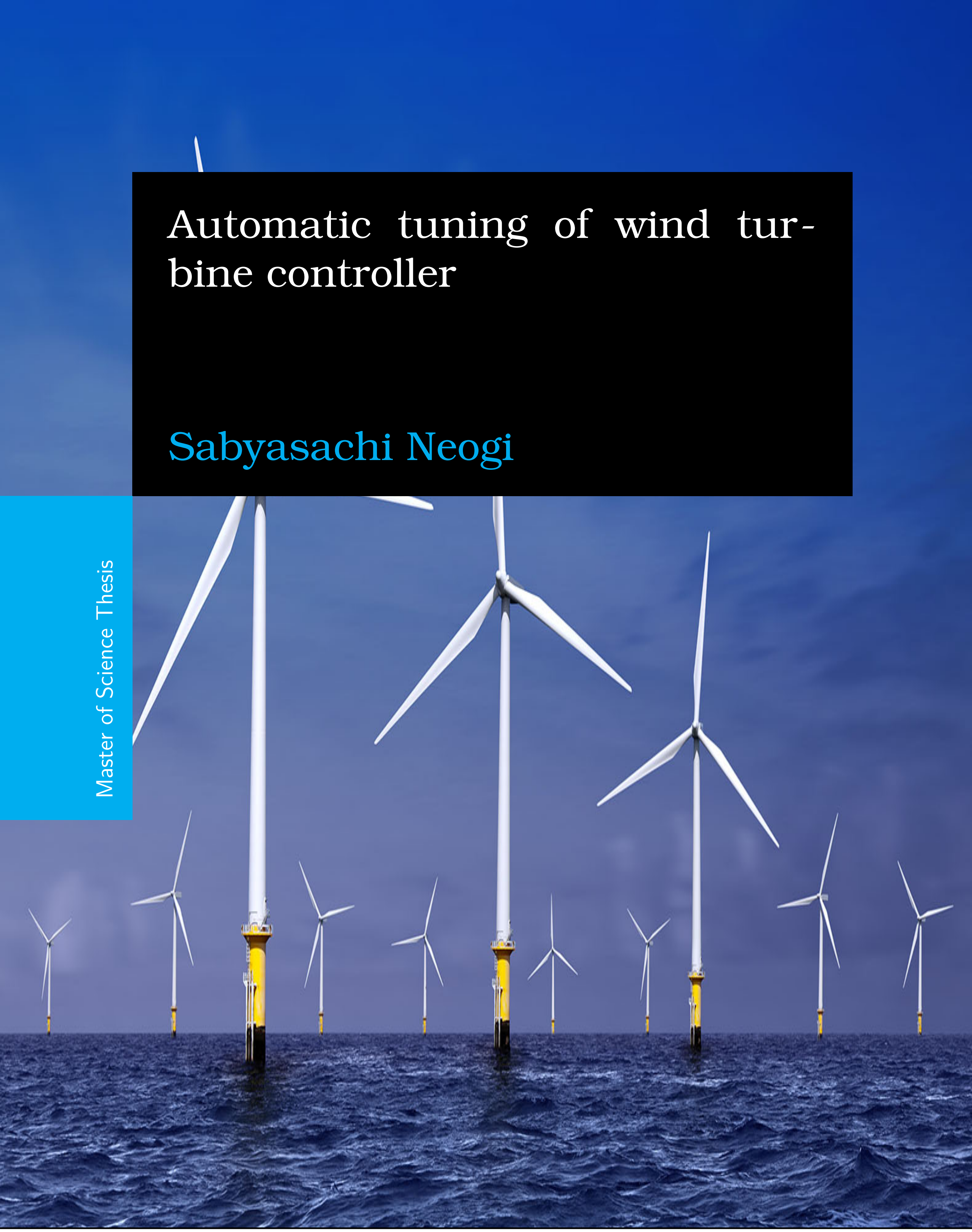


Automatic tuning of wind turbine controller

Sabyasachi Neogi

Master of Science Thesis

Delft University of Technology

Delft Center for Systems and Control

Automatic tuning of wind turbine controller

Master of Science Thesis

Sabyasachi Neogi

October 23, 2017

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

The work in this thesis was supported by Energy research Centre of the Netherlands . Their cooperation is hereby gratefully acknowledged.

Delft University of Technology
Department of

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

Automatic tuning of wind turbine controller

by

Sabyasachi Neogi

in partial fulfillment of the requirements for the degree of

Master of Science.

Dated: October 23, 2017

Supervisor:

Dr.ir Wouter Engels

Dr.ir Jan Willem Wingerden

Readers:

Dr.-Ing.J.Kober

Ir.S.Mulders

Abstract

The energy demand in current times has increased greatly in last few years. This increasing demand calls for a sustainable and clean energy resource that would reduce the load on non-renewable resources. Wind energy is a renewable resource which is harnessed by mankind from an ancient era. So as to meet this increasing energy demands, innovation in field of wind energy is required.

A wind turbine generates electricity but achieving optimal power is a difficult task. In this case the optimal power is defined as maximum power produced but at the constraints that the fatigue loading of the wind turbine structure should be as minimum as possible. Also, the wind turbine parameters such as rotor speed and pitch activity should be in a safe operational region. The problem in controlling wind turbines is that they work in a highly uncertain environment where managing so many factors at the same time is difficult. Optimal control of wind turbine has helped in achieving maximum power with-in safe working limits. Due to high uncertainty in the operating conditions of a wind turbine it is quite a daunting task to find optimal gains for a wind turbine controller.

This thesis focuses on achieving optimal gain parameters for wind turbine controller by using a algorithm from machine learning community. In this thesis the problem is formulated as a supervised learning problem where input-output mapping has to be predicted. For this purpose, Gaussian Process Regression Technique (GPRT) is used. The reason behind using GPRT is it takes fewer number of measurements to give good prediction compared to others. The property of GPRT where it deals with uncertain and non linear data with ease, making it a good choice for predicting wind turbine controller gains.

The second part of this thesis contains optimisation of the surrogate model achieved by performing regression. The optimisation is done by Monte Carlo Maximum distribution and improved results were generated by applying sequential sampling to this algorithm. This helps us to get a likelihood of optimal gains where the wind turbine gives out rated power with minimal fatigue loads, pitch activity and least deviation of rotor speed from rated.

The results obtained from the likelihood was tested for different operational wind speed and also tested for Extreme Operating Gusts as part of disturbance rejection and compared to current parameter used.
The comparison shows considerable improvement in the fatigue loads and pitch activity with having improvement in power production. In second case study, more parameters were predicted and optimised using the same algorithm so that the potential of this algorithm can be estimated. This was also performed successfully which proves that this technique can successfully be used to solve higher dimensional problems of wind turbine control.

Table of Contents

Acknowledgements **xi**

1 Introduction **1**

1-1 Wind Energy . . . 1
1-2 Offshore Wind Energy . . . 2
1-3 Motivation . . . 2
1-4 State of the art . . . 3
1-5 Problem Formulation . . . 4
1-6 Thesis Framework . . . 5

2 Model Description **7**

2-1 Wind turbine Model . . . 7
2-1-1 Aerodynamic modelling . . . 8
2-1-2 Structural modelling . . . 9
2-2 Baseline Controller Model . . . 10
2-2-1 State Estimator . . . 10
2-2-2 Controller design . . . 11
2-2-3 Shortcomings . . . 14

3 Theoretical Framework: Gaussian Process Regression and it's Optimisation **15**

3-1 Introduction to Gaussian Process . . . 16
3-2 Sparse and Online GP . . . 21

3-3 Gaussian Process Optimisation 21
3-3-1 Maximum Distribution 22
3-3-2 Sequential Monte Carlo Maximisation 23

4 Implementation 27
4-1 Measurements 27
4-1-1 Cost function 28
4-1-2 Measuring Noise 29
4-2 Regression 29
4-3 Gaussian Process Optimisation 31

5 Results 33
5-1 First case -Tuning with two inputs 33
5-1-1 Tuning for above rated case 33
5-1-2 Analysis with Near Rated Operation mode 37
5-1-3 Quantitative comparison 38
5-2 Second case - More tuning variables 39
5-2-1 Test function 39
5-2-2 On Wind turbine controller 43

6 Conclusions and Recommendations 45
6-1 Conclusions 45
6-2 Recommendations 47

A Appendix A 49
A-1 Time series analysis of above rated operational mode 50
A-2 Time series analysis for Extreme Operating Gust 55
A-3 Time series analysis at near rated Operational mode 60
A-4 Time series analysis with Extreme Event Controller 63

B Appendix B 69

Bibliography 71

Glossary 73
List of Acronyms 73
List of Symbols 73

List of Figures

1-1 Year wise cumulative Growth of Wind energy [1] 1
1-2 Year wise cumulative growth of Wind energy [1] 2
1-3 Different operational regions of a wind turbine 3

2-1 Representation of forces and their direction on wind turbine 9
2-2 drive train dynamics 10

3-1 Gaussian Process regression with different number of measurements 18
3-2 Variation in the signal variance (λ_f) and its affect on regression 19
3-3 Variation in the length scale (λ_x) and its affect on regression 20
3-4 Comparison of Maximisation algorithms 25

4-1 Comparison between magnitude of noise induced and cost values 29
4-2 Flow chart of Automated tuning 32

5-1 Prediction mesh for above rated tuning 34
5-2 Monte Carlo Maximum Distribution for above rated prediction 35
5-3 Optimisation using Sequential Monte Carlo (SMC) for above rated wind Speeds 35
5-4 Comparison of pitch activity between ECN optimum and optimum from likelihood under disturbance 36
5-5 Prediction mesh for near rated tuning 37
5-6 SMC optimisation plot for Near rated operational mode 38

5-7 True value v/s Predicted value 40
5-8 Comparing true value and predicted value for test data points 41
5-9 Maximum distribution using Monte Carlo Maximum Distribution (MCMD) along with true value plot 41
5-10 Maximum distribution using SMC along with true value plot 42
5-11 Particle Distribution with Sequential Monte Carlo Maximisation along with Predicted Mean Values 44

A-1 Pitch activity Comparison between ECN Optimum and Optimum from likelihood 50
A-2 Blade fatigue load comparison between ECN Optimum and Optimum from likelihood 51
A-3 Tower load fatigue comparison between ECN Optimum and Optimum from likelihood 52
A-4 Comparison of power production between ECN Optimum and Optimum from likelihood 53
A-5 Rotor speed comparison between ECN Optimum and Optimum from likelihood 54
A-6 Extreme Operating Gust used for testing 55
A-7 Blade fatigue load comparison between ECN optimum and optimum from likelihood under disturbance 56
A-8 Tower fatigue load comparison between ECN optimum and optimum from likelihood under disturbance 57
A-9 Comparison of generated power between ECN optimum and optimum from likelihood under disturbance 58
A-10 Comparison of rotor speed between ECN optimum and optimum from likelihood under disturbance 59
A-11 Blade fatigue load comparison between ECN Optimum and Optimum from likelihood for near rated case 60
A-12 Tower load fatigue comparison between ECN Optimum and Optimum from likelihood near rated case 61
A-13 Comparison of power production between ECN Optimum and Optimum from likelihood for near rated case 62
A-14 Pitch activity comparison between ECN Optimum and Optimum from likelihood for near rated case 63
A-15 Blade fatigue load comparison between ECN optimum and optimum from likelihood under disturbance Extreme event Control 64
A-16 Tower fatigue load comparison between ECN optimum and optimum from likelihood under disturbance Extreme event Control 65

A-17 Comparison of generated power between ECN optimum and optimum from likelihood under disturbance Extreme event Control 66
A-18 Pitch activity Comparison between ECN Optimum and Optimum from likelihood for Extreme event Control 67

List of Tables

5-1 Likelihood region for controllers gains for above rated operational mode . . 36

5-2 Likelihood region for controllers gains for near rated region. 38

5-3 Comparison of Below rated and above rated performance between current parameter and calculated parameter 39

5-4 Likelihood for an arbitrary function 42

5-5 Comparison Table for Parameter with EEC and without extreme event control 44

B-1 Information Table for ‘D4Rel Turbine’ 69

Acknowledgements

I would like to thank my supervisor Jan-Willem Wingerden to give the opportunity to carry out Master thesis under his supervision and helping me during the time period. Further, I would like to give special thanks my supervisor at ECN (Energy Research Center Netherlands). Wouter, thank you for selecting me to work under your supervision, your helpful suggestions and guidance that has helped me to complete the project and also helped me to become a better researcher. I would also like to thank Stoyan and Clym for helping me at times. Thanks to ECN for providing me with a friendly working environment.

I would like to thank Hildo Bijl for his quick responses to the emails to clarify my doubts on Sequential Monte Carlo.

On personal level, I would like to thank almighty God to keep me strong during the tough times and my parents without their motivation and support I couldn't have reached here.

I would also like to thank Ankit, Animesh, Divyam and Vimanyu and without them this journey wouldn't be this smooth. Finally, I would like to thank Priyanka for your relentless support and motivation during tough times.

Delft, University of Technology — Sabyasachi Neogi
October 23, 2017

Chapter 1

Introduction

1-1 Wind Energy

Energy plays a crucial role in development of humanity. From wood to coal and to renewable energy man has made transitions according to its needs. Growth in population and technological advancements created a huge energy demand. To solve this, wind energy caught an eye as its a clean and sustainable solution. Wind energy was historically used for various purposes like grinding grain, pumping water, pumps to drain polders. But nowadays wind energy is rapidly growing as a source of energy which has led to advancements in research in this field. By the end of 2016 the total installed wind power capacity world wide was around 4.86×10^5 MW and to give an idea of increase in wind power, the rise is almost 13% from 2015 [2].

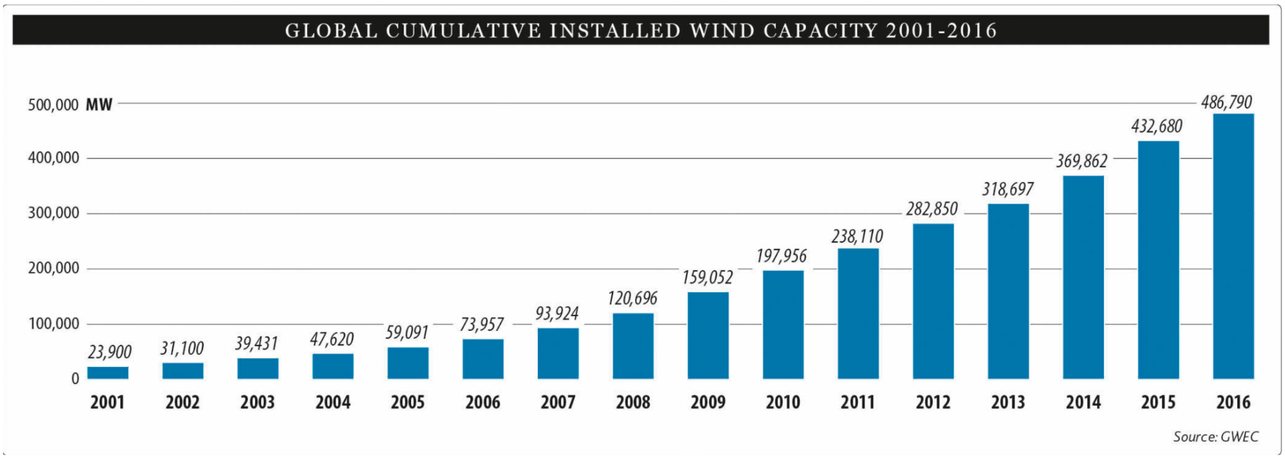


Figure 1-1: Year wise cumulative Growth of Wind energy [1]

This increase in power generation has also put up challenges such as designing more efficient wind turbines, safe operation of wind turbines, efficient transmission and much more. In last three decades, the wind turbine power production capacity has increased from 0.45 MW to 10 MW. Innovative ideas such as blade-less wind turbines, optimised blade designs, offshore vertical axis turbines and application of machine learning are some of the latest ideas which are currently being researched upon.

1-2 Offshore Wind Energy

An offshore wind energy is on an exponential rise. Only in EU a total of 43 GW is expected to be installed by 2020. The reason behind its growth is abundance of space, consistent wind resource and almost no opposition from the population. But offshore wind turbines are expensive due to complex logistics for installation, large structures and difficult grid connections [3].
Offshore wind turbine operates under very harsh conditions so their designs are kept relatively simple. Designing an optimal control scheme will help to reduce the fatigue loads along with an increase in power production which collectively will reduce the overall cost.

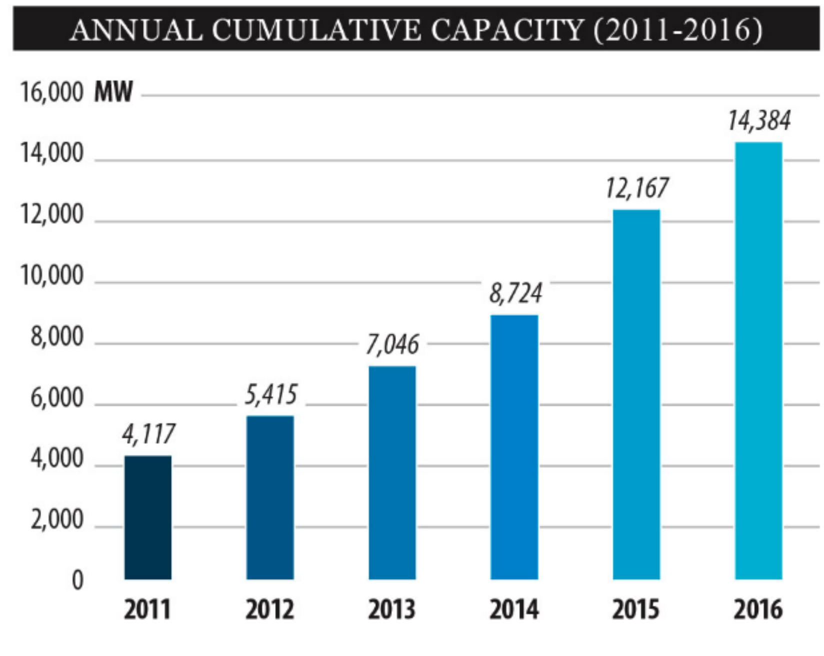


Figure 1-2: Year wise cumulative growth of Wind energy [1]

1-3 Motivation

To meet this increase in demand, production capability has to be increased. For this, size of wind turbines have increased rapidly as energy output is proportional to area of actuator disc [2].

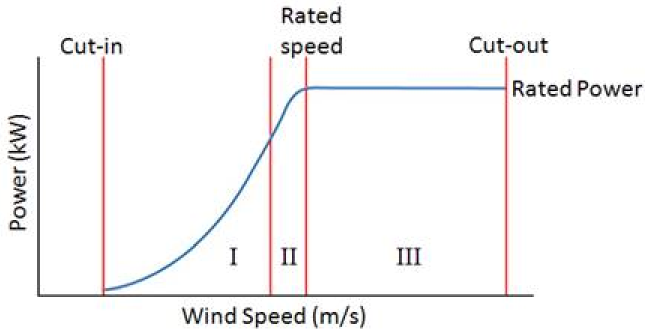


Figure 1-3: Different operational regions of a wind turbine

Levelized Cost of Electricity (LCOE) which is defined as sum of cost incurred over life time of the generating asset by sum of electrical energy generated in life time. For a wind turbine it is important to know about this factor as only generating maximum power is not the primary target, fatigue load reduction which plays important role in longevity of wind is crucial.
Wind turbine control is essential in reducing LCOE and better ways are to be looked out for so that power and load balance can be attained optimally.
Wind turbine is inherently a non-linear system which is constantly subjected to changing wind speeds. When the case is about offshore wind turbines then the conditions are harsher. A general guide to design a wind turbine control is to linearise the non-linear wind turbine model at a varying operating point and create a linear parameter varying model [4]. Then a parameter varying control is used using gain scheduling principles, sometimes along with H_∞ control principles to keep off the uncertainty LPV systems has [5].

1-4 State of the art

Control of wind turbine is a hard challenge because operation of wind turbine is dependent on the wind conditions, which is uncertain in behaviour. This change in operation with varying wind speeds bring us to different operation modes of a wind turbine.
Various operational regions for wind turbines are above cut-in speed, at rated speed and above rated speed (Figure 1-3). Each operational region has different control goals to achieve, so has different control actions. For example primary goal in above cut-in is maximising power production which is done by controlling generator torque, where as in above rated the goal is load reduction with using pitch manoeuvre techniques. These variations and large uncertainty in wind make it difficult to design a robust controller.
Some popular model based methods to design wind turbine controller are by using gain scheduling algorithm, by application of a robust controller or a pole placement technique.
In gain scheduling based controller, non-linear model is linearised around each operating point and then linear controller is designed for the linear system [5]. Gain scheduling is mostly applied in combination with Linear Parameter Varying (LPV) systems. LPV

systems are obtained by reformulating non-linear system as a linear system around the set of disturbances to the model [6]. For calculating optimal gain scheduling parameters, the problem is formulated as convex optimisation problem with help of Linear matrix inequalities (LMIs) [5].
For robust control grey box modelling is done where rigid body modelling followed by parametric identification for each components is done. Thereafter a H_∞ controller is applied to tackle the uncertainties and non-linear dynamics [5].
In pole placement, poles are forced in closed loop system so as to stabilise the system response. Here also non-linear model is linearised and gains of pitch and torque are tuned by pole placement method looking at the robustness of the design [7].
Data driven approach is also used to design controller. Where the first step is to perform system identification of wind turbine which is done by exciting the pitch system and the torque degrees of freedom, then an adaptive system identification of model is done to consider the case of constant variation of the system. An adaptive closed loop control algorithm is applied to the model to get the control parameter with varying system [8]. This technique does not require any kind of linearisation which makes the algorithm more reliable.
For optimal tuning, numerical optimisation techniques and Iterative feedback tuning methods are also used. In numerical optimisation technique a cost function is designed containing various parameters along with controller gain which is later optimised using a gradient based algorithm [9].
In iterative feedback control the control parameters are obtained iteratively by a gradient based method. The iterative optimisation is done by multiple closed loop systems which helps to estimate gradient with respect to a control parameter. This iterative updating of controller helps to minimise the user defined cost function [10]. The advantage of Iterative feedback tuning (IFT) is an in-depth knowledge of the system is not required but knowledge of internal stability of closed loop system is important [10].

1-5 Problem Formulation

So it can be observed from previous section that optimising a non linear cost function is a tough challenge. This brings us to a point where a new approach to achieve optimal gain needs to be formulated.
This brings us to an idea of data based tuning where prediction and optimisation is done in a model free manner which saves us from problems caused due to uncertainties in a wind turbine system.
The first step to perform this is to create data based cost function which is non linear in behaviour, due to the components such as damage equivalent loads, pitch activity and power produced which varying non-linearly. To mitigate this problem a self learning algorithm is sought for, which would learn the cost for different control parameters. The following parameters are obtained by performing aero-elastic simulation of wind turbine against different control parameters. Another reason to use a self learning it will predict uncertainties based on the uncertainty in current measurements.

We have a non linear cost function that varies with varying control parameters which are known to us, this motivates us to use supervised learning algorithm. In a supervised learning algorithm prediction of input-output mapping can be done based on a set of training data.

To perform this successfully, the ultimate goal has been broken down into smaller parts which collectively helps us to achieve the final goal :

To design a cost function which creates a trade off between wind turbine parameters that are essential for wind turbine control The measurement data is obtained from aero-elastic simulation an offshore wind turbine model of the 'D4Rel' turbine at a set of controller parameters. The controller is a closed loop negative feedback controller.
The data collected against a set of controller parameters after an aero-elastic simulation consists of parameters such as power generated, rotor speed, pitch angle and activity of all the three blades, fatigue loading of tower and blade, bending moments of blade and tower. The results also contain the standard deviation and mean values of these parameters. So, the first challenge that comes up is to design a cost function. The cost function should capture all the important factors of a wind turbine that are looked upon while designing a control algorithm and create a suitable trade off between them.

To apply learning algorithm to extrapolate number of control parameters from the measurements and predict the cost at these control parameters In the current system it is suggested that the model should be simulated for atleast 600 seconds for each set of controller parameters so as to get appropriate values for all the parameters. This is a considerably large time and limits the number of measurements.
So, for investigating gains and cost value it is required to have a set of large number control parameters. A learning algorithm is required that can interpolate or extrapolate the limited number of training data we have.

To create a suitable likelihood and achieve an optimal gain After obtaining a set of predicted gains and cost values corresponding to them, the task is to find the optimal gains for the designed cost function. For this, an optimisation scheme is required. The optimisation should be in two parts, first it should give out a likelihood of optimal gains and in the second part optimisation of likelihood is to be done to get the optimum out of the likelihood.

1-6 Thesis Framework

Chapter 2: Model Description To know and understand more about the model of wind turbine on which the experiment is performed. Details about the controller specifications and design that is being currently used to control this model. To know about the current tuning algorithm and the tuning parameters.

Chapter 3: Theoretical Framework This section contains description of the algorithm that is applied to reach our goal of achieving optimal gains without any manual interference. This section contains all the bits of theoretical aspects used to perform the automated tuning, interpolation and optimisation techniques.

Chapter 4: Implementation This section gives step by step pathway about how the algorithm has been applied to the model, what are the difficulties faced while applying the algorithm, intermediate results and how they were incorporated into the final results.

Chapter 5: Results In this chapter results are presented and its in-depth analysis is done.

Chapter6: Conclusions and Recommendations In the final chapter a conclusion is drawn from the results on the whole thesis work done. Also, some recommendations for future work is suggested.

Chapter 2

Model Description

The goal of this project is to automate wind turbine controller tuning. This experiment is performed on the 'Advanced Control Tool (ACT)' which is used for design and implementation of controller for wind turbines at Energy Centre Netherlands (ECN). The wind turbine used for simulation is a 'D4Rel turbine' whose specifications can be seen in appendix B. The controller consists of a baseline controller and a non linear model predictive controller that performs online optimisation of the baseline controller. The motive of this tool is to achieve some additional objective besides achieving baseline rotor speed and pitch angle, such as reduction fatigue loads in tower, active drive-train damping, fatigue load reduction on blades and optimal shutdown control minimising the tower loads during shut down [11].
The components in Advanced Control Tool (ACT) that facilitate it to get to the aforementioned goals

- A non-linear wind turbine model with control related wind turbine dynamics, for controller design and testing it.
- Observer for estimating states of wind turbine and stochastic nature of wind speed,
- A feedback based baseline controller which has trade offs to optimise rotor speed, power production and loads.d

Lets move on to discuss these components in details.

2-1 Wind turbine Model

The wind turbine model used in ACT is a non linear wind turbine model. The modelling has three parts structural, aerodynamic and wind modelling.

2-1-1 Aerodynamic modelling

Aerodynamic modelling is done in accordance with effects on each blade so that individual pitch action can be done with ease when required. The force and aerodynamic torques are defined as

$$\begin{aligned} F_{ax}^{(i)}(u_{ax}^{i}) &= F_{dyn}(u_{ax}^{i}) \cdot C_T(\lambda^{(i)}, \theta^{i})/B \\ T_a^{(i)}(u_{ax}^{i}) &= R \cdot F_{dyn}(u_{ax}^{i}) \cdot C_T(\lambda^{(i)}, \theta^{i})/B \end{aligned} \tag{2-1}$$

where F_{dyn} is force exerted on wind turbine, θ is pitch angle, λ is tip speed ratio, T_a^i is the generator torque, R is radius of wind turbine, u_{ax}^i is wind speed where i is each blade, B Number of blades. The wind turbine experiences forces and moments due to incoming wind are expressed as

$$\begin{aligned} \text{Flapwise moment}(M_{fl}^{(i)}) &= \frac{-2}{3} \cdot R \cdot F_{ax}^{(i)} \\ \text{Tilt moment}(M_{tilt}^{(i)}) &= \frac{2}{3} \cdot R \cdot F_{ax}^{(i)} sin(\psi^{(i)}) \\ \text{Tilt moment}(M_{yaw}^{(i)}) &= \frac{-2}{3} \cdot R \cdot F_{ax}^{(i)} cos(\psi^{(i)}) \\ \text{Sideward force}(F_{sd}^{(i)}) &= \frac{-2T_a^i}{R} sin(\psi^{(i)}) \end{aligned} \tag{2-2}$$

The direction of the forces follow right hand co-ordinate rule where the x-axis points down-wards, z-axis points downwards, $\psi^{(i)}$ denote azimuth position for i^{th} blade. To understand better a pictorial representation has been given in (Figure 2-1). The effect of abrupt change in pitch angle or wind is modelled by a simplified dynamic inflow model. In dynamic inflow model, effect of pitch angle variation on axial force per blade and aerodynamic torque per blade is modelled. Also, the pre-filtering of collective pitch angle using lag-lead filters is done to get filtered collective blade pitch angles. The coefficient of filters depend upon mean wind speed [11].

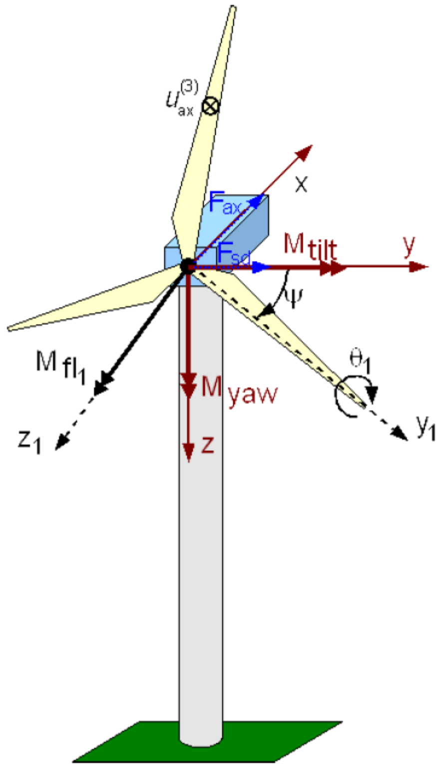


Figure 2-1: Representation of forces and their direction on wind turbine

2-1-2 Structural modelling

The structural dynamics of wind turbine is consists of three major components drive train dynamics, tower dynamics and wind. Each of them are modelled separately for ACT and then their impact on wind turbine is analysed.

Drive train modelling The slow shaft is modelled by assuming it as flexible, having torsional stiffness s_{dt} and damping d_{dt} to know it's frequency and damping. The gearbox is connected rigidly to nacelle, with transmission ratio (i_{tr}) which is positive when direction of the fast shaft rotor is same as slow shaft rotor else negative. The diagrammatic representation in figure (2-2) shows direction of torques and rotations acting on a drive train. The equations involved in drive train modelling

$$
\begin{aligned}
J_g\dot{\Omega}_g &= \frac{1}{i_{tr}}(T_{sh} - T_{loss}) - T_{gen}.sign(i_{tr}) \\
J_r\dot{\Omega}_r &= T_a - T_{sh} \\
T_{sh} &= s_{dt}\gamma - d_{dt}\dot{\gamma} \\
T_{loss} &= T_c + T_v\frac{1}{i_{itr}}\Omega_g + T_tT_{sh}
\end{aligned}
\tag{2-3}
$$

where J_r is the rotor inertia, J_g is generator inertia, Ω_g is the generator speed, Ω_r is rotor speed, T_{sh} is the shaft torque, γ is torsion angle, T_{loss} is model conversion losses,

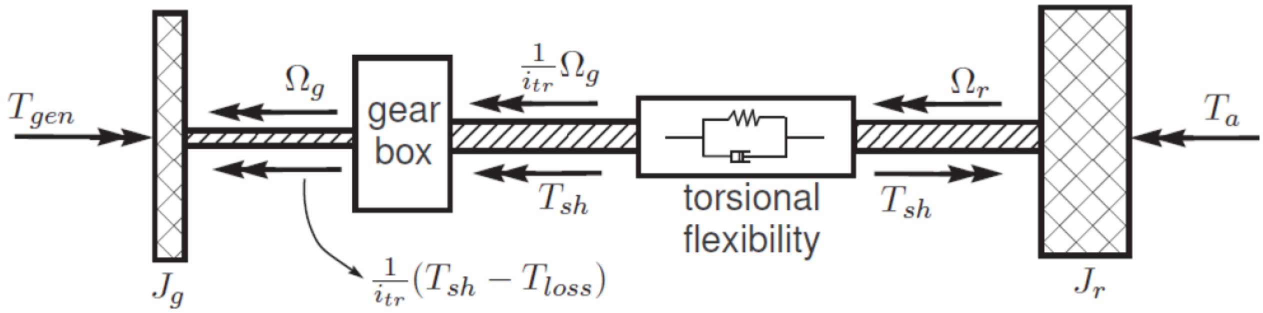


Figure 2-2: drive train dynamics

T_{gen} is generator torque, T_c and T_v are Coulomb and viscous forces respectively.

Tower top motion The tower top dynamics has two direction of dynamics fore-aft and sideways. Second order mass-spring-damper system model is used for modelling. The equation for tower fore-aft motion is

$$m_t \ddot{x}_{fa} + d_t \dot{x}_{fa} + s_t x_{fa} = \sum_{i=1}^{B} (F_{ax}^{(i)} - \frac{3}{2H} M_{tilt}^{(i)}) \tag{2-4}$$

where m_t is tower mass, d_t is damping and s_t is stiffness (these parameters are same for both the motions), x_{fa} is tower top position at fore-aft motion, M_{tilt}^i is tilting moment of tower, the force applied at the tower top is $\frac{3}{2H} M_{tilt}^i$. For side-ways motion

$$m_t \ddot{x}_{sd} + d_t \dot{x}_{sd} + s_t x_{sd} = \sum_{i=1}^{B} (F_{sd}^{(i)} + \frac{3}{2H} T_{nac}) \tag{2-5}$$

where sum of F_{sd}^i side ways force that excites the dynamics, F_{ax} is the axial force, x_{sd} is tower top position at sideways motion and T_{nac} is the effective torque acting on the nacelle.

Wind modelling The wind model used in ACT is takes a single wind signal that parameterizes the corresponding axial force and torque. The model consists stochastic wind speed signal containing slowly varying average wind and periodic components with frequency equal to multiples of rotors rotational frequency. It also has deterministic effects such as wind shear, tower shadow and tower fore-aft motion.

2-2 Baseline Controller Model

2-2-1 State Estimator

For achieving a feedback control, state estimation of the dynamics is required. The state estimation in ACT is done by observers. For estimating the states of linearised

model of tower, drive train and blade effective wind speed is used. A state estimation of linear structural dynamics is done using a Kalman filter, for this to be possible the available measurements of generator speed, pitch angle of each blade, tower aft and side ways acceleration are used to estimate axial force and torque.
The estimation has following steps, first the estimation of aerodynamic torque and drive train states are done, then wind speed is constructed and at last tower model states are estimated [11].

2-2-2 Controller design

Now the estimated states of wind turbine dynamics and wind will be used to design a baseline controller. This baseline controller will fulfils the objectives defined in the beginning of this chapter with help of several state feedback loops:

Power and rotor speed regulation loop The first control loop in the model is power and rotor speed regulation loop, the objective of this loop is to generate as much as power at below rated speed, limit power to rated power and to limit the fluctuation of rotor speed at above rated speed. To achieve maximum power capture below rated, λ and θ (collective) are set to optimum and Ω_r is kept controlled by generator torque. This is done by a non linear feedback law. The strategy used in power/rotor speed control loops are

- At a below rated condition when the $\Omega > \Omega_{r,min}$, the generator torque tries to control the rotor speed at $\Omega_{r,min}$ by means of PI control action on the error signal $(\hat{\Omega}_r - \Omega_{r,min})$. The generator torque is kept limited to $T_{gen}(\hat{\Omega}_r)$ that makes generator torque saturated at at very high wind speeds, where $\hat{\Omega}_r$ is estimated rotor speed.

- At the transition state, $\hat{\Omega}_r$ increase to $\frac{1}{2}(\hat{\Omega}_{r,rat} + \Omega_{r,min})$, the set point of PI torque controller is set to rated rotor speed and the upper limit of the torque is set to rated torque. When the $\Omega > \Omega_{r,min}$, T_{gen} will try to control rotor speed along with a proportional action on pitch to keep the pitch constant. But when the generator torque equals T_{gen}^{rat} then PI-pitch control is activated.

- The full load condition uses PI pitch action keeping the generator torque set at T_{gen}^{rat}

Closed loop for active reduction of tower fore-aft moment Reduction of tower fore aft moment is by means of collective pitch action control acting on the observer estimates of the tower-top position and speed in fore-aft directions. The estimation of tower top position and speed in foreaft and sidewards direction is done using Kalman filter. The linearised model of the tower fore-aft dynamics looks like

$$\begin{bmatrix}\delta\dot{x}_{fa}\\ \delta\ddot{x}_{fa}\end{bmatrix} = \begin{bmatrix}0 & 1\\ -\frac{s_t}{m_t} & -\frac{d_t+B\nabla_u F_{ax}(1+\frac{R^2}{2H^2})}{m_t}\end{bmatrix}\begin{bmatrix}\delta x_{fa}\\ \delta\dot{x}_{fa}\end{bmatrix} + \begin{bmatrix}0 & 0 & 0\\ \frac{B\nabla_\Omega F_{ax}}{m_t} & \frac{B\nabla_\theta F_{ax}}{m_t} & -\frac{RB\nabla_\Omega F_{ax}}{2Hm_t}\end{bmatrix}\begin{bmatrix}\delta\Omega_r\\ \delta\theta_{col}\\ \delta\theta_{tilt}\end{bmatrix} + \begin{bmatrix}0 & 0\\ \frac{B\nabla_u F_{ax}}{m_t} & \frac{RB\nabla_u F_{ax}}{2Hm_t}\end{bmatrix}\begin{bmatrix}\delta u_{sto,col}\\ \delta u_{sto,tilt}\end{bmatrix} \tag{2-6}$$

Closed loop for active reduction of tower side-ways moment The active reduction of the side-ways moment is done by generator torque control acting on the estimated tower top sidewards position and speed. The linearised model for tower side-ways moment look like

$$\begin{bmatrix}\delta\dot{x}_{sd}\\ \delta\ddot{x}_{sd}\end{bmatrix} = \begin{bmatrix}0 & 1\\ -\frac{s_t}{m_t} & -\frac{d_t}{m_t}\end{bmatrix}\begin{bmatrix}\delta x_{sd}\\ \delta\dot{x}_{sd}\end{bmatrix} + \begin{bmatrix}0\\ -\frac{B\nabla_u T_a}{Hm_t}\end{bmatrix}\delta\dot{x}_{fa} + \begin{bmatrix}0 & 0 & 0\\ -\frac{B\nabla_\theta T_a}{Rm_t} & \frac{3sign(i_{tr})}{2Hm_t} & -\frac{B\nabla_u T_a}{Rm_t}\end{bmatrix}\begin{bmatrix}\delta\theta_{tilt}\\ \delta T_{gen}\\ \delta u_{sto,tilt}\end{bmatrix} + \begin{bmatrix}0 & 0 & 0\\ \frac{3s_{dt}}{2H}\frac{i_{tr}-1+T_t}{m_t i_{tr}} & \frac{3d_{dt}}{2H}\frac{i_{tr}-1+T_t}{m_t i_{tr}} & \frac{3}{2H}\frac{T_\nu-d_{dt}(i_{tr}-1+T_t)}{m_t i_{tr}^2}\end{bmatrix}\begin{bmatrix}\delta\gamma\\ \delta\Omega_r\\ \delta\Omega_g\end{bmatrix} \tag{2-7}$$

This is more of a general model in the ACT, only the collective pitch action is applied for this project rather than having both individual pitch action and collective pitch action.

Current Tuning Process for Rotor speed and power regulation To achieve the gains of both the pitch and torque controller, at first the linearised drive train model is further modified so that PI controller can be designed by state feedback law. The optimum criterion is decided by a quadratic cost function for all the three operating regions, which brings LQR controllers into play. The linearised model to be used for the control is [11]

$$\begin{bmatrix}\delta\dot{\Omega}_r\\ \delta\dot{\Omega}_r^{int}\end{bmatrix} = \begin{bmatrix}\frac{(1-T_t)B\nabla_\Omega T_a(\bar{p})-T_v}{J_{eff}} & 0\\ 1 & 0\end{bmatrix}\begin{bmatrix}\delta\Omega_r\\ \delta\Omega_r^{int}\end{bmatrix} + \begin{bmatrix}\frac{-|i_{tr}|}{J_{eff}}\\ 0\end{bmatrix}\delta T_{gen} \begin{bmatrix}\frac{(1-T_t)B\nabla_\theta T_a(\bar{p})}{J_{eff}} & \frac{(1-T_t)B\nabla_u T_a(\bar{p})}{J_{eff}}\\ 0 & 0\end{bmatrix}\begin{bmatrix}\delta\theta_{col}\\ \delta T_{gen}\\ \delta u_{sto,col}-\dot{x}_{fa}\end{bmatrix} \tag{2-8}$$

where $\delta\Omega_r^{int}$ is the integrated rotor speed, $\delta\theta_{col}$ is the change in collective pitch angle, J_{eff} effective shaft dynamics, $\nabla_\theta T_a$ is the sensitivity of aerodynamic torque w.r.t pitch angle and $\nabla_\Omega T_a$ is the sensitivity of aerodynamic torque w.r.t rotor speed.
For each operational region different quadratic cost function formulation has been used

- Near cut-in region, only the torque controller is working to regulate rotor speed at rated rotor speed. As mentioned earlier, a LQR is designed on the model given in equation (2-8) with equilibrium rotor speed and equilibrium collective pitch angle with corresponding to the wind speed (u_{cutin}). The LQR design to compute PI-gains for feedback loop

$$\delta T_{gen} = \begin{bmatrix} K_p^{tq}(u_{cutin}) & K_I^{tq}(u_{cutin}) \end{bmatrix} \begin{bmatrix} \delta\Omega_r \\ \delta\Omega_r^{int} \end{bmatrix} \tag{2-9}$$

  As, this operation region is small so no gain scheduling was applied.

- In above rated region, the generator torque is kept at rated and Pitch PI-controller is used to regulate the rotor speed at rated. The pitch PI-controller has large operational region and aerodynamic torque varies quite a lot, to keep that in mind gain scheduling is done for wind speed (u_{high}) above rated apart from having the state feedback LQR. The equilibrium point for the LQR is $\bar{p}$=$\theta_{col}^{eq}(u_{high}), \Omega_{col}^{eq}(u_{high}), u_{high}$, the control law is

$$\delta\theta_{col} = \begin{bmatrix} K_p^{\theta}(u_{high}) & K_I^{\theta}(u_{high}) \end{bmatrix} \begin{bmatrix} \delta\Omega_r \\ \delta\Omega_r^{int} \end{bmatrix} \tag{2-10}$$

- In near rated has both the pitch and torque controller are working. The collective pitch action is done by a proportional controller. The torque controller gains are designed in a similar manner as done in above rated region .i.e. by calculation of K_p and K_i by solving a LQR design. The equation is same as equation (2-9) but the wind speeds are at near rated.

Rotor speed Filtering The estimated rotor speed has Bp and 2Bp frequency components from various components such as tower shadow, rotational sampling effects etc. These frequencies will result in unnecessary pitch and torque activity, so the $\hat{\Omega}_r$ is to be filtered before it goes into the Pitch and torque controller [11]. Bandstop filter around Bp and 2Bp frequencies is used, also a bandstop filter is applied around tower shadow and drive train frequency.

Gain Scheduling For above rated wind speed Gain scheduling is important, at constantly varying wind speeds as it operates at a wide range of wind speeds. This causes a high $\nabla_\theta T_a$ (sensitivity of aerodynamic torque to pitch angle) in (2-8), which means that small pitch angle change takes a large change in generator torque, leading to poor performance of pitch controller. To prevent this, a gain scheduling is done by calculating gain at a very high wind (u_{high}) speed and the scheduling it by gain at an estimated mean wind speed (u_{mean}). The gain factor is given by

$$\begin{aligned} g_\theta(u_{mean}) &\overset{\text{def}}{=} min\left\{ g_\theta^{max}, \frac{\nabla_\theta T_a(p_{eq}(u_{high}))}{\nabla_\theta T_a(p_{eq}(u_{mean}))} \right\} \\ p_{eq}(u) &\overset{\text{def}}{=} \{\theta_{col}^{eq}(u), \Omega_r^{eq}(u), u\} \end{aligned} \tag{2-11}$$

where the p_{eq}^{u} denotes the equilibrium operation curve. If seen closely $g_{\theta}(u_{mean})$ is inversely proportional to sensitivity of aerodynamic torque to pitch angle, this suggest that gain factor decreases with increasing mean wind speed. The g_{θ}^{max} is a factor kept to prevent from gain scheduling factor to become very high at close rated wind speed. The equilibrium operational curve in equation (2-11) is calculated assuming that the controller under steady state has power P_{rat} at high wind speeds, maximum power at rated wind speed u_{rat}.

For tower fore-aft moment fatigue load reduction To design the gain of the feedback controller, LQR design is applied to the linear top model (Equation 2-6). Some gain scheduling is also applied to cover the whole operational region. The final feedback law becomes

$$\delta\theta_{col}^{fa} = g_{fa}(u_{mean})K_{fa}(u_{high}) \begin{bmatrix} \delta\hat{x}_{fa} \\ \delta\hat{\dot{x}}_{fa} \end{bmatrix} \tag{2-12}$$

The estimation of tower top position and speed has some frequency components like Bp and 2BP. So, to avoid affects of these frequencies on pitch actuator bandstop filter is used. In addition to bandstop filter, highpass filter with a cutoff frequency much lower than the tower frequency is used to remove average from estimates of $\hat{x}_{fa}$ and $\hat{\dot{x}}_{fa}$.

For tower side-wards fatigue load reduction The state feedback gain is designed by LQR optimisation in a similarly as done for fore-aft fatigue load reduction. The linearised model (Equation 2-7) is used to design the controller.

2-2-3 Shortcomings

When this controller was simulated at a mean wind speed of 15 m/s, a better power yield is obtained with least fluctuation from rated rotor speed with a considerable reduction in tower bottom moment. As an estimate, over lifetime of 20 years damage equivalent load reduction of 6.5 % in fore-aft direction and 19% reduction in sidewards direction is achieved with a significant improvement in power production quality.
But it increase the pitch activity and ultimately a high pitch activity will give higher blade fatigue loads. It also uses a quadratic cost function to tune the controller gains which is a complicated task to perform due to non linearity involved.

Chapter 3

Theoretical Framework: Gaussian Process Regression and it's Optimisation

In introduction, explanation has been given about the difficulties faced in control of wind turbine and how capturing uncertainties is difficult while modelling wind turbine. Control algorithms such as H_∞, LQG and Gain scheduling which are being used predominantly in wind turbine cannot handle non-linear cost functions and require very detailed models to handle the non-white noise disturbances. So, this calls for need of a model free system of control where actual data from the model at varying conditions can be taken into consideration.

Machine learning approach for controller tuning The problem faced in our case study is an optimisation problem where we are trying to refrain from solving a non linear cost function. Also, model uncertainties needs to be mitigated to some extent due to stochastic behaviour of wind turbine.
To tackle this we try implement tools of machine learning which are quite popular as an optimisation tool. As it is possible to have the values of important factors for wind turbine control such as power produced, fatigue loads, rotor speed, pitch activity etc. at specific controller gains via aero-elastic simulation. To benefit from this feature the problem formulated in this thesis is a supervised learning problem, where prediction for input-output mapping is done. So, the data is analysed and more data is produced by inferring outputs from measured data points. The supervised learning consists of two types of problem: regression and classification [12]. Regression is a kind of supervised learning where prediction of continuous data is done from given training data [13]. The need of the problem is to get a model free approach for an uncertain and nonlinear system that self learns the non linearity and uncertainty to predict more data points in further space.

The aero-elastic simulation takes some time to perform the simulation that gives out the necessary values to be learned. This lead to look for an efficient method that can solve a regression problem with lesser training data. One of the solution for his type of problem is by 'kernel methods'.
In machine learning community 'kernel methods' are widely popular for tackling problems which has uncertainties and posses non linear dynamics which makes this method more useful in our case. Kernel methods are predominantly used in fields like geostatistics, cheminformatics, bioinformatics and handwriting recognition. Kernel methods include techniques such as Gaussian Process regression, support vector machines, kernel perceptron and many more. Kernel is a function of two arguments that are mapping a pair of inputs $x \epsilon \mathcal{X}$, $x' \epsilon \mathcal{X}$ into $\mathcal{R}$ [12].

3-1 Introduction to Gaussian Process

Regression models of Gaussian Processes are easy to implement, flexible and has Bayesian basis and thus a powerful tool in many application [14].
A Gaussian Process is a set of random variables which have a Gaussian joint distribution for any set of input x_n [15]

$$P(t|C, x_n) = \frac{1}{Z} exp(-\frac{1}{2}(t-\mu)^T C^{-1}(t-\mu)) \tag{3-1}$$

C is a covariance matrix defined by C(x_n,x_m;Θ) having Θ as hyperparameters and μ is mean function.
A simple Gaussian Process regression has two parts, a prior and a posterior distribution which are created by the joint distribution by predicted mean and variance [15].
The first step to perform Gaussian Process regression is to take measurements at random data points (x) in a space with each of them having output 'y'. These points can be said as measurements points (x_m) and consider that these measurements have noise to certain extent. The second step is to use this measurement to predict the values at random test data points in a space, this bit is called the regression.
The reason behind using GPRT is that it models the uncertainty present in the system efficiently while prediction is done. This property helps us to model the uncertainty in wind turbine behaviour. Also the ability of predicting non linear functions efficiently makes it a good choice for this problem.
As, mentioned before about the prior and posterior distribution, a short discussion about them is done.

Prior distribution Prior distribution is an assumption taken without performing any measurements. It fixes the properties of function considered, for example initial mean value or standard deviation or value region where measurements are going to be done [12] [16]. This assumption of prior is very important for the Bayesian formalisation of a

GP regression. Prior distribution function of f is given as probability density function

$$f \sim \mathcal{N}(f|m, \lambda_f^2) \tag{3-2}$$

where the Gaussian probability density function is given as

$$\mathcal{N}(f|m, \lambda_f^2) = \frac{1}{\sqrt{2\pi\lambda_f^2}} exp\left(-\frac{1}{2}\frac{(f-m)^2}{\lambda_f^2}\right) \tag{3-3}$$

Posterior distribution Before going to the posterior distribution, to show the expression for the joint distribution of the measurement points and the test points is important. This expression is basically mean and covariance between all the points (test and measurement) in the space. The joint distribution with noisy observations is given in Rasmussen *et.al* [12]

$$\begin{bmatrix} y \\ \underline{f}_\star \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(X) \\ m(X_\star) \end{bmatrix}, \begin{bmatrix} K(X,X)+\sigma^2 I & K(X,X_\star) \\ K(X,X_\star) & K(X_\star,X_\star) \end{bmatrix}\right) \tag{3-4}$$

where Y is the output of the measurements, $X_\star$ is set of trail data points, $\underline{f}_\star$ is output of the test points, K is the covariance matrix, σ is the noise covariance matrix, m is the mean of the both sets of data points values. For a non-noisy measurement the σ^2 value is set to zero and for a zero initial mean condition the prior mean functions are set as zero.
The σ^2 gives the uncertainty of the measurements giving the likelihood in the probability distribution [12]. The prior and the likelihood with the marginal likelihood, altogether can be combined using Bayesian formalism to give out the posterior distribution.

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}, p(f_\star \mid y, X) = \frac{p(y|X, f_\star)p(f_\star)}{p(y \mid f_\star)} \tag{3-5}$$

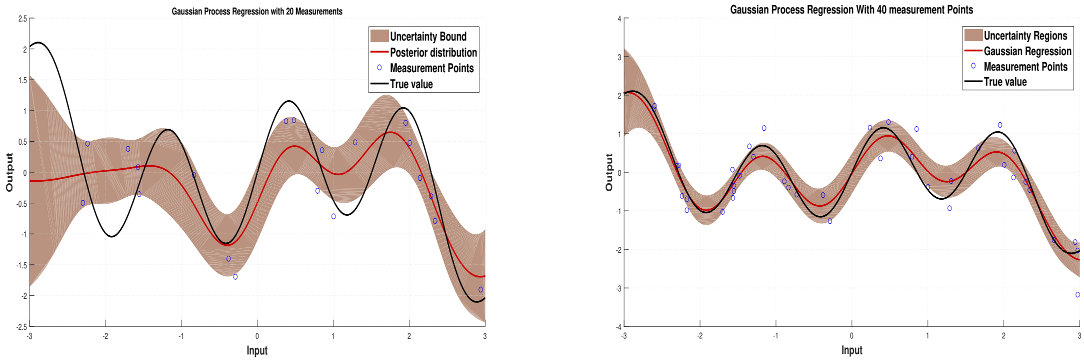
The expression for the posterior distribution for noisy measurements is given by $\mathcal{N}(\mu_\star, \Sigma_\star)$ which is expressed as:

$$\begin{aligned} \mu_\star &= K(X_\star, X)[K(X,X)+\sigma^2 I]^{-1} y \\ \Sigma_\star &= K(X_\star, X_\star) - K(X_\star, X)[K(X,X)+\sigma^2 I]^{-1} K(X, X_\star) \end{aligned} \tag{3-6}$$

These terms are valid for zero mean initial condition, for non zero initial mean term y is replaced by $(y - m(X_m))$. The posterior distribution gives us an estimate of distribution of the test set $x_\star$. This helps us to create a regression in the space where the desired test data points are chosen from. The complete GP regression which is obtained by merging prior distribution eq. (3-7) and the distribution of measurements will be

$$\begin{bmatrix} y \\ \underline{f}_\star \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu(X) \\ \mu(X_\star) \end{bmatrix}, \begin{bmatrix} \Sigma(X,X)+\sigma^2 I & \Sigma(X,X_\star) \\ \Sigma(X,X_\star) & \Sigma(X_\star,X_\star) \end{bmatrix}\right) \tag{3-7}$$

where
$\begin{bmatrix} \Sigma(X,X)+\sigma^2 I & \Sigma(X,X_\star) \\ \Sigma(X,X_\star) & \Sigma(X_\star,X_\star) \end{bmatrix}$ is the posterior covariance matrix which contains pre-



(a) Posterior distribution with 20 measurements **(b)** Posterior distribution with 40 measurements

Figure 3-1: Gaussian Process regression with different number of measurements

dicted variance between test and measurement points. $\begin{bmatrix} \mu(X) \\ \mu(X_\star) \end{bmatrix}$ is the posterior mean matrix containing predicted means of measurement and test points. So, by having this formulation we can calculate the posterior distribution of the test data points.
It is important to have the noises because it helps us to know the amount of uncertainty our posterior distribution will have, giving clarity about the process. For our project it's important to consider uncertainty, without considering it the uncertainty in posterior distribution won't be there. So, not considering uncertainty while calculation posterior distribution will miss out on quite some amount of data that would lead to failure in learning uncertainty along with non-linear variation in data will not be possible.
For improving a posterior distribution, more and more measurement points should be involved, by doing that the estimate of real function would be better refer to (Figure 3-1).

Example Simulations To show how this algorithm works, an example non linear function is taken for measurements and then further regression is performed for it. The function chosen is

$$f(x) = sin(4x) - \frac{x^3}{10} + \frac{2x}{5} \tag{3-8}$$

Some degree of noise is also added to the measurements. All the figures in this chapter are obtained by the measurements done using this function.

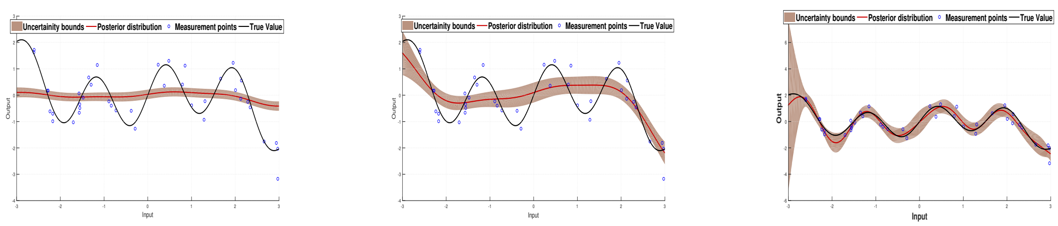
Covariance function Covariance function helps to mimic the kind of process which one wants to learn or create a model off. To large extent, they determine how the distribution will be, for example a stationary and smooth distribution can be achieved by using stationary covariance function.
In GP regression while performing the prior distribution term 'K' was introduced which is a covariance matrix. This is also called a kernel which helps to create relation between points in the space. This makes covariance function a very important component of Gaussian Process regression.

There are number of covariance functions and each have their own merits and demerits. So, being an important factor in creating a correct posterior distribution the choosing correct covariance function is crucial. In this project the kernel chosen is *squared exponential* as it is infinitely differentiable and widely used. It also gives very satisfactory results in our case. It is expressed as

$$k(x, x') = \lambda_f^2 exp\left(-\frac{1}{2}\frac{(x-x')^2}{\lambda_x^2}\right) \tag{3-9}$$

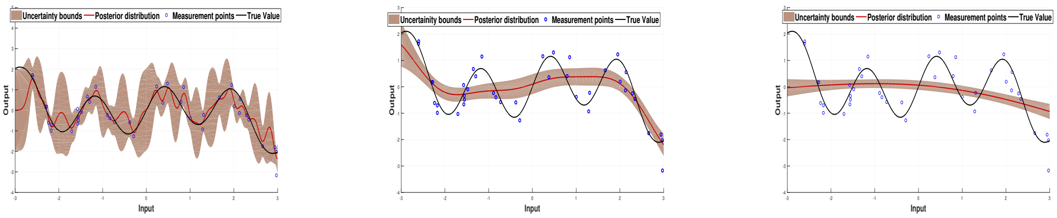
where λ_f is output variance, λ_x is length scale. Other parameter that decides the behaviour of covariance function is noise variance σ_n and $\bar{m}$ initial mean, these four parameters collectively are called hyperparameters. Tuning each of them changes the posterior distribution which is also shown in (Figures 3-2) and (3-3).

Hyperparameter tuning As mentioned in previous paragraph, covariance function consists of some parameters which basically govern the results we achieve from a covariance function. Covariance function being the most essential part in creating the GP regression, makes tuning hyperparameters important as tuning them will certainly affect in learning the non linear function. Generally, hyperparameters are collectively denoted as 'θ'.
Before moving to the tuning part let's look at how these hyperparameters affect the regression.
The length scales gives us an idea of the distance in the input space after which the data becomes uncorrelated to the function value [12], which be seen in (Figure 3-3). In a casual manner, the smaller the length scale the data points are more correlated and creates a distribution that's more crumbled and spiky whereas bigger the length scale the posterior becomes independent of input data.
The signal variance (λ_f) is a scaling factor where it determines how close or far the function predicted is from the mean. Lower the signal variance is low variation from mean it will have and a higher value has a higher variation, can be seen in (Figure 3-2).
Hyperparameter tuning is commonly known as Automatic relevance determination (ARD).



(a) Hyperparameters (λ_f,λ_x,σ_n)=(0.1,0.6,0.4) **(b)** Hyperparameters (λ_f,λ_x,σ_n)=(1,0.6,0.4) **(c)** Hyperparameters (λ_f,λ_x,σ_n)=(10,0.6,0.4)

Figure 3-2: Variation in the signal variance (λ_f) and its affect on regression



(a) Hyperparameters $(\lambda_f,\lambda_x,\sigma_n)$=(1,0.1,0.4) **(b)** Hyperparameters $(\lambda_f,\lambda_x,\sigma_n)$=(1,1,0.4) **(c)** Hyperparameters $(\lambda_f,\lambda_x,\sigma_n)$=(1,5,0.4)

Figure 3-3: Variation in the length scale (λ_x) and its affect on regression

Initially the hyperparameters are completely unknown, so an initial estimate has to be made, the most common estimate used is $\theta_i \sim (0,1)$ [15].
A widely used technique to tune hyperparameter is by 'Maximum A Posteriori method', which is a Bayesian optimisation based method. Here we like to maximise the posterior hyperparameter distribution. The posterior over the parameter by Bayes rule is

$$p(\theta|\hat{f}_m, X_m) = \frac{p(\hat{f}_m|\theta, X_m)p(\theta|X_m)}{p(\hat{f}_m|X_m)} \tag{3-10}$$

where X_m is measured data points (the known data), $\hat{f}_m$ is function values, θ is hyperparameters, $p(\theta|\hat{f}_m, X_m)$ is posterior hyperparameter distribution, $p(\hat{f}_m|\theta, X_m)$ is the likelihood, $p(\theta|X_m)$ prior values and $p(\hat{f}_m|X_m)$ is the marginal likelihood.
The expression which needs to be maximised is the logarithm of this a posteriori distribution which is

$$log(p) \propto -\frac{n_m}{2}log(2\pi) - \frac{1}{2}log|K_{mm} + \hat{\Sigma}_{f_m}| - \frac{1}{2}(\hat{f}_m - m_m)^T(K_{mm} + \hat{\Sigma}_{f_m})^{-1}(\hat{f}_m - m_m) \tag{3-11}$$

the reason for taking log of the distribution is increasing nature of logarithm.
Now to maximise this expression optimum choice of the hyperparameters should be made. To get the optimal hyperparameter a gradient based algorithm is used where gradient of the expression with each hyperparameter is calculated, and then parameters that maximise the gradient are chosen (gradient ascent). Some manual adjustment are also applied on top the gradient ascent method to create a better fit and maximise the log-likelihood value of a posterior distribution.
The disadvantage of performing Gaussian Process regression is it's high computaion cost and runtime. When the data points involved are large in number then it becomes troublesome to apply Gaussian Process.

Computational complexity The computational complexity of a regular Gaussian Process regression takes $\mathcal{O}(n_\star n_m^2 + n_m^3)$ for mean and for covariance $\mathcal{O}(n_\star^2 + n_m^2 n_\star^2 + n_m^3)$ where n_m is number of measurements, $n_\star$ is the number of test points. So, if the number of measurement data points are in range of 1000 then it'll be a huge runtime.
The reason for high runtime comes from the inversion of the $(K_{mm} + \Sigma_{fm})$ matrix which is $n_m \times n_m$ in dimension and inverting it takes $\mathcal{O}(n_m^3)$.

3-2 Sparse and Online GP

Sparse GP regression To tackle this high computational requirement and runtime Bijl *et.al* [17] suggested Sparse GP regression. This algorithm splits the regression in two parts to reduce the computational requirement. In first part some points are induced (inducing points(n_u)) and a prior joint distribution between them and measurement points are done. Then for the posterior distribution the normal GP regression is used as in equation (3-6), this step is called the training set.
The next step is the prediction step, here the prior distribution of $\underline{f}_\star$ and $\underline{f}_u$ is merged by a joint distribution with help of the calculated values of μ_u and Σ_{uu}. The sparse GP prediction equation for μ_{uu} and $\Sigma_{\star\star}$ are

$$\begin{aligned}\Sigma_{\star\star} &= K_{\star\star} - K_{\star u}K_{uu}^{-1}(K_{uu} - \Sigma_{uu})K_{uu}^{-1}K_{u\star} \\ \mu_\star &= m_\star + K_{\star u}K_{uu}^{-1}(\mu_u - m_u)\end{aligned} \tag{3-12}$$

The problem here which needs special attention while implementing this algorithm is $\underline{f}_u$ has been merged twice as it has taken part in joint distribution. So, it is necessary that $\begin{bmatrix}\underline{f}_u \\ \underline{f}_\star\end{bmatrix}$ is unmerged. The computational time for sparse regression is $\mathcal{O}(n_u^2 n_\star^2)$.
The choice of induced point is done manually such that it is sufficiently spread between measurement points and test points [16]. The reason behind having induced points between the measurement points and test points is to have better training from measurements and better prediction by induced points to the test points [16].

Online GP regression Another notion is to apply new points (suppose (x^+, f^+)) in the current regression without full re-calculation, which is done by online GP regression. But the Online GP regression has a high run time and computational requirement, so this concept is combined with sparse regression in Bijl *et.al* [16] and Sparse and online GP regression is used.
The combined sparse and online GP regression are with assumption that the inducing function values f_u and measurement function values f_m are completely independent. This assumption is also called FITC (Fully Independent Training Condition) initially given in Bijl *et. al* ([17]). The posterior mean and variance expressions are

$$\begin{aligned}\Sigma_{uu}^+ &= \Sigma_{uu} - \Sigma_{u+}(\Sigma_{++} + \hat{\sigma}_{f_+}^2)^{-1}\Sigma_{+u} \\ \mu_u^+ &= \mu_u + \Sigma_{u+}(\Sigma_{++} + \hat{\sigma}_{f_+}^2)^{-1}(\hat{f}_+ - \mu_+)\end{aligned} \tag{3-13}$$

here the superscript '+' is to indicate posterior distribution of test points in account of online measurements (x^+, f^+).

3-3 Gaussian Process Optimisation

Till now, the discussion was on Gaussian Process regression and its variants which were being used to extend the values of a nonlinear and noisy function. But this regression

needs to be optimised to get a likelihood of meaningful data from the distribution.
Gaussian optimisation is quite different from optimising a normal function, as in normal function we've idea that the output is from certain kind of function but here the function itself is predicted and contains uncertainty. So, rather than going for calculating a specific optimal point an optimal distribution is looked out for.

3-3-1 Maximum Distribution

The first part in this optimisation is to calculate the maxima of this joint posterior distribution. As Gaussian Process is distribution of random variables over the space, so it's maxima is not also fixed point rather it's a distribution of variables. The challenge is some what different from calculating an optimum point rather a maximum distribution is looked for.

Monte Carlo Maximisation Bijl *et.al* [16] used a Monte Carlo technique to create a maximum distribution. In a normal Monte Carlo technique future expectation is done on basis of samples of current mean and variance.
In Monte Carlo Maximum Distribution (MCMD) the goal is to find the part of Gaussian distribution where the function value will be maximum.
The terminology used in Bijl *et.al* [16] is a bit different, it calls the sample of points as *particles*, and the comparisons as *challenges* and each input data point where particles are stored is called *bin*. The current particles x_i is called *champion* and the random sample picked for x_j is called *challenger*. The number of particles used is denoted by n_p.

The steps to algorithm are

1. Divide all the particles (randomly or evenly) in 'n' bins so that at least there is one particle in each bin
2. Now n_p comparisons have to be done so each comparison should be in following manner
   - For a particle choose the corresponding bin (champion) say x_i
   - Pick a random bin (challenger) x_j
   - Set up a joint distribution of $\underline{f}(x_i)$ and $\underline{f}(x_j)$ like this

   $$\begin{bmatrix} \underline{f}(x_i) \\ \underline{f}(x_j) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} f_i \\ f_j \end{bmatrix} \mid \begin{bmatrix} \mu(x_i) \\ \mu(x_j) \end{bmatrix}, \begin{bmatrix} \Sigma(x_i, x_j) & \Sigma(x_i, x_j) \\ \Sigma(x_j, x_i) & \Sigma(x_j, x_j) \end{bmatrix} \right)$$

     and take samples from these distributions $\hat{f}_i$ and $\hat{f}_j$
   - If $\hat{f}_j > \hat{f}_i$ then move the particles to random bin x_j.
3. This procedure of comparison, moving and challenging is done till all the bins are swept through for a certain number of iterations (n_r) and until convergence of particles is observed.

This algorithm helps us to get an idea of the location of the maximum distribution but the problem this algorithm has is it does a comparison of samples in a distribution which are chosen smartly after every iteration. This makes the algorithm computationally demanding and at times it doesn't converge to the true maximum. So, for improved performance a modified version of Monte Carlo maximisation is used.

3-3-2 Sequential Monte Carlo Maximisation

Sequential Monte Carlo (SMC) algorithm is creating the maximum distribution incorporating sequential sampling techniques in Monte Carlo maximisation technique. This helps to make the process considerably fast and converge to the true maxima better than the MCMD algorithm. There are three major components that are being added so as to make the Monte Carlo maximisation algorithm faster and correct. These are Systematic Resampling, Importance Sampling and Defensive Sampling [18]. In (Figure 3-4), the number of simulations made for a Monte Carlo Maximisation is three times as the number of simulations done for Sequential Monte Carlo and the results are almost the same for particle distribution. This shows the superiority of Sequential Monte Carlo over Monte Carlo Maximisation in terms of run time and correct convergence.

Adding weights First, *importance sampling* is done. The reason behind doing this is to choose samples from the region were the amount of data is not sufficient but it is important to have that data as they are 'rare events'. So for this purpose a weight is added in that region of rare events. A distribution that has more of these 'rare events' $q(x_k)$ is chosen. The weight to be updated is given by $\frac{p(x_k)}{q(x_k)}$ also called likelihood ratio. $p(x_k)$ is the nominal distribution we had initially. Self normalisation is also done so that addition of weights don't disturb the convergence to the true values and divided by same value. The updated distribution looks like

$$p_m(x) = \frac{\sum_{i=1}^{n_p} w^i \delta(x, x^i)}{\sum_{i=1}^{n_p} w^i} \tag{3-14}$$

where $\delta(x, x^i)$ is a dirac delta function. Initially the function p_m denotes the number of samples at x^i for an input x, but after a few comparison this is the belief of maximum distribution (p_m). We also select the next challenger from this distribution as it makes our convergence fast.

Resampling This is done after first round of challenges are done. Now the different samples have different weights and it's required to remove the small and zero weights, thus assigning a common weight of 1 to non zero weights.
The reason behind performing resampling is to remove 'Weight degeneracy'. Weight degeneracy occurs when some weights are very small and some are very high, so the smaller weights are removed and new samples are chosen near higher weights. The need for removing smaller weight comes as they don't have any important information.
This is done by 'systematic resampling'. As described by Bijl *et.al* [18], In systematic

resampling cumulative sum of weights are taken which is further divided into number of blocks. Then a single random sample is chosen from first block and remaining block are kept as it is.

Correct Convergence As in Monte Carlo maximum distribution, maximum distribution has tendency to deviate from true maxima, this compromises the accuracy of the convergence. To prevent this, defensive sampling is done, where a new distribution is used for sampling of challengers which consists of data from another distribution $q(x)$ along with the maximum distribution $p_m(x)$. This new distribution looks like

$$q'(x) = \alpha p_m(x) + (1-\alpha)q(x) \tag{3-15}$$

This also makes us update the weights used for selecting the challengers. In this expression what happens is for α times the challenger is sampled from $p_m(x)$ (eqn. 3-14) that gives a random champion and part $(1-\alpha)$ a completely random challenger. The updated weight expression for a challenger looks like

$$w_c = \frac{q(x_c)}{q'(x)} \tag{3-16}$$

where x_c denotes a challenger. But this expression when used to calculate the weight will slow down the algorithm as $p_m(x)$ in equation (3-14) calculates through all samples. So, just to update a single weight $p_{max}(x_c)$ is done, which is redundant.
To remove this problem the idea used is calculate the random distribution after having the random champion in the distribution. So, what is done is rather calculating whole $p_m(x)$ again we select the champion (x^i) beforehand and then the random particle. The $q'(x)$ looks like

$$q'(x|x^i) = \alpha\delta(x, x^i) + (1-\alpha)q(x) \tag{3-17}$$

After combining equations 3-16, 3-17 and 3-14 the final challenger weight is

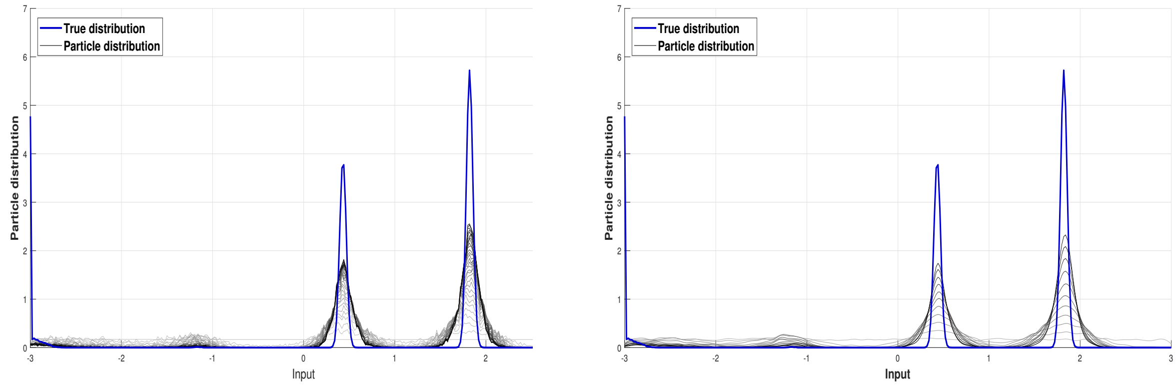
$$w_c = \frac{q(x_c)}{q'(x_c|x^i)} = \frac{q(x_c)}{\alpha\delta(x_c, x^i) + (1-\alpha)q(x_c)} \tag{3-18}$$

Application in continuous functions In previous discussions we used Dirac delta function for calculation of $p_m(x)$ but it's known that for a continuous function we calculate a probability density function. So, the $p_m(x)$ is calculated using a 'Kernel Density Estimation (KDE)' function. The KDE used is a Gaussian kernel. The Gaussian kernel looks like

$$k(x, x') = \frac{1}{\sqrt{2\pi h^2}} exp\left(\frac{\|x - x'\|^2}{h^2}\right) \tag{3-19}$$

This makes the maximum probability as

$$p_m(x) = \frac{1}{n_p}\sum_{i=1}^{n_p} w^i k_x(x, x^i) \tag{3-20}$$



(a) Particle Distribution by Monte Carlo Maximisation

(b) Particle Distribution by Sequential Monte Carlo Maximisation

Figure 3-4: Comparison of Maximisation algorithms

This maximum distribution is used for picking a random champion sample. The weighing equation for continuous domain looks like

$$w_c = \frac{q(x_c)}{q'(x_c|x^i)} = \frac{q(x_c)}{\alpha k_x(x, x') + (1-\alpha)q(x_c)} \tag{3-21}$$

Maximum Distribution In above three paragraphs, tools to remove the computational complexity and improving it's convergence to the true values are presented. The final distribution is created by a similar comparison methodology used in MCMD refer to (3-3-1). In which we first created a joint distribution of challenger and champion particles and then choose samples for both of them and perform comparison.

The extra bit which is done that all the challengers which wins a challenge is stored and the the challenger particles (x_c^j) are assigned a value, this value called *victory function value* $\hat{f}^i$ finally holds the maximum distribution. The final optimal distribution looks like

$$\underline{f}^\star = \frac{\sum_{i=1}^{n_p} w^i k_f(f, \hat{f}^i)}{\sum_{i=1}^{n_p} w^i} \tag{3-22}$$

So, to solve the problem the approach followed is to use Gaussian Process Regression for the prediction of the test data on the basis of training data set. For optimisation Monte Carlo Maximum Distribution is used at first and to improve the convergence sequential sampling techniques are used.

Chapter 4

Implementation

The previous chapter consisted of all the theoretical aspects that are to be applied in performing this project. This chapter contains all the details about the different steps taken and how the steps are accomplished.
For implementation, there were three major steps, first was to take correct measurements, second to create the prediction curve correctly and finally to achieve the likelihood of optimal gains.

4-1 Measurements

The type of learning problem we are using is a supervised learning problem, where the output at the different measurement values can be obtained. The training data points are used to further learn the function values. For continuous set of data the problem is often called as regression problem. The first step to take correct measurement is to create a suitable cost function in which all the important parameters that are generally looked into while wind turbine controller are used. Secondly, the measurement of the process noise should be done, so the amount of uncertainty in the posterior distribution of Gaussian Process regression is estimated correctly.

How many measurements are appropriate? In the second case study the amount of initial measurements (training data) becomes enormous. This phenomenon is called the 'Hughes phenomenon' or 'curse of dimensionality'. Hughes phenomenon say that having more dimensions help to get a better 'a posterior probability' or a better estimate of model to be learned but it also increase the bias of classification error (Covariance is Gaussian classifier) [19].
This increase of bias of classification error is due to increase in the number parameter to be predicted with same number of training inputs. When the increase in a posterior probability is less compared to increase in the bias of classification error, the purpose

of having additional dimension fails [19]. This proves that with increase in dimension, training data has to be increased enormous.
Another reason for having large number of training data is, increasing feature space increase the noise factor, and the error factor that there are not sufficient observations to get good estimate or the data is scattered [20].

4-1-1 Cost function

While designing the cost function four crucial factors which were kept in mind, are power production, fatigue loading (tower and blade), Pitch activity and rotor speed. The calculation of these parameters are done by simulating the wind turbine model against controller gains. The fatigue load calculation is done by rain flow counting technique by an aero-elastic simulation tool for span of atleast ten minutes.
After the data is obtained from the simulation, criteria are set different factors. For power production, the power production should be around 4 Megawatts (rated power) and so that the rotor speed is also in permissible limits, a factor for deviation of power from the 4MW mark is kept. The reason behind doing this is, in above rated state rotor speed tends to become very high for power production and we want minimum fluctuation from rated power. The maximum rotor speed for the 'D4rel' is around 13.6 rpm.
For the fatigue loading part, both tower fatigue loads and fatigue load of all the three blades are taken into consideration. As these values are very large parameters (in range the of 10^7- 10^8), they were normalised by the fatigue loads obtained at current optimal gains (gain at which the controller is optimum at above rated).
Similarly, for the pitch activity, the pitch rate of all the three blades were taken and here also it was normalised with the pitch rate of the current optimal gain.
All the individual costs were squared so that the cost function becomes convex and γ constant weight was added to the power production to create trade off between all the three costs. The cost function expression used is

$$\text{Cost due to power} = \left(\frac{\text{Power produced} - \text{Rated power}}{\text{Rated power}}\right)^2 \tag{4-1}$$

$$\text{Cost due to fatigue loads} = \left(\frac{\text{Mean of fatigue loads of all the three blades}}{\text{Reference blade fatigue loads}}\right)^2 + \left(\frac{\text{Rean of fatigue loads of Tower}}{\text{Reference tower fatigue loads}}\right)^2 \tag{4-2}$$

$$\text{Cost due to pitch activity} = \left(\frac{\text{Pitch activity}}{\text{Reference pitch activity}}\right)^2 \tag{4-3}$$

$$\text{Final cost} = \gamma \cdot \text{Cost due to power} + \text{Cost due to fatigue loads} + \text{Cost due to pitch} \tag{4-4}$$

where γ=10^2, and the weight of other two components are 1.
At a controller gain the non linear model of the 'D4rel' turbine with PI-controller is

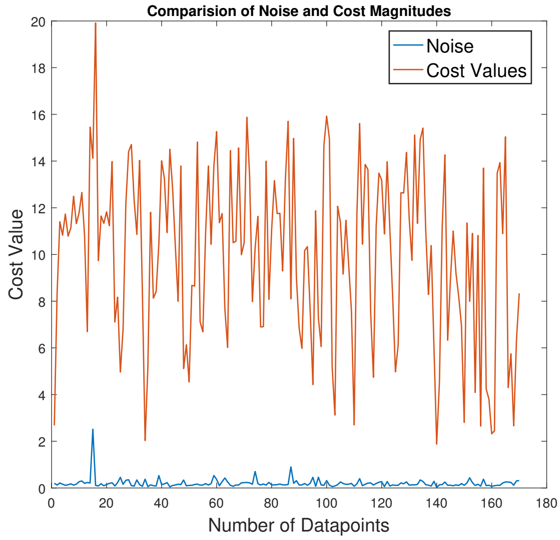


Figure 4-1: Comparison between magnitude of noise induced and cost values

simulated and then data obtained from simulation is used for cost function calculation. The expected cost of cost due to power is in range of 10^{-2}-10^{-3}, expected cost due to fatigue load is 0.8-1 and expected cost due to pitch activity is in range of 0.8-1.2 .
This is a minimisation cost function as we want to reduce the sum of all the three costs as we want to reduce pitch activity and fatigue loads with a power yield near to that of rated.

4-1-2 Measuring Noise

To have some randomness in the measurements, the gains that were chosen for simulation were completely random numbers in the space. To know the amount of process noise each simulation consisted of six different wind seeds so by doing that we could get six different results for the same input parameter, this helps us to estimate the uncertainty in the process. In (Figure 4-1) we can see the comparison between noise and cost values, also the randomness in measurement can be observed in it. The reason behind this step is to ensure that the process is a Gaussian process. It is crucial to take correct measurements for a supervised learning problem as unwanted and erroneous measurements could lead us giving erroneous function value prediction which can cause further problems in optimisation and getting correct final results.

4-2 Regression

As mentioned before, the kind of problem we're solving is known as a 'Supervised learning' problem and in continuous domain it's known as 'regression'. So in this section we want to predict the function values at test data points using the measurement we took. The technique we used to perform the regression is 'Gaussian Process Regression'. 'Sparse and Online Regression' which is a modified form of Gaussian Process Regression Technique (GPRT) can be used but only when the number training points becomes very high. The first step to get a good regression is to achieve suitable hyperparameters for the GPRT.

Tuning Hyperparameters As we are tackling a multi dimensional problem where we are trying to tune K_p and K_i initially and then some more parameters later on, so we have to create a multi-dimensional regression. The covariance function for a multi-dimensional GPRT looks like [16]:

$$k(x, x') = \lambda_f^2 exp\left(-\frac{1}{2}(x - x')^T \Lambda_x^{-1}(x - x')\right) \tag{4-5}$$

where Λ_x is the diagonal matrix containing the length scale of each of the input dimension.
Rather than manaully tuning the length scales, we used a gradient based method to maximise the log-likelihood of a posterior distribution i.e. equation (3-11). Initially, some prior values of hyperparameters are chosen as the initial condition for optimisation to start. The next step is to calculate the derivative of the equation (3-11) with respect to the hyperparameters, and calculate their values. A certain number of steps are pre-defined to carry out the iterations with varying size of the steps. The size of the step is decided by the value of log-likelihood of Maximum A Posteriori (MAP) distribution. With increasing value of the gradients the step size becomes smaller by a factor of 2 each time. Different initial condition (hyper-prior) was applied for the optimisation so as to know the best possible minima value.
Also, hyperparameters were checked by tuning them in Gaussian Process for Machine Learning (GPML) toolbox given by [12] which optimise the hyperparemeters on a line and direction search based algorithm. The toolbox also focuses on increasing the negative log likelihood of aposterior distribution. The hyperparameters were quite similar to what we obtained by gradient ascent method.

Gaussian Process Regression Till now we know that the process is Gaussian and how to tune the correct set of hyperparameters for a squared exponential covariance function. The next step is to create the regression where prediction of the function values at different data point is done.
The first bit of regression is to calculate the prior distribution between the measurements ($\underline{f}_m$) and the test values ($\underline{f}_\star$), which is done according to the equation (3-4). The second step is to calculate the posterior mean and variance values at the test points, which is done according to equation (3-6).
Number of measurements were increased gradually after achieving a set of points, as we wanted to look how many measurements will be satisfactory for getting good results.

4-3 Gaussian Process Optimisation

This section is about how the Regression was optimised to achieve the likelihood of the optimal controller gains.
The first step is to go for a maximisation, as the cost function we have is a minimisation cost function so the first step is to change it to maximisation by making it negative.

Maximum Distribution After changing the problem to a maximisation problem, first the Monte Carlo Maximum Distribution (MCMD) is used to calculate the maximum distribution. By calculating this we somewhat get an idea of the region where the likelihood of gains that maximise the predicted function value lies. We also run the comparison algorithm for certain number of rounds to check how many rounds of comparison gives a converging result and how much time it takes to achieve that, so that a comparison between MCMD and Sequential Monte Carlo (SMC) can be done.
We apply the sequential sampling techniques to sample the data in a better way, *sytematic resampling*, *Importance sampling* and *Defensive sampling* techniques were used. Following the sampling, similar kind of comparison is done to get a maximum distribution of particles.

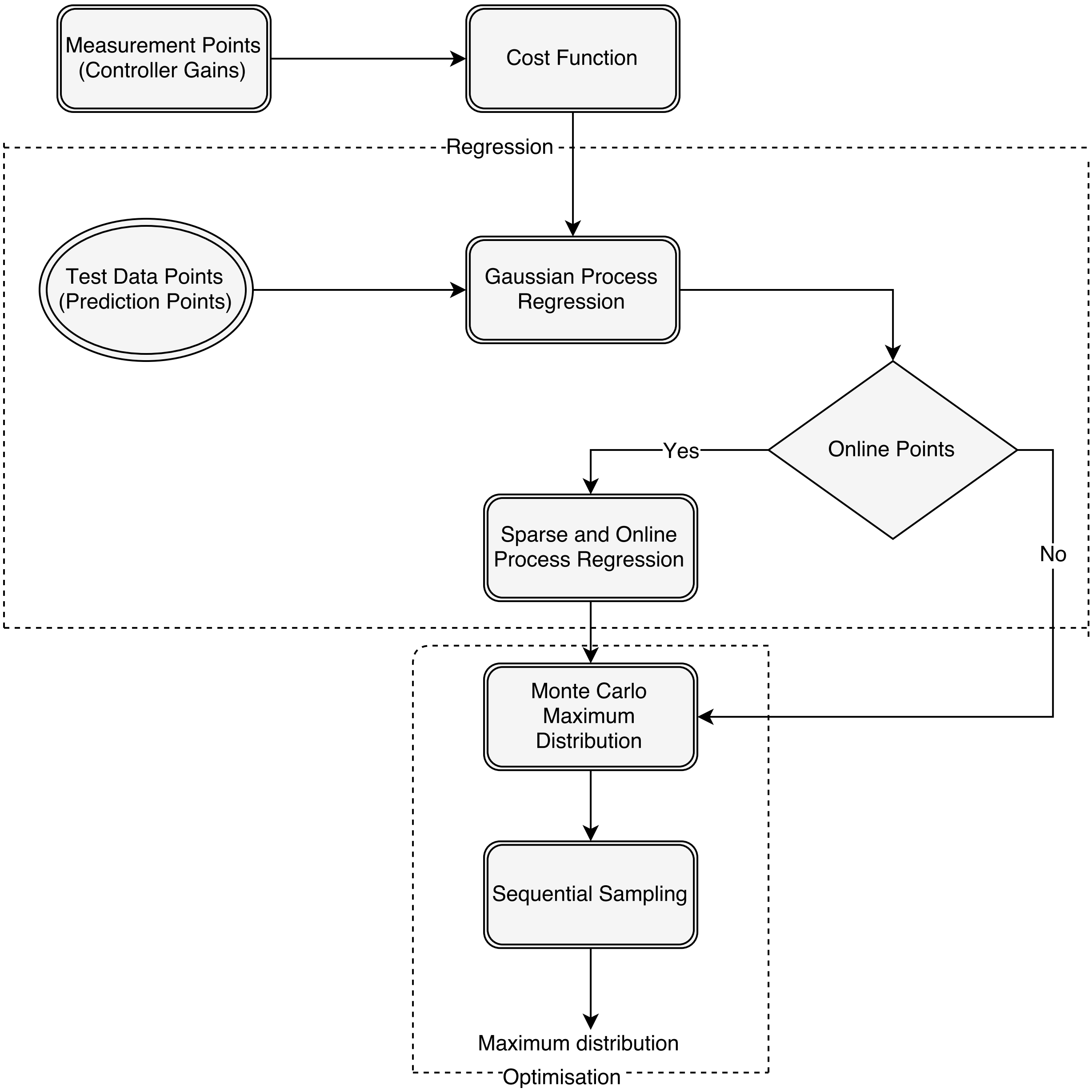


Figure 4-2: Flow chart of Automated tuning

Chapter 5

Results

This section presents the results obtained by applying the theory discussed to a wind turbine controller. In depth analysis of the results are also done which were supported by test results obtained by aero-elastic simulation.

In this thesis the application of the theory has been done in two parts the first one is the major part where the tuning the gains of Pitch K_p and K_i and is done. This done for both above rated and near rated mean wind speeds but as a different set of simulation. Finally a comparison between the obtained optimal gain and the optimal gain currently used by Energy Centre Netherlands (ECN).

In the second part of the experiment deals with expanding the tuning algorithm, as tuning is performed with few more variables. The aim was to look if the cumulative effect of tuning can be obtained or not.

5-1 First case -Tuning with two inputs

This case deals with tuning of pitch K_p and K_i. The tuning algorithm used in this thesis has two parts first of prediction where a training data set is taken and using Gaussian Process Regression Technique (GPRT) new predictions are made in the space, the second part is optimisation of this prediction region.

5-1-1 Tuning for above rated case

Prediction First the prediction is done for the above rated wind speeds and a mesh of points were created so as to show the fit between the measurement points and the

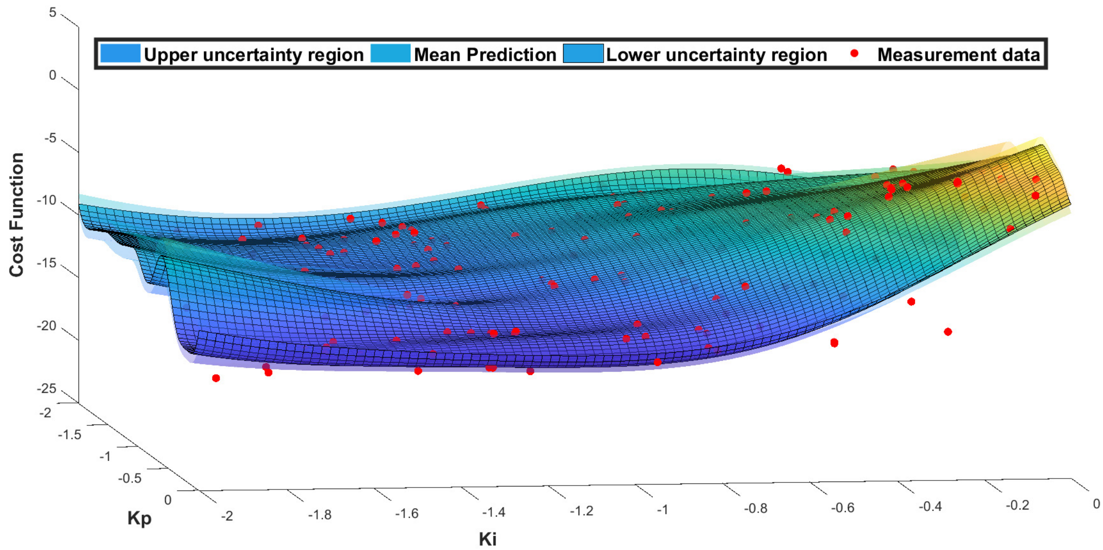


Figure 5-1: Prediction mesh for above rated tuning

space of prediction points. This fit is shown in (Figure 5-1). The upper and lower uncertainty bounds in (Figure 5-1) shows the uncertainty due to process noise in the wind turbine which was captured by having various wind seeds in our process.
It has been mentioned earlier that hyperparameters play a crucial role in having a correct Gaussian process regression. The hyperparameters obtained after tuning were $\begin{bmatrix} 1 & 0.15 & 0.65 & 0.22 \end{bmatrix}$ which denotes $\begin{bmatrix} \lambda_f & \lambda_{x_1} & \lambda_{x_2} & \sigma_{fm} \end{bmatrix}$ and the log likelihood which has to be maximised is -1.32×10^3.

Optimisation After the prediction, it can be clearly seen that there is no way the optimum can be perceived just by referring to (Figure 5-1), and the uncertainty bound also contains important information which needs to be accounted for. This calls for an optimisation of the mean prediction along with the uncertainty bounds. For that both Monte Carlo Maximum Distribution (MCMD) and Sequential Monte Carlo (SMC) are performed.
The (Figure 5-2) shows the likelihood of the maximum data points achieved by performing MCMD. In figure (5-2) we can see that there is a vast region that has no peaks and this suggests that there won't be any optimal value in that region. But in a small space there are lot of peaks of varying dimension, this makes it difficult to understand where the optimum will be truly converging. This is also stated in (3-3-1) while explaining MCMD. The advantage of doing this is we visually get an idea of where the optimum is, and when we've small number of points to optimise, this algorithm will be pretty quick, accurate and simple to apply.

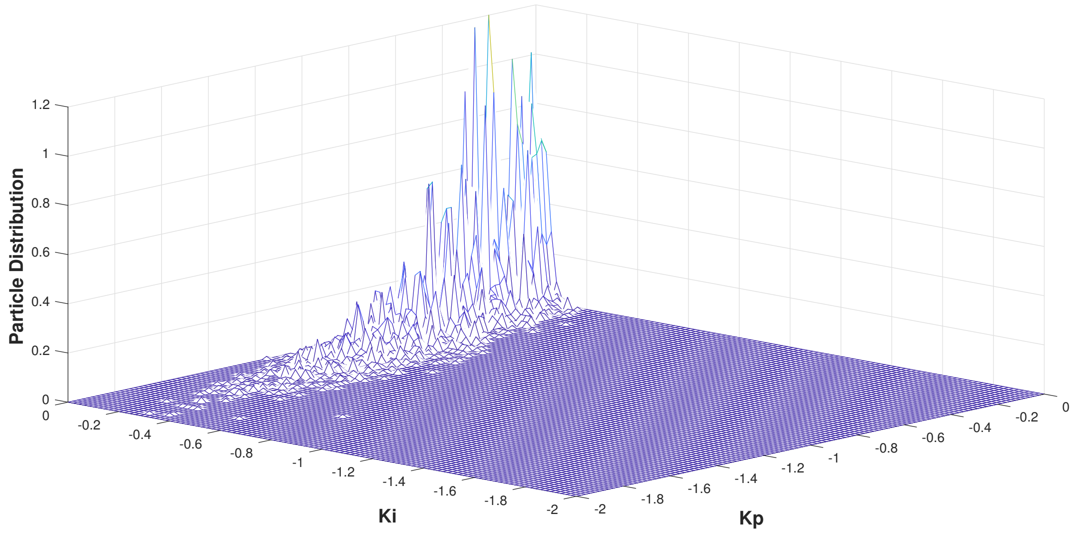


Figure 5-2: Monte Carlo Maximum Distribution for above rated prediction

The SMC gives a good convergence to the maxima in the region and so a good perception of likelihood can be made out of it. Though this algorithm is quite complex to apply, but computationally it is quite feasible when it comes for a big data set. For example, the results shown in figure 5-2 and 5-3 have been computed for different number of rounds. MCMD has been computed for n_r=30 where as for n_r=20 which is quite a difference and will affect more when done for a big data set.

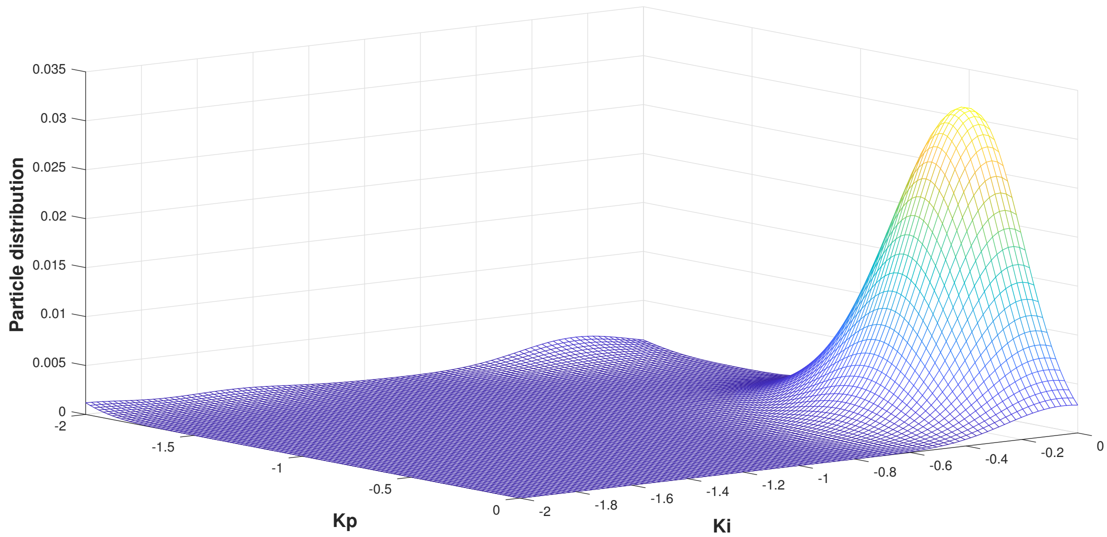


Figure 5-3: Optimisation using SMC for above rated wind Speeds

Test results So, looking at the results obtained from the optimisation a certain likelihood of gains can be obtained. To validate the correctness of the obtained result, a time series analysis of aero-elastic simulation is done for a set of controller gain parameters taken from the optimal likelihood. This is later compared to the optimum parameter currently used by ECN. The likelihood of optimum gain we've looked for taking a test for controller gain is

Tuning parameter	Likelihood region
K_p	(-0.7 - -0.3)
K_i	(-0.15 - -0.01)

Table 5-1: Likelihood region for controllers gains for above rated operational mode

For the time series analysis comparison of above case refer to appendix (A) where from (Figure A-1 to A-5) shows how the new optimum from the likelihood give a better performance.

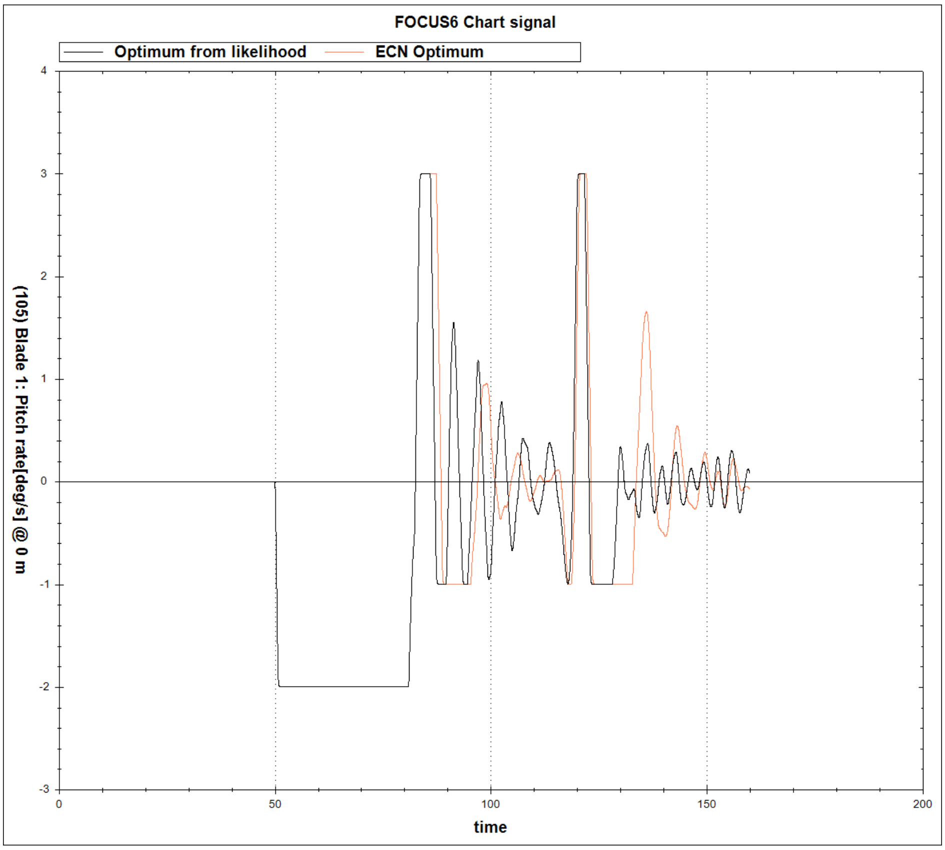


Figure 5-4: Comparison of pitch activity between ECN optimum and optimum from likelihood under disturbance

Also, a disturbance rejection test has been performed with extreme operating gust

(Figure 5-4). This gives an idea about the robustness of the gains, also a comparative analysis is done between the optimum used by ECN and the optimum from the likelihood. The result for all the parameters of disturbance rejection test is presented in appendix (A) from (Figure A-6) to (A-10).

5-1-2 Analysis with Near Rated Operation mode

In this part results are presented for a near rated operational mode, so that we can analyse whether the algorithm is valid for both the cases or not. Simulation at near rated condition will give a likelihood of optimal tuning parameter at the near rated region. This section has been divided in three parts prediction, optimisation and test.

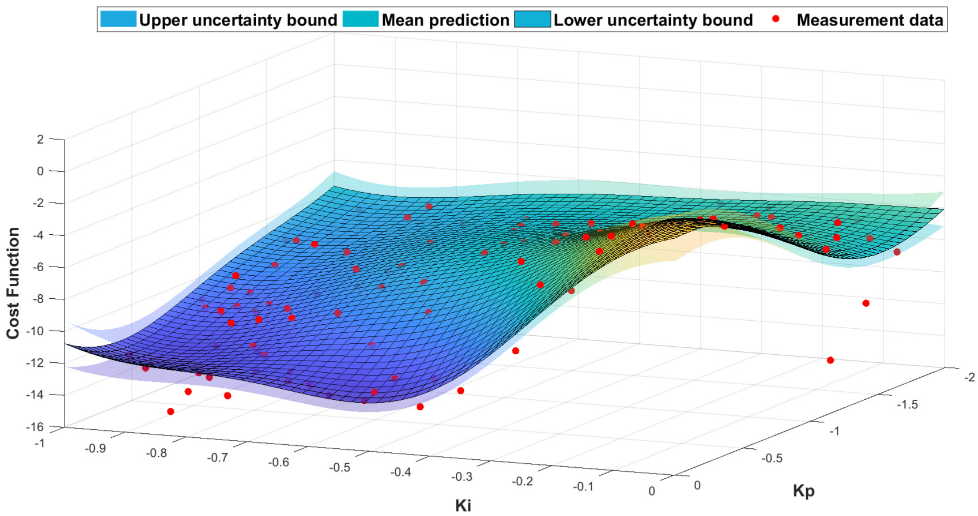


Figure 5-5: Prediction mesh for near rated tuning

Prediction Similar to above rating case, GPRT was applied for a set of measurement data and prediction for the trail data set was done. The hyperparameters were tuned using the gradient ascent algorithm. The hyperparemeter values in this case was $\begin{bmatrix}1.2 & 0.75 & 0.25 & 0.32\end{bmatrix}$ which denotes $\begin{bmatrix}\lambda_f & \lambda_{x_1} & \lambda_{x_2} & \sigma_{f_m}\end{bmatrix}$ and the log likelihood which was to be maximised is -275.89.
The mean prediction with upper and lower uncertainty bounds is shown in (Figure 5-5). This figure suggests that there is considerable fit between the measurement points and the prediction mesh.

Optimisation In a similar fashion as above rated, SMC and MCMD are performed for the prediction set of near rated data set. A result for SMC is presented in (Figure 5-6).

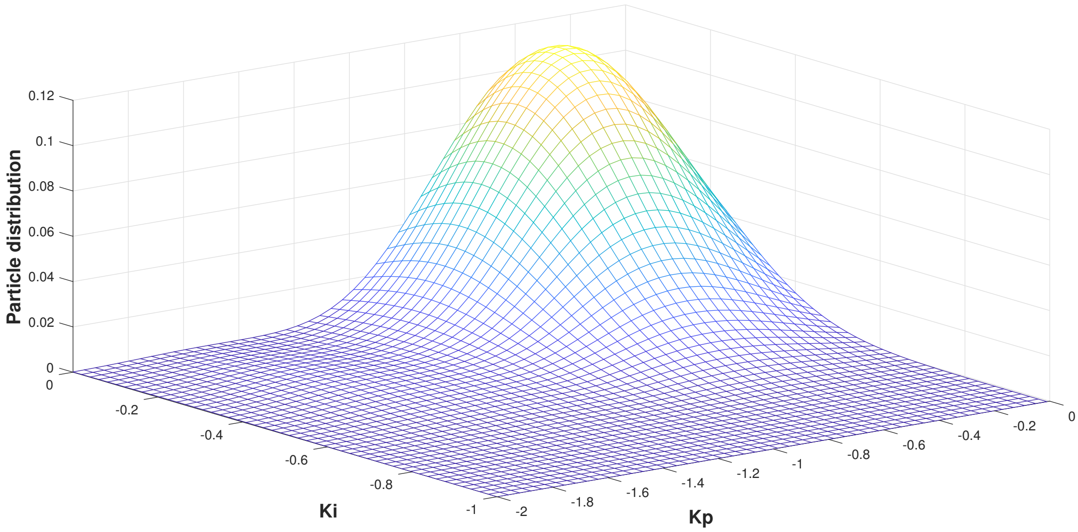


Figure 5-6: SMC optimisation plot for Near rated operational mode

Test Results A clear perception of likelihood of wind turbine gains is obtained from this optimisation plot. The likelihood for the optimum is

Tuning parameter	Likelihood region
K_p	(-0.7 - -0.2)
K_i	(-0.25 - -0.01)

Table 5-2: Likelihood region for controllers gains for near rated region.

Time series analysis of aero-elastic simulations are done for near rated wind speeds, where the gain was chosen from the likelihood region given in table 5-2. Time series analysis is presented in appendix A in section **??** which validates that the likelihood has some optimum values.

5-1-3 Quantitative comparison

After looking to the plots of SMC for both the cases, a time series analysis was done for gains that were perceived as maximum (which has maximum predicted cost value). For the plots of the comparison refer to Appendix (A). In table 5-3 quantitative analysis is done so that the amount of improvement can be analysed clearly.

Table 5-3: Comparison of Below rated and above rated performance between current parameter and calculated parameter

Parameters	Above rated		Near Rated		Improvement (%)	
	Current	Calcul.	Current	Calcul.	Ab. rated	near Rated
Generated Power (MW)	3.992	3.98	3.93	3.97	0.3	1
Pitch activity (deg/sec)	0.69	0.60	0.726	0.59	13	18.7
Max Rotor Speed	14.5	14.3	14.4	14.1	1.3	2
Tower Fatigue Loads(10^7 Nm)	2.81	2.61	2.65	2.42	7	8.67
Blade Fatigue Loads (10^6 Nm)	5.29	4.71	5.232	4.78	11	8

After looking at the table above, it can be said that there is possibility that more optimised parameter is present in the region of likelihood achieved after optimisation.
Also, when looked at the disturbance rejection results it can be said that not only these parameters are performing load reduction without any kind of loss in power but are also very robust to disturbances.
This answers the question of sensitivity of controller when disturbances are introduced in the system. This helps us further as wind turbines are exposed to uncertain working condition due to changing wind speeds.

5-2 Second case - More tuning variables

The aim of performing this case that how resourceful this algorithm is. As, performing a two dimensional optimisation is quite simple and can be done by other available techniques with ease.
Apart from increasing number of variables, both near rated operational mode and above rated operational mode load cases are kept in consideration. Previously controllers were separately tuned for near rated condition and above rated condition, but this case would provide a parameter that would be optimal for both the cases.
For this phase there are four parameters pitch K_p, K_i and detection threshold gains Vu_a & Vu_b. The main task of Vu_a & Vu_b is to detect gust in the incoming wind and then perform high pitch action.
Before directly moving to a higher dimension prediction and optimisation an attempt was made to perform task of similar complexity with an arbitrary non-linear function with some noise added to it.

5-2-1 Test function

The purpose of doing it with an arbitrary function is that the true values from the arbitrary function at the test points can be calculated then a comparison can be made with the results achieved from the algorithm. As in second case when the algorithm

will be tested for a wind turbine controller there won't be true values to validate the predictions. So, it helps us to validate the fit of the prediction and reliability one can have on this algorithm using an arbitrary function.
The chosen arbitrary function looks like:

$$Z = cos(x_1 - 2) + sin(x_2 + 2) + x_3^2 + 4x_4 + \sigma \tag{5-1}$$

where σ is random noise.

Prediction The first step is to take measurements and then to apply Gaussian process regression to predict the input-output mapping at set of test data points.
As we're working in a higher dimensional space the form of the covariance formulae changes as per described in equation (3-9). This makes hyper-parameter tuning more complex as there will be more gradients to calculate for maximising the negative log-likelihood. The increase in dimension leads to increase in number of length scales and the number of measurement points required shoots up to achieve a good covariance [20]. The hyper-parameters for this part of experiment were tuned using the gradient ascent algorithm same as in two dimensional case. The hyper-paremeter values in this cases were $\begin{bmatrix} 4.21 & 4.055 & 3.022 & 3.57 & 4.30 & 3.7 \end{bmatrix}$ which denotes $\begin{bmatrix} \lambda_f & \lambda_{x_1} & \lambda_{x_2} & \lambda_{x_3} & \lambda_{x_4} & \sigma_{f_m} \end{bmatrix}$. As it is difficult to say how good the prediction is because it has more than three dimensions. A technique which is quite popular to determine how good the fit is, by plotting the true value at the points where prediction is done versus the mean prediction value. Ideally if it is a good fit the plot should be linear [21].

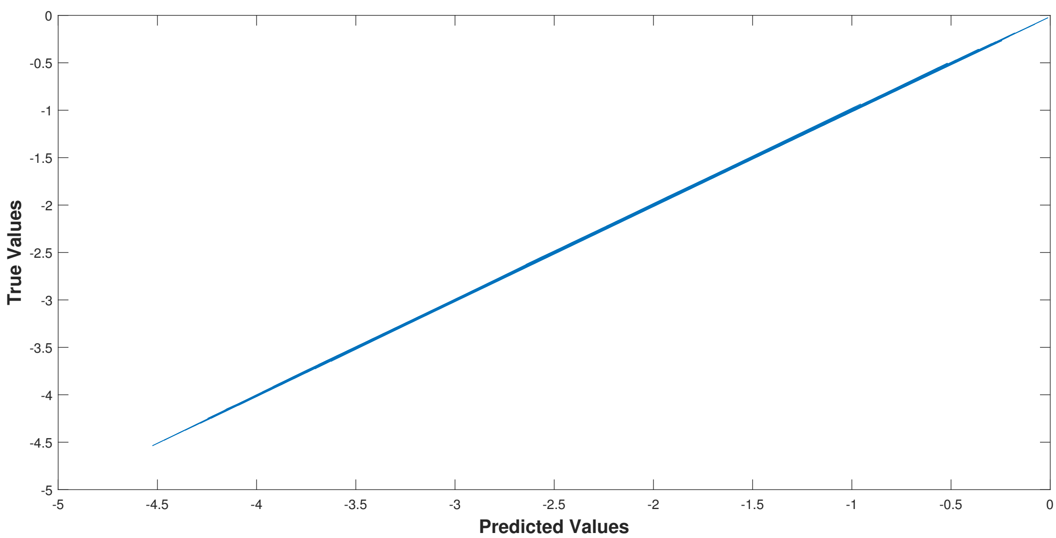


Figure 5-7: True value v/s Predicted value

So, looking at (Figure 5-7) it can be said the prediction is good.
To validate the fit, another way is to plot the true value of the function at the test points and plot the prediction obtained from Gaussian process regression, refer to (Figure 5-8).

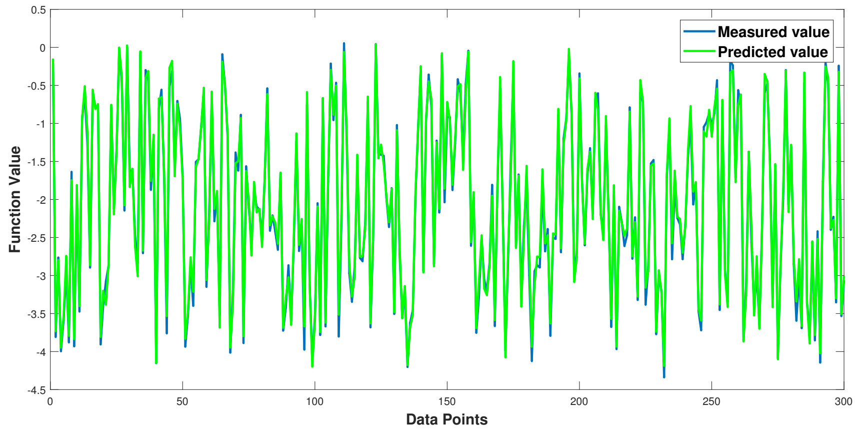


Figure 5-8: Comparing true value and predicted value for test data points

Optimisation This section both the optimisation techniques were used to show the points of maxima with respect to the data points and also a true function value plot is given. This helps to validate the optimisation. The (Figure 5-9) has particle distribution where the high particle distribution at certain points shows the maximum value of the arbitrary function at the corresponding data point. The (Figure 5-9) uses the MCMD algorithm to find maxima.

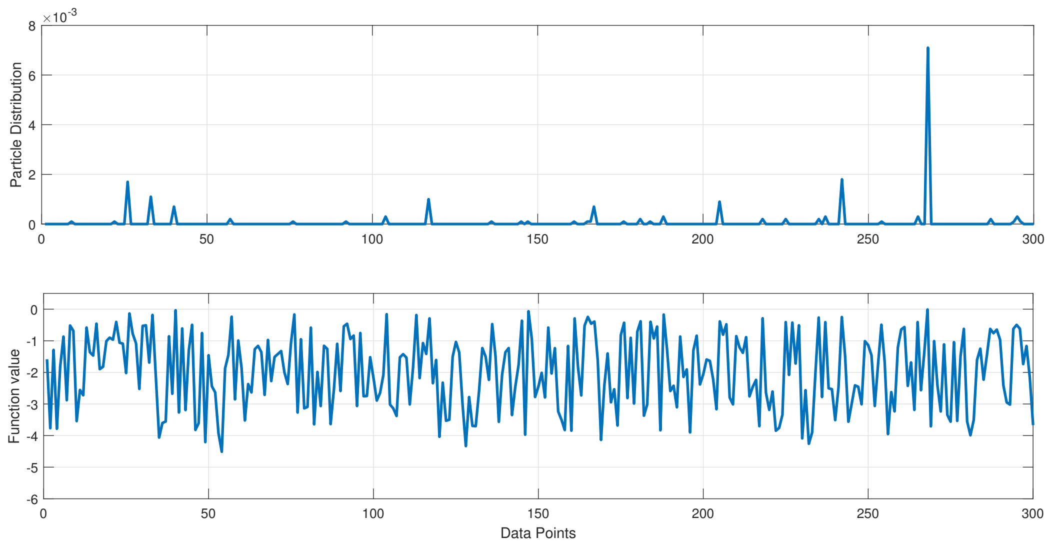


Figure 5-9: Maximum distribution using MCMD along with true value plot

As in this section the data points were taken randomly with some noise, thats why there is some randomness in the function value plot (not to be confused as a time series data).

Sequential Monte Carlo SMC is performed to get a better maximum distribution. As mentioned in section (3-3-1), SMC helps to get a true convergence to the maxima and is computationally cheaper. Here when a comparison made between figure (5-10) and (5-9) it can be clearly seen that some maximum values are missing out in (Figure 5-9). This helps us to understand that SMC is a superior algorithm as it is more precise compared to MCMD for optimisation.

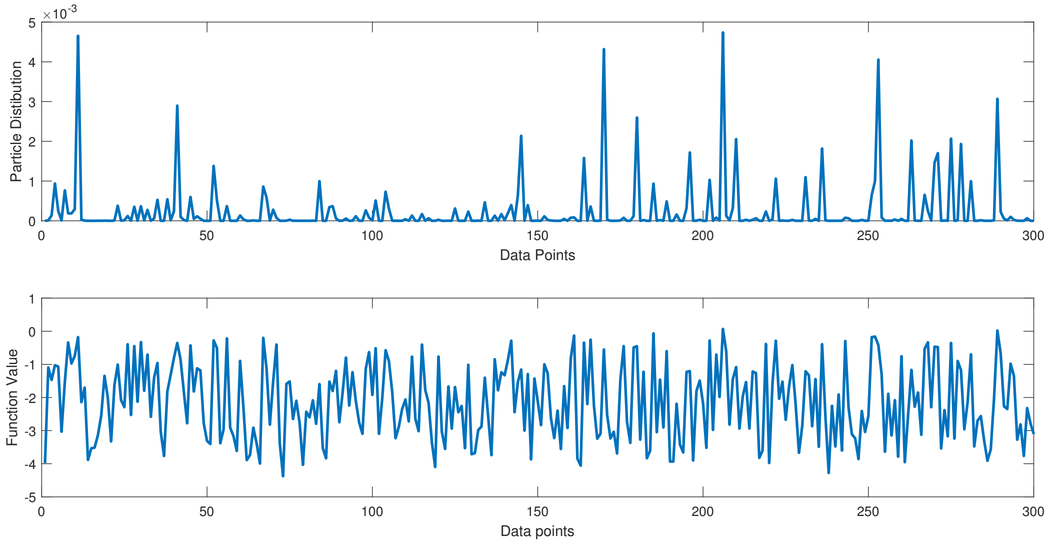


Figure 5-10: Maximum distribution using SMC along with true value plot

If we sort and put all the maxima's aside and analyse for at least 10% of the maximum distribution we can easily get a likelihood of values where there is a possible maxima but it's not that easy as compared to what we saw in two dimensional optimisation.

Another disadvantage to have more variable is the number of initial measurement increases by few folds and the length scales becomes big, that reduces the correlation between the data points.

Paramater	Likelihood Value
x_1	(-1.00 - -1.50)
x_2	(-0.1 - -0.45)
x_3	(0.005 - 0.02)
x_4	(0.3 - 0.05)

Table 5-4: Likelihood for an arbitrary function

The results in the table (5-4) is a local maxima, as the measurement and prediction is done in a chosen specific range. Thus, this results are optimum for that particular space.

5-2-2 On Wind turbine controller

For the wind turbine model it is not possible to get true value for a set of control parameter as the output value is generated from a simulator, and each measurement almost takes 10 minutes. In previous section it can be observed that for a large dimensional problem this algorithm was successful.
The variables added in this part of the problem are (pitch $K_p \& K_i$ and $Vu_a \& Vu_b$). In this section the wind turbines are simulated for the for both above rated wind speed and near rated wind speed combined.

Prediction In this section it is difficult to show how good our prediction is as we can't compare it with true value as we did it in previous section. The only thing that one can do is to look what is the variance given by the GP regression. Smaller the variance better the fit is. Also to show the amount of fit some of the points from the prediction region were actually simulated to check the error correctness. This doesn't helps to say that if the whole regression is correct or not because in a non linear system it's very uncertain how the behaviour would be changing.
The hyper parameter tuning is done similarly as done in all the cases. The hyperparemters are $\begin{bmatrix} 4.323 & 0.65 & 0.2924 & 5.4 & 1.082 & 3.2 \end{bmatrix}$ these denotes $\begin{bmatrix} \lambda_f & \lambda_{x_1} & \lambda_{x_2} & \lambda_{x3} & \lambda_{x4} & \sigma_{f_m} \end{bmatrix}$.

Optimisation Optimisation in this part is done only with SMC as it is shown clearly in the previous sections that SMC is more precise in pointing at maxima's rather than the MCMD.
The (Figure 5-11) gives the particle distribution with SMC along with the predicted mean values.

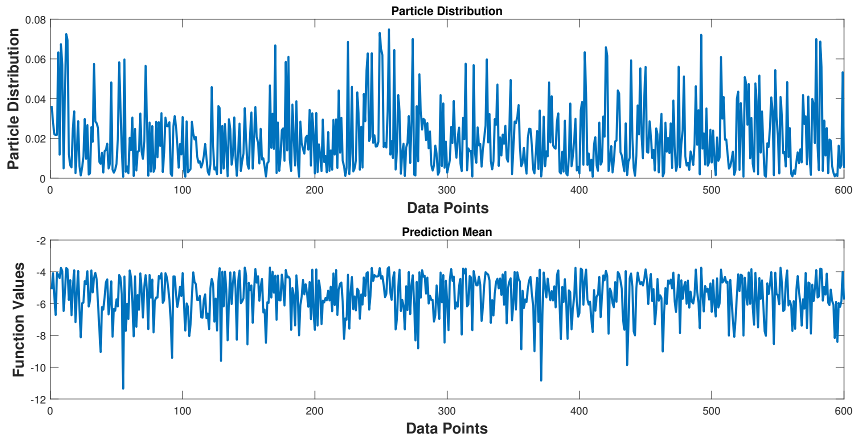


Figure 5-11: Particle Distribution with Sequential Monte Carlo Maximisation along with Predicted Mean Values

Similarly, as done in arbitrary function the particle distribution is sorted and it can help us to find the likelihood of the maximum distribution.

Test Results What we see in the time series analysis of aero-elastic simulation (in appendix A in section A-4) is that at an above rated wind speed, if the threshold detection gains ($Vu_a \& Vu_b$) are set at an very low values it'll start the Extreme Event Control (EEC) action where blades are pitched at a very high angles even at a small gust. Looking at the results it can be inferred that at above rated stochastic wind (mean wind speed is 16 m/s in this case) it is not very ideal to have EEC, maybe at a large sudden gust this parameter would be more efficient.
This can be proved by comparing fatigue loads of tower and the blades, along with the maximum and minimum tower bending moments and blade bending moment.
Also time series plots are shown in appendix A in section A-4. For a quantitative analysis the comparative values are given in the table.

Parameters	ECN Optimum	Calculated Optimum
Tower Fatigue Load (10^7N-m)	2.737	2.616
Blade fatigue Load (10^6N-m)	5.175	4.732
Maximum Tower Bending Moment (10^7N-m)	1.275	1.21
Minimum Tower Bending Moment (10^7N-m)	-0.303	-0.273
Maximum Blade Bending Moment (10^7N-m)	1.303	1.242
Minimum Blade Bending Moment (10^6N-m)	-1.13	-0.537

Table 5-5: Comparison Table for Parameter with EEC and without extreme event control

The reason behind having Maximum and Minimum Blade Bending Moment is this gives a better idea of the loads by the incoming gust.

Chapter 6

Conclusions and Recommendations

6-1 Conclusions

Gaussian process regression coupled with Monte Carlo based optimisation technique has given a new perspective for tuning a wind turbine controller. By achieving optimal control parameters for wind turbine problem which considers multiple factors for being optimal, shows a lot of prospect with this algorithm. The objectives mentioned in the problem formulation are answered in this section:

Designing of a cost function Having a good cost function is essential because in a supervised learning problem input-output mapping is to be predicted on basis of training data. For having a appropriate training data the cost function should define the dynamics of a wind turbine properly. For this a data based cost function consisting of net power production, fatigue loading of flapwise bending moment and tower bending moment and pitch activity is used. The aim of this cost function is to keep the power production as close to rated power i.e. 4 MW with smallest possible pitch activity and fatigue loads.
The cost function represents the turbine behaviour under various set of tuning parameters. For the second case where we use four variables an extra eye on maximum and minimum tower bending moment and blade bending moment is kept. As, at high wind case threshold gain for Extreme Operating region is used for detecting gust that could possibly damage the wind turbine.

To apply a learning algorithm to predict cost value from the training data at set of trail controller parameters Gaussian process regression was used to predict the cost values at a set of trail controller parameters. Gaussian process regression was applied along with tuned hyper-parameters. The tuning of hyper-parameter was done in order to maximise the log likelihood of a posterior distribution. Process noise was measured by

running various seeds for single simulation. For first case (two dimensional) as surface fit is created which fits with the measurement points which can be seen in (Figures 5-5) and (5-1). This give us a prediction of cost value at set of random trail points.
For the second case where Regression with multiple inputs, the accuracy of the fit is analysed by Figures (5-7) and (5-8).
Thus, prediction of the cost values at set of trail controller parameters helps to extend the set of controller parameters with corresponding cost values. This saves us to perform large number aero-elastic simulations at different controller parameters. Also, Gaussian process regression can successfully predict input-output mapping for a problem having large number of inputs. But with a small downside of having a big training set.

Creating a suitable likelihood of gains and tuning parameters After achieving a prediction of cost value at a set of test data points (controller gains), a likelihood needs to be found out where there is optimum cost. For achieving the likelihood Monte Carlo Maximum distribution and its modified version with sequential sampling technique is used.
The algorithms were applied for two cases, first is for tuning two parameters (pitch K_p & K_i). In this case, both algorithms predicted region of optimum points which can be clearly seen in (Figure 5-3), range of likelihood were also presented in (Tables 5-2 - 5-1). Upon doing a quantitative comparison with than current parameters used by Energy Centre Netherlands (ECN), the parameters in likehood were found to be more optimal than the currently used parameters.
For second case tuning of four parameters is performed, to know the potential of this optimisation technique. First this was performed upon an arbitrary function, where prediction space was created first and then optimisation was done (5-2-1). Then optimisation with wind turbine parameters (pitch $K_p \& K_i$ and $Vu_a \& Vu_b$) which were extrapolated by Gaussian Process regression (5-11).
Thus, performing this case successfully tells us that this algorithm can optimise a n-dimensional problem successfully. So, if a case study consists of 'n' number of variables and a optimal set of variable has to be calculated this algorithm can do it successfully.
The Sequential sampling techniques helped to give out a better likelihood of the optimum control parameters when compared to Monte Carlo Maximisation.

The goal to achieve a likelihood of optimally tuned parameter is successfully met and also an comparison has been done. In comparison it was shown that the parameters in the likelihood were quite much of improvement when compared to current parameters. This was confirmed by performing aero-elastic against the optimal gains from the likelihood.

6-2 Recommendations

This thesis work is a result of collaboration of machine learning and wind turbine controller. Which opens up wide possibilities as new machine learning algorithms are being developed everyday and wind turbine control is getting more and more complex. So, this thesis can pave the way for aspirants of wind turbine control to develop and apply machine learning principles in wind turbine control. Some recommendations are made so that this work can be improved further

- Wind turbine control is a difficult task due to large number of parameters to be look out for, for achieving best performance. So in future work some more parameters can be added to the cost function to improve it.
- For constantly growing data sets, Sparse and online GP regression has been shown in this thesis can be done so as the prediction can be be done with least computational complexity and runtime. This can help also for tuning multiple dimensions as they need much larger training set.
- As a maximum likelihood is obtained a Bayesian optimisation scheme for (regret/error minimisation) can be set up. This will further optimise the likelihood and will give further optimal controller parameters.
- In this thesis only a stand alone Wind turbine but it'll be good to see how this algorithm performs for a wind farm set up. As it will have more variables and more parameters to optimise, the optimisation and prediction will be more difficult. Though it is possible that a meta-heuristic optimisation will be better solution to wind farm problem.

Appendix A

Appendix A

This appendix consists of all the time series analysis comparison done for different optimum tuning parameters. The results of comparison for both above rated and near rated cases are done in this appendix. Also a disturbance rejection under 'Extreme Operating Gust' is given for the parameters tuned in two dimensional optimisation.

A-1 Time series analysis of above rated operational mode

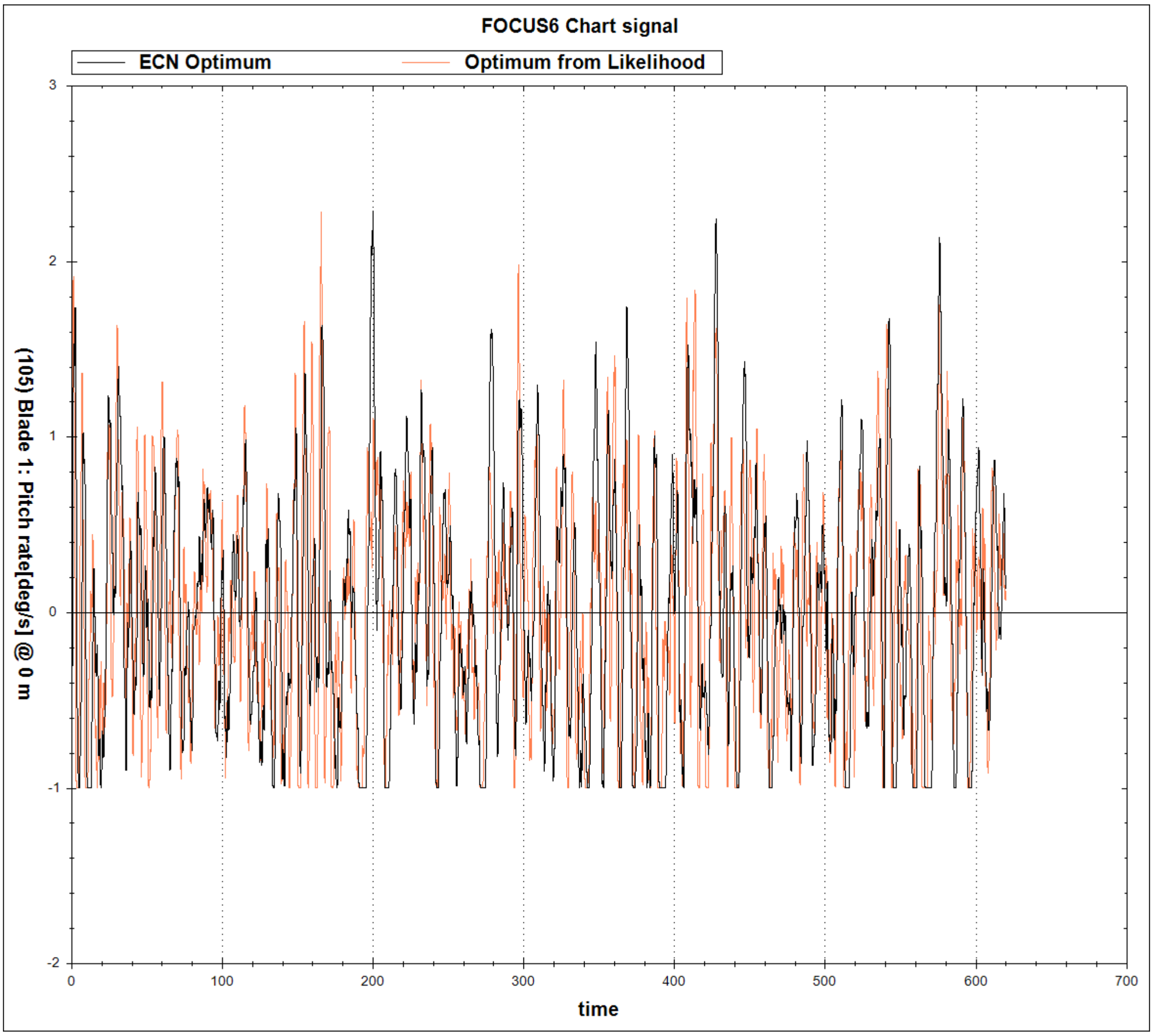


Figure A-1: Pitch activity Comparison between ECN Optimum and Optimum from likelihood

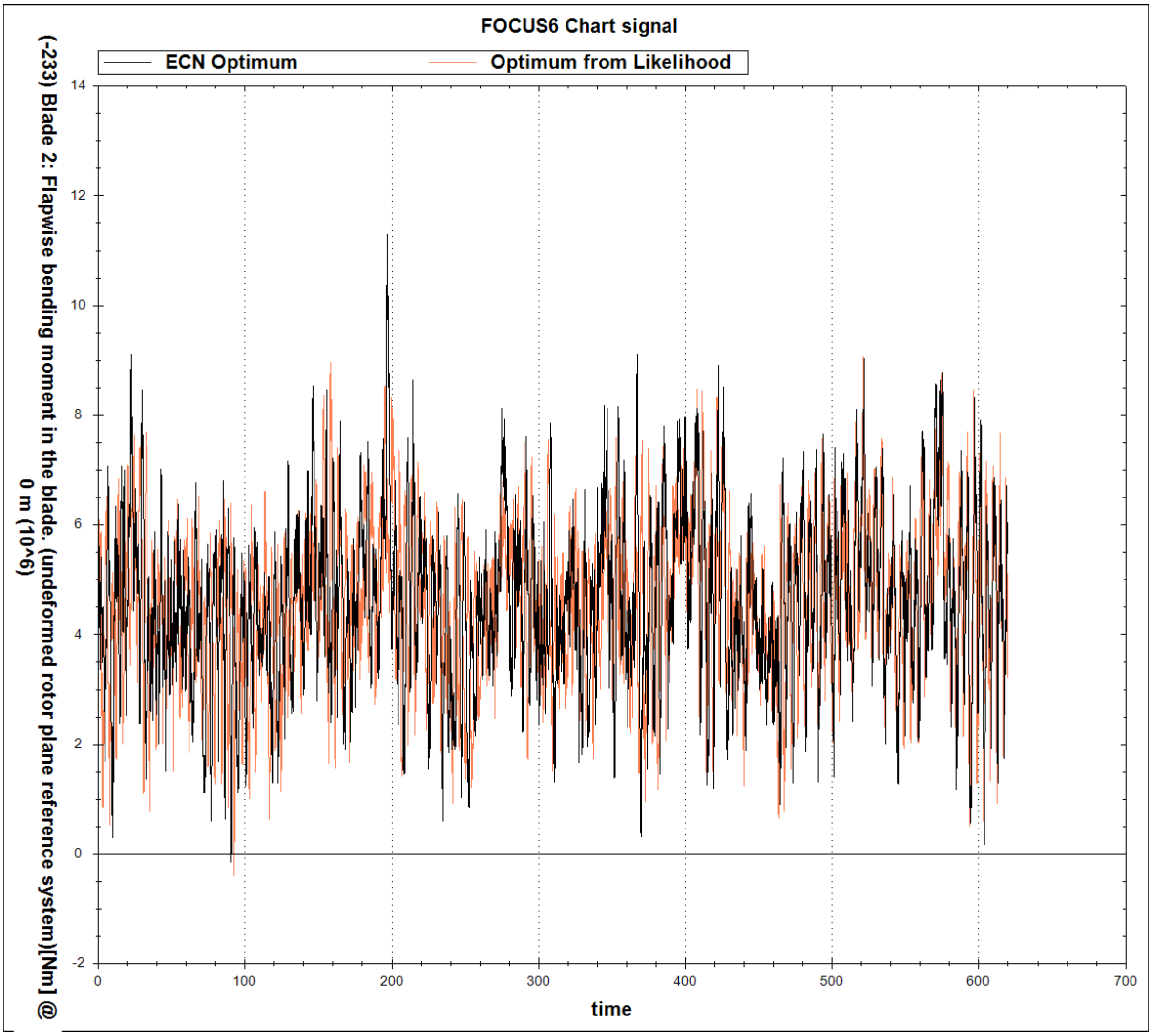


Figure A-2: Blade fatigue load comparison between ECN Optimum and Optimum from likelihood

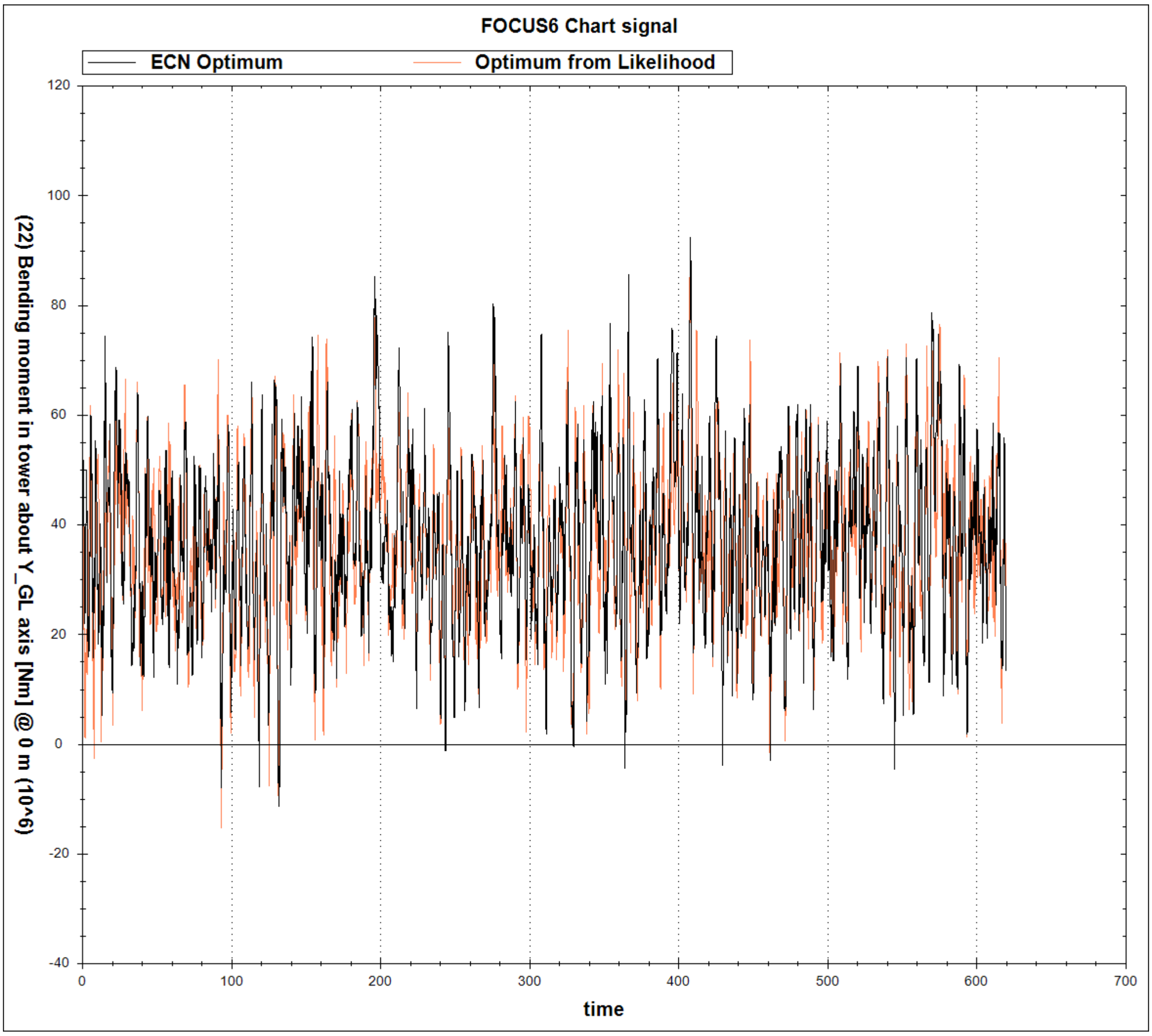


Figure A-3: Tower load fatigue comparison between ECN Optimum and Optimum from likelihood

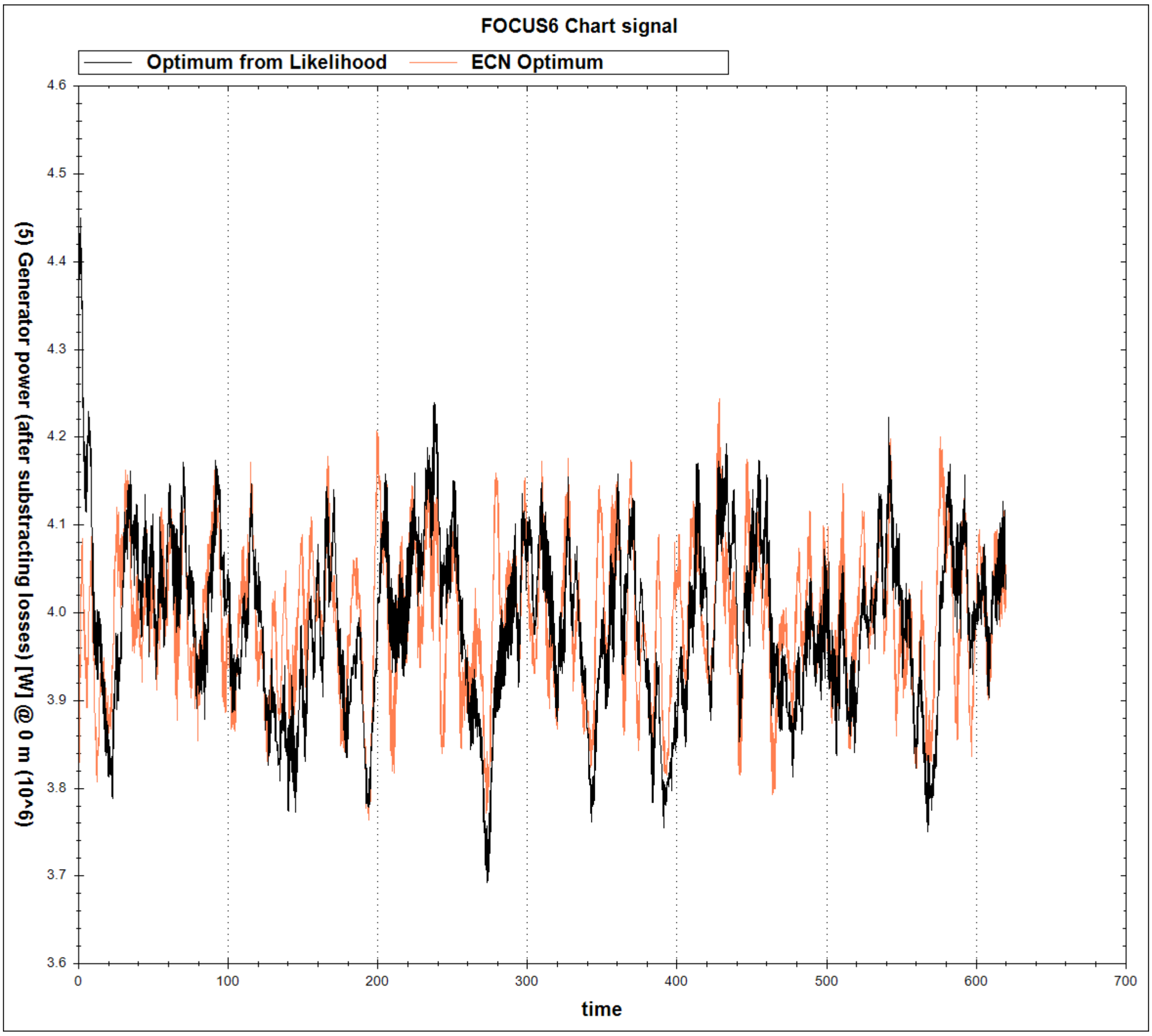


Figure A-4: Comparison of power production between ECN Optimum and Optimum from likelihood

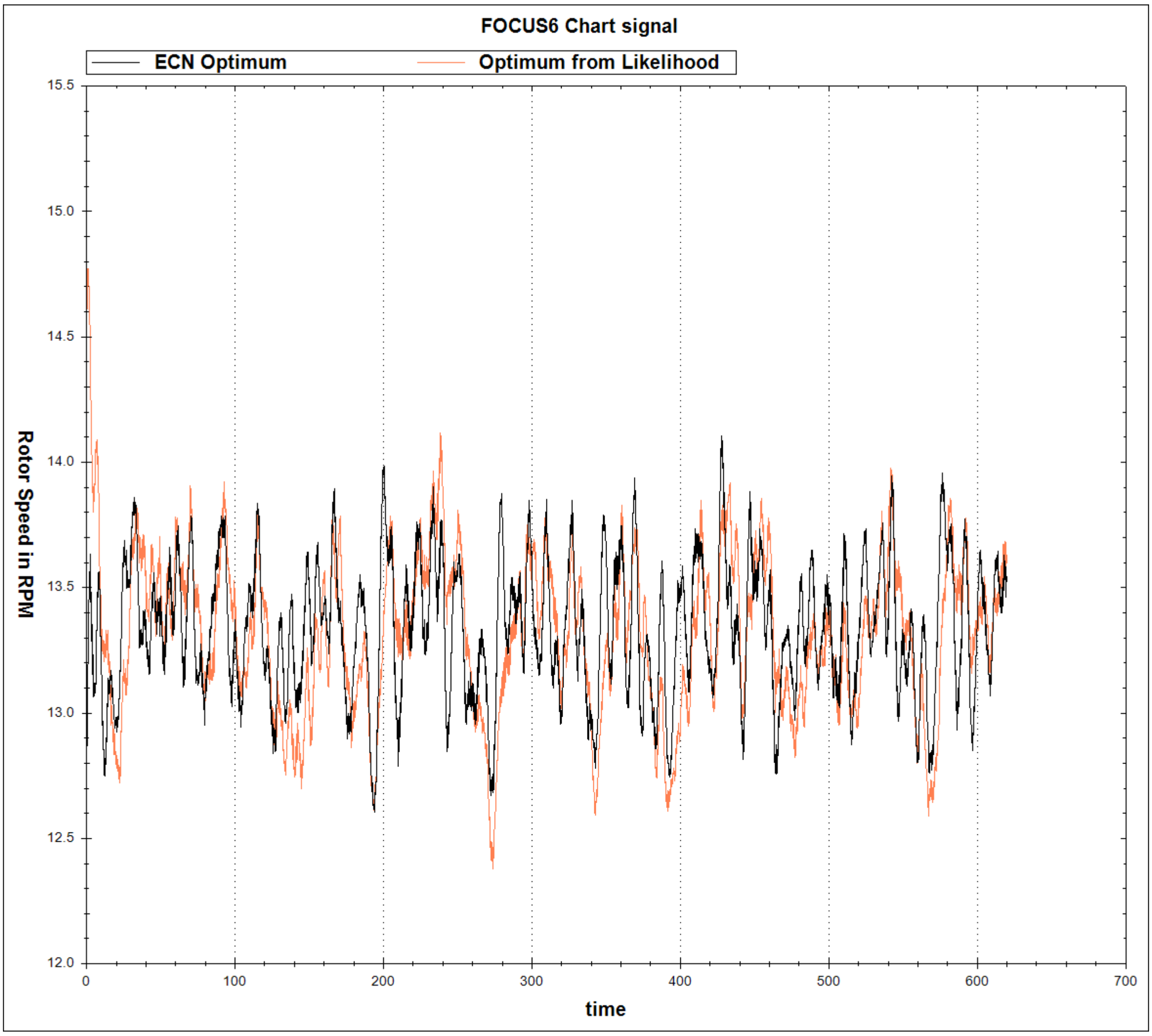


Figure A-5: Rotor speed comparison between ECN Optimum and Optimum from likelihood

A-2 Time series analysis for Extreme Operating Gust

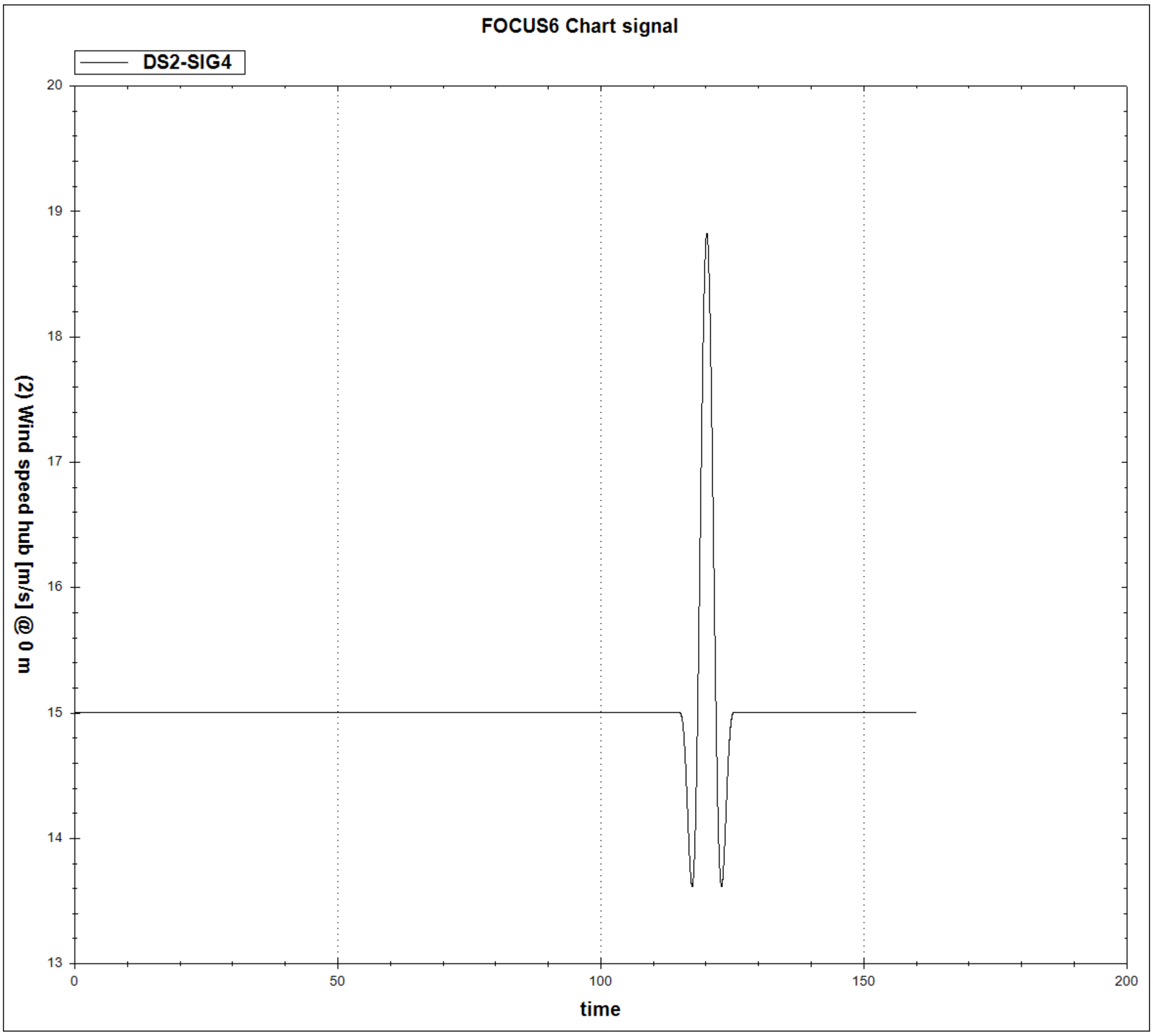


Figure A-6: Extreme Operating Gust used for testing

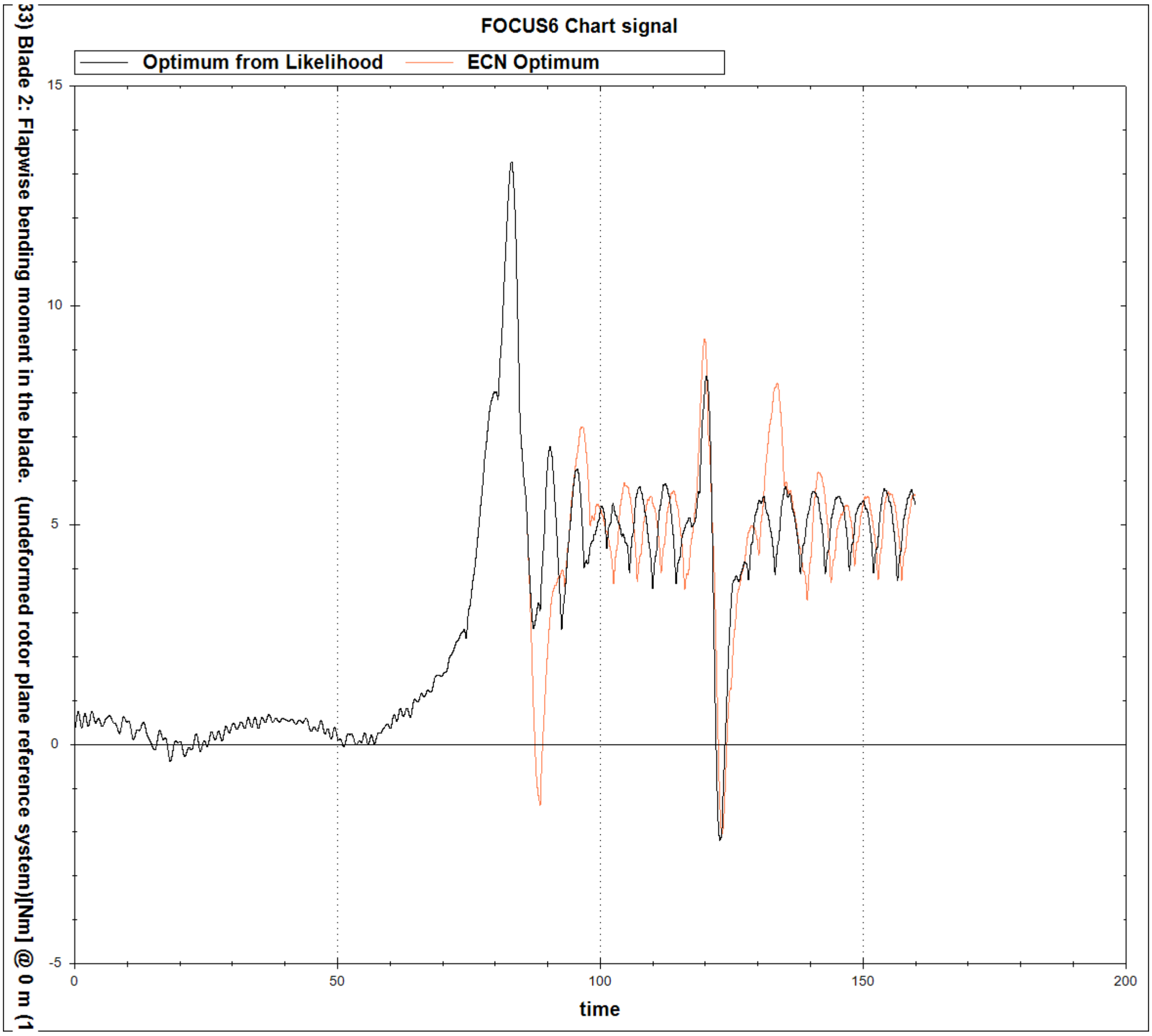


Figure A-7: Blade fatigue load comparison between ECN optimum and optimum from likelihood under disturbance

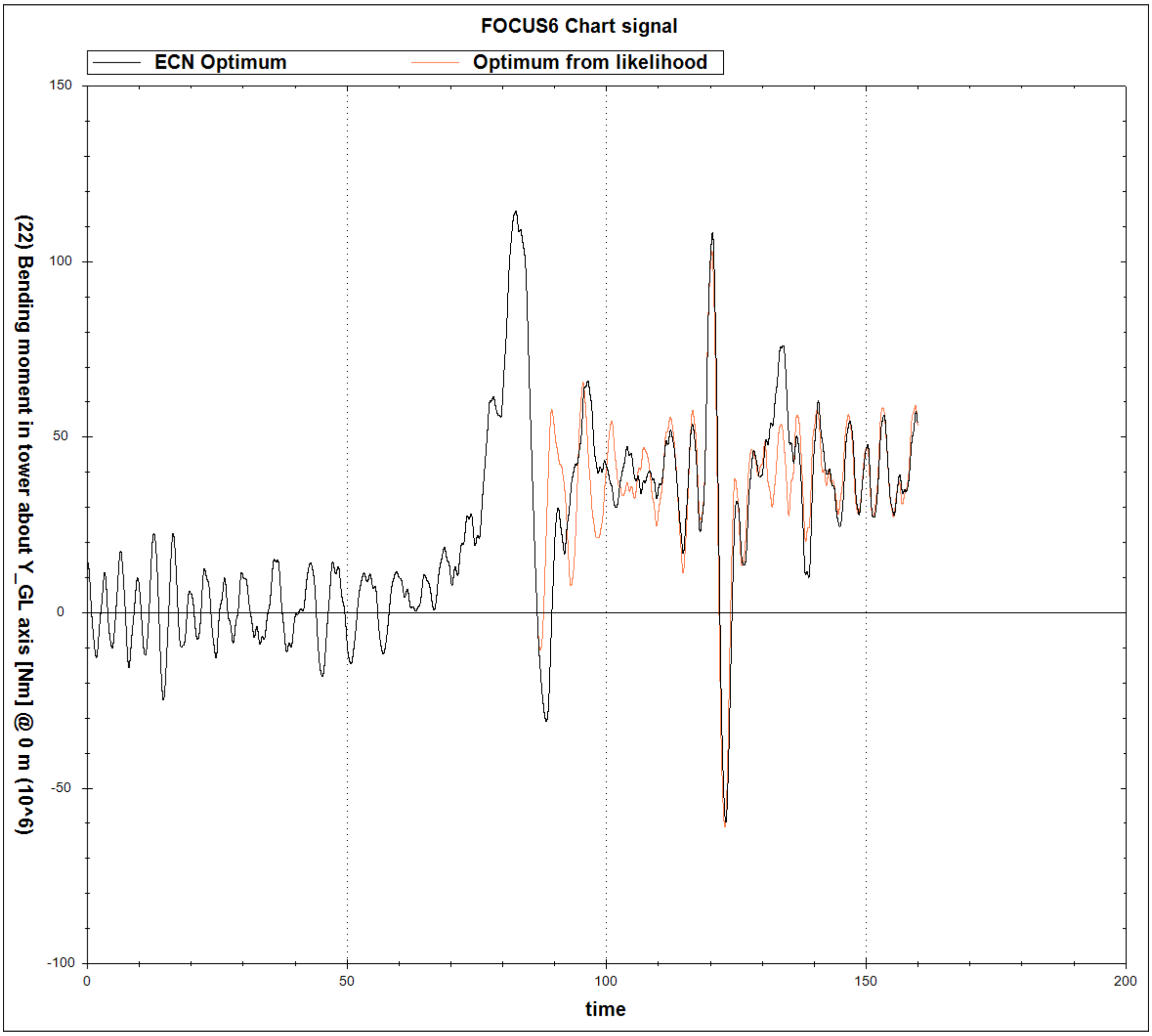


Figure A-8: Tower fatigue load comparison between ECN optimum and optimum from likelihood under disturbance

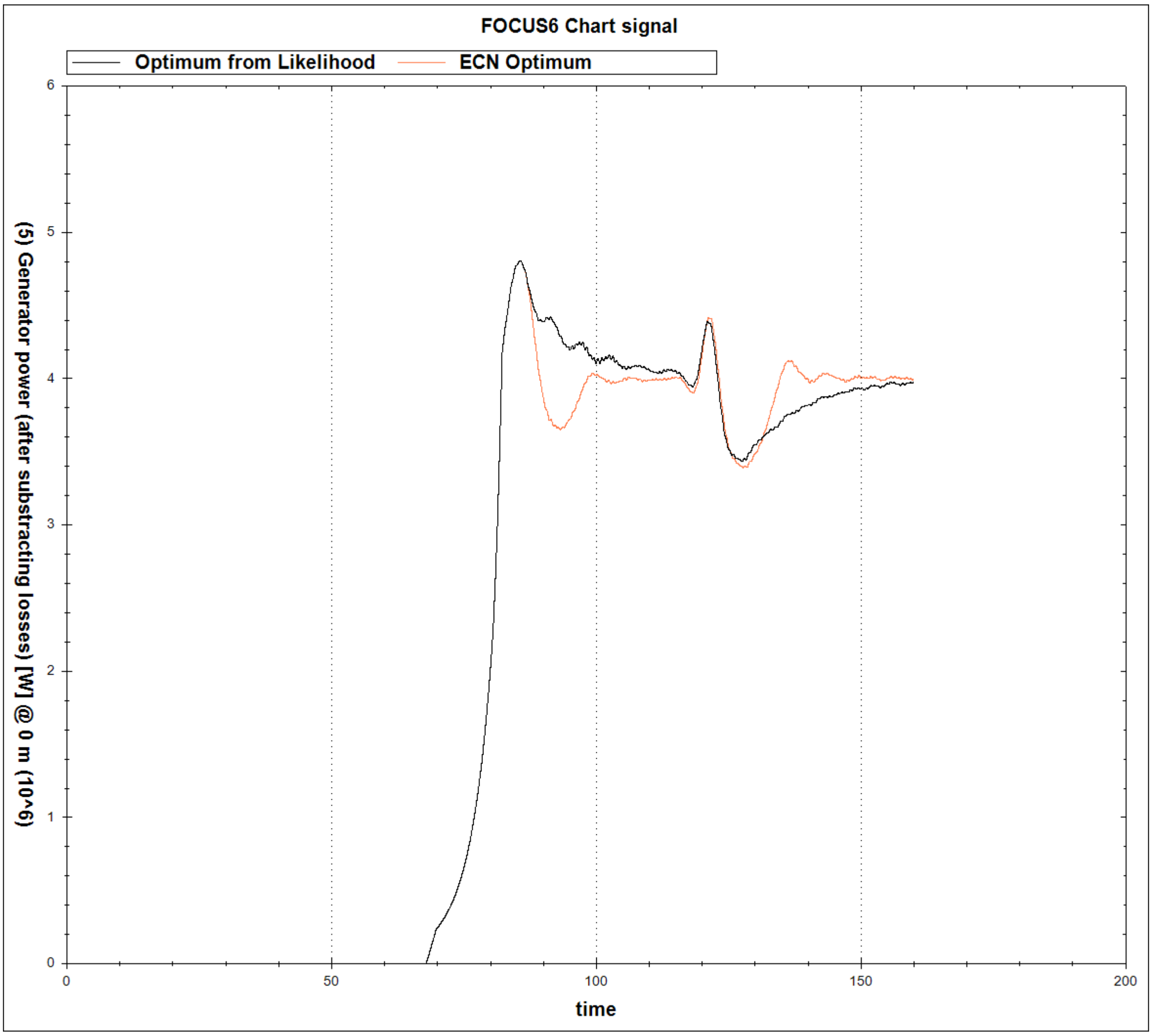


Figure A-9: Comparison of generated power between ECN optimum and optimum from likelihood under disturbance

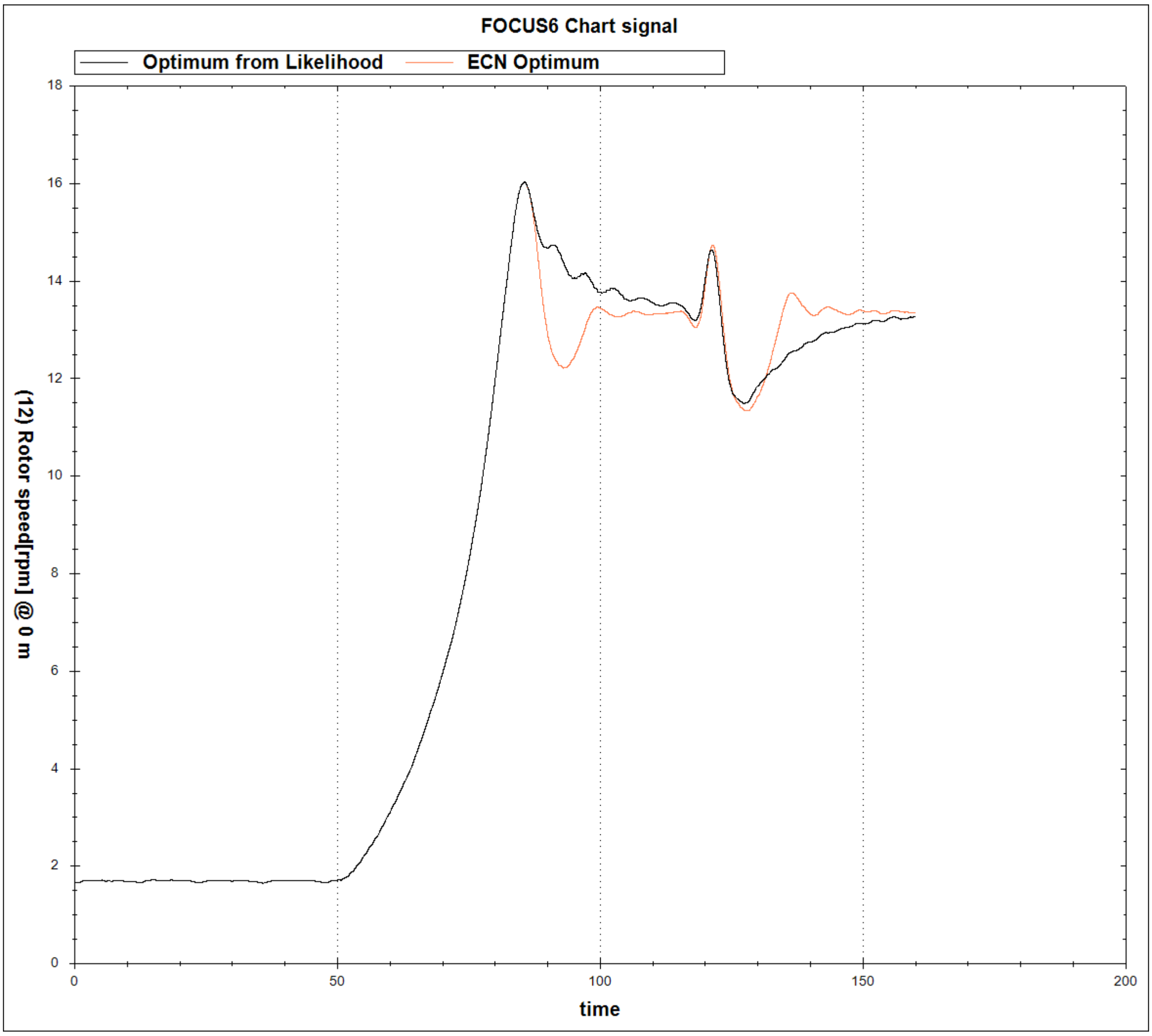


Figure A-10: Comparison of rotor speed between ECN optimum and optimum from likelihood under disturbance

A-3 Time series analysis at near rated Operational mode

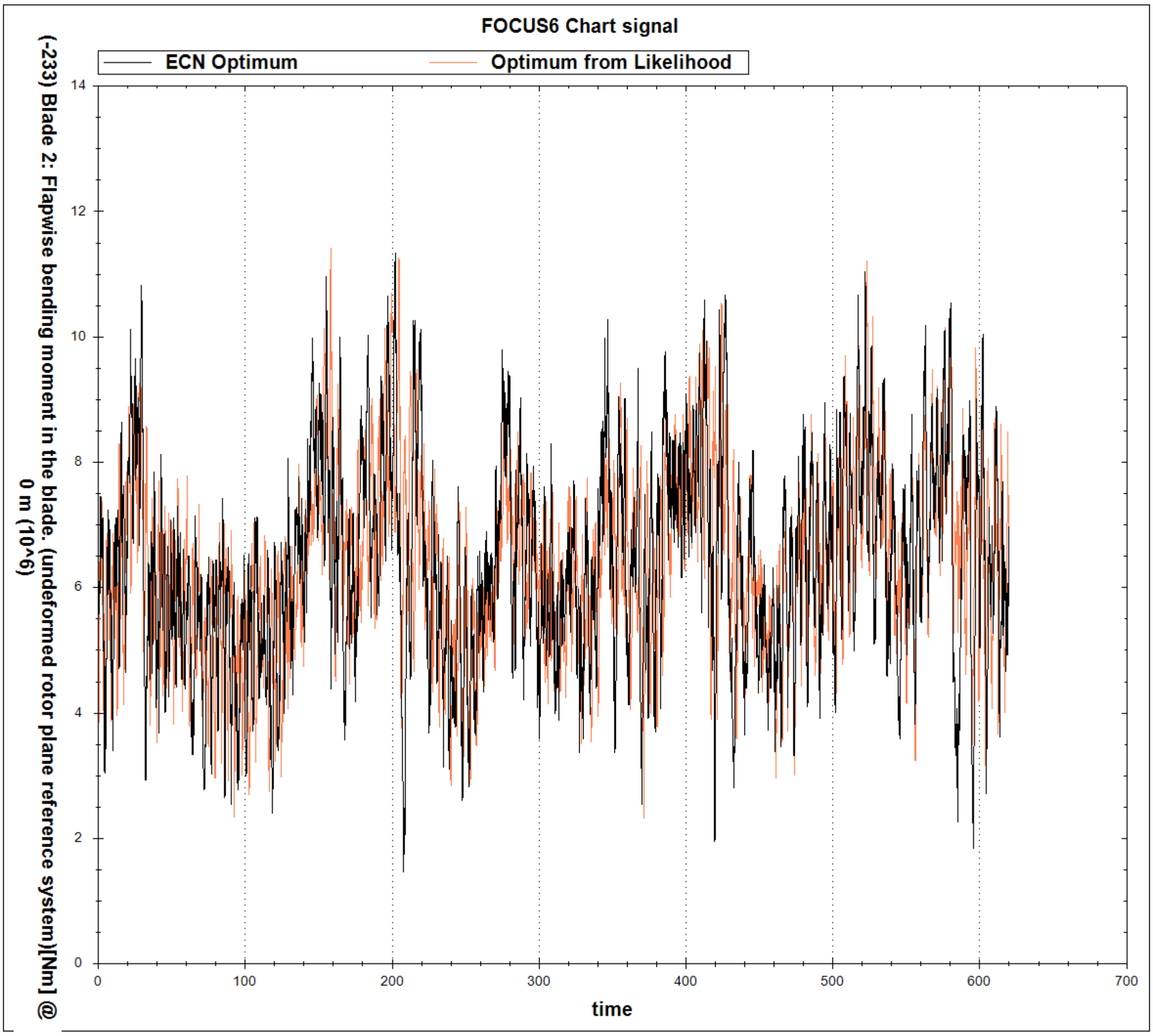


Figure A-11: Blade fatigue load comparison between ECN Optimum and Optimum from likelihood for near rated case

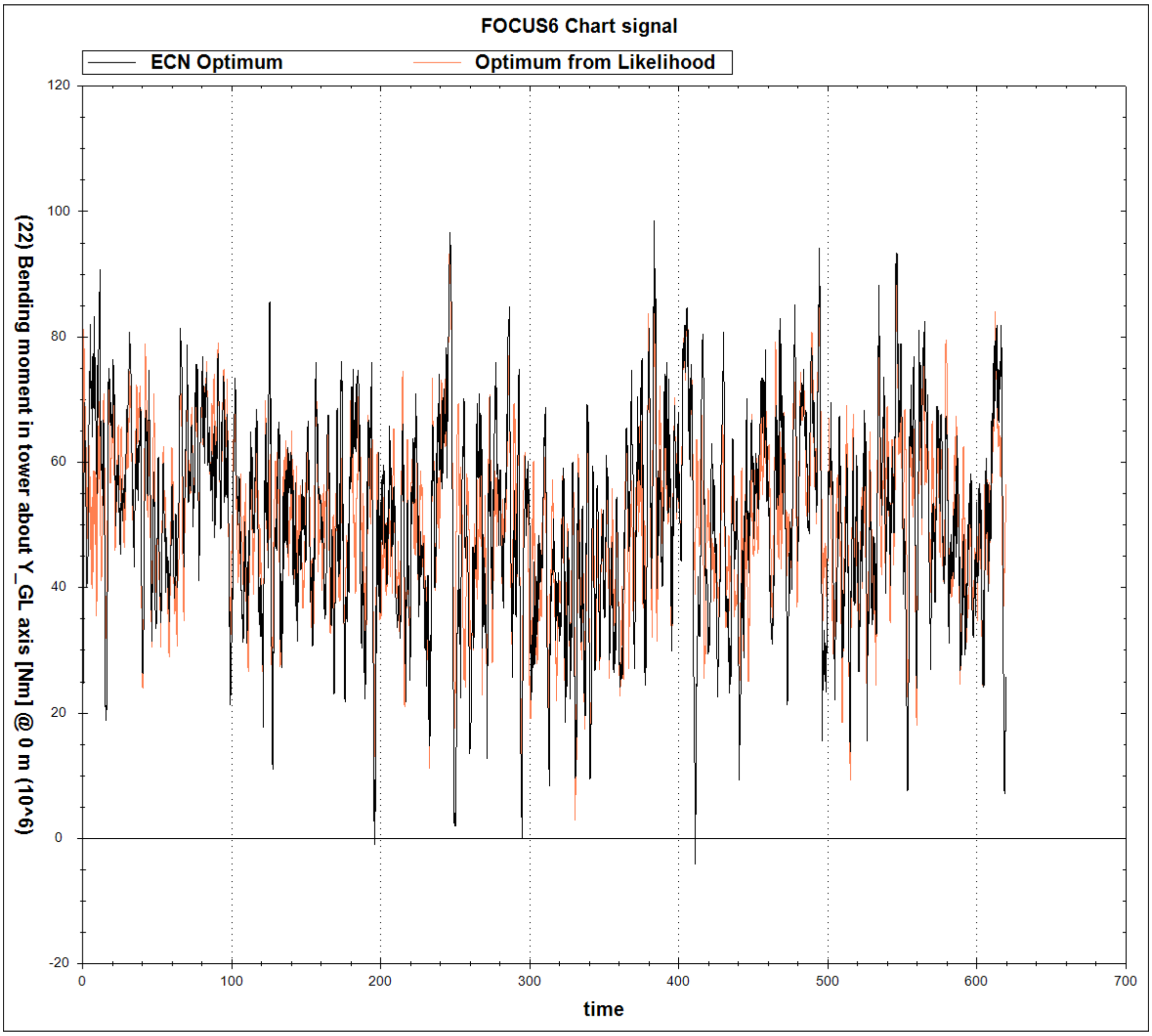


Figure A-12: Tower load fatigue comparison between ECN Optimum and Optimum from likelihood near rated case

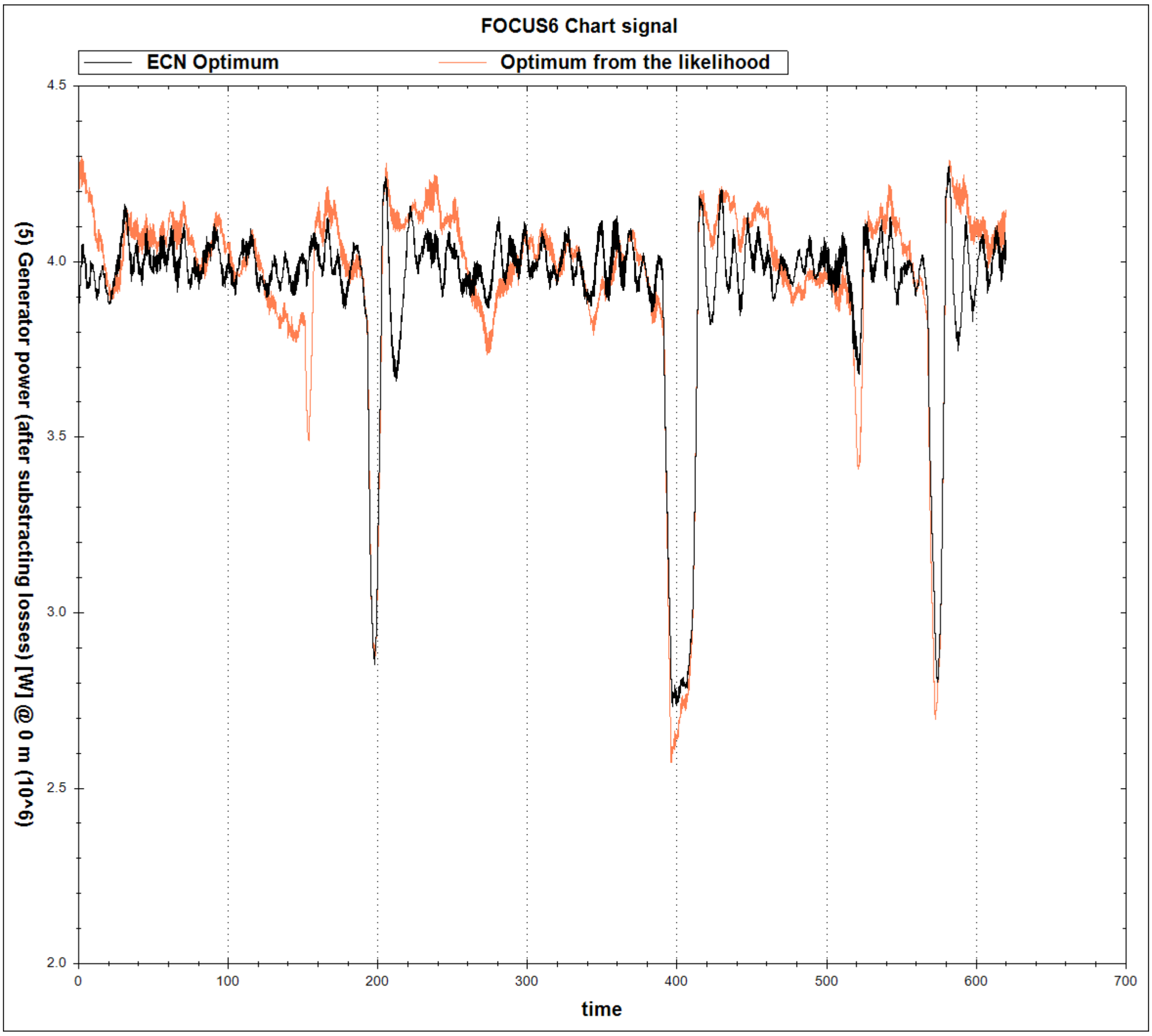


Figure A-13: Comparison of power production between ECN Optimum and Optimum from likelihood for near rated case

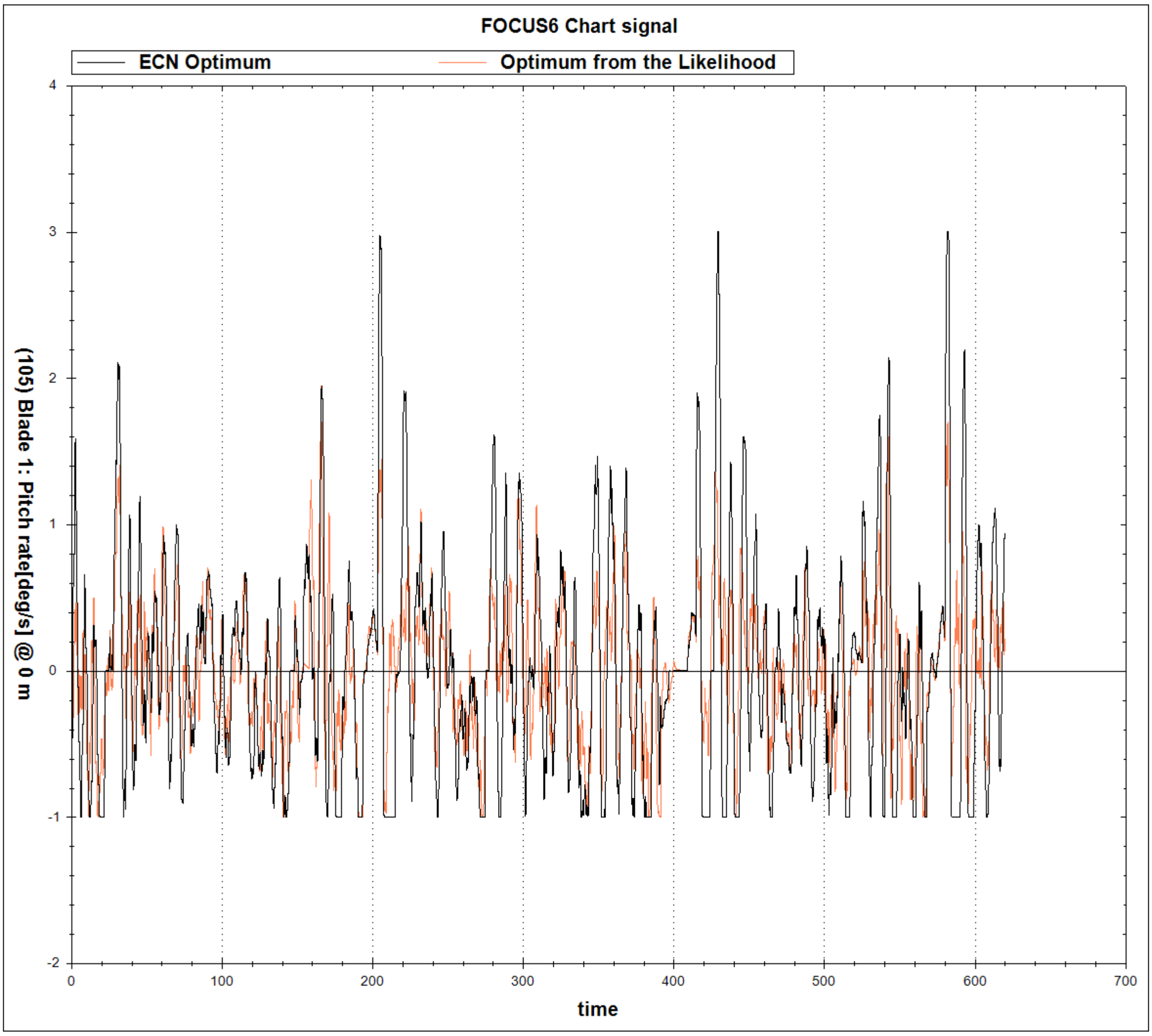


Figure A-14: Pitch activity comparison between ECN Optimum and Optimum from likelihood for near rated case

A-4 Time series analysis with Extreme Event Controller

This section contains test results when the threshold gains are very small and when they are considerably large for a continuous above rated wind speed. ECN uses currently a small threshold gain parameter which I've suggested against in this wind conditions.

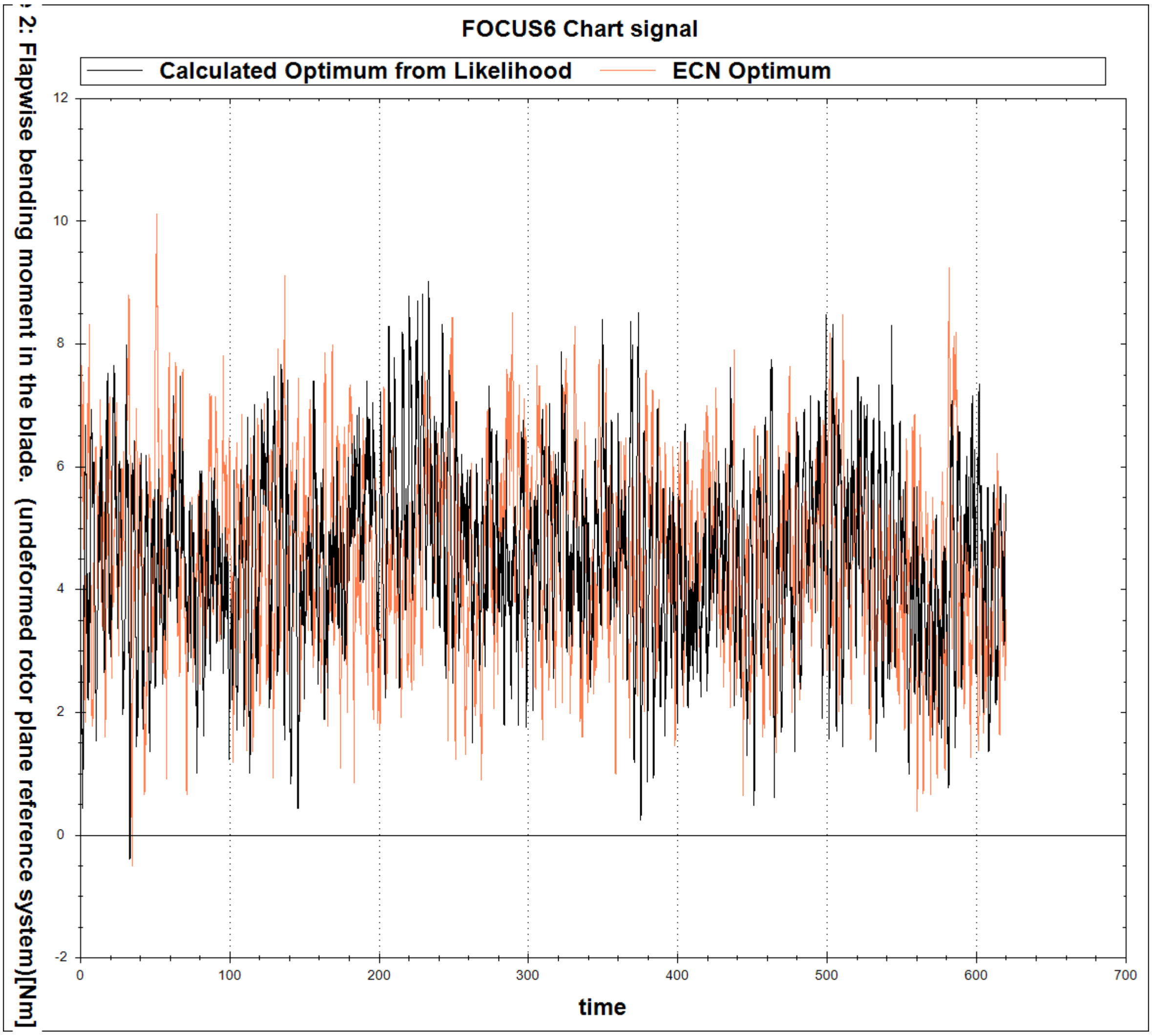


Figure A-15: Blade fatigue load comparison between ECN optimum and optimum from likelihood under disturbance Extreme event Control

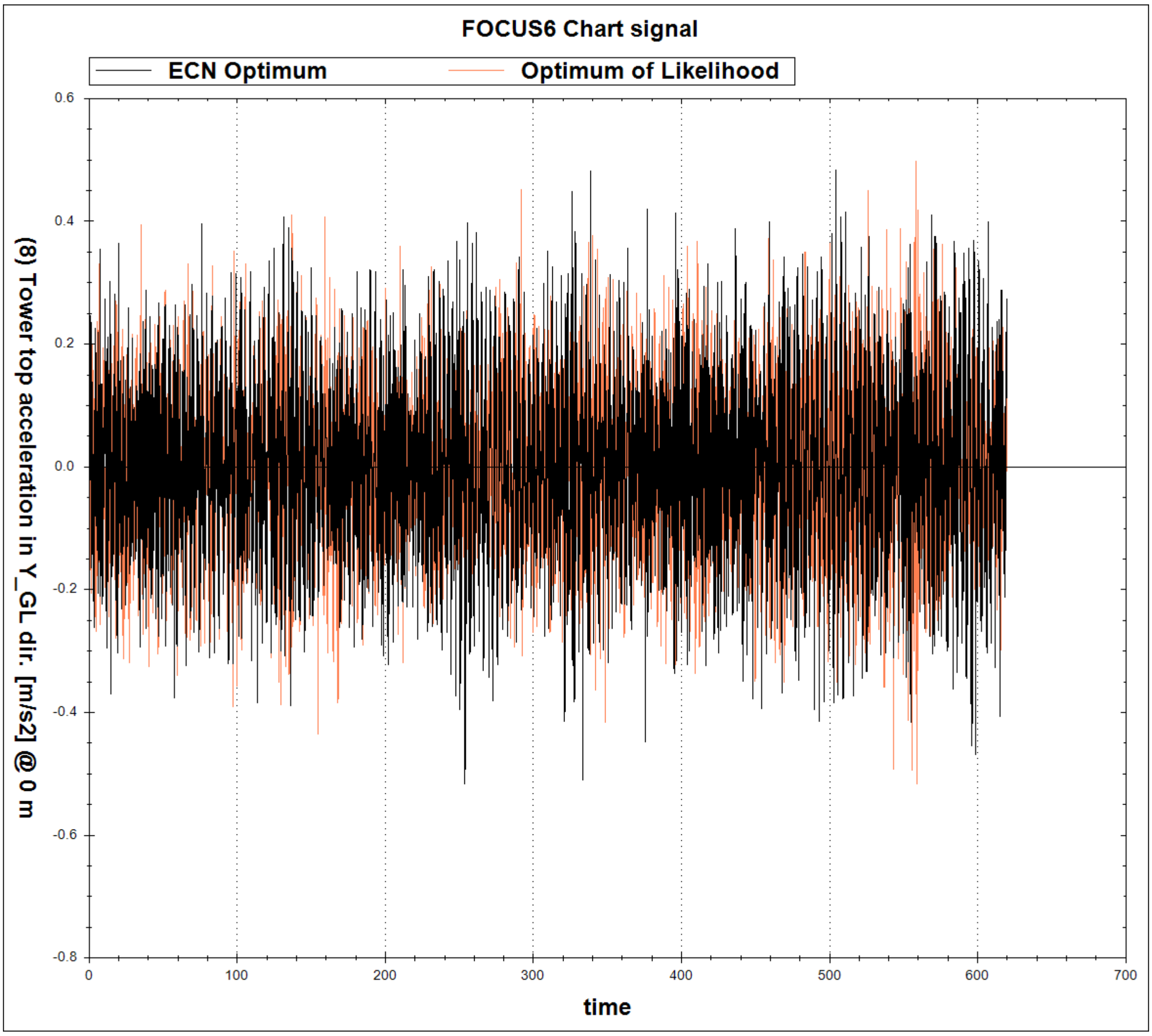


Figure A-16: Tower fatigue load comparison between ECN optimum and optimum from likelihood under disturbance Extreme event Control

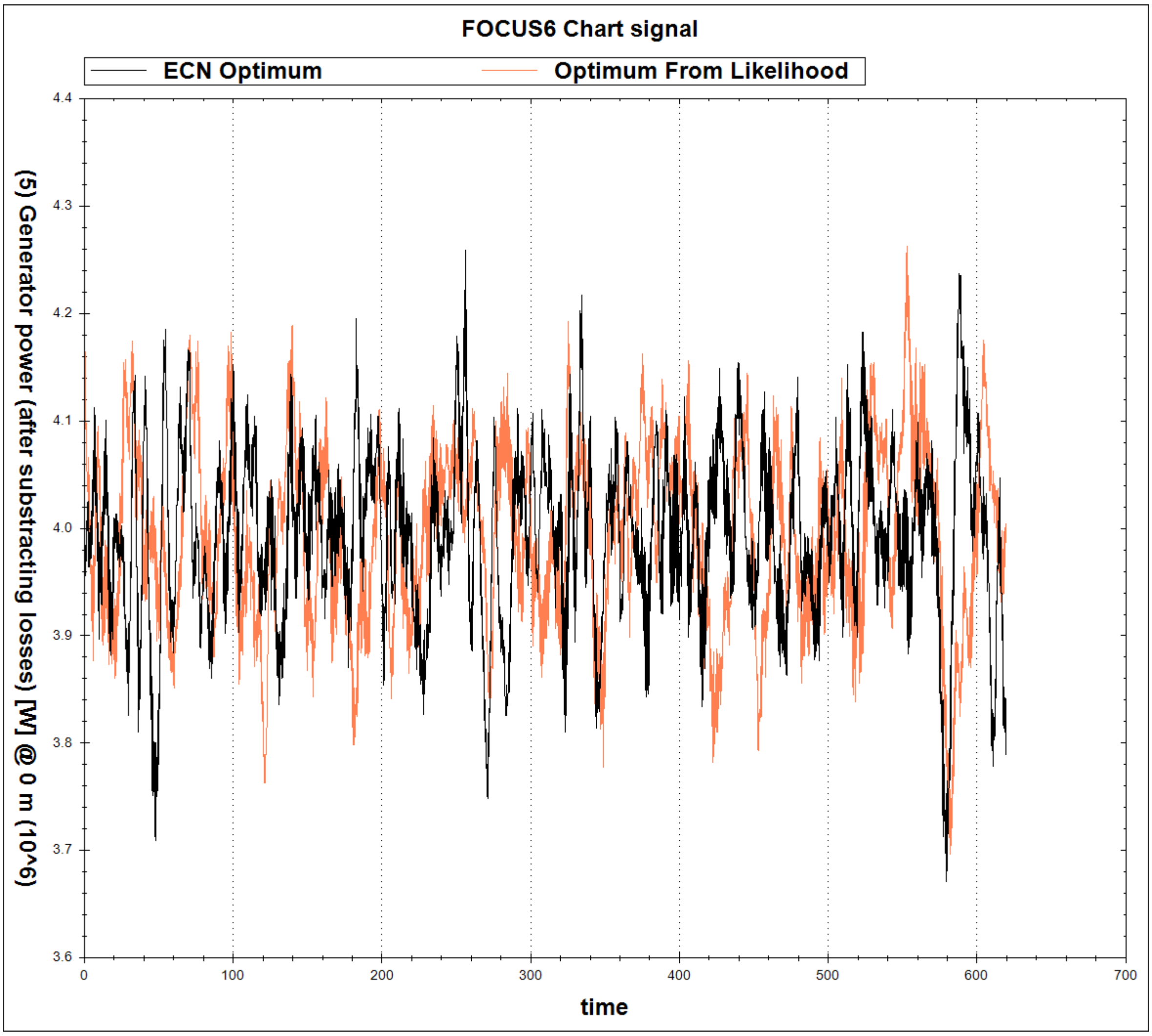


Figure A-17: Comparison of generated power between ECN optimum and optimum from likelihood under disturbance Extreme event Control

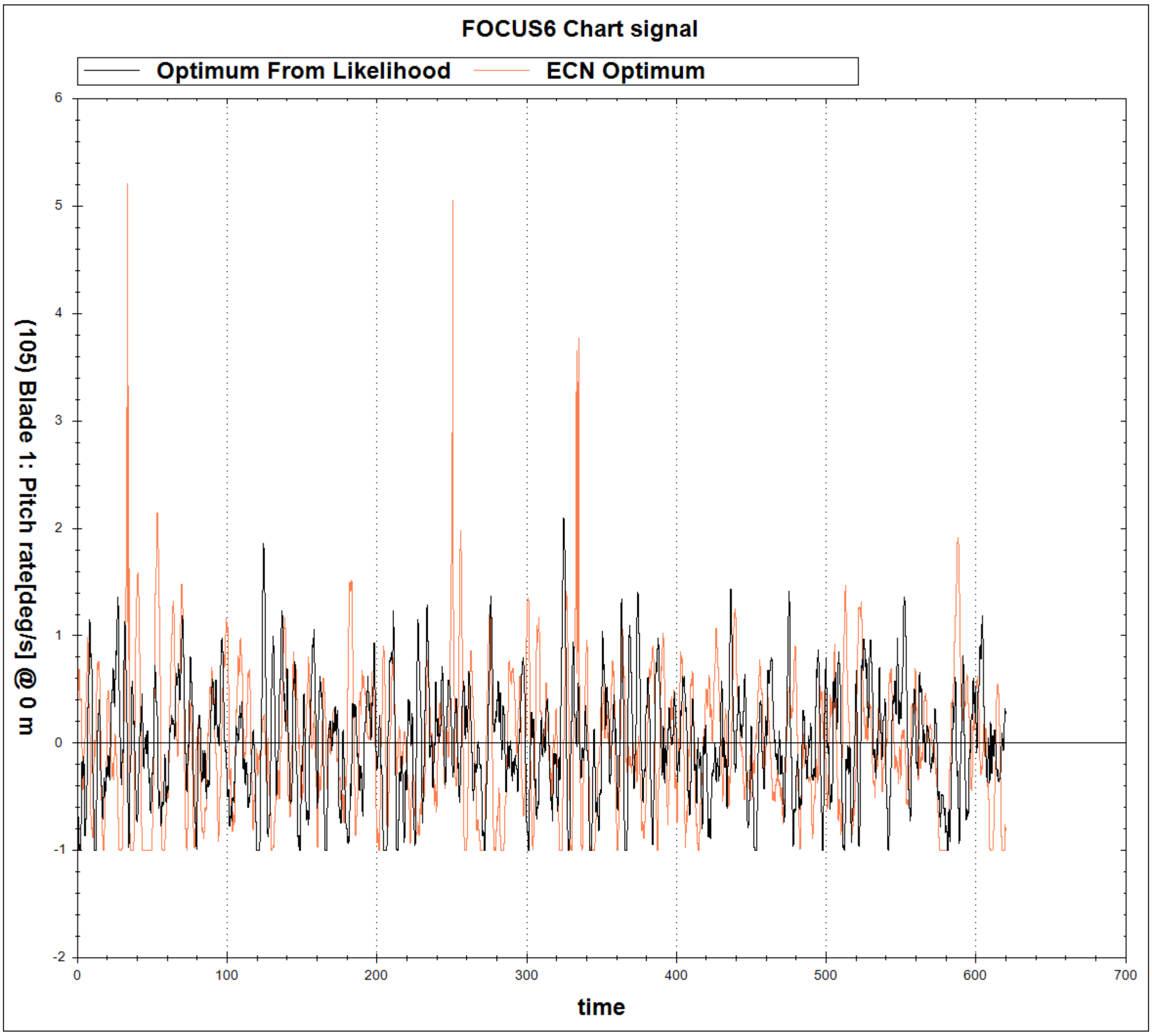


Figure A-18: Pitch activity Comparison between ECN Optimum and Optimum from likelihood for Extreme event Control

Appendix B

Appendix B

This appendix contains information about the 'D4Rel Turbine', whose model is used for simulation in this project.

Parameter	Values
Effective Hub Height (m)	89.80
Rotor Speed (rpm)	13.360
Rotor Diameter (m)	129.876
Number of Circles	18
Number of Points in a Circle	64
Radius of inmost Circle (m)	3.849
Radial distance between circles (m)	3.706
Spectrum Upper Cut-off Frequency (Hz)	12.603
Total Simulation Time (sec)	650
Characteristic Turbulence Intensity (%)	14.00
Actual Turbulence Intensity (%)	15.40
Annual Average Wind Speed (m/s)	10.00
Water Depth (m)	29.6000
Wave Peak Period (sec)	6.000
Significant Wave Height	1.6250
Peak Parameter	3.300
Number of Nodeset	8

Table B-1: Information Table for 'D4Rel Turbine'

Bibliography

[1] Global Wind Energy Council. Global wind report 2014. *Brussels, Belgium*, pages 9–10, 2016.

[2] Monique Maria Hoogwijk. *On the global and regional potential of renewable energy sources*. PhD thesis, 2004.

[3] Chong Ng and Li Ran. *Offshore wind farms: Technologies, design and operation*. Woodhead Publishing, 2016.

[4] E. A. Bossanyi. The design of closed loop controllers for wind turbines. *Wind Energy*, 8(3):149–163, 2000.

[5] Fernando D Bianchi, Hernan De Battista, and Ricardo J Mantz. *Wind turbine control systems: principles, modelling and gain scheduling design*. Springer Science & Business Media, 2006.

[6] Jeff S Shamma. An overview of lpv systems. In *Control of linear parameter varying systems with applications*, pages 3–26. Springer, 2012.

[7] Mahmood Mirzaei, Carlo Tibaldi, and Morten H Hansen. Pi controller design of a wind turbine: evaluation of the pole-placement method and tuning using constrained optimization. In *Journal of Physics: Conference Series*, volume 753, page 052026. IOP Publishing, 2016.

[8] Gijsbrecht Jan Van der Veen. Identification of wind energy systems. 2013.

[9] Carlo Tibaldi, Morten Hartvig Hansen, and Lars Christian Henriksen. Optimal tuning for a classical wind turbine controller. In *Journal of Physics: Conference Series*, volume 555, page 012099. IOP Publishing, 2014.

[10] Edwin van Solingen, Sebastiaan Paul Mulders, and Jan-Willem van Wingerden. Iterative feedback tuning of wind turbine controllers. *Wind Energy Science*, 2(1):153, 2017.

[11] Stoyan Kanev, J Schuurmans, R Rutterman, and E Nguyen. Towards new industrial software for advanced wind turbine control. In *Proceedings of European Wind Energy Conference & Exhibition (EWEA)*, pages 14–17, 2011.

[12] Carl Edward Rasmussen and Christopher KI Williams. *Gaussian processes for machine learning*, volume 1. MIT press Cambridge, 2006.

[13] Alex J Smola and Peter L Bartlett. Sparse greedy gaussian process regression. In *Advances in neural information processing systems*, pages 619–625, 2001.

[14] Joaquin Quiñonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6(Dec):1939–1959, 2005.

[15] Sambu Seo, Marko Wallat, Thore Graepel, and Klaus Obermayer. Gaussian process regression: Active data selection and test point rejection. In *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, volume 3, pages 241–246. IEEE, 2000.

[16] Hildo Bijl. *LQG and Gaussian Process Techniques-For Wind turbine control.* PhD thesis, TU Delft, Netherlands, 2016.

[17] Hildo Bijl, Jan-Willem van Wingerden, Thomas B Schön, and Michel Verhaegen. Online sparse gaussian process regression using fitc and pitc approximations** this research is supported by the dutch technology foundation stw, which is part of the netherlands organisation for scientific research (nwo), and which is partly funded by the ministry of economic affairs. the work was also supported by the swedish research council (vr) via the project probabilistic modeling of dynamical systems (contract number: 621-2013-5524). *IFAC-PapersOnLine*, 48(28):703–708, 2015.

[18] Hildo Bijl, Thomas B Schön, Jan-Willem van Wingerden, and Michel Verhaegen. A sequential monte carlo approach to thompson sampling for bayesian optimization. *arXiv preprint arXiv:1604.00169*, 2016.

[19] Behzad M Shahshahani and David A Landgrebe. The effect of unlabeled samples in reducing the small sample size problem and mitigating the hughes phenomenon. *IEEE Transactions on Geoscience and remote sensing*, 32(5):1087–1095, 1994.

[20] María C Alonso, José A Malpica, and Alex Martínez de Agirre. Consequences of the hughes phenomenon on some classification techniques. In *ASPRS 2011 Annual Conference, Milwaukee, Wisconsin May*, pages 1–5, 2011.

[21] Gervasio Piñeiro, Susana Perelman, Juan P Guerschman, and José M Paruelo. How to evaluate models: observed vs. predicted or predicted vs. observed? *Ecological Modelling*, 216(3):316–322, 2008.

Glossary

List of Acronyms

GPRT	Gaussian Process Regression Technique
MAP	Maximum A Posteriori
GPML	Gaussian Process for Machine Learning
MCMD	Monte Carlo Maximum Distribution
SMC	Sequential Monte Carlo
ACT	Advanced Control Tool
LPV	Linear Parameter Varying
LMIs	Linear matrix inequalities
ECN	Energy Centre Netherlands
KDE	Kernel Density Estimation
EEC	Extreme Event Control
IFT	Iterative feedback tuning

List of Symbols

$\delta\theta_{col}$	Collective pitch angle
$\delta\Omega_r^{int}$	Integrated rotor speed
γ	Torsion angle
$\hat{\Omega}_r$	Rotor speed estimate
$\hat{\Omega}_r$	Rotor speed estimate

λ	Tip-speed ratio
λ	Tip-speed ratio
Ω	Rotor speed
Ω_g	Generator speed
$\Omega_{r,min}$	Minimum rotor speed
Ω_r	Rotor speed
θ	Pitch angle
θ	Pitch angle
$\underline{f}_\star$	Function value of test points
$\underline{f}_\star$	Test points
$\underline{f}_m$	Measurement values
d_t	Damping
F_{dyn}	Force exerted
J_g	Generator inertia
J_r	Rotor inertia
M^i_{tilt}	Tilting Moment
m_t	Tower mass
n_r	Number of rounds
n_r	Number of rounds
s_t	Stiffness
T^i_a	Generator torque
T_c	Coulomb forces
T_{gen}	Generated Torque
T_{gen}	Generator Torque
T_{loss}	Model conversion losses
T_{sh}	Shaft torque
T_v	Viscous forces
u^i_{ax}	Wind speed
x_{fa}	Tower top position
x_{sd}	Tower sideways position
B	Number of blades
i	Per blade
R	Radius of wind turbine