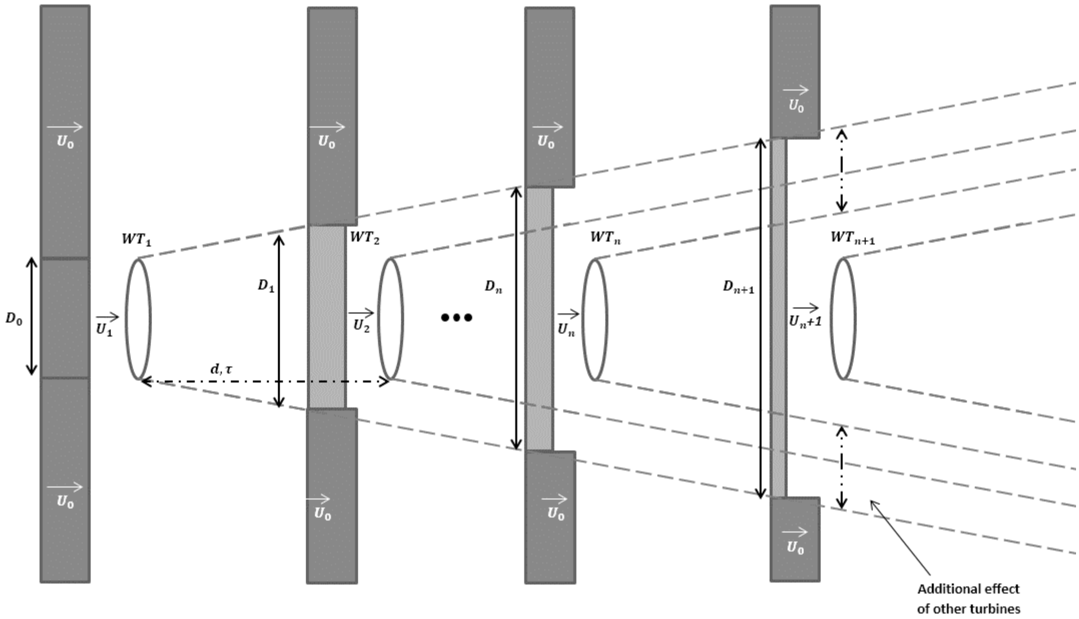


Figure 5-9: Wake deficit effects on the turbines in the wake

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 behind each other, another method for combining multiple wakes should 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 other 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 wakes from the upwind turbines as shown in Figure 5-10. Therefore, at each turbine the overlapping wake shadow area of the upwind turbines is computed and only a portion of the wake deficit from the upstream turbine is considered in the wake combination calculations.

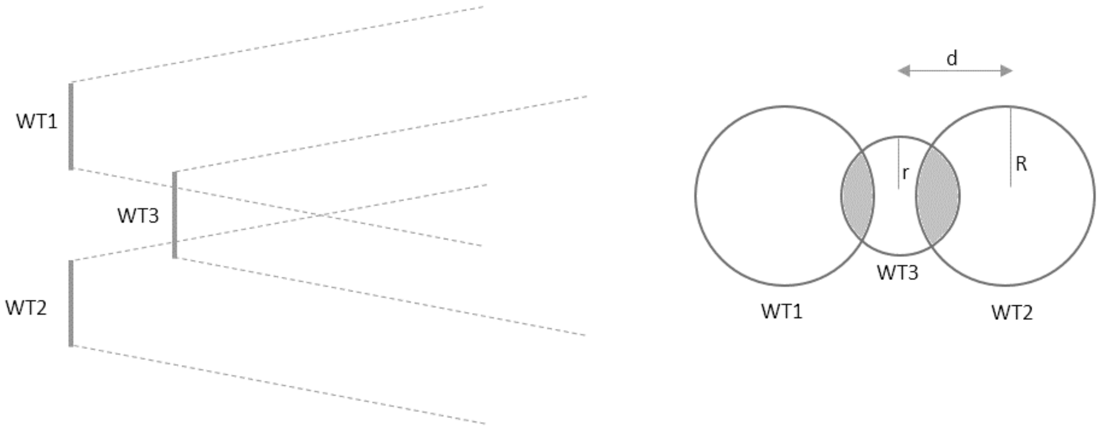


Figure 5-10: Partial wake shadowing effect

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

$$A_{shadow} = r^2 \cos^{-1}\left(\frac{d}{2r}\right) - \frac{d}{4}\sqrt{4r^2 - d^2} + R^2 \cos^{-1}\left(\frac{d}{2R}\right) - \frac{d}{4}\sqrt{4R^2 - d^2} \qquad (5\text{-}29)$$

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.

$$def_j = \sqrt{\sum_{\substack{j=1 \\ j\neq i}}^{n} def_{i,j}^2 \frac{A_{shadowi,j}}{A_0}} \qquad (5\text{-}30)$$

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

The calculations of the wake effects, are made in online simulation, using Matlab S-Functions in the wind-field model. The computed wake deficits at each sampling time are added to the mean wind speed at each turbine to realise the wake effects during the simulation.

5.9 Wind turbine model

According to the defined requirements, the wind turbine model must include adequate detail for load analysis and controller design and also should be suitable for fast simulation. The wind turbine model utilized in the wind farm modelling is a 5MW Supergen Exemplar model with relevant fidelity level for controller design and fast simulation. The wind turbine model includes the most significant dynamics and structural modes for load analysis as explained in Chapter 4. This model is implemented into discrete and continuous time sections to achieve fast simulation as shown in Figure 5-11.

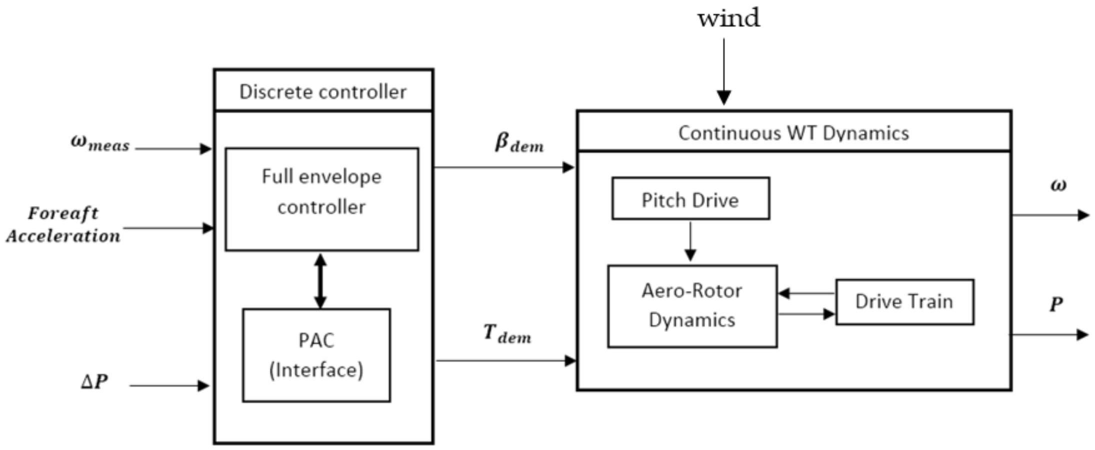


Figure 5-11: WT model structure

The wind turbine controller comprises of a full envelope controller, including additional feedback loops such as the tower feedback loop and the drive train filter to effectively alleviate the structural loads, and an interface for dynamically adjusting the power set-point of the wind turbine according to a power set-point change request from wind farm controller. The PAC acts as a feedforward modification to the full envelope controller in order to apply demanded power set-point change in a safe and quick operating manner.

The development of the PAC is given in [83]. PAC is used as an interface between the wind farm controller and the wind turbines' full envelope controller to achieve dynamic coordination of the wind turbines in the wind farm. PAC translates the power set-point request from the wind farm controller into additional torque demand and pitch demand signals according to the wind turbine operating condition. The pitch demand and torque demand outputs of the full envelope controller are adjusted by these additional signals to achieve the power set-point change request at each turbine. The wind turbine controller, including the PAC, discretisation and its implementation in C code is explained in Chapter 4.

5.10 High frequency turbulence data points

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

An algorithm is developed in Matlab to generate the high frequency components for the turbulence based on a random normal process. The random high frequency data points are generated using Matlab normal distribution random generator $normrnd(mue, sige)$ with the mean and the square of the standard deviation defined as

$$\mu = \frac{\sinh\big(a(t-t_i)\big)\,x\big(t_f\big) + \sinh\Big(a\big(t_f-t\big)\Big)\,x(t_i)}{\sinh(a(t_f-t_i)} \tag{5-31}$$

$$\delta^2 = 2\frac{\sinh\big(a(t-t_i)\big)\sinh\Big(a\big(t_f-t\big)\Big)}{\sinh(a(t_f-t_i)}\;b^2/(2a) \tag{5-32}$$

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

Consider two consecutive times, t_i and t_f, for the low resolution time series with values $x(t_i)$ and $x(t_f)$. These define an interval between t_i and t_f. To generate N intermediate values for a high resolution time series, this interval is divided into $(N+1)$ sub-intervals of length $(t_f - t_i)/(N+1)$. The first intermediate point is at $t = t_i + (t_f - t_i)/(N+1)$. The increment in the value of the time series at time, t, relative to the value at time, t_i, is chosen from a probability distribution with mean and variance as in (5-31) and (5-32), thereby, determining the value, $x(t)$. With $x(t_i) = x(t)$, the second intermediate value is obtained similarly. The procedure is repeated until all the intermediate values have been obtained. In this manner a higher resolution time series is obtained with the correct statistical properties. Figure 5-12 illustrates the high resolution time series generated from a low resolution time series with the intermediate values indicated by red dots.

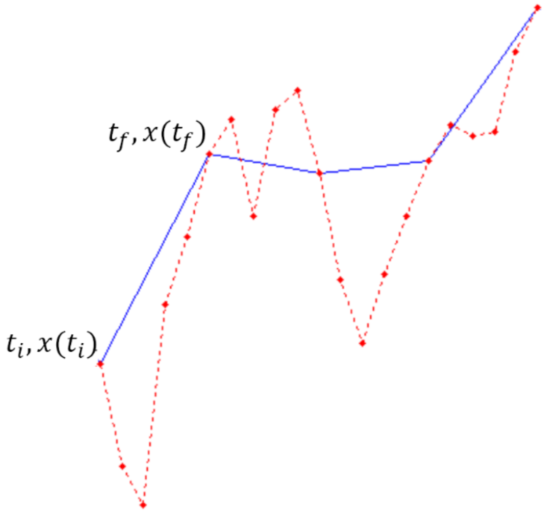


Figure 5-12: Dryden interpolation for high frequency data points

Figure 5-13 and Figure 5-14 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. As can be seen the high frequency components are randomly generated between each two points in the low frequency data with correct spectrum.

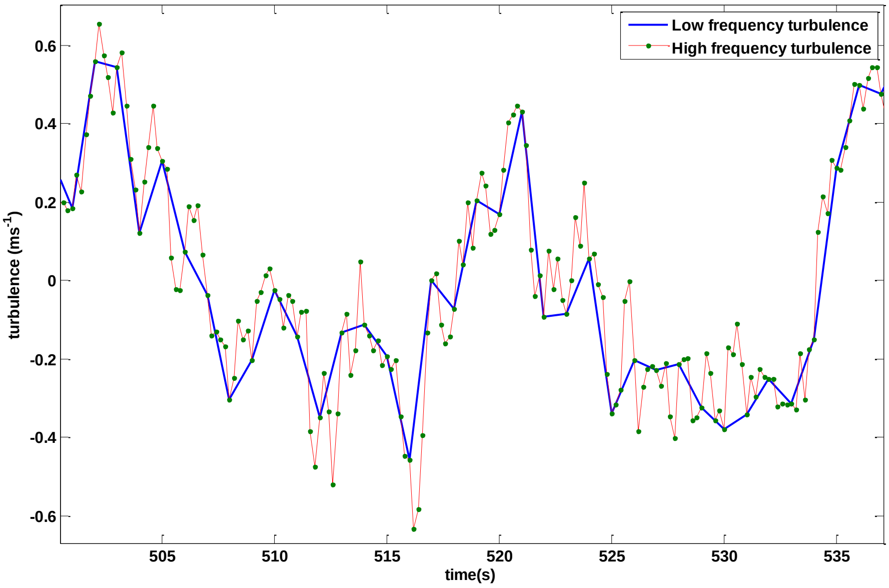


Figure 5-13: High frequency turbulence

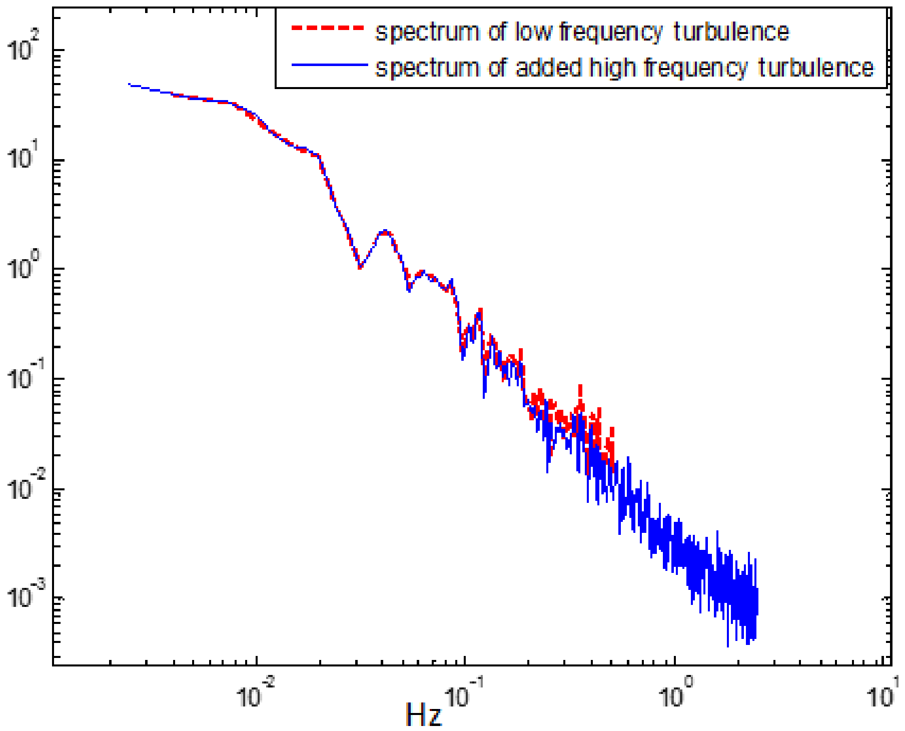


Figure 5-14: Spectrum of the low frequency and high frequency turbulence time series

5.11 Generating the effective wind speed time series

In [84], the interaction between the rotor and the stochastic wind-field is modelled by simple ordinary differential equations. The rotor effective wind speed is obtained by spatially filtering the point wind speed over the swept area of the rotor. This effective wind speed is modelled in such a way as to produce the same forces and torques on the rotor as the wind-field.

To generate the effective wind speed at each turbine, the generated turbulence time series from the previous stage are filtered using the spatial filter described in [84],[85].

The perturbations in the torque are related to the point wind speed using the model shown in Figure 5-15. The spatial filter utilised in the model is

$$f(s) = \frac{(\sqrt{2} + \sigma s)}{(\sqrt{2} + \sqrt{a}\sigma s)(1 + \sigma s/\sqrt{a})} \tag{5-33}$$

where, $\sigma = \frac{\gamma R}{\bar{V}}$

Figure 5-16 illustrates the input and output of the spatial filter model.

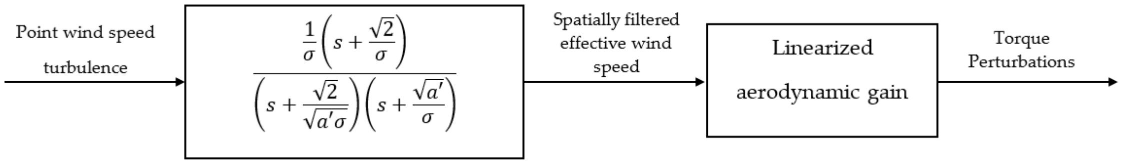

Point wind speed turbulence
$\frac{\frac{1}{\sigma}\left(s+\frac{\sqrt{2}}{\sigma}\right)}{\left(s+\frac{\sqrt{2}}{\sqrt{a'}\sigma}\right)\left(s+\frac{\sqrt{a'}}{\sigma}\right)}$
Spatially filtered effective wind speed
Linearized aerodynamic gain
Torque Perturbations

Figure 5-15: Spatial filter

The combination of the low and high frequency turbulent time series are passed through the filter to obtain the effective wind speed. The output of the filler is added to the mean wind speed and the wake deficit at each turbine during simulation to obtain the effective wind speed influenced by the wake effects at each wind turbine.

Figure 5-16 illustrates an example of input point wind speed to this spatial filter and its corresponding output effective wind speed. The effective wind speed time series are saved in files to be read during the simulation.

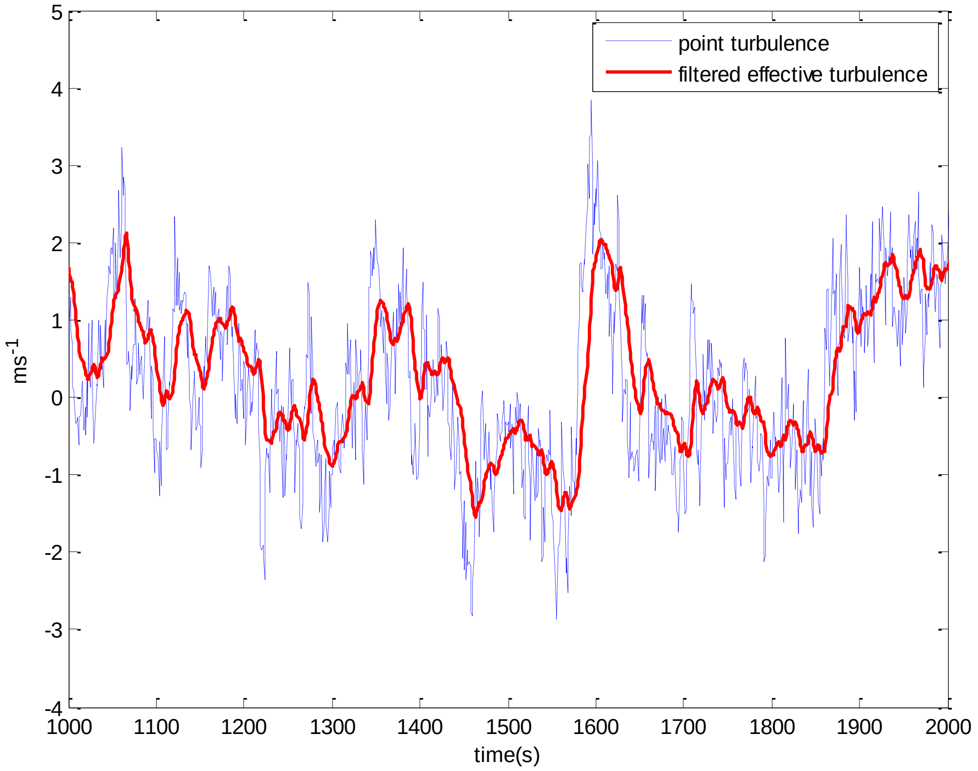


Figure 5-16: Filtered effective turbulence

5.12 Wind farm modelling tool in Matlab

In order to ease the process of developing a wind farm model with a large number of wind turbines (50>), a Matlab script is developed to build a wind farm model according to user defined parameters automatically.

The wind farm builder Matlab script, utilizes various Matlab functions and Simulink libraries to construct different sections of the model. A model library is implemented in Simulink that includes different components required by the construction tool to build a wind farm model. The model library contains a 5MW Supergen exemplar model, a wind turbine template for user defined wind turbine models, and a wind farm template for developing the wind farm model as shown in Figure 5-17.

Once the wind farm generation function is executed the wind farm model builder requests the required wind farm model parameters from the user. A wind farm template then will be copied from the model library into the new wind farm model. According to the defined number of wind turbines, the wind turbine model in the model library will be automatically copied for as many as wind turbines are required and will be populated into the turbines model template in the new wind farm model. Furthermore, the ports on the wind farm subsystems are automatically connected to complete the wind farm model generation.

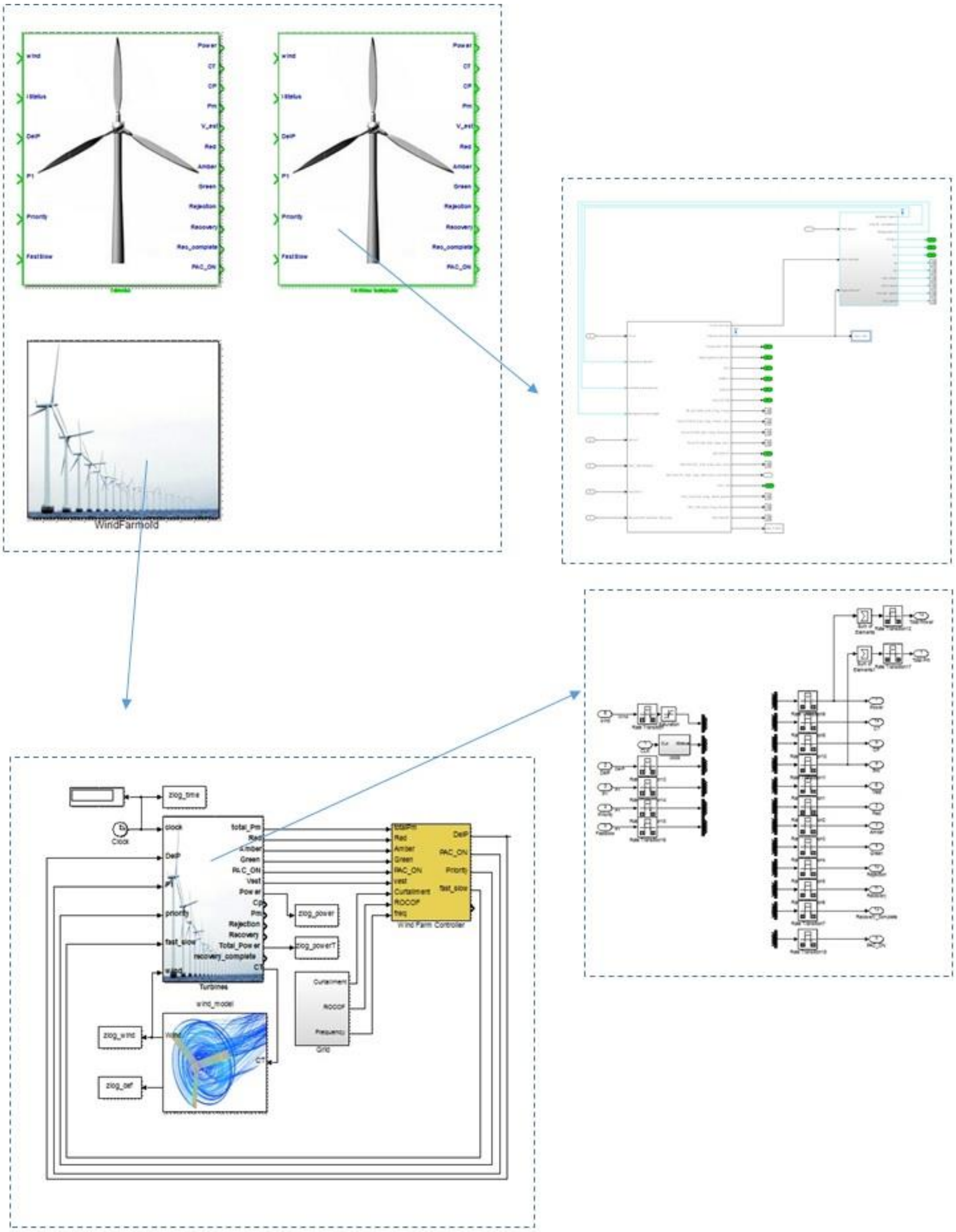


Figure 5-17 wind farm model layout

The available wind turbine model template in the library can be used to include a user defined wind turbine model if required. The wind turbine model template consists of necessary input/output ports that are required by the Matlab script to create the model.

The wind-field and wind farm parameters, specifically, wind turbine position, mean wind speed, turbulent intensity, sampling time, wind-field length and width must be defined by the user during the model development process.

Subsequently, the Matlab script calls various methods in the code, to generate the wind-field and construct the wind farm model. The wind-field generation function, spatial filter, high frequency turbulence data interpolation function and wind turbine cluster construction function are the methods that the main function in the code utilises to construct a full wind farm model.

The wind-field generator function generates the effective wind speed time series as explained in this chapter and saves the wind-field parameters and data points in a data structure in the Matlab current directory.

The wind farm model builder then saves the created model in the current directory and loads the wind farm model and its parameters to start the simulation.

5.13 Concluding remarks

In this chapter, the development process of a wind farm model for fast simulation and controller design is presented. The wind farm model includes a wind-field model which is fast enough for simulation and detailed enough for wind turbine load analysis. The wind-field model includes the model of wake effects which are used during the simulation. The wind farm model contains a wind farm control algorithm that is developed for coordinating control of the wind turbines in the farm. The wind farm model is divided into continuous and discrete sections where appropriate to increase the simulation speed. The model runs on the basis of multiple sampling rate simulation that allows for flexible wind farm controller design. The wind farm model can be used to investigate different power reference tracking objectives of the wind farm controller.

Chapter 6: Wind farm controller design

IN this chapter a wind farm control algorithm is developed with the main objective of dynamically coordinating the wind turbines' power set-points. The wind farm control algorithm contains different control paths corresponding to different control objectives. The main objective of the wind farm controller is to provide power reference tracking. Additionally, primary frequency response is also included in the control routine. Droop control and synthetic inertia response are implemented alongside the power reference tracking objective of the controller. The execution of each control path depends on satisfaction of the conditions specified for that path. The wind farm controller execution starts with performing the power reference tracking control mode of the algorithm. At the same time, the primary frequency control routine checks the measured power system frequency against a predefined threshold. Once the frequency exceeds the thresholds, the wind farm controller performs the primary frequency response accordingly.

The wind farm control algorithm, coordinates the wind turbines' operation in different control modes on the basis of the PAC supervisory flags. The supervisory flags indicate the operating status of individual wind turbines, and are used as decision making variables. These flags are produced by the PAC of each wind turbine, to indicate the region where the turbine is operating. As explained in Chapter 4, the PAC includes the supervisory rules to prevent the wind turbine operating outwidth the defined safe operating region. The supervisory rules including, black limits and traffic light limits, are considered in the implementation of the wind farm control algorithm. The traffic light rules are applied for most control objectives with the exception of synthetic inertia response.

Figure 6-1 depicts a specific wind farm control arrangement, which is used to develop a wind farm control algorithm for different control objectives. As shown, in this arrangement, the wind farm controller communicates with the wind turbines, through the flags produced by their PAC. Power set-point increments $\left(\Delta p_{i=1,N}\right)$ are computed by the wind farm controller, corresponding to the requested wind farm reference power, and dispatched between the wind turbines while considering their operating status.

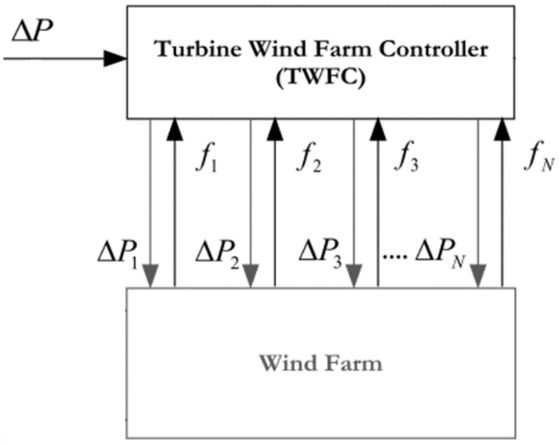


Figure 6-1: The wind farm control arrangement [97]

Since only the status flags, produced from the feed-forward interface controller (PAC), are used for dispatching the wind farm reference power (ΔP) between the turbines, no significant feedback loop between the wind farm controller and each individual wind turbine is introduced.

Under the PAC supervisory rules, each wind turbine can only provide a specific amount of power adjustment level, according to the region in which the turbine is operating. Thus, the wind farm control algorithm must dispatch the farm reference power between the turbines in agreement with these supervisory rules and their corresponding power adjustment levels. Here the maximum adjustable power allocated for each operating region within the traffic-light regimes, in this control algorithm, are $0MW$ for the Red region, $\pm 0.2MW$ for the Amber region, and $\pm 0.5MW$ for the Green region. The wind farm controller utilises these power adjustment levels and the status flags of individual turbines to compute the required power adjustment level for each wind turbine accordingly.

Figure 6-2 shows a flowchart corresponding to the implementation of the wind farm control algorithm for different control objectives including power reference tracking, synthetic inertia response and droop control. As can be seen, the control algorithm consists of two control paths. The droop control path includes the power curtailment control function and the droop control function. The synthetic inertia control path is implemented to operate simultaneously with the droop control path and includes a decision making routine to initiate the synthetic inertia response function corresponding to large rate of change of frequency.

The grid frequency measurement and its corresponding rate of change of frequency are defined as decision making variables in the wind farm control algorithm. Once a deviation from the nominal frequency is detected, the control algorithm computes the corresponding rate of change of frequency (ROCOF) and performs the relevant control functions in the algorithm according to the requirements. If the frequency deviation is within an acceptable threshold, the power curtailment control mode of the controller is realised. The power curtailment level is defined to achieve 5% spinning reserve for primary frequency response [98], [99]. The traffic-light rules are applied for the curtailment control mode. In the case of large frequency deviations, outwith the acceptable thresholds with slow ROCOF, the droop control function computes the corresponding power adjustment level for compensating the frequency deviation. If the rate of change of frequency exceeds a predefined threshold, the control algorithm switches to the synthetic inertia control mode and computes the power adjustment level corresponding to the wind farm synthetic inertia response accordingly.

Additionally, a power set-point dispatch function is implemented, which operates interactively with the other control functions in the algorithm, to distribute the required wind farm reference power between the turbines according to their operating status. Once the required adjustment power level for each control mode (synthetic inertia, droop control, or power curtailment) is computed, the power set-point dispatch unit checks the availability of the wind turbines' PAC, and distributes the computed adjustment power demand between the turbines according to their operating status.

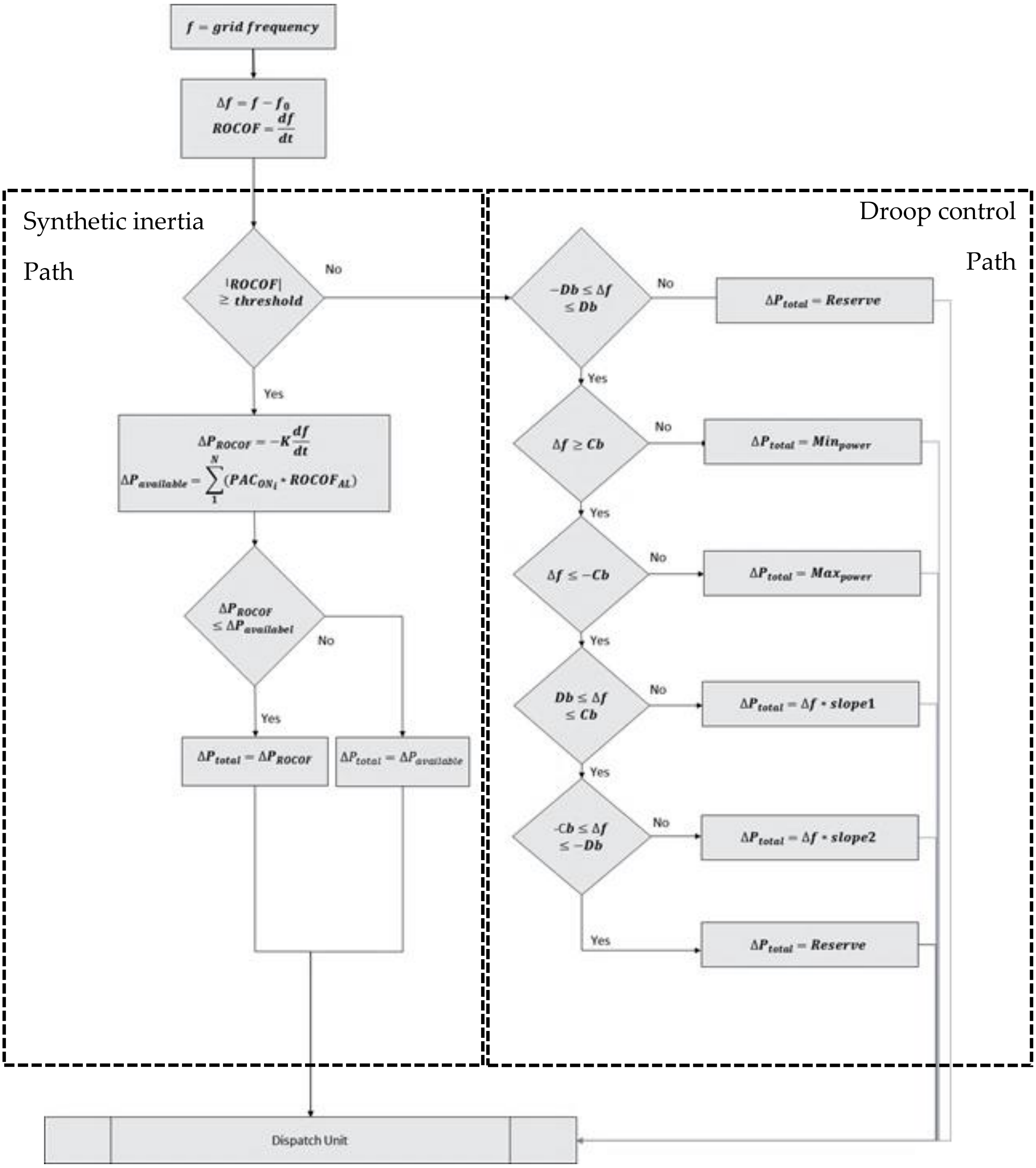

$f = grid\ frequency$
$\Delta f = f - f_0$
$ROCOF = \frac{df}{dt}$
Synthetic inertia Path
Droop control Path
$|ROCOF| \geq threshold$
No
Yes
$\Delta P_{ROCOF} = -K\frac{df}{dt}$
$\Delta P_{available} = \sum_{1}^{N}(PACON_i * ROCOF_{AL})$
$\Delta P_{ROCOF} \leq \Delta P_{availabel}$
No
Yes
$\Delta P_{total} = \Delta P_{ROCOF}$
$\Delta P_{total} = \Delta P_{available}$
$-Db \leq \Delta f \leq Db$
No
$\Delta P_{total} = Reserve$
Yes
$\Delta f \geq Cb$
No
$\Delta P_{total} = Min_{power}$
Yes
$\Delta f \leq -Cb$
No
$\Delta P_{total} = Max_{power}$
Yes
$Db \leq \Delta f \leq Cb$
No
$\Delta P_{total} = \Delta f * slope1$
Yes
$-Cb \leq \Delta f \leq -Db$
No
$\Delta P_{total} = \Delta f * slope2$
Yes
$\Delta P_{total} = Reserve$
Dispatch Unit

Figure 6-2: Flowchart for the wind farm control algorithm including droop and synthetic inertia control

6.1 Implementation of the power set-point dispatch routine

The power set-point dispatch unit distributes the required farm power adjustment ΔP_{dem} between turbines according to their operating status as

$$\Delta p_i = -\Delta P_{dem}\frac{P_{traffic-light_i}}{P_{total}} \tag{6-1}$$

where, $P_{traffic-light_i}$ is the adjustment power allocated for turbine i at its operating region in traffic light regime, P_{total} is the sum of $P_{traffic-light_i}$ of all turbines, and ΔP_{dem} is the required wind farm power adjustment level (e.g. curtailment level).

Since, all wind turbines may not be available to contribute to the wind farm power adjustment operation (e.g. operating at wind speeds lower than the PAC wind speed operating limit of 6.5ms^{-1}), the required wind farm power demand ΔP_{dem} ,therefore, may be adjusted according to the total available power adjustment level at the wind farm. As a result, the delivered adjustment power level may differ from the requested wind farm reference power. The possible level of power adjustment, which can be delivered at different control modes, is computed according to the adjustment power level assigned to each turbine at each control mode. The wind farm power adjustment level in the droop control and curtailed control mode is calculated by

$$\Delta P_{dem} \leq \sum_{i=1}^{N} \left(p_{curtailed_i}\right) * PAC_{ON_i}, p_{curtailed_i} = p_{green} \; or \; p_{amber} \; or \; p_{red} \qquad (6\text{-}2)$$

where, $p_{green} = 5MW, p_{amber} = 0.02MW,$ and , $p_{red} = 0MW$, are the adjustment power levels available for WT_i in the traffic light region where it is operating, PAC_{ON_i} is a binary condition to check if WT_i power adjusting controller is available.

In the case of synthetic inertia, the wind farm power adjustment level is computed by

$$\Delta P_{dem} \leq \sum_{i=1}^{N} \Delta p\_synthetic * PAC_{ON_i} \qquad (6\text{-}3)$$

where $\Delta p\_synthetic$ is the adjustment power level for each turbine for the synthetic inertia response (i.e. 0.5MW in this case).

In the synthetic inertia response, the wind farm control algorithm requests the priority flag set from each wind turbine's PAC. Ultimately, the PAC either accepts or refuses the wind farm controller requests according to its supervisory rules.

The control algorithm is used in the following sections to test the wind farm model and wind farm controller performances.

6.2 Implementation of the synthetic inertia response routine

The synthetic inertia control mode of the wind farm controller is implemented to operate simultaneously with the droop control path. The synthetic inertia response is triggered once the rate of change of frequency exceeds a predefined threshold. In the literature, it is a common practice to use the effective rate of change of frequency by averaging the measured rate of change of frequency over a time period, normally $100ms$ [100].

$$\frac{df}{dt} = \frac{1}{5}\sum_{i=1}^{5} \frac{\Delta f_i}{\Delta t_i} \tag{6-4}$$

where, Δf_i is the measure frequency at each sampling time $\Delta t_i = 20ms$.

To implement the synthetic inertia response, again the swing equation (6-5) is used. The power adjustment level required for synthetic inertia response is determined using equation (6-5). In order to trigger the synthetic inertia function, a condition check routine monitors the averaged ROCOF. If the rate of change of frequency is within an acceptable threshold, the droop control function, compensates for the slow deviations in the grid frequency. A droop control curve is defined to compute the required wind farm power adjustment level, for compensating the slow frequency deviations from the nominal. In the synthetic inertia response control mode, the required power adjustment level of the wind farm is computed proportional to the rate of change of frequency according to equations **(6-5)** to **(6-10)**. The traffic-light rules are ignored in this control mode to provide maximum possible response to the grid under-frequency events. The swing equation is written as

$$J\omega_{grid}(t)\frac{d\omega_{grid}}{dt} = P_{mec}(\mathrm{t}) - P_{elc}(t) \tag{6-5}$$

where, ω_{grid} is the system nominal frequency, and P_{mec}, P_{elc} represent the mechanical power and electrical power of the generator.

The speed of generating nominal power by extracting only the kinetic energy stored in the generator's rotating mass is related to the rotor inertia by a parameter known as the inertia constant H

$$H = \frac{J\omega_n^2}{2S_n} \tag{6-6}$$

where, S_n is the nominal apparent power of the generator, ω_n is the nominal rotor speed.

Therefore, the level of inertia response according to the swing equation in terms of inertia constant is

$$J = \frac{2S_nH}{\omega_n^2} \tag{6-7}$$

$$\underbrace{\frac{2S_nH}{\omega_n}}_{K}\frac{d\omega}{dt} = \underbrace{P_{mec} - P_{elc}}_{\Delta P} \tag{6-8}$$

The total additional power ΔP required for the inertia response is then proportional to the ROCOF by a constant gain $K = \frac{2S_nH}{\omega_n}$.

The S_nH term of the constant gain is computed as the equivalent value for all the machines in a wind farm. Therefore, the power adjustment required in the synthetic inertia by the wind farm is computed by

$$K = \frac{2\sum_{i=1}^{N} S_{n_i}H_i}{\omega_n} \tag{6-9}$$

$$\Delta P = K\frac{df}{dt} \tag{6-10}$$

The inertia constant is computed from Equation **(6-7)**. The 5MW Supergen wind turbine parameters required for calculating the inertia constant are given in Table 6. 1 and the inertia constant for a single turbine is

$$H = \frac{3.53715e^7 * 1.237^2}{2 * 5e^6} = 5.4124s \tag{6-11}$$

5MW WT Characteristics

Rotor inertia	J	$3.53715e^7\ kgm^2$
Nominal rotor speed	ω_n	$1.237\ rad/s$
Nominal power	S_n	$5MW$
Nominal frequency	f_0	$50Hz$

Table 6. 1: 5MW Supergen wind turbine parameters

As mentioned earlier, the maximum achievable adjustment power level is bounded by the total adjustment power level available for the synthetic inertia response. Since the traffic light rules are ignored in this control mode, only the black boundary defined by the PAC supervisory rules applies. If the black boundary limits are reached, the corresponding turbine's PAC enters the recovery process. During the PAC recovery, the power adjustment level available at these turbines is zero. Therefore, the total adjustment power available at the farm level reduces. If the synthetic inertia response power demand is higher than the total power adjustment power level in the farm, the wind farm controller adjusts the synthetic inertia reference power to achieve a sensible power adjustment at the farm level.

$$\Delta P_{SIR} = \sum_{i=1}^{N} \Delta p\_inertia_i * PAC\_ON_i \qquad (6\text{-}12)$$

where, ΔP_{SIR} is the total adjustment power level available for the synthetic inertia response control mode, $\Delta p\_inertia_i$ is the power adjustment level at turbine i for synthetic inertia response (in this case $0.5MW$), and PAC_ON_i is a condition check that determines if each individual wind turbine is available for power adjustment.

The ROCOF setting is defined to only react to the $\left|\frac{df}{dt}\right| \geq 0.5Hz/s$ as recommended in the UK Grid code[101].

6.3 Implementation of droop control routine

The droop control routine in the wind farm control algorithm includes a power reserve (curtailment) control mode, to provide 5% to 10% spinning reserve relative to the available power, at any given wind speed for primary frequency response. Generally, a droop curve with 5% slope is used to provide droop control [102], [103].

The synthetic inertia and droop control parameters used in the wind farm control algorithm are defined in Table 6. 2 and Figure 6-3.

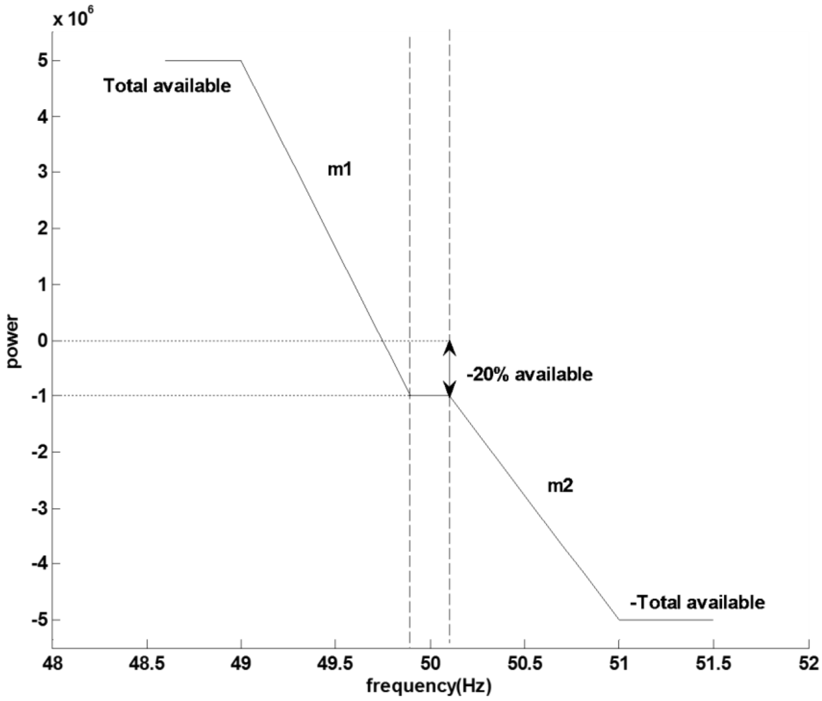


Figure 6-3: Droop curve

Variables	
$ROCOF$	Rate of change of frequency
ΔP_{ROCOF}	Power computed for the ROCOF events
$\Delta P_{available}$	Total available adjustment power level for inertia response
ΔP_{total}	Total wind farm reference power
Db	Droop curve Dead-band limit
Cb	Droop curve control-band
$Reserve$	Curtailment power level
K	Inertia response constant
$Threshold$	ROCOF threshold

Table 6. 2: Control algorithm variables

Figure 6-3 shows the droop curve that is implemented in the wind farm control algorithm with specific dead-band and droop slope characteristics. The maximum and minimum power in the curve are computed based on the sum of adjustment power level available at each turbine according to the traffic light regime. Initially, all turbines are operating in the Green region with available adjustment power of $\pm 0.5MW$. The slopes of the droop curve are designed to be adjusted dynamically according to the total available adjustment power level at any time.

$$m1 = \frac{P_{available} * (1 + \%reserve)}{(50Hz - contrlband) - (50Hz - deadband)} \tag{6-13}$$

$$m2 = \frac{P_{available} * (1 - \%reserve)}{(50Hz + deadband) - (50Hz + contrlband)} \tag{6-14}$$

where, m1 and m2 are the slope rates of the droop curve for negative and positive frequency deviations respectively, $P_{available}$ is the available adjustment power level in the farm, $deadband$ and $controlband$ are the characteristics of the droop curve defined in the grid codes and $\%reserve$ is the amount of reserve power required for the droop control.

The dead-band, control-band and percentage reserve characteristics of the droop curve can be configured as required.

6.4 Concluding remarks

In this chapter a wind farm control algorithm with different control objectives is developed. The wind farm control algorithm includes several control routines for power curtailment control, synthetic inertia response, and droop control. The control algorithm is implemented using a Matlab S-Function. The sampling time of the wind farm controller can be adjusted as required. The wind farm controller communicates with the individual wind turbines' controller through their PAC. The PAC works as an interface between the wind farm controller and the wind turbines' full envelope controller. Wind farm controller dispatches a wind farm reference power between the turbines corresponding to their operating condition and wind farm control mode (e.g. synthetic inertia response).

The wind turbines' PAC decides whether to deliver the requested power set-point adjustment from the wind farm controller depending on the wind turbines' operating condition. Power set-point adjustment requests from the wind farm controller could be rejected by the PAC if the supervisory limits are exceeded. Therefore, the wind farm controller distributes the wind farm reference power between the turbines corresponding to the operating status flags generated by each individual wind turbines’ PAC during the simulation. The wind farm controller can be modified to adopt different control strategies if required.

Chapter 7: Testing of the wind farm Simulation model

IN this chapter the wind farm model builder tool and the wind farm control algorithm that are developed in the previous chapters are tested. The wind farm model builder is used to develop a wind farm model with large number of turbines to investigate the performance of the developed wind farm model under different wind farm control objectives.

7.1 Wind farm model with large number of turbines

A wind farm model with 50 5MWwind turbines is developed. Wind turbines are placed in 5 rows parallel to the wind direction with 800m distance between them. The model is simulated for $4000s$ at $8ms^{-1}$ mean wind speed with 10% turbulence intensity. The wind farm controller is configured, to operate the wind farm in power curtailment control mode. The traffic-light rules are applied in this control mode and the wind farm power output is reduced by 5% of its total available power. Power curtailment control mode of the wind farm controller can be used to provide reserve power for primary frequency response.

Figure 7-1 to Figure 7-4 illustrate the operation of the 250MW wind farm in curtailed control mode. As can be seen wind turbines are operating below $6.5ms^{-1}$ wind speed for a significant amount of time. During these low wind speed periods the wind turbines' PAC starts a recovery process and the incremented Δp of the turbine in recovery mode is reduced to zero. Since the number of available turbines (with their PAC not in recovery mode) dictates the level of possible adjustment power at any time, in order to dispatch a sensible level of power set-point between the available turbines, the wind farm dispatch unit adjusts the total farm reference power to be less than or equal to the total available adjustment power level of the farm. The wind farm controller identifies the wind turbines operating at these low wind speed conditions and sets their power set-point adjustment level to zero allowing them to complete their recovery process.

This low speed recovery process is evident in the thrust coefficient of WT1 depicted in Figure 7-1. As can be seen during low wind speeds, the PAC starts the recovery process. Therefore, the incremented power set-point Δp value is reduced to zero. Thus the thrust coefficient also moves back to the value corresponding to the normal operation of the wind turbine before power set-point adjustment.

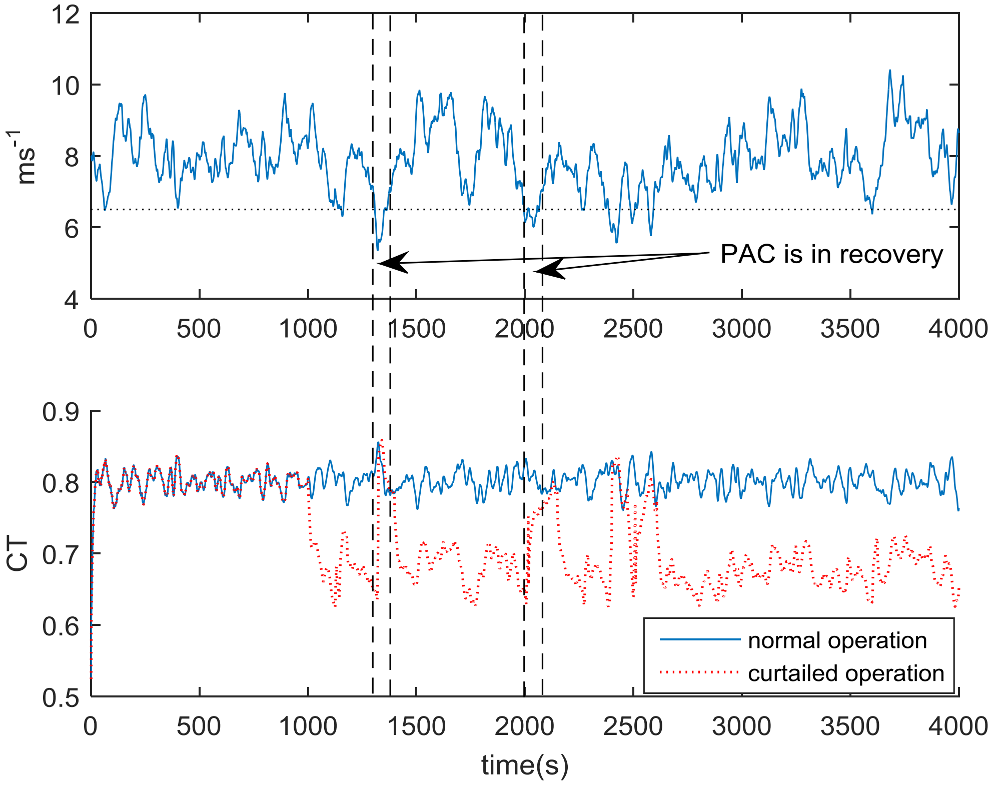


Figure 7-1: PAC recovery in the low wind speeds

Figure 7-2, illustrates the velocity deficits at each turbine. The initial transient in the velocity deficit takes place in the first 900sec of the simulation. At 1000sec the wind farm power output is curtailed by 5% of its available power. Since all turbines are operating in the green traffic light region, all turbines are participating in the wind farm curtailment control operation.

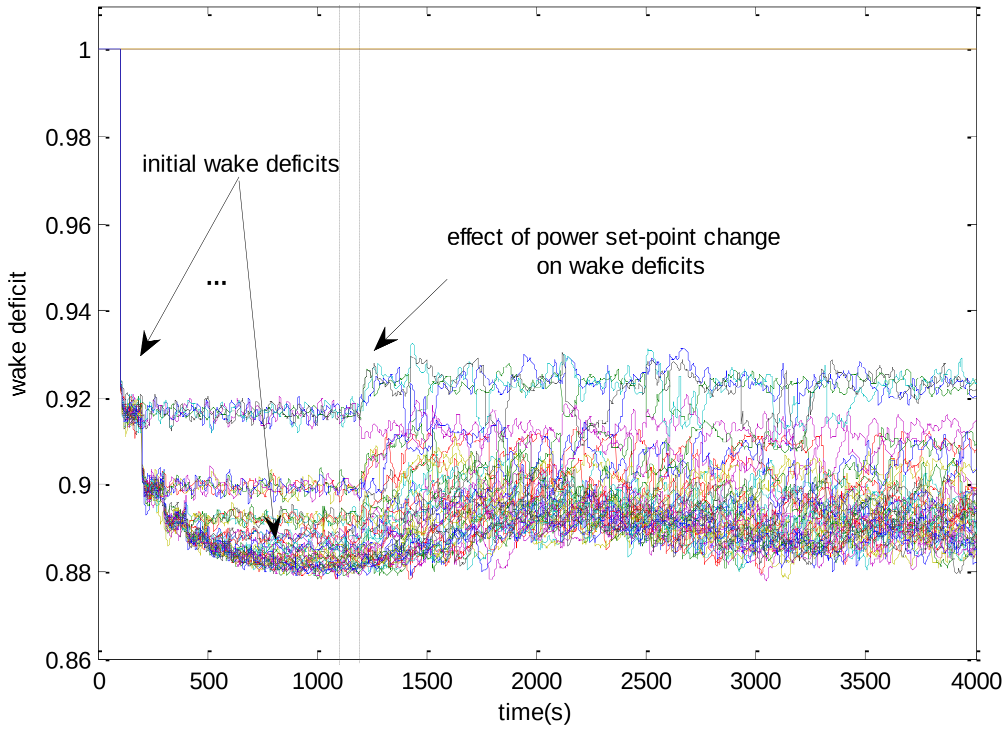


Figure 7-2: Wake deficits after power curtailment at 1100sec

Figure 7-3 shows the total power output of the wind farm and the difference between before and after curtailment. The difference between the power outputs clearly shows the level of power curtailment achieved in this control mode.

Initially, the power curtailment is achieved as requested. However, since the wind farm controller is an open-loop controller and due to the change in the wake deficits as a result of power set-point change for wind turbines, the deviation between the power output and the reference power is actually increased. The closed-loop controller design of the wind farm control system will be investigated in the future work. Figure 7-4 illustrates the power curtailment level achieved immediately after the power adjustment request at 1000sec. It shows that the power output of the farm is adjusted in a quick and safe operating manner which satisfies one of the main requirements defined for the wind farm modelling in this chapter.

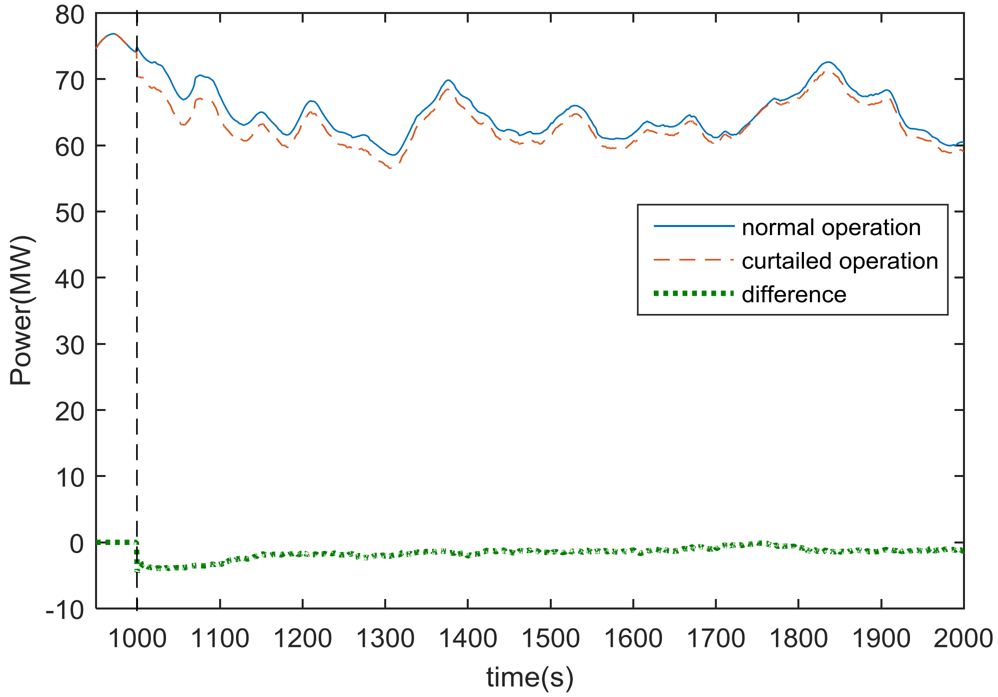


Figure 7-3: Total power output before and after curtailment of 250MW wind farm in below rated $8ms^{-1}$ wind speed

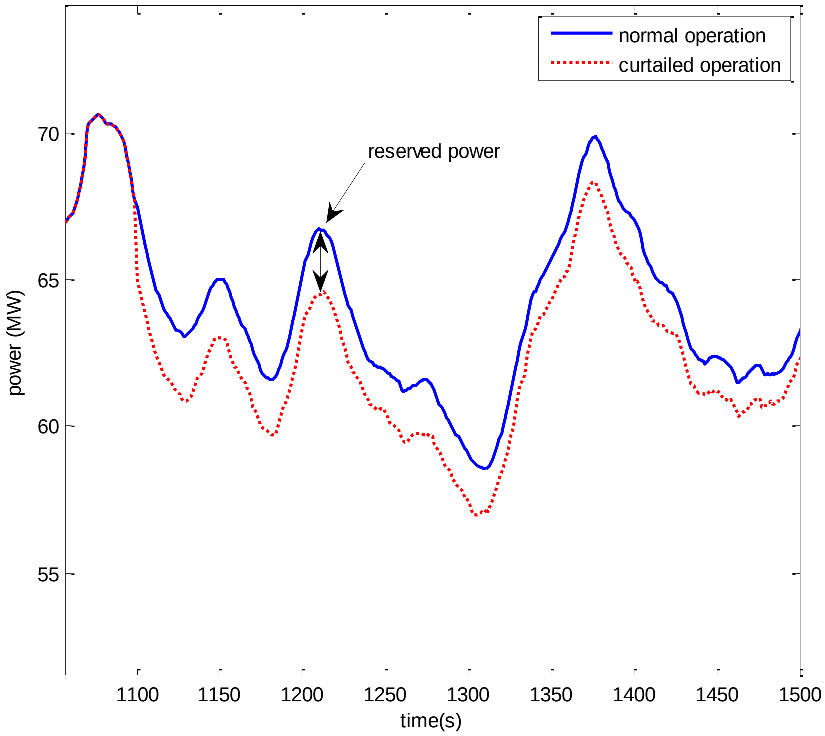


Figure 7-4: Total power output at the power curtailment control mode

7.2 Power de-loading effect on wind turbines in wake

The power output control of most wind farm control algorithms in the literature is only possible in above rated operation, or where the available power is higher than the farm reference power. Therefore, wind farm optimisation and power output control mode of these models is limited to certain operating conditions. The wind farm model is capable of dynamically altering the power output of the wind turbines in all operating region using the PAC of wind turbines. Therefore, wind farm power/load optimisation control can be investigated in any operating conditions.

In order to investigate the effect of power de-loading operation of a wind turbine on the performance of the other wind turbines in the wind farm and on the total power output, a number of simulations are performed for a cluster of 10 wind turbines placed behind each other parallel to the wind direction with $800m$ distance between them as shown in Figure 7-5. A set of simulations are conducted at $8ms^{-1}$ mean wind speed with 1% turbulence intensity. The wind farm control objective is defined to reduce the power set-point of the most upstream turbine in the line and investigate the effect of this power adjustment on the wake effects and other turbines performances.

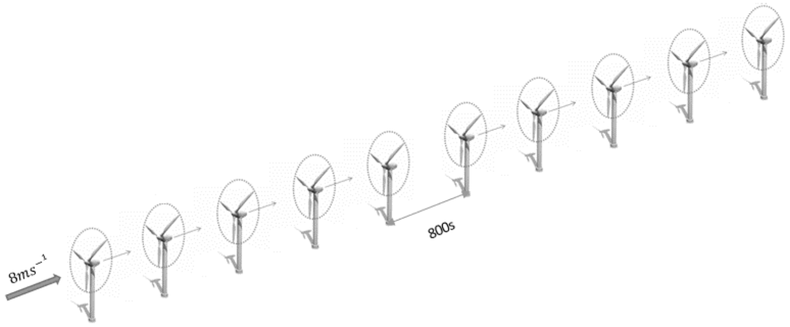


Figure 7-5: Wind farm layout, de-loading operation of WT1

Figure 7-6 shows the wake deficits experienced by each turbine after the power adjustment of $-0.25MW$ being realised at 1100 seconds at WT1. As can be seen, the wake deficits are reduced as a result of this power adjustment at each turbine in the wake consecutively after a time delay. Since wind turbines operate above the minimum wind speed, when the PAC starts the recovery process, and the level of power set-point adjustment is chosen to be less than the available adjustment power level at WT1, it can be seen that the power curtailment of WT1 is achieved for a long time period. Therefore, the effect of this power de-loading operation can be investigated on the other turbines performances.

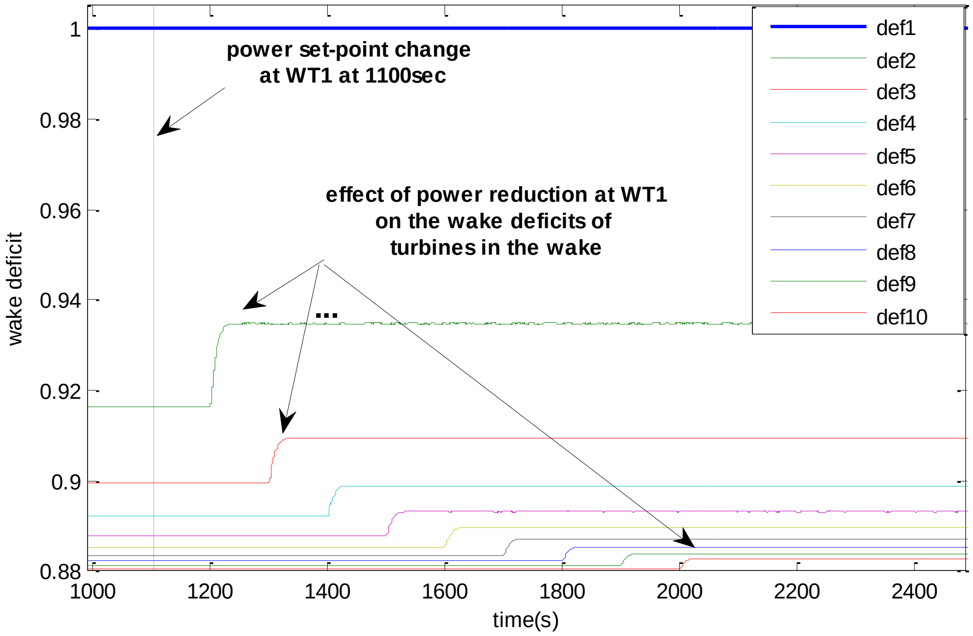


Figure 7-6: Wake deficits at WT1 to WT10

Figure 7-7, depicts the thrust coefficient of the wind turbines after power adjustment at WT1. The power de-loading operation of WT1 has negligible effect on the thrust coefficient of the other turbines in the wake.

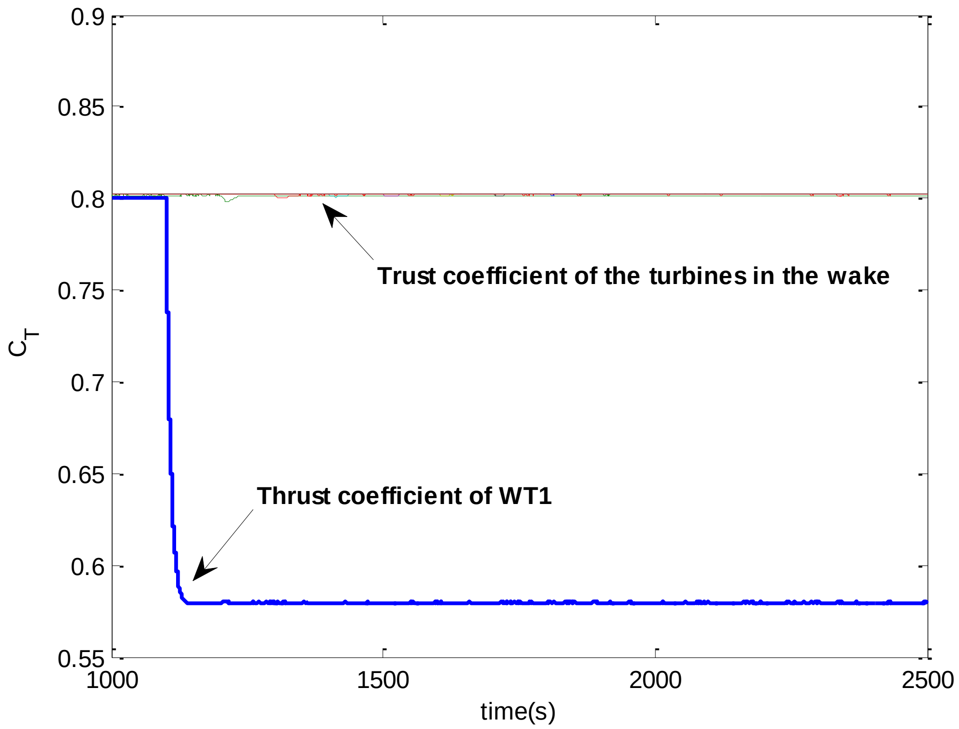


Figure 7-7: CT of WT1-10

The effect of this power adjustment on the wind turbines in the wake is also evident in the individual power outputs, wind speed, and total power output of wind turbines in Figure 7-8, Figure 7-9, and Figure 7-10. As can be seen, power output of WT1 is reduced at 1100 seconds and the effect of this power adjustment is experienced by the other wind turbines in the wake after consecutive time delays. Figure 7-9 shows the change in the wind speeds and wake condition at each turbine in the wake due to this power set-point change at WT1. Figure 7-10, depicts the effect of this power set-point change on the total wind farm power output. Since the wake deficits at the turbines in the wake are reduced, their experienced wind speeds and power outputs are increased. However, by comparing the total power output before and after de-loading operation of WT1, it can be seen that the power output is reduced by almost 0.2% in comparison with the normal operation. However, further adjustment of the WT1 power set-point reveals that, the total power output of the farm can be increased by de-loading operation of the upstream wind turbines.

In order to achieve power maximisation by de-loading operation of WT1, the power set-point of WT1 is adjusted by -0.1MW. Figure 7-11 shows the effect of this power adjustment on the total power output of the wind farm. As can be seen by reducing the power of WT1 by 2% from its rated, the total power can be increased by 0.16%. It is shown that by coordinating the operation of wind turbines the total power output of the wind farm can be increased. The power maximisation objective of the wind farm controller is further investigated in Chapter 7.

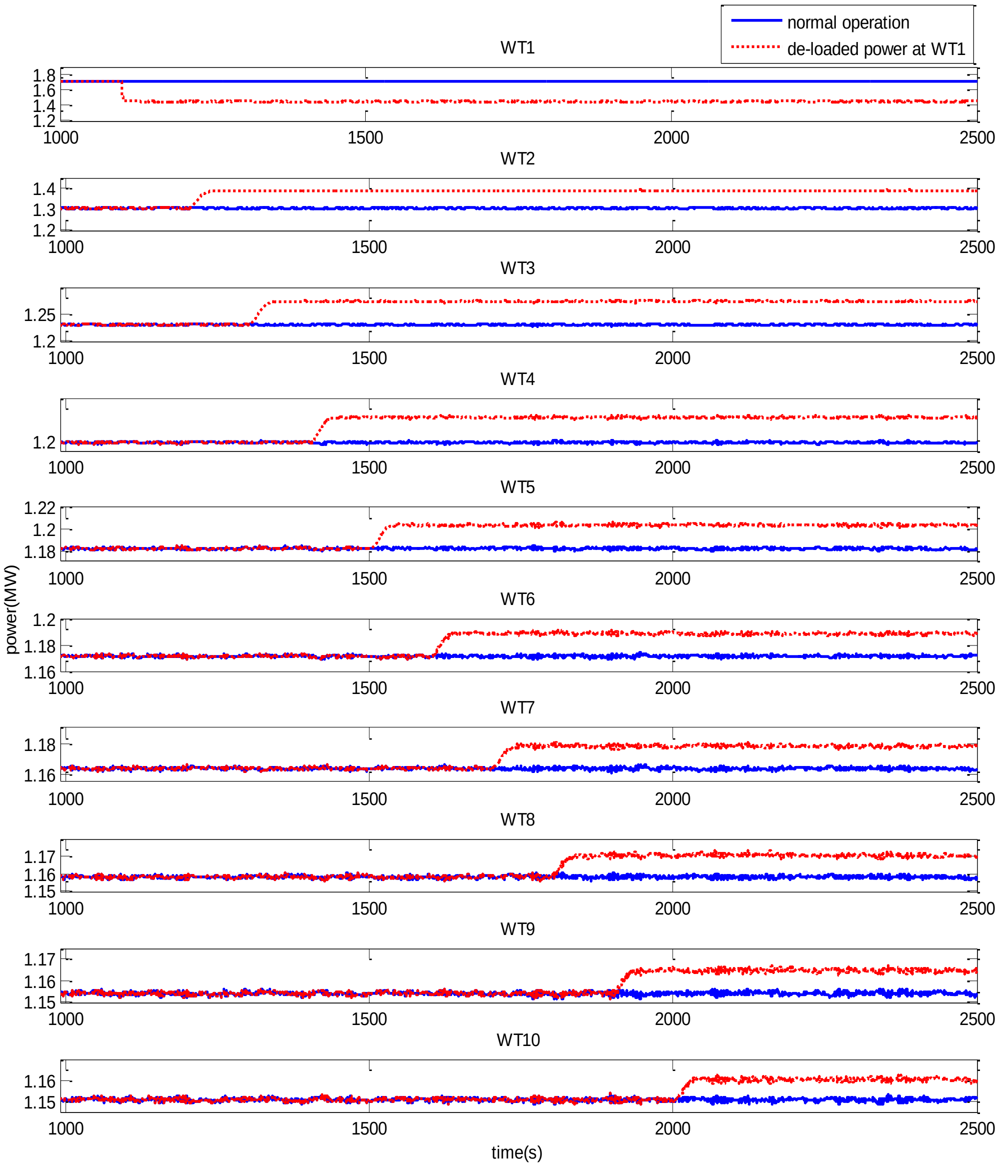


Figure 7-8: Power output of WT1-10 before and after power reduction at WT1

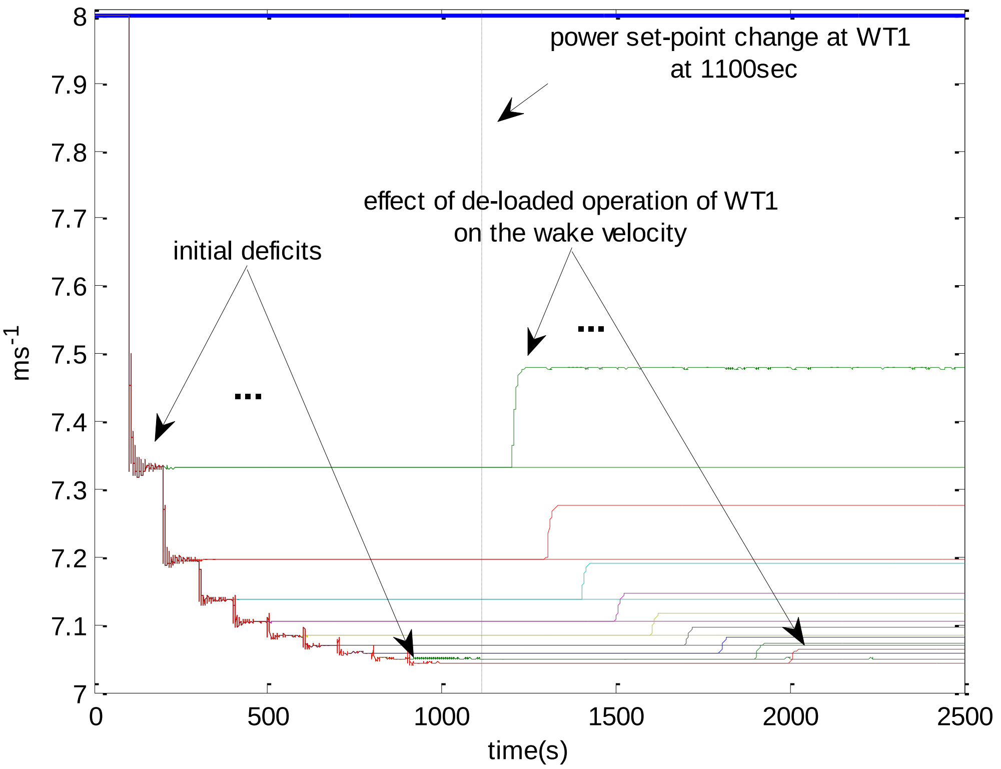


Figure 7-9: Wind speeds at WT1-10 before and after power set-point change at WT1

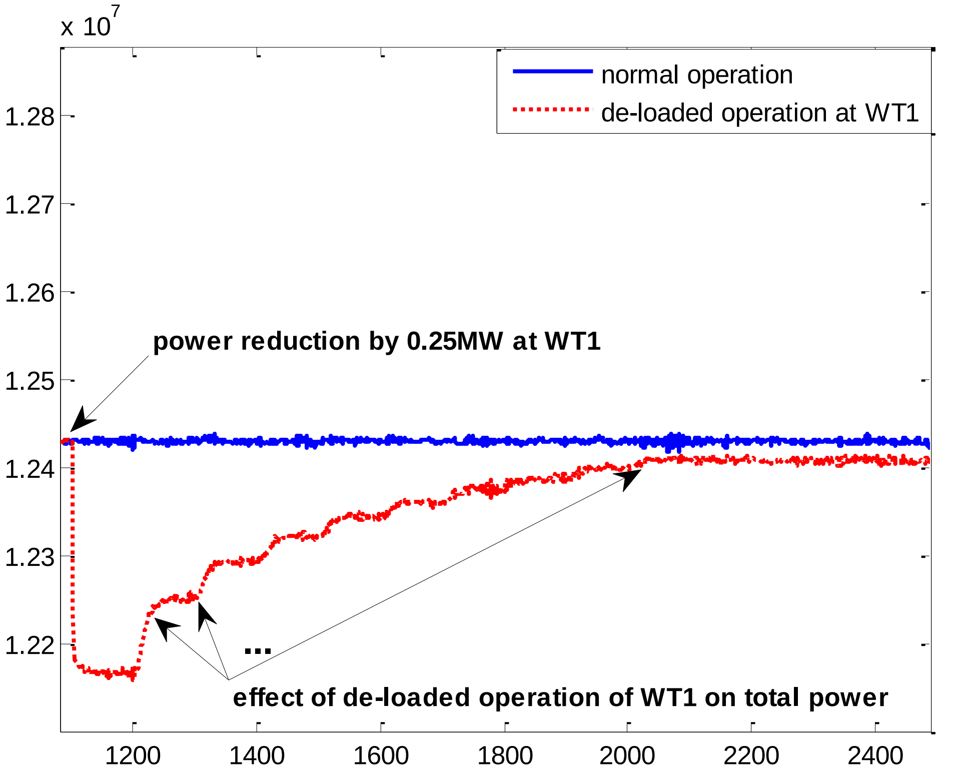


Figure 7-10: Total power output before and after power set-point change at WT1

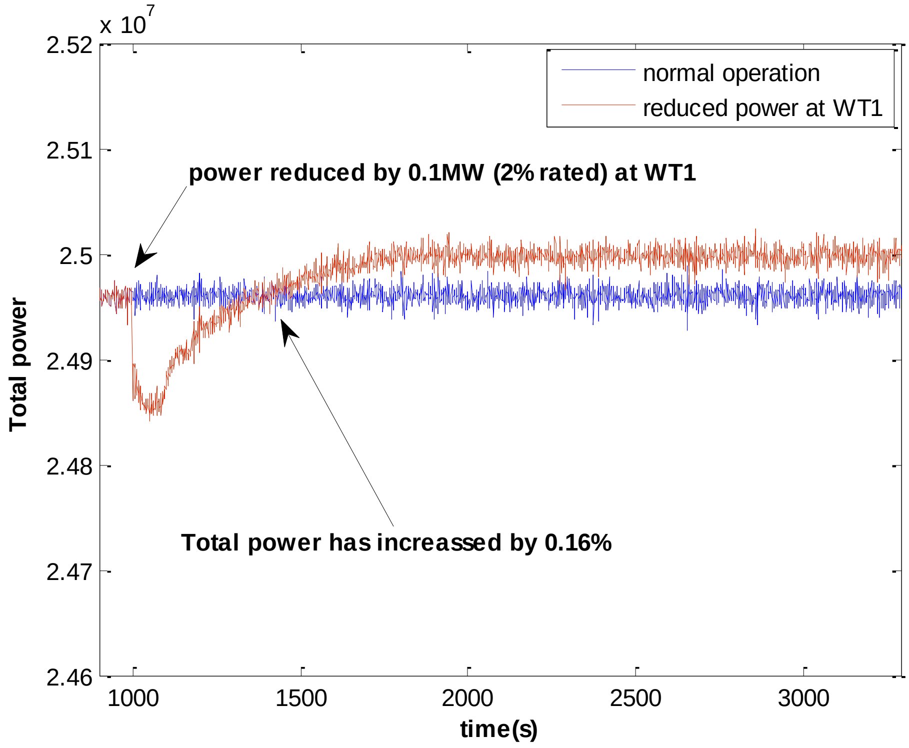


Figure 7-11: Effect of power de-loading operation on wind farm power maximisation

Another simulation is conducted at $8ms^{-1}$ mean wind speed with 10% turbulence intensity with the same control objective for the same wind farm layout to measure the level of power performance improvement in more realistic wind condition. The wake deficits experienced by the wind turbines are depicted in Figure 7-12. Similarly, the effect of power set-point adjustment at WT1 on the other turbines is evident in a consecutive manner. The power outputs of the first three turbines in the line are depicted in Figure 7-13. Clearly, the power performances of the individual wind turbines in the wake are improved in this scenario. The wake propagation through the wind farm and its effect on the wind farm power production is evident from the results. As it is shown, by adjusting the power output of the turbines and coordinating the operation of the turbines, it is possible to alleviate the wake losses in the wind farm.

Figure 7-14 and Figure 7-15 illustrate the total power output and the mean power output of each turbine before and after this power adjustment. As can be seen, the mean of total power output has increased by almost 0.2% for a 2% power reduction of WT1 from its rated in comparison with the optimal operation of all wind turbines. The average power of each turbine is calculated after the power de-loading at WT1 from 1100 seconds to

4000 seconds. However, for longer simulation in this control mode the mean total power output may increase further as the initial change in the wake condition lasts for 900 seconds after power set-point change at WT1. Therefore, it is shown that the power maximisation through de-loading operation of WT1 in this scenario can also be achieved for turbulent wind condition.

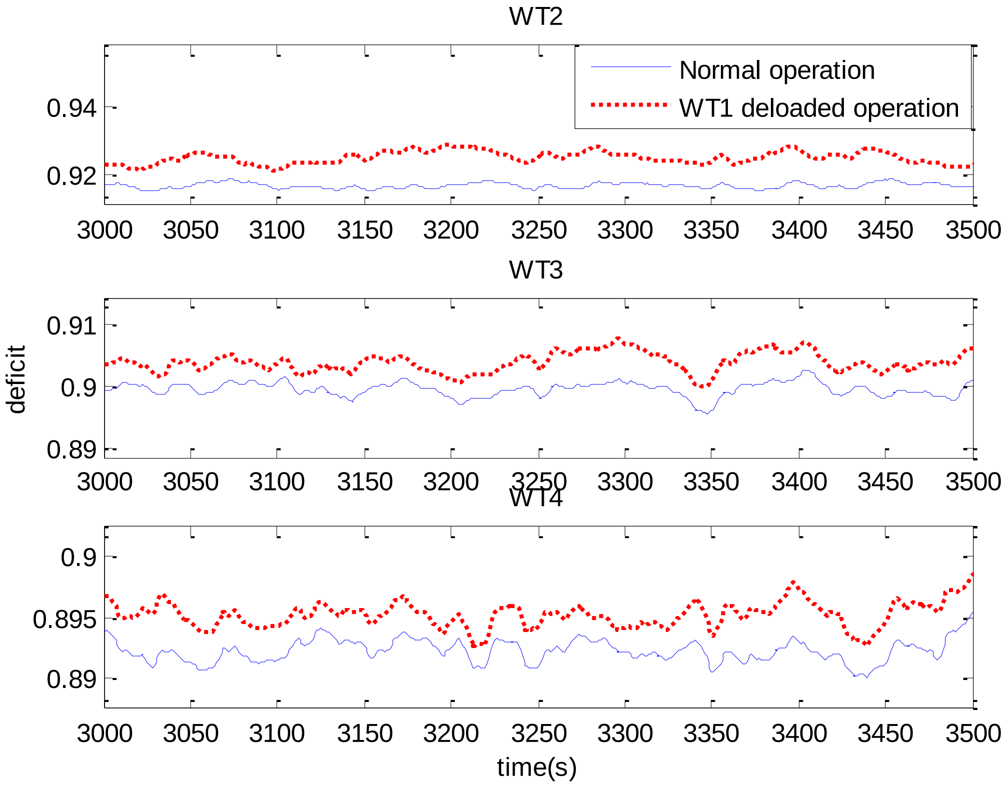


Figure 7-12: Wake deficit at turbulent wind for WT2 to WT4 before and after power reduction at WT1

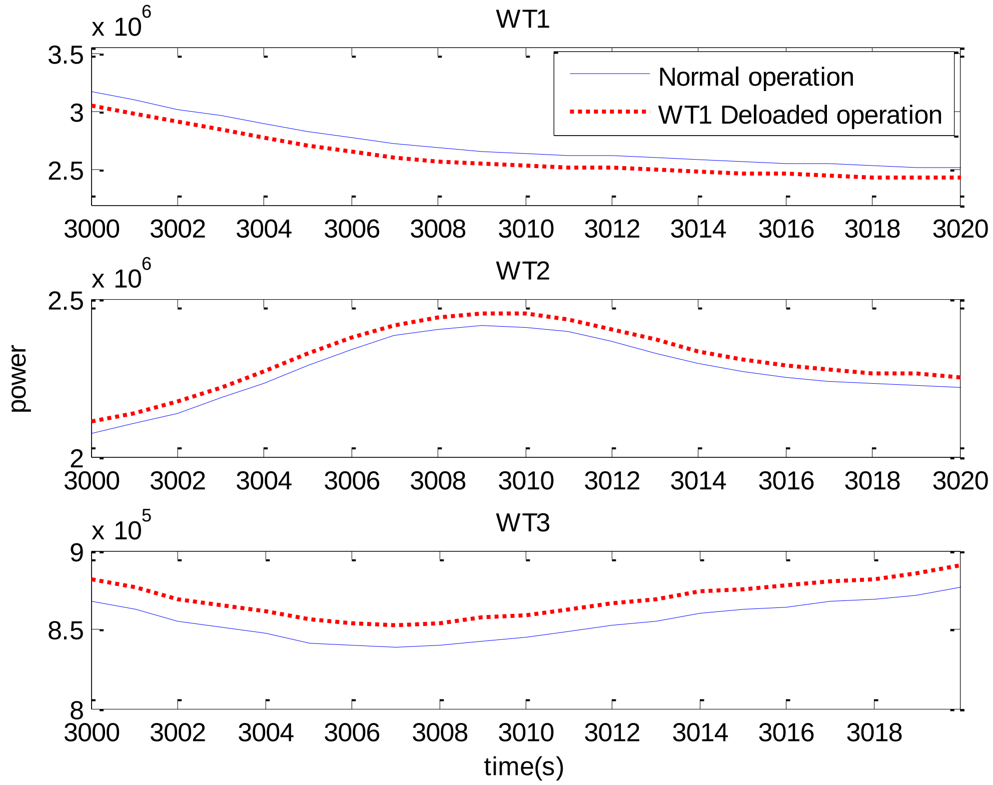


Figure 7-13: Power output at WT1 to WT3 before and after power adjustment at WT1

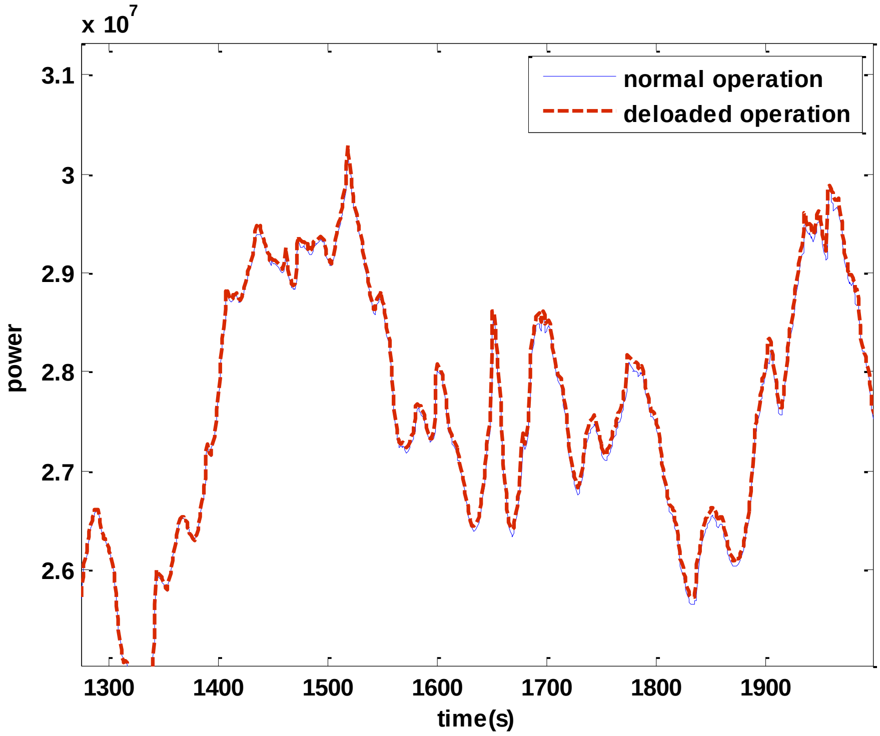


Figure 7-14: Total power output before and after power adjustment at WT1

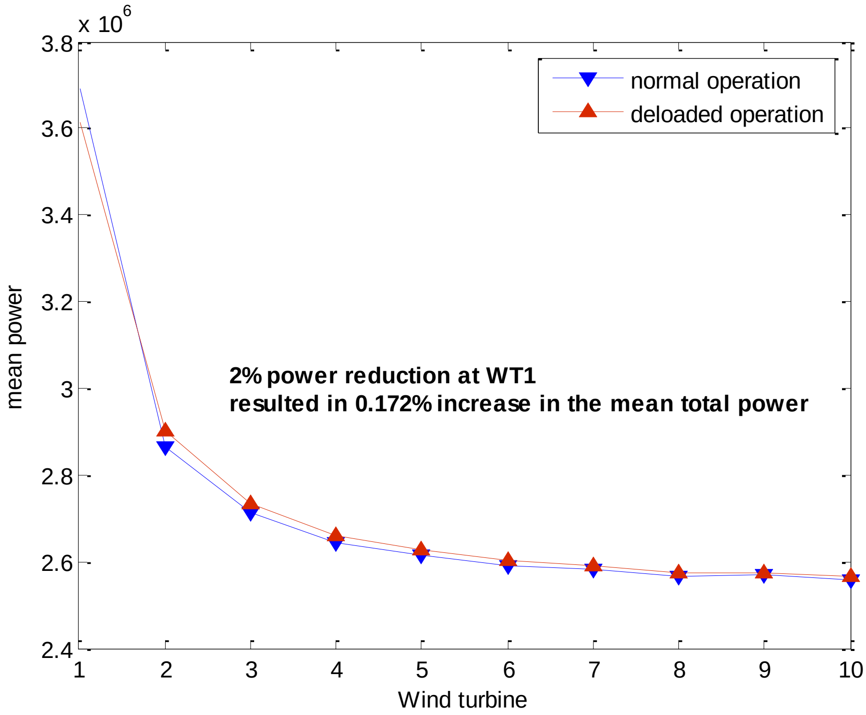


Figure 7-15: Mean power output of WT1 to WT10 before and after power adjustment at WT1

7.3 Wind farm positive power demand

Extra power may be requested from a wind power plant during normal operation mainly for primary frequency response. Wind farm power output must be increased in a short period of time, to compensate for grid under frequency events. In this section the ability of the wind farm model and PAC to provide a positive power demand is investigated. Similar to the power curtailment operation, the wind farm controller allocates positive power set-points for each turbine according to their operating condition. The interface between the wind farm and wind turbines' controllers is through the PAC. In below rated, where wind power is limited, the positive extra power delivery by the wind farm is limited to the maximum power adjustment level that the wind turbines can produce. In the case of synthetic inertia response, PAC adjusts the power output of the turbines in order to produce the maximum power adjustment level, regardless of their traffic light limits, until the black limits are reached. Therefore, the available turbines can provide as much extra power as possible until they hit the black limits.

The below rated positive power adjustment of the wind farm is investigated utilising the same wind farm layout depicted in Figure 7-5 at $8ms^{-1}$ constant wind speed. At 1100sec the wind farm reference power is adjusted to provide an extra amount of 2.5MW power during 200 seconds. The aim is to investigate the ability of the wind farm to provide this extra power demand in below rated operation and measure the time period over which the extra power is delivered. Figure 7-16 shows the total power output of the farm before and after this power reference request at 1100 seconds. As can be seen the power output of the farm is increased immediately after 1100 seconds to a level equivalent to the power reference request. The power set-point increase causes the wind turbines' operating point to eventually hit the black supervisory boundary and consequently go to recovery stage. In this case, the wind farm controller reduces the total power adjustment level of the farm to match the total available power adjustment in the farm. This power reference reduction is evident in Figure 7-16.

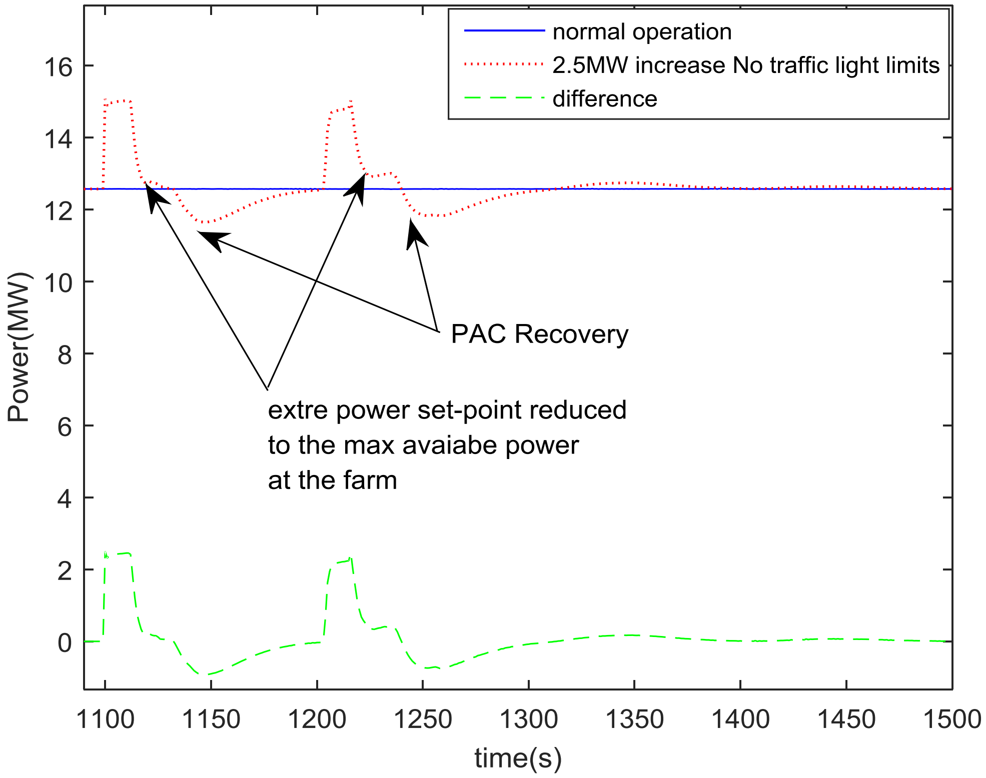


Figure 7-16: Below rated wind farm power increase

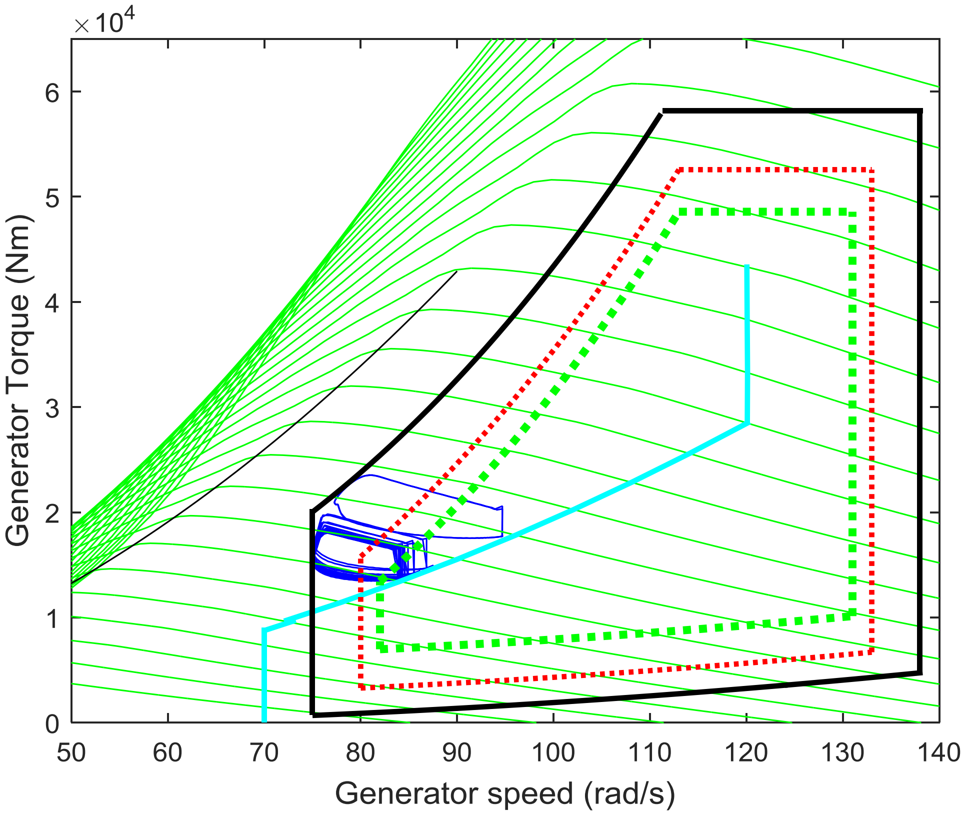


Figure 7-17: Wind turbines operation in the generator speed- generator torque plane for 2.5MW power set point increase with no traffic light rules applied

The operating condition of the wind turbines in the generator speed-generator torque graph is depicted in Figure 7-17. As can be seen, the power set-point of wind turbines are increased until they hit the black boundary. At this point wind turbines' PAC recovery is started. Since the traffic light rules are not applied, the wind turbines can deliver the requested extra power regardless of their operating condition in the traffic light regions. The wind farm power set-point is adjusted when the total power available in the farm is less than the requested power.

Figure 7-18 illustrates the traffic light flags' status during this control mode. Wind turbines' operating points cross the traffic light limits towards the black boundary. The synthetic inertia response is only required from the wind turbines for a limited period of time. As can be seen from the power output graph, in this control mode, an extra power delivery of 5% of the wind turbines rated power is achieved within about 15 seconds before the wind turbines hit the black boundary.

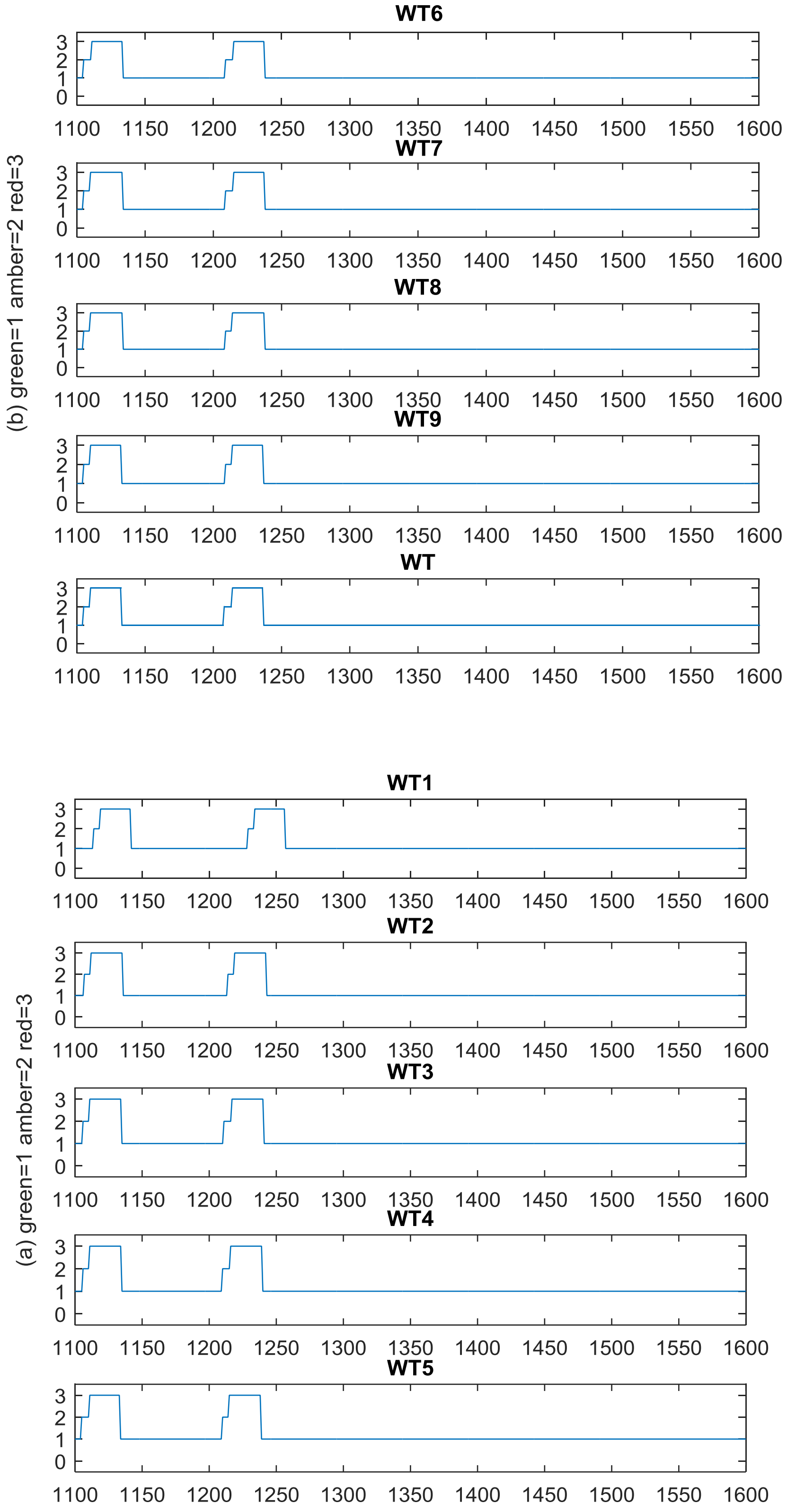


Figure 7-18: Traffic lights flags for the win farm simulation with 2.5MW power increase

Another simulation is conducted using the same wind farm model with similar conditions to investigate the wind farm positive power delivery under the traffic light rules. Figure 7-19 shows the total power output of the turbines in this control mode.

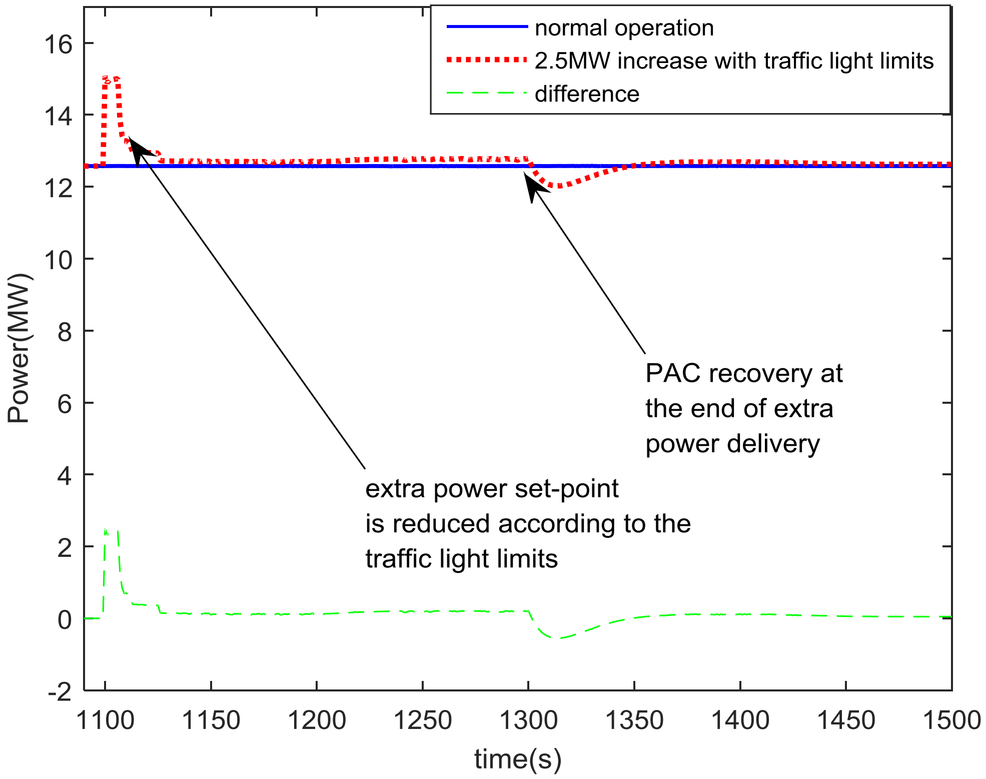


Figure 7-19: 2.5MW power increase traffic light flags

As can be seen the extra power delivery can be achieved for longer period of time by preventing the wind turbines operating point from moving towards the black boundary using the traffic light limits. In this control mode the wind turbines' power set-points are adjusted according to their operating region. Since the power set-point for the wind turbines operating in the red region is zero, according to the traffic light rules, the extra power delivery by the wind turbines crossing into the red region is reduced to zero, allowing them to move back towards the green zone and normal operating condition.

The wind turbines operation on the generator speed-generator torque graph in Figure 7-20 shows the operation of the wind turbines under the traffic light rules. Initially, wind turbines provide the maximum power required to contribute for extra power delivery. Once the wind turbines cross the traffic light boundaries, the total wind farm reference power is adjusted to achieve the extra power delivery corresponding to the maximum available power in the farm. Thereby, the wind turbines' operating points are kept within the traffic light limits and prevented from moving towards the black boundary.

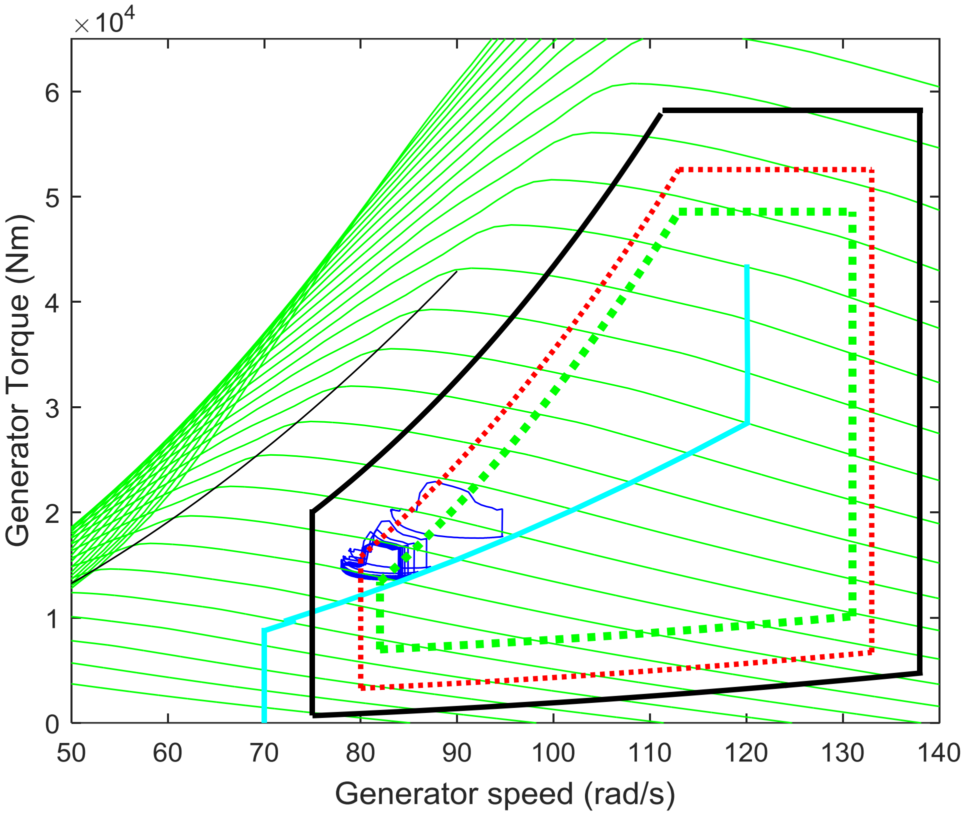


Figure 7-20: Wind turbines operation in the generator speed-generator torque plane for 2.5MW wind farm power increase with traffic light limits applied

Figure 7-21 depicts flags corresponding to the wind turbines' operation in different traffic light zones. The power set-points of individual turbines are also depicted in the graph. The power set –point scale in the graph is adjusted for clarity (i.e. 1=0.1MW). As can be seen, during the extra power delivery, wind turbines with higher available power contribute more to the power delivery and compensate for the turbines entering the amber or red regions. The power set-point compensation dispatch is evident in the WT1, WT2, WT3, and WT4 plots. The effect of traffic light limits on the power set-point dispatch for each wind turbine can also be seen from the graphs. Wind farm control with the objective of primary frequency response is further investigated in Chapter 8.

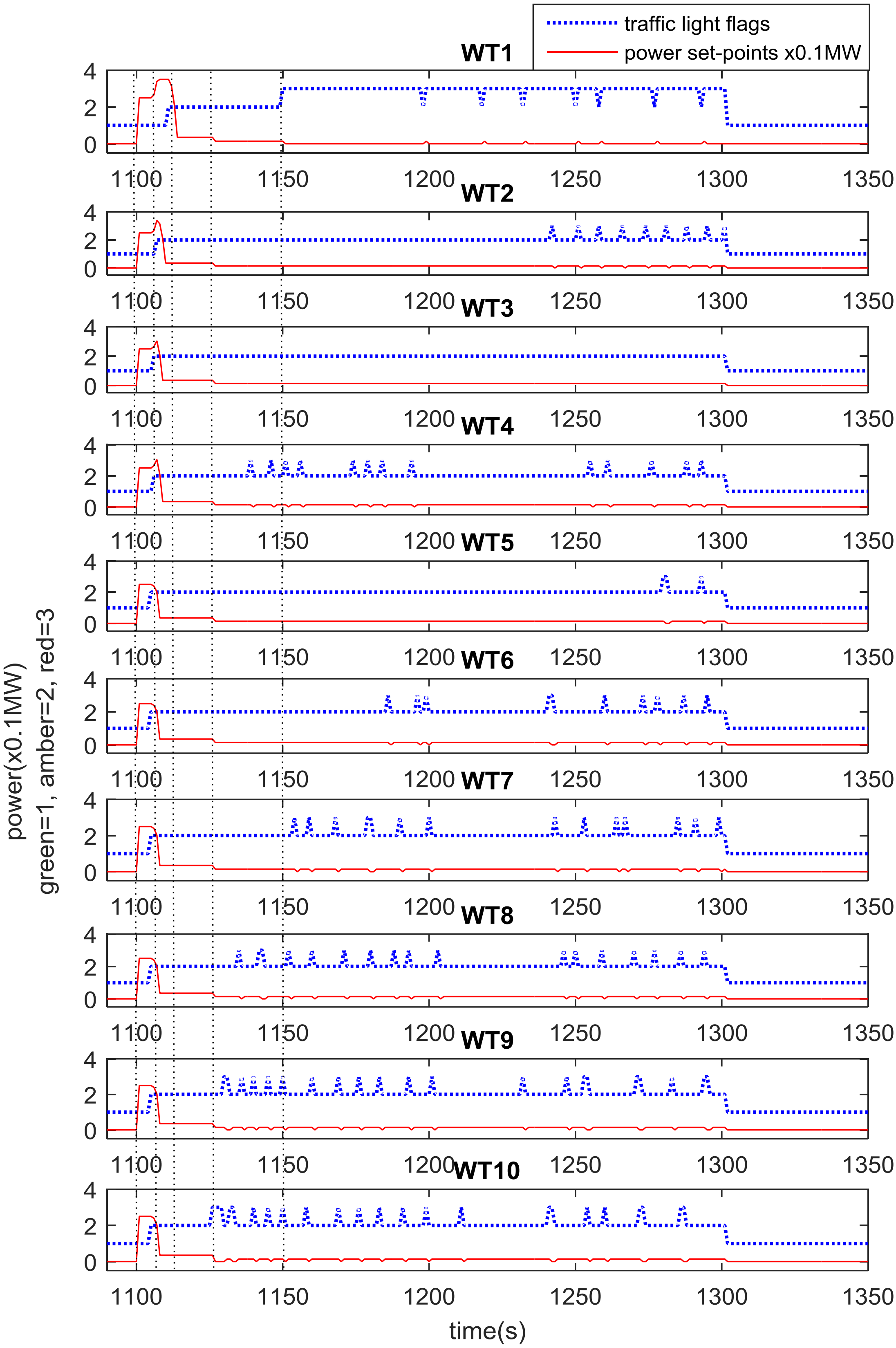


Figure 7-21: Traffic light Flags for 0.25MW power increase

Positive power adjustment is more achievable when wind turbines are operating in higher wind speeds. Therefore, another simulation is conducted for the same wind farm layout at $14ms^{-1}$ mean wind speed with 10% turbulent intensity. Figure 7-22 illustrates the total power output of the turbines before and after 2.5MW power set-point increase. It can be seen that the power set-point increase is achieved for longer period of times in wind speeds above rated wind speed.

Figure 7-23 depicts the operation of the wind turbines in the generator speed-generator torque plane. As can be seen the wind turbines are kept within a safe operating boundary.

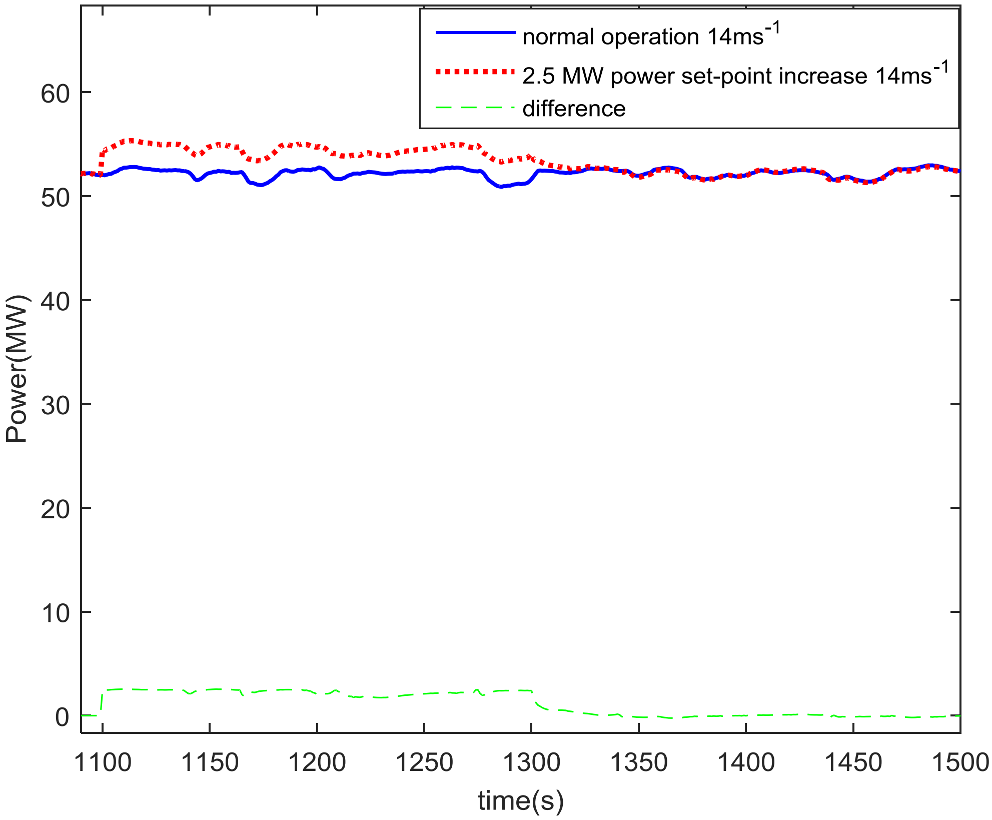


Figure 7-22: Total power output of the wind farm for 2.5MW power increase at 14m/s

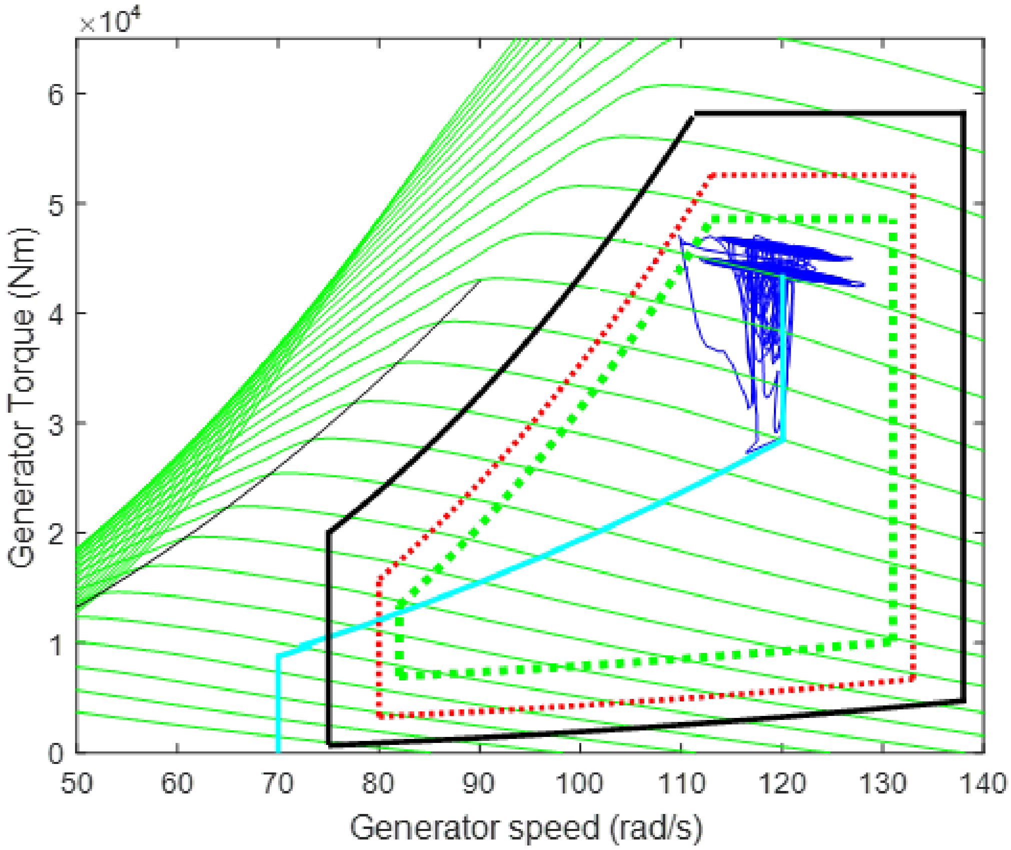


Figure 7-23: Wind turbines operation in generator speed-generator torque plane for 2.5MW power increase at 14m/s

Figure 7-24 shows the traffic light flags of the wind turbines in this control mode. The dispatched power set-point of the wind turbines are also depicted on the graph. The scale of the power set-points in the graph is zoomed for clarity (i.e. 1=0.1MW). As can be seen WT3 power set-point is reduced to zero as its operating point crossed into the red traffic light region. This can also be seen on the generator speed-generator torque plane. The reason for this power set-point reduction is the reduction in the wind speed at WT3. It can be seen that the other turbines are compensating for WT3 power set-point reduction during this control mode.

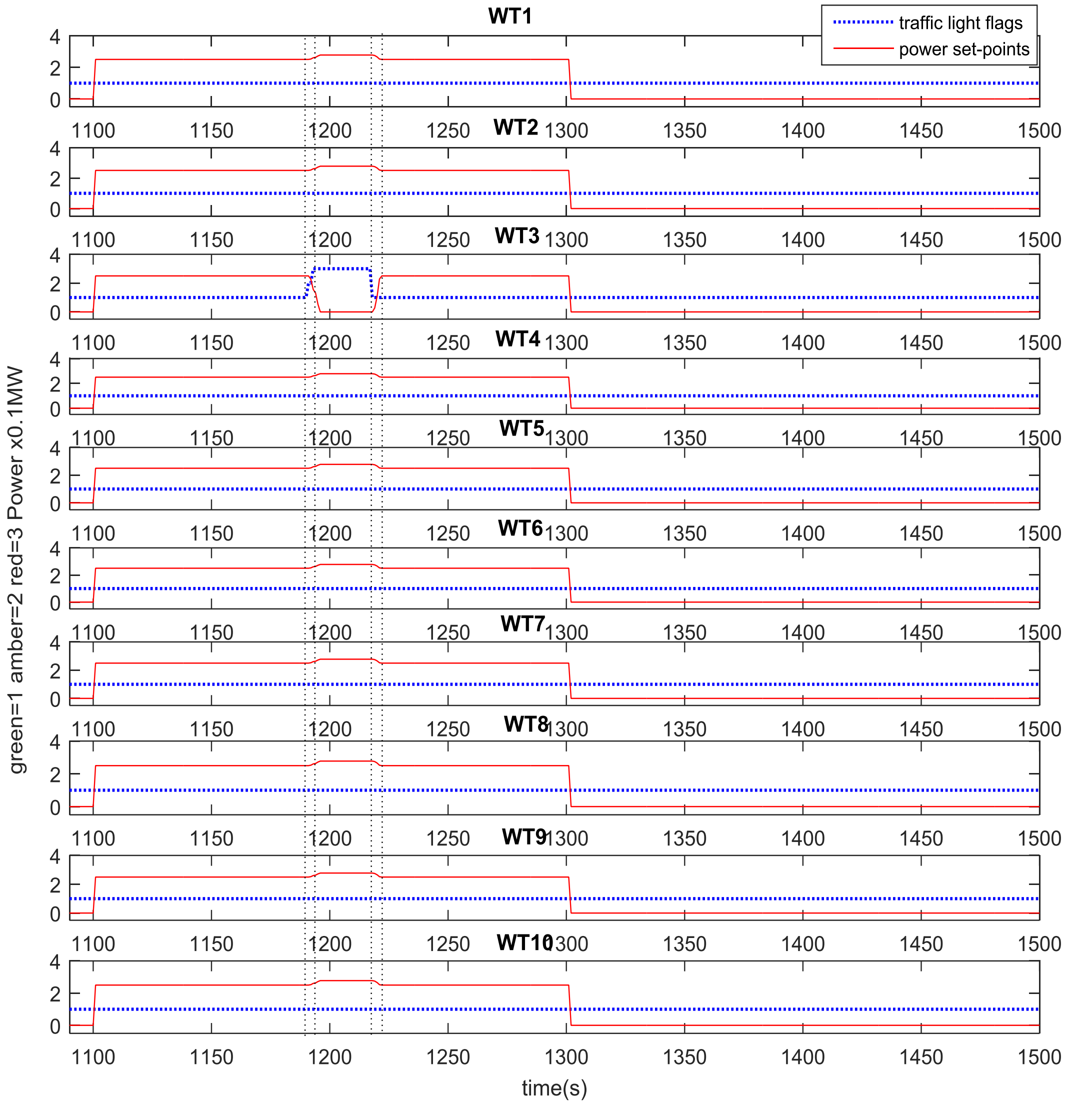


Figure 7-24: Power set-points and traffic light flags for 2.5MW power increase at 14ms^{-1}

7.4 Closely spaced wind turbines

Since a wind turbine in a wind farm can be affected by multiple partial wakes from other turbines, a simulation is performed to investigate the effect of partial wakes on the operation of the turbines in the wakes. The wind farm model layout is shown in Figure 7-25. Wind turbines are placed in two lines parallel to the wind direction with 800m separation between the turbines in the same line and 400m separation between the two lines. Simulation is conducted at 10ms^{-1} mean wind speed with 1% turbulent intensity to show the wake effects clearly. The spacing between turbines is chosen to illustrate the partial wake from neighbouring turbines in opposite line on the turbines in the other line.

In this case WT7 experiences the wake from WT6 and also from WT1 in the neighbouring row. Similarly, WT8, WT9, and WT10 also experience wake effects from the neighbouring turbines WT1, WT2, WT3, and WT4 in the same manner.

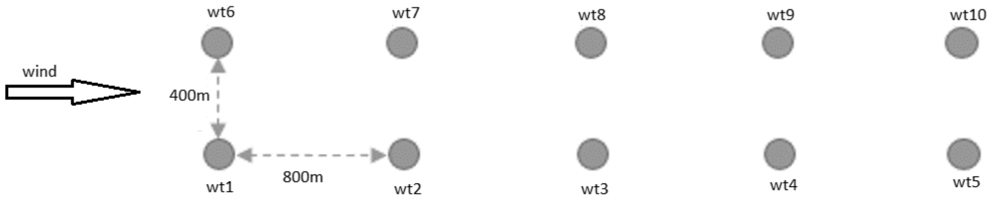


Figure 7-25: Wind farm layout

Figure 7-26 and Figure 7-27 illustrate the combined effect of partial wakes from neighbouring turbines on the turbines in their wake. At 1100 seconds the power set-point of WT1 is reduced by 0.1MW. The effect of this power set-point adjustment is evident in the velocity deficit of turbines in its wake. As can be seen WT2 experiences the effect of this power set-point adjustment after a time delay of 80 seconds at 1180 seconds and WT7 experiences the change after 90seconds at 1190 seconds. It is evident that WT7 is experiencing a partial wake since the wake deficit change in WT7 is lower than WT2. Similarly, other turbines in the wake are also experiencing this power set-point change in WT1 after appropriate time delays.

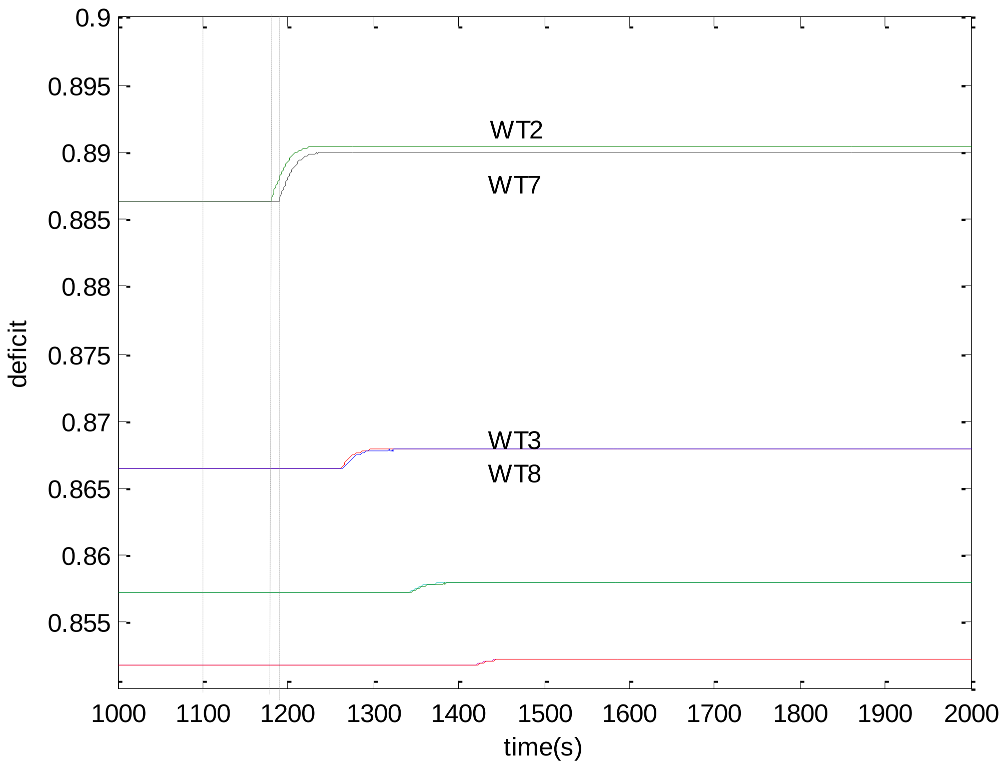


Figure 7-26: Wake deficit at different neighbouring turbines

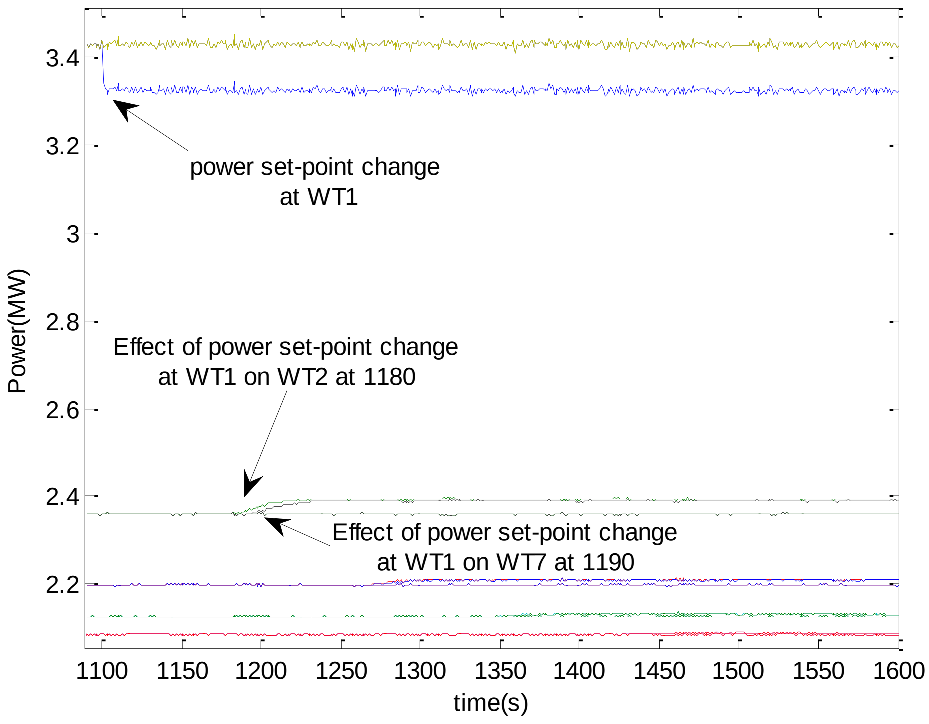


Figure 7-27: Wake effect on the neighbouring turbines

Figure 7-28 illustrates the effect of this power set-point change in the total power output of the farm. As can be seen by adjusting the wind turbines' power set-point, the power losses in the wind farm with closely positioned wind turbines can be alleviated.

The developed wind farm model also can be used to compare different wind farm layouts.

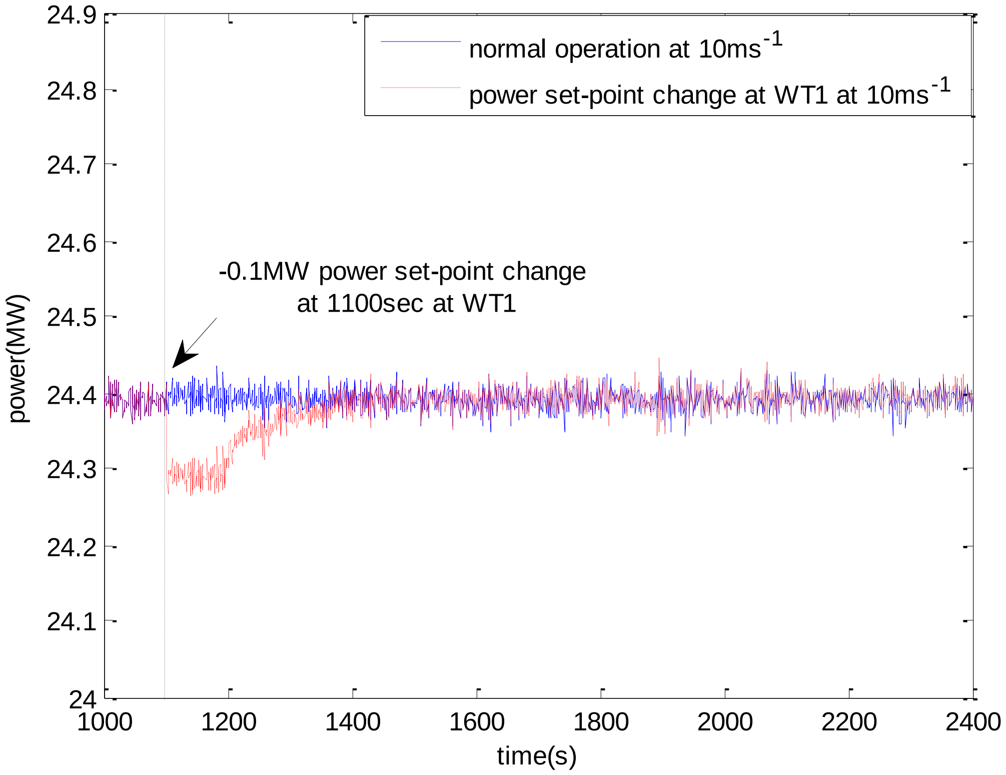


Figure 7-28: Total power output before and after power set-point change at WT1

7.5 Concluding remarks

In this chapter, the wind farm model and wind farm controller that are developed in the previous chapters are tested. Various control scenarios including power curtailment, power set-point increase, and power set-point coordination of the wind turbines corresponding to their operating conditions are investigated. It is shown that the wind farm controller structure is suitable for designing different control objectives. Hence, the wind farm model and its controllers are suitable for fast simulation and controller design.

Chapter 8: Providing primary frequency response by wind farms

INCREASING levels of wind power installed capacity in the grid system can potentially reduce the total system inertia and consequently increase the risk of system instability. Since variable speed wind turbines are decoupled from the AC network by their power converters, they cannot inherently contribute to the system inertia response in the same way as conventional power plants. Therefore, maintaining the system stability can become a challenging task without compensating for the reduction in the system inertia level. Consequently, wind power plants are being required to participate in the primary frequency response by the transmission network operators.

Numerous reports in the literature deal with the primary frequency response and provision of ancillary services at the turbine level [102], [98]. Many of these implementations, approach this control problem through modifying the structure of the wind turbines’ full envelope controller and its control strategy. The majority of these algorithms are only applicable in above rated operation conditions. However, wind power plants must be able to provide inertia response for grid primary frequency support, in any operating conditions, when an under frequency event is detected. At the farm level, uncoordinated response to a frequency event by each individual wind turbine, makes the total level of inertia response provided by the wind power plant uncertain. Moreover, modifying the structure of existing full envelope controllers in the commercial wind turbines cannot be achieved easily since they are protected by the manufacturers and insurance companies. Therefore, in this chapter the primary frequency response of the wind farms is investigated through coordinating the power set-point of individual wind turbines, by utilising an interface between the wind farm controller and the full envelope controller of each individual turbine. The proposed interface, PAC, acts as a feed-forward modification to the existing standard HAWTs' full envelope controller. Thus, the operation of the wind turbines' full envelope controller is not affected by the operation of the PAC. The wind farm controller then coordinates the operation of the wind turbines, to achieve a collective power output of in the wind farm as a whole.

Ideally, wind power plants are desired to operate similarly to conventional power plants. Conventional synchronous generators are directly connected to the grid via a power transformer. Therefore, their electromagnetic torque is coupled to the grid frequency. Hence, any frequency deviations in the grid can be detected at their terminals and compensated automatically by their governor and automatic generation control systems.

In contrast, variable speed pitch regulated wind turbines are normally decoupled from the grid system through power electronics. Thus, the rotational speed of the wind turbine rotor is decoupled from the grid frequency. As a result, the kinetic energy available in the turbine rotor cannot be used directly to compensate for the system frequency fluctuations.

The large rotating rotor of multi megawatt wind turbines can be seen as a source of "hidden inertia". The available kinetic energy stored in the rotor can be released to increase the power output of the turbine temporary for the grid primary frequency response.

The level of kinetic energy stored in the wind turbines rotor is related to the rotor inertia and the rotating speed by

$$E = \frac{1}{2} J \omega^2 \tag{8-1}$$

where, J is the inertia of the rotating mass and ω is the rotational speed of the rotating mass.

Therefore, by adopting an appropriate control regime, the stored kinetic energy can be extracted and used for the grid frequency support.

Normally, the primary frequency response for wind power plants is implemented as a combination of synthetic inertia response, responding to a rapid rate of change of frequency deviation, see Figure 8-1, and droop control, responding to slow rates of change. It is anticipated that wind turbines would mainly support the grid with primary frequency response, whilst the secondary response which is for eliminating the steady state frequency error, is covered by the conventional power plants [104].

The principle behind the wind turbine inertia response is to adjust the generator torque demand, corresponding to the rate of change of frequency fluctuations, by releasing the available kinetic energy stored in the wind turbine rotor. In this control regime, the wind turbine's operating point moves towards the stall region. Since operating in the stall region is not recommended for the variable speed pitch regulated HAWTs, a suitable control strategy is required to prevent the wind turbine operating outwith a predefined safe operating boundary, whilst providing the synthetic inertia response.

Figure 8-1 shows a typical recommended frequency response given by UK National grid [105]. The frequency service is divided into continuous and occasional services with their corresponding time frames. The occasional service consists of a primary response with corresponding time frame of 10 seconds to 30 seconds a secondary response with corresponding response time between 30 seconds to 30 minutes for abnormal frequency event occurrence. The maximum allowed frequency deviation is 0.8Hz in an event of large generating unit loss, and must be maintained within $\pm 0.5 Hz$ of the nominal frequency. The full delivery of the inertia response must occur within 200ms after the event [106].

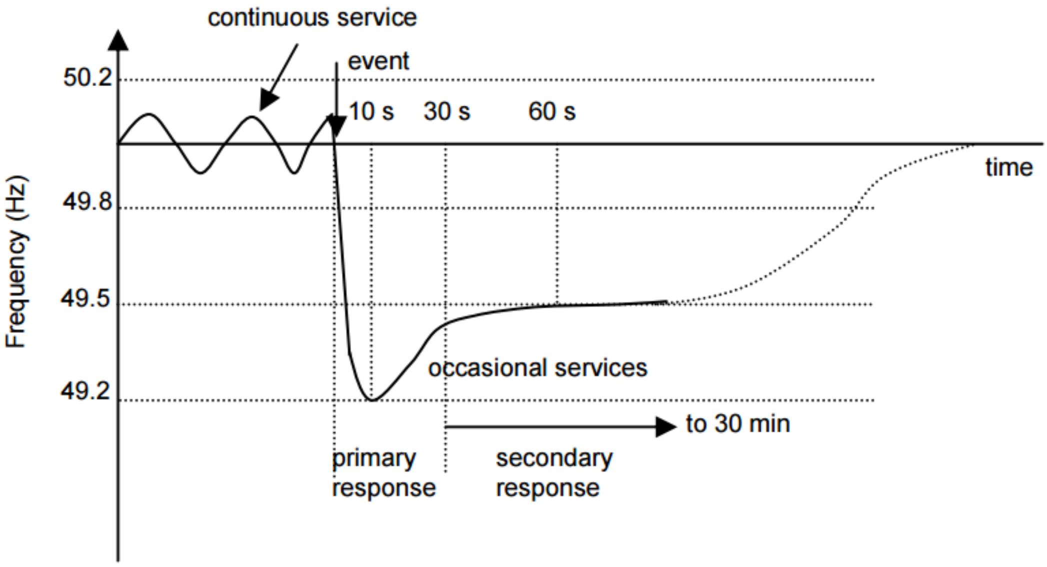


Figure 8-1: A proposed frequency control scenario, Source: [105]

A 10% power output increase above the available power for about 10 seconds after a large loss of generation is recommended for wind turbines synthetic inertia response [107]. In [108], it is shown, when over-rating the wind turbines by 20% in above rated operating condition, it takes at least 20 minutes for converters to reach their thermal limit. Thus, demanding 10% extra power for few seconds in above rated operation, for synthetic inertia response, does not cause any harm to the wind turbines' power converters. Moreover, power converters are normally designed to safely deliver about 10% extra power above their rated value [99].

The wind farm control algorithm developed in Chapter 6 includes synthetic inertia response and droop control for primary frequency response. The primary frequency response is implemented according to the recommendations in the UK grid code [105].

8.1 Testing the wind farm model for Primary frequency response

A wind farm model is developed using the developed wind farm model builder tool to investigate the primary frequency response of the wind farm controller. Specifically, power curtailment, synthetic inertia and droop control modes are simulated for a wind farm model of 10 turbines with the farm layout shown in Figure 8-2. In this layout wind turbines are positioned in two parallel lines with their rotors perpendicular to the wind direction. Wind turbines are separated by 800m in both lateral and longitudinal directions.

8.2 Providing reserve power for droop control

In the first simulation, the power curtailment control mode of the wind farm controller is investigated at $8ms^{-1}$ mean wind speed with 10% turbulence intensity in a 50MW wind farm. The wind farm controller is configured to achieve a 2MW power curtailment.

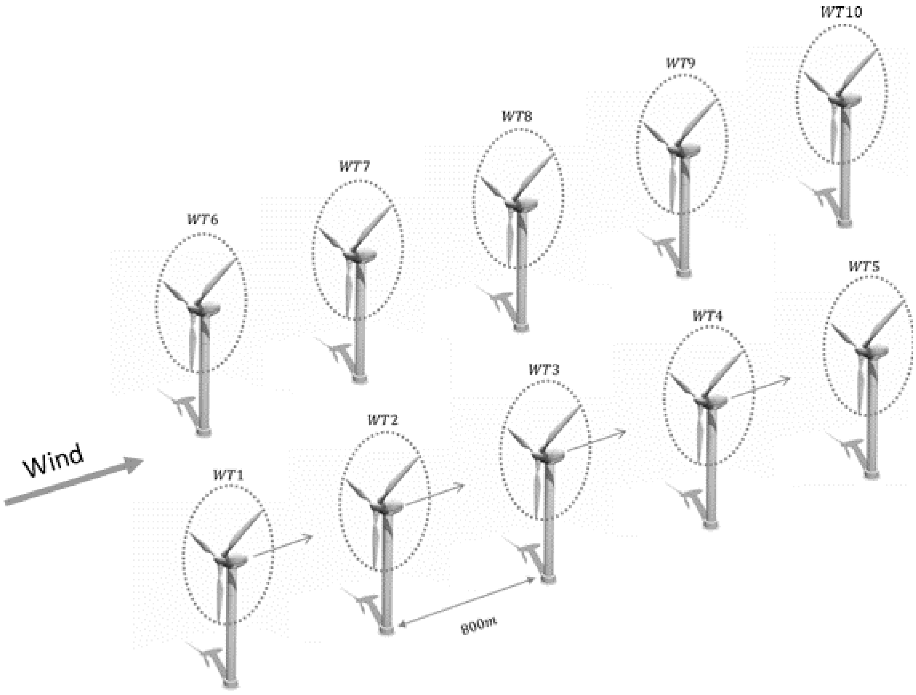


Figure 8-2: Wind farm layout with 10 turbines

Figure 8-3 shows the total power output of the wind farm before and after 2MW power curtailment. The difference between the power outputs is also depicted in the graph. In order to avoid introducing a feedback loop between wind turbines and wind farm controller, the curtailment control algorithm is developed in an open loop manner. Figure 8-4, illustrates the power set-point adjustments of individual turbines. The change in the power set-points in different traffic light regions is evident. Figure 8-5 depicts the wind turbines' traffic light flags. Level 1 is where wind turbines are operating in the green regions. Similarly, level 2 and level 3 indicate where wind turbines are operating in the amber and red regions. Therefore, transition between different traffic light regions for each wind turbine can be seen in one graph. Before the power set-point adjustments at 1100sec, all wind turbines are operating in the green region (level 1 in the graph of traffic light flags). After the initial power set-point adjustments, wind turbines move away from the green region to different operating regions in the traffic light regime. The wind turbines' power set-point are then adjusted according to the power adjustment levels defined for each traffic light region.

The operation of the wind turbines in the generator speed-generator torque plane is depicted in Figure 8-7. As shown wind turbines are operating on a curve below the CP_{max} tracking curve in a safe and controlled operating manner. Some of the wind turbines' operating point move to the amber or red traffic light region and their contribution to the wind farm power curtailment is reduced to the levels allocated for these traffic light regions. Moreover, some of the wind turbines operate in the wind speeds below the PAC wind speed operating limit (6.5 ms^{-1}). As a result, their PACs go to the recovery process and became unavailable at some points during this control mode. In this circumstance, the power set-points of the wind turbines are readjusted, according to their operating condition, to deliver the required wind farm reference power. Figure 8-8 illustrates the wind speeds at which each wind turbine is operating and the low wind speed limit at which the PAC enters into recovery process. Some of the turbines are operating at wind speeds below this low wind speed limit, and thus other turbines must contribute more to meet the requested wind farm reference power delivery. The individual power set-point of the wind turbines in Figure 8-6 clearly shows the power set-point coordination by the wind farm controller in these conditions. The power set-point adjustment of the individual wind turbines can also be mapped to the traffic light status flags of each individual turbine. Once a wind turbine operating point moves from the green to the amber region, the level of its power set-point reduces gradually to the amber region power adjustment level. Similarly wind turbines' power set-points reduce to zero, for those turbines operating in the red region.

As the results suggest, the power curtailment of the wind farm control algorithm is effectively implemented and can be used in the power curtailment control mode or to provide spinning reserve in the droop control mode.

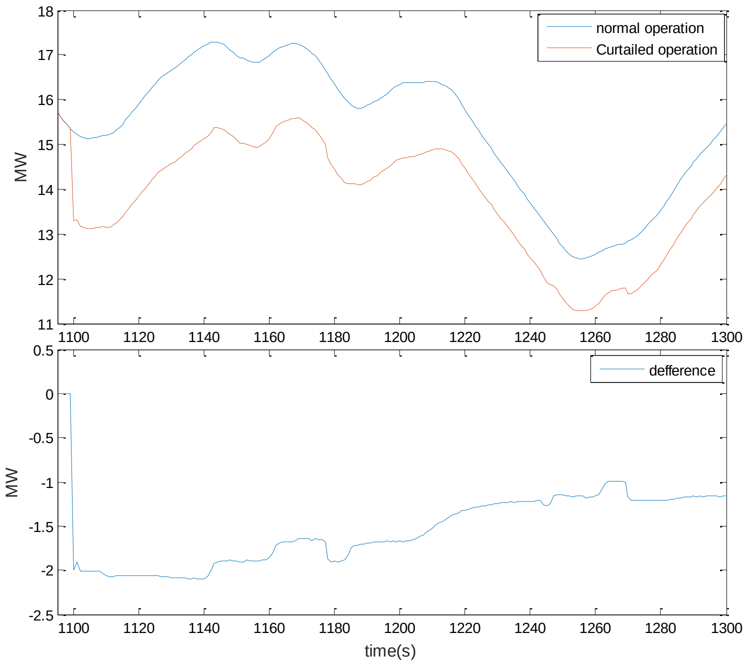


Figure 8-3: Total wind farm output in the power curtailment control mode

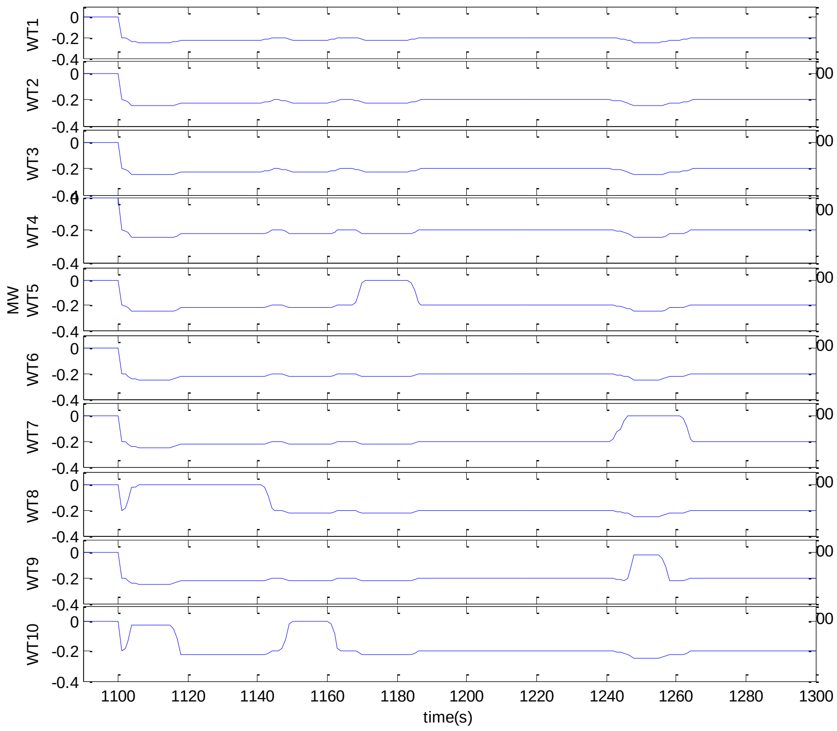


Figure 8-4: Wind turbines' power set-point in the power curtailment control mode

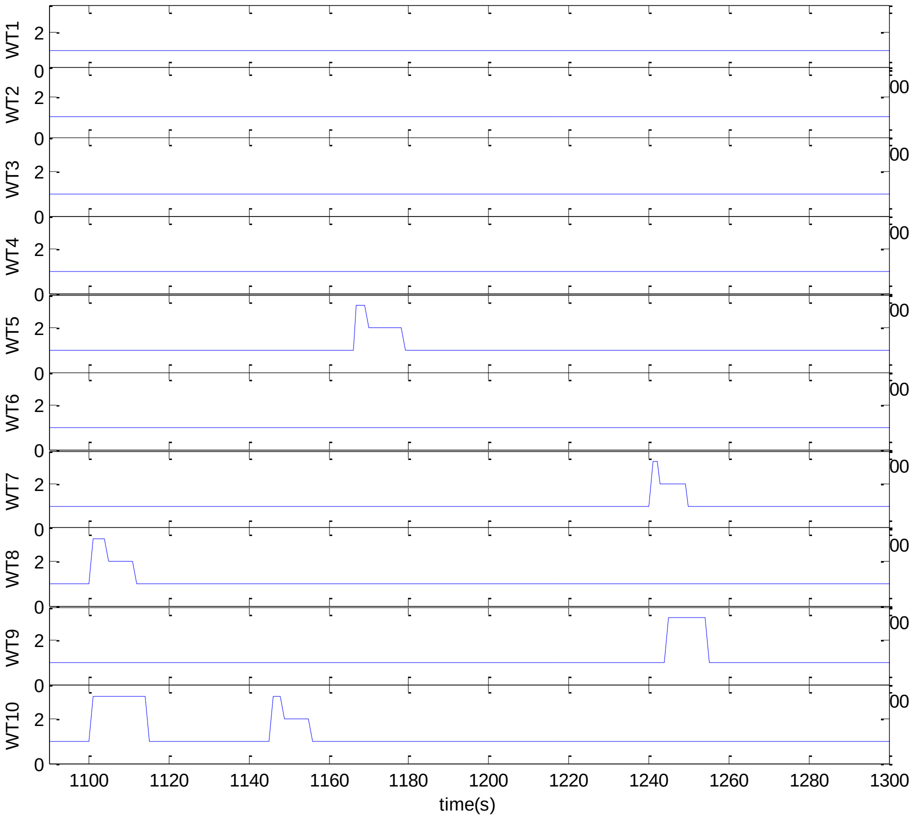


Figure 8-5: Wind turbines' traffic light flags in the power curtailment control mode

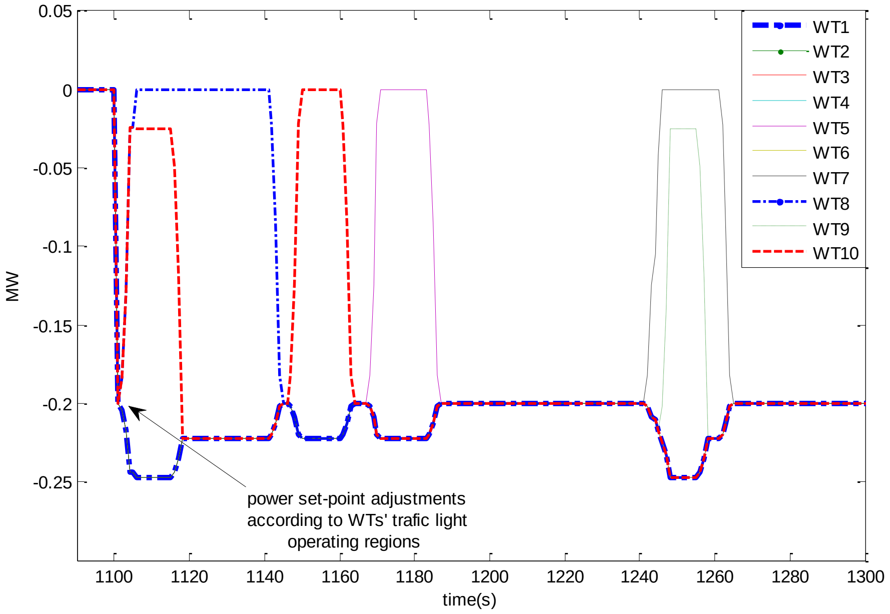


Figure 8-6: Wind turbines' power set-point compensation

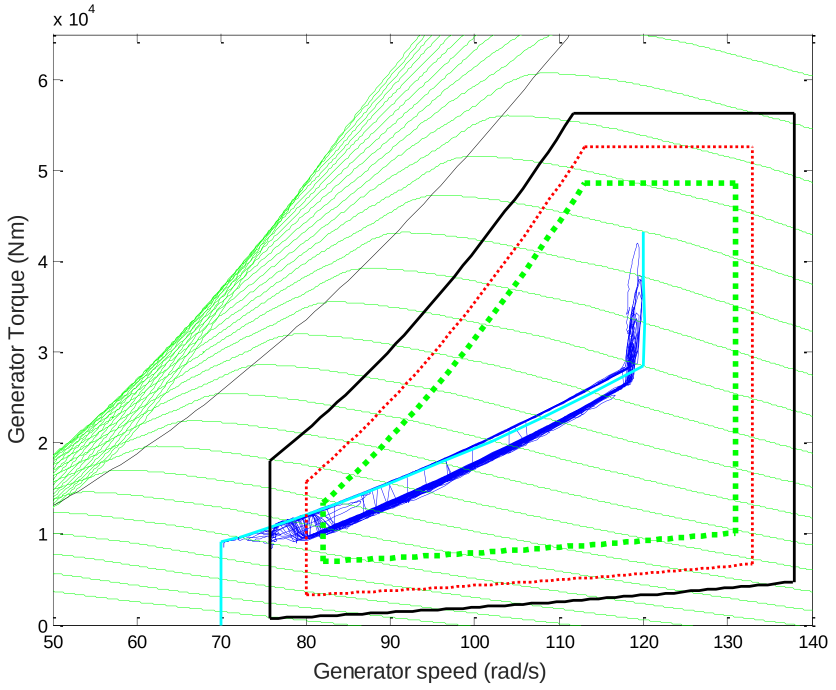


Figure 8-7: Wind turbines operation in the Generator speed-Generator torque plane

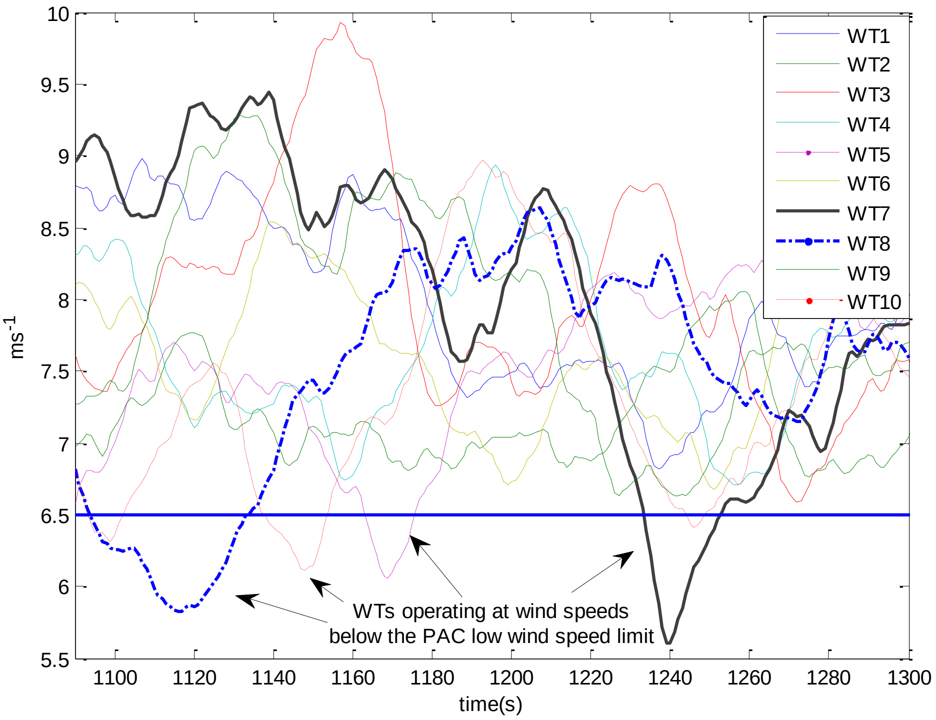


Figure 8-8: Wind turbines with their PAC unavailable due to low wind speed

8.3 Providing synthetic inertia response

In order to evaluate the performance of the synthetic inertia response of the wind farm controller, data from a real under frequency event in UK is collected and analysed. Figure 8-9 and Figure 8-10 illustrate the measured under-frequency event signal and its corresponding rate of change of frequency averaged over $100ms$. A number of simulations are conducted to evaluate the primary frequency response of the wind farm controller. A wind farm model with the layout shown in Figure 8-2 is used for the simulations.

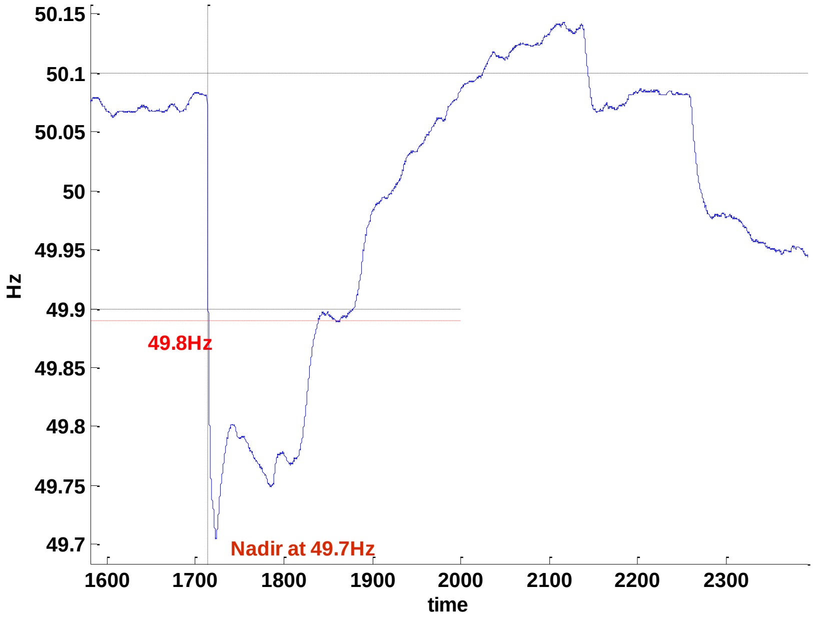


Figure 8-9: UK under-frequency event data 2011

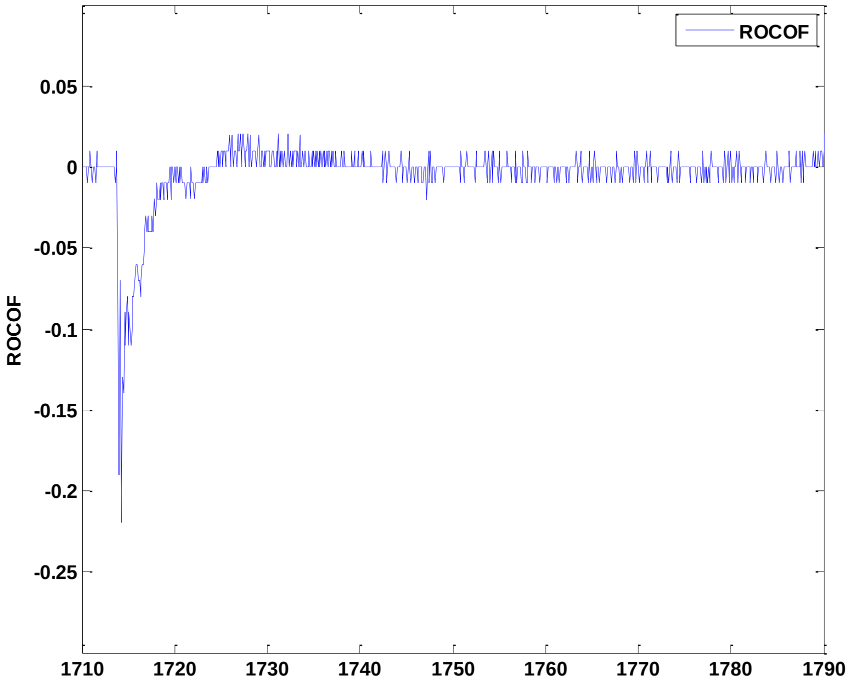


Figure 8-10: rate of change of frequency (ROCOF) of the frequency signal

In the first simulation the synthetic inertia response of the wind farm controller is investigated. Thus, in the wind farm control algorithm only the synthetic inertia control mode is activated and the droop control mode is bypassed. Since the operation of the synthetic inertia control mode is constrained by the rate of change of frequency, the small frequency deviations with slow rate of change are ignored. From the measured frequency graph in Figure 8-9 and its corresponding averaged rate of change of frequency in Figure 8-10, it can be observed that the synthetic inertia response is required at roughly 1714 sec in the simulation. In this control mode, all operating wind turbines contribute equally to the extra power delivery required by the synthetic inertia function. The level of inertia response of the wind farm at any given time depends on the total available power level adjustment in the farm. In this case, the power adjustment increment ($\Delta p_{inertia}$) for the synthetic inertia response for each turbine is 10% of its rated power (0.5MW). The value of $\Delta p_{inertia}$ can be modified by the wind farm controller as required.

Figure 8-11, depicts the required wind farm power level adjustment, computed by the wind farm synthetic inertia algorithm, to compensate for this under frequency event and the total power level adjustment, delivered by the wind turbines. The total power before and after synthetic inertia response is also depicted in the figure.

As can be seen, the total power adjustment delivered, is lower than the required power adjustment computed by the synthetic inertia function in the algorithm. In this control mode each wind turbine provides an increment $\Delta p_{inertia} = 0.5\ MW$, ignoring the traffic light regime, the total power adjustment level computed by the synthetic inertia function in the wind farm controller must be 5MW. However, since WT9 is operating at low wind speeds, below the PAC operating low speed limit during the synthetic inertia response around 1714sec, only 9 turbines are available for extra power delivery during this control mode. Figure 8-12 shows the wind speed condition at WT9 during the synthetic inertia response. Since the computed power adjustment level for the synthetic inertia response is higher than the total available power adjustment level in the farm, the wind farm controller only dispatches the available power adjustment level of 4.5MW between the remaining operating wind turbines.

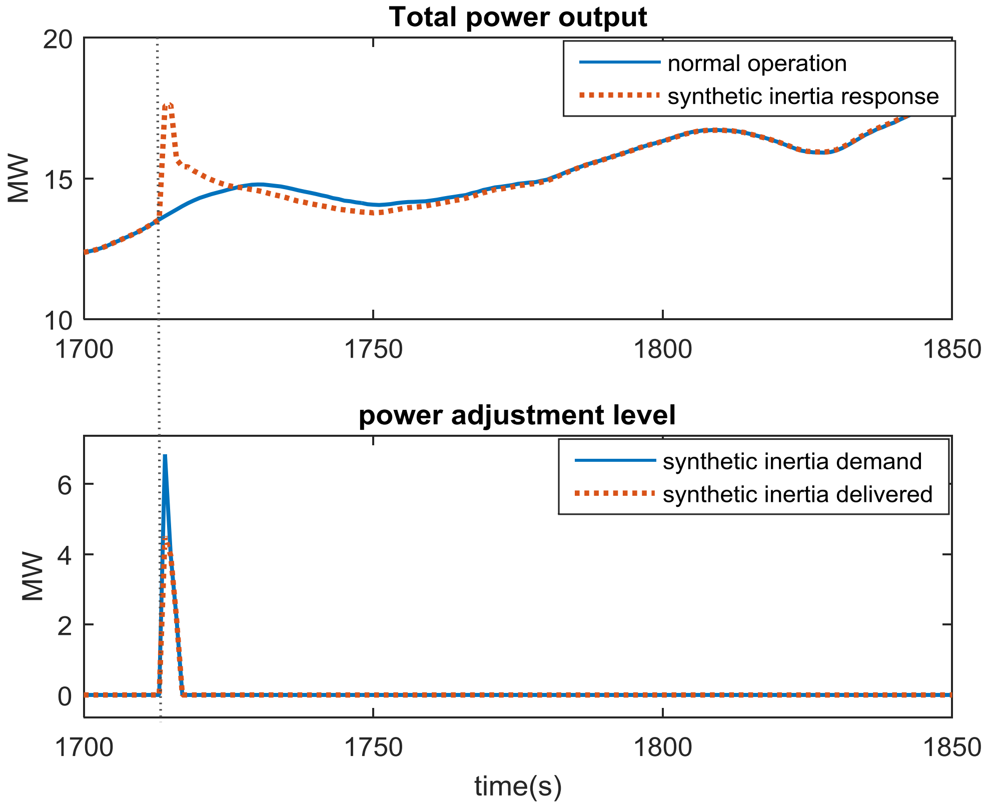


Figure 8-11: Wind farm synthetic inertia response required power and the actual delivered power

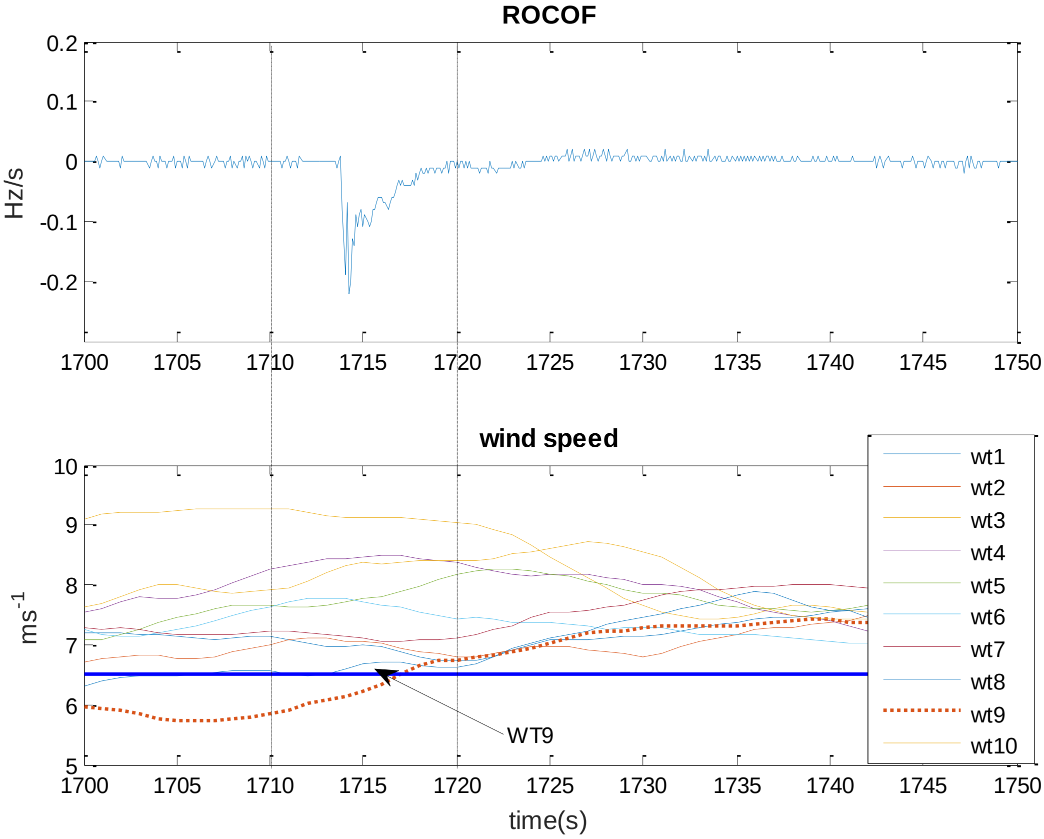


Figure 8-12: Wind farm synthetic inertia response

Figure 8-13 depicts the operating condition of the turbines in the generator speed-generator torque plane in this control mode. It is evident that the available wind turbines for power adjustment control are participating in the extra power delivery, ignoring the traffic light limits. As can be seen only 9 turbines are participating in the extra power delivery since WT9 is operating at low wind speed condition.

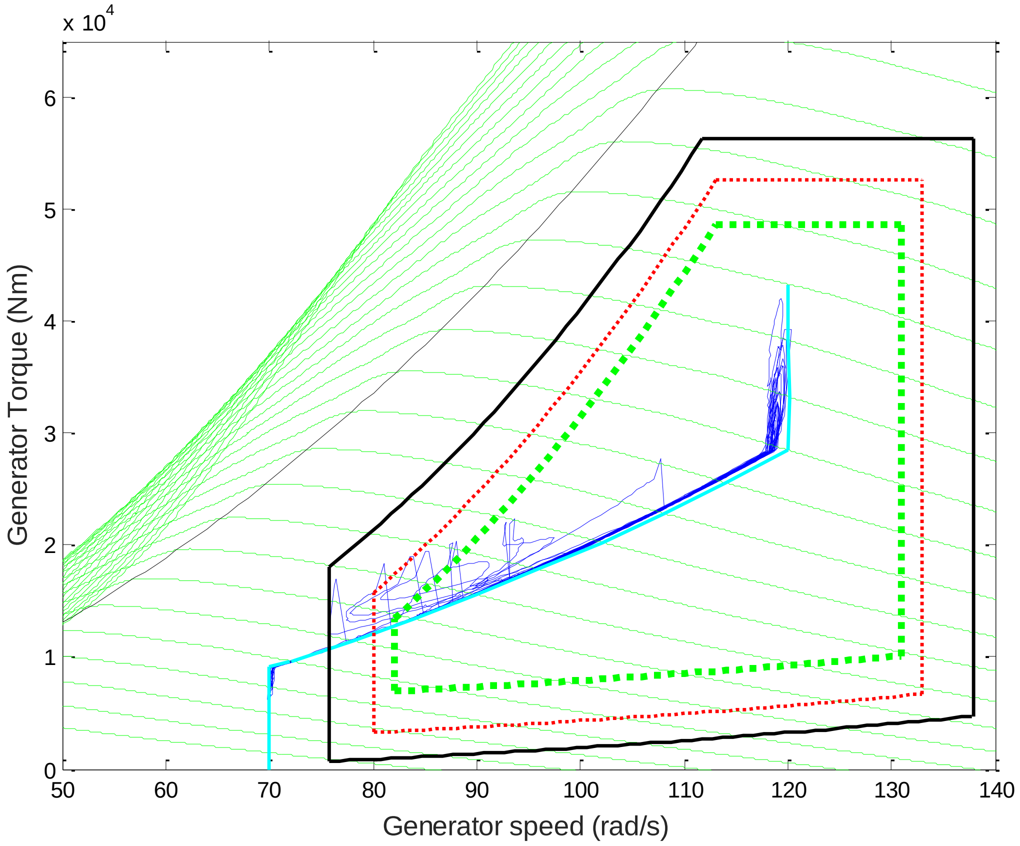


Figure 8-13: Wind farm synthetic inertia response

The recovery process of the wind turbines' PAC can be configured by the wind farm controller to achieve a fast or slow recovery process. The slow recovery mode is recommended to reduce the effect of PAC on the structural loads and allow the wind turbines to go back to the normal operating point in a safe operating manner. Figure 8-14 and Figure 8-15 show the difference between the fast and slow recovery process of one of the wind turbine's PAC.

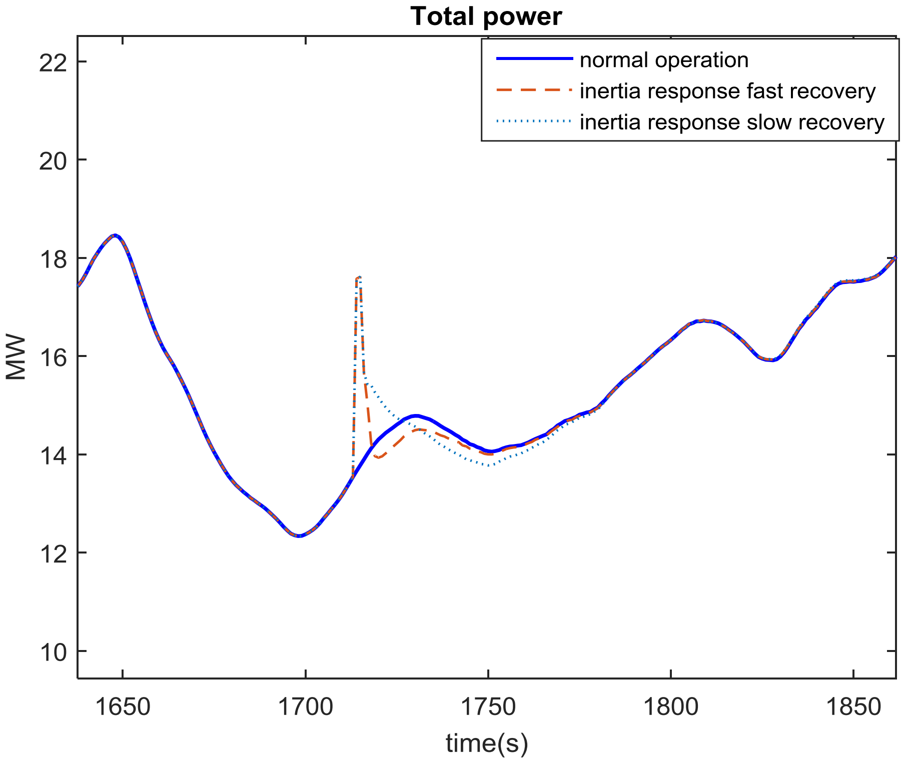


Figure 8-14: Wind farm synthetic inertia response with fast/slow PAC recovery

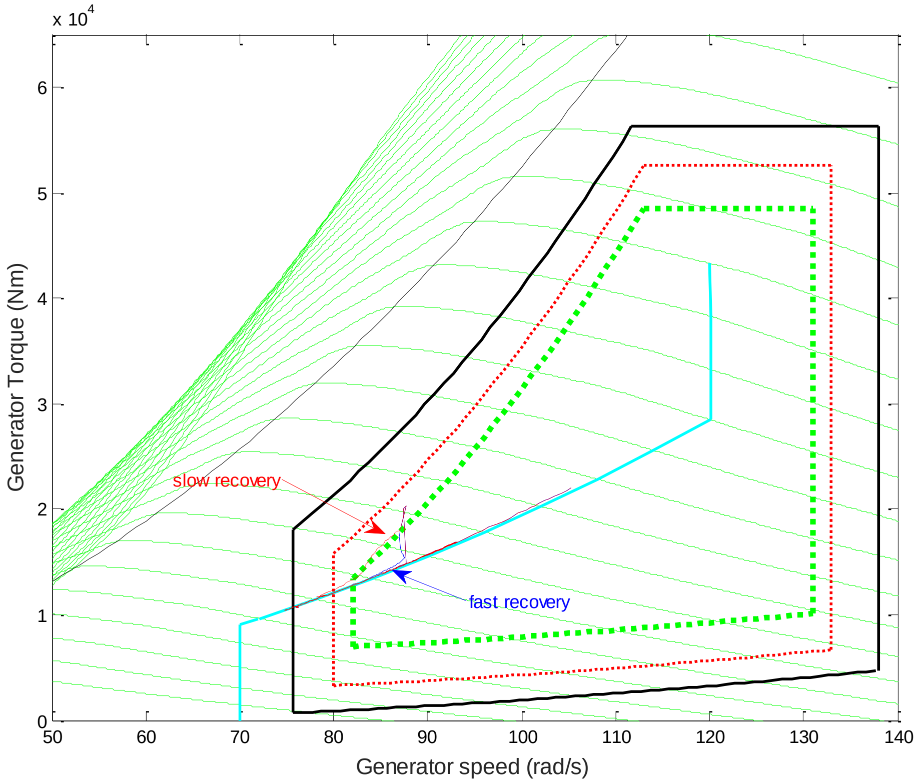


Figure 8-15: Fast Slow recovery comparison for one WT

From the generator speed-generator torque graph, it is observed that wind turbines may be able to provide more extra power adjustment for the synthetic inertia response before they hit the black limits and start their recovery process. Therefore, another simulation is performed with higher level of synthetic inertia response at each turbine. The synthetic inertia adjustment power level of 1MW is considered for each turbine. Figure 8-16 and Figure 8-17 show the level of synthetic inertia achieved in this simulation. As can be seen, the total dispatched power adjustment between the turbines is equal to the computed adjustment power for synthetic inertia response in the algorithm. The generator speed-generator torque graph shows the increased level of power adjustment in comparison to those in Figure 8-13.

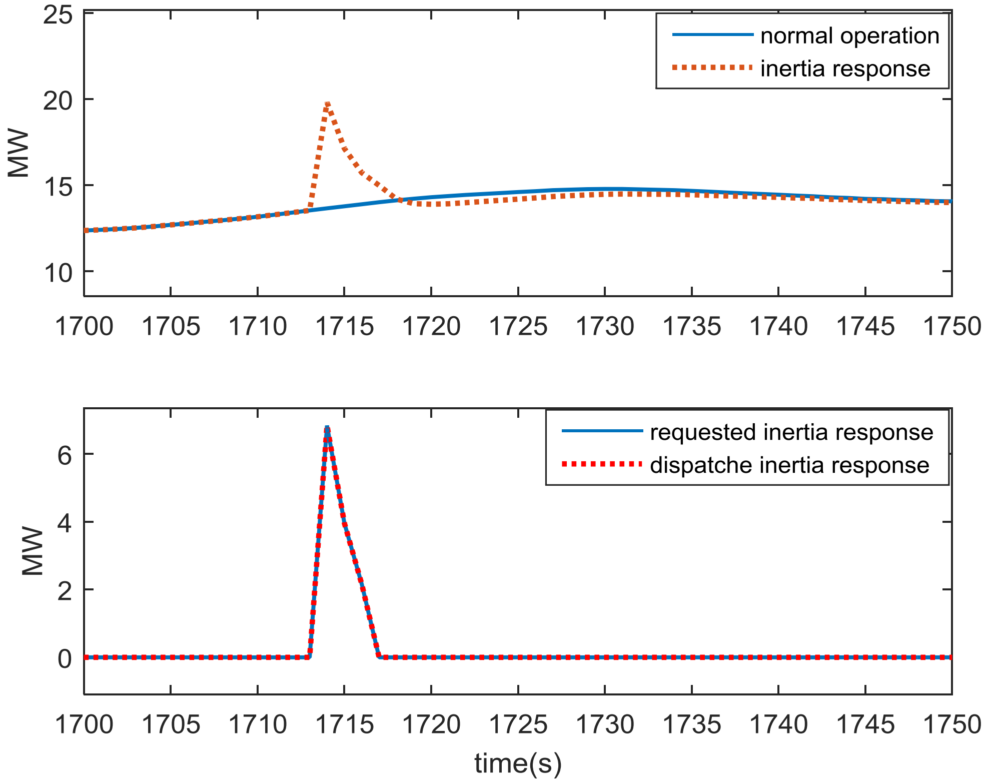


Figure 8-16: Synthetic inertia response with higher adjustment power level at each WT

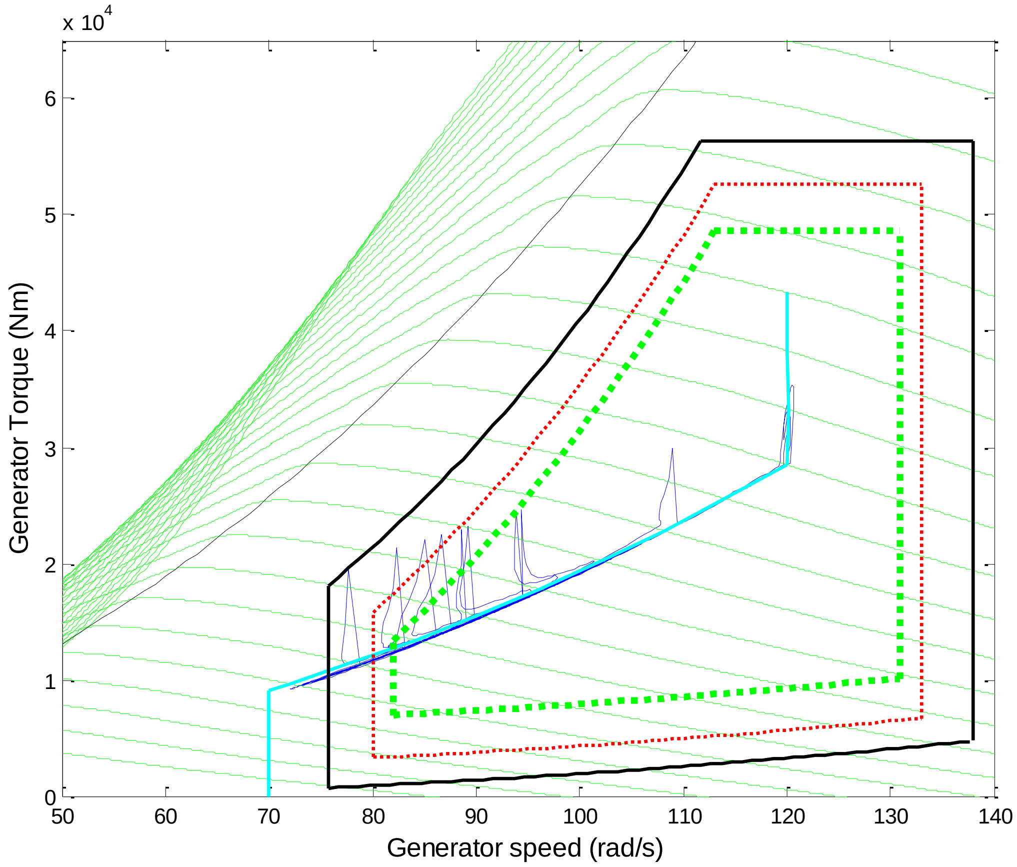


Figure 8-17: Wind turbines operation with 1MW adjustment power in the synthetic inertia control mode

Once the rate of change of frequency is maintained within the defined thresholds for synthetic inertia, the wind farm control algorithm switches to the droop control mode to help stabilise the grid frequency. The synthetic inertia and droop control functions are defined to operate simultaneously. Therefore, a combined control action of the synthetic inertia response and droop control is possible. In the next simulation, the same frequency event shown in Figure 8-9 is used to investigate the performance of the wind farm controller for the combination of Droop control and synthetic inertia response.

8.4 Droop control

Synthetic inertia response alone is not adequate to maintain the grid frequency at its nominal value. Additionally, a droop controller is required to reduce the grid frequency deviations to a level within a defined thresholds [103]. The secondary frequency response can also be implemented to eliminate the steady state error in the grid frequency using controllers with integral action.

Droop control of the conventional power generators is implemented using the governor response of conventional generators to deviations in the grid frequency. Since, the speed of generators is directly coupled to the grid frequency, any deviation in the grid frequency, as a result of mismatch between the generated power and load in the system, can be detected and compensated by the governor controller of the generators. The governor controller of the conventional generators compensates for the grid frequency deviations by controlling the speed of the generator through the prime movers. In a similar manner, the droop control response can be implemented for the wind power plants. A droop control curve is used to compensate for the perturbations in the grid frequency.

A droop slope of 3-5% is required by the UK National Grid code for the droop control curve [109]. The grid frequency must be maintained within a defined threshold of $49.8Hz \leq f \leq 50.2Hz$, by adjusting the active power of the wind turbines.

Normally, wind power plants are required to operate in a curtailed control mode, with 5% spinning reserve at any wind speed, to provide enough head room for droop control response [99]. Therefore, the available power curtailment control function, which is implemented within the droop control function, is configured to provide the required spinning reserve in this control mode.

In order to demonstrate the performance of the wind farm controller for primary frequency response, including droop control and synthetic inertia, a number of simulations are performed using the wind farm model with the layout shown in Figure 8-2. The frequency input to the controller is the UK under-frequency event data depicted in Figure 8-9. The droop curve depicted in Figure 6-3 is implemented in the wind farm control algorithm for this control mode.

The first simulation is conducted for the case where the wind turbines are participating in the primary frequency response with no spinning reserve.

Figure 8-18 and Figure 8-19 illustrate the individual and total power set-point of the wind turbines computed for this under-frequency event. As can be seen, both synthetic inertia and droop control functions compute the required power adjustment level for the wind turbines corresponding to their operating conditions. For the periods where the frequency fluctuations are within the predefined dead-band according to the droop control curve the total reference power of the wind farm is zero. Simultaneously, the rate of change of frequency is measured and compared with the defined thresholds.

During the synthetic inertia response wind turbines provide the maximum extra power available to them to reduce the rate of change of frequency. Once the rate of change of frequency is slowed down, the droop control function becomes responsible for reducing the frequency fluctuations back to the nominal grid frequency. From the individual wind turbines power set-point adjustments in Figure 8-19, it can be seen that the wind turbines are constrained by the traffic light limits during the wind farm droop control mode. It is also evident that WT9 is not contributing to the synthetic inertia response due to low wind speed condition during the inertia response.

The operating condition of wind turbines during this under frequency event is depicted in Figure 8-21 in terms of traffic light flags. The transition between traffic light regions' power adjustment limits is evident in the graphs. In the graphs level 1, level 2 and level 3 represent operation of the wind turbines in the green, amber and red regions respectively.

Similarly, wind farm dispatch unit coordinates the power set-point of each wind turbine corresponding to its operating status so as to meet the total farm reference power demand. In this control manner, available wind turbines compensate for those with unavailable PAC, to fulfil the required wind farm power output adjustment.

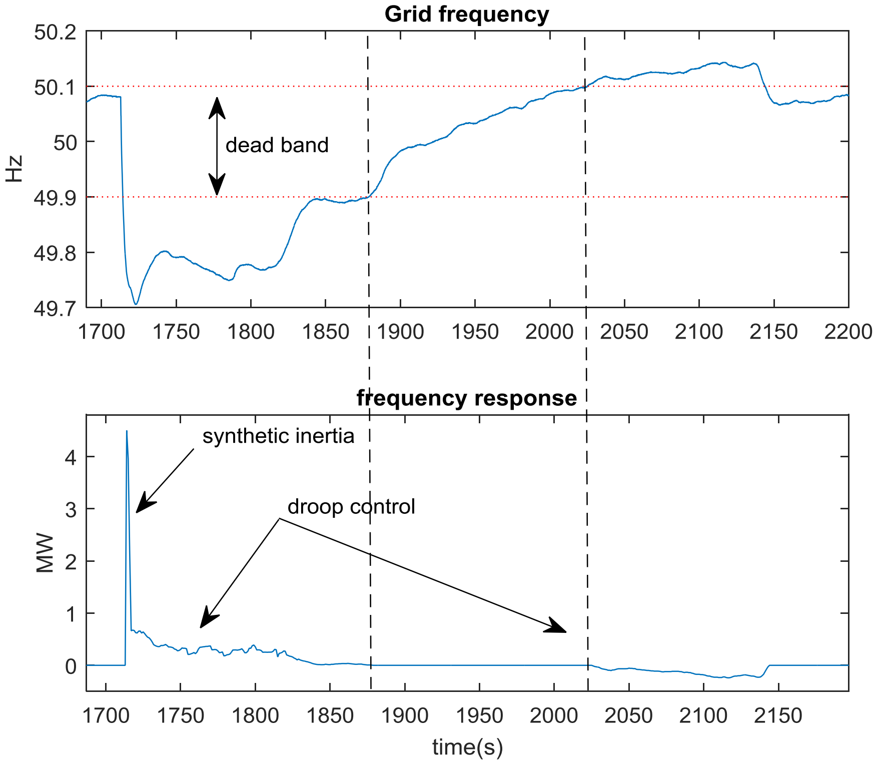


Figure 8-18: Wind farm reference power for primary frequency response no reserve

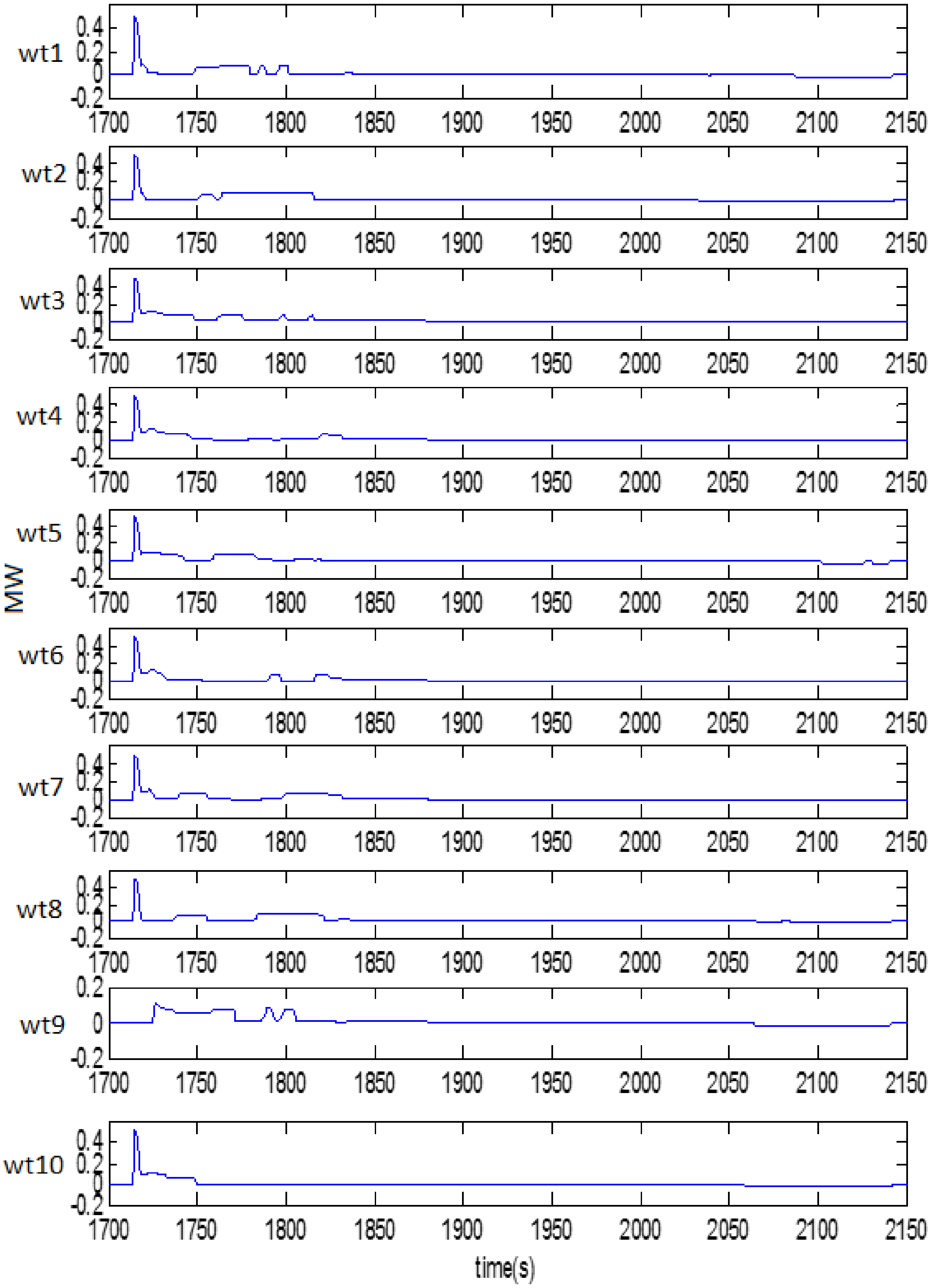


Figure 8-19: Wind turbines' power set-point for primary frequency response no reserve

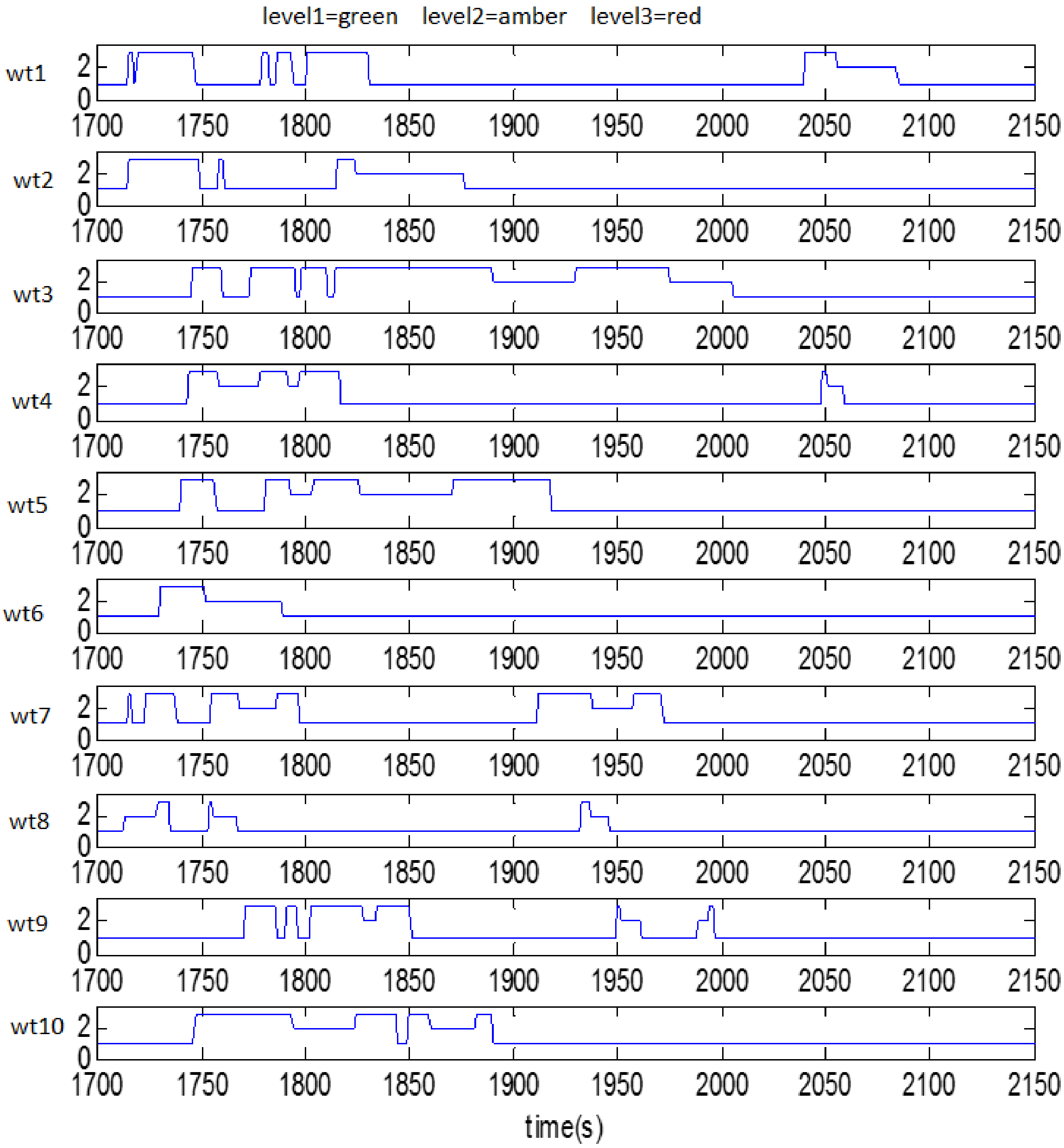


Figure 8-20: Wind turbines' traffic light status flags

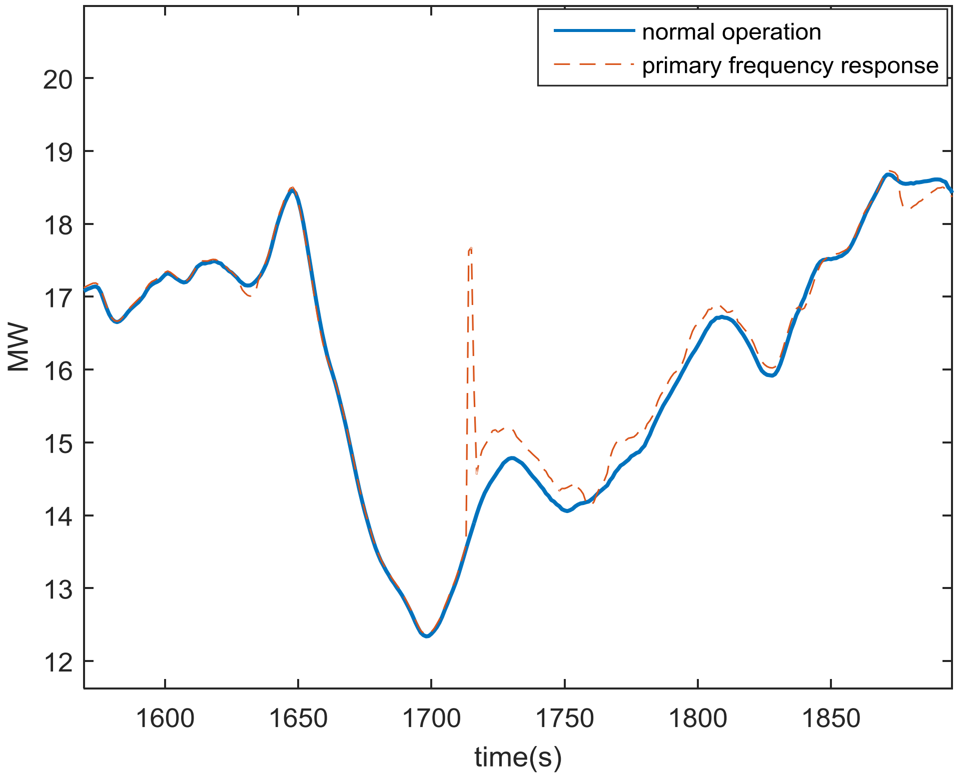


Figure 8-21: Wind farm primary frequency response total power output

Figure 8-21 illustrates the total power output of the wind farm corresponding to a combination of synthetic inertia response and droop control. The initial ROCOF response is evident in the graph. The sharp spike in the total power of the farm, shows the level of inertia response of the wind farm. As seen, the synthetic inertia response is performed immediately, after the ROCOF is exceeded from the defined thresholds, to help bring the system back to stability. As can be seen from Figure 8-20, during the droop control, roughly around 1750 seconds to 1770 second, most of the wind turbines either operate in the amber or red traffic light regions. As a result, the level of adjustment power is reduced by the wind farm dispatch unit to achieve sensible power adjustment increments Δp_i for each wind turbine. The total power adjustment is determined as the minimum between the total available adjustment power level in the farm and the computed adjustment power by the droop control function.

The difference between computed and dispatched wind farm power adjustment level for synthetic inertia response is depicted in in Figure 8-22. Once again, since the WT9 is not available for the synthetic inertia response, the total power adjustment level is altered to be 4.5MW.

Figure 8-23 illustrates the wind turbines operating condition in the generator speed-generator torque plane. After completion of the synthetic inertia response, the wind farm controller coordinates the power set-point of the turbines according to their operating condition to realise the droop control mode. The power set-point of the turbines are adjusted while considering traffic light limits in this control mode. As can be seen, some of the wind turbines hit the black boundary limit and their PAC recovery process is triggered. In the recovery mode, wind turbines' operating point are driven back towards the optimal operating point on the C_{Pmax} tracking curve. It is shown that, even without available spinning reserve, the synthetic inertia response of the wind turbines is achievable.

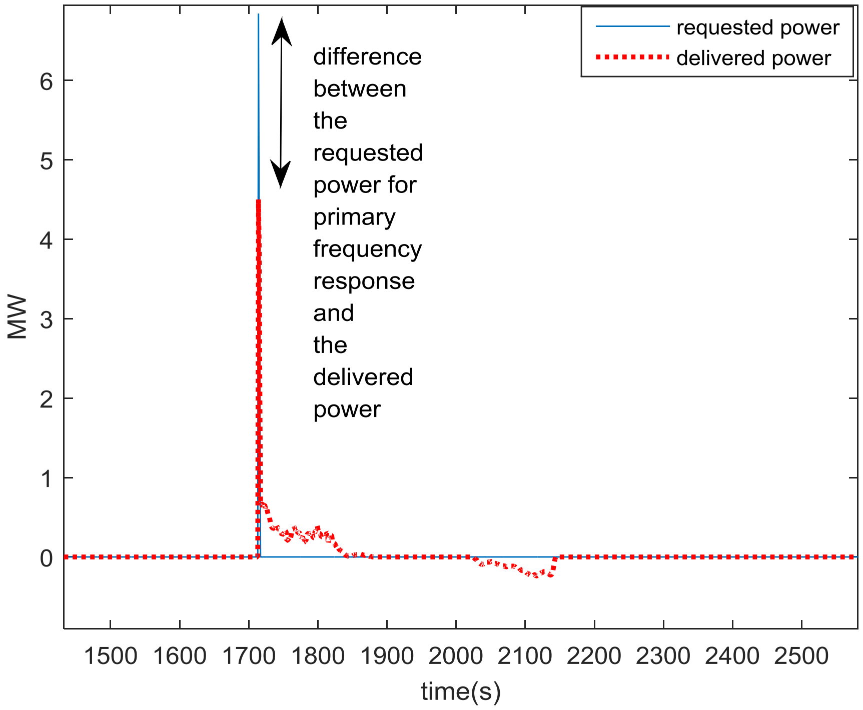


Figure 8-22: Difference between requested power and applied reference power at inertia response mode

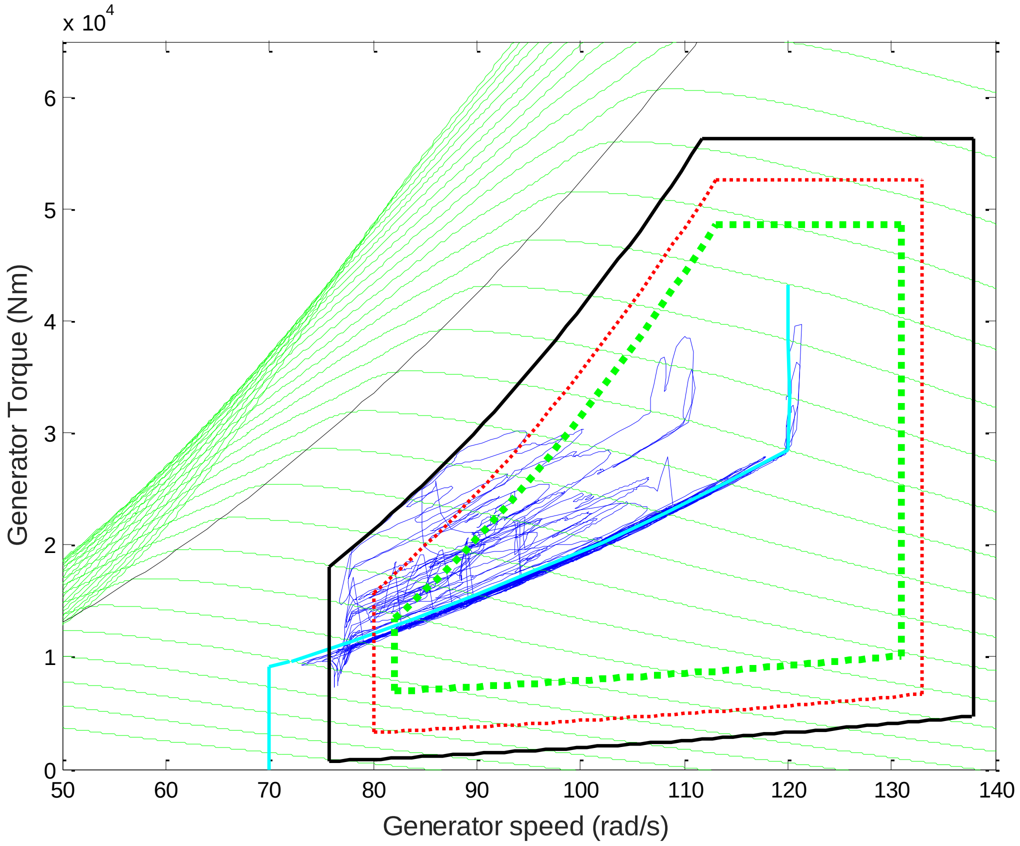


Figure 8-23: Wind turbines operating condition at primary frequency response with no reserve

In the next simulation, the primary frequency response with spinning reserve available in the wind farm is investigated and compared with the previous simulation.

In this control mode, the curtailment control function in the droop control path is configured to provide some spinning reserve for droop control response. The required power adjustment level, to deliver a percentage of spinning reserve, is determined as the basis of the total available power level adjustment of the turbines taking into account traffic-light limits. A 5% spinning reserve from the total available power in the farm is considered for this simulation.

Figure 8-24 illustrates the total power output of the farm before and after the frequency response for both control modes with and without spinning reserve. As can be seen, a continuous droop control is achieved for the case with available spinning reserve, whereas in the case with no spinning reserve, poor droop control operation of the wind farm can be observed around 1760 seconds.

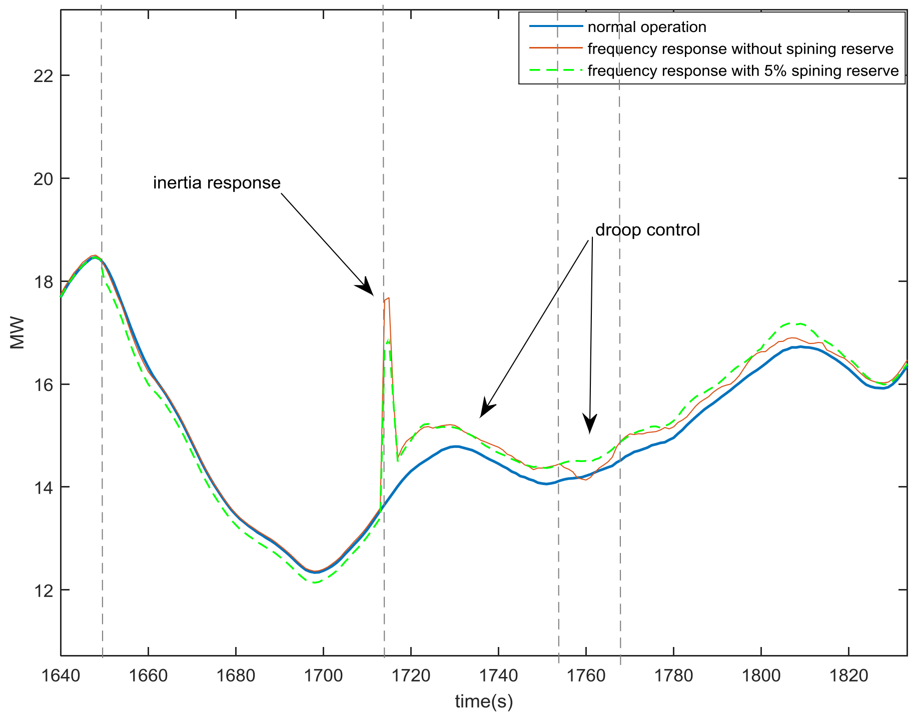


Figure 8-24: Wind farm power output at primary frequency response control mode with and without reserve

Figure 8-25 and Figure 8-26, depict the difference between the total power delivered before and after wind farm operation in the primary frequency control mode. The power set-points dispatched between the turbines are also shown in the graph. As can be seen, during the curtailed control mode, to deliver the required spinning reserve, the wind turbines are affected by the traffic light limits. The power set-point of the turbines entering the amber or red region are reduced according to the traffic light regime. Therefore, safe operating conditions of the wind turbines are ensured during this control mode. Similarly, as shown in Figure 8-27, during the power adjusting control modes of the wind farm controller, the wind turbines are kept operating within the predefined safe operating boundaries.

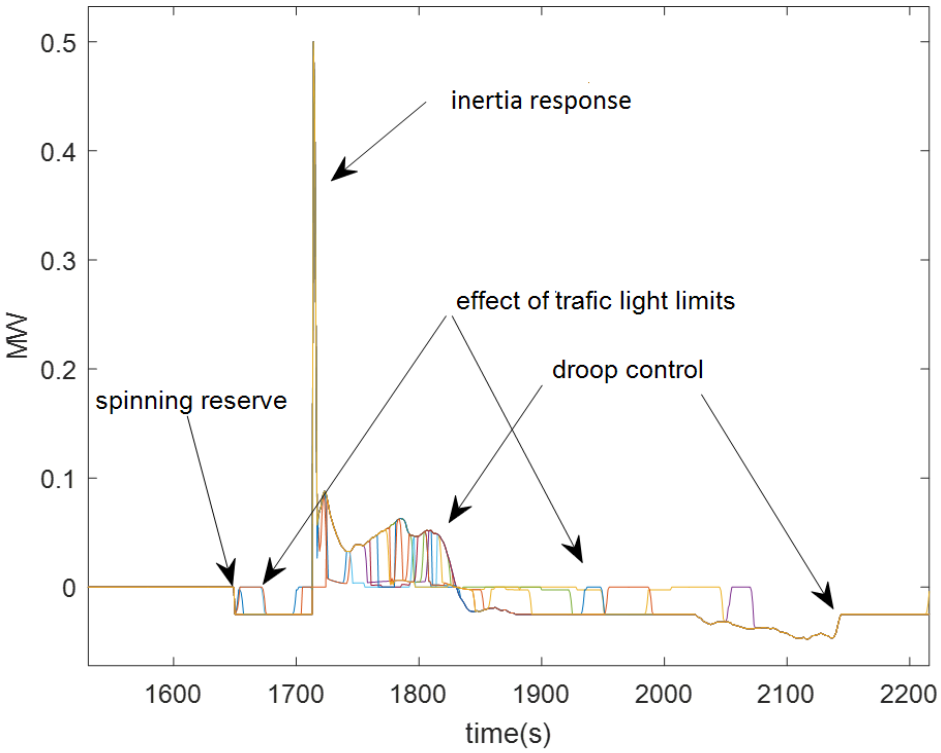


Figure 8-25: Wind turbines' power set-point for primary frequency response with reserve

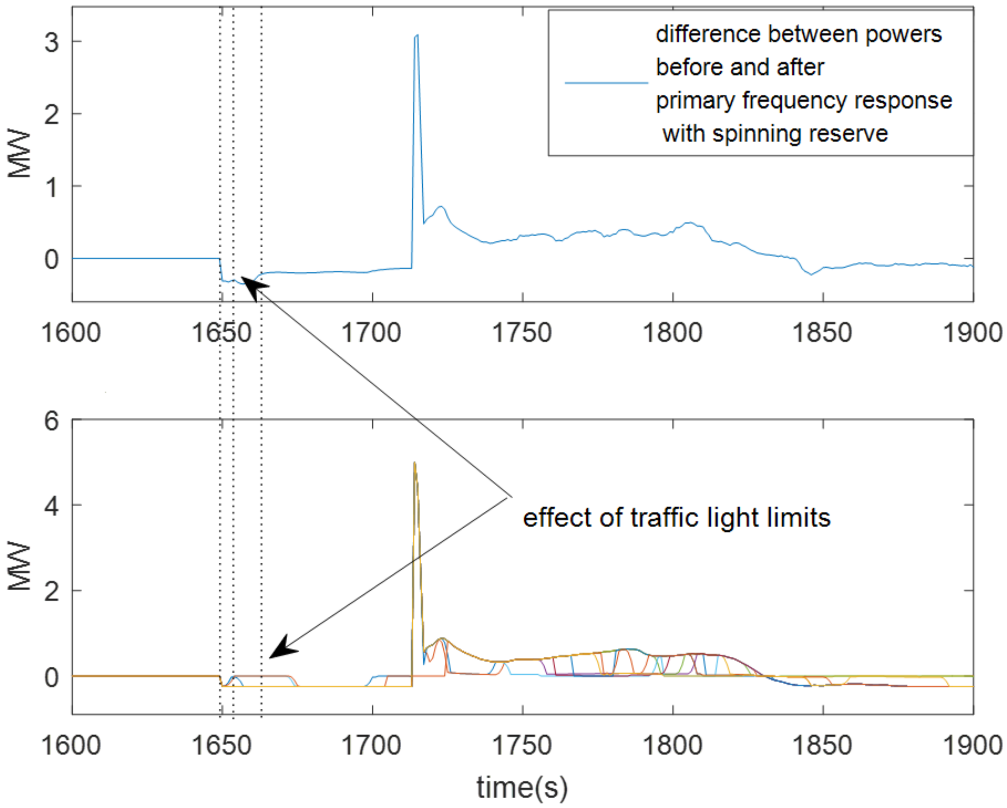


Figure 8-26: Wind turbines operating condition at primary frequency response with reserve

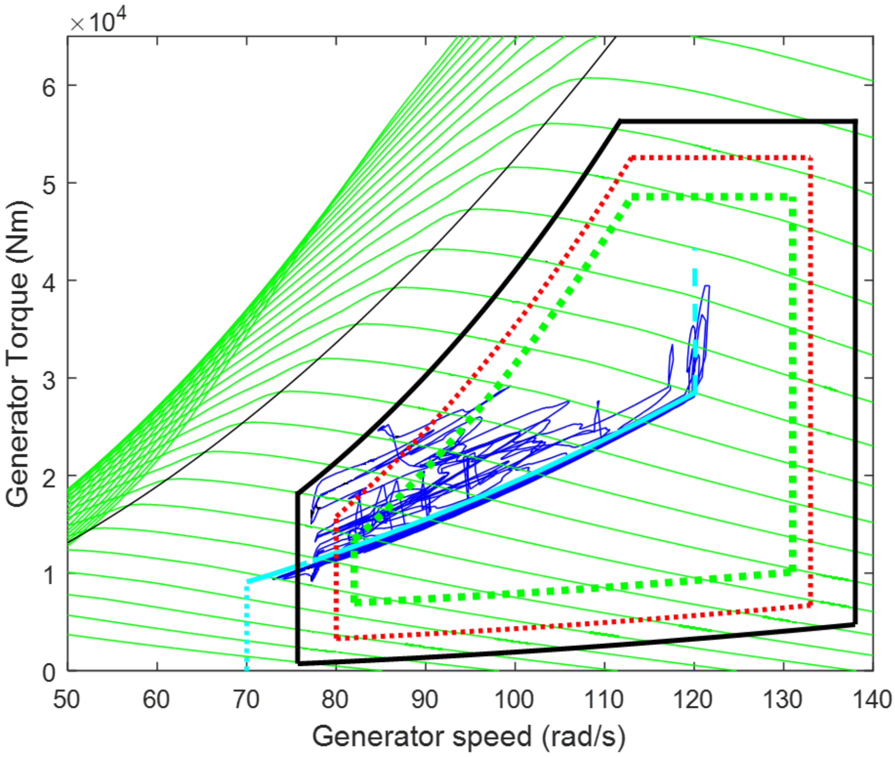


Figure 8-27: Wind turbines operating condition in generator torque/speed plane

8.5 Concluding remarks

Primary frequency response is becoming an essential control objective for the grid connected wind power plants. It is shown that by adopting a proper control strategy wind turbines are capable of participating in the primary frequency response in a safe operating manner. Simulation studies suggest that the wind power plant performance improves, when a small spinning reserve is available in the droop control mode. In this wind farm control configuration, a certain level of extra power adjustment is possible by the wind turbines corresponding to their operating status.

The level of synthetic inertia response depends on the wind turbines' PAC availability and can be adjusted as required. Wind turbines' PACs, can reject adjustment of the power set-points during its operation in the recovery mode. Therefore, wind farm controller adjusts the total power adjustment level according to the wind turbines' PAC availabilities and their operating status in the traffic light regime.

The wind farm control algorithm utilises a combination of droop control, synthetic inertia and power curtailment to provide adequate control action for primary frequency response. Safe operation of the wind turbines is ensured by the PACs taking account of their operating status. The simulation results suggest that primary frequency response with adequate spinning reserve is possible for large wind power plants.

Chapter 9: Wind farm coordination control for performance improvement

WIND farm control with power/load optimisation objectives, particularly for closely-spaced wind turbines, has been investigated in the recent literature [114], [115]. Since, available sites for large wind farm developments are limited, it is desirable to deploy more wind turbines in wind farms [118]. However, the control strategies of most variable speed wind turbines are defined so as to maximise the power output by adopting the optimal operating points at any operating condition. Thus, closely positioning wind turbines in a wind farm, potentially may increase the wake effect losses. According to EWEA, wake effects in a wind farm could account for a significant power reduction, during the operating life of the wind turbines [96]. It is desirable to reduce the wake effects and increase the array efficiency in a wind farm with closely-spaced wind turbines.

There are two main concepts widely utilised in the literature for reducing aerodynamic interaction between wind turbines. The first concept is known as Heat & Flux, which refers to an idea of adjusting the power set-point (axial induction factor) of the wind turbines at the upwind side, on the boundaries of a wind farm, to reduce the wake deficits and increase the total yield [116]. The second concept is on the basis of deflecting the wake utilising the yaw control of the upwind turbines and reducing the wake interaction effect on the downwind turbines.

The principle behind the Heat & Flux control concept is to alter the tip speed ratio of the upwind turbines, in order to reduce the wake effect on the downwind turbines [116]. In theory, by derating the upwind turbines operation, wind power at downwind turbines increases. Hence, by coordinating the wind turbines operation the total power output of the wind farm can be increased.

In [116], a 1:400 scaled model of an offshore wind farm is utilised to investigate the power performance of the wind farm model in derated control strategy. The wind turbines' axial induction factor are adjusted, by trial and error, to achieve improvements in the total power performance. It was suggested that by reducing the axial induction factor of the most upwind turbines by 20% the total power capture can be increased by 2%.

In [120], an optimisation control approach using genetic algorithm methods is adopted to obtain a set of optimal pitch angles for individual wind turbines, so that the total power production in the farm is maximised. The author suggested that by optimising the wind turbines' pitch angle the total aerodynamic output power can be increased by 4.5%.

Similarly, in [117], [66], optimisation techniques are used, based on the game theoretic algorithms and extremum seeking control methods to obtain a set of joint axial induction factor that maximises the total power output of the farm. A total power output increase of around 4% is reported in these literatures.

Since heat and flux concept is widely used in the literature, in this Chapter a similar approach is used to investigate the power maximisation objective of the wind farm controller through coordinating the power set-points of the wind turbines. The wind farm control algorithm developed in Chapter 6 is modified to accommodate the power maximisation objective of the controller. Because the wakes meander and the extent of wake interaction varies and because the penalty on operating a turbine with a 20% reduction in the induction factor [116], when there is no wake interaction, is so large compared to the potential gain, strategies dependent on complex optimisation of the operating points for the wind turbines in a farm are avoided. Instead, two simple strategies are investigated. The first, Section 9.1, is to maintain a fixed profile for the set points that is independent of wind conditions. The second, Section 9.2, is to seek to keep the power output of all turbines the same.

9.1 Power maximisation with constant reduction profile

The power maximisation follows the heat and flux control concept explained earlier. The simulations are performed for a wind farm model with a layout shown in Figure 9-1.

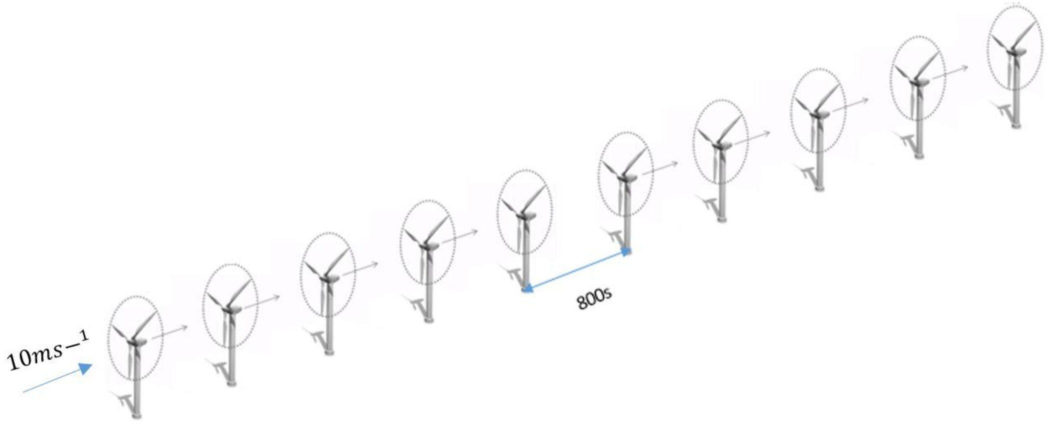


Figure 9-1: Wind farm layout for power optimisation control and power maximisation control

Initially, only the power set-point of the most upfront turbine in the row is adjusted to reduce the wake effect on the other turbines and possibly increase the total power output. Once a power set-point for the first turbine, in which the total power output of the farm cannot be increased any further by adjusting this power set-point is obtained, the power set-point of the first turbine is kept at that value, and the process is repeated for the next wind turbines in the row. A more precise optimisation process is not required since the same power set-point reduction pattern is to be used in all wind speeds and turbulence intensities.

Table 9. 1 contains a set of power set-points obtained from trial and error process for power maximisation at $10ms^{-1}$ wind speed with no turbulence.

WT1	WT2	WT3	WT4	WT5	WT6	WT7	WT8	WT9	WT10	Mean power
-10%	-5%	-5%	-3%	-1%	-0%	-0%	-0%	-0%	-0%	25.152 MW
-10%	-7%	-5%	-3%	-1%	-0%	-0%	-0%	-0%	-0%	25.153 MW

Table 9. 1: Power set-points reduction for power maximisation

Figure 9-2 illustrates the mean power outputs of the turbines for these power set-point adjustments. The increased level of power production at downstream wind turbines is evident. Figure 9-3, depicts the total power output of the wind farm before and after power set-point adjustments. It is clear that the power output of the wind farm is increased by almost 1% for both set of power set-point adjustments.

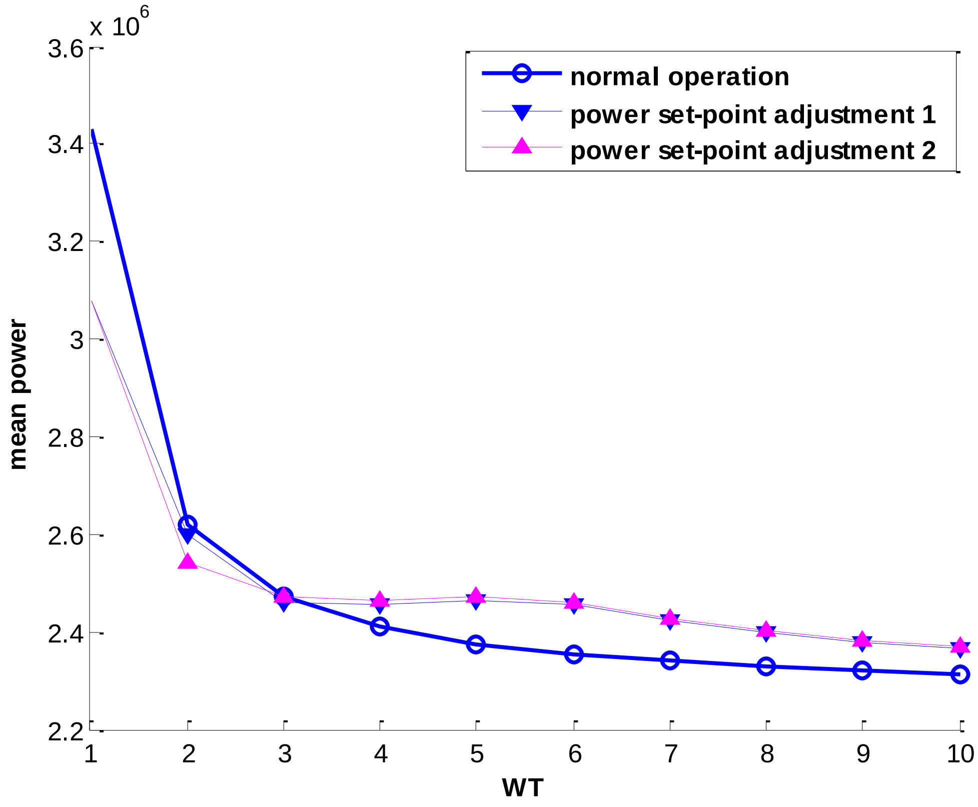


Figure 9-2: Power set-point adjustment for power maximization

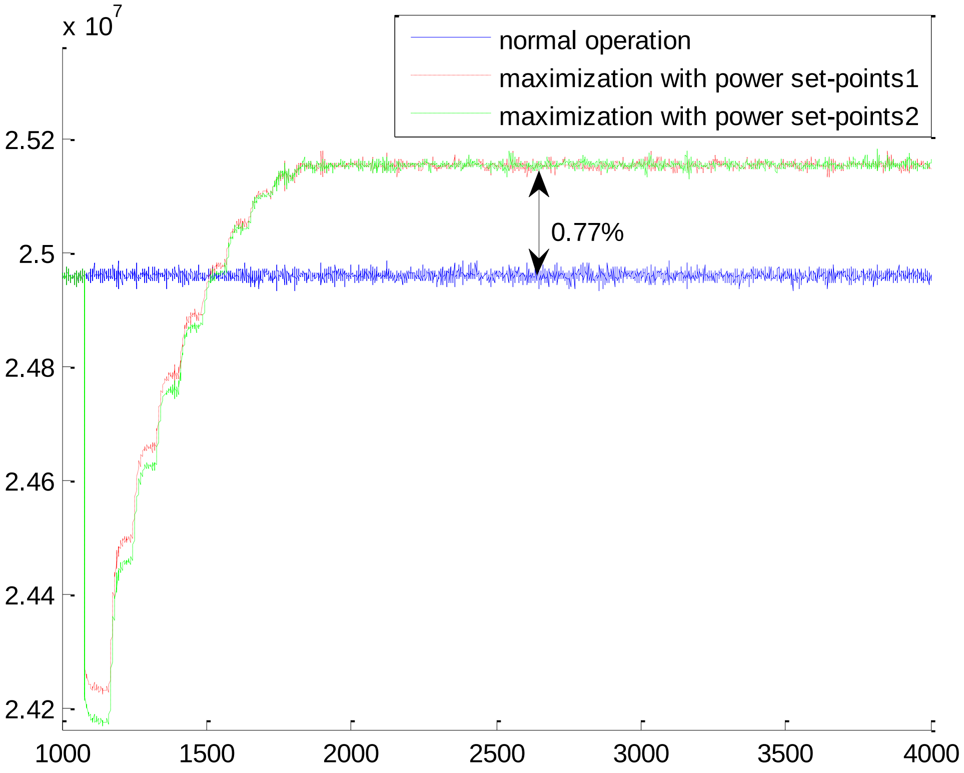


Figure 9-3: Total power before and after maximisation

The power set-point reductions obtained for the power maximisation at $10ms^{-1}$ is used at different below rated wind speeds as a common power set-point pattern for power maximisation. As can be seen from Figure 9-4 and Figure 9-5, this power set-point reductions pattern is applicable to a range of below rated wind speeds for power maximisation.

The mean total power outputs of the wind farm before and after power maximisation at different wind speeds are given in Table 9. 2. The percentage power increase is also given for each simulation at different wind speeds. As can be seen, around 1% power output increase is achieved at different below rated wind speeds, using the power set-points obtained at $10ms^{-1}$mean wind speed.

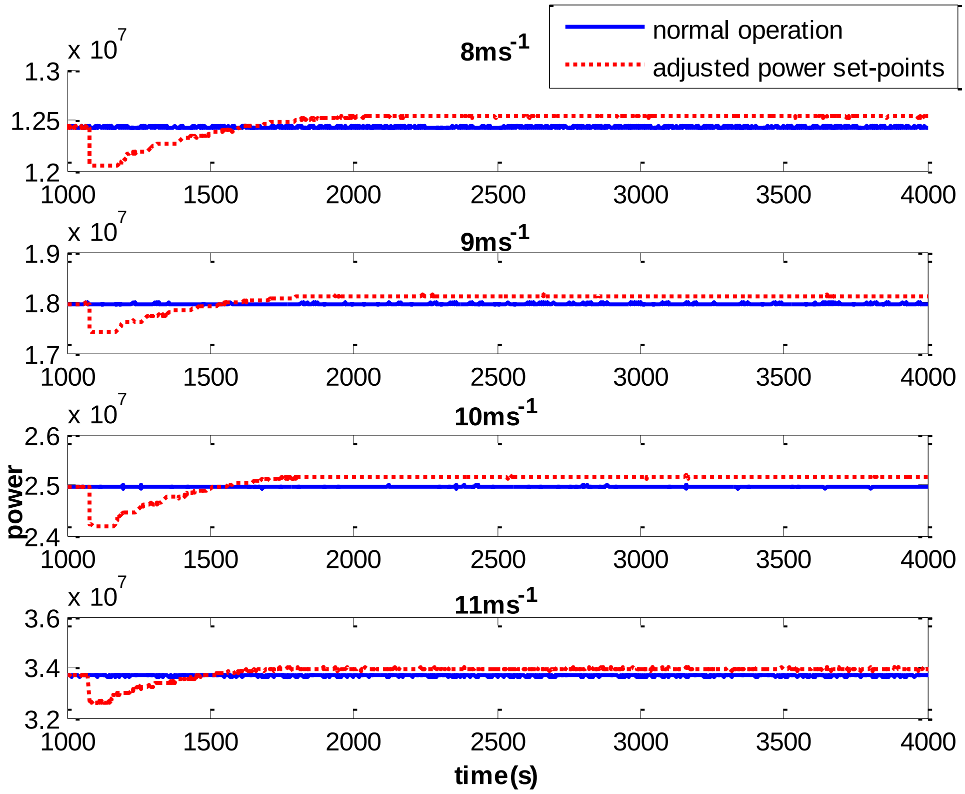


Figure 9-4: Total power for different wind speeds for maximization control

	Total mean before	*Total mean after*	*% power increase*
$8ms^{-1}$	12.43MW	12.531MW	0.8128%
$9ms^{-1}$	17.975MW	18.124MW	0.8278%
$10ms^{-1}$	24.96MW	25.166MW	0.8244%
$11ms^{-1}$	33.663MW	33.93MW	0.7928%

Table 9. 2: percentage power increase after adjusting power set-points through Heat & Flux concept

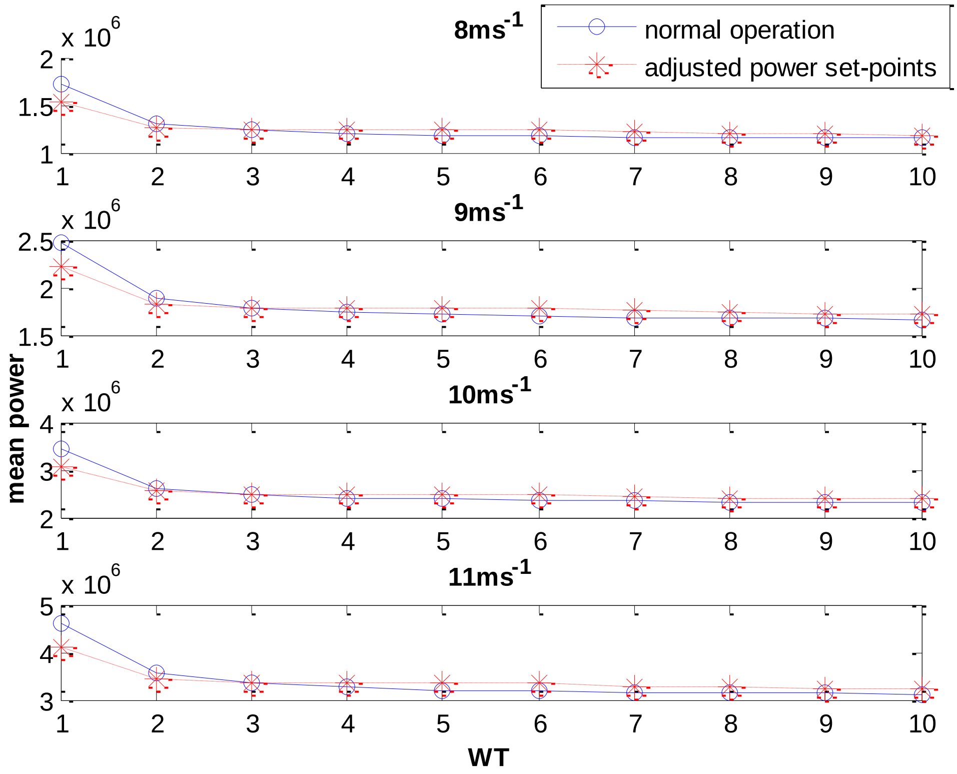


Figure 9-5: Mean power for different wind speeds for maximisation control

To further evaluate the strategy, more simulations are conducted at different below rated mean wind speeds with 10% turbulent intensity.

Simulation results in Figure 9-6 and Figure 9-7 suggest that, the power set-point pattern obtained from the Heat & Flux method at constant wind speeds, show a little improvement in the total power output of the wind turbines for turbulent wind speeds. Table 9. 3 contains the percentages of the power output changes due to the power set-points adjustments at different turbulent wind speeds. The poor performance at 11m/s arises from turbines on occasion going above rated. Accordingly, the power maximisation strategy should only be utilised in lower wind speed

	Total mean before	*Total mean after*	*% power increase*
$8ms^{-1}$	16.716MW	16.79MW	0.4434%
$9ms^{-1}$	24.111MW	24.238MW	0.5255%
$10ms^{-1}$	32.957MW	33.021MW	0.1942%

$11ms^{-1}$	39.768MW	39.606MW	-0.4081%

Table 9. 3: Effect of power set-point adjustments on the total power output

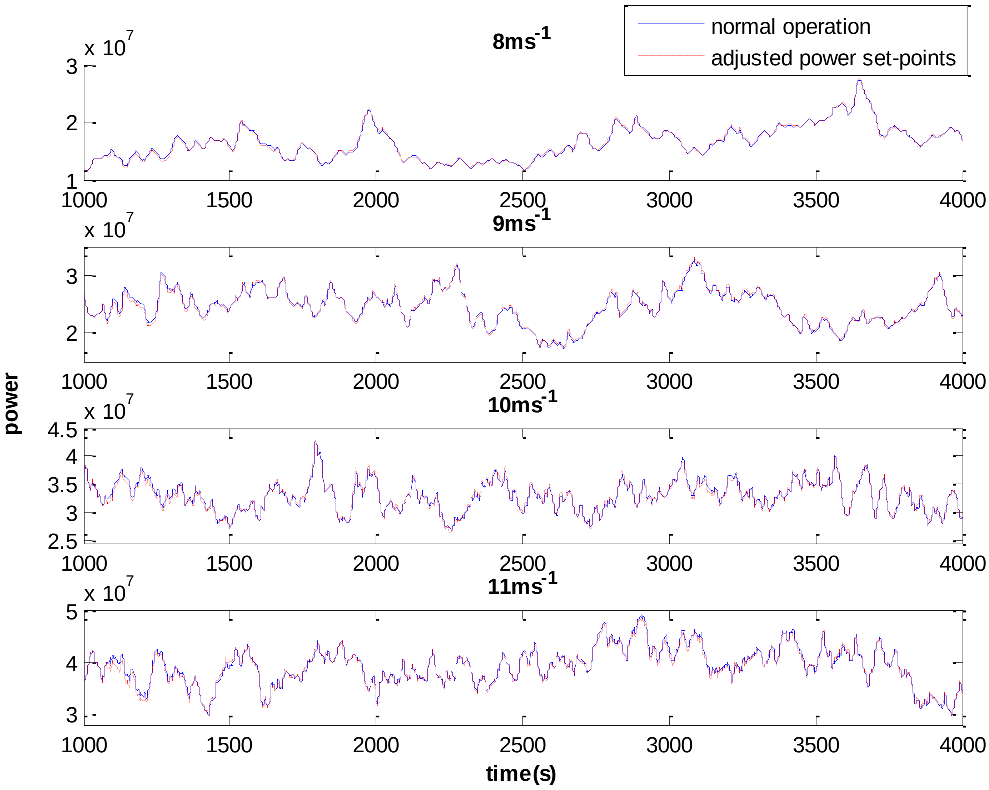


Figure 9-6: Total power output before and after power set-point adjustments for turbulent wind

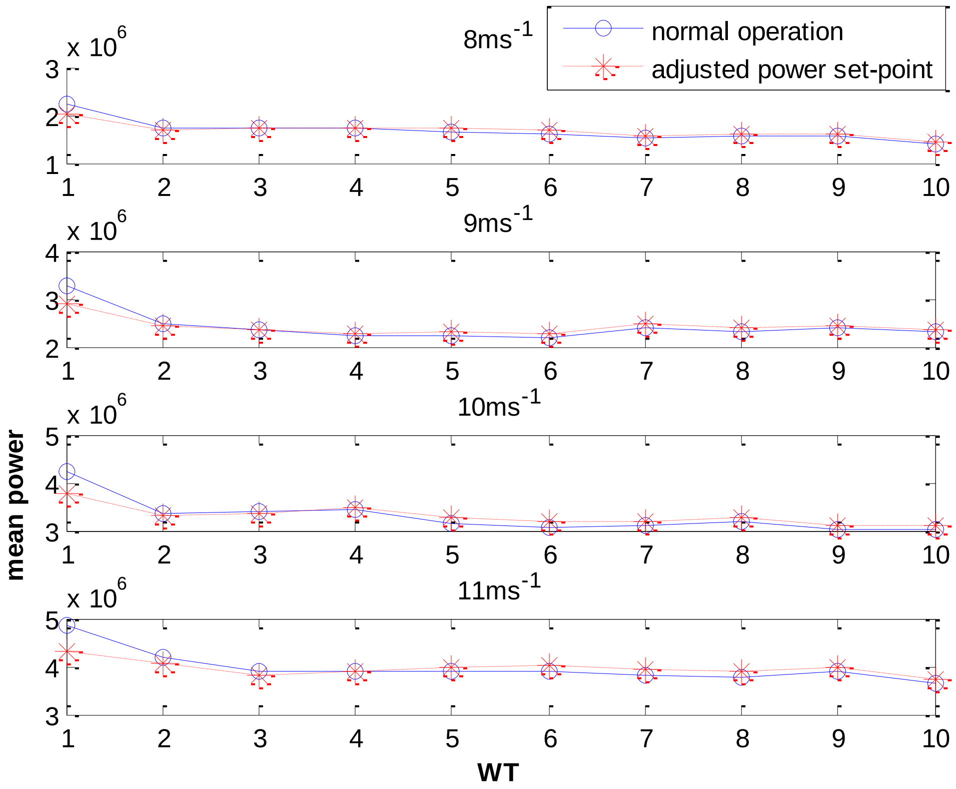


Figure 9-7: Mean power outputs before and after power set-point adjustments for turbulent wind

The main drawback of the power maximisation control using the Heat and Flux concept is the potential of losing power instead of gaining extra power in the farm level. In a wind farm layout depicted in Figure 9-1, downstream wind turbines may not be affected by the upstream wind turbines' wake during their operation. Considering situations where the wind direction is not parallel to the wind turbines array or where the wake of upstream wind turbines meanders around the downstream wind turbines. In these conditions, downstream wind turbines may not always remain in the wake of the upstream wind turbines. Therefore, de-loading operation of the upstream wind turbines may not have a significant effect on the wake condition at downstream wind turbines.

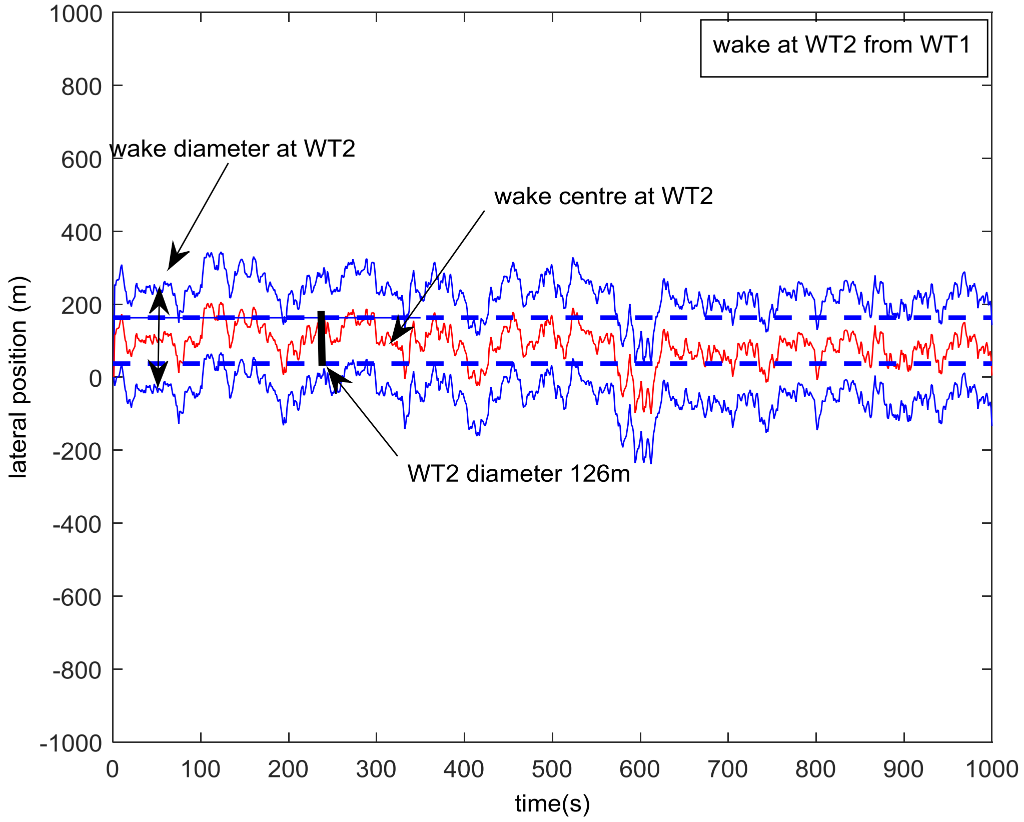


Figure 9-8: Wake centre and diameter at a downstream wind turbine affected from an upstream wind turbine

Figure 9-8, illustrates a situation where WT2 is in the wake of WT1 in this simulation. The twin blue lines depicted in the graph represent the wake diameter of WT1 at the position of WT2. The centre red line represents the wake centre at position of WT2. The thick vertical black line shows the rotor diameter and lateral position of WT2 in the wind farm. As can be seen WT2 is in the full wake of WT1 for most of the simulation time. However, since the wake of WT1 meanders at WT2 position, in some simulation times between 600 seconds to 700 seconds WT2 is only affected by the partial wake of WT1. This is an example where de-loading operation of WT1 may not effectively increase the wind power at WT2.

Thus, derating power set-point of upstream turbines, in conditions where the wake of upstream turbines does not affect the downstream turbines, could adversely cause the total wind farm power output to reduce. Hence, to implement an effective power maximisation control algorithm, wind turbines' operation must be coordinated in a more intelligent manner using estimation of the wind-field condition at each turbine. The power maximisation control will be investigated more closely in the future work.

9.2 Wind farm optimisation with balanced power generation

In this approach the aim is to find a general pattern for wind turbines' power set-point adjustment in order to produce almost the same power at each individual turbine in the farm. The advantage of such an approach is that, when the wakes are not interacting, operating all the turbines to produce the similar power would still be an appropriate strategy. Since wind turbines experience different wind speed due to the wake effects, by adjusting the power set-point of the upwind turbines, the wake effect in the downwind turbines can be altered to generate a similar level of power at all turbines in the farm. To obtain a set of optimal power set-points for this control approach a quasi-static optimisation problem is solved using Matlab toolbox.

9.2.1 *Optimal power set-point pattern using a quasi-static wind farm model*

Here, a quasi-static optimisation problem for an array of 10 turbines, positioned in a line behind each other in the direction parallel to the wind speed with 800 meter distance between the turbines, is defined to estimate a set of power set-points, at steady state operating condition, so as to maximise the total power output.

The optimisation problem is scripted in Matlab and solved with the Matlab optimisation toolbox. The optimisation objective is to minimise the difference between the average of total power output of the wind turbines and the individual wind turbines' power output. The outcome of this optimisation is a set of C_T values that are used to compute an estimate of the power set-points for individual wind turbines. The computed power set-points are then used in online simulations to coordinate the wind turbines operation and evaluate the optimisation problem approach.

The optimisation goal is to produce the same level of power at all wind turbines in the farm. Since the power capture at all turbines is expected to be almost the same, it can be anticipated that the wind turbines may experience similar levels of aerodynamic loading on their structure. Although this control strategy may cause the total power output of the wind farm to slightly decrease, in theory it is a simple and effective control strategy to reduce the aerodynamic loading on wind turbines.

The optimisation function is defined using a combination of a power model, wind speed model, and wake deficit model for each wind turbine as

$$
\begin{gathered}
P_{i+1} = \frac{1}{2}\rho A v_{i+1}^3 C_{P\,i+1} \\
v_{i+1} = U_0 \cdot deficit_{i+1} \\
deficit_{i+1} = 1 - \left[\frac{WD_i^2}{WD_{i+1}^2}(1 - deficit_i) + \frac{1}{2}\frac{CT_i \cdot D_0^2}{WD_{i+1}} deficit_i\right] \qquad (9\text{-}1) \\
WD_{i+1} = \left[\beta_{i+1}^{k/2} + \alpha \frac{dist}{D_0}\right]^{1/k} D_0, \quad k = 2, \alpha = 0.5 \\
\beta_{i+1} = \frac{1 + \sqrt{1 - CT_i}}{2 \cdot \sqrt{1 - CT_i}}
\end{gathered}
$$

where, $P_{i+1}, v_{i+1}, deficit_{i+1}, WD_{i+1}$ represent the power, wind speed, wind speed deficit, and wake diameter at wind turbine $i + 1$ respectively.

The optimisation algorithm utilises Matlab *fmincon* function that searches for a number of variables defined in a local function to minimise the objective function within the local function. The local function computes the deficits, wind speeds and output powers in each iteration using the estimated C_T values. The objective function is defined within the local function by

$$
\min f(C_T) = \sum_{i=1}^{N} \left(P_{avg} - P_i\left(C_{T_i}\right)\right)^2 \qquad (9\text{-}2)
$$

$$
Subj. \qquad \begin{cases} 0 \leq C_{T_i} \leq 0.8889 \\ P_i \leq rated \end{cases}
$$

where, P_{avg} is the average power of N wind turbines, p_i is the power output of each wind turbine with subscribe index i, and CT_i is the thrust coefficient of the $i - th$ turbine.

The $fmincon$ function yields a set of C_T coefficients that minimises the above constrained multivariable objective function.

9.2.2 *Power adjustment pattern*

The optimisation problem is solved at different mean wind speeds to identify a common power set-point adjustment pattern for the whole operating envelope

Figure 9-9 illustrates the power coefficients obtained from the optimisation problem at $10ms^{-1}$. The C_P coefficients are computed from C_T values by

$$C_P = C_T\left(\frac{1+\sqrt{1-C_T}}{2}\right) \tag{9-3}$$

Figure 9-10 and Figure 9-11 show the wind speeds and wake deficits corresponding to these power coefficients. As expected, the mean wind speed at individual turbines is decreasing towards the last turbine in the line. As can be seen from Figure 9-12, the derived C_T values force the power output of each individual turbine to be equal to the total mean power output of the farm as defined in the objective function. At each optimisation iteration the total mean power output is changing and therefore the optimisation function yields a new set of optimisation variables (C_T) to satisfy the objective function.

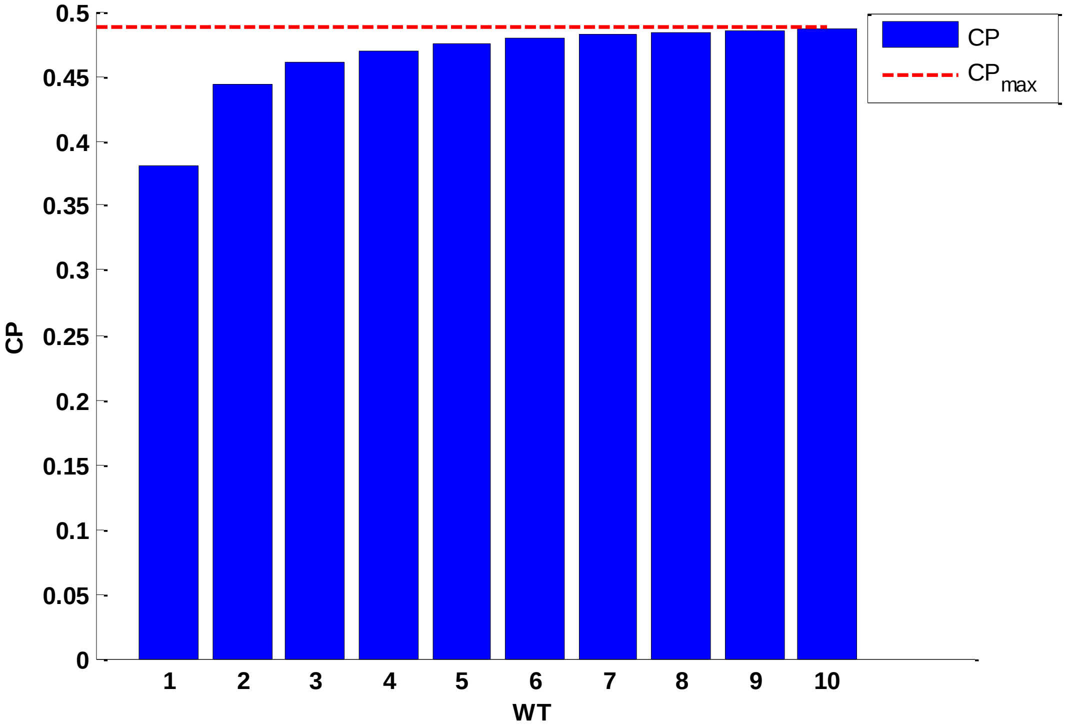


Figure 9-9: Power coefficients obtained from the static optimisation

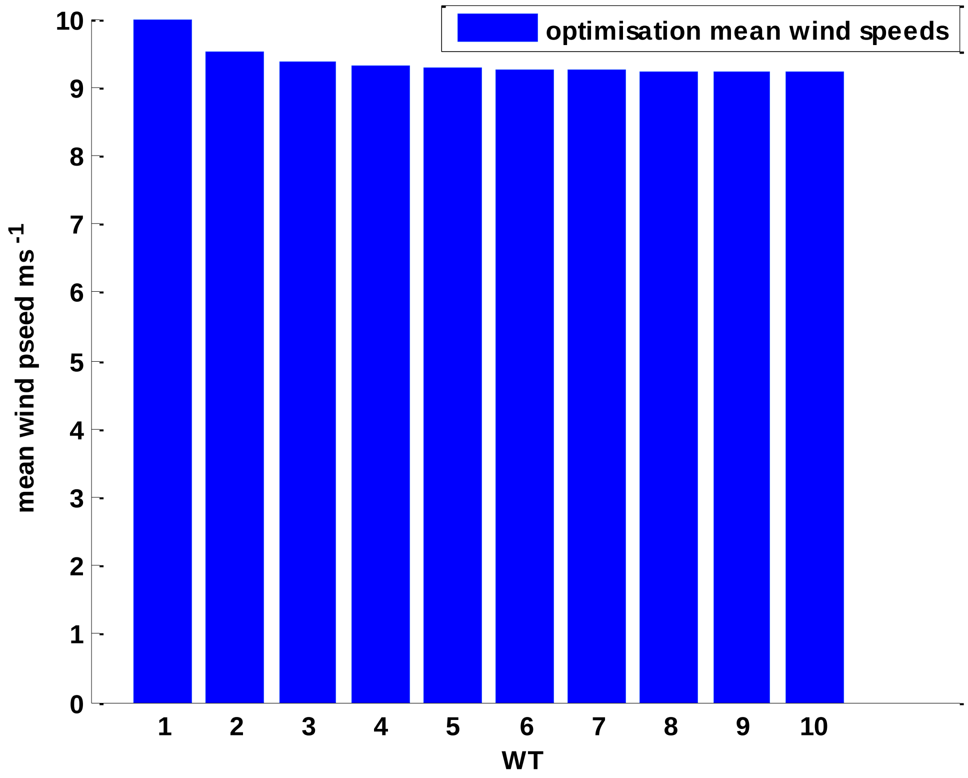


Figure 9-10: Wind speeds of the turbines in the static optimisation

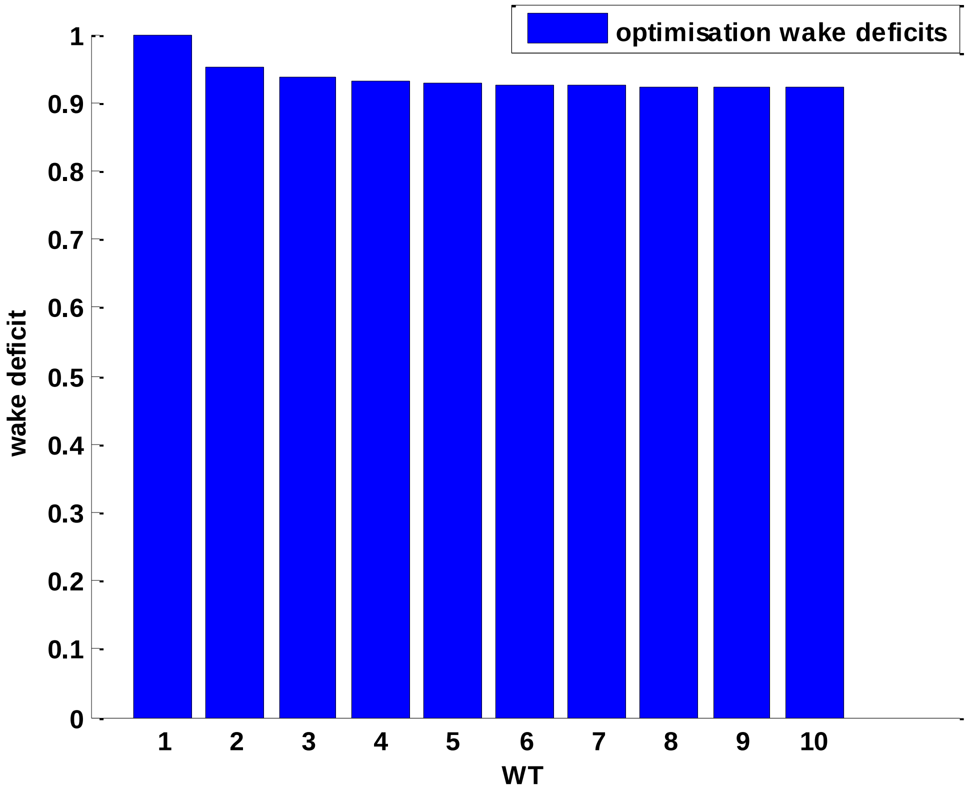


Figure 9-11: Wake deficits of the turbines in the static optimisation

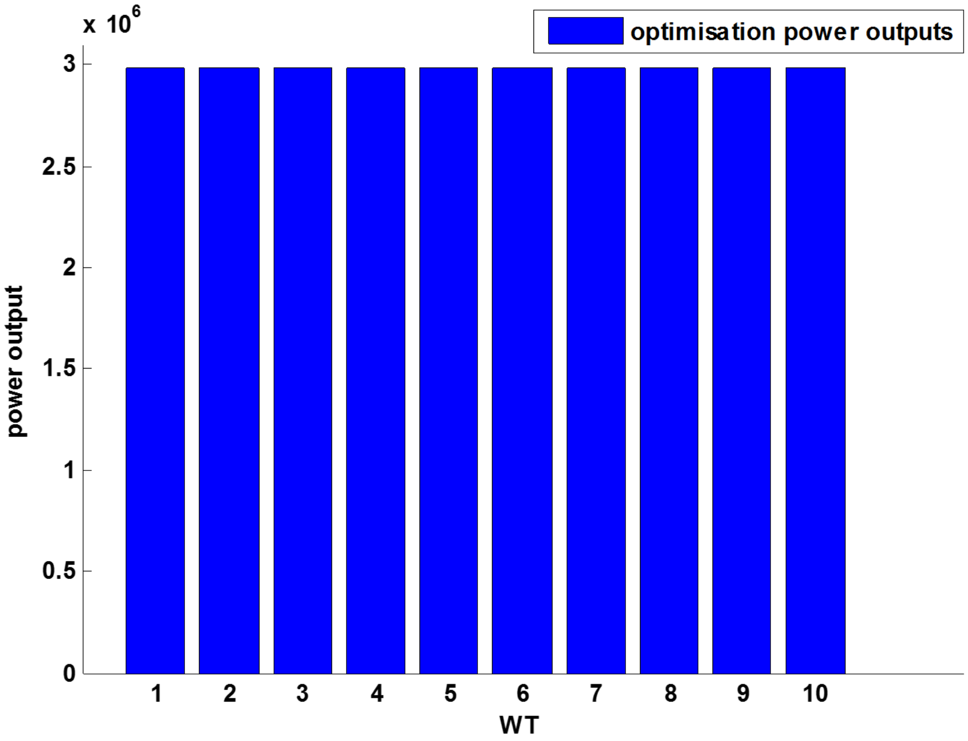


Figure 9-12: Power output of wind turbines in the static optimisation

By comparing the estimated power adjustment levels, obtained from the optimisation problem at different mean wind speeds, a common pattern is observed.

Figure 9-13 illustrates the C_P values for individual wind turbines obtained at different below and above rated wind speeds. For below rated wind speeds a consistent pattern is evident. This pattern is used to define a control strategy for below rated wind speeds.

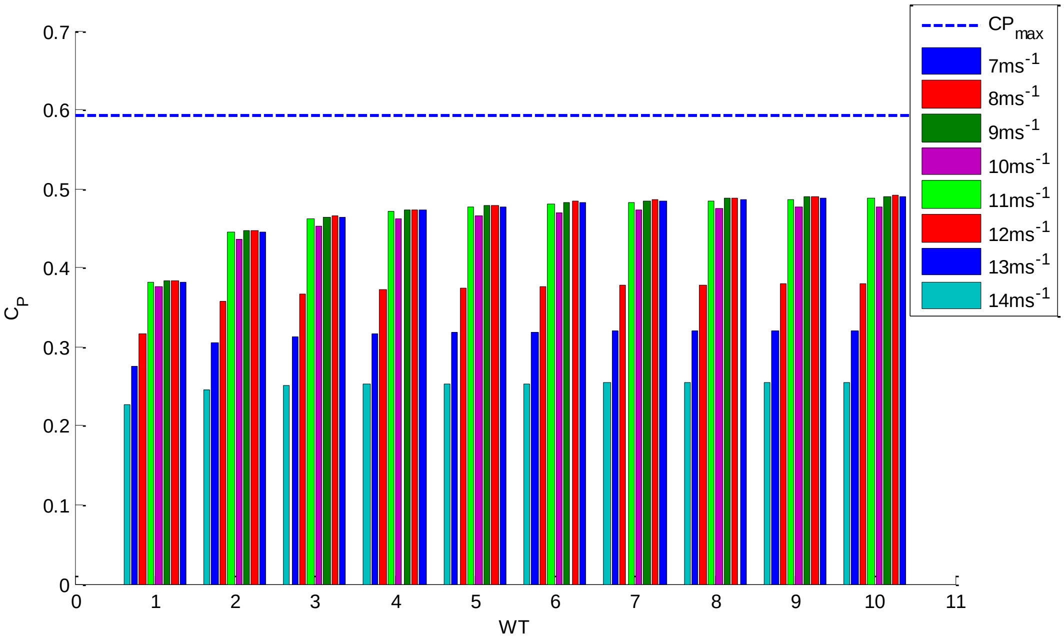


Figure 9-13: The required Cp adjustment levels in this optimisation at different wind speeds

The adjustment power level for each wind turbine is computed using the derived C_P values as a percentage of the optimal power by

$$\%\ adjustment\ level = \frac{(C_{pi} - C_{pmax})}{C_{Pmax}} * 100 \tag{9-4}$$

Figure 9-14 illustrates the percentage drop required in the power output of each wind turbine to make the output power of all wind turbines equal to the mean power output of the wind farm at different wind speeds. Since the optimisation problem is mostly applicable at below rated wind speeds, only below rated adjustment power levels are computed. In above rated operating conditions, wind farm controller switches to the reference power tracking control mode.

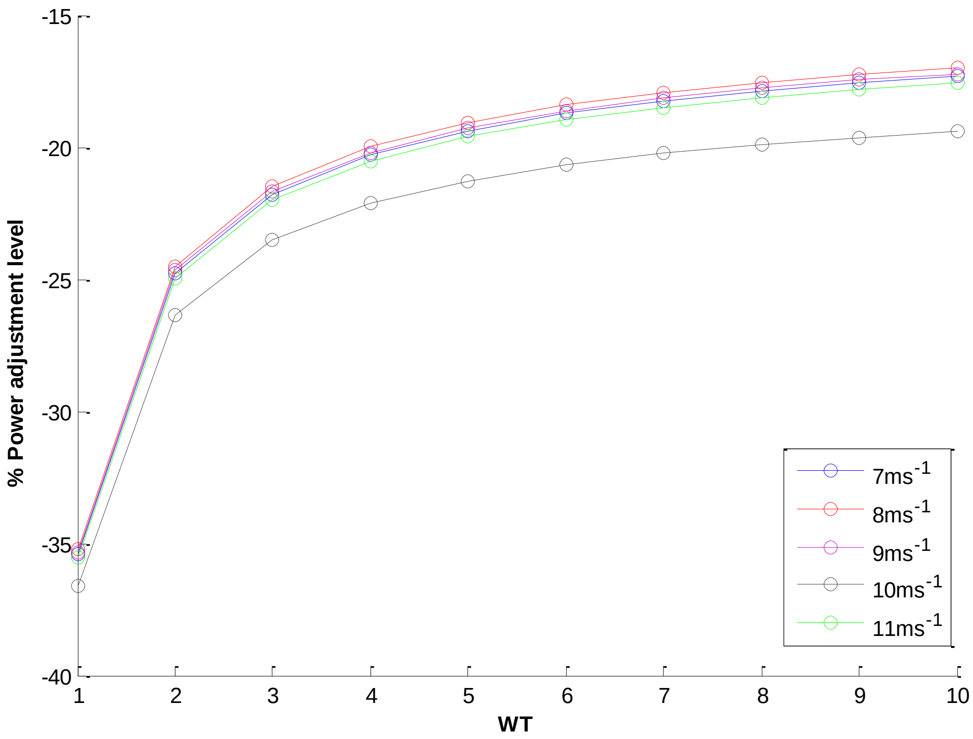


Figure 9-14: Wind turbine power adjustment patterns at different wind speeds obtained from the static optimisation

9.2.3 *Simulation results*

In order to evaluate the optimisation results, a number of simulations are conducted using the power set-points obtained from the static optimisation problem. Simulations are conducted at $10ms^{-1}$ mean wind speed with and without turbulence.

The power adjustment pattern obtained from the static optimisation is given in **Table 7. 1**.

WT1	WT2	WT3	WT4	WT5	WT6	WT7	WT8	WT9	WT10
-36%	-26%	-23%	-22%	-21%	-20%	-20%	-20%	-19%	-19%

Table 7. 1: Power set-point adjustment pattern

Figure 9-15 shows the mean power output of the wind turbines before and after power set-point adjustments in this optimisation control approach.

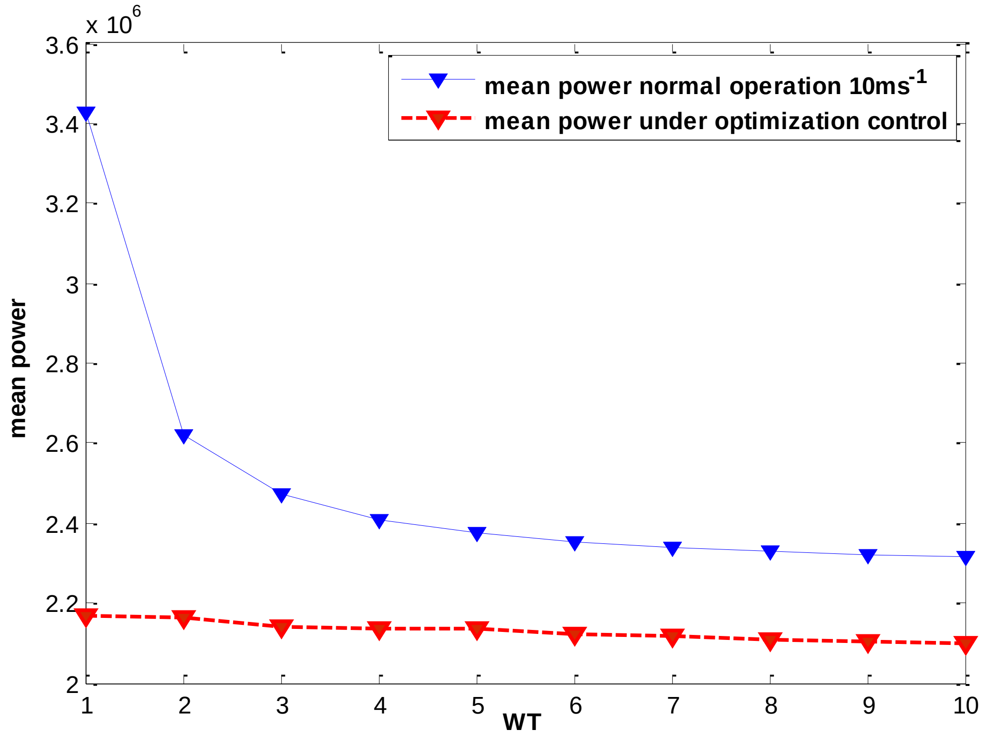


Figure 9-15: Mean power before and after optimisation

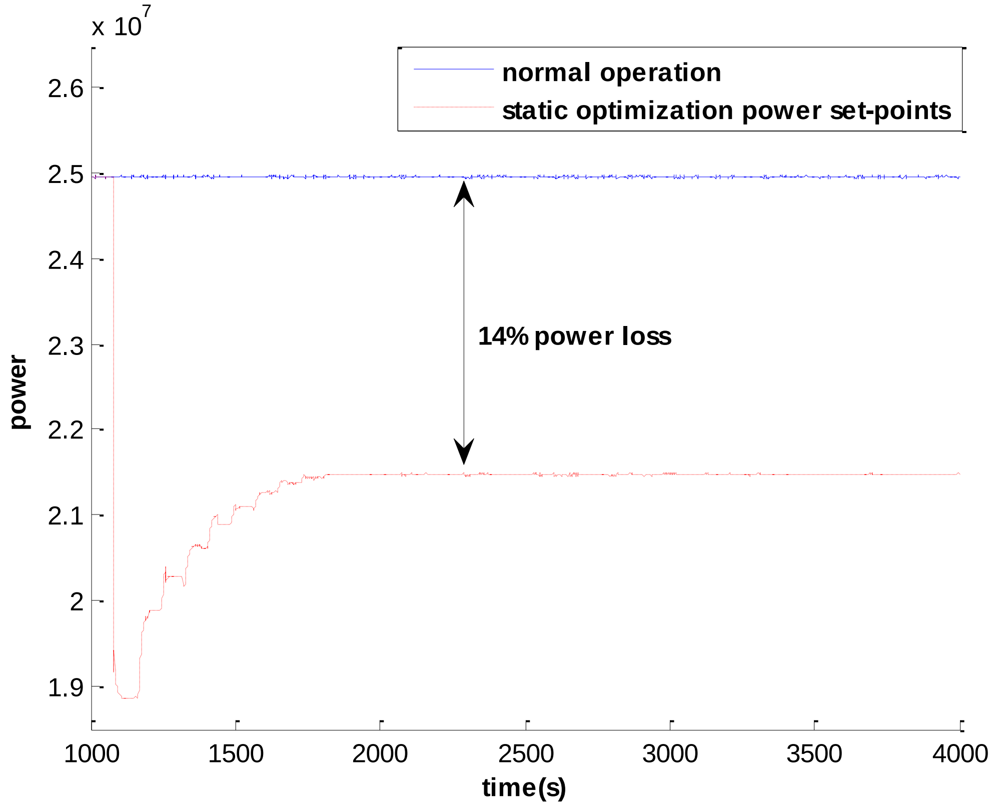


Figure 9-16: Total power output before and after optimisation

As can be seen from Figure 9-15 and Figure 9-16, this optimisation control approach resulted in almost equal power output production at all turbines. Although the total power output is reduced by almost 14%. The power set-points, obtained from the static optimisation problem, are obtained by minimising the difference between the total mean power output of the wind turbines and the individual wind turbines' power. Therefore, to reduce the power losses due to this optimisation objective, the objective function in the static optimisation is modified to achieve a new set of power set-points, with the aim to increase the total power output of the wind farm, whilst keeping the individual power output of the turbines almost equal. This modification is done by replacing the average power in the equation (9-2) by

$$Pav = Pav + 1e4. \tag{9-5}$$

The corresponding simulation results using the new set of the power set-points, obtained from the adjusted optimisation, suggest that by increasing the average power in the optimisation function the total power output of the wind farm can be increased further, whilst the power output across all wind turbines is kept equal.

Figure 9-17 illustrate the mean power output of individual wind turbines, using different set of power set-points obtained from the optimisation for different level of average power in the optimisation function.

Similarly as shown in Figure 9-17 and Figure 9-18, for a $0.1MW$ increase in the total mean power in the optimisation function, the total power output is increased significantly.

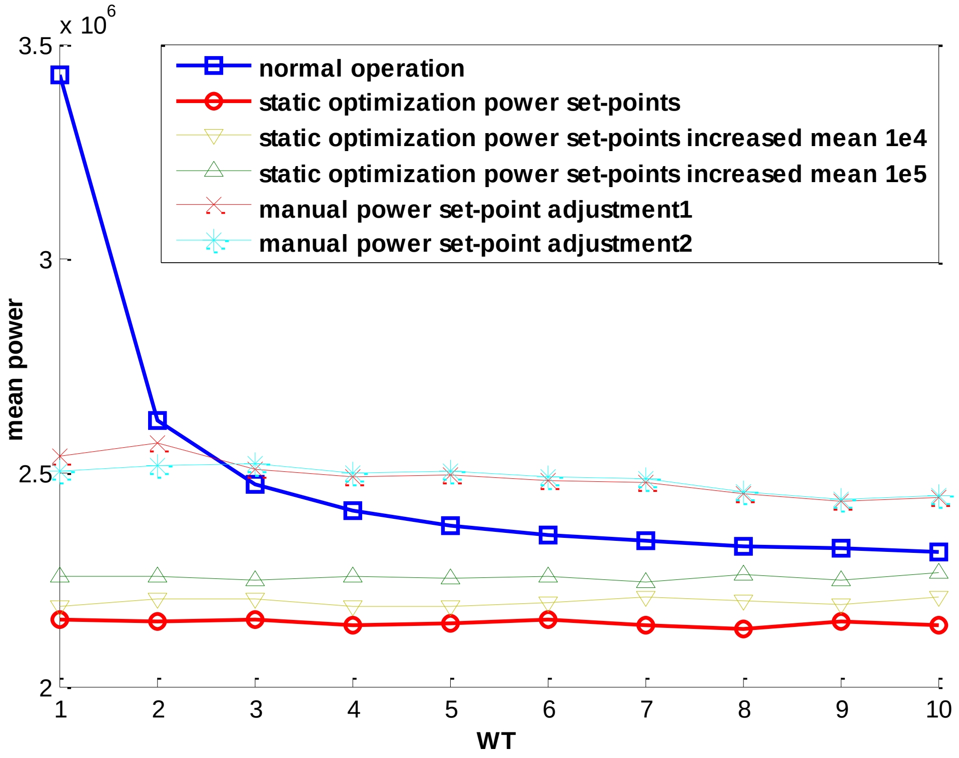


Figure 9-17: Mean power output of the turbines after power set-point adjustments

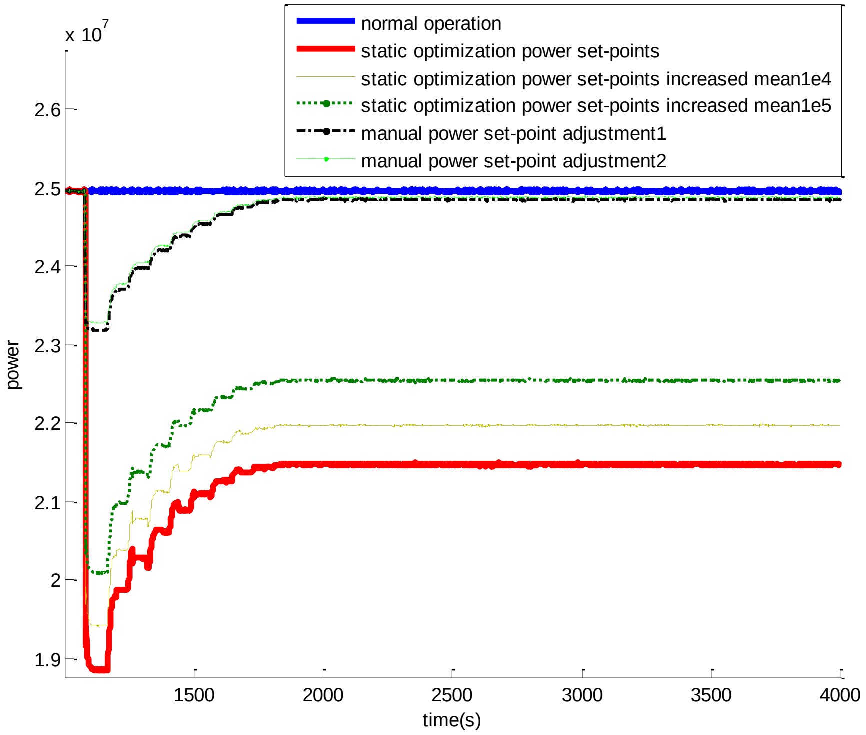


Figure 9-18: Total power output for different power set-points' adjustment

The power adjustment levels for all optimisation solutions are given in Table 9. 4. Row 1 represents the power set-points obtained from the initial static optimisation, Row 2 corresponds to the 0.01MW adjustment in the averaged power in the optimisation function, Row 3 represents the power set-points obtained by 0.1MW adjustment in the averaged power, and finally Row 4 and Row 5 correspond to the sequential adjustment, following a similar procedure to that of Section 9.1, of the power set-point obtained from the static optimisation problem.

The total mean power output of the turbines and the percentage of the power loss for each optimisation solution is given in the Table 9. 5.

	WT1	*WT2*	*WT3*	*WT4*	*WT5*	*WT6*	*WT7*	*WT8*	*WT9*	*WT10*
1	-36%	-26%	-23%	-22%	-21%	-20%	-20%	-20%	-19%	-19%
2	-35%	-24%	-21%	-20%	-19%	-18%	-17%	-17%	-17%	-16%
3	-33%	-22%	-19%	-17%	-16%	-15%	-15%	-14%	-14%	-13%
4	-25%	-10%	-7%	-5%	-3%	-2%	-1%	-1%	-1%	0%
5	-26%	-12%	-7%	-5%	-3%	-2%	-1%	-1%	-1%	0%

Table 9. 4: Power set-points set obtained from the optimisation

	Total mean	*% power loss*
1	21.5MW	13.9%
2	22MW	11.9%
3	22.5MW	9.7%
4	24.88MW	0.35%
5	24.85MW	0.45%

Table 9. 5: Total mean power output and power loss for each optimisation power set-points set

As can be seen, the most upstream turbines in the line are most significantly affected. The sequential adjustment of the power set-points is made to reduce the losses due to this optimisation. From Figure 9.17, it can be seen that the sequential adjustment strategies result in a slow decrease in power from WT1 WT10. It is not possible to reproduce this pattern with a strategy that seeks to have each turbine producing the same power, hence, the poorer performance. Instead, the cost function (9-2) should be modified so that the turbines have decreasing power. Figure 9-19 illustrates the thrust coefficient values with the power set-point adjustment corresponding to strategy in the Row 5 of Table 9. 4 . Clearly, the level of thrust coefficients and therefore the thrust force on the wind turbines are reduced.

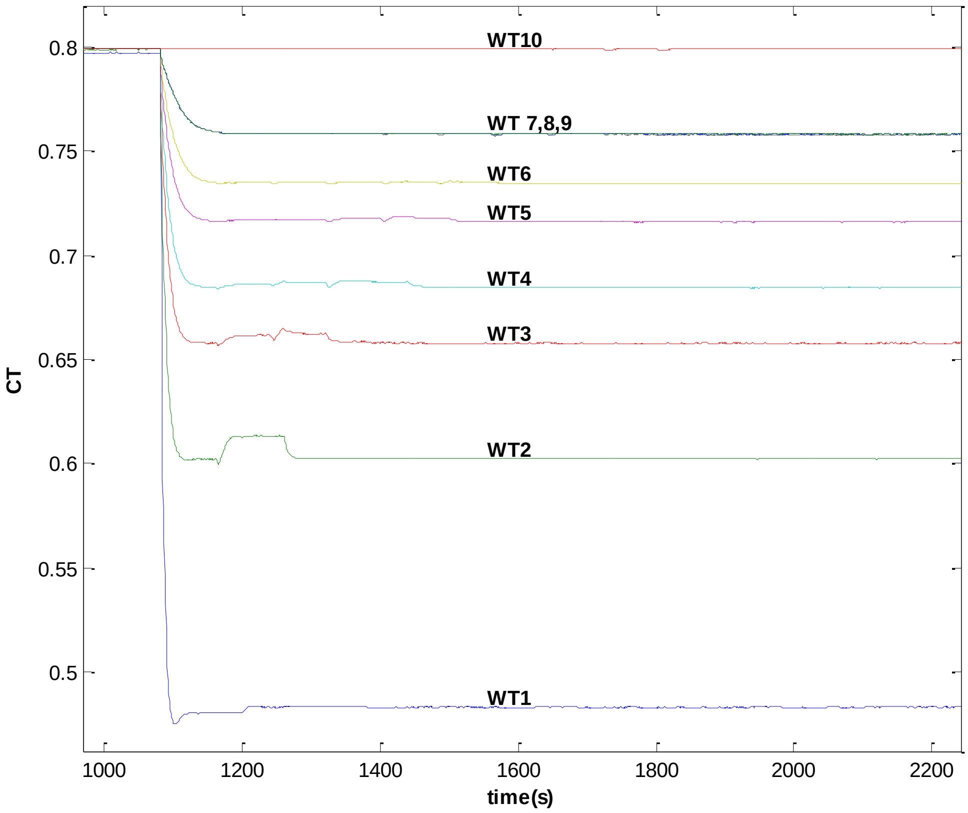


Figure 9-19: CT values after power set-points obtained from the optimisation applied

To validate the optimisation pattern obtained, more simulations at different wind speeds are performed. For each simulation the power set-points in Row 5 of Table 9. 4 is utilised.

Figure 9-20 and Figure 9-21 illustrate the total power output of the wind turbines and the mean power output of individual wind turbines before and after power set-point adjustments at different wind speeds.

As can be seen the power set-point adjustment pattern obtained from the optimisation at $10ms^{-1}$ wind speed is applicable for a range of below rated wind speeds. Therefore, a general pattern for optimising the wind farm power at different below rated wind speed for this wind farm model is achieved.

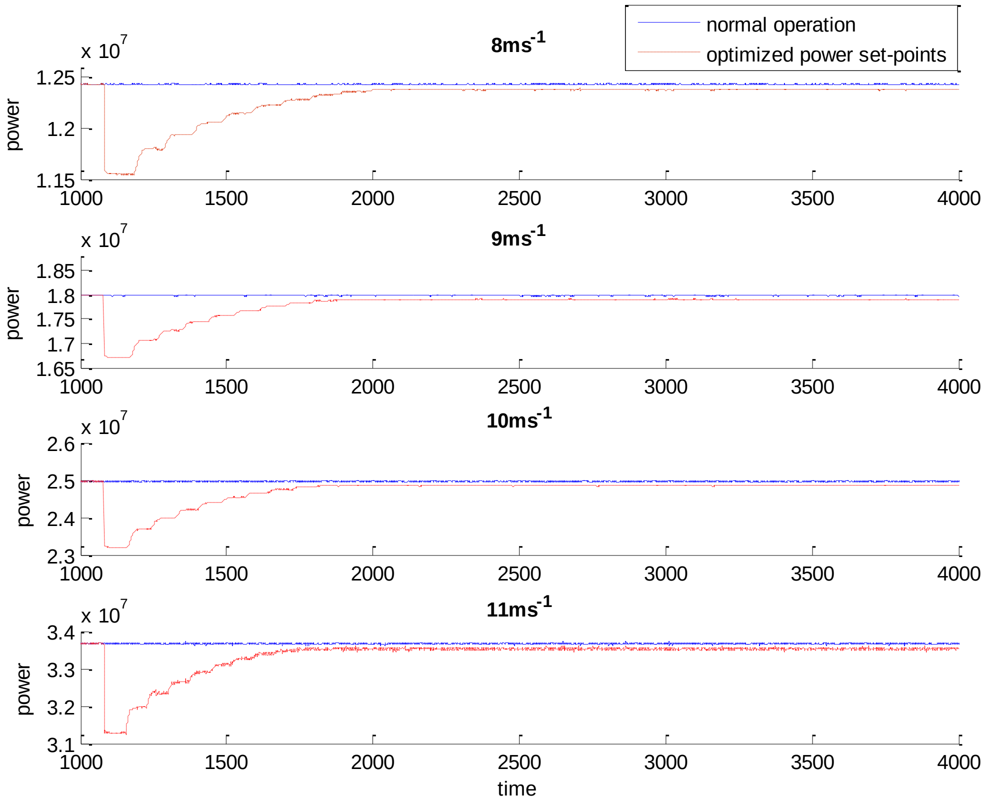


Figure 9-20: Total power output at different wind speeds before and after power set-point adjustments

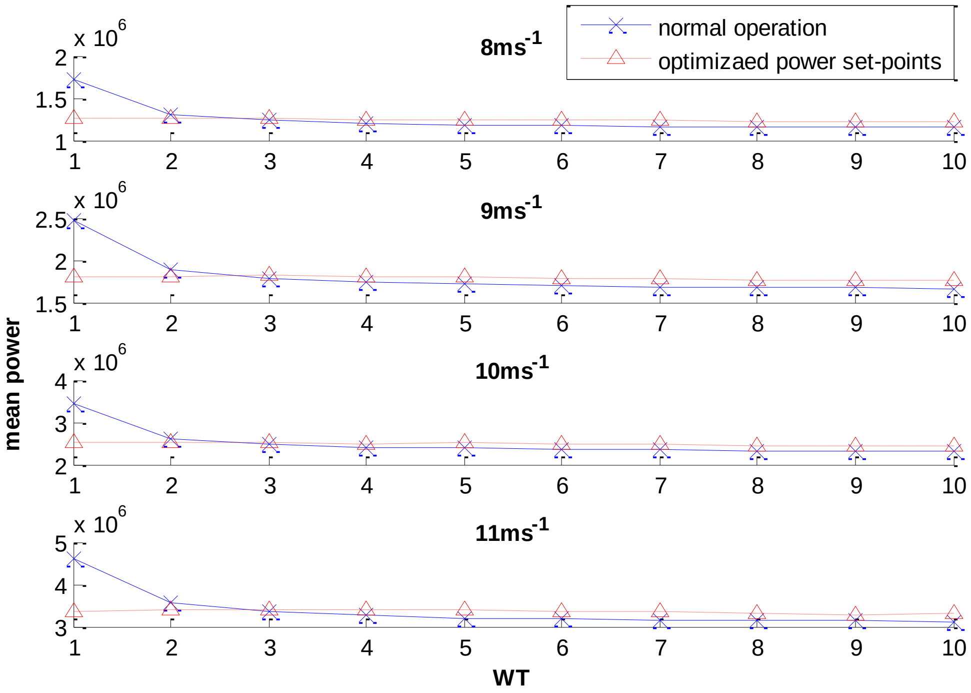


Figure 9-21: Mean power output of individual turbines at different wind speeds before and after power set-points' adjustment

9.2.4 *Load analysis*

In this simulation the equal power generation control strategy in a wind farm with the layout as shown in Figure 9-22 is investigated. The objective is to examine the effect of this control strategy on the wind turbines aerodynamic loads. The simulation is conducted at $8ms^{-1}$ mean wind speed with 10% turbulent intensity.

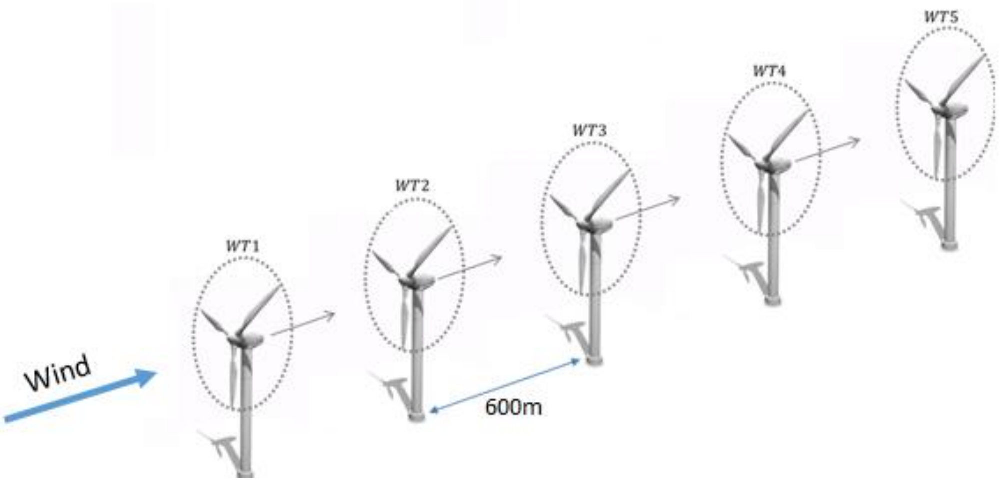


Figure 9-22: Wind farm layout

Initially, only the power set-point of the most upwind turbine (WT1) is reduced by 20% to examine the aerodynamic loads on WT1 and the turbines in the wake. The effect of this power set-point adjustment can be seen in the other turbines power generation in Figure 9-23 and Figure 9-24. As a result of this power adjustment the wind power at downwind turbines is increased. Therefore, the power output of the turbines are increased.

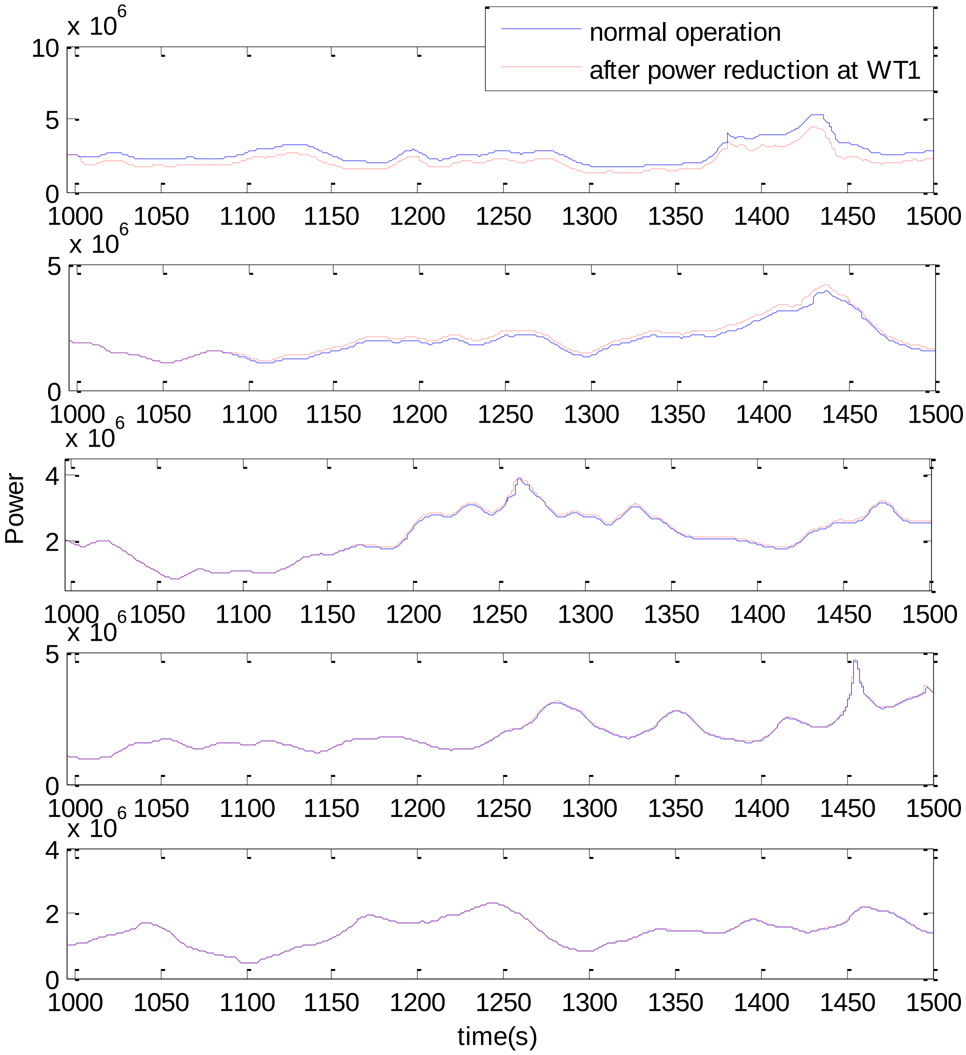


Figure 9-23: Effect of power set-point adjustment at WT1 on the turbines in the wake

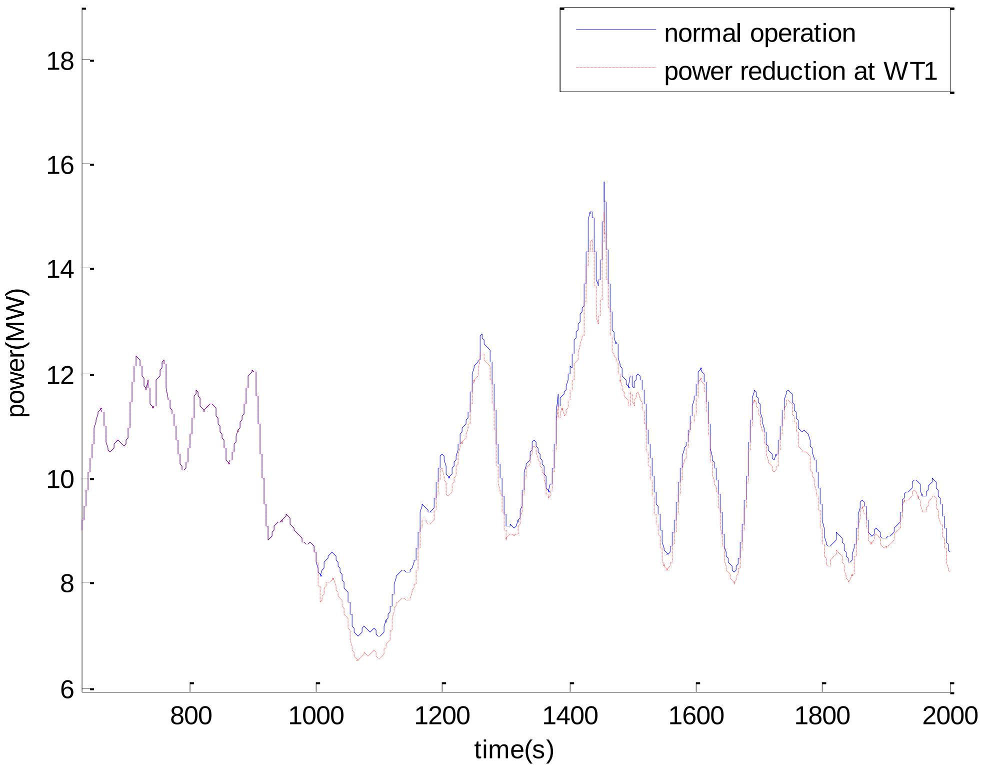


Figure 9-24: Total power output before and after power adjustment at WT1

The effect of this power adjustment on the wind turbines load is investigated by examining the thrust and the tower fore-aft acceleration time series at each turbine. Figure 9-25 and Figure 9-26 illustrate the mean of the power output and thrust force at each wind turbine before and after this power set-point adjustment.

As can be seen, the power reduction in WT1 resulted in power and the thrust force increase in the other turbines. The effect of this power set-point adjustment is also evident the cumulative PSD of the fore-aft acceleration and the thrust force in Figure 9-27 and Figure 9-28. The tower fore-aft acceleration at WT1 is reduced whilst the fore-aft acceleration and the thrust force at the downwind turbines is increased. This is due to increase in the wind power in the wake of WT1.

These results suggest that by coordinating the operation of the wind turbines through their power set-points adjustment, the aerodynamic loads can be distributed between the wind turbines in a wind farm.

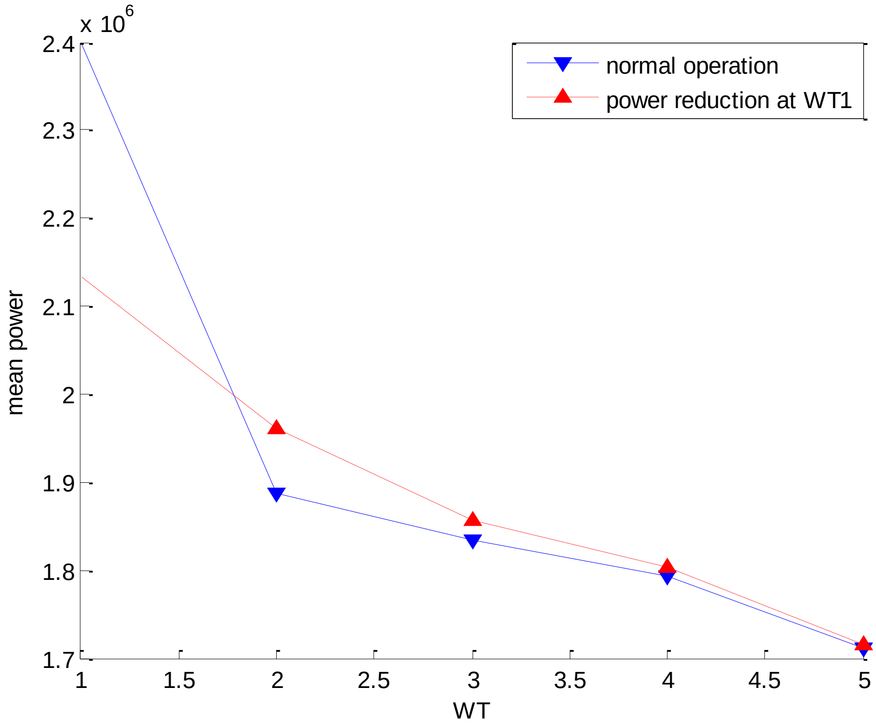


Figure 9-25: Mean power output of the turbines before and after power adjustment at WT1

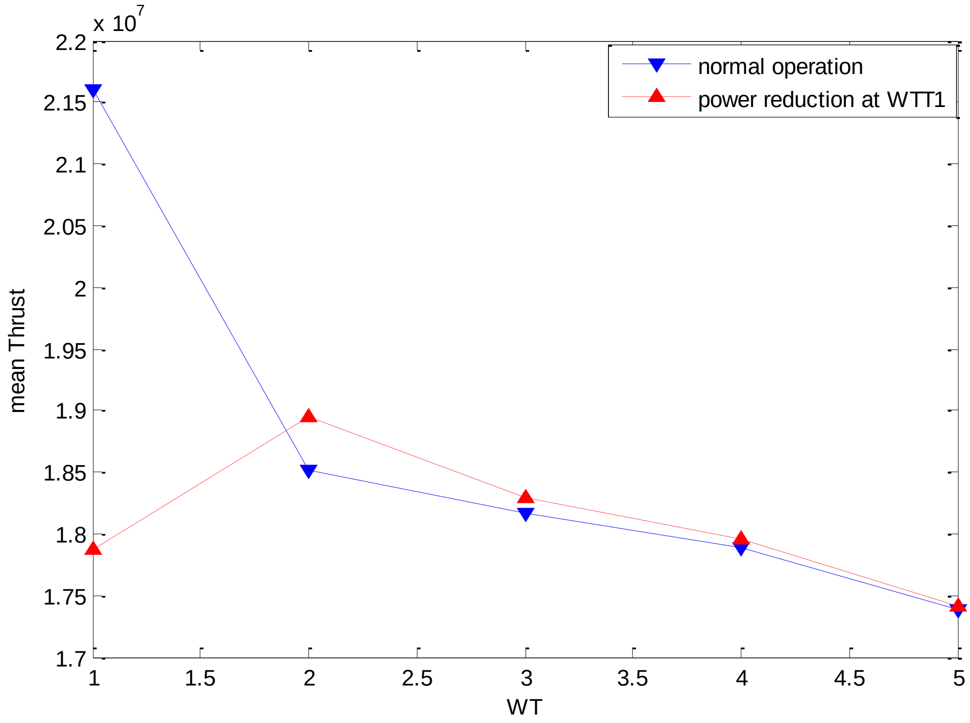


Figure 9-26: Mean of the thrust at turbines before and after power adjustment at WT1

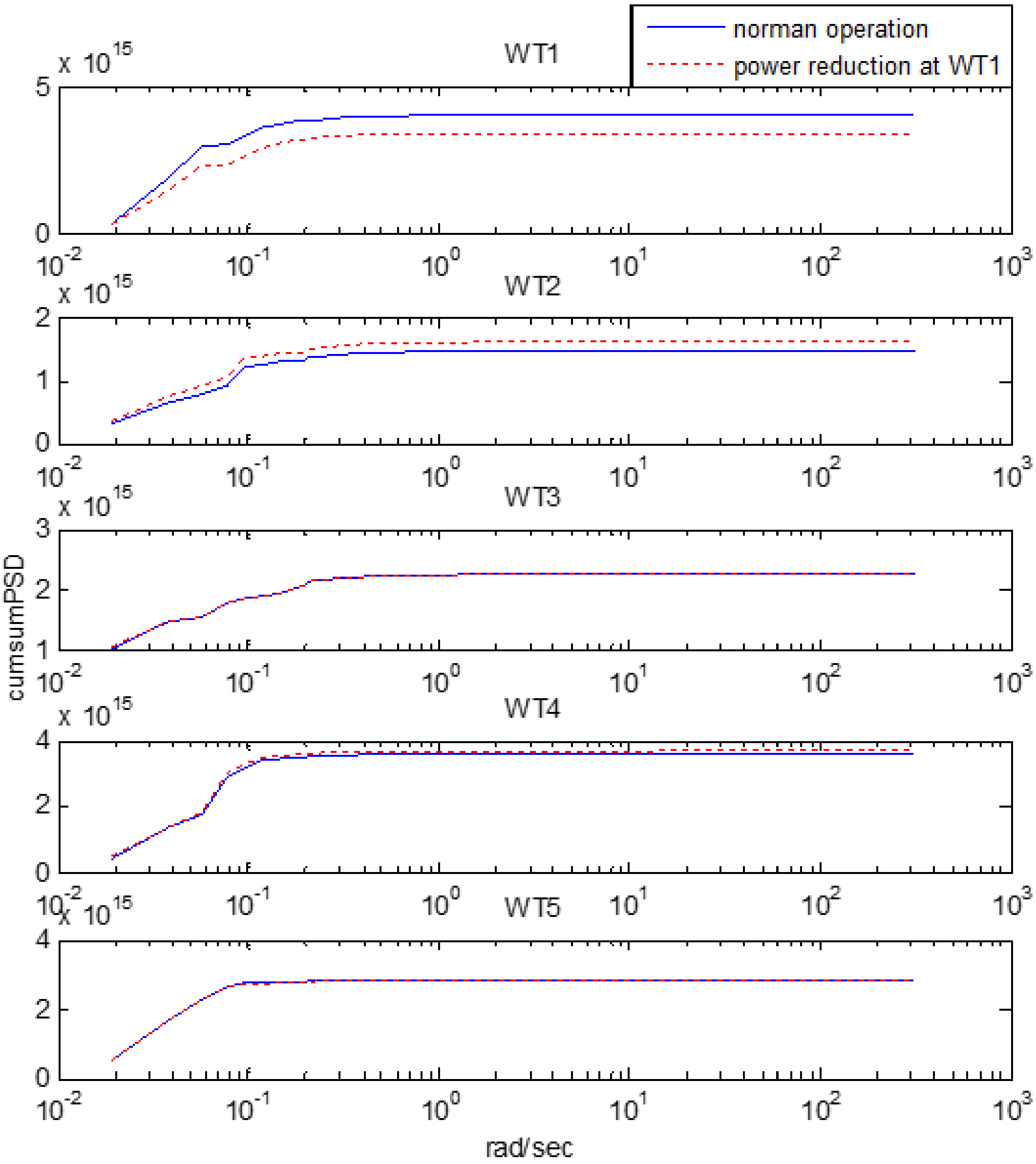


Figure 9-27: Cumulative PSD of the Thrust force at the turbines before and after power adjustment at WT1

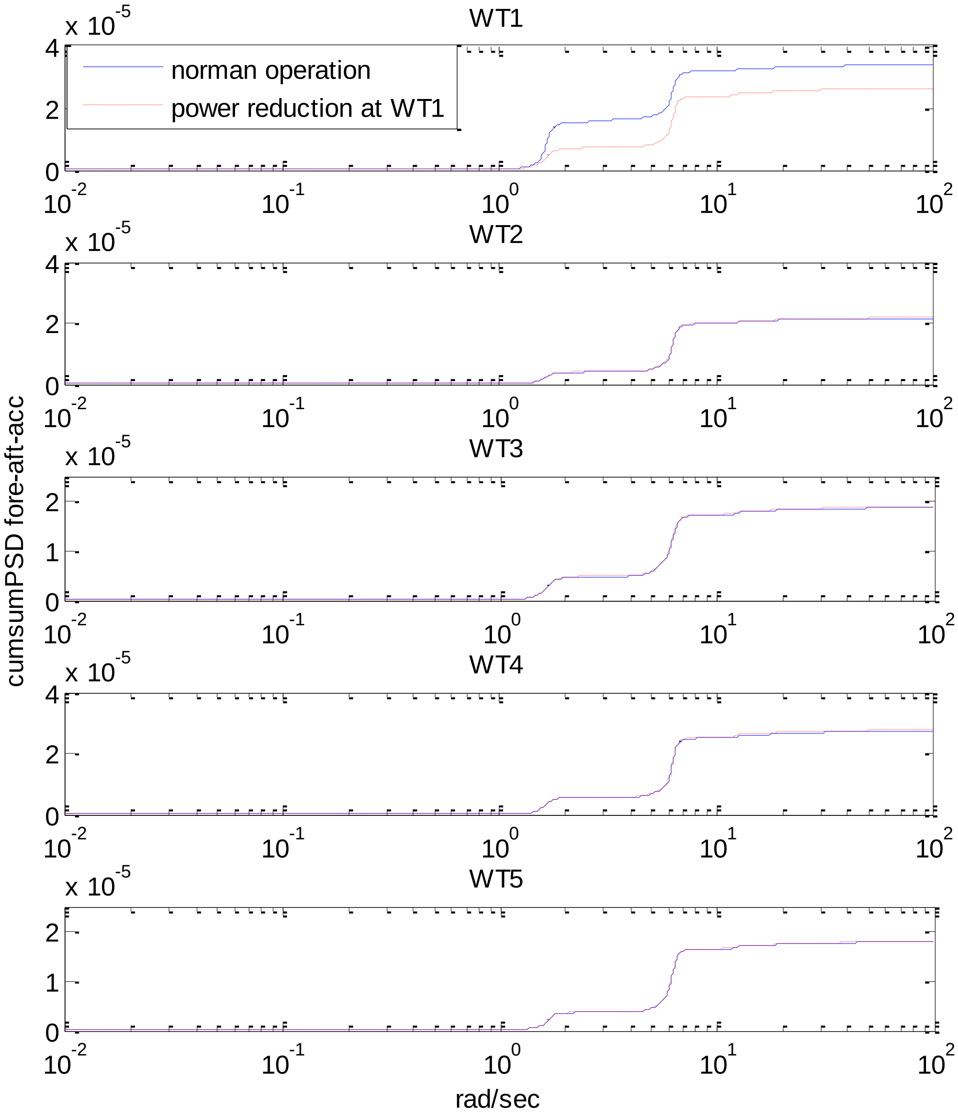


Figure 9-28: Cumulative PSD of the fore-aft acceleration at the turbines before and after power adjustment at WT1

In order to examine the effect of equal power generation control strategy on the loads at the turbines, the power set-points of the WT1, WT2, WT3, and WT4 are adjusted sequentially. The wind turbines' power set-point are adjusted to generate almost the same level of power at each turbine.

The effect of these power set-point adjustments is evident in the total power output and individual power output of the turbines in Figure 9-29 Figure 9-30. The total power loss, as a result of this control strategy, can be reduced by carefully adjusting the power set-points at each turbine in the farm. However, the main objective of this control strategy is to distribute the aerodynamic loads over the turbines.

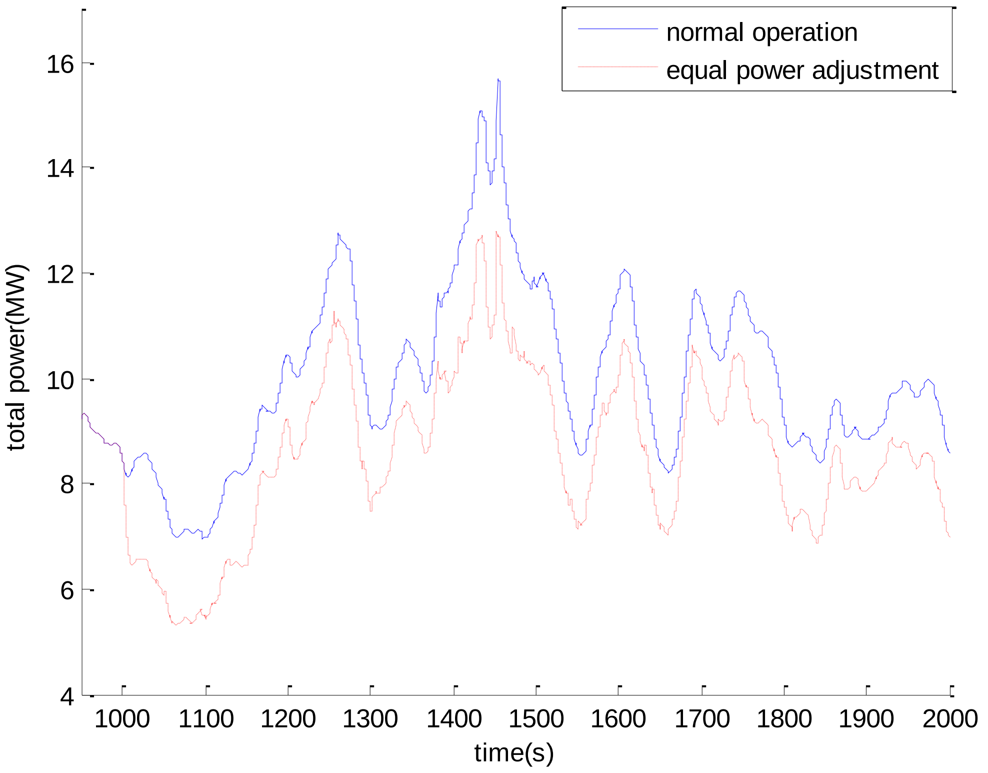


Figure 9-29: Total power output before and after power adjustment at WT1, WT2, WT3, and WT4

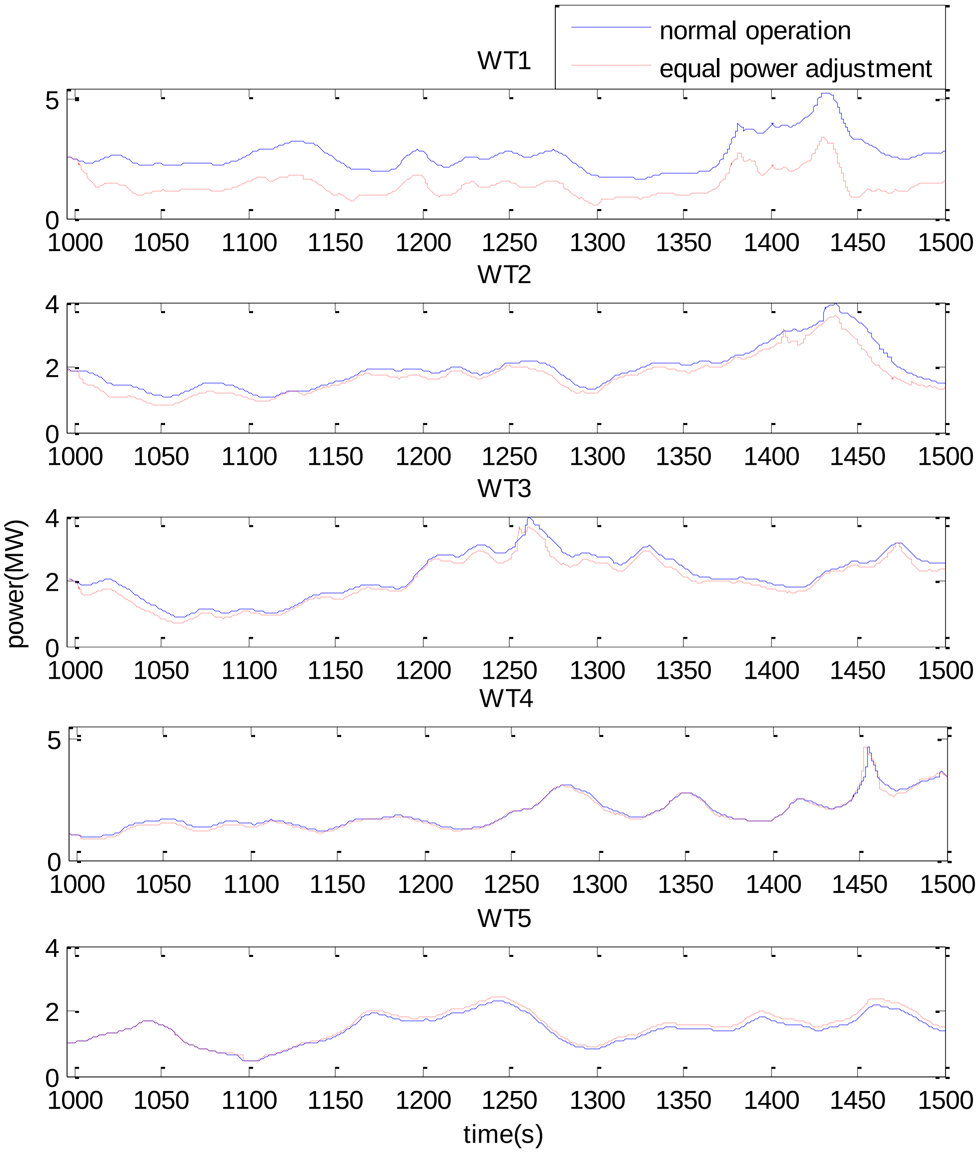


Figure 9-30: Power outputs of the turbines before and after power adjustments

The fore-aft acceleration and the thrust force time series of the wind turbines are examined to investigate the effectiveness of this control strategy in this simulation.

The cumulative PSD of the for-aft acceleration and the thrust force time series are shown in Figure 9-31and Figure 9-32. As can be seen, the level of fore-aft acceleration and thrust at the most upwind turbine (WT1) is reduced significantly. It can be seen that after these power set-point adjustments the level of fore-aft acceleration at all turbines become almost equal.

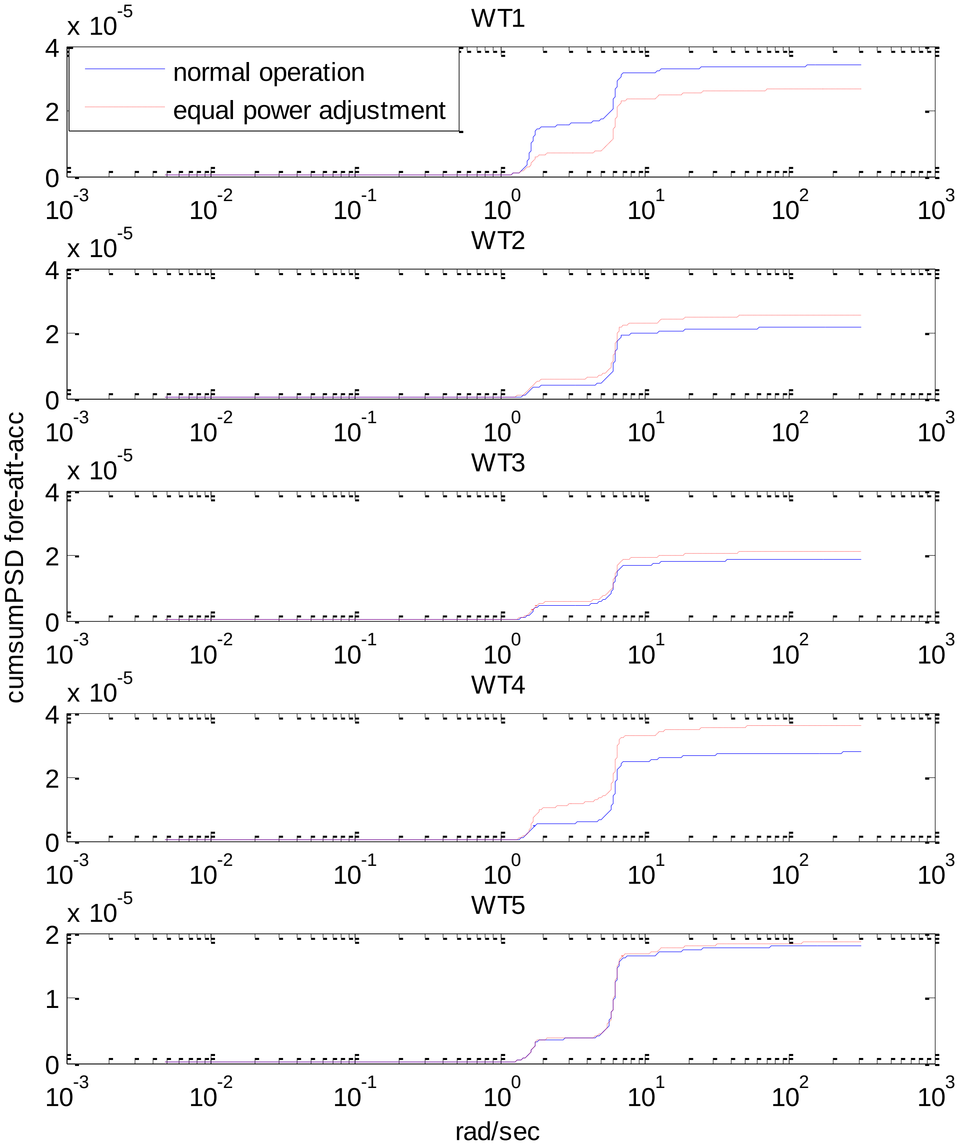


Figure 9-31: Cumulative PSD of the fore-aft acceleration at the turbines before and after power set-point adjustments

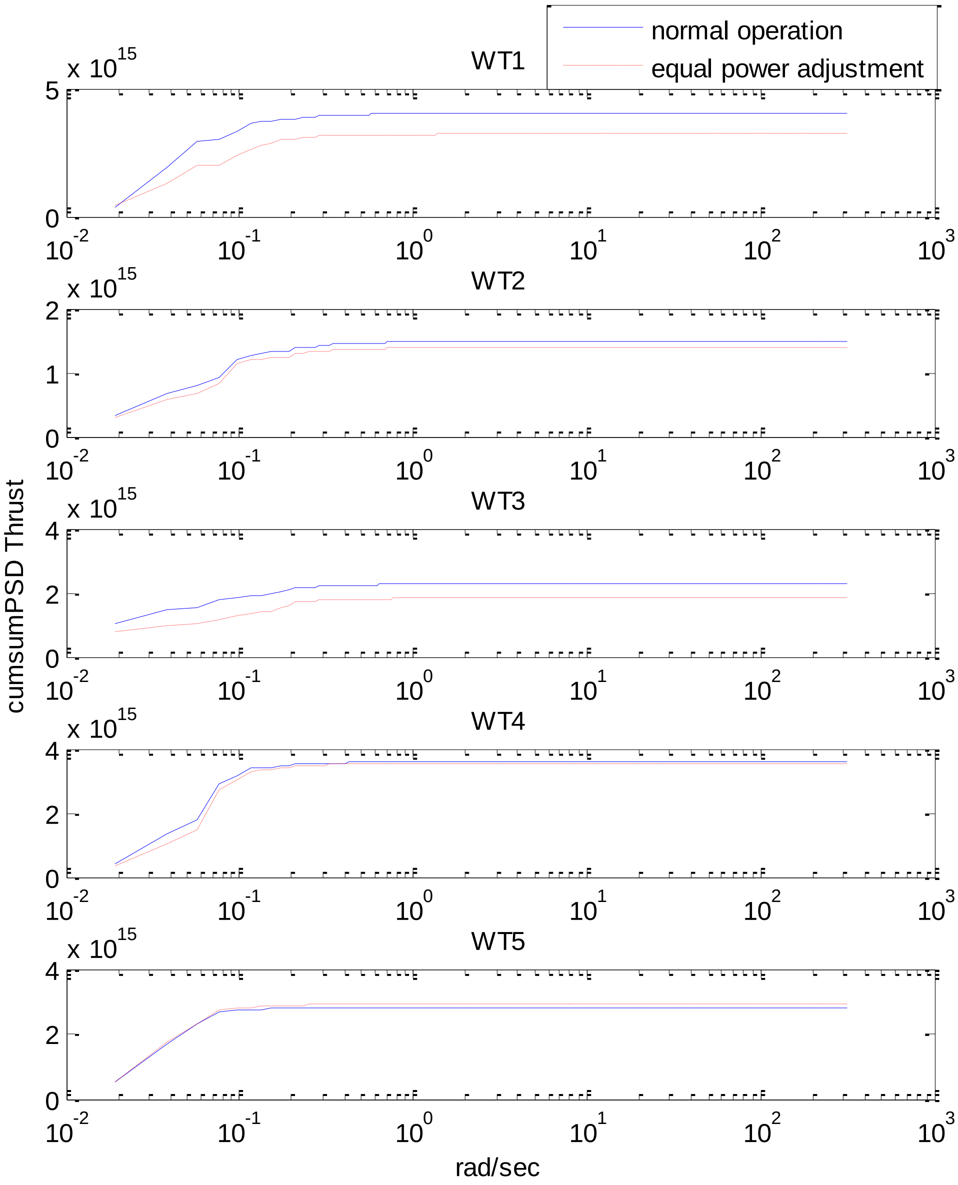


Figure 9-32: Cumulative PSD of the thrust at the turbines before and after power set-point adjustments

The level of thrust at WT1, WT2, WT3, and WT4 are reduced whilst at WT5 is slightly increased. This is due to increase in the wind power in the wake as a result of power set-points adjustments which resulted in higher power generation at WT5.

Figure 9-33 and Figure 9-34 illustrate the mean of the power output and the thrust of the turbines before and after these power set point adjustments. As can be seen, the power output of the turbines are almost equal. The mean of the thrust at most of the turbines are also reduced significantly.

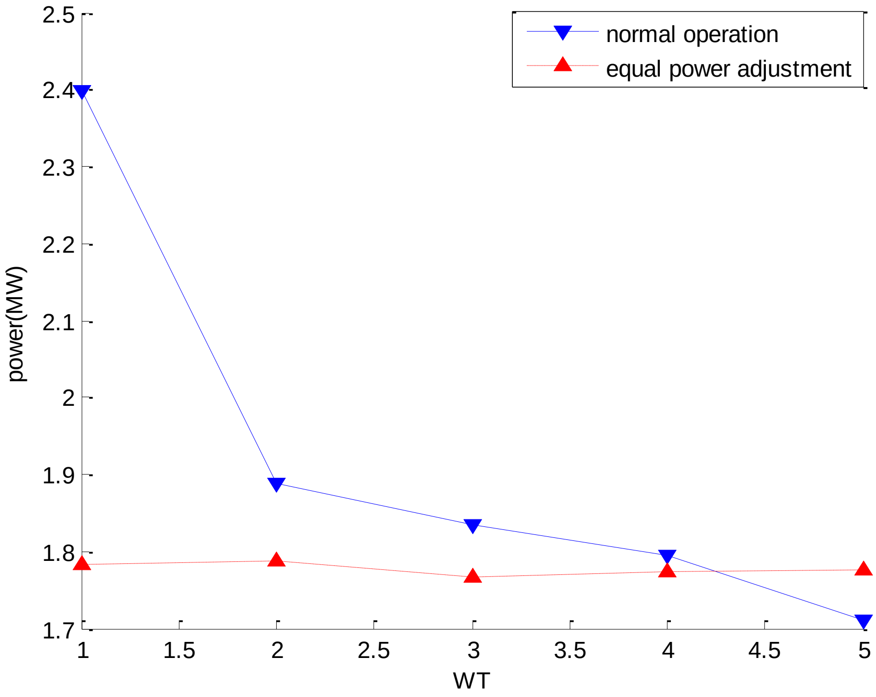


Figure 9-33: Mean of the power output at each turbine before and after power set-point adjustments

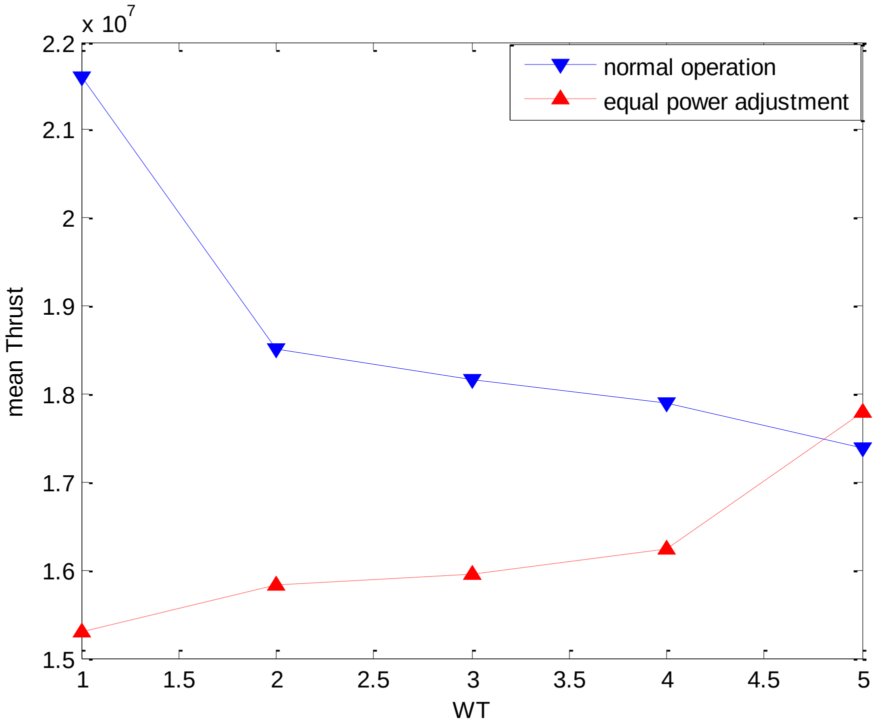


Figure 9-34: Mean of the thrust at each turbine before and after power set-point adjustments

The results suggest that the equal power output generation control strategy is an effective way to distribute the aerodynamic forces between the turbines which consequently results in equal distribution of the aerodynamic loads between the turbines. The power/ load optimisation based will be further investigated in the future work.

9.3 Concluding remarks

Wind farm power optimisation control is investigated in this chapter. The power maximisation of the wind farm is obtained using the Heat & Flux control method. It is shown that, the power maximisation algorithm is mostly effective at below rated wind speeds with small turbulent level. The total power output of the wind farm is increased by about 1% at different constant wind speeds. Furthermore, it is shown that the Heat & Flux control approach is not always effective for power maximisation. Since wake is meander through the wind farm, wind turbines downstream might not always be in the wake of the upstream turbines. Therefore, the power de-loading operation of the upstream turbines may affect the power production in the downwind turbines. Hence, wind farm optimisation control is investigated with different objective.

A simple and effective control strategy is presented for power/load optimisation. The optimisation goal is to generate the same level of power at all turbines in the farm despite the wind and wake conditions at turbines. It is shown that this control strategy is effective to distribute the aerodynamic loads between the turbines. In this control manner all turbines experience almost the same level of structural loads and therefore their operating lifetime may increase. The wind farm optimisation will be investigated more in the future work.

Chapter 10: Conclusion

IN this thesis wind farm modelling and control design through dynamic adjustment of the wind turbines' power set-point in Simulink is addressed. The thesis is divided into two main sections of modelling and controller design. In the modelling section, a Simulink wind farm model suitable for fast simulation and controller design is presented. A literature review is presented in Chapter 2 and Chapter 3 that overviews the necessary requirements and modelling process suitable for developing a wind farm model for controller design and analysis. In Chapter 4, the 5MW Supergen wind turbine model which includes the necessary dynamics and structural modes for controller design and load analysis is detailed. The wind turbine's full envelope controller and the power adjusting controller (PAC) are described. The wind turbine model is tested for different operating scenarios and capability of the wind turbine model for wind farm controller design and load analysis is investigated. It is shown that the wind turbine model includes the most significant structural modes relevant for the load analysis and controller design. It is shown that the wind turbine controllers, including the PAC, are suitable for dynamic wind turbine power set-point adjustment. It is shown that the de-loading control of a wind turbine can reduce the fore-aft acceleration of the wind turbine tower. Therefore, the structural loads on the tower can be reduced by adopting an appropriate control strategy. The wind farm modelling process is defined in Chapter 5. The wind farm modelling process includes, wind field modelling, wake effects modelling, and wind farm modelling. The wind farm modelling requirements for developing a fast simulation wind farm model are reviewed and a Matlab script is developed to automate the process of building a wind farm model with a large number of wind turbines (>50). The wind-field model generates effective wind speed time series at different sampling times allowing for fast simulation using low frequency data points at the farm level but higher frequency data points at the turbine level to investigate load analysis. The wind-field model provides a suitable representation of wake propagation in the wind farm. The wind farm builder code in

Matlab utilises a Simulink library containing the 5MW Supergen wind turbine model, a wind farm template, and a wind turbine template to build wind farm models. The wind turbine template in the library can be used to replace the Supergen 5MW wind turbine model with a user defined wind turbine model.

In the controller design section, the wind farm model is utilised to design control algorithms through coordinating wind turbines power set-points for different control objectives such as primary frequency response, power curtailment, and power maximisation. It is shown that coordination control of wind turbines is an effective control method for improving the performance of wind turbines in presence of the wake. Chapter 6 describes a control structure defined for the wind farm model and utilised for designing different control algorithms. A dispatch power set-point routine is also developed to distribute the total wind farm power adjustment between the turbines according to their operating condition. The wind farm controller coordinates the wind turbines operation in an open-loop manner eliminating the introduction of any additional feedback to the individual wind turbines' full envelope controllers.

The performance of the wind farm model and its controller are demonstrated in Chapter 7 using a wind farm model of 50 turbines. A power curtailment control scenario is investigated to evaluate the wind farm model and its controller performances. Results show that this wind farm model is suitable to design wind farm control algorithms for coordinating operation of the wind turbines. Furthermore, a wind farm model with 10 turbines is built to investigate different wind farm control scenarios including positive and negative wind farm power set-point adjustment in below and above rated mean wind speeds with and without turbulence. The wake propagation and its effect on the wind turbines' performance is shown in these simulations. It is shown that by coordinating the wind turbines' power set-point, the wind farm could be controlled as a single controllable power generation unit.

In Chapter 8 the wind farm model and the wind farm control algorithm are used to investigate primary frequency response, including synthetic inertia and droop control. The status flags from each individual wind turbine are used to dispatch the wind farm reference power between the turbines in a quick and safe operating manner considering their operating conditions. The synthetic inertia response of the wind farm is investigated for an under frequency event, using only the synthetic inertia control routine of the control algorithm and also in combination with the droop control. The combined synthetic inertia response and droop control is investigated for two different control scenarios with and without spinning reserve. The results show that, the primary frequency response is achieved, whilst the operation of the individual wind turbines is kept in a safe operating boundary.

In Chapter 9, the wind farm control algorithm is modified to include the power optimisation objective of the wind farm controller. An optimisation problem is solved for static operating condition to obtain a set of power set-points for wind turbines. The objective of the optimisation problem is to minimise the difference between the wind turbine power outputs. Therefore, power outputs of the wind turbines are forced to be equal to the average power of the wind turbines in the optimisation problem. Initially, the power output of the wind farm is reduced significantly utilising the optimal power set-points obtained from the optimisation problem. However, the optimisation problem is modified and the power losses due to this control strategy are minimised. The optimisation problem is solved for different wind speeds and a general pattern is obtained for a wide range of below rated wind speeds. The power maximisation objective of the wind farm controller is also implemented on the basis of the Heat & Flux control concept. In this control mode a set of power set-points is obtained to maximise the energy yield using trial and error. It is shown that in some wind farm conditions where the upstream wind turbines' wake are not significantly affecting the downstream wind turbines' operation, the Heat and Flux power maximisation control may not effectively increase the total power output of the wind farm. The simulation results show that for the conditions where the downstream wind turbines are fully in the wake of the upstream wind turbines, almost 1% increase in the total power output of the wind farm is achieved using this control strategy.

Possible area for future work includes:

It is shown that this wind farm model includes necessary wind turbine dynamics and structural modes that are necessary for investigating wind turbines' performance under different control scenarios. Since dynamic wake propagation also is included in the model, the wind turbines' performance in the wake can be investigated. It is desirable to study structural loads of the wind turbines operating in the wake. The wind farm model can include a gateway model that connects the Simulink model with the wind turbine model in the GL Bladed. Therefore, load analysis can be done in detail utilizing this capability of the model.

The Supergen wind turbine model can be modified to include a more detailed model of rotor/wind-field interaction. Therefore, stochastic and deterministic loads can be considered in more detail during controller design and performance analysis.

Wind farm layout optimisation is also possible since the wake effects are included in the model.

Since the wind farm controller allows for coordinating the wind turbines operation, it is possible to regard the wind farm as a single controllable unit. Therefore, issues regarding grid connection of high capacity wind farms can be investigated.

Chapter 11: References

[1] D. Le Gouriérès, *Wind Power Plants Fundamentals, Design, Construction and operation,*. Springer, 2012.

[2] J. K. Kaldellis and D. Zafirakis, “The wind energy (r)evolution: A short review of a long history,” *Renew. Energy*, vol. 36, no. 7, pp. 1887–1901, 2011.

[3] P. Gadonneix, *2013 World Energy Issues Monitor*. London: World Energy Council, 2013.

[4] S. Bassi and S. Fankhauser, “The economics of wind power : submission to the inquiry by the House of Commons Select Committee on Energy and Climate Change Samuela Bassi and Sam Fankhauser Policy paper Grantham Research Institute on Climate Change and,” London, 2012.

[5] Giorgio Corbetta & Thomas Miloradovic, “Wind in power,” *EWEA Eropean Stat.*, no. February, pp. 1–12, 2014.

[6] Dolf gielen, “renewable energy technologies: cost analysis series,” International Renewable Energy Agency RENEWABLE, Boon, Germany, 2012.

[7] T. Ashuri, M. B. Zaaijer, J. R. R. a. Martins, G. J. W. van Bussel, and G. a. M. van Kuik, “Multidisciplinary design optimization of offshore wind turbines for minimum levelized cost of energy,” *Renew. Energy*, vol. 68, pp. 893–905, 2014.

[8] L. D. S. Coelho and C.-S. Lee, “Solving economic load dispatch problems in power systems using chaotic and Gaussian particle swarm optimization approaches,” *Int. J. Electr. Power Energy Syst.*, vol. 30, no. 5, pp. 297–307, Jun. 2008.

[9] I. M. de Alegría, J. Andreu, J. L. Martín, P. Ibañez, J. L. Villate, and H. Camblong, “Connection requirements for wind farms: A survey on technical requierements and regulation,” *Renew. Sustain. Energy Rev.*, vol. 11, pp. 1858–1872, 2007.

[10] K. E. Johnson and N. Thomas, “Wind farm control: Addressing the aerodynamic interaction among wind turbines,” *2009 Am. Control Conf.*, pp. 2104–2109, 2009.

[11] N. Fichaux, J. Beurskens, P. H. Jensen, and J. Wilkes, “Design Limits and solution for very large wind turbines,” *UpWind Sixth Framew. Program.*, 2011.

[12] E. Kulunk, *Aerodynamics of wind turbines, Fundumental and advanced topics in wind power. Dr. Rupp Carriveau (Ed.)*. 2011.

[13] T. Burton, N. Jenkins, D. Sharpe, and E. Bossanyi, *Wind Energy Handbook*. 2011.

[14] P. E. Morthorst, “WIND ENERGY - The Facts- Costs & Prices” *EWEA*, vol. 2, no. NEI-DK--4555, 2004.

[15] J. Van Der Tempel, D. Molenaar, and C. N. Delft, “Wind Turbine Structural Dynamics – A Review of the Principles for Modern Power Generation , Onshore

and Offshore," *Wind Eng.*, vol. 26, no. 4, pp. 211–220, 2002.

[16] J. Licari, "Control of a Variable-Speed Wind Turbine," Institute of Energy Cardiff University, 2013.

[17] W. E. Leithead and V. Neilson, "A novel approach to structural load control using Intelligent Actuators," in *17th Mediterranean Conference on Control & Automation, Thessaloniki, Greece*, 2009.

[18] E. Hau, *Wind Turbines, Fundamentals, Technologies, Application, Economics*. Springer, 2006.

[19] F. Bianchi, H. d. Battista, and R. . Mantz, *wind turbine control system principle modelling and Gain scheduling design*. Springer, 2007.

[20] P. J. Schubel and R. J. Crossley, "Wind Turbine Blade Design," *Energies*, vol. 5, no. 12, pp. 3425–3449, 2012.

[21] V. Gupta, "Aerodynamic and structural analyses of the 5 MW wind turbine using BEM and lifting line theories," Eindhoven University of Technology, 2012.

[22] W. E. Leithead and S. Dominguez, "Coordinated Control Design for Wind Turbine Control Systems," *Proc. EWEC*, 2006.

[23] E. Bitar and P. Seiler, "Coordinated Control of a Wind Turbine Array for Power Maximization," *2013 Am. Control Conf.*, pp. 2898–2904, 2013.

[24] Â. Gutie, "Survey of Modelling Methods for Wind Turbine Wakes and Wind Farms," vol. 24, pp. 1–24, 1999.

[25] B. Sanderse, "Aerodynamics of wind turbine wakes," *Energy Res. Cent. Netherlands*, no. ECN-E--09-016, 2009.

[26] D. J. Renkema, "Validation of wind turbine wake models," Delft University of Technology, 2007.

[27] G. C. Larsen, H. A. Madsen, J. Mann, S. Ott, J. N. Sørensen, V. Okulov, N. Troldborg, M. Nielsen, K. Thomsen, T. J. Larsen, A. G. C. Larsen, F. Bingöl, and R. Mikkelsen, "Dynamic wake meandering modeling," RISO, Denmark, 2007.

[28] Andrew karl Scholbrock, "optimizing winf farm control strategies to minimize wake loss effects," University of Colorado, 2011.

[29] G. C. Larsen, H. A. Madsen, J. Torben, and N. Troldborg, "Wake modeling and simulation," Riso, National Laboratory for Sustainable Energy, 2008.

[30] S. F. A.Crespo, J. Hernandez, "Survey of modelling methods for wind turbine wakes and wind farms," *Wind Energy*, no. 2, pp. 1–24, 1999.

[31] A. Makridis and J. Chick, "CFD Modeling of the wake interactions of two wind turbines on a Gaussian hill," in *Eacwe5, Florance*, 2009, vol. 44, no. July, pp. 1–9.

[32] J. Prospathopoulos, "Modelling wind turbine wakes in complex terrain," *Proc. EWEC 2008, Brussels, Belgium*, 2008.

[33] G. van Bussel, "offshore wind farm aspects," *Push. Offshore Wind Energy Reg.*, pp. 1–30, 2006.

[34] J. Schepers, "ENDOW: Validation and improvement of ECN's wake model,"

Riso National Laboratory, 2003.

[35] J. F. Ainslie, “Calculating the flowfield in the wake of wind turbines,” *J. Wind Eng. Ind. Aerodyn.*, vol. 27, pp. 213–224, 1988.

[36] D. Quarton, “The loading of wind turbines in wind farms,” *Garrad Hassan Partners Ltd*, 1996.

[37] W. Tong, S. Chowdhury, J. Zhang, and A. Messac, “Impact of Different Wake Models On the Estimation of Wind Farm Power Generation,” *12th AIAA Aviat. Technol. Integr. Oper. Conf. 14th AIAA/ISSM*, pp. 1–13, 2012.

[38] N.O. Jensen, “On A Note Wind Generator Interaction,” Riso, National Laboratory, 1983.

[39] G. . Larsen, “A simple wake calculation procedure,” RISO National Laboratory, 1988.

[40] S. Frandsen, R. Barthelmie, S. Pryor, O. Rathmann, S. Larsen, J. Højstrup, and M. Thøgersen, “Analytical modelling of wind speed deficit in large offshore wind farms,” *Wind Energy*, vol. 9, no. 1–2, pp. 39–53, Jan. 2006.

[41] M. Méchali, R. Barthelmie, S. Frandsen, L. Jensen, and P.-E. Réthoré, “Wake effects at Horns Rev and their influence on energy production,” *Eur. Wind Energy Conf. Exhib.*, no. 1, pp. 10–20, 2006.

[42] D.J. MILBRROW, “the performance of arrays of wind turbines,” *J. Ind. Aerodyn.*, vol. 5, pp. 403–430, 1980.

[43] E. a. Bossanyi, C. Maclean, G. E. Whittle, P. D. Dunn, N. H. Lipman, and P. J. Musgrove, “The Efficiency of Wind Turbine Clusters,” in *Third International Symposium on Wind Energy Systems*, 1980, pp. 1–16.

[44] D. Madjidian and A. Rantzer, “A Stationary Turbine Interaction Model for Control of Wind Farms,” in *Preprints of the 18th IFAC World Congress*, 2011, pp. 4921–4926.

[45] H. Kooijman and W. F. Engineering, “Reducing Cost of Wind Energy Opportunities to reduce the cost of wind energy,” no. January, pp. 29–30, 2013.

[46] J. T. G. Pierik and P. Bauer, “Wind Farm as Power Plant Dynamic modelling studies,” ECN, Delft University of Technology, 2007.

[47] M. Singh, “Dynamic Models for Wind Turbines and Wind Power Plants Dynamic Models for Wind Turbines and Wind Power Plants,” National Renewable Energy Laboratory, 2011.

[48] V. Gevorgian and E. Muljadi, “Wind Power Plant Short Circuit Current Contribution for Different Fault and Wind Turbine Topologies Preprint,” in *9th Annual International Workshop on Large-Scale Integration of Wind Power into Power Systems*, 2010.

[49] S. Skolthanarat, “The Modeling and Control of a Wind Farm and Grid Interconnection in a Multi-machine System The Modeling and Control of a Wind farm,” Virginia Polytechnic Institute and State University, 2009.

[50] J. Deleuran, S. M. Nourbakhsh, M. Nygaard, M. Soltani, T. Knudsen, M.

Kragelund, and T. Bak, “Aeolus Toolbox for Dynamics Wind Farm Model , Simulation and Control,” in *European Wind Energy Conference and Exhibition*, 2010.

[51] T. G. Couto, B. Farias, A. C. G. C. Diniz, and M. V. G. De Morais, “Optimization of Wind Farm Layout Using Genetic Algorithm,” *10th World Congr. Struct. Multidiscip. Optim.*, pp. 1–10, 2013.

[52] L. Shampine and S. Thompson, “Stiff systems,” *Scholarpedia*, vol. 2, p. 2855, 2007.

[53] L. C. Henriksen, “Model Predictive Control of a Wind Turbine,” *Math. Model.*, pp. 1–7, 2007.

[54] P. Fleming, P. Gebraad, J.-W. van Wingerden, S. Lee, M. Churchfield, Scholbrock, J. Michalakes, K. Johnson, and P. Moriarty, “The SOWFA super-controller: a high-fidelity tool for evaluating wind plant control approaches,” NREL Technical Report, 2013.

[55] T. Knudsen, T. Bak, and M. Svenstrup, “Survey of wind farm control-power and fatigue optimization,” *Wind Energy*, p. n/a-n/a, May 2014.

[56] J. R. Marden, S. D. Ruben, and L. Y. Pao, “A Model-Free Approach to Wind Farm Control Using Game Theoretic Methods,” *IEEE Trans. Control Syst. Technol.*, pp. 1–11.

[57] J. R. Marden and H. P. Young, “Payoff Based Dynamics for Multi-Player Weakly Acyclic Games,” *SIAM J. Control Optim.*, vol. 48, no. 1, pp. 373–396, 2008.

[58] J. Marden, S. Ruben, and L. Pao, “Surveying game theoretic approaches for wind farm optimization,” *50th AIAA Aerosp. Sci. Meet.*, 2012.

[59] T. Horvat, V. Spudic, and M. Baotic, “Quasi-stationary optimal control for wind farm with closely spaced turbines.” pp. 829–834, 2012.

[60] T. Knudsen, T. Bak, M. Soltani, F. B. Vej, and D.-A. Ø, “Distributed Control of Large-Scale Offshore Wind Farms ∗,” *Eur. Wind Energy Conf. Exhib.*, pp. 1–8, 2009.

[61] V. Spudic, “Coordinated Optimal Control of Wind Farm Active Power,” University of Zagreb, 2012.

[62] F. van Dam, “A Maximum Power Point Tracking Approach for Wind Farm Control,” in *Making Torque from Wind*, 2012.

[63] A. J. Brand and J. W. Wagenaar, “Final maps of wind fields and mechanical loads and energy output,” AEOLUS,ICT, 2010.

[64] J. C. S. Lui, “Introduction to Game Theory : Cooperative Games,” Department of Computer Science & Engineering, 2010.

[65] K. Leyton-brown, *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Shoham and Leyton-Brown, 2010.

[66] W. Ren and Y. Cao, *Distributed Coordination of Multi-agent Networks*. London: Springer London, 2011.

[67] K. E. Johnson and G. Fritsch, “Assessment of Extremum Seeking Control for Wind Farm Energy Production,” *Wind Eng.*, vol. 36, no. 1, pp. 701–716, 2012.

[68] J. E. Seem, Y. Li, and Z. Yang, "Maximising Wind Farm Energy Capture VIA Nested-Loop Extremum Seeking Control," *Proc. ASME 2013 Dyn. Syst. Control Conf.*, no. Ilc, pp. 1–8, 2013.

[69] M. A. Rotea, "Dynamic Programming Framework for Wind Power Maximization," *Prepr. 19th World Congr. Int. Fed. Autom. Control*, pp. 3639–3644, 2014.

[70] S.-J. Liu and M. Krstic, *Stochastic Averaging and Stochastic Extremum Seeking*. Springer, 2012.

[71] J. Jonkman, S. Butterfield, W. Musial, and G. Scott, "Definition of a 5-MW Reference Wind Turbine for Offshore System Development Definition of a 5-MW Reference Wind Turbine for Offshore System Development," National Renewable Energy Laboratory, 2009.

[72] V. Spudic and N. Peri, "Hierarchical wind farm control for power / load optimization," in *Torque*, 2010, pp. 1–3.

[73] M. Soltani, R. Wisniewski, P. Brath, and S. Boyd, "Load Reduction of Wind Turbines Using Receding Horizon Control," *2011 IEEE Int. Conf.*, pp. 852–857.

[74] V. Spudi, "Wind Farm Load Reduction via Parametric Programming Based Controller Design," in *The 18th IFAC World Congress*, 2011, pp. 1704–1709.

[75] B. Biegel, "Distributed Control of Wind Farm," AALBORG University, 2011.

[76] F. Heer, P. M. Esfahani, M. Kamgarpour, and J. Lygeros, "Model Based Power Optimisation of Wind Farms," *Eur. Control Conf.*, pp. 1145–1150, 2014.

[77] J. D. Grunnet, M. Soltani, T. Knudsen, M. Kragelund, and T. Bak, "Aeolus Toolbox for Dynamics Wind Farm Model , Simulation and Control," *Eur. Wind Energy Conf. Exhib. EWEC*, 2010.

[78] T. Knudsen and M. Soltani, "Aeolus Toolbox for dynamic wind farm modeling , simulation , and control," ICT aeolus, 2011.

[79] M. Jelavic, M. Baotic, M. Vasak, and V. Spudic, "Reconfigurable control extension," ICT aeolus, 2010.

[80] Paul S. Veers, "Three-Dimensional wind Simulation," Sandia National Laboratories, 1988.

[81] S. Frandsen, R. Barthelmie, S. Pryor, O. Rathmann, and S. Larsen, "analytical modelling of wind speed deficit in large offshore wind farms," in *WIND ENERGY*, 2006, pp. 39–53.

[82] S. Dominguez and P. W. E. Leithead, "Control Design Toolbox for Large-Scale PRVS Wind Turbines," in *Milan, Italy, EWEC-Conference*, 2007.

[83] A. Stock, "Augmented Control for Flexible Operation of Wind Turbines," PhD Thesis, University of strathclyde, 2014.

[84] W. E. Leith, "Effective wind speed models for simple wind turbine simulations," *Br. Wind Energy Conf.*, pp. 321–326, 1992.

[85] D. Robb and W. Leithead, "derivation and validation of simple correlated wind speed models," Internal report, Department of Electrical and Electronic

Engineering, Glasgow, Scotland, 1995.

[86] W. E. Leithead and B. Connor, "Control of variable speed wind turbines: Design task," *Int. J. Control*, vol. 73, no. 13, pp. 1189–1212, Jan. 2000.

[87] W. E. Leithead and D. J. Leith, "Implementation of wind turbine controller," *Int. J. Control*, vol. 63, no. 3, pp. 349–380, 1997.

[88] W. E. Leithead, D. J. Leith, F. Hardan, and H. Markou, "Global Gain-Scheduling control for variable speed wind turbines," *EWEC-Conference*, pp. 853–856, 1999.

[89] S.Poushpas and W. E. Leithead, "application of coordinated and power coordinated control of a 5MW Supergen wind turbine model," *8th_PhD_Seminar_on_Wind_Energy_in_Europe_proceedings*, p. 92, 2012.

[90] T. Lei, M. Barnes, S. Member, S. Smith, S. Hur, A. Stock, and W. E. Leithead, "Using Improved Power Electronics Modeling and Turbine Control to Improve Wind Turbine Reliability," *IEEE PES Trans. Energy Convers.*, vol. 1, pp. 1–9, 2015.

[91] M. Shinozuka, "Simulation of Multivariate and Multidimensional Random Processes," *J. Acoust. Soc. Am.*, no. 1, pp. 357–368, 1971.

[92] B. J. Jonkman and L. Kilcher, "TurbSim User's Guide: Version 1.06.00," NREL Technical Report, 2012.

[93] I. Std, "International Standard IEC 61400-1," vol. 62032, 2005.

[94] S. Krohn, "The Economics of Wind Energy," *Eur. Wind Energy Assoc. EWEA*, 2009.

[95] S. Lynum, "Wind turbine wake meandering," Norwegian University of Science and Technology, 2013.

[96] M. Of and S. Thesis, "Development of a Wind Farm Power Forecast Model," Eindhoven University of Technology, 2012.

[97] S. Hur and W. Leithead, "Curtailment of Wind Farm Power Output Through Flexible Turbine Operation Using Wind Farm Control" *EWEA*, 2014.

[98] A. Buckspan, J. Aho, L. Pao, P. Fleming, and Y. Jeong, "Combining Droop Curve Concepts with Control Systems for Wind Turbine Active Power Control Preprint," no. June, 2012.

[99] E. Muljadi, V. Gevorgian, and M. Singh, "Understanding Inertial and Frequency Response of Wind Power Plants Preprint," *IEEE Symp. power Electron.*, no. July, 2012.

[100] D. Paper, "Rate of Change of Frequency (RoCoF) Modification to the Grid Code," CER, 2014.

[101] D. Spillett, "Frequency changes during large distrurbances and their impact on the total system," National Grid, 2014.

[102] P. Sørensen, A. D. Hansen, F. Iov, F. Blaabjerg, and M. H. Donovan, "Wind farm models and control strategies," RISO National Laboratory, 2005.

[103] M. Tsili and S. Papathanassiou, "A review of grid code technical requirements for wind farms," *IET Renew. Power Gener.*, vol. 3, no. 3, p. 308, 2009.

[104] X. U. E. Yingcheng and T. a I. Nengling, "System frequency regulation investigation in doubly fed induction generator (DFIG)," *WSEAS Trans. power Syst.*, vol. 7, no. 1, pp. 18–26, 2012.

[105] I. a. Erinmez, D. O. Bickers, G. F. Wood, and W. W. Hung, "NGC experience with frequency control in England and Wales-provision of frequency response by generators," *IEEE Power Eng. Soc. 1999 Winter Meet. (Cat. No.99CH36233)*, vol. 1, 1999.

[106] "Security and Quality of Supply Standard," National Grid, 2004.

[107] J. Licari, J. Ekanayake, and I. Moore, "Inertia response from full-power converter-based permanent magnet wind generators," *J. Mod. Power Syst. Clean Energy*, vol. 1, pp. 26–33, 2013.

[108] T. Lei, M. Barnes, S. Member, S. Smith, S. Hur, A. Stock, and W. E. Leithead, "Using Improved Power Electronics Modeling and Turbine Control to Improve Wind Turbine Reliability," pp. 1–9, 2015.

[109] "The Grid Code," National Grid, 2010.

Appendix A: Matlab scripts

A.1 Wind farm generator function

This Matlab function uses various models in a Simulink library that includes required subsystems to build a wind farm model. The Matlab function requests inputs from user and generates wind-field data points for the wind farm layout defined by the user. The generated wind-field data points and the wind farm model then will be saved in the current directory and will be loaded for simulations.

```
%%
% this code works with Matlab 2013

function wind_farm_generator() % wiind farm generator tool

clear all;
options.Resize='on';
options.WindowStyle='modal';
options.Interpreter='none';

%% wind turbine data
radius=63;
%% Number of wind turbines input
qs={'Number of turbines'}
answer={'10'};
Number_WT=inputdlg(qs,'Wind Farm',1,answer,options);
Number_WT=str2double(Number_WT);
%% wind turbines' position input
Grid_N=max(ceil(sqrt(Number_WT)),10);
turbDist = 800;
turbDist1=500;
[yvec xvec] = meshgrid(0:(Grid_N-1),0:(Grid_N-1));
xvec = xvec(1:Number_WT)*turbDist;
yvec = yvec(1:Number_WT)*turbDist + 100;
pos = [xvec(:) yvec(:)]';
dfl={mat2str(pos')};
pos=inputdlg('Enter turbine position matrix (x1 y1 ; x2 y2 ; ... ; xn
yn)','Wind Farm Creator',1,dfl,options);
pos1=pos{1};
pos=str2num(pos{1})';
%% wind_field data input
Lx = max(pos(1,:)) + 100;
Ly = max(pos(2,:)) + 200;
qs={'Mean wind speed [m/s]',...
    'Turbulence intensity',...
    'Wind field lengt [m]',...
    'Wind field width [m]',...
    'Sample Time [s]',...
    'Simulation time [s]',...
                  };
answer={'8','.1',num2str(Lx),num2str(Ly),'1','4000'};
wndparm=inputdlg(qs,'Wind Farm Creator',1,answer,options);

U0=str2double(wndparm{1});
Ti=str2double(wndparm{2});
Lx=str2double(wndparm{3});
Ly=str2double(wndparm{4});
Ts=str2double(wndparm{5});
SimTime=str2double(wndparm{6});
grid.size=Ts*U0;
grid.xsize=Lx;
%% creat and save a model
[file,path]=uiputfile('*.mdl','Save As','windfarm_model');
mdlname=file(1:end-4);
load_system('model_library');
load_system('simulink');
bdclose(mdlname);
mainsys=new_system(mdlname,'Model','model_library/WindFarm');
save_system(mainsys,[path file],'BreakUserLinks',true)
Nstr=int2str(Number_WT);
```

```
%% set wind farm model subsystems' parameters, input/output ports,
positions,

set_param([mdlname '/Turbines'],'UserDataPersistent','on');
set_param([mdlname '/Turbines/Demux'],'Outputs',Nstr);
set_param([mdlname '/Turbines/Demux1'],'Outputs',Nstr);
set_param([mdlname '/Turbines/Demux2'],'Outputs',Nstr);
set_param([mdlname '/Turbines/Demux3'],'Outputs',Nstr);
set_param([mdlname '/Turbines/Demux4'],'Outputs',Nstr);
set_param([mdlname '/Turbines/Demux5'],'Outputs',Nstr);
set_param([mdlname '/Turbines/Mux'],'Inputs',Nstr);
set_param([mdlname '/Turbines/Mux1'],'Inputs',Nstr);
set_param([mdlname '/Turbines/Mux2'],'Inputs',Nstr);
set_param([mdlname '/Turbines/Mux3'],'Inputs',Nstr);
set_param([mdlname '/Turbines/Mux4'],'Inputs',Nstr);
set_param([mdlname '/Turbines/Mux5'],'Inputs',Nstr);
set_param([mdlname '/Turbines/Mux6'],'Inputs',Nstr);
set_param([mdlname '/Turbines/Mux7'],'Inputs',Nstr);
set_param([mdlname '/Turbines/Mux8'],'Inputs',Nstr);
set_param([mdlname '/Turbines/Mux9'],'Inputs',Nstr);
set_param([mdlname '/Turbines/Mux10'],'Inputs',Nstr);
set_param([mdlname '/Turbines/Mux11'],'Inputs',Nstr);
set_param([mdlname '/Turbines/clock/Mux'],'Inputs',Nstr);
set_param([mdlname '/wind_model/CT/ctconcat'],'NumInputs',Nstr);
set_param([mdlname '/wind_model/CT/CTdemux'],'Outputs',Nstr);
set_param([mdlname '/wind_model/ambient/uyMux'],'NumInputs',Nstr);
set_param([mdlname '/Turbines'],'LinkStatus','none');
set_param([mdlname '/wind_model'],'LinkStatus','none');
%% set the blocks' position
y=0;
x=150;
x1=500;
x2=750;
%Initial x and y
tl1x=100;
tl1y=100;
tl2x=10;
tl2y=10;

x1_wt=300;
y1_wt=0;
x2_wt=400;
y2_wt=100;
ofset_wt=0;

%% calculate the wake radius at all downstrean distances from the turbnes
according to equation (5-14)
for i=1:Number_WT
          meand(i,:)=sqrt((radius)^2+(0:grid.size:grid.xsize).*radius/2);
%Equation
          parame.meandr=meand(i,:);
end
%% calculate wake delays
for i=1:Number_WT
    for j=1:Number_WT
        gdelta=max(min(pos(1,i)-pos(1,j),grid.xsize),0);
        parame.dist(i,j)= sqrt(abs(pos(2,j)-pos(2,i))^2+abs(pos(1,j)-
pos(1,i))^2);
        %For downwind turbines
        if (gdelta>0)
                  %Calculate wake delays
            delays(i,j)=parame.dist(i,j)/U0;
        else %Upwind turbines
            delays(i,j)=0;
        end
    end
end
%% save the generated data in the parame data structure
parame.gridsize=Ts*U0;
parame.Ts=Ts;
parame.gridxsize=Lx;
parame.xgrids=floor(pos(1,:)/grid.size)+1;
parame.pos=pos;
parame.U0=U0;
parame.farmx=pos(1,:);
parame.farmy=pos(2,:);
parame.k=2;
parame.alpha=0.5;
parame.r=radius;
parame.Number_WT=Number_WT;
parame.delay=delays;
%% add wind turbine blocks to the wind farm model
count=0;
for i=1:Number_WT
parame.wtx=pos(1,i);
parame.wty=pos(2,i);
parame.wtnum=i;
wt(i).block= add_block('model_library/turbine/', [mdlname
'/Turbines/turbine' int2str(i)]);
set_param(wt(i).block,'Position',[x1_wt
y1_wt+ofset_wt,x2_wt,y2_wt+ofset_wt]);
set_param([mdlname '/Turbines/turbine' int2str(i) ],'LinkStatus','none');
set_param([mdlname '/wind_model/ambient'],'LinkStatus','none');
set_param([mdlname '/wind_model/CT'],'LinkStatus','none');
set_param([mdlname '/Turbines/turbine' int2str(i) '/S-Function'],'Function
Name',['controller' int2str(i)]);
ofset_wt=ofset_wt+150;
if ofset_wt>=1500
        x1_wt=x1_wt+105;
        x2_wt=x2_wt+105;
        y1_wt=0;
        y2_wt=100;
        ofset_wt=0;

end

%% data logging for output signals if required
h=get_param([mdlname '/Turbines/turbine' int2str(i) '/WIND
TURBINE'],'PortHandles');
set_param(h.Outport(1),'DataLogging','on');
set_param(h.Outport(1),'DataLoggingName',['OmGen' int2str(i)]);
set_param(h.Outport(1),'DataLoggingNameMode','Custom');

h=get_param([mdlname '/Turbines/turbine' int2str(i) '/S-
Function'],'PortHandles');
set_param(h.Outport(2),'DataLogging','on');
set_param(h.Outport(2),'DataLoggingName',['GenMom' int2str(i)]);
set_param(h.Outport(2),'DataLoggingNameMode','Custom');
```

```
h=get_param([mdlname '/Turbines'],'PortHandles');
set_param(h.Outport(2),'DataLogging','on');
set_param(h.Outport(2),'DataLoggingName',['Red']);
set_param(h.Outport(2),'DataLoggingNameMode','Custom');

h=get_param([mdlname '/Turbines'],'PortHandles');
set_param(h.Outport(3),'DataLogging','on');
set_param(h.Outport(3),'DataLoggingName',['Amber']);
set_param(h.Outport(3),'DataLoggingNameMode','Custom');

h=get_param([mdlname '/Turbines'],'PortHandles');
set_param(h.Outport(4),'DataLogging','on');
set_param(h.Outport(4),'DataLoggingName',['Green']);
set_param(h.Outport(4),'DataLoggingNameMode','Custom');

h=get_param([mdlname '/Turbines'],'PortHandles');
set_param(h.Outport(14),'DataLogging','on');
set_param(h.Outport(14),'DataLoggingName',['CT']);
set_param(h.Outport(14),'DataLoggingNameMode','Custom');

%% connect wind turbines' ports in the Wind farm block subsystem in the
wind farm model
add_line([mdlname '/Turbines/turbine' int2str(i) ],'iStatus/1','S-
Function/1','Autorouting','on');
add_line([mdlname '/Turbines/turbine' int2str(i) ],'WIND TURBINE/1','S-
Function/2','Autorouting','on');
add_line([mdlname '/Turbines/turbine' int2str(i) ],'WIND TURBINE/2','S-
Function/3','Autorouting','on')
add_line([mdlname '/Turbines/turbine' int2str(i) ],'DelTP/1','S-
Function/5','Autorouting','on');
add_line([mdlname '/Turbines/turbine' int2str(i) ],'P1/1','S-
Function/6','Autorouting','on');
add_line([mdlname '/Turbines/turbine' int2str(i) ],'WIND TURBINE/3','S-
Function/4','Autorouting','on');
add_line([mdlname '/Turbines/turbine' int2str(i) ],'Priority/1','S-
Function/7','Autorouting','on');
add_line([mdlname '/Turbines/turbine' int2str(i) ],'Fast_Slow/1','S-
Function/8','Autorouting','on');
add_line([mdlname '/Turbines/turbine' int2str(i) ],'S-Function/1','WIND
TURBINE/2','Autorouting','on');
add_line([mdlname '/Turbines/turbine' int2str(i) ],'S-Function/2','WIND
TURBINE/3','Autorouting','on');
add_line([mdlname '/Turbines/turbine' int2str(i) ],'S-
Function/3','Pm/1','Autorouting','on');
add_line([mdlname '/Turbines/turbine' int2str(i) ],'S-
Function/4','V_est/1','Autorouting','on');
add_line([mdlname '/Turbines/turbine' int2str(i) ],'S-
Function/5','Red/1','Autorouting','on');
add_line([mdlname '/Turbines/turbine' int2str(i) ],'S-
Function/6','Amber/1','Autorouting','on');
add_line([mdlname '/Turbines/turbine' int2str(i) ],'S-
Function/7','Green/1','Autorouting','on');
add_line([mdlname '/Turbines/turbine' int2str(i) ],'S-
Function/8','Rejection/1','Autorouting','on');
add_line([mdlname '/Turbines/turbine' int2str(i) ],'S-
Function/9','Terminator1/1','Autorouting','on');
```

```
add_line([mdlname '/Turbines/turbine' int2str(i) ],'S-
Function/10','Terminator2/1','Autorouting','on');
add_line([mdlname '/Turbines/turbine' int2str(i) ],'S-
Function/11','Terminator3/1','Autorouting','on');
add_line([mdlname '/Turbines/turbine' int2str(i) ],'S-
Function/12','Terminator4/1','Autorouting','on');
add_line([mdlname '/Turbines/turbine' int2str(i) ],'S-
Function/13','Recovery/1','Autorouting','on');
add_line([mdlname '/Turbines/turbine' int2str(i) ],'S-
Function/14','Terminator5/1','Autorouting','on');
add_line([mdlname '/Turbines/turbine' int2str(i) ],'S-
Function/15','recovery_complete/1','Autorouting','on');
add_line([mdlname '/Turbines/turbine' int2str(i) ],'S-
Function/16','PAC_ON/1','Autorouting','on');
add_line([mdlname '/Turbines/turbine' int2str(i) ],'S-
Function/17','Terminator13/1','Autorouting','on');
add_line([mdlname '/Turbines/turbine' int2str(i) ],'S-
Function/18','Terminator14/1','Autorouting','on');
add_line([mdlname '/Turbines/turbine' int2str(i) ],'S-
Function/19','Terminator15/1','Autorouting','on');
add_line([mdlname '/Turbines/clock'],'CLk/1',['Mux' '/'
int2str(i)],'Autorouting','on');
add_line([mdlname '/Turbines'],['Demux/' int2str(i)],['turbine' int2str(i)
'/1'],'Autorouting','on');
add_line([mdlname '/Turbines'],['Demux1/' int2str(i)],['turbine' int2str(i)
'/2'],'Autorouting','on');
add_line([mdlname '/Turbines'],['Demux2/' int2str(i)],['turbine' int2str(i)
'/3'],'Autorouting','on');
add_line([mdlname '/Turbines'],['Demux3/' int2str(i)],['turbine' int2str(i)
'/4'],'Autorouting','on');
add_line([mdlname '/Turbines'],['Demux4/' int2str(i)],['turbine' int2str(i)
'/5'],'Autorouting','on');
add_line([mdlname '/Turbines'],['Demux5/' int2str(i)],['turbine' int2str(i)
'/6'],'Autorouting','on');
add_line([mdlname '/Turbines'],['turbine' int2str(i) '/1'],['Mux/'
int2str(i)],'Autorouting','on');
add_line([mdlname '/Turbines'],['turbine' int2str(i) '/2'],['Mux1/'
int2str(i)],'Autorouting','on');
add_line([mdlname '/Turbines'],['turbine' int2str(i) '/3'],['Mux2/'
int2str(i)],'Autorouting','on');
add_line([mdlname '/Turbines'],['turbine' int2str(i) '/4'],['Mux3/'
int2str(i)],'Autorouting','on');
add_line([mdlname '/Turbines'],['turbine' int2str(i) '/5'],['Mux4/'
int2str(i)],'Autorouting','on');
add_line([mdlname '/Turbines'],['turbine' int2str(i) '/6'],['Mux5/'
int2str(i)],'Autorouting','on');
add_line([mdlname '/Turbines'],['turbine' int2str(i) '/7'],['Mux6/'
int2str(i)],'Autorouting','on');
add_line([mdlname '/Turbines'],['turbine' int2str(i) '/8'],['Mux7/'
int2str(i)],'Autorouting','on');
add_line([mdlname '/Turbines'],['turbine' int2str(i) '/9'],['Mux8/'
int2str(i)],'Autorouting','on');
add_line([mdlname '/Turbines'],['turbine' int2str(i) '/10'],['Mux9/'
int2str(i)],'Autorouting','on');
add_line([mdlname '/Turbines'],['turbine' int2str(i) '/11'],['Mux10/'
int2str(i)],'Autorouting','on');
add_line([mdlname '/Turbines'],['turbine' int2str(i) '/12'],['Mux11/'
int2str(i)],'Autorouting','on');

%% add wind-field block components
selector(i).block=add_block('simulink/Signal
Routing/Selector',[mdlname '/wind_model/ambient' '/uy' int2str(i)]);
sel1=get_param(selector(i).block,'Position');
set_param(selector(i).block,'Position',[tl2x+x1 tl2y+y,sel1(3)+x1-sel1(1)
sel1(4)+y-sel1(2)]);
set_param(selector(i).block,'Indices',['n/a,' int2str(i)]);
%
set_param(selector(i).block,'NumberOfDimensions',int2str(2),'IndexOptions',
'Select all,Index Vector (dialog)');

y=y+10;

add_line([mdlname '/wind_model/ambient'],['Selector1/1'],['uy' int2str(i)
'/1'],'Autorouting','on');
add_line([mdlname '/wind_model/ambient'],['uy' int2str(i) '/1'],['uyMux/'
int2str(i)],'Autorouting','on');

mux(i).block=add_block('simulink/Signal Routing/Mux',[mdlname
'/wind_model/CT' '/ctMux' int2str(i)]);
set_param([mdlname '/wind_model/CT/ctMux'  int2str(i)],'Inputs',Nstr);
add_line([mdlname '/wind_model/CT'],['ctMux' int2str(i) '/1'],['ctconcat/'
int2str(i)],'Autorouting','on');
for j=1:Number_WT%length(delays(i,:))
    if(delays(i,j)>0)
       count=count+1;
        ctd(j).block=add_block('simulink/Discrete/Integer Delay',[mdlname
'/wind_model/CT' '/ctDelay' int2str(i) ',' int2str(j)]);
        set_param([mdlname '/wind_model/CT' '/ctDelay' int2str(i) ','
int2str(j)],'NumDelays',int2str(floor(delays(i,j))));
        set_param([mdlname '/wind_model/CT' '/ctDelay' int2str(i) ','
int2str(j)],'samptime',num2str(parame.Ts));
        add_line([mdlname '/wind_model/CT'],['CTdemux/'
int2str(j)],['ctDelay' int2str(i) ',' int2str(j) '/1'],'Autorouting','on');
        add_line([mdlname '/wind_model/CT'],['ctDelay' int2str(i) ','
int2str(j) '/1'],['ctMux' int2str(i) '/' int2str(j)],'Autorouting','on');

    else
       add_block('simulink/Sinks/Terminator',[mdlname '/wind_model/CT'
'/TermWdde' int2str(i) ',' int2str(j)]);
       add_line([mdlname '/wind_model/CT'],['CTdemux' '/' int2str(j)
],['TermWdde' int2str(i) ',' int2str(j) '/1'],'Autorouting','on');
       add_line([mdlname '/wind_model/CT'],['Constant/1'],['ctMux'
int2str(i) '/' int2str(j)],'Autorouting','on');

    end
end

end

%% generate wind-field data
wind=wind_field_final(U0,Ti,Ts,Lx,Ly,SimTime,pos);
%save wind-field data to the parame data structure
parame.Uy=wind.Uy;
parame.Ux=wind.ave_turb;
parame.meanV=wind.ave_mean;
parame.wind=wind;
%% generate high frequency points using Dryden spectra "interpolation"
L=200;
a=1.14*U0/L;
b=Ti*U0*sqrt(2*a);
```

```
fHz=2;
t=0:1/fHz:1;
x_t=zeros(length(t),1);
V_highfq=zeros(fHz*SimTime+1,Number_WT);
z_sigma=sqrt((2*b^2)*((sinh(a*(1/fHz))*sinh(a*(1/fHz)))/(sinh(a*(1/fHz))))/
(2*a));%
for k=1:Number_WT
n=1;
m=length(t);
 for i=1:SimTime+1
x_ti=wind.ave_turb(i,k);
x_tf=wind.ave_turb(min(i+1,SimTime+1),k);
t_i=i-1;
t_f=i;
t=t_i:1/fHz:t_f;
   for j=1:length(t)

      if(t(j)==t_i)
         x_t(j)=x_ti;

     elseif (t(j)==t_f)
           x_t(j)=x_tf;

      else
        z_mu=(sinh(a*(t(j)-t_i))*x_tf+sinh(a*(t_f-t(j)))*x_ti)/sinh(a*(t_f-
t_i));
        x_t(j)=normrnd(z_mu,z_sigma);
        t_i=t(j);
        x_ti=x_t(j);

     end

   end

 V_highfq(n:m,k)=x_t;
 n=n+length(t)-1;
 m=m+length(t)-1;

  end

end

parame.v_highfq=V_highfq;% save high frequency data points in the parame
data structure

%% generating effective wind speed by spatially filtering the wind-field
turbulence data+ high frequency interpolation points
open_system ('simulink_wind');% open the spatial filter model
for m=1:Number_WT%Number_WT
mean_wind_speed =U0;
% Var_wind = var(parame.meanV(:,m));
Var_wind = var(parame.v_highfq(:,m));
Turb_inten = Ti;
Start_point = parame.v_highfq(1,1);
Ts_wind = Ts;
Total_time = SimTime;
External_point_wind(:,1)=1:1:Total_time+1;
External_point_wind(:,2)=parame.v_highfq(:,m);
Point_wind_switch = 1;
```

```
sample_time=Ts;
h = 90;%Turbine hub height
R = 63; %Turbine rotor radius
Gamma = 1.3;
a_cont = 0.55; % This is a fixed value and it is in wind_filter.m for
discrete sys
Time_constant = 30;
w1 = randn(1);
w2 = randn(1);
w3 = randn(1);
w4 = randn(1);
w_tot = [w1;w2;w3;w4];
set_param(gcs,'Stop time',num2str(Total_time));
set_param(gcs,'LoadInitialState','off');
set_param(gcs,'SaveFinalState','on');
sim('simulink_wind')% simulate the wind filter model for number of turbine
times

parame.Veff(:,m)=v3_sim; % save effective wind-field data points in the
parame data structure
end
bdclose('simulink_wind');% close the wind-filter model



%% set parameters for the wind farm model subsystems
set_param([mdlname '/wind_model'],'UserData',parame);
set_param([mdlname '/wind_model'],'UserDataPersistent','on');

set_param([mdlname '/Turbines'],'UserData',parame);
set_param([mdlname '/Turbines'],'UserDataPersistent','on');

set_param([mdlname '/Wind Farm Controller'],'UserData',parame);
set_param([mdlname '/Wind Farm Controller'],'UserDataPersistent','on');

set_param([mdlname '/Wind Farm Controller'],'UserData',parame);
set_param([mdlname '/Wind Farm Controller'],'UserDataPersistent','on');

set_param([mdlname '/Wind Farm
Controller'],'Parameters',int2str(parame.wtnum));

set_param([mdlname '/wind_model/ambient'],'UserData',parame);
set_param([mdlname '/wind_model/ambient'],'UserDataPersistent','on');
%% set the simulink solver parameters
set_param(mainsys,'Solver','ode23tb');
set_param(mainsys,'SimulationMode','accelerator');
set_param(mainsys,'StopTime',num2str(SimTime));
set_param(mainsys,'InlineParams','on');
set_param(mainsys,'EnhancedBackFolding','on');
%% save the wind farm model, wind farm model parameters
save_system(mainsys);
open_system(mdlname);
%% open the wind farm model and load the parameters for the simulation
load('Turbine_Controller_parameters_new.mat');
save([mdlname 'parameters.mat'],'parame')
```

A.2 Wind-field time series generator function

This Matlab function generates wind speed time series for each turbine in the farm.

```
function wind=wind_field_final(U0,Ti,Ts,Lx,Ly,SimTime,pos)

waitH = waitbar(0,'Please wait...');
 waitbar(1 / 2)
%% wind field parameters
Lambdau = 42;
cu= 7.1; cv=4.2;
d=Ts*U0;
Nx = ceil(Lx/d);
Ny = ceil(Ly/d);
N = ceil(SimTime/Ts) + Nx;
Lv= 2.7*Lambdau;
Sigmau= Ti*(0.75*U0+5.6);
Sigmav= 0.8*Sigmau;
gridSize=d;
[pom,index] = sortrows(pos');
Vm=U0;%input('Specify the mean speed value [m/s] \n');
TI=Ti;%input('Specify the turbulence intensity [%] \n');
time_increment=Ts;%input('Specify the sampling interval \n');
sample_time=time_increment;
standard_deviation=TI*Vm;
L=170.1;%turbulence length scale IEC standard for Kaimal model >30m of the
ground
w1=randn(1,SimTime+1);%generates random 30001 random numbers, variance 1.
N=length(w1);
time=time_increment:time_increment:time_increment*N;
w1=w1./sqrt(time_increment);% conversion of random numbers to discrete time
white noise
frequency_increment=(2*pi)/(N*time_increment);
frequency2=(-
N*frequency_increment:frequency_increment:frequency_increment*(N-1));
%% Generate Kaimal spectrum
kaimal=real((1/3*(standard_deviation^2)*(L/Vm))./((1+((abs(frequency2).*L).
/Vm)).^(5/3)));
kaimal_periodic=kaimal(N+1:2*N)+kaimal(1:N);
%% Generate wind data
Ns=length(kaimal_periodic);
N_turb=length(pos);
for k=1:N_turb
    for h=k:N_turb
        d_turb(k,h)=sqrt(abs(pos(2,h)-pos(2,k))^2+abs(pos(1,h)-
pos(1,k))^2);
```

```
        d_turb(h,k)=d_turb(k,h);
    tau(k,h)=d_turb(k,h)/U0;
    tau(h,k)=-tau(k,h);
        alpha1(k,h)=atan2(abs(pos(2,k)-pos(2,h)),abs(pos(1,k)-pos(1,h)));
        alpha1(h,k)=alpha1(k,h);
    end;
end;
a=sqrt((cu*cos(alpha1)).^2+(cv*sin(alpha1)).^2);

gamma=zeros(N_turb,N_turb,Ns);
ff=frequency2(Ns+1:end);

for i=1:N_turb
   for j=1:N_turb
        gamma(i,j,1:Ns)=exp(-a(i,j)*d_turb(i,j)/U0*ff);
    end;
end;

Su1=sqrt(2*pi*kaimal_periodic);
S1=zeros(N_turb,N_turb,Ns);

for i=1:N_turb
    S1(i,i,1:Ns)=Su1;
end;

for i=1:N_turb
    S_ii1(1:Ns)=S1(i,i,1:Ns);
    for j=1:N_turb
        S_jj1(1:Ns)=S1(j,j,1:Ns);
        if (i~=j)
            gamma_ij1(1:Ns)=gamma(i,j,1:Ns);
            S1(i,j,1:Ns)=gamma_ij1.*sqrt(S_ii1.*S_jj1);
       end;
    end;
 end;
HH1=zeros(N_turb,N_turb,Ns);
 H1=zeros(N_turb);
 V_turb1=zeros(N_turb,Ns);

 for f=1:Ns

 for k=1:N_turb

   for j=1 :N_turb
       b=0;
    if (j==k)
      for l=1:k-1
       b=b+H1(k,l)^2;
      end

     H1(k,k)=(S1(k,k,f)-b)^0.5;
    else
c=0;
     for l=1:k-1
       c=c+H1(j,l)*H1(k,l);
     end
```

```
if(H1(k,k)==0);
        H1(j,k)=0;
      else
          H1(j,k)=(S1(j,k,f)-c)/H1(k,k);
      end

    end

   end

 end

HH1(:,:,f)=H1(:,:);

phi_r=randn(N_turb,1).*(2*pi);
    X=exp(sqrt(-1).*phi_r);
    V_turb1(1:N_turb,f)=HH1(:,:,f)*X;
    for i=1:N_turb
V_turb1(i,f)=sum(V_turb1(i,f));
    end;
 end

V_turb1(1:N_turb,1)=zeros(1,N_turb);
V_turb1=[V_turb1 fliplr(conj(V_turb1(1:N_turb,2:end)))];
v_turb1=zeros(size(V_turb1));

parfor i=1:N_turb
    v_turb1(i,:)=ifft(V_turb1(i,:));
end;

for i=1:N_turb
    pom1(index(i),1:size(v_turb1,2))=v_turb1(i,:);
end;
v_turb1=pom1;

%% wind data time series
waitbar(2 / 2)
T=length(frequency2)-1;
t=0:1:T-1;
tsim1=0:Ts:SimTime;
v_turb_sim_mean1=interp1(t,v_turb1',tsim1)+U0;
v_turb_sim_mean1=interp1(t,v_turb1',tsim1)+U0;
v_turb_sim1=interp1(t,v_turb1',tsim1);
[M,N_turb] = size(v_turb_sim1);
gridN=1;
for k = 1:N_turb/gridN
        wind_turb1{k} = v_turb_sim1(:,((k-1)*gridN+1):((k-1)*gridN+gridN));
        wind_turb_mean1{k} = v_turb_sim_mean1(:,((k-1)*gridN+1):((k-
1)*gridN+gridN));
end
amb_wind.wind = wind_turb1;
amb_wind.wind_mean = wind_turb_mean1;
for m=1:N_turb/gridN
wind_turbav1(:,m) = sum(amb_wind.wind{1,m},2)./(gridN);
wind_turbav_mean1(:,m) = sum(amb_wind.wind_mean{1,m},2)./(gridN);

Ns= 2^ceil(log(N)/log(2));
```

```
DF=1/(2*Ns*Ts);
F= (0:Ns)'*DF;
Sv= Sigmav^2*4*Lv/U0./(1+6*F*Lv/U0).^(5/3);
Cv= exp(-cv*d/U0*F);
[V,t] = Simlateral(Sv,Cv,Ny,DF);
Uy = V';
%% save wind data

amb_wind.time = tsim1;
amb_wind.num_of_turb = N_turb/gridN;
amb_wind.gridN = gridN;
amb_wind.gridSize = gridSize;
amb_wind.turb_pos = pos;
amb_wind.wind = wind_turb1;
amb_wind.wind_mean = wind_turb_mean1;
wind.gridSize=d;
wind.Ti=Ti;
wind.ave_turb= wind_turbav1;
wind.ave_mean=wind_turbav_mean1;
wind.Ts=Ts;
wind.Uy=Uy;
wind.Umean = U0;
wind.grid.size=d;
wind.grid.ysize=Ly;
wind.grid.xsize=Lx;
wind.SimTime = SimTime;

end

close(waitH)

end
```

A.3 Effective Wind-field spatial filter

The effective wind speed data points are generated using a spatial filter model. The model filters the generated wind-field time series and saves the results in a data structure.

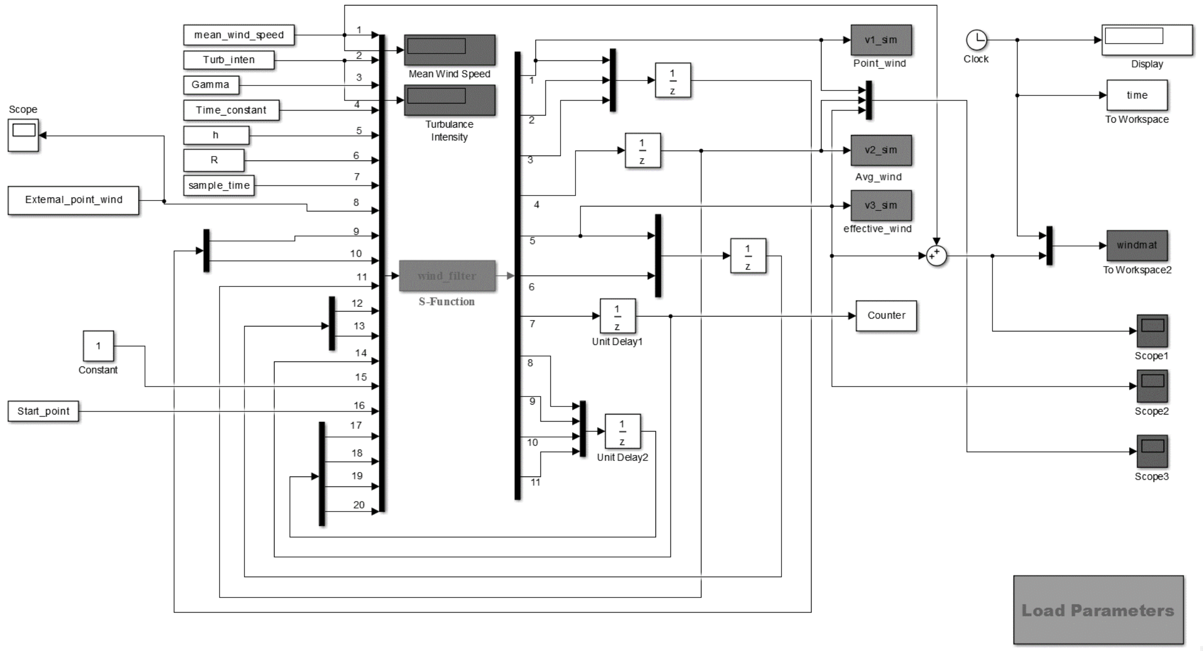


Figure 12. 1: wind filter

Matlab function used in the Wind filter model

```
function [sys,x0,str,ts] = wind_filter(t,x,u,flag)
% Dispatch the flag. The switch function controls the calls to
% S-function routines at each simulation stage.
switch flag,

    case 0
        [sys,x0,str,ts] = mdlInitializeSizes; % Initialization

    case 3
        sys = mdlOutputs(t,x,u); % Calculate outputs

    case { 1, 2, 4, 9 }
        sys = []; % Unused flags

    otherwise
        error(['Unhandled flag = ',num2str(flag)]); % Error handling
end;
% End of function wind_filter.

%% ============================================================
% Function mdlInitializeSizes initializes the states, sample
% times, state ordering strings (str), and sizes structure.
%==============================================================
function [sys,x0,str,ts] = mdlInitializeSizes
% Call function simsizes to create the sizes structure.
sizes = simsizes;
% Load the sizes structure with the initialization information.
sizes.NumContStates= 0;
sizes.NumDiscStates= 0;
sizes.NumOutputs=    11;
sizes.NumInputs=     20;
sizes.DirFeedthrough=1;
sizes.NumSampleTimes=1;
% Load the sys vector with the sizes information.
sys = simsizes(sizes);
%
x0 = []; % No continuous states
%
str = []; % No state ordering
%
ts = [-1 0]; % Inherited sample time
% End of mdlInitializeSizes.
%% ============================================================
% Function mdlOutputs performs the calculations.
%==============================================================
function sys = mdlOutputs(~,~,u)
mean_wind_speed = u(1);
Turb_inten = u(2);
Gamma = u(3);
Time_constant = u(4);
h = u(5);
R = u(6);
Ts_wind = u(7);
v1= u(8);
v1_minus1 = u(9);
```

```
v1_minus2 = u(10);
v2= u(11);
v3= u(12);
v3_minus2 = u(13);
Count = u(14);
Point_wind_switch = u(15);
Start_point = u(16);
w1_in =u(17); w2_in =u(18);w3_in =u(19);w4_in =u(20);

%% Calculate Parameters
Var_wind = (mean_wind_speed*Turb_inten/100)^2;
Sigma = (Gamma*R/(v2+mean_wind_speed));
L_a = (200/30)*h; % Turbulence length scale - assumes this scales linearly
with R and 200m is a reasonable value for L_a when h=30m

a = 1.14*(mean_wind_speed/L_a);
b = 1/Time_constant;
c = sqrt(2)/(sqrt(0.55)*Sigma); % Assumes a'=0.55
d = sqrt(0.55)/Sigma;           % as above
e = sqrt(2)/Sigma;

k1 = sqrt(Var_wind)*sqrt(2*a);
k2 = 1/Time_constant;
k3 = 1/Sigma;

%% Calculate state matrices A_sm, B_sm and C_sm
A_sm = ...
    [-a 0 0 0;...
     0 -b 0 0;...
     0 0 -c 0;...
     0 0 0 -d];

%Calculate A_smd for discretisation
A_smd = A_sm;
for i = 1:size(A_sm,2)
    A_smd(i,i)=exp(A_sm(i,i)*Ts_wind);
end
clear i;

Eig_vector = ...
    [1, -1, 1, -1;...
    -a, b, -c, d;...
    a^2,-b^2,c^2,-d^2;...
    -a^3,b^3,-c^3,d^3];

Inv_eig_vector = inv(Eig_vector);

B_sm = Inv_eig_vector*[0;0;0;k1*k2*k3];

C_sm = [b*c*d/(k2*k3),(d*c+b*c+b*d)/(k2*k3),(b+c+d)/(k2*k3),1/(k2*k3);...
        c*d/k3,(c+d)/k3,1/k3,0;...
        e*b/k2,(e+b)/k2,1/k2,0]*Eig_vector;

%% Calculate Covariance:
P = zeros(size(A_sm));
P(1,1)=B_sm(1)^2*(1-exp(-2*a*Ts_wind))/(2*a);
P(2,2)=B_sm(2)^2*(1-exp(-2*b*Ts_wind))/(2*b);
```

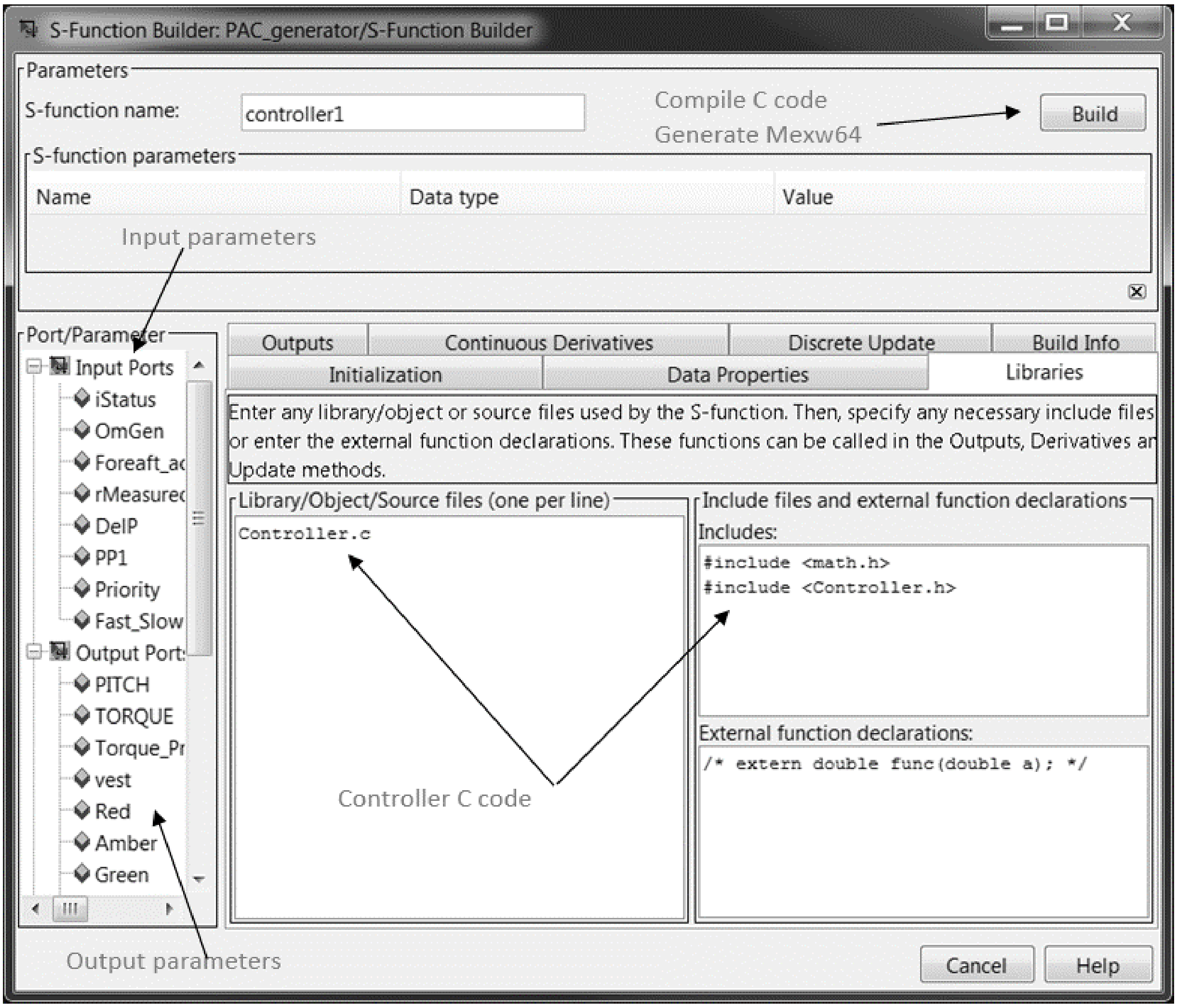
```
P(3,3)=B_sm(3)^2*(1-exp(-2*c*Ts_wind))/(2*c);
P(4,4)=B_sm(4)^2*(1-exp(-2*d*Ts_wind))/(2*d);

P(1,2)=B_sm(1)*B_sm(2)*(1-exp(-(a+b)*Ts_wind))/(a+b);      P(2,1)=P(1,2);
P(1,3)=B_sm(1)*B_sm(3)*(1-exp(-(a+c)*Ts_wind))/(a+c);      P(3,1)=P(1,3);
P(1,4)=B_sm(1)*B_sm(4)*(1-exp(-(a+d)*Ts_wind))/(a+d);      P(4,1)=P(1,4);
P(2,3)=B_sm(2)*B_sm(3)*(1-exp(-(b+c)*Ts_wind))/(b+c);      P(3,2)=P(2,3);
P(2,4)=B_sm(2)*B_sm(4)*(1-exp(-(b+d)*Ts_wind))/(b+d);      P(4,2)=P(2,4);
P(3,4)=B_sm(3)*B_sm(4)*(1-exp(-(c+d)*Ts_wind))/(c+d);      P(4,3)=P(3,4);
Pn = P;
%% Calculate coefficients (k_mn)
k = diag([1 1 1 1]);
D1 = Pn(1,1);k(2,1)=Pn(2,1)/D1;k(3,1)=Pn(3,1)/D1;k(4,1)=Pn(4,1)/D1;
D2 = Pn(2,2)-k(2,1)^2*D1;
k(3,2) = (Pn(3,2)-k(2,1)*D1*k(3,1))/D2;
D3 = Pn(3,3)-k(3,1)^2*D1-k(3,2)^2*D2;
k(4,2) = (Pn(4,2)-k(2,1)*D1*k(4,1))/D2; k(4,3) = (Pn(4,3)-k(3,1)*D1*k(4,1)-
k(3,2)*D2*k(4,2))/D3;
D4 = Pn(4,4)-k(4,1)^2*D1-k(4,2)^2*D2-k(4,3)^2*D3;

Do = [D1,D2,D3,D4];
D = diag(Do);

for i = 1:size(D,1)
    if abs(D(i,i)) <= 1e-4 && D(i,i) < 0
        D(i,i) = 0;
    end
end

Dd = sqrt(D);
k = k*Dd;

iskreal = isreal(k);
if iskreal~=1
    disp('error');
    disp(k);
    disp(P);
    disp(Pn);
    disp(D);
end
%% Calculate numerical output
%Initialisation:
w1_out = randn(1);
w2_out = randn(1);
w3_out = randn(1);
w4_out = randn(1);

% Calculate v1 and v2:
if Point_wind_switch == 0
    v1_Count = A_smd(1,1)*v1 + C_sm(1,1)*k(1,1)*w1_out(1); % Calculate v1
v1
else v1 = v1 - Start_point; % v1 is already given, use v1 to first
calculate w1
    v1_Count = v1;
    w1_in = (1/C_sm(1,1)/k(1,1))*(v1_minus1-A_smd(1,1)*v1_minus2);
    w1_out = (1/C_sm(1,1)/k(1,1))*(v1_Count-A_smd(1,1)*v1_minus1);
end
```

```
x_1_minus2 = (1/C_sm(1,1))*v1_minus2;
x_1_minus1 = (1/C_sm(1,1))*v1_minus1;

w_tot_in = [w1_in;w2_in;w3_in;w4_in];
w_tot_out = [w1_out;w2_out;w3_out;w4_out];

x_1 = (1/C_sm(1,1))*v1_Count;

v2_Count = ((C_sm(2,1)/C_sm(1,1))*(A_smd(1,1)-
A_smd(2,2)))*v1_minus1+A_smd(2,2)*v2...
    +(C_sm(2,1)*k(1,1)+C_sm(2,2)*k(2,1))*w_tot_out(1)
+C_sm(2,2)*k(2,2)*w_tot_out(2);

v1_minus2 = v1_minus1;
v1_minus1 = v1_Count;

% Calculate v3:
Delta = inv([C_sm(3,3),C_sm(3,4);C_sm(3,3)/exp(-c*Ts_wind),C_sm(3,4)/exp(-
d*Ts_wind)]);

if Count < 3
    v3_Count = 0;
elseif Count >= 3
    temp = A_smd(3:4,3:4)*(Delta*([v3;v3_minus2]...
        +[0,0;C_sm(3,3)/exp(-c*Ts_wind),C_sm(3,4)/exp(-
d*Ts_wind)]*k(3:4,:)*w_tot_in...
        -C_sm(3,1)*[x_1_minus1;x_1_minus2]))+k(3:4,:)*w_tot_out;

    v3_Count=[C_sm(3,3),C_sm(3,4)]*temp+C_sm(3,1)*x_1;
end

v3_minus2 = v3;
Count_plus1 = Count + 1;

%% Assign Outputs
sys(1) = v1_Count;
sys(2) = v1_minus1;
sys(3) = v1_minus2;
sys(4) = v2_Count;
sys(5) = v3_Count;
sys(6) = v3_minus2;
sys(7) = Count_plus1;
sys(8) = w_tot_out(1);sys(9) = w_tot_out(2);sys(10) = w_tot_out(3);sys(11)
= w_tot_out(4);
% End of mdlOutputs.
```

A.4 Wind turbine controller s-function builder

This Simulink model builds the wind turbine controllers written in C code and generates a Mexw64 file for each wind turbine.

The controller.c and controller.h files must be saved in the directory of the S-Function builder model. The input and output ports of the controller must be set-up in the Builder GUI tool. The build button generates the Mexw64 library with the name specified in the S-function name field. For each wind turbine the S-function name must be different, controller1, controller2, etc.



S-Function Builder: PAC_generator/S-Function Builder

S-function name: controller1

Enter any library/object or source files used by the S-function. Then, specify any necessary include files or enter the external function declarations. These functions can be called in the Outputs, Derivatives an Update methods.

Library/Object/Source files (one per line)

```
Controller.c
```

Include files and external function declarations

Includes:

```
#include <math.h>
#include <Controller.h>
```

External function declarations:

```
/* extern double func(double a); */
```

Figure 12. 2: C-MEX builder

A.5 Wind farm controller

A Matlab leve2 S-function is used to develop the wind farm controller function. The wind farm controller can be modified to include different control objectives.

```
function FarmController2(block)

setup(block);
%%
function setup(block)

  block.NumDialogPrms     = 1;
  block.DialogPrmsTunable = {'Tunable'};
  block.NumInputPorts  = 9;
  block.NumOutputPorts = 6;

 %set up input output ports

  block.InputPort(1).Dimensions        = 1;
  block.InputPort(2).Dimensions        = block.DialogPrm(1).Data;
  block.InputPort(3).Dimensions        = block.DialogPrm(1).Data;
  block.InputPort(4).Dimensions        = block.DialogPrm(1).Data;
  block.InputPort(5).Dimensions        = block.DialogPrm(1).Data;
  block.InputPort(6).Dimensions        = block.DialogPrm(1).Data;
  block.InputPort(7).Dimensions        = 1;
  block.InputPort(8).Dimensions        = 1;
  block.InputPort(9).Dimensions        = 1;
  block.InputPort(1).DirectFeedthrough = false;
  block.InputPort(2).DirectFeedthrough = false;
  block.InputPort(3).DirectFeedthrough = false;
  block.InputPort(4).DirectFeedthrough = false;
  block.InputPort(5).DirectFeedthrough = false;
  block.InputPort(6).DirectFeedthrough = false;
  block.InputPort(7).DirectFeedthrough = false;
  block.InputPort(8).DirectFeedthrough = false;
  block.InputPort(9).DirectFeedthrough = false;
  block.InputPort(1).DatatypeID  = 0;  % double
  block.InputPort(2).DatatypeID  = 0;  % double
  block.InputPort(3).DatatypeID  = 0;  % double
  block.InputPort(4).DatatypeID  = 0;  % double
  block.InputPort(5).DatatypeID  = 0;  % double
  block.InputPort(6).DatatypeID  = 0;  % double
  block.InputPort(7).DatatypeID  = 0;  % double
  block.InputPort(8).DatatypeID  = 0;  % double
  block.InputPort(9).DatatypeID  = 0;  % double
  block.InputPort(1).Complexity  = 'Real';
  block.InputPort(2).Complexity  = 'Real';
  block.InputPort(3).Complexity  = 'Real';
  block.InputPort(4).Complexity  = 'Real';
  block.InputPort(5).Complexity  = 'Real';
  block.InputPort(6).Complexity  = 'Real';
  block.InputPort(7).Complexity  = 'Real';
  block.InputPort(8).Complexity  = 'Real';
  block.InputPort(9).Complexity  = 'Real';
  block.OutputPort(1).Dimensions       = block.DialogPrm(1).Data;
  block.OutputPort(1).DatatypeID  = 0; % double
  block.OutputPort(1).Complexity  = 'Real';
  block.OutputPort(2).Dimensions       = block.DialogPrm(1).Data;
  block.OutputPort(2).DatatypeID  = 0; % double
  block.OutputPort(2).Complexity  = 'Real';
block.OutputPort(3).Dimensions       = block.DialogPrm(1).Data;
  block.OutputPort(3).DatatypeID  = 0; % double
  block.OutputPort(3).Complexity  = 'Real';

  block.OutputPort(4).Dimensions       = block.DialogPrm(1).Data;;
  block.OutputPort(4).DatatypeID  = 0; % double
  block.OutputPort(4).Complexity  = 'Real';

  block.OutputPort(5).Dimensions       = 1;
  block.OutputPort(5).DatatypeID  = 0; % double
  block.OutputPort(5).Complexity  = 'Real';

  block.OutputPort(6).Dimensions       = 1;
  block.OutputPort(6).DatatypeID  = 0; % double
  block.OutputPort(6).Complexity  = 'Real';


  block.SampleTimes = [0.01 0];

  % methods
  block.RegBlockMethod('Outputs', @Outputs);
  block.RegBlockMethod('SetInputPortSamplingMode',
  @SetInpPortFrameData);
  block.RegBlockMethod('PostPropagationSetup', @DoPostPropSetup);
  block.RegBlockMethod('Start', @Start);
  %%
 function SetInpPortFrameData(block, idx, fd)

  block.InputPort(idx).SamplingMode = fd;
  block.OutputPort(1).SamplingMode  = fd;
  block.OutputPort(2).SamplingMode  = fd;
  block.OutputPort(3).SamplingMode  = fd;
  block.OutputPort(4).SamplingMode  = fd;
  block.OutputPort(5).SamplingMode  = fd;
  block.OutputPort(6).SamplingMode  = fd;
  %%
   function DoPostPropSetup(block)
  block.NumDworks = 11;

  block.Dwork(1).Name            = 'x1';
  block.Dwork(1).Dimensions      = block.DialogPrm(1).Data;
  block.Dwork(1).DatatypeID      = 0;      % double
  block.Dwork(1).Complexity      = 'Real'; % real
  block.Dwork(1).UsedAsDiscState = true;

  block.Dwork(2).Name            = 'x2';
  block.Dwork(2).Dimensions      = block.DialogPrm(1).Data;
  block.Dwork(2).DatatypeID      = 0;      % uint32
  block.Dwork(2).Complexity      = 'Real'; % real
  block.Dwork(2).UsedAsDiscState = true;

  block.Dwork(3).Name            = 'count';
  block.Dwork(3).Dimensions      = block.DialogPrm(1).Data;
  block.Dwork(3).DatatypeID      = 0;      % uint32
  block.Dwork(3).Complexity      = 'Real'; % real
  block.Dwork(3).UsedAsDiscState = true;
  block.Dwork(4).Name            = 'sum';
  block.Dwork(4).Dimensions      = 1;
  block.Dwork(4).DatatypeID      = 0;      % uint32
  block.Dwork(4).Complexity      = 'Real'; % real
  block.Dwork(4).UsedAsDiscState = true;

  block.Dwork(5).Name            = 'countgreen';
  block.Dwork(5).Dimensions      = block.DialogPrm(1).Data;
  block.Dwork(5).DatatypeID      = 0;      % uint32
  block.Dwork(5).Complexity      = 'Real'; % real
  block.Dwork(5).UsedAsDiscState = true;

  block.Dwork(6).Name            = 'countamber';
  block.Dwork(6).Dimensions      = block.DialogPrm(1).Data;
  block.Dwork(6).DatatypeID      = 0;      % uint32
  block.Dwork(6).Complexity      = 'Real'; % real
  block.Dwork(6).UsedAsDiscState = true;



  block.Dwork(7).Name            = 'countred';
  block.Dwork(7).Dimensions      = block.DialogPrm(1).Data;
  block.Dwork(7).DatatypeID      = 0;      % uint32
  block.Dwork(7).Complexity      = 'Real'; % real
  block.Dwork(7).UsedAsDiscState = true;



  block.Dwork(8).Name            = 'gmf';
  block.Dwork(8).Dimensions      = 1;
  block.Dwork(8).DatatypeID      = 0;      % uint32
  block.Dwork(8).Complexity      = 'Real'; % real
  block.Dwork(8).UsedAsDiscState = true;



  block.Dwork(9).Name            = 'amf';
  block.Dwork(9).Dimensions      = 1;
  block.Dwork(9).DatatypeID      = 0;      % uint32
  block.Dwork(9).Complexity      = 'Real'; % real
  block.Dwork(9).UsedAsDiscState = true;



  block.Dwork(10).Name            = 'rmf';
  block.Dwork(10).Dimensions      = 1;
  block.Dwork(10).DatatypeID      = 0;      % uint32
  block.Dwork(10).Complexity      = 'Real'; % real
  block.Dwork(10).UsedAsDiscState = true;

  block.Dwork(11).Name            = 'Delpold';
  block.Dwork(11).Dimensions      = block.DialogPrm(1).Data;
  block.Dwork(11).DatatypeID      = 0;      % uint32
  block.Dwork(11).Complexity      = 'Real'; % real
  block.Dwork(11).UsedAsDiscState = true;
  % Register all tunable parameters as runtime parameters.
  block.AutoRegRuntimePrms;
  %%
```

```
function Start(block)

    for i=1:block.DialogPrm(1).Data

    block.Dwork(1).Data(i) = 0;
    block.Dwork(2).Data(i) = 0;
    block.Dwork(3).Data(i) = 0;
    block.Dwork(5).Data(i) = 0;
    block.Dwork(6).Data(i) = 0;
    block.Dwork(7).Data(i) = 0;
    block.Dwork(11).Data(i) = 0;

    end

    block.Dwork(4).Data = 0;
    block.Dwork(8).Data = 1;
    block.Dwork(9).Data = 0;
    block.Dwork(10).Data = 0;
    DelP_Total_droop=10*5e6;
    for i=1:block.DialogPrm(1).Data

             block.OutputPort(1).Data(i)=0;
             block.OutputPort(2).Data(i)=0;
    end
%%
      function Outputs(block)

       droop =1;
       synthetic_inertia=0;
       DelP=zeros(block.DialogPrm(1).Data,1);
       DelPd=zeros(block.DialogPrm(1).Data,1);
       DelPdR=zeros(block.DialogPrm(1).Data,1);
       NTTT=zeros(block.DialogPrm(1).Data,1);



%% inputs
        totalPm=block.InputPort(1).Data;%add 100 sec wind speeds
        red=block.InputPort(2).Data;
        amber=block.InputPort(3).Data;
        green=block.InputPort(4).Data;
        PAC_ON=block.InputPort(5).Data;
        vest=block.InputPort(6).Data;
        demand=block.InputPort(7).Data;
        ROCOF=block.InputPort(8).Data;
        grid_freq=block.InputPort(9).Data;
        NT=block.DialogPrm(1).Data;

%% calculate the maximum avaiable DelP based on the flags

        Gn=find(green==1);
        An=find(amber==1);
        Rn=find(red==1);
        pg=length(Gn);
        pa=length(An);
        pr=length(Rn);
        Delp_green=0.5e6;

 %% frequency Droop control, calculation demand DelP based on the grid
frequency
```

```
DelP_Total_droop=5e7;
     dead_band=0.1;
     cont_band=1;
     maxpower=NT*0.25e6;%
     minpower=-NT*0.25e6;%;%NT*5e6/5;
     ref_freq= 50;
     freq=grid_freq-ref_freq;
%% ROCOF inertia control
DelP_Total=0;
DelP_ROCOF=0;
block.OutputPort(5).Data=0;

if freq~=0 && abs(ROCOF)>5

   synthetic_inertia=1;
   droop=0;
    H=18;%5.4;
    Sn=5e6;
    K=-2*NT*Sn*H/ref_freq;
    DelP_ROCOF=ROCOF*K;
    DelP_Total=DelP_ROCOF;
    priority=1;
    fast_slow=0;
    block.OutputPort(5).Data= DelP_ROCOF;

  for i=1:block.DialogPrm(1).Data

         if    PAC_ON(i)==0% in green region
                  DelPdR(i)=0.5e6;
                  NTTT(i)=1;

         else
            DelPdR(i)=0;
            NTTT(i)=0;
         end

  end
  DelP_ROCOF=2.5e6;
  DelP_Total_ROCOF=sum(DelPdR);
  DelP_Total=min( DelP_Total_ROCOF,DelP_ROCOF);
  block.OutputPort(6).Data=DelP_Total_ROCOF;
 end

%% synthetic inertia

if droop==1 %
    for i=1:block.DialogPrm(1).Data

         if   green(i)==1 && block.Dwork(9).Data==0 &&
block.Dwork(10).Data==0 && PAC_ON(i)==0% in green region

                 block.Dwork(8).Data=1;%gmf=1;
                 block.Dwork(9).Data=0;%amf=0;
                 block.Dwork(10).Data=0;%rmf=0;
                 DelPd(i)=0.5e6;
                 block.Dwork(5).Data(i)=0;
                 block.Dwork(6).Data(i)=0;
                 block.Dwork(7).Data(i)=0;
                 block.Dwork(11).Data(i)=DelPd(i);
```

```
        else if green(i)==0 && amber(i)==1 && block.Dwork(8).Data==1 &&
PAC_ON(i)==0% from green to amber

                block.Dwork(8).Data=1;%gmf=1;
                block.Dwork(9).Data=1;%amf=0;
                block.Dwork(10).Data=0;%rmf=0;
                block.Dwork(6).Data(i)=block.Dwork(6).Data(i)+1;
                DelPd(i)=max(block.Dwork(11).Data(i)-
(block.Dwork(6).Data(i)*10),0.5e5);
                block.Dwork(5).Data(i)=0;
                block.Dwork(7).Data(i)=0;
                block.Dwork(11).Data(i)=DelPd(i);

             else if  green(i)==1 && block.Dwork(9).Data==1 && PAC_ON(i)==0
%from amber to green

                 block.Dwork(8).Data=1;%gmf=1;
                 block.Dwork(9).Data=1;%amf=0;
                 block.Dwork(10).Data=0;%rmf=0;
                 block.Dwork(5).Data(i)=block.Dwork(5).Data(i)+1;

DelPd(i)=min(block.Dwork(11).Data(i)+(block.Dwork(5).Data(i)*10),0.5e6);
                 block.Dwork(6).Data(i)=0;
                 block.Dwork(7).Data(i)=0;
                 block.Dwork(11).Data(i)=DelPd(i);

                 else if  red(i)==1 && block.Dwork(9).Data==1 &&
PAC_ON(i)==0%from amber to red

                 block.Dwork(8).Data=0;%gmf=0;
                 block.Dwork(9).Data=1;%amf=0;
                 block.Dwork(10).Data=1;%rmf=1;
                 block.Dwork(7).Data(i)=block.Dwork(7).Data(i)+1;
                 DelPd(i)=max(block.Dwork(11).Data(i)-
(block.Dwork(7).Data(i)*10),0);
                 block.Dwork(6).Data(i)=0;
                 block.Dwork(5).Data(i)=0;
                 block.Dwork(11).Data(i)=DelPd(i);
%
                        else if  amber(i)==1 && block.Dwork(10).Data==1 &&
PAC_ON(i)==0%from red to amber

                 block.Dwork(8).Data=0;%gmf=1;
                 block.Dwork(9).Data=1;%amf=0;
                 block.Dwork(10).Data=1;%rmf=0;
                 block.Dwork(6).Data(i)=block.Dwork(6).Data(i)+1;

DelPd(i)=min(block.Dwork(11).Data(i)+(block.Dwork(6).Data(i)*10),0.2e5);
                 block.Dwork(5).Data(i)=0;
                 block.Dwork(7).Data(i)=0;
                 block.Dwork(11).Data(i)=DelPd(i);

                            end
                    end
                end
           end
       end

    end
    DelP_Total_droop=sum(DelPd);
    avail=DelP_Total_droop;

    percentage=1;
if (freq==0 && droop==1)

    DelP_Total=-avail* percentage;% avail demand power
    priority=0;
    fast_slow=0;
else if(freq<=dead_band && freq>= -dead_band && droop==1)
    priority=1;
    fast_slow=0;
    DelP_Total=-avail*
percentage;%demand;%(0.5*(maxpower+minpower));%avail;

else if(freq>=cont_band && droop==1)
    DelP_Total=minpower;%%negative

    priority=1;
    fast_slow=0;
else if(freq<=-cont_band && droop==1)
    DelP_Total=(maxpower);%%positive
    priority=1;
    fast_slow=0;
else if(freq>dead_band && freq< cont_band && droop==1)
DelP_Total= (((-avail* percentage)+avail)/-0.9)*(grid_freq-50.1)-(avail*
percentage);
priority=1;
    fast_slow=0;

else if (freq<-dead_band && freq>-cont_band && droop==1)
DelP_Total= ((avail+avail* percentage)/-0.9)*(grid_freq-49)+avail;
priority=1;
    fast_slow=0;
    else
        DelP_Total=-avail* percentage;%demand;
        priority=0;
        fast_slow=0;
    end
    end
    end
    end
    end

end



DelP_Total=min(2.5e6,DelP_Total_droop);
end
% synthetic_inertia=1;



%% dispatch power
if (block.CurrentTime>=500 && (synthetic_inertia==1||droop==1))
%% closed loop
   for i=1:block.DialogPrm(1).Data
```

```
       block.OutputPort(3).Data(i)=1; % Priority
       block.OutputPort(4).Data(i)=0; % fast/slow
     if synthetic_inertia==10 && PAC_ON(i)==0
%          droop=0;

         DelP(i)=DelP_Total*NTTT(i)/sum(NTTT);
         block.OutputPort(2).Data(i)=1; % PAC_ON

     else if droop ==1
%       DelP(i)=DelP_Total*DelPd(i)/DelP_Total_droop;

        block.OutputPort(2).Data(i)=1; % PAC_ON
         else
             DelP(i)=0;
             block.OutputPort(2).Data(i)=0; % PAC_ON
%
         end
     end

   end

       block.OutputPort(1).Data=DelP;


 else
            for j=1:block.DialogPrm(1).Data
            block.OutputPort(1).Data(j)=0;
            block.OutputPort(2).Data(j)=0;
            block.OutputPort(3).Data(j)=0;
            end
  end
```

A.6 Wind farm power Optimisation algorithm

This algorithm is used to generate optimal power set-point for the wind farm optimization control strategy in Chapter 9. This algorithm uses the Matlab fmincon function to generate the optimal power-set point in a static manner. The optimization algorithm runs for a number of iteration to obtain optimal power set-points that satisfy the optimization function.

```
function [history]=run
history.Power=[];
history.Pav=[];
history.CT=[];
history.CT1=[];
history.CP=[];
history.vv=[];
history.defii=[];

CT0=(0.8889).*ones(10,1); % initial CT values
lower=zeros(10,1);
upper=0.8889*ones(10,1);
options=optimset('LargeScale','off','MaxIter',100,'TolFun',1e-
20,'MaxFunEvals',100000,'Algorithm','sqp' ,...
   'Display','iter', 'TolCon',1e-1000,'TolX',1e-
1000,'PlotFcns',{@optimplotx,@optimplotfval});
% fmincon function
fmincon(@deficit,CT0,[],[],[],[],lower,upper,@confun,options);

function f=deficit(CT)

CT1=(0.8)*ones(10,1);

V=zeros(10,1);
CP=zeros(10,1);
Power=zeros(10,1);

beta=zeros(10,1);
W_D=zeros(10,1);
def=zeros(10,1);
dist=[0;800;1600;2400;3200;4000;4800;5600;6400;7200];
R=63;
D=2*R;
U0=10;
V(1)=U0;
W_D(1)=D;
def(1)=1;

k=0.5*1.255*pi*R^2;

for i=1:10
 beta(i)=(1+sqrt(1-CT(i)))/(2*sqrt(1-CT(i)));
 CP(i)=CT(i)*(1+sqrt(1-CT(i)))/2;
end
 history.CP=[history.CP;CP];
Power(1)=k*V(1)^3*CP(1);
```

```
for i=1:9
   W_D(i+1)=((beta(i+1)+0.5*dist(i+1)/D))^(1/2)*D;
   def(i+1)=1-(W_D(i)^2/W_D(i+1)^2*(1-
def(i))+0.5*D^2/W_D(i+1)^2*CT(i)*def(i));
    V(i+1)=U0*def(i+1);
   Power(i+1)=k*(V(i+1))^3*CP(i+1);
end
history.defii=def;
history.vv=V;
Pav=(((sum(Power)/10)+500000)*ones(10,1));
history.Power=[history.Power;Power];
history.Pav=[history.Pav;Pav];
history.CT=[history.CT;CT];
history.CT1=[history.CT1;CT1];
f=sum((Pav-Power).^2);
 end

end

%% optimisation constraint function used in the Matlab fmincon function
   function [c,ceq]=confun(CT)

V=zeros(10,1);
CP=zeros(10,1);
Power=zeros(10,1);

beta=zeros(10,1);
W_D=zeros(10,1);
def=zeros(10,1);
dist=[0;800;1600;2400;3200;4000;4800;5600;6400;7200];
R=63;
D=2*R;
U0=10;
V(1)=U0;
W_D(1)=D;
def(1)=1;
k=0.5*1.255*pi*R^2;
for i=1:10

 beta(i)=(1+sqrt(1-CT(i)))/(2*sqrt(1-CT(i)));
 CP(i)=CT(i)*(1+sqrt(1-CT(i)))/2;
end
 Power(1)=k*V(1)^3*CP(1);

for i=1:9
   W_D(i+1)=((beta(i+1)+0.5*dist(i+1)/D))^(1/2)*D;
   def(i+1)=1-(W_D(i)^2/W_D(i+1)^2*(1-
def(i))+0.5*D^2/W_D(i+1)^2*CT(i)*def(i));
    V(i+1)=U0*def(i+1);
   Power(i+1)=k*(V(i+1))^3*CP(i+1);

end
c=(sum(Power)/10)-5e6;
ceq=[];

    end
```