

Investigation and Assessment of the Benefits For Power Systems From Wind Farm Control

Matthew D. Cole, Professor William E. Leithead & Dr David Campos-Gaona

Industrial partner - SgurrControl

University of Strathclyde - Wind and Marine Energy Systems CDT

2nd December 2023

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

The copyright of this thesis belongs to the author under the terms of the United Kingdom Copyright Acts as qualified by University of Strathclyde Regulation 3.50. Due acknowledgement must always be made of the use of any material contained in, or derived from, this thesis.



“All models are wrong, but some are useful.” George E. P. Box
Dedicated to my parents Andrea and Michael without whom none of this would
have been possible.

Abstract

As wind turbines are increasingly situated in large arrays offshore, connected to power grids by a single long cable, it is necessary to consider the operation of the whole wind farm as a single plant rather than as a series of individual units. To achieve this, the development of advanced wind farm modelling software is required to test and evaluate new control strategies for wind farm operation. This thesis considers the use of Strathfarm, The University of Strathclyde's in-house wind farm modelling software, presenting novel wind farm control algorithms which significantly reduce the fatigue of wind turbine towers and wind turbine blades. The thesis also further develops Strathfarm in two key areas, presenting improvements to the modelled wakes and also details the development of a novel power system model. The power system model can be used to show the efficacy of previously developed dispatch algorithms for wind farms to support power grids.

Contents

Abstract	iii
Nomenclature	x
0.1 Wind Farm Control Nomenclature	xi
0.2 Wake Modelling Nomenclature	xii
0.3 Power System Modelling Nomenclature	xiii
List of Figures	xvi
List of Tables	xxiv
Preface/Acknowledgements	xxvii
1 Introduction	2
1.1 Motivation	3
1.2 Publications and Research Outputs	3
1.2.1 Individual contribution	5
1.3 Summary of Thesis	5
2 Context	8
2.1 A History of Windmills	8
2.1.1 Electricity Generating Wind Turbines	9
2.1.2 Wind Farms as Power Plants	13
3 Background	15
3.1 Operational Strategy For A Wind Turbine	15

Contents

3.2	Wind farm Modelling Software	17
3.3	Wind Farm Control	19
3.3.1	Why use control at the wind farm level?	19
3.3.2	Aims of wind farm control	19
3.3.3	Maximising Power Output	20
3.3.4	Minimizing Wind Turbine Component Fatigue	20
3.4	Aims and objectives	22
4	A Literature Review of Wind Farm Control for Grid Support	23
4.1	Introduction	23
4.2	Review of Ancillary Services with Participation of Wind Power	24
4.2.1	Case Study: National Grid	25
4.2.2	Case Study: EirGrid	26
4.2.3	Case Study: Energinet	28
4.2.4	Discussion	30
4.3	Review of Ancillary Services without Participation of Wind Power	31
4.3.1	Black Start	31
4.3.2	Inertial Response	32
4.3.3	Future Participation	32
4.3.4	Discussion	33
4.4	Virtual Synchronous Machines	33
4.5	Wind Farm Set-Point Power Control	34
4.6	Optimisation-Based Dispatch Algorithms	37
4.7	Stored-Energy-Based Solutions	40
4.8	Other Approaches	47
4.9	Discussion	48
4.9.1	Structural Modelling	49
4.9.2	Farm Size Consideration	49
4.9.3	HVDC vs. AC Connection	50
4.10	Concluding Remarks	50

Contents

5	Development of a Farmwide Power Set-Point Controller for Strathfarm	51
5.1	What is Strathfarm?	51
5.1.1	The Power Adjusting Controller	53
5.2	Why Set-point Control?	57
5.2.1	Wind Farm Controller	57
5.2.2	Limitations and Restrictions to be Considered	59
5.2.3	Additional Practical Restrictions	61
5.3	Development of a Wind Farm Control Strategy for Set-point Power Provision	62
5.3.1	Traffic Light Flag Based Allocation Without Anti-Windup	62
5.3.2	Inclusion of First Order Anti-Windup for the Farm Wide Power	64
5.3.3	Inclusion of Second Order Anti-Windup for the Farm Wide Power	66
5.3.4	Discussion of Turbine Level Windup in Curtailment Requests Despite Farm Level First and Second Order Anti-Windup	68
5.3.5	Inclusion of First and Second Order Anti-Windup for the Turbine Level Curtailment Requests	71
5.3.6	Additional Wind Farm Controller Considerations	76
5.4	OVERRATING A WIND FARM	76
5.4.1	OVERRATING A TURBINE WITH A BELOW RATED WIND SPEED	80
6	Designing a Wind Farm Controller to Reduce Wind Turbine Fatigue	83
6.1	Fatigue Estimation	83
6.2	Existing Wind Turbine Bending Moments	86
6.2.1	Curtailment to Reduce Bending Moments	91
6.3	Steady State Loads	93
6.4	The wind speed estimation from the PAC	97
6.5	Wind Farm Curtailment Allocation Strategies	97
6.5.1	Equal Curtailment	98
6.5.2	Power Estimation Control	99
6.5.3	Intelligent Tower Control	103
6.6	Validation Methodology	112
6.6.1	Validation	112

Contents

6.6.2	Weibull Comparison	112
6.6.3	Damage Equivalent Loads	112
6.6.4	Choice of Farm Power Set-Points	113
6.6.5	Turbine Level Results Methodology	113
6.6.6	Power Spectral Density Analysis	114
6.6.7	The Impact of the Wind Rose on Turbine Tower Fatigue	114
6.7	Unweighted Farm Level Simulation Results	115
6.7.1	Upper Curtailment Level Power Curve Results	115
6.7.2	Middle Curtailment Level Power Curve Results	125
6.7.3	Lower Curtailment Level Power Curve Results	129
6.7.4	Weibull Weighted DEL Changes for Weibull with mean wind speeds of 8, 9 and 10 m/s	132
6.7.5	Turbine Level Results	137
6.8	Concluding Remarks	139
7	Development of a dynamic power system model in Strathfarm	140
7.1	The Model Development Process	140
7.2	The Solver Code	142
7.2.1	The Jacobian Matrix	147
7.2.2	The Levenberg-Marquardt Method	149
7.2.3	Non-linear Solver Algorithm	150
7.3	The Solver S-function in Strathfarm	152
7.3.1	Example Solver Results	153
7.4	The Power System Model	156
7.5	The Model Between the DLL Blocks	157
7.5.1	The Mechanical Model	159
7.6	The Electrical Model	162
7.6.1	The Excitation System	166
7.7	Power System Linking Model	168
7.8	The Wind Farm Controller	171
7.8.1	Drop Controller Example Results	174

Contents

7.8.2	Example Inertial Response	183
7.9	Concluding Remarks	188
8	Development and Inclusion of a Gaussian Wake Model in Strathfarm	189
8.1	Review and History of Engineering Models of Wakes	189
8.1.1	The Jensen Wake Model	189
8.1.2	The Frandsen Wake Model	191
8.1.3	The Gaussian Wake Model	192
8.2	The Existing Strathfarm Wake Model	194
8.2.1	The Wake Centre Calculation	194
8.2.2	The Wake Diameter Calculation	195
8.2.3	The Wake Deficit Calculation	195
8.3	Development of a Gaussian Wake Model for Strathfarm	195
8.4	Implementation in Strathfarm	199
8.4.1	Example Plots	201
8.5	Comparison of Wake Models	206
8.5.1	1 m/s wake movement	206
8.5.2	2 m/s wake movement	208
9	Conclusions	210
9.1	Future Work	211
9.1.1	Wind Farm Control	211
9.1.2	Future PAC developments	213
9.1.3	Wake Model	213
9.1.4	Power System Model	213
	References	215
A	Using Rotor Inertia as Stored Energy in Below Rated Wind Farms to Provide Primary Frequency Response	228
A.1	Introduction	228
A.2	Methodology	231

Contents

A.2.1	Wind Turbine level Approach	231
A.2.2	Farm Level implementation	235
A.3	Results	237
A.3.1	5% Turbulence Intensity	238
A.3.2	10% Turbulence Intensity	239
A.4	Conclusions	241
B	Power system Model Discretization	242
B.1	Discretization of the Governor	242
B.1.1	Actuation	244
B.1.2	The Governor's 2nd Actuator	245
B.2	Discretization of the Generator Excitation System	246
B.2.1	Low Pass Filter	246
B.2.2	Main Regulator	246
B.2.3	Lead Lag Compensator	247
B.2.4	Damping	247
B.2.5	Exciter	248
C	Alternative model for the Gaussian wake	249
C.1	Bessel Function Derivation	250
C.1.1	Even Terms	251
C.1.2	Odd Terms	251
C.2	Continuation	251
C.3	Gamma Substitution	252
C.4	Final Equations	253
C.5	Example Results	253
D	Variables For Power System Model S-function External Tuning	256
E	Wind Farm Distribution Controller C++ Code	260
F	Wind Farm Network Controller C++ Code	299

Nomenclature

Due to the multidisciplinary nature of this thesis it is not possible to follow all conventions for variable names within each field of engineering without having conflicting naming. While it is clear from the surrounding information what each variable means in context for completeness duplication will be avoided. Where the conventions conflict a subscript is used in each convention for clarity. For convenience to the reader the nomenclature is listed in three sections: wind farm control, wake modelling and power system modelling.

0.1 Wind Farm Control Nomenclature

This nomenclature lists the variables used in chapter five, chapter six and appendix A .

Symbol	Description
f_i	the status flag of the i th turbine in a wind farm
J	mechanical inertia of a turbine generator
k_i	the integral gain of the wind farm controller
k_p	the proportional gain of the wind farm controller
N_t	the total number of turbines in a wind farm
P_d	the demanded power out of a wind farm
$P(t)$	The power output of a wind farm
ΔP_{NB_i}	The preallocated curtailment of the i th turbine in a wind farm for the intelligent control strategy
ΔP_{farm}	The requested change in power output of a wind farm
ΔP_i	The requested change in power output of the i th turbine in a wind farm
$R_{WT}(v)_i$	a wind speed based reserve of power for a wind turbine for the strategy used in appendix A
$R_{WF}(v)_i$	a wind speed based reserve of power for a wind farm for the strategy used in appendix A
S_i	The PAC flags sent by the i th turbine in a wind farm
T_s	A timestep in the wind farm controller (normally 0.01 seconds)
v_i	the estimated wind speed of the i th turbine in a wind farm
\dot{x}_{limit}	The maximum change of curtailment of a turbine in Strathfarm for a single timestep.

0.2 Wake Modelling Nomenclature

This nomenclature lists the variables used in chapter eight and appendix C.

Symbol	Description
a	the axial induction factor
a_{nP}	Fourier coefficients in the effective wind speed model
b_{nP}	Fourier coefficients in the effective wind speed model
β	the wake factor
c	weighing coefficient
C_t	The coefficient of thrust
def_j	wake deficit at the j th turbine
l	the lateral offset of the wake from a turbine centre
r_0	the initial wake radius
R	rotor radius
r	the radius of the wake at a downstream distance of x
u	the unimpeded wind speed
v	the wind speed at a distance x downstream
v_0	the wind speed behind the upwind turbine
WD_j	the wake diameter of the j th wake

0.3 Power System Modelling Nomenclature

This nomenclature lists the variables used in chapter seven and appendix B.

Symbol	Description
\bar{d}	the incremental change to the solution to the nonlinear solver
E_0	the grid voltage
$e_{\zeta d}^g$	the d component of the filter bus voltage from the ζ th turbine
$e_{\zeta q}^g$	the q component of the filter bus voltage from the ζ th turbine
$F(x)$	the functions of the nonlinear solver
H	the grid inertia
H_k	the Jacobian of $F(x)$
i_d	the d component of the current
i_q	the q component of the current
i_{fd}	the current of the field winding
i_{fk}	the d component of the current of the damper winding
i_{kq}	the q component of the current of the damper winding
$i_{d_{sum}}$	d component of the sum of the currents from the wind farm
$i_{q_{sum}}$	q component of the sum of the currents from the wind farm
$i_{c\zeta d}^g$	the d component of the current flowing from the ζ th turbine
$i_{c\zeta q}^g$	the q component of the current flowing from the ζ th turbine
$i_{t\zeta d}^g$	the d component of the current flowing to the network from the ζ th turbine
$i_{t\zeta q}^g$	the q component of the current flowing to the network from the ζ th turbine
i_{2d}	the d component of the current flowing from the load to the ground
i_{2q}	the q component of the current flowing from the load to the ground
L_n	the inductance of the AC connection from the farm to the grid
L_{node}	the capacitance of the node
L_t	The inductance of the transmission line from the wind turbine to the offshore network PCC
L_c	The inductive component of the phase reactor of the converters
μ	the tuning parameter of the non-linear solver

Chapter 0. Nomenclature

Symbol	Description
ω	the grid frequency
$d\omega_1$	the incremental change of the rotational speed of the generator model in the power system model
$d\omega$	the deviation from the synchronous speed in the power system model
P_m	the mechanical power in the power system model
P_ζ	The power output at the ζ th converter
P_{demand}	the modelled power demand in the grid
ψ_d	the d component of the flux linkage
ψ_q	the q component of the flux linkage
ψ_{fd}	the flux linkage of the field winding
ψ_{fk}	the d component of the flux linkage of the damper winding
ψ_{kq}	the q component of the flux linkage of the damper winding
R_s	the resistance of the stator
R_{fd}	the resistance of the field winding
R_{kd}	the d component of the resistance of the damper winding
R_{kq}	the q component of the resistance of the damper winding
R_{node}	the resistance of the node
R_c	The resistance component of the phase reactor of the converters
R_c	The resistance component of the phase reactor of the converters
R_t	The resistance of the transmission line from the wind turbine to the offshore network PCC
T_1	time constant in the governor controller
T_2	time constant in the governor controller
T_3	time constant in the governor controller
T_4	time constant in the governors first actuator
T_5	time constant in the governors first actuator
T_6	time constant in the governors second actuator
T_e	the electrical torque in the power system model
T_a	time constant for the main regulator
T_c	time constant for the lead lag compensator
T_b	time constant for the lead lag compensator
t_{fd}	time constant for the excitation system damping
T_{ex}	time constant for the exciter
t_r	time constant for the exciter low pass filter

Chapter 0. Nomenclature

Symbol	Description
v_d	the d component of the voltage at the node.
v_q	the q component of the voltage at the node.
V_{base}	the voltage of the power system
V_ζ	the bridge voltage of the ζ th turbine
$v_{\zeta d}^g$	the d component of the bridge voltage from the ζ th turbine
$v_{\zeta q}^g$	the q component of the bridge voltage from the ζ th turbine
V_d	the d component of the voltage
V_q	the q component of the voltage
V_{fd}	the voltage of the field winding
x^k	The k th iteration of the solution in the nonlinear solver

List of Figures

2.1	left:A European four sail windmill design[105]. Right: Mill in Sint Hubert, Netherlands [30].	8
2.2	An example American style windpump [78]	9
2.3	James Blyth’s wind turbine [31]	10
2.4	The Richborough 1MW Wind Turbine. Credit Jock Blakley	11
2.5	The increasing size of wind turbines over time from [88]	12
2.6	Wind turbines at the Whitelee Windfarm near Glasgow. Own work.	13
3.1	The modes of operation for the Supergen 5MW wind turbine model.[3]	16
3.2	An example of a typical wind turbine power curve.[21]	17
4.1	(a) The limited sensitivity mode of droop control from [49]. (b) Sensitive mode of droop control from [49].	25
4.2	(a) Mandatory droop requirements from the Irish grid [33] at dangerously low frequencies. (b) Mandatory droop requirements from the Irish grid at dangerously high frequencies.	27
4.3	Droop requirements from the Irish grid for when the frequency sensitive mode is active.	28
4.4	The mandatory droop control required at dangerously high frequencies from the Danish grid code [34].	29
4.5	The plots shown here are the sensitive mode of droop control in the Danish grid from Danish grid code [34] The plots show (a) a case with low levels of reserve; (b) a higher level of reserve.	30
4.6	The generic form of the wind farm control architecture as shown in [100]. . . .	35

List of Figures

4.7	An example of the different responses to a grid frequency event [94].	41
4.8	A diagram of the proposed controller infrastructure proposed in [59].	44
4.9	Offshore reference circuit proposed by [82].	46
5.1	A schematic diagram showing how each section of Strathfarm connects together.	52
5.2	The general form of the Power Adjusting Controller (PAC) [99]	53
5.3	The wind farm controller layout used in Strathfarm.	54
5.4	The traffic light boundaries for the PAC in the Supergen 5MW Wind Turbine Model.	56
5.5	A generic form block diagram of the wind farm controller.	60
5.6	The layout of the wind farm used for the anti-windup examples presented in this chapter.	62
5.7	Example farm power output without any anti-windup with set-point control activated at the red dashed line at after 200 seconds.	63
5.8	Example farm requested curtailment without any anti-windup.	64
5.9	Example farm requested curtailment without any anti-windup compared to 1st order farm level anti-windup.	65
5.10	Example farm requested curtailment without any anti-wind up compared to 1st order farm level anti-windup.	66
5.11	Example farm requested curtailment with only first order anti-windup compared to when first and second order farm level anti-windup are included. . . .	67
5.12	Example farm requested curtailment with only first order anti-windup compared to when first and second order farm level anti-windup are included. . . .	68
5.13	Example farm turbine level curtailment requests and the total number of activate rate flags.	69
5.14	Example farm sum of each type of traffic light flag.	70
5.15	Histogram of the number of rate flags active in the farm.	71
5.16	Example farm requested curtailment with anti windup applied at the farm level compared to when it is apply at both the farm and turbine level.	73

List of Figures

5.17	Example farm requested curtailment with anti windup applied at the farm level compared to when it is apply at both the farm and turbine level.	74
5.18	The sum of the traffic light flags when anti-windup is applied at both the farm and turbine level.	75
5.19	Example farm with an overrated level of power output set-point.	77
5.20	The Torque speed trajectories of the wind turbines in normal operation	78
5.21	The Torque speed trajectories of the wind turbines in overrated operation. It should be noted that the black limits have been ignored in this simulation for demonstration purposes.	78
5.22	The sum of the traffic light flags in overrated operation	79
5.23	The power output of the overrated wind turbines in the example simulation.	80
5.24	The power output of Turbine 8 in the example farm.	81
5.25	The generator speed of turbine 8 in the example farm.	82
6.1	A diagram of the rainflow counting and S-N curve lookup process. [32]	84
6.2	An example S-N Curve for a wind turbine blade.	85
6.3	The interaction between different turbine components within the wind turbine model used in Strathfarm. From [11].	87
6.4	The power curve of the Supergen exemplar 5MW wind turbine over a range of wind speeds	88
6.5	Pitch angle by wind speed from BLADED demo 5MW model.	89
6.6	The fore-aft tower bending moment of the Supergen exemplar 5MW wind turbine over a range of wind speeds	89
6.7	The out of plane blade bending moment of the Supergen exemplar 5MW wind turbine over a range of wind speeds	90
6.8	The fore-aft bending moment of the Supergen 5MW wind turbine tower by wind speed.	91
6.9	The out-of-plane bending moment of the Supergen 5MW wind turbine blade by wind speed.	92
6.10	A 3D plot of the steady state fore-aft tower bending moment by wind speed and curtailment	93

List of Figures

6.11	A 3D plot of the steady state out of plane bending moment by wind speed and curtailment	94
6.12	A 3D plot of the steady state side-side tower bending moment by wind speed and curtailment	95
6.13	A 3D plot of the steady state hub torque moment by wind speed and curtailment	95
6.14	The Power spectral density of the in plane blade bending moment at 12 m/s . .	96
6.15	Power curve adjustments of the power estimation control strategy	99
6.16	A contour plot of the steady state fore-aft bending moment of the tower with the power estimation control strategy at different curtailment levels.	100
6.17	A contour plot of the steady state out of plane bending moment of the blade with the power estimation control strategy at different curtailment levels. . . .	101
6.18	A contour plot of the steady state hub torques with the power estimation control strategy at different curtailment levels.	102
6.19	The “New Baseline” Power curve for the Intelligent Tower Control strategy . .	103
6.20	Power curve adjustments of the intelligent tower control strategy	104
6.21	A contour plot of the steady state fore-aft bending moment of the tower with the intelligent tower control strategy at different curtailment levels.	105
6.22	A contour plot of the steady state out of plane bending moment of the blade with the intelligent tower control strategy at different curtailment levels.	106
6.23	A contour plot of the steady state hub torques with the intelligent tower control strategy at different curtailment levels.	107
6.24	The idealized fore-aft tower bending moment with the intelligent tower control strategy	109
6.25	The idealized out of plane blade bending moment with the intelligent tower control strategy	110
6.26	The idealized side-side tower bending moment with the intelligent tower control strategy.	111
6.27	The wind farm layout used in the validation process.	114
6.28	Wind farm power curve for the upper set-points	115
6.29	Mean fore-aft tower DELs by wind speed and curtailment strategy	116

List of Figures

6.30	Mean side-side tower DELs by wind speed and curtailment strategy	117
6.31	Mean combined tower DELs by wind speed and curtailment strategy for the upper set-points	118
6.32	Fore-aft tower PSD at 12m/s with farm curtailment set-point of 64MW.	119
6.33	Mean out of plane blade DELs by wind speed and curtailment strategy for the upper set-points	120
6.34	Out of plane blade PSD at 14m/s with farm curtailment set-point of 64MW.	121
6.35	Mean hub torque DELs by wind speed and curtailment strategy for the upper set-points	122
6.36	Hub torque PSD at 14m/s with farm curtailment set-point of 72MW.	123
6.37	In plane blade PSD at 14m/s with farm curtailment set-point of 64MW.	124
6.38	Wind farm power curve for the middle set-points	125
6.39	Mean combined tower DELs by wind speed and curtailment strategy for the middle set-points	126
6.40	Mean out of plane blade DELs by wind speed and curtailment strategy for the middle set-points	127
6.41	Mean hub torque DELs by wind speed and curtailment strategy for the middle set-points	128
6.42	Wind farm power curve for the lower set-points	129
6.43	Mean combined tower DELs by wind speed and curtailment strategy for the lower set-points	129
6.44	Mean out of plane blade DELs by wind speed and curtailment strategy for the lower set-points	130
6.45	Mean hub torque DELs by wind speed and curtailment strategy for the lower set-points	131
6.46	Bins used for the Weibull distribution used with a mean speed of 8m/s.	132
6.47	Bins used for the Weibull distribution used with a mean speed of 9m/s.	133
6.48	Bins used for the Weibull distribution used with a mean speed of 10m/s.	133
7.1	Offshore power system model diagram from [42].	142
7.2	Solver block from the Power system model.	152

List of Figures

7.3	An example of the solver error by iteration for a single turbine system.	154
7.4	The full power system model as it appears in Strathfarm	157
7.5	A schematic diagram of the power system in Strathfarm	157
7.6	The Mechanical model block from the Power system model.	159
7.7	The contents of the governor block in the diesel generator system from [80]. . .	161
7.8	The electrical model block from the power system model.	162
7.9	An example circuit from [79] showing the generator circuits	163
7.10	The continuous time model of the generator excitation system from the original 15MW generator model.	167
7.11	The power system block from the power system model	168
7.12	Diagram of Dynamic node from [18].	169
7.13	The wind farm controller blocks in Strathfarm.	171
7.14	The dispatch algorithm used in the inertial response shown in this chapter from [90]	172
7.15	Wind Farm Layout for the inertial and droop response results	173
7.16	The droop curve for the wind farm showing the requested set-point power by the grid frequency.	174
7.17	The additional current added to the power system model to create a frequency change to activate the droop controller.	175
7.18	The farm wide power output of the wind farm in the droop control stimulations by distribution strategy.	176
7.19	The farm wide power output of the wind farm in the droop control stimulations by distribution strategy. at the time of the droop event.	177
7.20	The requested change to the wind farm power in the droop simulations for three distribution strategies and when the droop controller is not active	178
7.21	The requested change to the wind farm power in the droop simulations for three distribution strategies and when the droop controller is not active at the time of the droop event.	178
7.22	The grid frequency in the droop simulations for three distribution strategies and when the droop controller is not active	179

List of Figures

7.23	The grid frequency in the droop simulations for three distribution strategies and when the droop controller is not active at the time of the droop event.	179
7.24	The power output and set-point from the droop controller in the equal distribution strategy droop response.	180
7.25	The power output and set-point from the droop controller in the power estimation distribution strategy droop response.	181
7.26	The power output and set-point from the droop controller in the intelligent tower distribution strategy droop response.	181
7.27	The additional current added to the power system to simulation an inertial event.	183
7.28	The power output of the wind farm in the inertial event for different wind farm algorithms	184
7.29	The power output of the wind farm in the inertial event for different wind farm algorithms at the time of the inertial event.	185
7.30	The power system frequency of the wind farm in the inertial event for different wind farm algorithms.	185
7.31	The power system frequency of the wind farm in the inertial event for different wind farm algorithms at the time of the inertial event	186
7.32	The rate of change of frequency of the power system frequency of the wind farm in the inertial event for different wind farm algorithms.	187
7.33	The rate of change of frequency of the power system frequency of the wind farm in the inertial event for different wind farm algorithms at the time of the inertial event.	187
8.1	An image from [57] showing the wake expansion as the wake travels downwind.	190
8.2	Diagram showing the grid used for node based turbulence. From [90].	194
8.3	Diagram showing the wake interaction with a turbine's swept area.	196
8.4	Diagram showing the wake interaction with a turbine's swept area.	197
8.5	The turbine level implementation of the lookup tables in Strathfarm.	200
8.6	The turbine level implementation of the wake model in Strathfarm.	201
8.7	Gaussian wake with a CT value of 0.8 at the upwind turbine, a turbulence intensity of 0.1 and a downwind distance of 4 turbine diameters.	202

List of Figures

8.8	Gaussian wake with a CT value of 0.4 at the upwind turbine, a turbulence intensity of 0.1 and a downwind distance of 4 turbine diameters.	203
8.9	Gaussian wake with a CT value of 0.8 at the upwind turbine, a turbulence intensity of 0.065 and a downwind distance of 4 turbine diameters.	203
8.10	Gaussian wake with a CT value of 0.8 at the upwind turbine, a turbulence intensity of 0.1 and a downwind distance of 8 turbine diameters.	204
8.11	Gaussian wake with a CT value of 0.8 at the upwind turbine, a turbulence intensity of 0.1 and a downwind distance of 4 turbine diameters with a lateral offset of 0.4 turbine diameters.	204
8.12	Gaussian wake with a CT value of 0.8 at the upwind turbine, a turbulence intensity of 0.1 and a downwind distance of 4 turbine diameters with a lateral offset of 1 turbine diameters.	205
8.13	Gaussian wake with a CT value of 0.4 at the upwind turbine, a turbulence intensity of 0.065 and a downwind distance of 4 turbine diameters with a lateral offset of 0.4 turbine diameters.	205
8.14	Gaussian wake with a CT value of 0.2 at the upwind turbine, a turbulence intensity of 0.065 and a downwind distance of 4 turbine diameters with a lateral offset of 0.4 turbine diameters.	206
8.15	The wake deficit for the three wake models by lateral wake centre offset.	207
8.16	The fore-aft tower bending moment of the downwind turbine by lateral wake centre offset.	207
8.17	The power output of the downwind turbine by lateral wake centre offset.	208
8.18	The wake deficit for the three wake models by lateral wake centre offset.	208
8.19	The fore-aft tower bending moment of the downwind turbine by lateral wake centre offset.	209
8.20	The power output of the downwind turbine by lateral wake centre offset.	209
A.1	The general form of the Power Adjusting Controller (PAC) [98]	230
A.2	A torque-speed diagram of a wind turbine extracting energy from its rotor for 30 seconds at 9, 10 and 11 m/s.	232

List of Figures

A.3 The tip speed ratio of a wind turbine which is extracting energy from its rotor for 30 seconds at 9, 10 and 11 m/s. 233

A.4 The ratio of the turbine level C_P divided by the steady state C_p as energy is extracted from the turbine's rotor. 234

A.5 The layout of the wind farm used for the simulations presented here. 237

A.6 An example of the Reserve Curtailment strategy in a 9.5 m/s windfield with 10% turbulence intensity in a 16 turbine wind farm. 237

C.1 Gaussian wake 1200 meters down wind with an upwind turbine C_T of 0.4 . . . 254

C.2 Gaussian wake 1200 meters down wind with an upwind turbine C_T of 0.8 . . . 254

C.3 Gaussian wake 2400 meters down wind with an upwind turbine C_T of 0.4 . . . 255

C.4 Gaussian wake 2400 meters down wind with an upwind turbine C_T of 0.8 . . . 255

List of Tables

5.1	Mean squared error of each of the levels of anti-windup shown in this chapter.	75
6.1	Change in combined tower DELs by control strategy and wind speed compared to not curtailing.	134
6.2	Change in out of plane blade DELs by control strategy and wind speed compared to not curtailing.	134
6.3	Change in in plane blade DELs by control strategy and wind speed compared to not curtailing.	135
6.4	Change in hub torque DELs by control strategy and wind speed compared to not curtailing.	135
6.5	Change in energy capture by control strategy and wind speed compared to not curtailing.	136
6.6	Change in energy capture by control strategy and wind speed compared to not curtailing with the 22m/s and 24m/s bins ignored.	136
6.7	Change in combined tower DELs by control strategy and wind speed at the turbine level.	137
6.8	Change in out of plane blade DELs by control strategy and wind speed at the turbine level.	137
6.9	Change in in plane blade DELs by control strategy and wind speed at the turbine level.	138
6.10	Change in hub torque DELs by control strategy and wind speed	138
7.1	The inputs of the solver S-Function block in Strathfarm	152
7.2	The outputs of the solver S-Function block in Strathfarm	153

List of Tables

7.3	The inputs to the mechanical S-Function	159
7.4	The outputs of the mechanical S-Function	159
7.5	The inputs to the electrical S-Function in the power system model.	162
7.6	the outputs of the electrical S-Function in the power system model.	162
7.7	The inputs to the linking S-Function in Strathfarm.	168
7.8	The outputs from the linking S-function in Strathfarm.	168
8.1	The values used in the figures shown in this section	202
A.1	Maximum sustainable power extraction from the rotor for 30 seconds and al- located turbine level curtailment by wind speed.	234
A.2	The average reduction in tower DELs of each of the three strategies by wind speed when compared to not curtailing the turbines at 5% TI.	238
A.3	The average reduction in blade DELs of each of the three strategies by wind speed when compared to not curtailing the turbines at 5% TI.	239
A.4	The average reduction in energy capture of each of the three strategies by wind speed when compared to not curtailing the turbines.	239
A.5	The average reduction in tower DELs of each of the three strategies by wind speed when compared to not curtailing the turbines at 10% TI.	240
A.6	The average reduction in blade DELs of each of the three strategies by wind speed when compared to not curtailing the turbines at 10% TI.	240
A.7	The average reduction in energy capture of each of the three strategies by wind speed when compared to not curtailing the turbines at 10% TI.	240
D.1	The external tuning parameters of the solver S-Function	256
D.2	The external tuning of the Mechanical model S-function.	257
D.3	The external tuning parameters for the electrical S-Function for the power sys- tem model.	258
D.4	The external tuning variables for the power system linking section of the power system model in Strathfarm	259

Preface/Acknowledgements

I would like to thank my supervisor Professor Bill Leithead for his consistent friendship and warm-hearted support throughout my PhD. I would also like to thank Adam Stock, David Campos-Gaona, Lindsey Amos and Alexander Giles for sharing their wealth of knowledge without which this work would not have been possible. I would also like to thank in no particular order: Drew Smith, Edward Hart, Keith Bell, Marcel Nedd, Hong Yue, Alasdair McDonald, Julian Feuchtwang, Jethro Browell, Luis Recalde Camacho, Abbas Mehrad Kazemi Amiri, Michael Smailes, Sheila Campbell, Max Parker, Cagatay Cebeci, Edgar Lucas MacLeod, Alan Turnbull, Nicola Greive, Grant Leishman, Becky Corley-Morgan, George Elderfield, Gabriele Pisetta, Anastasis Charalambous, Ahmed el-Bozie, Eva Topham, Elena Gonzalaz, Alfred Alsop, Alfred Cotton, Ciaran Gilbert, Jonathan Bowes, Sofia Koukoura, Oliver Tulloch, Adam Harris, Conor McKinnon, James Sterling, Leo May, Marilou Jourdain De Thieulloy, Mathieu Kervyn, Paul Pirrie, Peter Taylor, Rosemary Tawn and Rebecca Hall as well as everyone else in the Wind and Marine Energy Systems CDT and the Smart Grids CDT. In addition to this I would like to thank my friends: William Hebden, Evie Tucker, Emily Smith, Benjamin Jones and Jock Blakley as well as my family, in particular my parents, for their unwavering belief and support throughout my PhD.

Chapter 0. Preface/Acknowledgements

Chapter 1

Introduction

As wind power increases its share of grid penetration, especially through very large offshore arrays with capacities into hundreds of megawatts, there are increased requirements on wind farms. The requirements which this project will be focused on implementing are providing ancillary services from wind arrays which are connected by a single AC connection to a power grid. Curtailment is essential for wind farms to be integrated into the grid, as if wind turbines were to always generate at the wind speed dictated output, the grid would not be able to match supply and demand. While this presents a challenge to wind farm operation there are potential benefits in curtailing wind farms such as reduced loadings on blades and turbine towers which will in turn reduce operation and maintenance (O&M) costs leading to a reduction in the cost of energy. This is particularly essential for offshore arrays as they are often designed with a single grid connection meaning that the grid sees the wind farm as a single source of power. Therefore, research is required in this area as wind turbines operating at their individual optimum will not necessarily constitute the wind farm as a whole operating at its optimum. There is much research on each of the individual stages of the path from the wind through the blades and drive train, the generator, the farm power system and grid connection. However, there is little research which attempts to combine all of these elements into a single simulation. This research is essential as grid penetration from wind power is increasing and so will be increasingly essential in the provision of ancillary services as many conventional sources of electricity which are reliant on fossil fuels are being replaced by renewable sources of energy.

1.1 Motivation

In response to the climate crisis there is an urgent need to rapidly decarbonise electricity generation. Wind energy offers a sustainable, affordable alternative to fossil fuel sources of electricity. In order to improve wind farm operation and operate wind farms in a flexible way software must be developed first as a cost effective method of testing any hypothesized improvements in wind farm operation. The work shown in this thesis presents new wind farm control algorithms using the existing software developed from previous work. It then discusses further developments that improve the accuracy and scope of the existing software by developing a power system model and by adding a more accurate wake model.

1.2 Publications and Research Outputs

- FarmConnors wind farm flow control benchmark - Part 1 : blind test results
Göçmen Tuhfe, Campagnolo Filippo, Duc Thomas, Eguinoa Irene, Andersen Søren Juhl, Petrović Vlaho, Imširović Lejla, Braunbehrens Robert, Feng Ju, Liew Jaime, Baungaard Mads, van der Laan Maarten Paul, Qian Guowei, Aparicio-Sanchez Maria, González-Lope Rubén, Dighe Vinit, Becker Marcus, van den Broek Maarten, van Wingerden Jan-Willem, Stock Adam, Cole Matthew, Ruisi Renzo, Bossanyi Ervin, Requate Niklas, Strnad Simon, Schmidt Jonas, Vollmer Lukas, Blondel Frédéric, Sood Ishaan, Meyers Johan *Wind Energy Science* Vol 7, pp. 1791-1825 (2022) <https://doi.org/10.5194/wes-2022-5> [45]
- Using rotor inertia as stored energy in below rated wind farms to provide primary frequency response Cole Matthew, Stock Adam, Leithead WE, Amos Lindsey *Journal of Physics: Conference Series* Vol 1618 (2020) <https://doi.org/10.1088/1742-6596/1618/2/022026> [27]
- Distributed control of wind farm power set points to minimise fatigue loads Stock Adam, Cole Matthew, Leithead Bill, Amos Lindsey 2020 American Control Conference (ACC) American Control Conference (ACC) 2020 American Control Conference, pp. 4843-4848 (2020) <https://doi.org/10.23919/ACC45564.2020.9147732> [99]

Chapter 1. Introduction

- Wind farm control work at the University of Strathclyde, past, present and future Stock Adam, Leithead William, Anaya-Lara Olimpo, Amos Lindsey, Cole Matthew, Taylor Peter, Pirrie Paul, Campos-Gaona David 5th Wind Energy Systems Engineering Workshop (2019) [101]
- Supergen Wind Hub : D3.3 Report on Wind Farm Control Algorithms to Meet Asset Management Requirements, Kazacoks Romans, Cole Matthew, Leithead Bill (2019) <https://doi.org/10.17868/77130> [108]
- A critical review of current and future options for wind farm participation in ancillary service provision, : Cole, Matthew. Campos-Gaona, David. Stock, Adam. Nedd, Marcel. *Energies A3: Wind, Wave and Tidal Energy* (2023)[25]
- Development of a power system model For Strathfarm: Cole Matthew, Stock Adam, Campos-Gaona David, Smailes Michael. *EERA DeepWind 2023 Conference Poster* (2023) [26]

1.2.1 Individual contribution

The work presented in this thesis is built on many other people's previous research. Hence for clarity it is important to state my individual contribution to Strathfarm's development. I was the lead author on the publication which chapter 4 is based on, with assistance from David Campos-Gaona, Adam Stock and Marcel Nedd who are all credited on the paper as co-authors. Much of Strathfarm's design was developed through combining existing work, with most of the underlying code having been developed by Lindsey Amos who developed it from the preceding MAXFARM project as well as other research projects in the Wind and Marine energy systems CDT and the Wind energy Control Centre. My individual contributions to Strathfarm's development as presented in this thesis are built on this previous work. As shown in chapter 5, I developed the discrete WFC controller and anti-windup developed from the previous continuous time approach [54]. As shown in chapter 6, I developed the wind farm control distribution algorithms and designed the steady state assessment used to develop the intelligent tower control. Additionally, I wrote the codes for the batching process used to automate the running of such a large number of simulations in the validation. As shown in chapter 7, I implemented existing techniques from [42] into Strathfarm by developing a novel solver code for the scalable matrix system and developed the power system model based on existing models Simulink models and implemented the new power system model into Strathfarm. As shown in chapter 8, I collaborated with Adam Stock to develop a Gaussian wake model compatible with the existing models in Strathfarm and implementing the developed approach within Strathfarm's code.

1.3 Summary of Thesis

Chapter 2 of the thesis details the historical context for the development of wind energy and power grids.

Chapter 3 of this thesis covers some of the background of wind turbine and wind farm control including different simulation software programs used for academic research of wind farm behaviour.

Chapter 1. Introduction

Chapter 4 contains a review of the academic literature for wind farm operation in provision of ancillary services. This chapter has been published as a review in *Energies* [25].

Chapter 5 of this thesis reviews the previous work and model development that the research presented was build upon. The chapter also builds through some of the basic initial wind farm control research done to ensure farm wide set-point power tracking.

Chapter 6 of this thesis reviews how wind turbine component fatigue is measured using rainflow based Damage equivalent load approaches. It then considers the steady state bending moment properties of the Supergen 5MW exemplar wind turbine when curtailed using the Power Adjusting Controller (PAC). This analysis is then used to develop a wind farm control algorithm for dispatching curtailments across a scalable wind arm to reduce component fatigue. Finally the chapter presents a verification exercise to quantify any changes to turbine fatigue through a range of curtailment scenarios, comparing the novel strategy with two other approaches.

Chapter 7 discusses the development of the power system model in Strathfarm being with the development of a nonlinear solver algorithm for the wind farm power system model developed by [42]. The chapter then shows the development of a dynamic power system model for Strathfarm to allow for the study of the efficacy of frequency provision from the wind farm using the dispatch approach from [90]. An example of the inertial dispatch and the droop dispatch are shown.

Chapter 8 reviews the history of engineering models for wind turbine wakes, then considers the implementation of a Gaussian wake model in Strathfarm.

Chapter 9 contains the conclusions of the Thesis.

Appendix A contains the conference paper [27] which was written for Torque 2020. It discusses a control strategy for using energy stored in the rotation of wind turbine generators for providing a primary frequency response at below rated wind speeds.

Appendix B contains additional information for the creation of the power system model in Strathfarm. This consists of additional difference equations and working from the discretization process of the continuous time model.

Appendix C presents and unused alternative approach for the inclusion of the Gaussian wake model in Strathfarm.

Appendix D lists the external tuning variables used in the power system model.

Chapter 1. Introduction

Appendix E shows the C++ code used in the wind farm distribution controller.

Appendix F shows the C++ code used in the wind farm network controller.

Chapter 2

Context

This short chapter details the history of windmills including the development of wind turbines for generating electricity. It then discusses the formation of the National Grid and the invention of the governor control system.

2.1 A History of Windmills

The earliest windmills were found in antiquity with some historians claiming their invention as early as the period of Emperor Hammurabi's rule of Babylonia[2].



Figure 2.1: left:A European four sail windmill design[105]. Right: Mill in Sint Hubert, Netherlands [30].

In Europe, a four bladed horizontal axis design saw widespread use as a source of energy of industrial uses such as grinding cereals, sawing wood and pumping water. Figure (2.1) shows

Chapter 2. Context

two examples of the traditional European wind mill design. This design has four blades made from timber and canvas, the tower is often of stone construction but sometimes timber. [67]



Figure 2.2: An example American style windpump [78]

Another common design of wind pump is seen in Figure (2.2) which shows a waterpump in the USA. This design has a higher number of blades, as while this is less efficient in energy capture the high number of blades allows for a steadier output torque which is sufficiently high for pumping at low wind speeds.

2.1.1 Electricity Generating Wind Turbines

A key pioneer in the development of wind turbines was Scottish inventor James Blyth. In 1885 he began building wind turbines at his home in Marykirk Aberdeenshire. The description of the turbine in Aberdeenshire Council's historic record is: "... being of tripod design, with a 33

Chapter 2. Context

foot windshaft, four arms of 13 feet with canvas sails, and a Burgin dynamo driven from the flywheel using a rope.”[28].

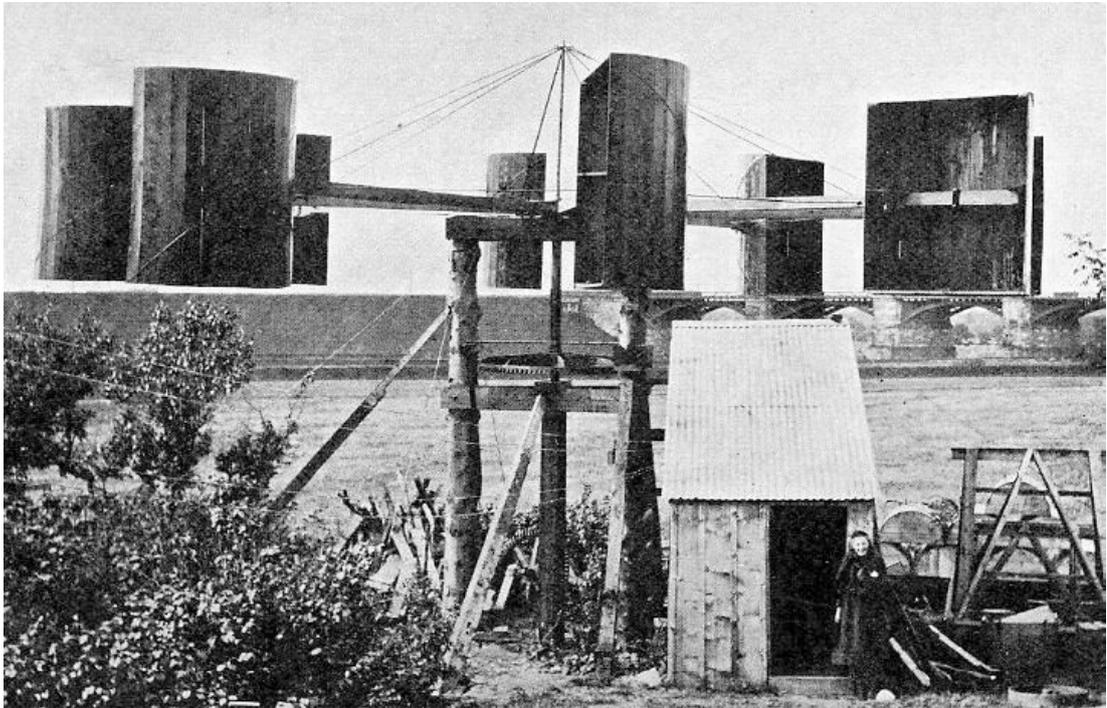


Figure 2.3: James Blyth's wind turbine [31]

Figure (2.3) shows James Blyth's 1891 vertical axis wind turbine design. When Blyth offered to provide electricity to light the main street of the village he was rejected as the electricity was thought by the locals to be "the work of the Devil" [70]. The turbine powered lights in his home. He subsequently constructed a larger version of the turbine for providing electricity to the Royal Asylum mental hospital at Montrose [70].

The origin of modern research in renewable energy sources begins in the 1970s as a result of repeated oil crises [44]. While many of the initial designs were two bladed (occasionally four or more) these early designs were later refined to three bladed based on research from Denmark [44] which became the modern standard "Danish model" wind turbine with three blades on a horizontal axis turbine. Through the 1970s and 1980s the United Kingdom's government invested heavily in wind power research. [91] testing many different turbine prototypes in the Carmarthen bay test site in Wales. However, funding cuts in the 1990s stalled the UK's progress with wind power resulting in the UK having to import many wind turbines in the 2010s due to

Chapter 2. Context

the lack of UK based manufacturing.



Figure 2.4: The Richborough 1MW Wind Turbine. Credit Jock Blakley

Over time the design of wind turbines has been refined to the typical three blade design that is commonly now seen.

Figure (2.4) shows the Richborough 1MW Wind Turbine. This turbine was erected in 1989 and was at the time, one of the biggest wind turbine installed in the United Kingdom. Around the late 1980s several large test turbines were developed in the UK some of which were ambitiously large for the time. For example the LS1 turbine was developed in 1985 and produced a rated power of 3MW [81].

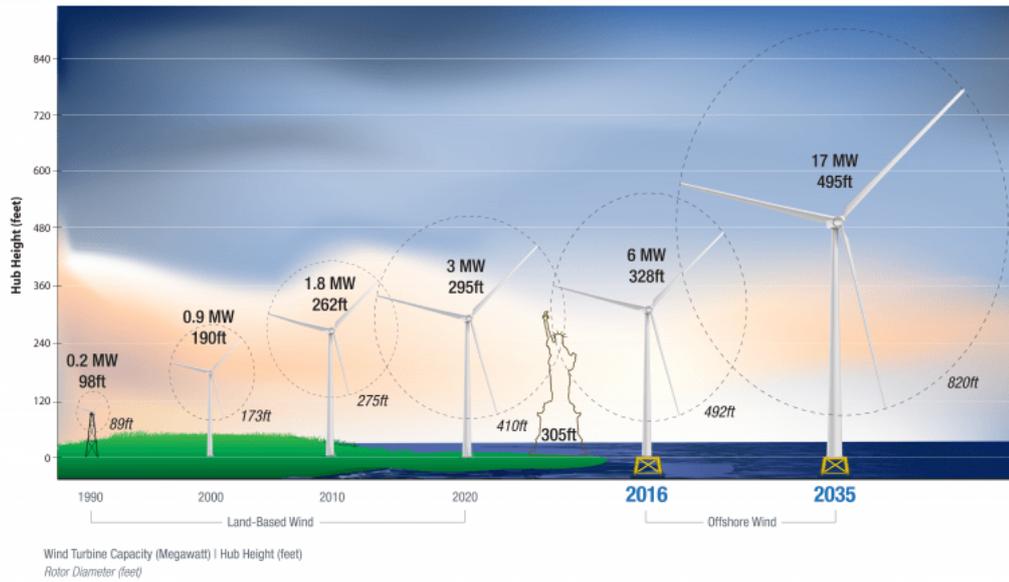


Figure 2.5: The increasing size of wind turbines over time from [88]

Figure (2.5) shows the increasing size of wind turbines over time. It can be seen that the Richborough 1MW Wind Turbine would have been considered small compared with commercial wind turbines developed 10 years ago. It can also be seen that within the next 10-15 years turbines could be as large as 17MW with rotors 250 meters in diameter.

2.1.2 Wind Farms as Power Plants



Figure 2.6: Wind turbines at the Whitelee Windfarm near Glasgow. Own work.

Figure (2.6) shows an example of a modern onshore wind farm. As wind farms are increasingly being situated offshore in large arrays with a single connection to the grid they can be operated as a single entity. As these wind farms are connected through a single cable they act as a single generator when seen from the grid. Hence there is scope for considering the whole of a wind farm as a single electricity plant, considering a collective control strategy across the farm rather than at the wind turbine level. Historic research in wind energy has focused on the maximization of energy capture at the wind turbine level but as wind penetration increases in grids around the world wind farms will be expected to operate in a way similar to how synchronous power plants currently operate by:

1. Outputting a requested set-point power as instructed by grid operators
2. Increase or decrease power output to maintain supply and demand in a power grid

Chapter 2. Context

3. maintain a reserve of power that can be used if required to match supply and demand

While it may seem counter intuitive given the climate crisis to reduce energy capture from renewable sources such as wind there are benefits to increased controllability and flexibility of operation of wind farms such as reduced component fatigue and scope for ancillary service provision that make wind energy more cost effective and reduces the requirement for reserve fossil fuel generation on the grid.

Chapter 3

Background

This chapter details the existing operation of wind turbines and considers how their operation is computationally modelled at the turbine and farm level.

3.1 Operational Strategy For A Wind Turbine

Most wind turbines used today are operated using a variable speed, variable pitch strategy. This strategy controls turbine operation through two mechanisms: torque control and pitch control. The standard operational strategy has four modes of operation: A low wind speed constant speed mode, a maximum power tracking variable speed mode, a higher speed constant speed mode and a rated power model.

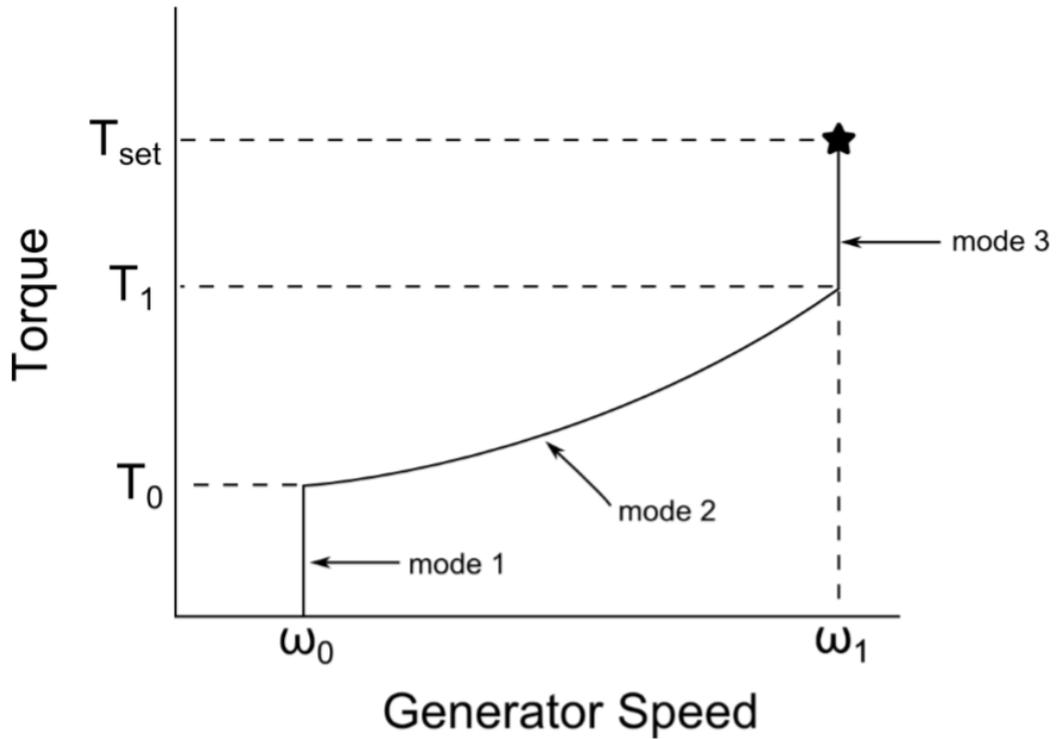


Figure 3.1: The modes of operation for the Supergen 5MW wind turbine model.[3]

Figure (3.1) shows the standard operational strategy used in most modern wind turbines. Mode one consists of a small constant speed section as the wind turbine will start with an initial speed of zero. Mode two consists of a curve which tracks the maximum power output which the wind turbine can provide at a given wind speed. When the turbine reaches its maximum rotor speed it enters mode three where it maintains a higher constant speed. When the turbine reaches rated power it enters mode four where it maintains its torque and speed by the pitching of the blades.

The operational strategy seen in Figure (3.1) results in a typical power curve for a wind turbine as seen in Figure (3.2) .

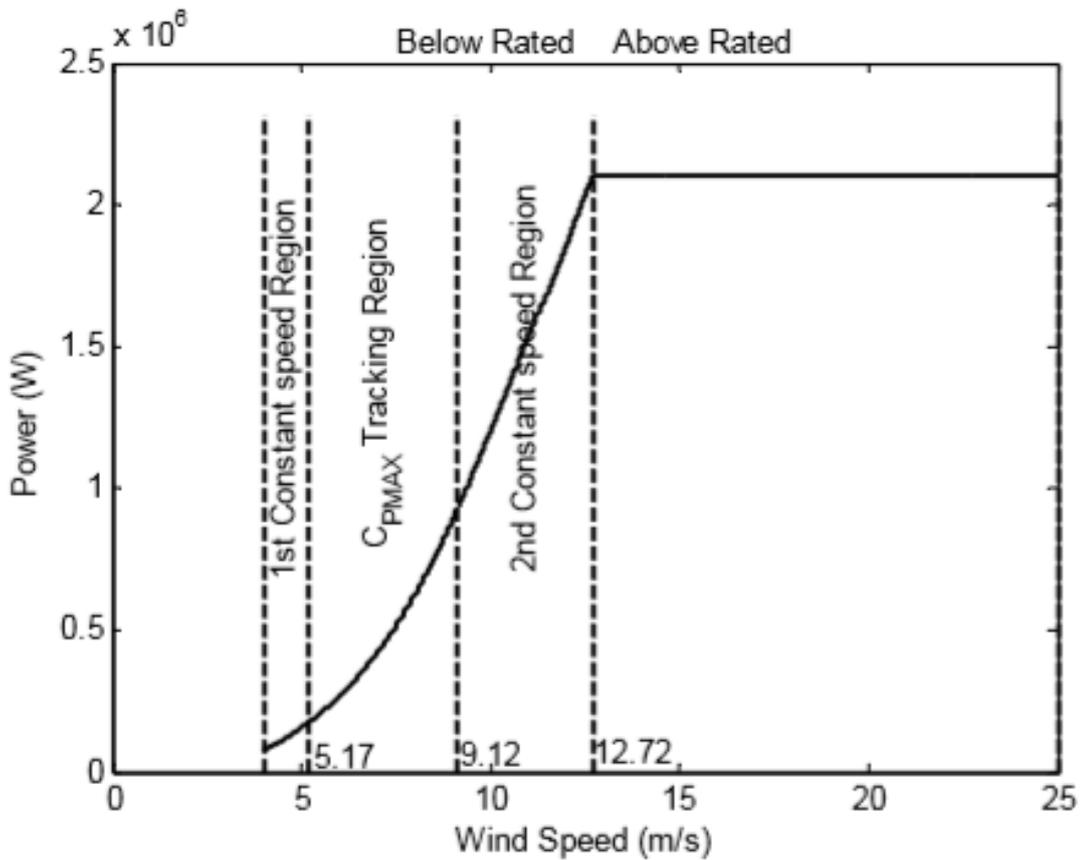


Figure 3.2: An example of a typical wind turbine power curve.[21]

Figure (3.2) shows the four modes of operation of the turbine over a power curve.

3.2 Wind farm Modelling Software

There is an inevitable trade-off that inherently exists when modelling a large wind field over a wind farm between fidelity and computational speed, so a range of differing modelling strategies have been used depending on whether computational workload or windfield detail is required. The least computationally expensive but lowest fidelity models tend to have steady state or quasi-steady state wind fields whereas, the highest fidelity models tend to use a complete computational fluid dynamics (CFD) approach which is highly accurate but can take a

Chapter 3. Background

long time to simulate. Sometimes a medium fidelity approach, which sometimes has been validated against the higher fidelity methods, is used as a compromise.

Simulator fOr Wind Farm Applications (SOWFA) is the National Renewable Energy Laboratory (NREL)'s high fidelity wind farm modelling software. SOWFA uses a wind field model based on a comprehensive three dimensional Navier-Stokes equations approach and so has a very high level of accuracy but this comes at the cost of being very computationally expensive. Some larger models can require weeks of computational time to simulate a wind farm over ten minutes.[58]. Due to its accurate but slow nature, SOWFA is often used for calibration for lower fidelity models. Of particular interest for this research is the inclusion of a wind farm controller, called a super controller, for wind farm level control.

FLow Redirection and Induction in Steady state (FLORIS) is a model which has been designed by NREL. However it is a low fidelity approach as the name suggests it is in steady state. It utilites a version of the Jensen model, a wake model developed in the 1980s which considers a wake as a speed reduction downwind, still being used due to it being computationally inexpensive [89]. This approach has been used in an extended manner to improve the accuracy when considering multiple and deflected wakes and uses uniform wakes and steady wind flow across the farm based on average rotor thrust. Due to the use of a steady state approach damage equivalent loads cannot be found through models of this type.

NREL's current mid-fidelity model is called The Dynamic Wake Meandering Model (DWM) which uses a Naiver-Stokes approximation and considers wake expansion resulting from turbulence diffusion and rotor field diffusion[77] , a wake meandering model which stochastically simulates the wakes transportation by large turbulent structures and also considers the additional turbulence caused by the existence of the wakes. The model can be used to simulate the loads across a whole wind farm on a standard PC but does so sequentially from the most upwind turbine considering only downstream wake effects so is not always sufficiently accurate.

NREL's newest modelling software which is currently being designed is FAST.farm. The aim in developing FAST.farm appears to be a mid-fidelity model without DWM's limitations and inaccuracies, of particular interest for this research is the introduction of the super-controller which has previously only been used in SOWFA which is used for wind farm control. As this model is being developed it is being calibrated using higher fidelity data from SOWFA

[58]. One of the key differences between FAST.farm and DWM is the inclusion of wind farm controller (WFC), in the paper referred to as a super controller, which will be identical to the super controller in SOWFA.

WindFarmSimulator (WFSim) is a mid-fidelity model which is based on two dimensional Navier-Stokes equations modified with an estimated vertical component to improve accuracy [17]. The aim of the development of this software appears to be for use modelling wind farm control, which would not necessarily require the high level of accuracy that comes from 3D Navier-Stokes. The result have been validated against results from SOWFA which uses 3D Naiver-Stokes.

3.3 Wind Farm Control

3.3.1 Why use control at the wind farm level?

The key reason to consider control at the wind farm level is that when a wind farm is connected to the grid, it, in most cases, is connected through a single connection to the rest of the grid. Hence, from the grids perspective the wind farm is seen as a single generator. Additionally, when considering power set-point control the farm level power will always have less proportional variation in power output than a single wind turbine as the stochastic variations in wind speed seen at each of the turbines can cancel out and it is unlikely that all turbines experience a rapid change in wind speed simultaneously. Another key advantage to using wind farm level control is that the impacts turbines are having on each other can be considered and mitigated as the optimum strategy for a single turbine might not align with the collective optimum for the whole farm. An example of where this is very beneficial is in wake steering when a turbine yaws to deflect its wake to reduce the impact of the wake on a downwind turbine.

3.3.2 Aims of wind farm control

Research into wind farm control is usually motivated by one or more of the following:

1. minimising fatigue
2. maximising total power output

Chapter 3. Background

3. providing ancillary services

Often the provision of ancillary services is considered by having the wind farm follow a set-point for total power output which can be set as required by a Transmission System Operator (TSO).

3.3.3 Maximising Power Output

The research in this area has normally consisted of considering a wind farm in a quasi-steady state wind field. As the wakes of the upwind wind turbines cause a reduction in the effective wind speed experienced by turbines directly downwind there is a reduction in total power output of the wind farm. However, by curtailing the power output of the upwind turbines the wakes effect on the down wind turbines is reduced.

Some of the research in wind farm control looks at maximising the total power output of the wind farm through the reduction of wakes from the upwind turbines. For example, [90] looks at a heat and flux approach to optimizing the power output of a wind farm by curtailing upwind turbines. The model used ten turbines in a row aligned with the wind direction separated by 800 meters in a turbulence free wind field. A trial and error based approach was applied resulting in the optimised curtailments across the wind farm at 8, 9, 10 and 11ms. The total power output in all optimisations increased by less than a percent in each wind field. when 10 percent turbulence intensity is used the improvements reduce and at 11ms the optimization resulted in a lower energy capture.

3.3.4 Minimizing Wind Turbine Component Fatigue

Research from the previous section is often unsuitable for considering fatigue as a simulation using a quasi-static wind field will not have an accurate variation in wind speed and therefore cannot output an accurate damage equivalent load.

In [63], a strong explanation of how fatigue of a wind turbine tower should be considered is given, reasoning that as the tower acts a large beam, force is transmitted from the wind to the blades and subsequent to the tower via the hub resulting in the tower moving and this movement is proportional to stress in the tower. The tower stress can be separated into cycles

Chapter 3. Background

using a rainflow counting approach which can then be used to find a damage equivalent load. The review observes that any level of constant tower thrust will result in zero fatigue. It is also noted that using the standard deviation of tower thrust is a poor estimate of tower stress as the same value could represent frequencies near the towers resonant frequency which would result in increased tower movement and also that as the frequency contents are not known then the cycle counts are also unknown so no calculation of fatigue can be made. As a result of this, the review questions the validity of approaches to reduce fatigue using an objective function with thrust used as an input.

In [52], a description of the methodology of finding damage equivalent loads is detailed. By counting cycles and half cycles in the moment profile using rainflow counting and comparing them to a single frequency (chosen as 1HZ) using an S-N curve based approach a single damage equivalent load can be found for a component.

Turbine Level Control Strategies for Load Reduction

Many strategies of turbine control exist of reducing wind turbine component fatigue. One example is individual pitch control [68] which presents a system of alleviating rotor loads by independently adjusting the pitch of each individual blade independently using a sensor-control system within each individual blade's actuator. However, rotor imbalance is not considered in Strathfarm. One assumption that is made in the turbine models used in Strathfarm is that each of the blades is in alignment, this is done as it reduces the complexity of the model as only one blade needs to be modelled. Peak shaving (also known as thrust clipping), is an approach for reducing fatigue of a wind turbine in normal operation. This approach reduces energy capture near the rated wind speed in order to reduce the maximum steady state thrust on the turbine. It is often considered as a last resort option due to the reduction in energy capture [36]. By considering peak shaving in conjunction with the need to curtail, a wind farm control strategy can be developed which prioritizes reductions in power near to the rated wind speed.

3.4 Aims and objectives

The aim of this thesis is to show how wind farms can be flexibly operated in a future power system. Hence, the objectives for this work are first to review the existing rules for wind farm operation in a power system, then to consider the academic literature for future wind farm operation strategies. Following this, the second aim is to develop Strathfarm such that novel strategies for flexible operation of a wind farm can be considered, including any impact on component fatigue. After this Strathfarm will be developed further through the inclusion of a power system model to study the efficacy of grid support from ancillary service based wind farm control strategies.

Chapter 4

A Literature Review of Wind Farm Control for Grid Support

This chapter reviews the academic literature of how wind farm control can be used for ancillary service provision. This chapter contains a review paper published in Energies journal [25].

4.1 Introduction

The use of advanced control for wind farms/turbines for providing ancillary services to the grid has been the subject of several review papers in the last decade [14, 96, 106, 61]. However, because of the rapid changes in climate policies, the subject has seen significant developments in the published literature. Hence, this review provides an updated perspective on the topic, with a focus on published work from the last five years. The scope of this review chapter covers any service provided to the grid by wind farms, with a focus on wind farm control level perspectives, covering the following:

- Mandatory requirements;
- Market incentivised services;
- Academic proposals for future contributions;
- Policy case studies of grids with high levels of wind generation.

Chapter 4. A Literature Review of Wind Farm Control for Grid Support

This scope does not cover individual turbine-level strategies or power electronic solutions other than where this intersects with farm-level control strategies such as virtual synchronous machines (VSMs). The three grid systems with high levels of wind penetration that will be considered as case studies are Great Britain, Denmark, and the Island of Ireland. Due to the high levels of wind generation in these grids, they can be considered as examples of how other power grids will adapt as they increase their share of generation from wind power. The chapter then considers how these grid codes might change in future, focusing mainly on grid code changes for Great Britain, in the view of the recent “Operating Zero Carbon GB in 2025” strategy issued by the British government in several publications [84]. The chapter then reviews the existing academic literature for how wind farm control can be used to provide grid support services. By considering novel academic research, possibilities for future provision technologies can be seen as the increase in grid penetration of wind energy necessitates innovative solutions.

The scope of the review was chosen to consider only the academic literature touching the topics of wind farm control, including three level strategies for clusters of wind arrays. This omits research and modelling based solely on single wind turbines. In addition to this, the control systems of turbine-level power electronics are reviewed at only a basic level.

4.2 Review of Ancillary Services with Participation of Wind Power

This section mostly focuses on power grids that have a high level of wind generation; this includes the grid codes of the Danish grid (Energinet), the Great Britain grid (National Grid), and the Ireland grid (EirGrid). As these three regions have high levels of wind generation at present, they provide a glimpse of how other power grids might look as wind energy increases rapidly as a source of electricity generation. A grid code is the rules and regulations that any generator connected to a power grid must abide by.

Several papers [87, 8, 83, 29, 16, 110] compared grid codes of different power systems. However, due to the rapid technological progress in decarbonising energy grids, and subsequent policy changes, a new review of the updated grid codes is required as the most recent of these was conducted in 2019. This review considers three power grids with high levels of wind energy generation, as their policies and rules are likely to be adapted into other power grids as they increase their wind power penetration.

4.2.1 Case Study: National Grid

The National Grid code [49] and Guidance Notes for Power Park Modules [48] state that, presently, the requirements on wind parks for providing grid support are in three areas: voltage control, reactive power control, and droop control. The first two of these are provided by the power electronics and so are outside the scope of this review. For wind farms with a capacity above 50 MW, droop control and frequency support must be provided through different response strategies at the instruction of National Grid. The “limited sensitivity mode” is the default operation of a wind farm [48], but large wind farms are required to switch to frequency sensitive mode when requested.

Figure 4.1a shows the “limited sensitivity” required frequency droop response of a large wind array as stipulated by National Grid. In this limited sensitivity mode, the wind farm must be curtailed at times where the grid frequency is too high; this mode does not require action when the frequency is too low.

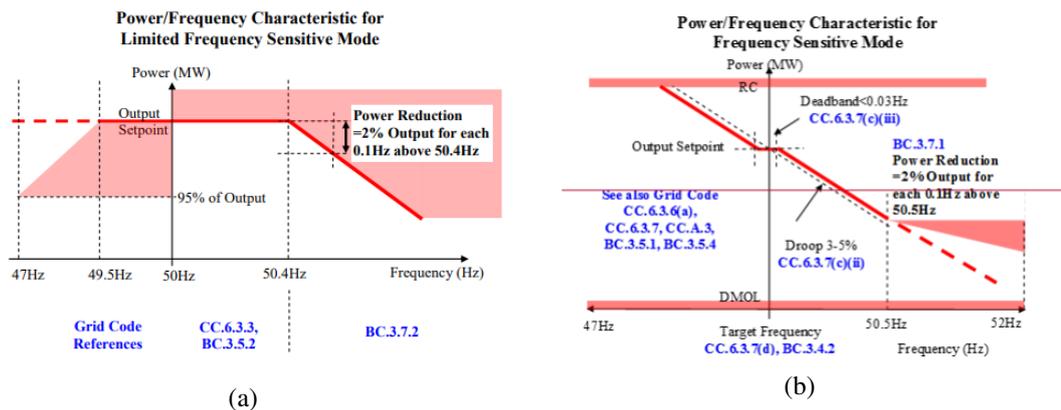


Figure 4.1: (a) The limited sensitivity mode of droop control from [49]. (b) Sensitive mode of droop control from [49].

Figure 4.1b shows the required droop curve when sensitive mode is active. In this mode, the wind farm is required to provide changes in power output at grid frequencies both higher and lower than the normal grid frequency of 50 Hz. The wind farm is required to hold a 10% reserve of power output so that at times of low frequency the power output of the farm can be increased. At the desired grid frequency of 50 Hz, there is a narrow deadband of 50 ± 0.015 Hz where no droop action is required.

Chapter 4. A Literature Review of Wind Farm Control for Grid Support

A more detailed analysis of how the wind farm operators abide by the rules of NG is discussed in [84], so only a brief summary is provided here. In situations of high grid frequency, the response of the wind farm is provided relative to the power output of the wind farm at the time when the grid first exceeds the upper frequency threshold of 50.015 Hz. This locks the set-point of the wind farm until the frequency moves back below 50.015 Hz, regardless of changes in the wind speed at the wind farm. This paper notes that this can lead to large steps in power when the wind farm is released from the lock, which has undesirable impacts on both the wind turbines and the grid, which is an area where the policy should be reviewed as this is undesirable for all stakeholders. Reference [84] suggests that Delta control could be a way to mitigate this issue. Delta control is the strategy of curtailing by a set amount, usually 10% of available power, to ensure a generating unit has a reserve of power that it can deliver in the event of a deviation of grid frequency. This would be a far more beneficial approach as the level of reserve is held in relation to the level of power available across a wind farm, so would be adjusted if there was a significant change in the wind speed, including allowing more power generation if the wind speed increases.

4.2.2 Case Study: EirGrid

The Irish grid code has similar requirements to the GB grid code for how a wind plant should provide droop control [33], with wind farms able to operate in both sensitive and nonsensitive modes.

Figure 4.2a,b show the required droop curves for times when the wind farm is not in sensitive mode and where the frequency of the grid is either very low or very high, respectively. It can be seen that if taken as a combined strategy, this would constitute a mandated droop control with a very wide deadband.

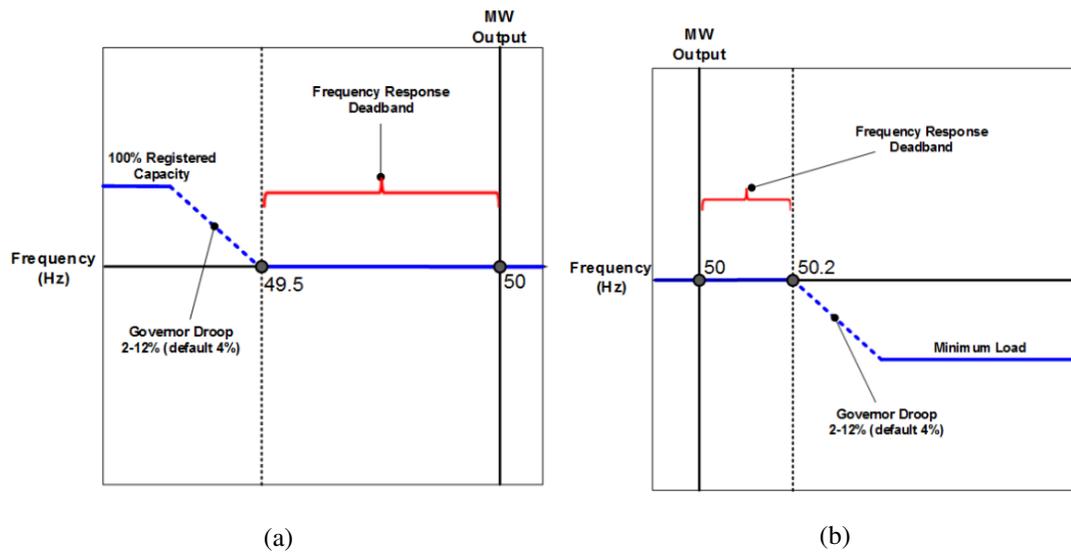


Figure 4.2: **(a)** Mandatory droop requirements from the Irish grid [33] at dangerously low frequencies. **(b)** Mandatory droop requirements from the Irish grid at dangerously high frequencies.

Figure 4.3 shows the droop curve for when a wind farm is in sensitive mode. The grid code stipulates that the minimum reserve in frequency sensitive mode is 5% reserve. It can be seen that, just as in the GB grid, there is a narrow deadband of frequencies where no action is taken.

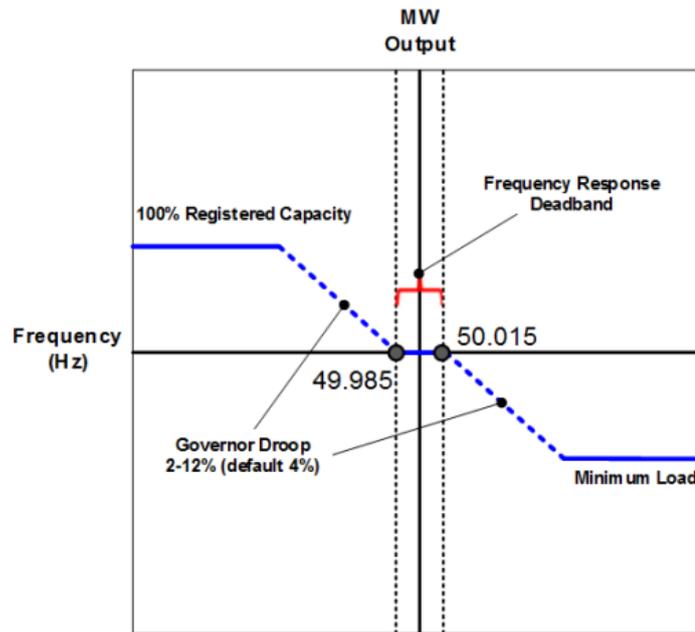


Figure 4.3: Droop requirements from the Irish grid for when the frequency sensitive mode is active.

4.2.3 Case Study: Energinet

The Danish grid [34] has a set of rules for a nonsensitive mode and a sensitive mode for droop control. As in the GB grid code, there is a requirement that generators must curtail their power output at times of very high grid frequency.

Figure 4.4 shows the requirements for droop control at high grid frequencies. As with the GB grid code and the island of Ireland grid code, this requirement is always stipulated due to the essential requirement to balance supply and demand with a power grid. This is required for all grid-connected wind turbines and wind farms.

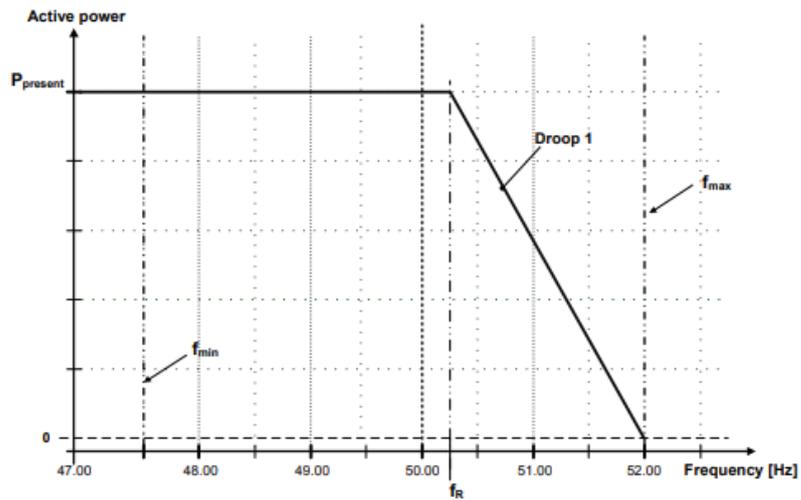
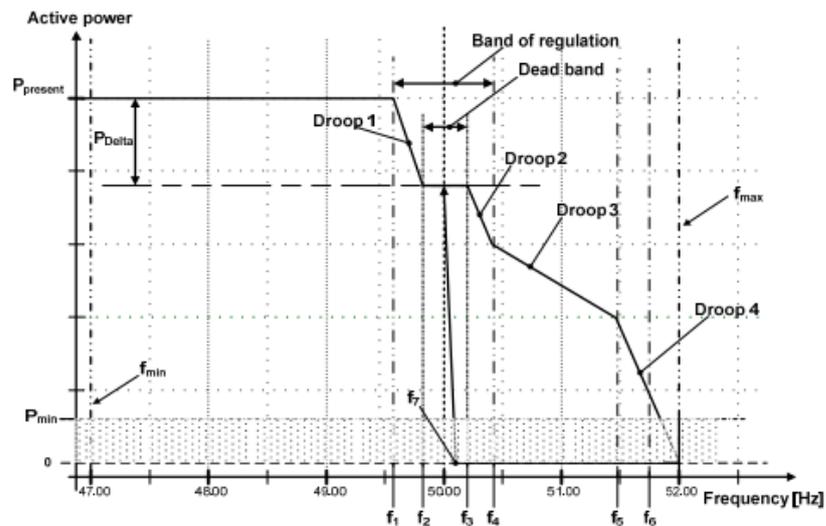
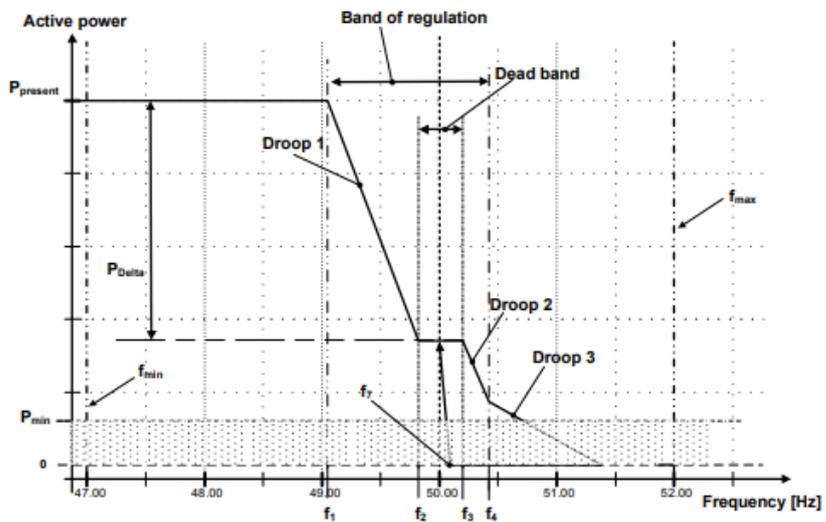


Figure 4.4: The mandatory droop control required at dangerously high frequencies from the Danish grid code [34].

Figure 4.5 shows the droop control frequency strategies for sensitive mode operation as stipulated by the Danish grid code [34]. In this mode, the wind plant must have the ability to set the values of the frequencies (f_{min} , f_{max} , f_1 to f_7) shown in Figure 4.5 to any values between 47 and 52 Hz. The grid code states that the reasoning behind this is so that the droop curve can be changed to produce different curves as required to improve power stability. It also states that “In case of grid frequencies above f_5 , upward regulation of the wind power plant cannot be commenced until the grid frequency is lower than f_7 ”. This is only a grid requirement for wind farms with a rating higher than 25 MW.



(a)



(b)

Figure 4.5: The plots shown here are the sensitive mode of droop control in the Danish grid from Danish grid code [34] The plots show (a) a case with low levels of reserve; (b) a higher level of reserve.

4.2.4 Discussion

In each of these three case studies, there is either a requirement or an allowance for wind farms to participate in grid support through droop control responses. All three grids have similar rules where there are mandatory rules for extreme events such as overfrequency that will always require some form of droop controller. It can also be seen that each of the codes requires a

similar frequency-sensitive approach where the wind farm does not have to change its power output within a narrow deadband near to the desired grid frequency. However, the Danish grid has far more strict requirements for large wind plants, which gives an idea of where the Irish and GB grid codes could develop further in the future. These codes are likely to require further development over the next five years as more nonsynchronous generation is added into the energy mix. Loss of system inertia means that in Denmark further developments in the grid code could still be required, as it is supported by synchronous connection to Germany in its western continental grid and to Sweden in its eastern grid. The connection to other power grids provides Denmark with synchronous generation that can support its grid. While the future design of power systems is uncertain, some ideas of how they could develop are provided by considering the academic literature. A key area where the grid codes will need to be developed is in inertial response from wind farms. It seems likely that this will be mandated for wind in the future due to the wealth of proposals for how wind farms can provide inertial response in the academic literature. These ideas are extensively covered in Sections 4.4–4.7 of this review. Section 4.4 details how virtual synchronous machines can be used as a method for providing grid support from a wind farm.

In Section 4.5, wind farm set-point power tracking for wind farms is reviewed. Building on Section 4.5, Section 4.6 discusses farm-level optimisation approaches for controlling wind farms. The utilisation of stored energy that already exists in the current hardware, such as rotor inertia or capacitance in an HVDC cable, is reviewed in Section 4.7.

Finally, Section 4.8 presents approaches that do not fit easily into any other category.

4.3 Review of Ancillary Services without Participation of Wind Power

4.3.1 Black Start

Wind farms do not currently provide black start capability. A National Grid report [47] notes large wind arrays’ “inability to self start” and wind energy being “grid following” rather than “grid forming” as specific weaknesses. It does, however, suggest that large wind energy arrays have strengths in supporting the latter stages of any restart, stating that “Large wind sites are

Chapter 4. A Literature Review of Wind Farm Control for Grid Support

a good supporting asset for network restoration as the majority of the sites can latch onto a grid signal and then provide balancing services, reactive power and frequency control". The report proposes that energy storage could be used to provide enough power to restart a wind farm following a blackout, suggesting that an auxiliary power supply of 5 to 10% of the farm's nameplate capacity would be required to accommodate black start capability. It continues on to suggest that the level of storage could be reduced if the farm was restarted in stages, allowing for the restarted turbine's power to be used to restart other areas of a wind farm. It is clear that any black start approach by a wind farm would require additional hardware, such as a backup generator or battery storage, without much scope for wind farm control strategies.

4.3.2 Inertial Response

Since wind turbines are decoupled from the grid through their power electronics, they do not naturally provide an inertial response. Small-scale studies show that wind turbines could provide fast frequency response close to inertial timescales through a change in the algorithms used in the power electronic converters. It should be noted that the converters are not directly contributing inertia, but are providing a fast power change that has an equivalent impact on providing inertia.

4.3.3 Future Participation

There has been a focus on the impact of the changing makeup of energy generation in the GB power system. For example, [84] considered how a zero-carbon grid with limited levels of inertia can still have a stable frequency. It was found that events with a lower rate of change of frequency (RoCoF) (up to 0.125 Hz/s) can be contained with present measures. However, these measures are insufficient for events of greater magnitude, becoming inadequate for events with higher rate of change of frequency (0.5 Hz/s). In the specific context of wind energy, the paper notes that wind can currently participate in dynamic primary, secondary, and high-frequency response via the mandatory frequency response (MFR) market. It suggests that wind energy can provide dynamic moderation and containment services in the future but that this will only be the case at times of high wind resource, implying that energy storage will be required for offering an all-time solution.

4.3.4 Discussion

There are several areas of ancillary service provision in which wind farms do not currently participate. This is primarily the result of technical barriers based around the present design of wind turbines, which requires decoupling of the turbine generator from the grid. However, this barrier can be overcome with innovation, and this will likely need to happen in the next decade as the grid is decarbonised and asynchronous energy source begins to dominate generation. Through innovation, grid support can be provided in part from existing hardware without the requirement of large amounts of new expensive equipment such as batteries needing to be added into power systems. These potential innovations are detailed in the remainder of this review, which considers novel academic research.

4.4 Virtual Synchronous Machines

A virtual synchronous machine (VSM) is a control algorithm for a power inverter that emulates the grid-supporting properties of a traditional synchronous generator.

A comprehensive assessment of virtual synchronous machine algorithms is detailed in [7]. The paper breaks the approaches used in VSM down into two groups: high- and low-order VSM algorithms. The high-order algorithms are based on models of synchronous machines with complexity of greater than second order, whereas the lower-order algorithms are based on either second-order models or a droop-control-based approach. Some models require an external energy storage in the system, whereas others do not. The higher-order models generally try to mimic the inertial behaviour of a synchronous machine, whereas the lower-order approaches only mimic the droop control mechanism based on the swing equation.

Reference [74] discusses the mechanism for how VSMs work in the specific context of uses with wind turbine in the case of DFIG turbines and type 4 turbines (turbines that pass all of their output power through one back-to-back converter) that are decoupled from the grid through their power electronics. The paper suggests that the inclusion of energy storage improves the inertial response of wind turbine in below rated conditions for the type 4 case of turbines.

A trial inclusion of applying grid-forming converters to a wind farm to study the impacts

that they could on have contributing to grid stability is presented in [93]. The study shows that small extractions of power for inertia do not have a noticeable effect on the wind turbines, but they did show that there would be a noticeable impact in a synthetic event with a large change in frequency.

It is clear from the academic literature that VSMs can provide a method of grid support but are usually considered at the wind turbine level; hence, further discussion is outside the scope of this review.

4.5 Wind Farm Set-Point Power Control

Wind farm set-point control is the method of having a wind farm output a desired power output. This is usually achieved by having a central wind farm controller that distributes either turbine-level power set-points or reductions in power to each wind turbine in a farm so that the desired power output is achieved.

The research from the University of Strathclyde considers a decentralised hierarchical two-level approach based on the method detailed in [54, 100]. The two level control structure is facilitated through a power-adjusting controller (PAC) fitted to each turbine, as described in [98, 100]. This method allocates changes in power to the turbines based on turbine operational information which is sent from the turbines' PACs to a central controller as a series of binary flags. The central controller distributes changes in power to each of the wind turbines within a wind farm based on the turbine level information. These power changes are then implemented by the wind farm controller. This farm level controller architecture can be seen in Figure 4.6.

The PAC is designed so that it can be fitted to any wind turbine, including retrospectively, in order to enable any wind turbine to vary its power output flexibly and dynamically, including for the purposes of curtailment. The PAC implements changes in power through increments to the demanded generator torque. Where possible, changes in rotor speed caused by the increments in torque are ameliorated through increments in the pitch angle. Any remaining induced change to the rotor speed is input as a negative increment on the measured rotor speed, hence preventing the turbine controller from taking action to countermand the action of the PAC. The PAC is an augmentation to a wind turbine's controller that is, from the turbine's controller's perspective, essentially feedforward in nature (in the sense that from the controller's perspect-

ive the system dynamics are unchanged). The advantage of the PAC approach is that when requesting a reduction from the uncurtailed power, it maintains the generator speed at the same value, which means that curtailment is unlikely to excite any structural modes in the wind turbine more than would be the case without the PAC. The wind farm controller uses the PAC flags to keep all the wind turbines in a safe operational envelope through a system of flags which signal how far from the safety limits the wind turbine is operating.

This PAC-based wind farm control approach is expanded on in [108], which details wind farm dispatch strategies to reduce turbine component fatigue. These strategies are also shown to be effective at reducing fatigue in a delta control curtailment [99].

A closed-loop approach of active power control at the wind farm level is presented in [22]. The paper shows an improved power tracking at the farm level in a disturbance through the presented method. However, whilst this work is interesting and is a useful first look at this issue, the turbine models used are simple second-order transfer models and so might not be representative of an actual turbine.

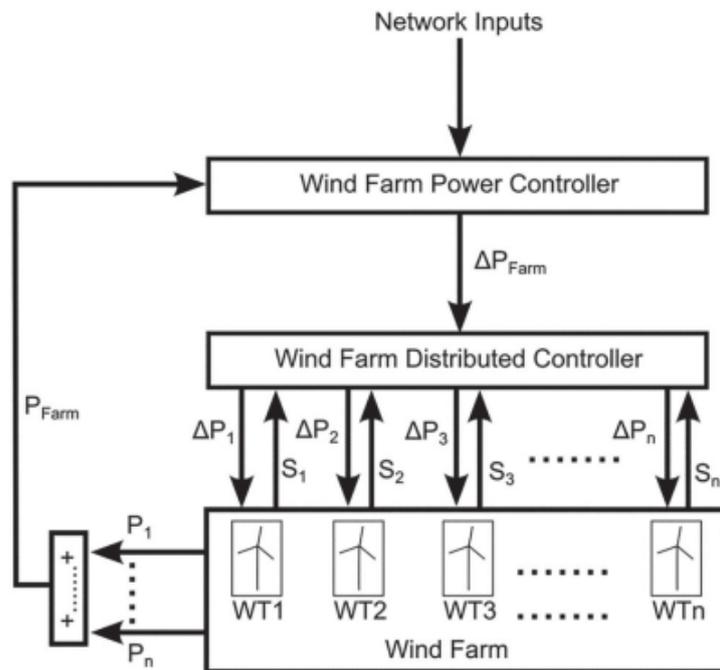


Figure 4.6: The generic form of the wind farm control architecture as shown in [100].

Chapter 4. A Literature Review of Wind Farm Control for Grid Support

The research presented in [53] proposes an algorithm for a cluster of wind farms to control active and reactive power outputs with a two-level approach. The upper-level algorithm distributes active and reactive power set-points between the wind farms, and the lower-level algorithm at the wind farm level distributes power set-points amongst the wind turbines to achieve the active and reactive power set-points using a model-predictive-control approach. The paper shows, in a case study, that the point of connection voltage can be controlled using this approach. This is important as this controllability of power output is the underlying mechanism through which ancillary services such as synthetic inertial responses and droop control can be provided from a wind farm.

A three-layer approach (similar to [53]) using nonlinear model-predictive control for providing a frequency response from a wind farm is presented in [64]. The highest layer calculates a change to the farm active power reference to provide a proportionate frequency response. This change in power reference is then distributed amongst the wind turbines using a nonlinear model-predictive control approach through the use of a cost function. The lowest level of control is the turbine-level controllers where the change in power is implemented. The paper shows that the algorithm is effective but does not consider factors such as wake effects or structural fatigue in its analysis.

An active power dispatch algorithm for a wind farm to provide grid frequency regulation while also considering fatigue loads is considered in [72]. It achieves this through an optimisation-based approach based on a cost function which minimises variation of shaft torque and thrust forces of the wind turbines. While the results show a decrease in damage-equivalent loads, only one wind speed was used in the case studies, so these results might not be representative and a full validation should be performed to prove that this method works over a range of wind speeds, turbulences, and wind directions.

In [104], an open-loop wind farm controller is used to provide active power control to a set-point value provided by the transmission system operator (TSO). As the wind farm is providing frequency support, it is necessary to maintain a level of reserve power. The requested curtailment of the wind farm as a whole is regulated by a PI controller, with the total amount of curtailment distributed around the wind farm equally. Each turbine in the wind farm feeds its available power back to the WFC, which feeds the total to the TSO. Each wind turbine in the

wind farm has a separate PI controller used to track local power demand. There is a provision for curtailment to be redistributed when individual turbines cannot reach the local requested level of curtailment. The main goal of the paper is to expand the model to work in a heavily waked environment. The model for a wind farm controller is tested in SOWFA (Simulator fOr Wind Farm Applications) in a 12 m/s windfield using the NREL (National Renewable Energy Laboratory) 5 MW reference model for a three-by-three wind farm arrangement: firstly with derating allocated at 50% across all of the wind turbines, and again but with deratings of 80% for the first row of wind turbines, 50% for the next row, and 20% for the third. The results show that the controller in both scenarios reduces RMS error and improves CAISO (California Independent System Operator) accuracy score (a requirement in California for ancillary service payments). However, both approaches generally show an increase in damage-equivalent loadings in the tower fore-aft and out-of-plane bending moments, with the 80–50–20 approach generally resulting in higher loads than the 50% approach. The paper does not explicitly state which approach it uses for curtailing the turbines; however, it does cite appropriate methods from the literature.

4.6 Optimisation-Based Dispatch Algorithms

The design of optimisation functions to distribute active power set-points across wind farms has become a popular topic in recent years. These approaches use cost-function-based approaches to find an optimal farm-wide operation rather than only considering the turbine level. In the literature, a variety of criteria are used in the cost function, depending on the goal of the model.

A three-layer approach (with a similar farm approach to [53, 102]) to dispatch active power set-points across a wind farm is used in the research shown in [69]. This approach considers the wind farm as a series of clusters of wind turbines and considers the operational state of the turbines within these clusters when allocating the turbine-level power set-points. The results show that this yields a more stable accurate power generation at the farm level compared to an approach of derating the turbines at the turbine level. The approach shown in [24] also considers a three-layer approach to dispatch active power set-points across a wind farm using a receding horizon distribution strategy to improve the wind farm power tracking error and transmission loss. The paper discusses the practicality of implementing this strategy on an

Chapter 4. A Literature Review of Wind Farm Control for Grid Support

actual wind farm, stating that in practice it would need to operate at a time step of over 1 s rather than the simulated time step of the order of microseconds.

The research presented in [6] considers optimisation strategies for maximising farm power while maintaining kinetic stored energy, maximising stored kinetic energy while maintaining power output, and maximising stored kinetic energy while deloading by a preset amount. As with other approaches in this area, this paper does not include a dynamic wake model so might not be as effective in a real-world environment.

Reference [75] considers an optimisation strategy based on the clustering of wind turbines with similar wind profiles with the aim of improving frequency support in a similar vein to [23]. The paper considers the method of switching between different wind farm operating strategies to maximise injected power stored as kinetic rotational energy in the turbines for frequency support. The three modes considered are maximising total power output, maximising stored kinetic energy while deloading, and minimising stored kinetic energy with nearly maximum power output. The paper shows an improvement in inertial grid response in a case study. While this approach does show improvement, it does not include a dynamic wake model which, when included, could see these improvements diminish or disappear, as the meandering effect of the wakes has been found to impact other research strategies with similar methodologies [63].

The approach shown in [73] is similar to [23, 75], considering an optimisation-based approach to providing a variable droop strategy considering stored kinetic energy in turbine rotors. The paper shows that through the optimisation and adaption of the droop coefficient, more energy could be released when needed for grid support. As with other papers in this area, a static wind field including wakes was used, so these results might not be reproducible in a dynamic wind field simulation.

The research presented in [23] considers an optimisation algorithm for allocating active power set-points with a cluster of wind turbines using a model-predictive approach for improving power output stability and frequency support provision. The paper shows that the algorithm is effective but notes that further work could be required to improve solving times and eliminating disturbances.

The approach presented in [109] considers two optimisation function approaches for active power dispatch in a wind farm to reduce fatigue loads of the turbines. While these approaches

Chapter 4. A Literature Review of Wind Farm Control for Grid Support

are based on minimising the standard deviation of turbine bending moments (which is known to be an inaccurate approach [63]), the paper does acknowledge this and uses a modified approach to mitigate this flaw. The results show that the two strategies result in lower levels of fatigue while approximately tracking a power set-point. However, the results shown only represent a single simulation without wake effects or variations in global turbulence at the farm level considered, so further work would be required to ensure that the approach is generalisable for a range of scenarios and at different scales.

Fatigue modelling using the NREL 5 MW wind turbine is shown in [76] for a wind farm comprising five wind turbines curtailed to 10 MW. A baseline was set of curtailing the power output to 2 MW. A wind farm controller was then implemented to control the power output using a linear–quadratic–Gaussian control optimisation strategy to allocate set-points across the wind farm based on the thrust at each wind turbine, reasoning that variations in the tower moment are mostly the result of variation in the thrust force. The result is that the total power output from the wind farm is maintained at 10 MW but the average reduction in fatigue load is 35%. With the level of curtailment used in this paper, there is a large scope for optimising for the minimisation of fatigue, so a large improvement in the level of fatigue would be expected. It would be interesting to see the impact that this approach would have in a more realistic curtailment environment such as a delta curtailment of 10% or following a set-point which is much higher than the one used. The model used does not account for wakes within the farm, which is a significant omission as wakes can have a large impact on the damage-equivalent loads in the wind farm.

The use of the partial derivative of load with respect to reference power in order to reduce the fatigue across a wind farm is considered by [111]. The wind farm control architecture used is one where most of the control is centralised to the wind farm level with each turbine’s generator torque reference and pitch reference being controlled at this level. Using measurements from the wind turbines, the turbine level controller sends a partial derivative of the structural loads with respect to the power reference to the wind farm controller, which sends back a power reference. These power references are found through the use of an optimisation at the wind turbine level which finds the optimal set of power references for the wind farm to produce a requested farm-wide-level power total while also minimising the damage-equivalent loads of

the turbines. The paper does not discuss the computational time required by the optimisation process (which would need to be at least real time for any hardware application), and the paper also fails to mention the turbulence intensity of the wind field used, which makes it difficult to draw concrete conclusions from the results presented.

4.7 Stored-Energy-Based Solutions

This section reviews the academic literature detailing the use of existing energy stored within a wind farm for ancillary service provision.

It is useful, first, to consider some of the different types of frequency response that are typically delivered following a large grid frequency event, as shown in Figure 4.7. The grid frequency is determined by the balance of generation and demand, with frequency increasing when generation is greater than demand and decreasing when the reverse is true, i.e., when demand is greater than generation. In an event of this kind, historically, the initial change in system frequency is dominated by the size of the power imbalance and by the inertia of the large mechanical inertia of synchronous generators, which are electromagnetically linked to the grid and so rotate at grid frequency. As large offshore arrays are decoupled from the grid due to the power being transported by HVDC links, they cannot directly provide inertia in this way. Hence, the topic of how HVDC-connected wind power can be used for the provision of synthetic inertia is an active area for research.

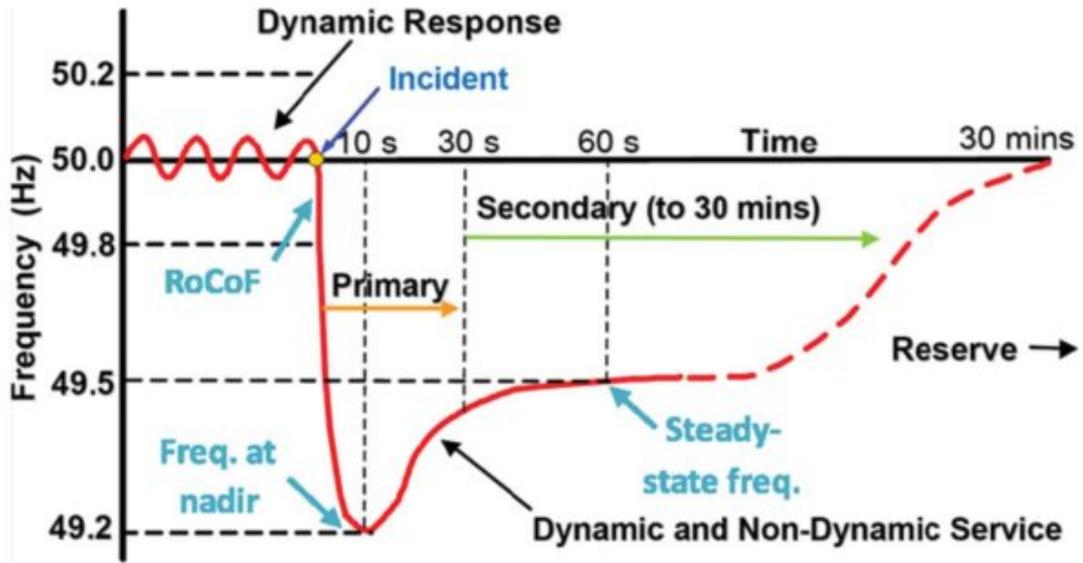


Figure 4.7: An example of the different responses to a grid frequency event [94].

Frequency events such as the one shown in Figure 4.7 are not common, with the National Grid’s Frequency Risk and Control Report [50] describing frequencies between 49.2 Hz and 49.5 Hz as happening infrequently, and conditions of frequencies less than 49.2 Hz as being not often tested in real-world conditions. As large grid frequency deviations are not common, particularly of the magnitude seen in Figure 4.7, but must be corrected urgently, wind turbine loads are considered to be less important than maintaining grid stability. This urgency is necessary, as if a blackout occurs it is often disastrous due to society’s dependence on electricity and the difficulty of a black start, and will also be far more expensive than the cost of a slight increase in fatigue.

The response of the wind farm can be broken down into two parts: control of the wind turbines and control of the HVDC.

In the event that inertial response is required, wind turbines can temporarily increase their power output above the steady-state power taken from the wind as there is some energy stored in the rotating inertia of each wind turbine’s rotor. The amount of energy in the rotor is given by:

$$E = \frac{1}{2} J \omega^2 \quad (4.1)$$

where E is the energy stored in the rotor, J is the inertia of the rotor, and ω is the rotational

speed of the rotor. Energy stored in the rotating inertia is often considered for very fast responses where a rapid increase in power is required. After the inertia of the wind turbine's rotor has been used, the wind turbines must be derated briefly in order to return the generator speed to its standard operating value as the inertia of the rotor is restored.

An approach for utilising the stored kinetic energy in the rotors of DFIG turbines is presented in [46], with results shown for a wind farm over a range of wind speed conditions. The aforementioned power-adjusting controller (PAC) and hierarchical structure [100] were used to provide ancillary services in [90]; specifically, a wind farm controller was developed that provides droop control and synthetic inertia. The controller is capable of providing both droop and synthetic inertia simultaneously or either response individually. When providing synthetic inertia, the usual limits on rate of change of power from the PAC are suspended to enable an appropriately fast response. Built-in safety measures in the PAC ensure that operation is kept within a safe operational envelope defined by the control designer. The controller accounts for turbine availability within the wind farm and, if the requested change in power exceeds that which is available, the maximum available response is provided. The droop controller is similar to that initially proposed in [97], incorporating a deadband as per the National Grid standards described in Section 4.2.1. The droop controller described does not consider the wind farm as a whole, but each turbine in the farm instead provides a droop response individually. There is scope for this work to be expanded to account for alternative distributed control approaches such as the method described in [99], which demonstrates that a distributed control approach can greatly reduce the damage-equivalent loads on the wind turbines. Further work by the same research group demonstrated the implementation of a dispatch algorithm for holding sufficient reserve in a wind farm so that it can provide a 30 s primary response if required [27]. There was a significant reduction in the fatigue loads on the wind turbine structural components; however, this was coupled with a reduction in energy capture.

A novel approach is presented in [59], which suggests using the energy stored in the HVDC link and also the rotational inertia of the wind turbines in order to provide frequency response. The model used is a simplified version of an HVDC connection with resistance in the cable and higher-order dynamics neglected. In this simplified model, the rate of change of the DC link voltage can be found from the difference between the current at the wind farm side and the

grid side. Using the energy stored in the HVDC link has the advantage of being a quick way of bringing increased amounts of power into the grid when required, with the amount of energy stored in the link given by:

$$E_{DC}(t) = \frac{1}{2}CV_{DC}(t)^2 \quad (4.2)$$

and the power extracted from it given by:

$$P(t) = CV_{DC}(t)\frac{d}{dt}V_{DC}(t) \quad (4.3)$$

Three methods are presented to provide frequency support to the grid using an HVDC-connected wind farm. The first is to only change the power output of the wind farm considering a communication lag to account for the signal moving from the grid-side converter to either the wind turbines or a wind farm controller. The second is to use the energy stored in the HVDC link using the grid voltage at the point of common coupling (PCC). The limit of this approach is the capacitance of the DC link which can be expensive to scale. The third is a combination of the first two approaches. The proposed approach is that very-high-frequency responses can be provided from the stored capacitance in the cable, whereas lower-frequency responses can be provided using wind farm control. While the paper omits fatigue loads modelling of the wind turbines, the proposed strategy could potentially result in reduced fatigue on the wind farm as the proposed method would reduce changes in the wind farm power output, resulting in less fatigue. The paper discusses the trade-off between the capacitance of the DC link and the time constant, which is used to approximately separate the response of the DC link and the wind farm. This proposed controller infrastructure can be seen in Figure 4.8.

Chapter 4. A Literature Review of Wind Farm Control for Grid Support

line of communication from the synchronous generator and the other using changes in the voltage of the HVDC link to change the frequency reference of the wind farm side VSC which then changes the offshore grid frequency, which can be used by the wind turbine converters as a signal to increase power output. The idea behind the signal from the synchronous generator is to send a request for a fast inertial response from the wind turbine, whereas the communication-less approach is for inertial frequency support. The paper considers two approaches based on differing sizes of deadband. The results show that having a narrow deadband for frequency with a shallower slope results in improved operation in normal conditions, but that having a wider deadband with a steeper gradient is better in fault conditions. The paper uses a model with a small synchronous generator that has one quarter of the capacity of the wind farm and subsequent HVDC link, but suggests that five percent of the size should be sufficient, or even another approach without a synchronous machine, but notes that this would lead to a weaker response. Further work is needed in this concept as it decouples the wind turbines from the offshore grid using a back-to-back converter with the assumption that the wind turbines can provide sufficient power.

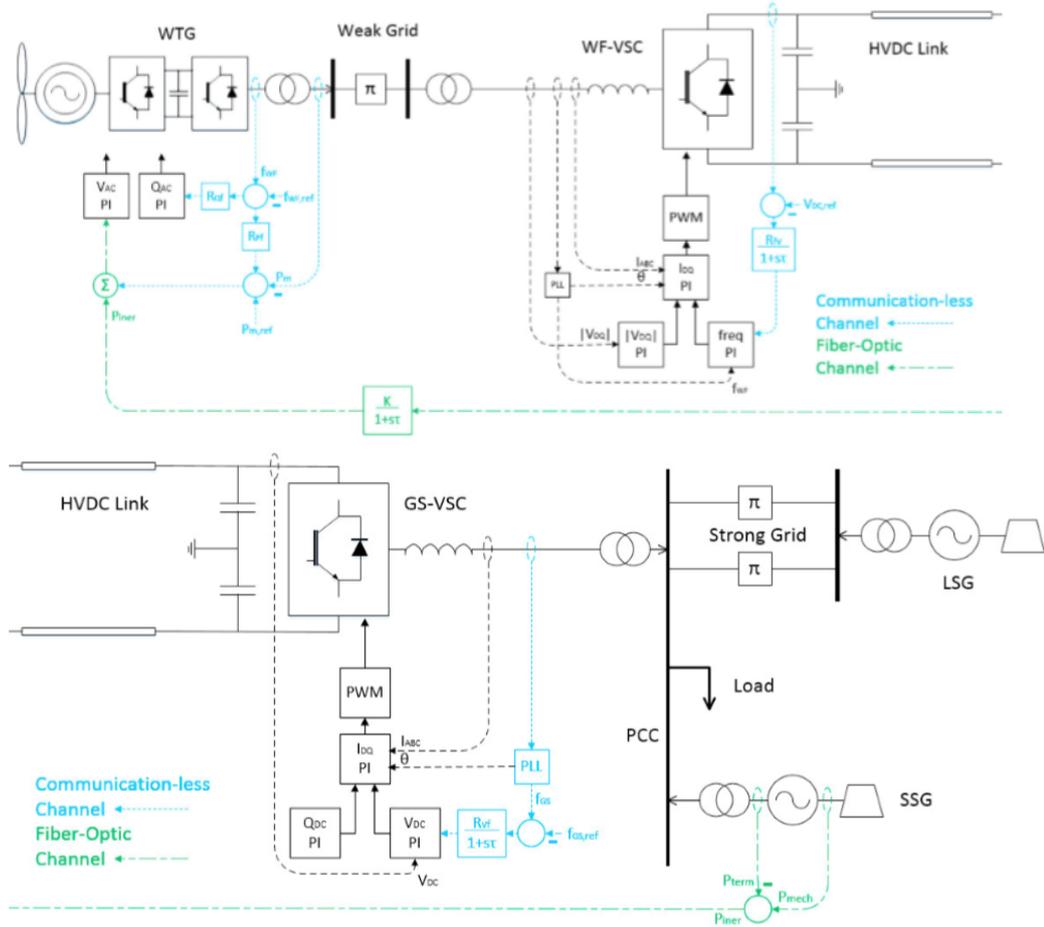


Figure 4.9: Offshore reference circuit proposed by [82].

Further work in this area is shown in [13], expanding on [82], by considering a large synchronous generator communicating the information to a large offshore wind farm of a similar rating. This paper considers a more wind-farm-based approach as it considers how the turbines would need to be derated in order to have some capacity to increase power output when required. The additional consideration here is that a conventional synchronous generator that already exists can be used for obtaining the information for the wind farm as long as it is connected near the PCC of the wind farm, which will make the implementation of this more cost-effective. The paper suggests that this would be beneficial for load balancing for times when the wind farm cannot produce a sufficient amount of wind. The proposed approach is one which considers both frequency support and also voltage support with the frequency

support being provided through wind turbine control and the voltage support by the onshore HVDC converter. This echoes other papers' ideas that the energy stored in the HVDC link can be utilised for services which are required on electrical time steps and that the wind farm can provide the support for the slower requirements. The frequency support approach is similar to the others discussed, using the frequency of the synchronous generator as a signal to the wind farm generators to change their power output to provide frequency response. The provision of voltage support is supplied by using the synchronous generator. It performs this by estimating the voltage and reactive power at the synchronous generator and then used by the HVDC converter on a per-unit basis to emulate the output of the synchronous generator by changing the reactive power set-point of the converter. The approach used considers the wind turbines only electrically, so this approach does not provide any information about the impact on wind farm damage-equivalent loads.

This research theme is continued in [20] by considering how a wind farm with wind turbines equipped with PACs (discussed in Section 4.5 and detailed in [100]) can be included in a GRF concept while also (similarly to [59]) considering the energy stored in the capacitance of the HVDC connection cabling for a rapid response. Despite considering how the turbine can provide an increase in power output when required, this paper also assumes this power rather than considering how this power increase would impact the turbines.

The research presented in [41] considers a model for controlling the grid frequency of a wind farm connected to a conventional generator, including a control system to make the wind farm aware of power variations. The paper focuses on the development of an optimisation-based approach for a multiobjective function using a particle swarm optimisation algorithm to find the tuning parameters for this signal of active power variation fed to the wind farm. The paper shows that the proposed approach has positive results but only in a single simulation case study. While this is not exactly the same as GRF, it is a similar approach, which shows that the concept has potential for further development.

4.8 Other Approaches

Reference [71] considers a three-level approach to dispatch active power set-points across a wind farm based on the fuzzy c-means clustering algorithm and model-predictive-control

method. The paper shows that the proposed approach is effective when compared to a proportional curtailment at the lower layer. The paper also includes a robustness analysis for the results. However, it does acknowledge several limitations in the modelling such as the models being simple first-order systems and requiring shorter controller time steps. It also states that for an industrial implementation, further modelling including wake effects, wind conditions, and load conditions would be required. The method shown in [95] considers an algorithm for derating wind turbines to maximise the amount of power the wind farm has in reserve for the purpose of ancillary service provision. Similar to other approaches, which aim to maximise the total power output, it achieves this by reducing the power output from the wind turbines which are in the front row more than those in subsequent rows. Hence, the wake effects downwind are reduced, meaning that the downstream wind speeds are increased and so more power is available. The levels of curtailment considered in scenario 1 are very high as scenario 1 considers a set-point power of 20 MW compared to an uncurtailed power output of 48 MW. In the case of scenario 3, the level of curtailment is lower but the improvement in total power is smaller. When the wind direction is varied, there is a large change in the increase of potential power, with scenario 3 at times having no improvement. This paper considers wake effects in a quasi-static wind field, and other research in this area shows that when turbulence is introduced into the model, increases in power output are likely to disappear [63]. When the prevailing direction of the wind changes even slightly, the potential power could be liable to change. In practice, wakes have a tendency to meander as they propagate through the wind farm, which would make it difficult to know what the optimum allocation of set-points would be. This paper does not consider damage-equivalent loads or fatigue, but suggests it as a future area of research.

4.9 Discussion

It is clear that the present grid codes of even the most wind-penetrated power grids will need to be developed further in the near term as wind energy increasingly dominates generation in high-resource areas. These advancements will likely be driven by the pipeline of innovation from academia to deployment; hence, by reviewing the present academic literature, it can be seen that the existing grid codes are likely to increasingly require inertial provision from wind

Chapter 4. A Literature Review of Wind Farm Control for Grid Support

farms in addition to the present requirements to provide droop control. The exact mechanism of how this inertial response will be provided remains to be seen, as there are a range of proposals in the academic literature covering areas from virtual synchronous machines to provision from novel communication topographies, to allow dispatch from existing energy stored in the current hardware found in wind farms, and there is also a wealth of novel wind farm control strategies. Due to the need to rapidly decarbonise in response to the climate crisis, it is likely that many of these areas will be deployed in the next decade to facilitate a higher level of supply not just from wind farms but also from other nonsynchronous renewable resources such as wave energy or solar power. There could also be further changes in market incentives on wind farms. Previously, governments and grid operators wanted to incentivise the maximum amount of renewable generation possible to combat climate change; as grids become decarbonised, these incentives may need to change further to prioritise stability and ancillary service provision. This will need to be carried out in a careful way to not stifle innovation, by preventing the implementation of novel approaches, while also ensuring that the grid has as little fossil-fuel-based generation as possible.

4.9.1 Structural Modelling

Much of the research in the wind farm control modelling for ancillary service provision neglects structural loading. There is a clear disconnect between the modelling focusing on the power system integration aspect of the modelling and the research looking at fatigue modelling. This disconnect should be addressed, as droop provision could require the power output of a wind farm to change in response to maintain frequency stability. This variation could increase wind turbine component fatigue.

4.9.2 Farm Size Consideration

The presented research in this review covers a range of sizes of wind farms. Much of the research on smaller wind farms only presents a single simulation case. While this can show an indicative example of a control strategy efficacy, a single simulation cannot fully account for the range of conditions seen when accounting for wake meandering. There is a clear gap in the literature for an exercise in quantifying the uncertainty of any single simulation. This would

require a comprehensive validation exercise simulating a vast number of conditions to have a mathematically sound quantification of the inherent uncertainty.

4.9.3 HVDC vs. AC Connection

Much of the research seen in the literature assumes an AC grid connection, as in cases where an HVDC link is used there is a requirement for additional communication pathways due to the decoupling of the HVDC connection. This would incur additional costs, but as wind farms are increasingly being located far offshore, they require a HVDC link. This would also be required in cases where the HVDC link exists and communication pathways need to be retrofitted.

4.10 Concluding Remarks

From this review, it is clear that there is a comprehensive amount of high-quality research on wind farm participation in ancillary service provision. This research covers a wide range of perspectives such as power system modelling, control-based approaches, and fatigue modelling. However, most of the research is in a singular research area and ignores or negates the other areas of the modelling. While it has been shown that wind farms can provide certain services, such as frequency regulation to a grid, the impact of this provision on the fatigue of a wind farm has not been modelled in a simulation that also shows that the provision is actually contributing to the grid stability effectively. There is a clear disconnect between the two approaches to research in this area, between research that is based on a power systems approach and either neglects or simplifies the structural impact on the wind turbines, and research that is based on the structural modelling of the wind turbines and assumes or neglects the power system model. Hence, there is a significant gap in the academic literature regarding how the provision of ancillary service from a wind farm impacts fatigue loads in a dynamic model.

Chapter 5

Development of a Farmwide Power Set-Point Controller for Strathfarm

This chapter introduces Strathfarm, the University of Strathclyde's in-house wind farm modelling software, from the perspective of its existing wind farm controller architecture. This chapter then introduces the Power Adjusting Controller (PAC) including the system of binary flags sent between the central wind farm controller (WFC) and each of the wind turbines in a farm through which the control system operates. Finally, the wind farm controller is discussed with a demonstration of how the anti-windup measures at the farm and turbine level allow for set-point power tracking of the wind farm. Some of the contents of this chapter and chapter six overlap with the Supergen wind hub report 'D3.3 Report on wind farm control algorithms to meet asset management requirements' [108] which was a precursor to this work. This chapter's contribution to knowledge is to develop for the first time a discrete time, PAC based wind farm set-point power tracking controller.

5.1 What is Strathfarm?

Stathfarm is a software program that has been developed at the University of Strathclyde, written primarily as C++ code but also using MATLAB and Simulink for some code and as an interface, for modelling wind farms. Strathfarm is the culmination of many different researchers combining aerodynamic modelling([9, 10]), wind turbine modelling([39, 103]), control

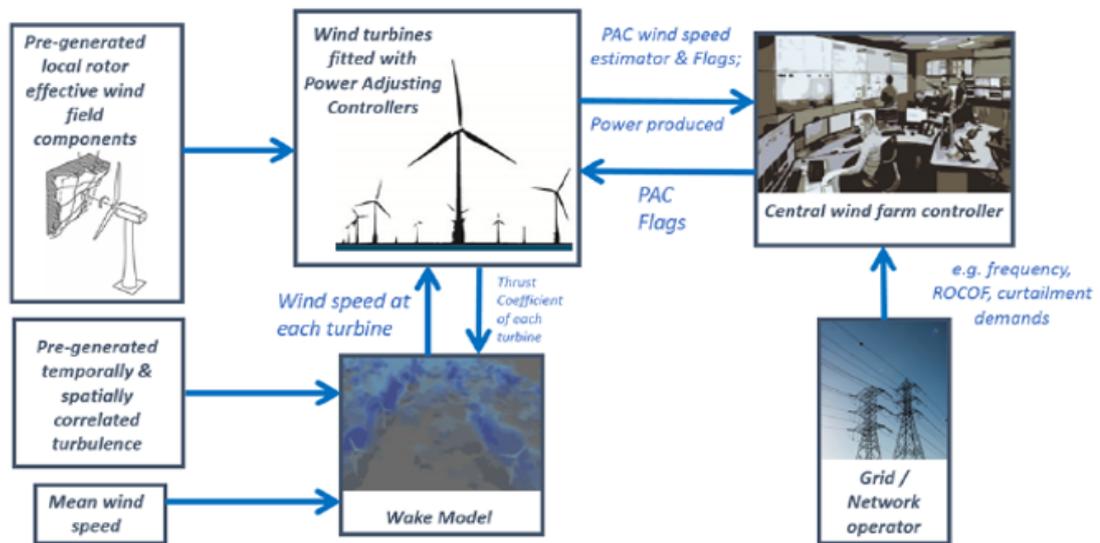


Figure 5.1: A schematic diagram showing how each section of Strathfarm connects together.

engineering ([54, 98]) and power system engineering research ([65, 90, 43]) into a single wind farm simulation software.

Strathfarm is a medium fidelity model which has been validated using higher fidelity but more computationally expensive models. The wake models were validated against high fidelity results from PALM [107] and the structural modelling of the individual wind turbine model (Strathturb) has been validated against Bladed [55, 11]. However, despite being a medium fidelity model Strathfarm has a relatively low computational cost and can simulate up to one hundred wind turbine in real time on a standard desktop computer.

Another advantage of Strathfarm is that it has been designed such that the protocols for the controllers for the wind turbines and the wind farm are industry standard. This means that any controller used in either Strathfarm or Strathturb can be used in a commercial wind turbine or a commercial controller could be tested using either Strathfarm or Strathturb.

Strathfarm's control architecture uses a decentralised, hierarchical, scaleable approach, which is possible due to each wind turbine in Strathfarm having its control system augmented with a Power Adjusting Controller (PAC) which wraps around the existing full envelope controller. The purpose of the PAC is twofold, as it both allows each wind turbine to receive operational instructions, such as requested changes to power output or rated generator speed, while also providing information about the operation of the turbine as an output. In Strathfarm's

control architecture these signals are sent to and from the Wind Farm Controller (WFC).

5.1.1 The Power Adjusting Controller

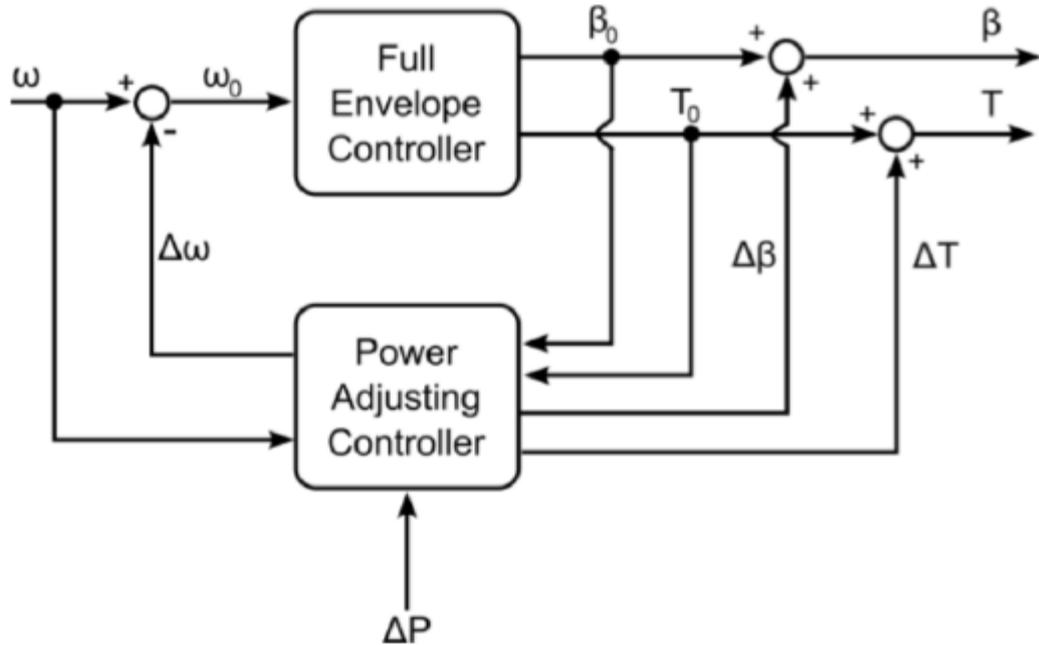


Figure 5.2: The general form of the Power Adjusting Controller (PAC) [99]

A schematic for the PAC's augmentation of the full envelope controller can be seen in Figure (5.2) where ΔP is the requested change in power output of the wind turbine, ω is the measured generator speed, $\Delta\omega$ is an adjustment to the measured generator speed by the PAC, ω_0 is the adjusted generator speed as seen by the existing Full Envelope Controller, β_0 the resultant pitch demand from the Full Envelope Controller, $\Delta\beta$ is the change to the pitch demand from the PAC, β is the pitch demand sent to the turbine, T_0 the resultant torque demand from the full envelope controller, ΔT is the change to the torque demand from the PAC and T is the Torque demand sent to the turbine. Figure (5.2) shows that the PAC wraps around the existing wind turbine controller by feeding forward changes in pitch angle and generator torque to adjust the wind turbine's power output while also modifying the signal of the turbine's generator speed. It should be noted that the PAC does not feedback any signals in it's inclusion to a wind turbine's control system as doing so could result in adverse operational conditions in the existing full

envelope controller. A key advantage of using the PAC for curtailing a wind turbine is that when a reduction in power is requested of a wind turbine fitted with a PAC, the reduction in power is done through pitching even at below rated wind speeds. This is done through the forward paths which modify the torque and pitch outputs from the full envelope controller. This is done such that the operational speed of the turbine does not change from what it would be if the PAC were not present. Hence, any wind turbine curtailment (down-rating) strategy using PACs will not lead to increased excitation of structural modes through increased variation in wind turbine generator speeds.

Control Structure of Strathfarm

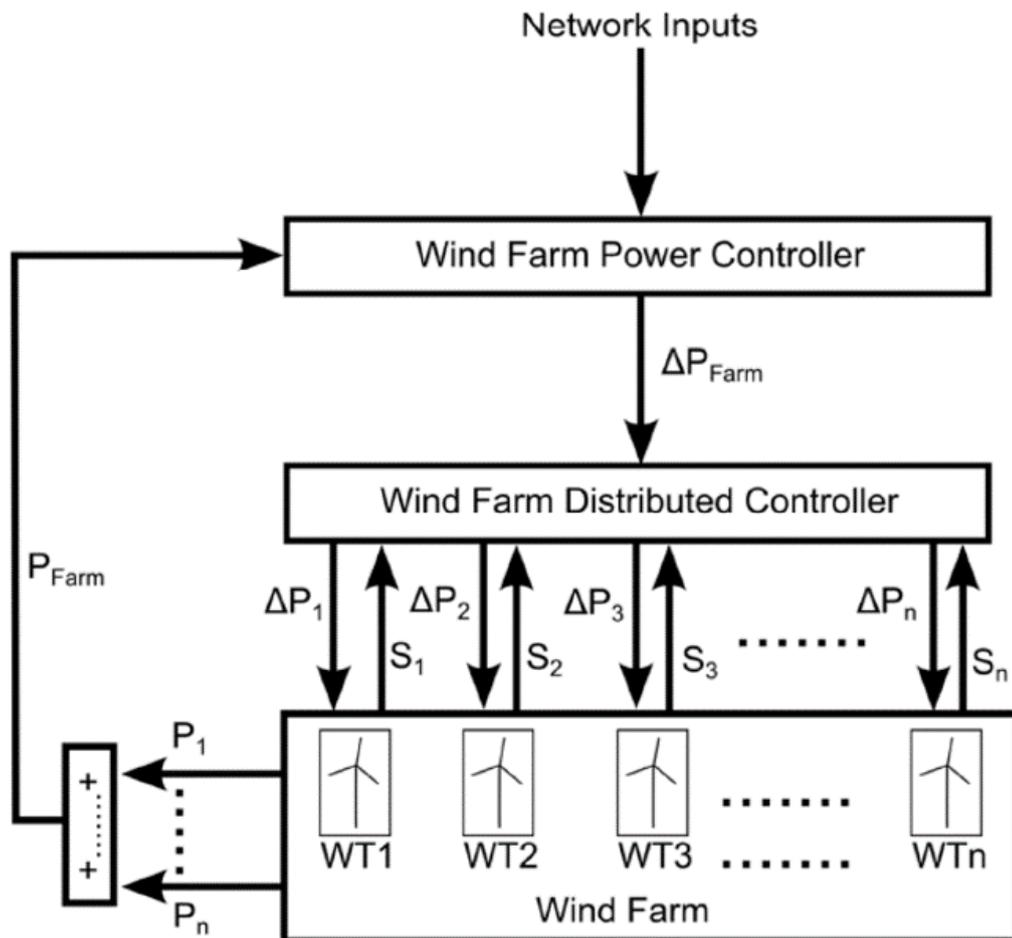


Figure 5.3: The wind farm controller layout used in Strathfarm.

Chapter 5. Development of a Farmwide Power Set-Point Controller for Strathfarm

A diagram of the farm level control system used to regulate power in Strathfarm can be seen in Figure (5.3).

This control structure has the option to include information into the wind farm controller. The external data could be from the network such as frequency or Rate of Change of Frequency (RoCoF), or it could be O & M data related to turbine fatigue, protecting turbines which are known to have evidence of wear or it could be live electricity wholesale price data which could be used to set a level of curtailment for the wind farm. This data is fed into the Wind Farm Power Controller which calculates a required change in power for the wind farm (ΔP_{farm}). The ΔP_{farm} is then fed to the wind farm distributed controller which allocates changes in power to each turbine (ΔP_i) based on information sent through the turbine's PAC's flags (S_i). The total power output of the wind farm is fed back to the Wind Farm Power Controller closing the loop. An example of Strathfarm's operation is presented in this thesis showing a method of outputting a set-point power from the wind farm through including a PI controller within the Wind Farm Power Controller.

Why Use Flags?

The information which is fed from each turbine's PAC to the WFC is almost entirely comprised of binary flags. The reason that flags are used is that they do not introduce feedback into any signals within the wind turbines' or wind farm's control systems. As an additional protection against this the binary flags also, where required, operate with hysteresis loops (Schmitt triggers) which stop certain signals from chattering. The binary flags contain both operational information such as: flags that indicate how close the turbine is to its limit of safe operation as well as turbine level protection information for when a request cannot be fulfilled such as: flags which indicate that requested changes in power (and rates or requested changes in power) are not possible, flags that indicate that the turbine needs protecting due to high turbulence conditions and flags that indicate that the wind speed is too low to use the PAC.

The singular case for where a numerical signal is used is the estimated wind speed at each turbine which can be included as a signal sent to the WFC because it is sufficiently decoupled from wind turbine operation that it will not introduce any feedback into the wind farm control system [98].

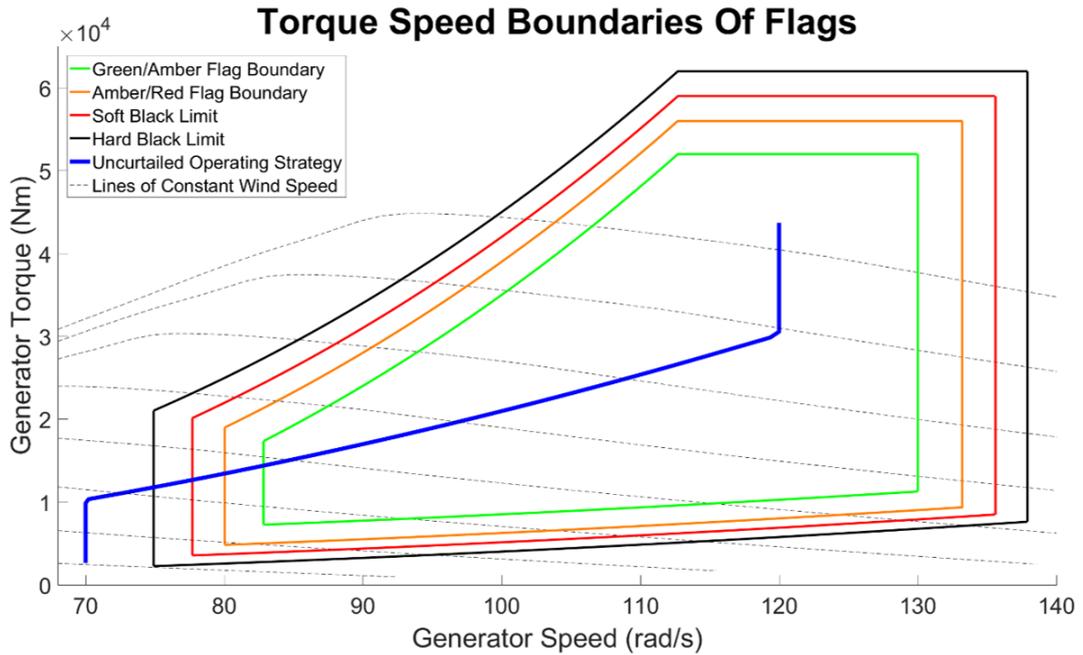


Figure 5.4: The traffic light boundaries for the PAC in the Supergen 5MW Wind Turbine Model.

The boundaries for the binary flag which tell the WFC how close each turbine is to the limits of safe operation, can be seen in Figure (5.4). These “traffic light” flags have been shown to be an effective tool in wind farm control [54]. These flags are used to indicate how close the turbine is to reaching an unsafe operational point which is set by the black limit flag. As the turbine approaches the black limit it passes from a “green” flag to an amber flag then to a red flag. Beyond the red flag the turbine hits a “soft” black limit which is the absolute limit for a wind turbine under normal conditions. The “hard” black limit is the absolute safe operational limit of the wind turbine which should only ever be reach under extreme conditions such as inertial response provision. The traffic light boundaries are tunable within the PAC with only the hard black limits being mandatory as they are usually set by the turbine manufacturer’s specifications. The reasoning behind the hard black limits varies at different locations as the upper speed limit and the upper torque limit are set by the turbine’s maximum operational speed and torque respectively, the upper left limit is to avoid the turbine’s stall region. The low speed limit is used to ensure that the PAC is not active while a wind turbine is in the early stages of startup. The lower torque limit is used to prevent a turbine’s minimum generator torque limit been breached.

5.2 Why Set-point Control?

As wind turbines are increasingly being situated in large offshore arrays with a single point of connection to a power grid they are effectively seen by a power system as a single collective source of electricity generation. This means that if wind farms can act in a similar way to a conventional, traditional thermal generation plant, they can not only be more easily integrated into power grids but also help support them and provide ancillary services.

5.2.1 Wind Farm Controller

The initial step was to design a controller which could achieve set-point regulation at the wind farm level. This had already been shown in previous research [54], However, that model was in continuous time, hence the controller has been discretized in this chapter. This discretization also aligns with the long term development plan for Strathfarm which aims to convert all of Strathfarm's code into C++ to improve the computational speed of the simulations and to remove the requirement that Matlab and Simulink are needed to run Strathfarm. The approach chosen here was the same as used in the continuous time model [54] beginning with a PI controller.

The wind farm controller used began from the one used in [54] in the s-domain form of:

$$\Delta P = k_p(P_d(t) - P(t)) + k_i \int (P_d(t) - P(t))dt \quad (5.1)$$

where P_d is the demanded power and $P(t)$ is the power output of the wind farm. This can be considered in the Laplace domain in the form:

$$\Delta P = (k_p + \frac{k_i}{s})P_s \quad (5.2)$$

where $P_s = P_d(t) - P(t)$. Equation (5.2) can be considered in discrete time, using the bilinear transform, as:

$$\Delta P = (k_p + k_i \frac{T_s}{2} \frac{1 + z^{-1}}{1 - z^{-1}})P_s \quad (5.3)$$

where T_s is the time-step.

$$\Delta P = \left(\frac{k_p(1 - z^{-1}) + k_i \frac{T_s}{2}(1 + z^{-1})}{1 - z^{-1}} \right) P_s \quad (5.4)$$

$$\Delta P = \left(\frac{k_p + k_i \frac{T}{2} + (k_i \frac{T_s}{2} - k_p)z^{-1}}{1 - z^{-1}} \right) P_s \quad (5.5)$$

$$\Delta P = \left(k_p + k_i \frac{T}{2} + \frac{k_i T_s z^{-1}}{1 - z^{-1}} \right) P_s \quad (5.6)$$

This can be considered as a difference equation of the form:

$$\Delta P[n + 1] = \Delta P[n] + (k_p + k_i \frac{T}{2})P_s[n + 1] - (k_p + k_i \frac{T}{2})P_s[n] + k_i P_s[n]T_s \quad (5.7)$$

$$\Delta P[n + 1] = \Delta P[n] + k_p(P_s[n + 1] - P_s[n]) + k_i \frac{T_s}{2}(P_s[n + 1] + P_s[n]) \quad (5.8)$$

We must consider the governing equations as difference equations so that they can be implemented within Strathfarm's code structure which requires these calculations to be in a DLL called in an s-function in Simulink.

After the power adjustment has been found at the wind farm level it is then distributed across the wind turbines.

The wind farm controller has been tuned for this discrete time implementation.

The tuning was done for a wind farm with ten turbines arranged in a row perpendicular to the direction of the wind to prevent wake effects.

The controller has been tuned in turbulence free wind fields at below near and above rated wind speeds of 9m/s, 11.5m/s and 14 m/s. As the wind farm controller has anti-windup, overshooting from the integral action is prevented. The tuned parameters for the WFC are $k_p = 1$ and $k_i = 4$

Following this step the WFC then allocates the level of curtailment to each turbine based on the flag status of each turbine. The initial distribution of allocating curtailment was proposed

by [54] as

$$\Delta P_i = \frac{\Delta P f_i}{\sum_{j=1}^{N_t} \hat{f}_j} \quad (5.9)$$

where ΔP_i is the requested curtailment of the i th turbine, ΔP is the total requested curtailment of the wind farm, f_i is the status flag of the i th turbine, \hat{f}_j is the status flag of the j th turbine and N_t is the number of wind turbines in the wind farm.

5.2.2 Limitations and Restrictions to be Considered

Black Limits

The black limits of the PAC are the absolute limit of where a wind turbine can operation while remaining in a safe operational state. The black limits are usually only reached in extreme operational events such as inertial and primary response provision as in these cases maximising the amount of power at that instant of time supersedes any other requirement to avoid a grid wide blackout. If a “soft” black limit is breached a wind turbine will need to enter a ‘recovery mode’ to return it to the normal operational strategy for the turbine. The PAC prevents the turbine operating outside of the outer ‘hard’ black limits as they are the absolute limit of where the turbine can safely operate.

Traffic Light Flags

In normal operation of a PAC fitted wind turbine the WFC should allocate power changes based on the traffic light flags which indicate how far away from the black limit a turbine is operating. The PAC restricts the amount of curtailment that a turbine with each flag colour can deliver.

First and Second Order Anti-Windup in Power Allocation

In practice there are limits to the operation of wind turbines in terms of how much curtailment can be delivered based on the operational status of the wind turbine. When a turbine is requested to go beyond these limits the power output will saturate resulting in the controller failing to accurately track the wind farm set-point. Saturation occurs in both the level of power curtailment which a wind turbine can deliver, which varies depending on its location on the torque speed plane, and also in the rate at which it can vary its level of curtailment.

The limits on the level of curtailment are calculated based on a series of “traffic light” flags which are conveyed from the PACs to the WFC and indicate how far each wind turbine has deviated from its default operational strategy with green flags being closest, amber slightly further and red the furthest away. The limits used in Strathfarm for the Supergen 5MW wind Turbine Model can be seen in Figure (5.4)

The lower bounds of power adjustment for the PAC tuning for the 5MW Supergen turbine ([108]) are:

1. -2.5MW for green flags,
2. -1.5MW for amber flags,
3. 0MW for red flags .

In addition to these limits, the PAC also has a limit on the rate of change of curtailment that a wind turbine can deliver. These limits however do not vary with the traffic light flags.

The solution to this saturation used in Strathfarm’s WFC is to first consider the saturation at the total windfarm wide level and then secondly consider each wind turbine’s ability to deliver a requested level of curtailment. This is done in the wind farm controller using the known rate limits in the change of power that the PACs will be able to implement. Initially, the WFC uses the status flags of all of the wind turbines and then finds the total upper and lower bounds of the windfarm for both the total level of curtailment and the rate of change of curtailment. Using these limits a back-calculation based approach to anti-windup can be added to the wind farm controller.

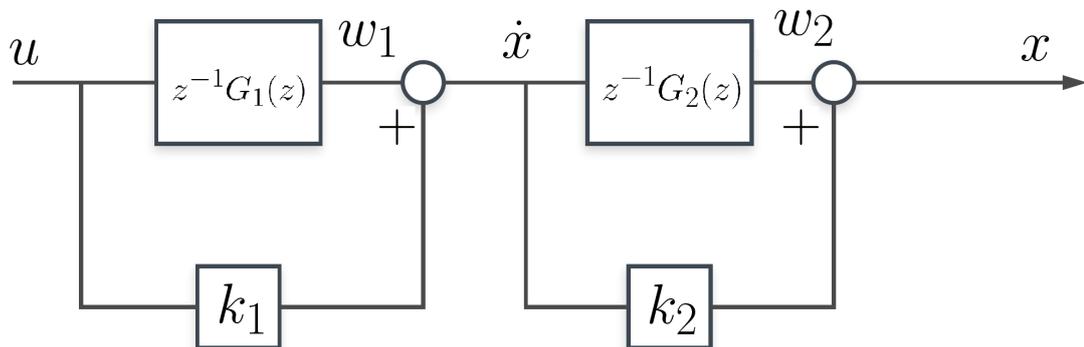


Figure 5.5: A generic form block diagram of the wind farm controller.

Figure (5.5) shows the generic form of the wind farm power controller. By considering a controller in this form, the limits of the power adjustment and rate of change of power adjustment for the wind farm are both calculated.

The controller first checks whether $\Delta P[n + 1]$ is saturated, if $\Delta P[n + 1]$ is saturated, then it is set to be equal to the saturation limit and $P_s[n + 1]$ is subsequently back calculated. The controller then checks if the rate (\dot{x}) is saturated, if \dot{x} is saturated $P_s[n + 1]$ back calculated and then both values are used to find an updated value for $\Delta P[n + 1]$.

5.2.3 Additional Practical Restrictions

Minimum Wind Speed

The PAC has a minimum operating wind speed of approximately 7 m/s below which it enters recovery or turns off. This does not have much of an impact on control strategies as firstly at such low wind speeds the turbine is normal operating in a red flagged state preventing the turbine's power adjustment.

Overtaking Wind Turbines

The PAC is capable of allowing the wind turbine to produce more than the rated power of the wind turbine. It achieves this in two different ways: Firstly in above rated wind speeds the PAC increase the torque of the turbine generator, while maintaining rated speed, to increase power output. Secondly, in below rated wind speeds the PAC extracts energy stored in the rotational inertia of the rotor resulting in a reduction in the turbine's rotational speed.

5.3 Development of a Wind Farm Control Strategy for Set-point Power Provision

To demonstrate why both the total wind farm curtailment and also the individual turbines curtailment both need anti-wind up measures, a series of simulation results are presented below with anti-windup measure being introduced incrementally to demonstrate why each of them is necessary. These simulations are all from the same wind farm model with the farm layout shown in Figure (5.6) at a mean wind speed of 11 m/s, a turbulence level of 10% and a farm-wide set-point of 35MW. The models turbines used are the Supergen exemplar 5MW wind turbine. The simulation results presented here are 800 seconds long with the wind farm controller active from 200 seconds into the simulation. This initial 200 second period is included as it allows the transients at the start of the model time to dissipate.

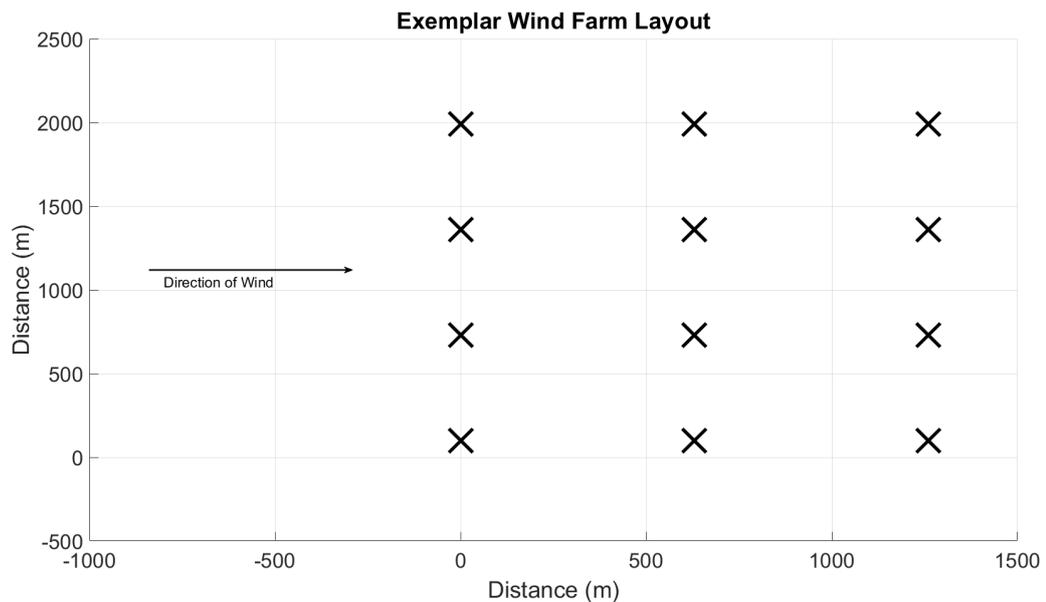


Figure 5.6: The layout of the wind farm used for the anti-windup examples presented in this chapter.

5.3.1 Traffic Light Flag Based Allocation Without Anti-Windup

As has been shown by [54] a traffic light based approach to curtailment where turbines are allocated reduction in power output based on turbine level traffic light flag information, can be

used to curtail a wind farm to a set-point power output.

When curtailing the farm only using the PI controller and allocation system without any anti-wind up leads to integral wind up in the power output with the curtailment allocated through Equation (5.9).

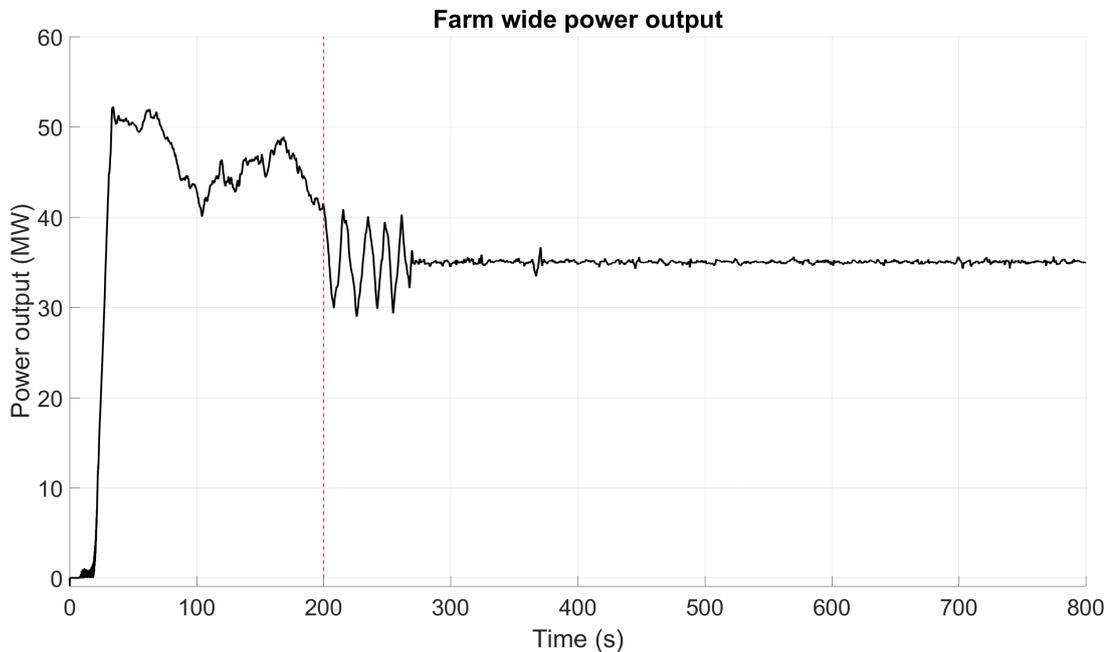


Figure 5.7: Example farm power output without any anti-windup with set-point control activated at the red dashed line at after 200 seconds.

Figure (5.7) shows the power output of the example wind farm without any anti-wind up measure included with the start of curtailed operation indicated by the red dashed line at 200 seconds. There is windup in the set-point tracking after the wind farm controller is turned on at 200 seconds. The windup is seen as the power output of the wind farm moves between the upper and lower PAC hard limits at the maximum rate of power change set by the PAC.

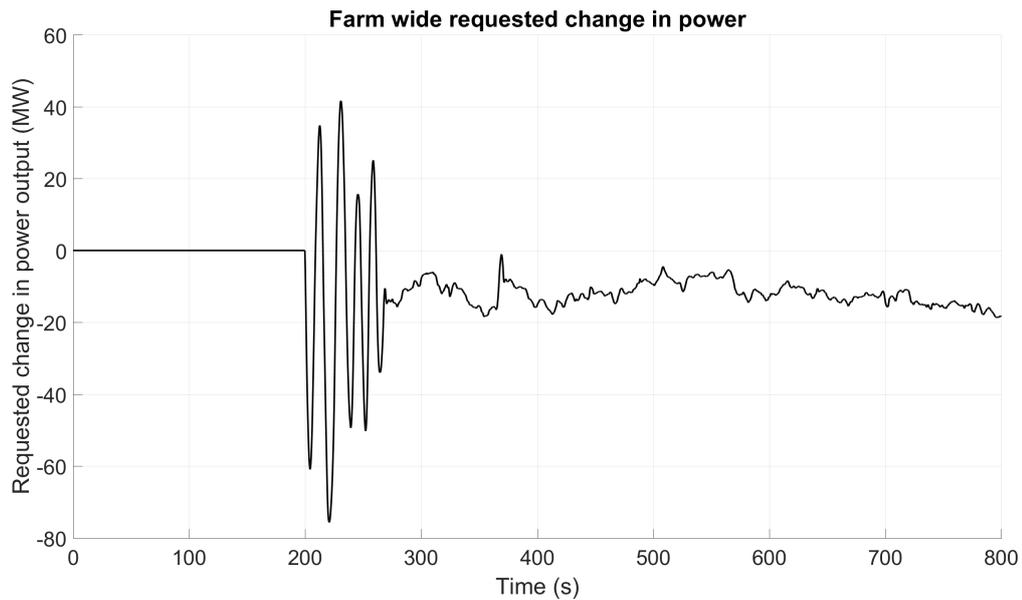


Figure 5.8: Example farm requested curtailment without any anti-windup.

The cause of the windup in Figure (5.7) can be seen in Figure (5.8) which shows the total requested reduction in power across the wind farm over time. After the controller is turned on there is a gigantic oscillation in the total curtailment. This amount of curtailment at times exceeds the maximum rating of the farm so obviously cannot be achieved.

5.3.2 Inclusion of First Order Anti-Windup for the Farm Wide Power

The first inclusion of anti-wind up considered here is to add anti-windup on the sum of the curtailment so that a level of curtailment beyond what the farm can provide is not requested. This information is known to the WFC as the maximum curtailment that a PAC fitted turbine can provide is governed by the traffic light flag system. By summing the limits of each turbine across the farm a farm wide range of possible curtailments that can be achieved is calculated.

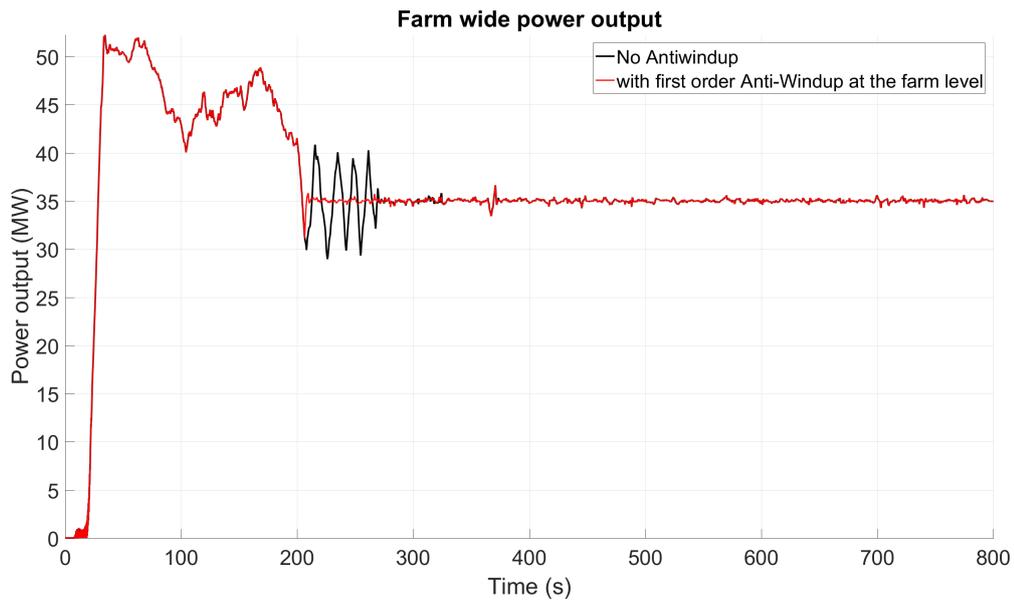


Figure 5.9: Example farm requested curtailment without any anti-windup compared to 1st order farm level anti-windup.

Figure (5.9) shows the power output when the farm-level first order anti-wind up is included compared with not including any anti-windup. It can be clearly seen that the inclusion of first order anti-windup at the farm level significantly reduces the oscillatory effects compared to when it is not included. However, this is still not yet satisfactory in terms of providing a set-point power output as there is a large overshoot just after 200 seconds as the controller activates. There is also still a noticeable deviation from the set-point between 300 and 400 seconds which is also undesirable.

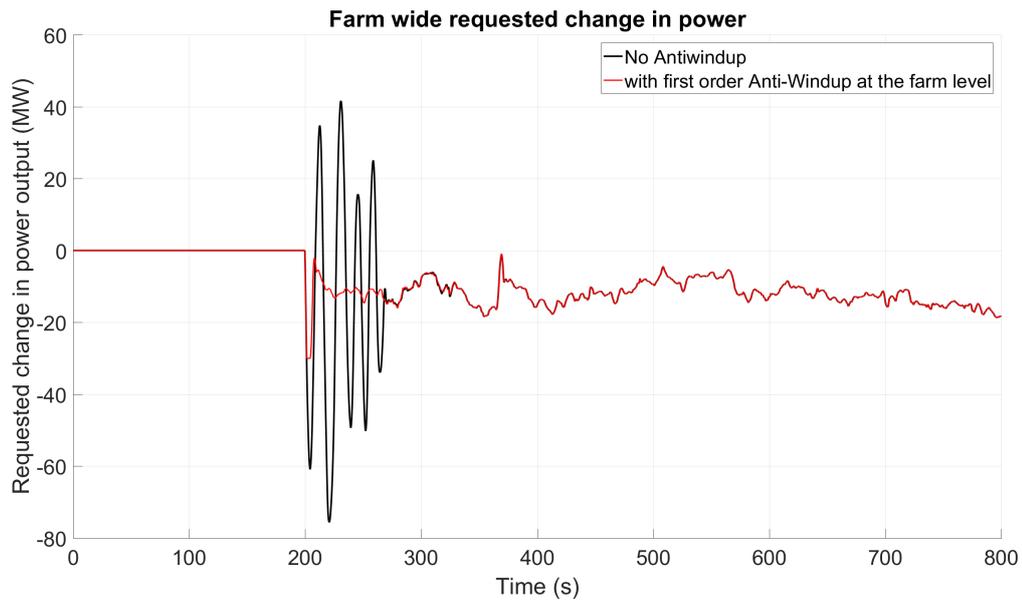


Figure 5.10: Example farm requested curtailment without any anti-wind up compared to 1st order farm level anti-windup.

Figure (5.10) shows the requested change in power output when the farm-level first order anti-windup is included compared with not including any anti-windup. It is clear that the initial overshoot in the power reduction is mitigated by the inclusion of the first order anti windup measures when the WFC is activated at 200 seconds. By comparing the times where there are deviations from the set-point in Figure (5.9) with the same times in Figure (5.10) it can be seen that they are due to the maximum rate of change of power, implemented by the PACs, being exceeded. Hence, further anti-windup measures to mitigate windup are required.

5.3.3 Inclusion of Second Order Anti-Windup for the Farm Wide Power

Second order anti-windup can be included at the wind farm level through the traffic light and recovery flags. If a turbine has either a green or amber flag it will be able to change its power output at a rate of 150Kw per second (assuming this does not cause the turbine to exceed a positional limit). If a turbine has either a red flag or a recovery flag active, it cannot change its power output. By taking the sum of turbines within the farm with green and amber flags, an upper and lower limit for how much power can change at a given time step can be known allowing for rate limits at the wind farm level. In the case of the Supergen 5MW exemplar

turbine PAC the limit would be $\pm 150\text{Kw/s}$ per turbine with either a green or amber flag.

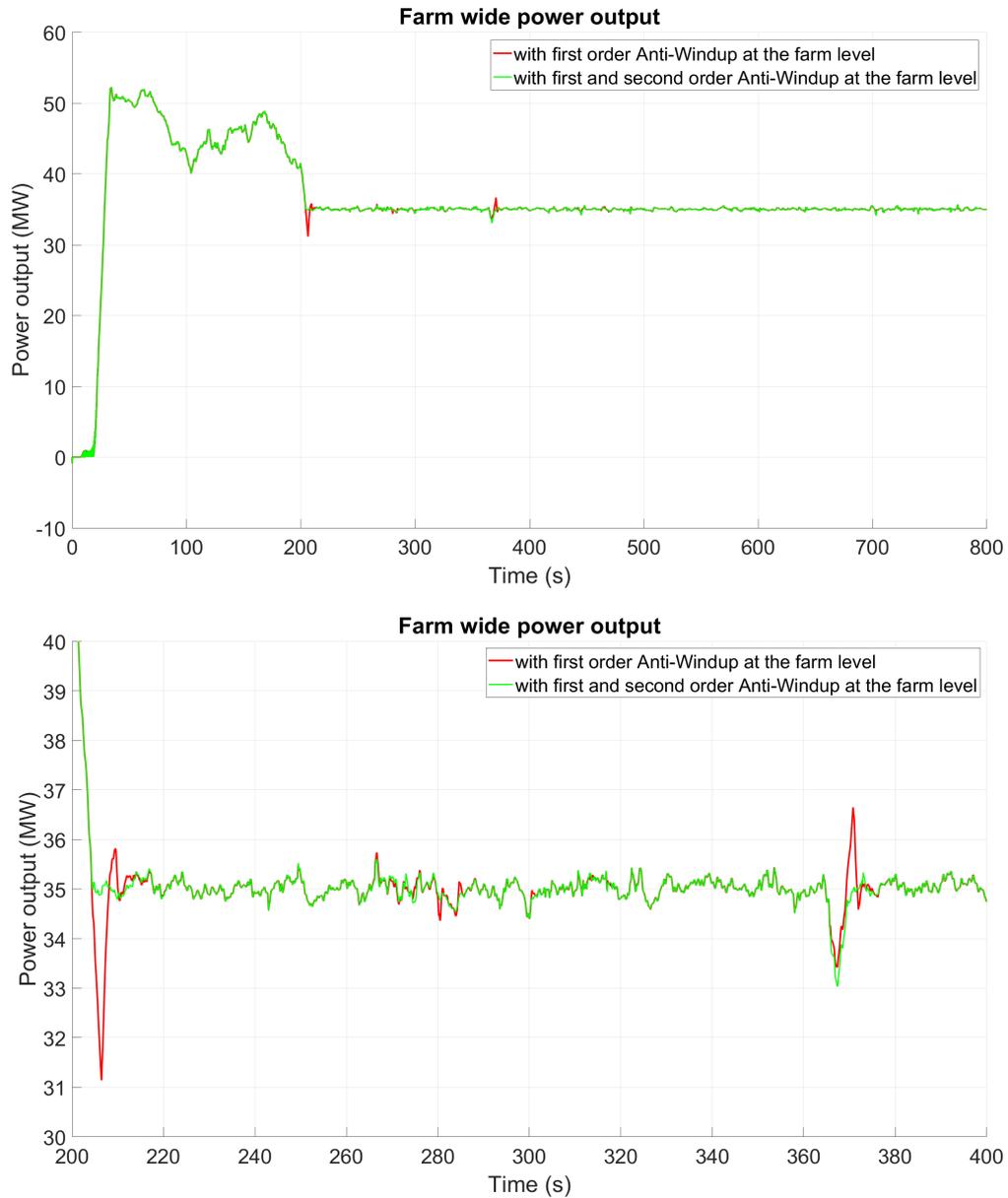


Figure 5.11: Example farm requested curtailment with only first order anti-windup compared to when first and second order farm level anti-windup are included.

Figure (5.11) shows the power output when the farm-level first and second order anti-windup are included compared with when only the farm-level first order anti-windup is included. It can be seen that when the farm-level first and second order anti-windup are included

the overshoot seen when only the farm-level first order anti-windup is included is no longer present. It can also be seen that there is a reduction (but not complete elimination) in the deviation from the set-point between 200 and 300 seconds when the farm-level first and second order anti-windup are included.

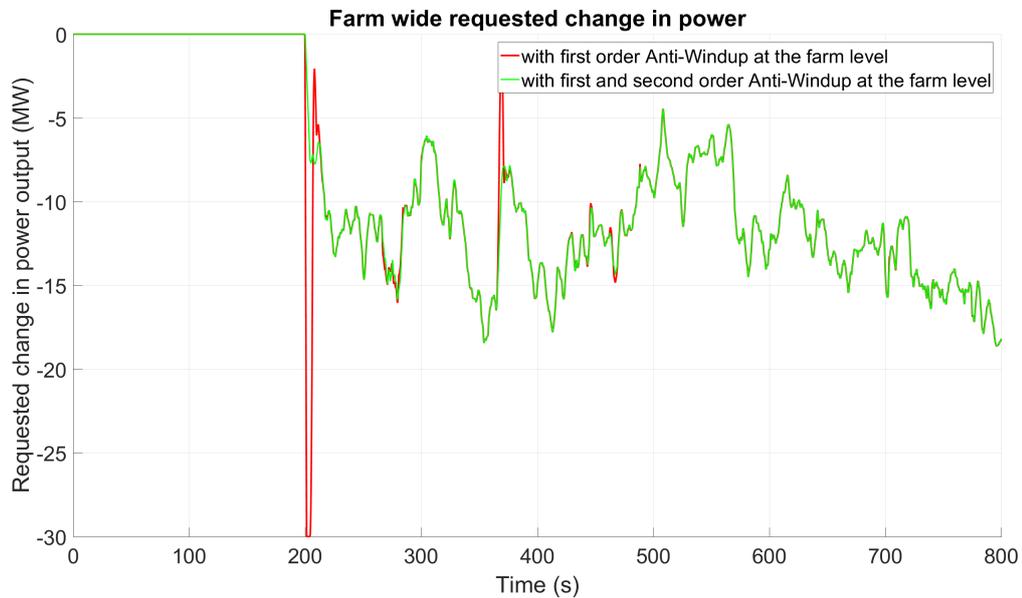


Figure 5.12: Example farm requested curtailment with only first order anti-windup compared to when first and second order farm level anti-windup are included.

Figure (5.12) shows the farm wide requested change in power output when the farm-level first and second order anti-windup are included compared with when only the farm-level first order anti-windup is included. The cause of the improvements in set-point tracking seen in Figure (5.11) can be seen to be the removal of the large changes in requested curtailment at the corresponding times in Figure (5.12).

5.3.4 Discussion of Turbine Level Windup in Curtailment Requests Despite Farm Level First and Second Order Anti-Windup

While the inclusion of anti-windup to the set-point control at the wind farm level is clearly successful in improving set-point tracking, there is room for further improvement by considering the limits of requested curtailment between the turbines in a wind farm, ensuring that no turbine is allocated a curtailment request its PAC will reject.

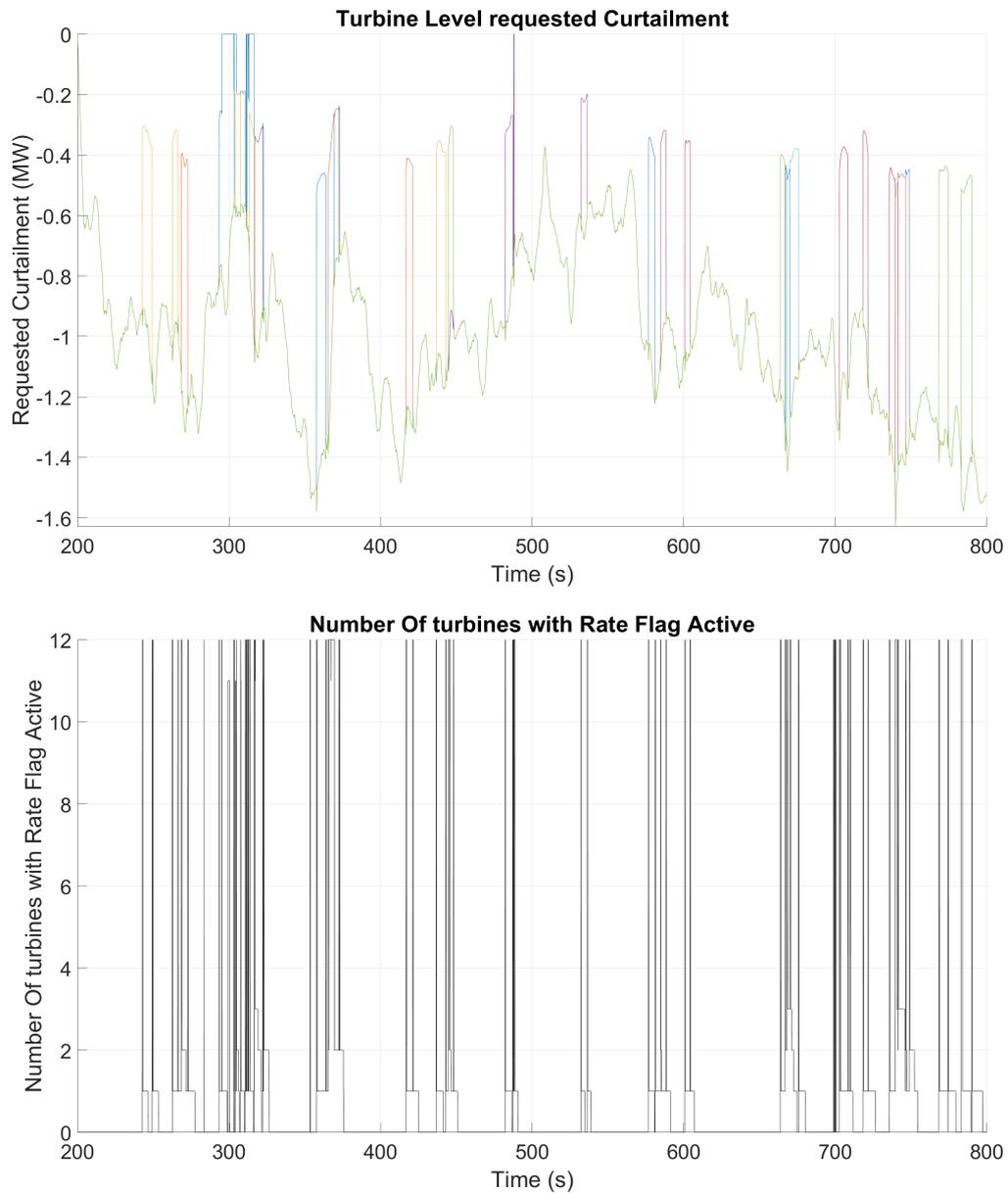


Figure 5.13: Example farm turbine level curtailment requests and the total number of activate rate flags.

Figure (5.13) shows the requested curtailment of each of the 12 turbines in the example wind farm for the simulation shown in Figure (5.11) and Figure (5.12) on the upper plot, whereas the lower plot shows the sum of turbines that have their power rate flag active meaning that the rate of change in requested curtailment for a turbine is changing at a faster rate than the turbine can deliver. What is noticeable in the lower plot in Figure (5.13) is that when there

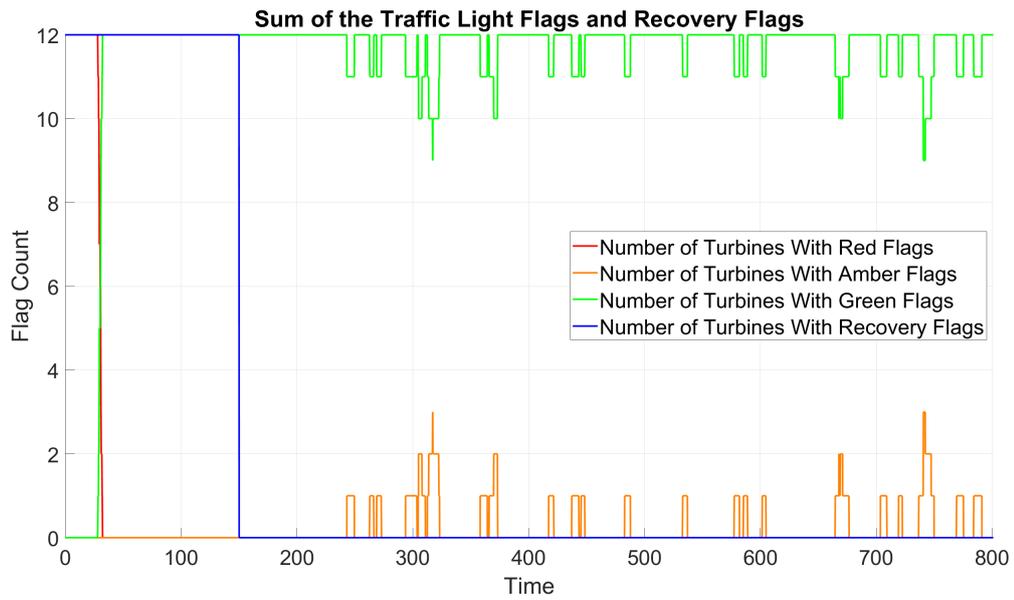


Figure 5.14: Example farm sum of each type of traffic light flag.

is a large reduction in the magnitude of curtailment in a single turbine all 12 turbine rate flags are activated.

Figure (5.14) shows the number of turbines in the farm with green, amber, red or Recovery flags active at a given time in the simulations presented in Figure (5.11) and Figure (5.12). Comparing Figure (5.14) and Figure (5.13) confirms that turbines that are not seeing a change in flag are also seeing an activation of their curtailment rate saturation flag.

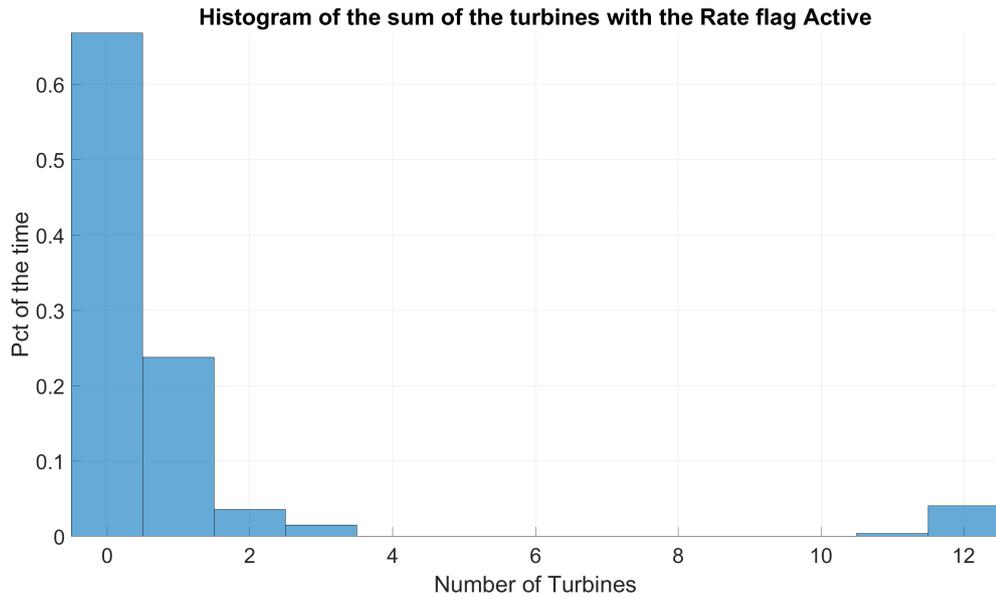


Figure 5.15: Histogram of the number of rate flags active in the farm.

Figure (5.15) shows a histogram of the number of turbines in the farm with an active rate flag for the results shown in Figure (5.13) . From this figure it is clear that when a single turbine changes between flags the reaction of the other turbines is resulting in all of the farm to experiencing a wind up effect as the turbines are allocated changes in power that they cannot react fast enough to deliver.

5.3.5 Inclusion of First and Second Order Anti-Windup for the Turbine Level Curtailment Requests

Through the inclusion of anti-windup considerations at the turbine level the windup effects seen in the previous subsection can be mitigated. The anti-windup considers the rate limits at both the farm and turbine level reallocating turbine level curtailment that cannot be fulfilled. In cases where a turbine is requested a rate of change in power adjustment it will either happen in a positive or negative direction. In the positive direction the WFC will know that the rate is being exceeded for a given turbine if:

$$\Delta P_i[n] - \Delta P_i[n - 1] > \dot{x}_{limit}. \quad (5.10)$$

In the negative direction the WFC will know that the rate is being exceeded for a given turbine if:

$$\Delta P_i[n] - \Delta P_i[n - 1] < -\dot{x}_{limit}. \quad (5.11)$$

in these cases the controller can limit the power change for the given time step to:

$$\Delta P_i[n] = \Delta P_i[n - 1] + \dot{x}_{limit} \quad (5.12)$$

in the positive direction and

$$\Delta P_i[n] = \Delta P_i[n - 1] - \dot{x}_{limit} \quad (5.13)$$

for the negative direction. Following this step the WFC then checks the power adjustment limits for the turbine to ensure that the requested power adjustment can be delivered. Finally the WFC repeats the steps in Equations (5.10) to (5.13) to ensure that the position limit anti-windup complies with the rate limits.

As the WFC adjusts the levels of curtailment through the farm it takes the cumulative sum of the adjustment across all the turbines. Once the WFC has passed through all of the turbines it does a second pass of the turbine power adjustments, reallocating curtailment to turbines which are not at a saturation limit (or are at the limit in the opposite direction).

The reallocation algorithm prioritises curtailment to turbines with green or amber flags ordered by estimated wind speed starting with the turbine with the fastest estimated wind speed as they are the most likely to be able to fulfil any change.

The reallocation algorithm can be summarized as the following steps:

1. Set "recurt"=0
2. Allocate wind turbines with curtailment from the wind farm control curtailment allocation strategy
3. Where turbines cannot fulfil their allocated curtailment flag them as being saturated in either the positive or negative direction, add the unfulfilled curtailment from each turbine to "recurt".

Chapter 5. Development of a Farmwide Power Set-Point Controller for Strathfarm

4. Pass through all of the unsaturated (and reverse saturated) wind turbines from the fastest estimated wind speed to the slowest, changing their requested power adjustment to the maximum value within their saturation limits until either all of the turbines are passed through or the total adjustment in the second pass is equal to "recurt".

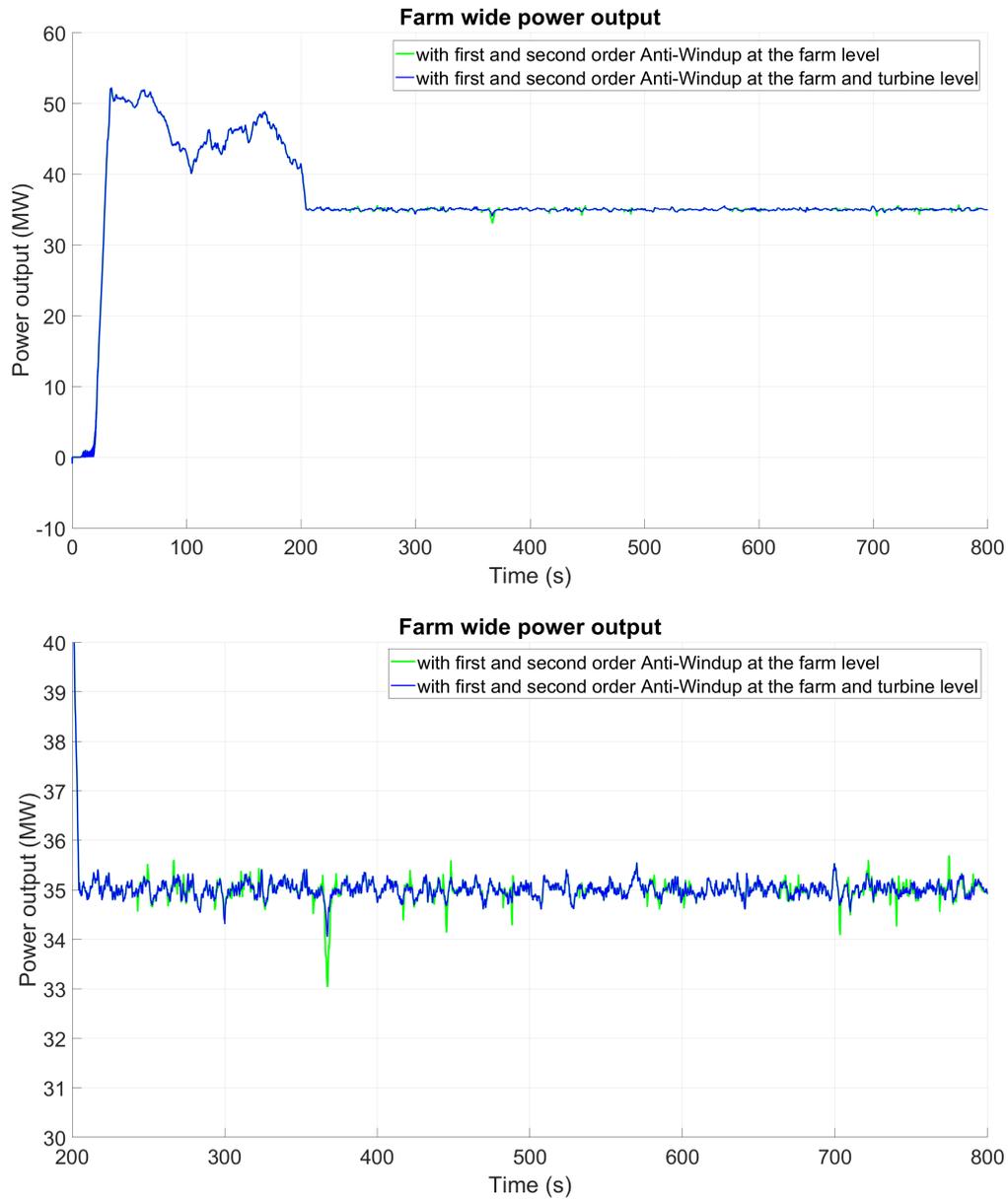


Figure 5.16: Example farm requested curtailment with anti windup applied at the farm level compared to when it is applied at both the farm and turbine level.

Chapter 5. Development of a Farmwide Power Set-Point Controller for Strathfarm

Figure (5.16) shows the power output when the farm-level and turbine level first and second order anti-windup are included compared with when only the farm-level first and second order anti-windup are included. It can be seen that the large deviation from the set point between 300 and 400 seconds is sufficiently mitigated when the turbine level anti-windup is included.

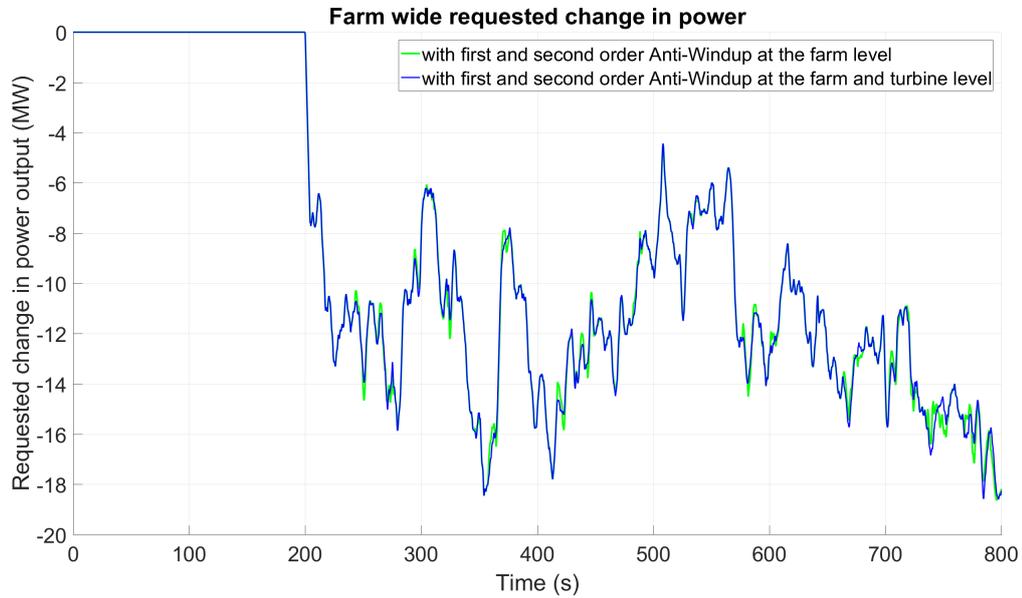


Figure 5.17: Example farm requested curtailment with anti windup applied at the farm level compared to when it is apply at both the farm and turbine level.

Figure (5.17) shows the requested change in power output when the farm-level and turbine level first and second order anti-windup are included compared with when only the farm-level first and second order anti-windup are included.

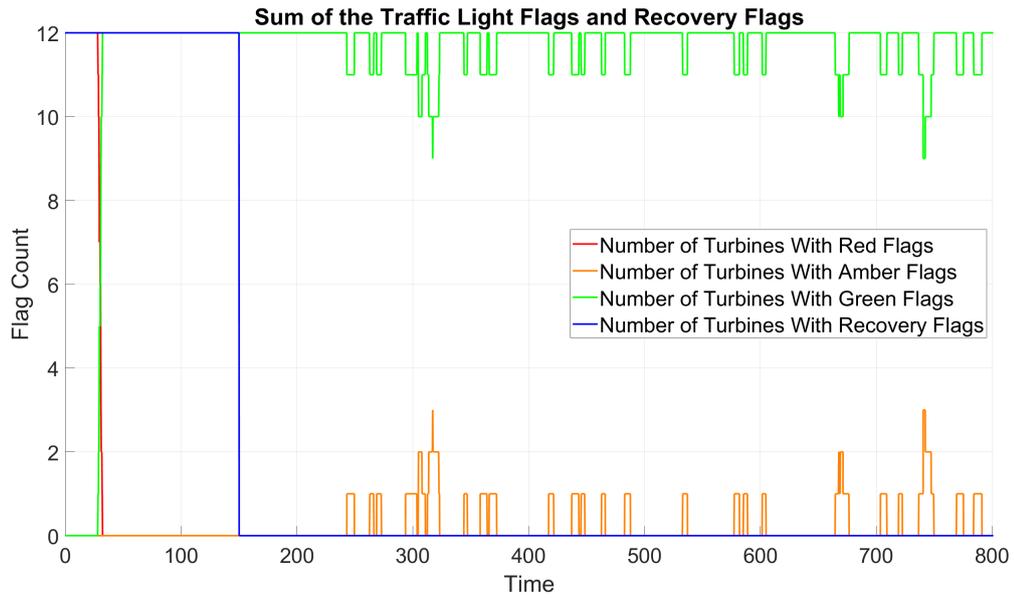


Figure 5.18: The sum of the traffic light flags when anti-windup is applied at both the farm and turbine level.

Figure (5.18) shows the number of turbines in the farm with Green, Amber, Red or Recovery flags active at a given time in the simulations presented in Figure (5.16) and Figure (5.17). The plot confirms that the reductions in deviations from the set-point is not simply because the changes in the traffic light flags have somehow been removed.

Anti-Windup Included:	No Anti-Windup	1st Order at Farm Level	2nd Order at Farm Level	Individual Turbine
error:	$1.0524 \cdot 10^6$	$0.2567 \cdot 10^6$	$0.1920 \cdot 10^6$	$0.1475 \cdot 10^6$

Table 5.1: Mean squared error of each of the levels of anti-windup shown in this chapter.

Table (5.1) shows the error in farmwide set-point power tracking that each level of anti-windup measures included in the wind farm controller results in for the simulations shown in this section. The error values shown have been calculated using a root mean square approach beginning at 205 seconds into each simulation to allow the controller reasonable time to reach the set-point accounting for rate limits. It can be seen from the table that each additional level of anti-windup included in the controller reduces the error in the set-point tracking demonstrating each addition's cumulative efficacy.

5.3.6 Additional Wind Farm Controller Considerations

Recovery Flags

When a wind turbine deviates too far from its operational strategy, to the extent that it is no longer safe, it is allocated a recovery flag. In this case the allocation of curtailment to that wind turbine is reduced to zero and the rest of the wind turbines compensate for it. The result of this implementation is that there is no noticeable change in the power output of the wind farm when a wind turbine enters recovery as can be seen by the results in this chapter.

Low Wind Speed Flag

When the wind speed in a PAC fitted wind turbine reaches a too low wind speed the turbine's power can no longer be adjusted. This is tracked in the PAC outputs through the low wind speed flag. Due to how the wind turbine operates this should normally occur after the activation of the amber and red traffic light flags so the turbine should not be curtailed when this happens, or if it is to only a minimal extent.

5.4 OVERRATING A WIND FARM

This section demonstrates Strathfarm's ability to maintain a set-point power output above the rated power of a wind farm in an above rated wind field. This is achieved through the PACs on each turbine which changes the requested torque in the wind turbine so that the turbine will produce the required power while maintaining the turbine's generator speed. In cases where more power is requested than is available from the wind the PAC will extract the energy from the energy stored as inertia in the rotor causing the wind turbine to slow down. It will do this until it reaches the black limit which causes the turbine to enter a period of recovery to speed the turbine back up to the baseline rotational wind speed. It should be noted that the results presented in this section should be considered illustrative as in practice turbines cannot be overrated for the durations of time shown. In a real wind turbine producing more power than a turbine's rated power results in the generator and power electronics overheating which leads to significant damage to those components. In practice overrating should be limited to emergency situations, such as inertial response provision. However, it has been shown [108]

that allowing wind turbines to produce a level of power above the rated limit for long durations of time could lead to significant reductions in tower and blade fatigue, so the ability of the PAC to overrate wind turbines in Strathfarm is discussed here.

For the results presented here the example wind farm layout used in Figure (5.6) is used again however the wind field is different. The wind field used here has been generated with a mean wind speed of 15 m/s and a turbulence intensity of 10%. As before, the wind farm controller is activated from 200 seconds into the simulation to allow for transients in the turbine models to dissipate. The farm wide power set-point has been set as 65MW.

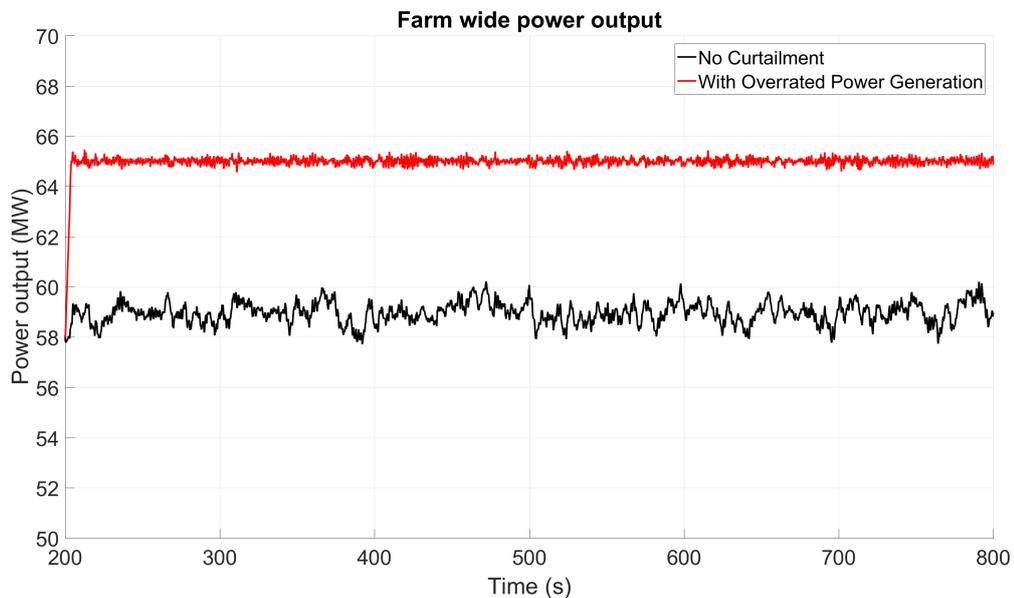


Figure 5.19: Example farm with an overrated level of power output set-point.

Figure (5.19) shows the total farmwide power output of the wind farm when there is no overrating of the wind turbines and when the farm has an overrated set-point of 65MW. It is clear from the graph that the set-point tracking for the farm is as effective for overrated set-points as it is for curtailed set-points as was shown in the previous section.

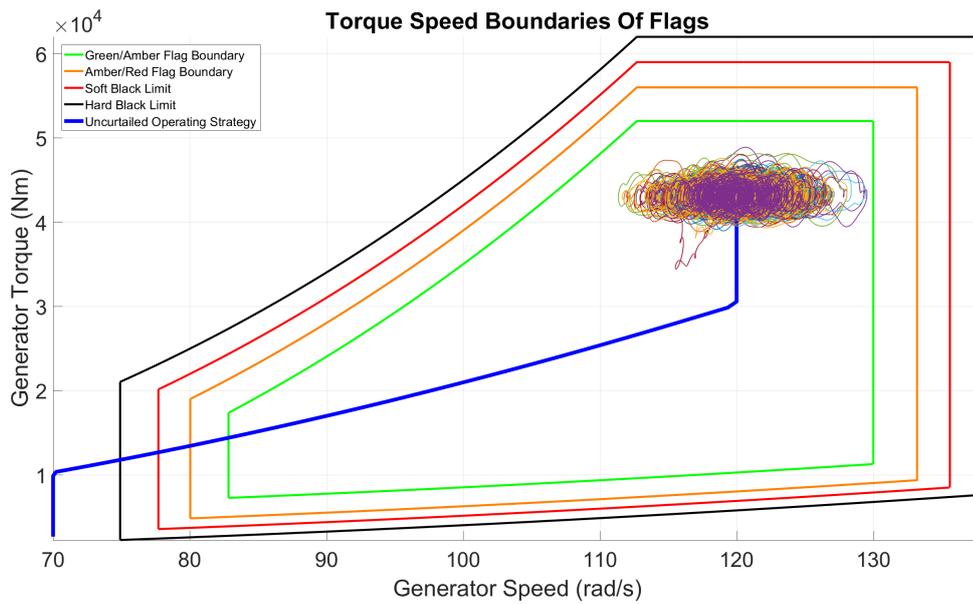


Figure 5.20: The Torque speed trajectories of the wind turbines in normal operation

Figure (5.20) shows the operation of each of the turbines in the wind farm when the turbines are not overrated on a torque speed plane.

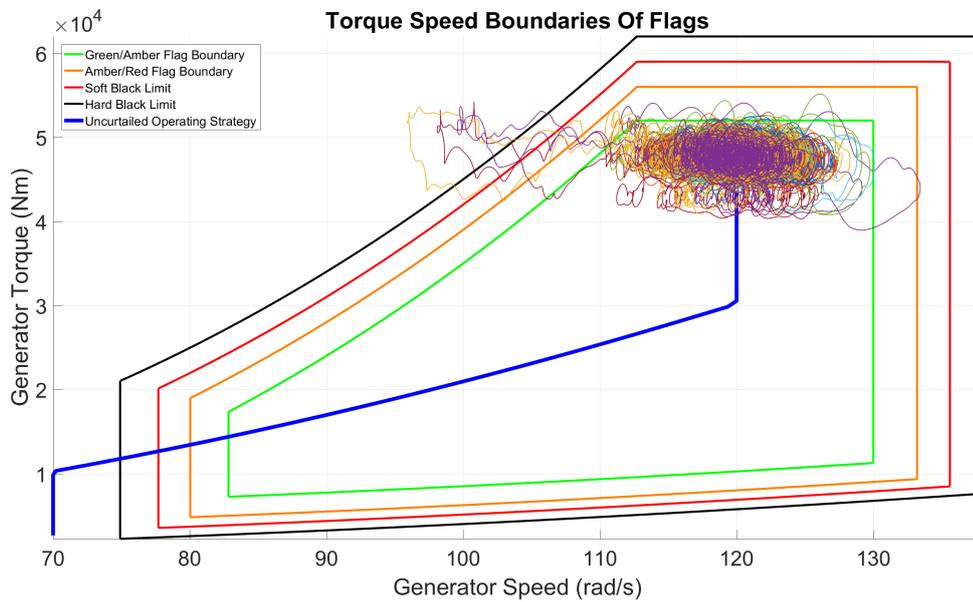


Figure 5.21: The Torque speed trajectories of the wind turbines in overrated operation. It should be noted that the black limits have been ignored in this simulation for demonstration purposes.

Chapter 5. Development of a Farmwide Power Set-Point Controller for Strathfarm

Figure (5.21) shows the operation of each of the turbines in the wind farm on a torque speed plane for the simulation where the wind farm is overrated. By comparing Figure (5.21) and Figure (5.20) it can be seen that the additional power output of the wind turbines when they are overrated is generated by increasing the generator torque. The stable power output seen in Figure (5.19) can be seen to be because the turbine are mostly operating within the green flag region, This is confirmed by Figure (5.22). It is worth noting that there are a few times that turbines significantly deviate from the green flag region with the trajectories exceeding the outer black region.

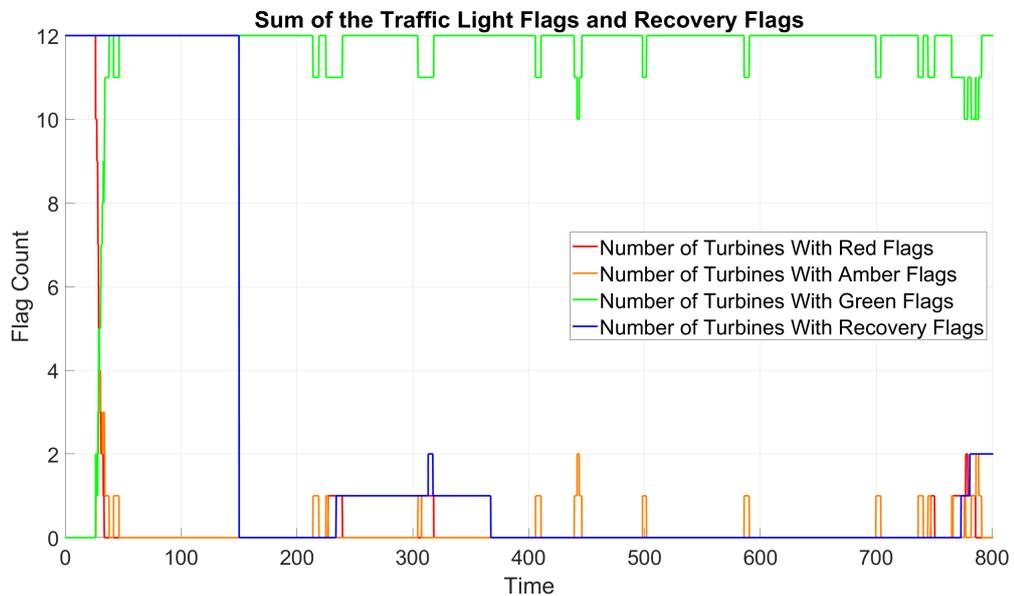


Figure 5.22: The sum of the traffic light flags in overrated operation

Figure (5.22) shows the number of turbines in the farm with green, amber, red or Recovery flags active at a given time when the wind farm generates an overrated set-point power as seen in Figure (5.19) .

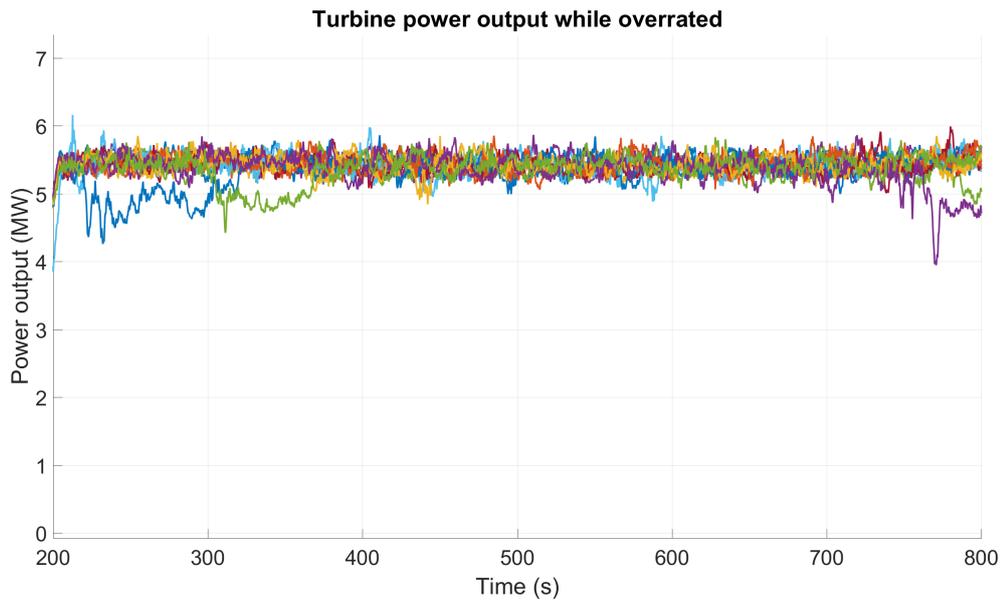


Figure 5.23: The power output of the overrated wind turbines in the example simulation.

Figure (5.23) shows the power output of each turbine in the wind farm when the wind farm is overrated. It is noticeable in the figure that there are a few turbines that are generating significantly less than rated power. This is the case for these turbines as they have been produced an overrated level of power while the wind speed was below rated, causing the PAC to extract the energy out of the rotational inertia of the turbine’s rotor. This has been done until the turbine exceeded the “soft” black limit boundary resulting in the turbines entering recovery. This can be confirmed by comparing the times where there are turbines in recovery in Figure (5.22) with these cases in Figure (5.23).

5.4.1 Overtating a Turbine With a Below Rated Wind Speed

By looking at the outputs for just Turbine 8 the mechanisms of how the PAC overrates a wind turbine at below rated wind speeds can be seen in more detail.

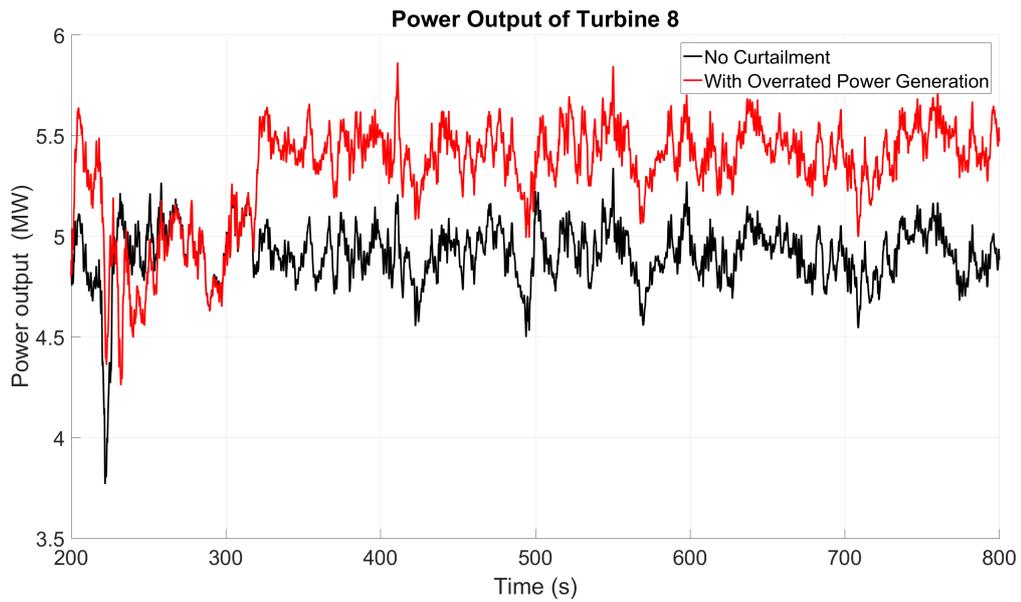


Figure 5.24: The power output of Turbine 8 in the example farm.

Figure (5.24) shows the power output of Turbine 8 for both the simulation where there is no wind farm controller active and the simulation where the wind farm is overrated. The overrating of turbine 8 begins just at 200 seconds. When the turbine enters recovery it must produce less than the rated power as the energy extracted from the rotational kinetic energy in the rotor must be replaced before normal operation can resume.

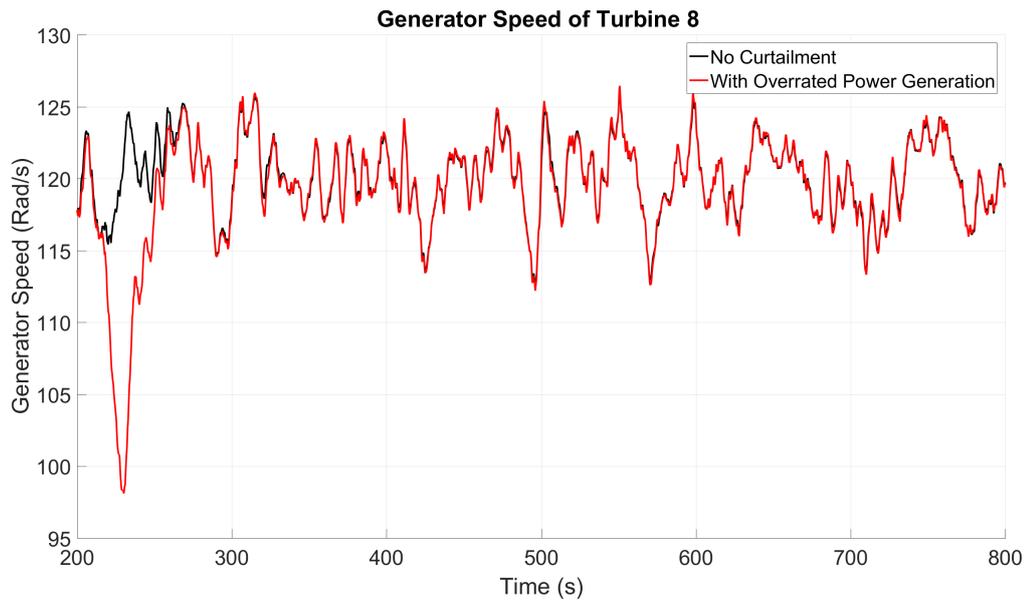


Figure 5.25: The generator speed of turbine 8 in the example farm.

Figure (5.25) shows the generator speeds of turbine 8 for both the simulation where there is not a wind farm controller active and the simulation where the wind farm is overrated. It can be seen that as discussed previously the overrating mechanism in the PAC (like the curtailment process) does not alter the operational speed of a wind turbine under normal conditions. The deviation in the generator speed is initially caused by the requirement to generate more power than is available for capture in the wind field causing the PAC to operate the wind turbine to extract the energy from the stored inertia in the rotor causing the generator to slow down. Following this, when the turbine enters recovery the generator speed increases until it is back at the speed it would have been at if the PAC was not operating, where it remains for the rest of the simulation.

Chapter 6

Designing a Wind Farm Controller to Reduce Wind Turbine Fatigue

This chapter begins by detailing how fatigue of mechanical components in a wind turbine can be estimated through time series bending moment data, through rainflow counting method and damage equivalent loads (DELs). It continues by discussing how wind turbines existing control strategies can result in fatigue of turbine components, due to the desire for maximal energy capture up to a rated wind speed. This chapter continues by discussing how turbine level wind speed estimation can be used in a wind farm controller to reduce damage equivalent loads. It concludes with a validation of the proposed wind farm control algorithms over a comprehensive range of wind speeds, turbulence intensities, wind directions and farm level curtailment power set-points. The distribution algorithms detailed in this chapter are novel and therefore given the significant change to lifetime fatigue are an important contribution to knowledge.

While it is possible that turbine level factors such as yaw and pitch misalignment are a significant cause of wind turbine fatigue they are not considered in this analysis. This is a possible area of future work but goes beyond the scope of this chapter and thesis.

6.1 Fatigue Estimation

This subsection contains a review of the relevant industrial standards, IEC 61400-1, IEC TS 61400-13 and DNV-DS-J102, to detail any beneficial information related to wind turbine loading

and fatigue.

Equivalent loads are discussed using the technical specification which summarizes them as a convenient, short-hand description of the fatigue impact of a given load measurement time history [19]. The equivalent load is described by the standards as “the single load amplitude, when applied for a total number of cycles in a given time history appearing at a given frequency, does the same damage as the sum of all the different rainflow-counted load amplitudes in the measured load spectrum”. This provides a single value for the amount of fatigue that a component has experienced regardless of operational condition, difference in wind speed or turbulence intensity.

The equivalent load is the “weighted average rainflow range, with the S-N curve slope, m , for the relevant material as the weighting exponent”. The process of rainflow counting and S-N curve look up are shown in Figure (6.1) from [32]:

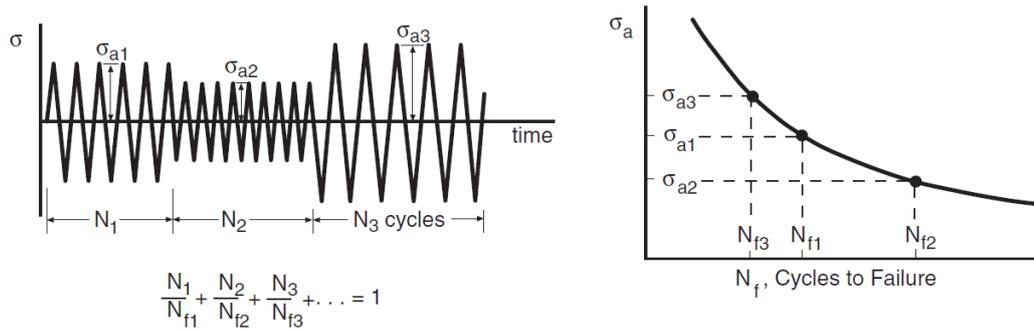


Figure 6.1: A diagram of the rainflow counting and S-N curve lookup process. [32]

The equivalent load is found using the equation:

$$R_{eq} = \left(\frac{\sum R_i^2 n_i}{n_{eq}} \right)^{\frac{1}{m}} \quad (6.1)$$

where R_{eq} is the equivalent load, R_i is the load of the i th class of the fatigue load spectrum, n_i is the number of cycles in the i th class of the fatigue load spectrum, n_{eq} is the equivalent number of cycles, and m the inverse slope of the S-N curve.

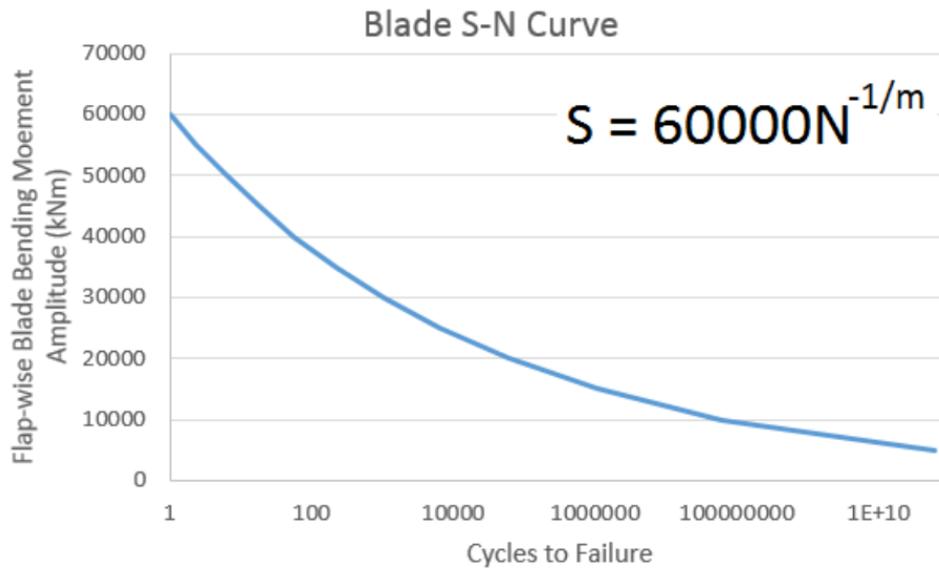


Figure 6.2: An example S-N Curve for a wind turbine blade.

Damage equivalent loads in this thesis have been calculated using an approach which decomposes a time series of bending moment data into a number of cycles and half cycles using rainflow counting [52]. These half cycles are then compared to a 1Hz frequency damage equivalent load baseline frequency. The amplitude of these cycles is weighted on an S-N curve in comparison with the 1 Hz baseline to find the damage equivalent loads for each component. It is worth noting that comparisons between Bladed and Strathfarm have differences in exact values of Damage Equivalent Loads. However, the changes in the damage equivalent loads in Strathfarm have been validated against Bladed, so the impact that wind turbine controllers and wind farm controllers have on the DELs of the turbines are accurate.

6.2 Existing Wind Turbine Bending Moments

Existing wind turbine control systems focus on operating the turbine for maximising energy capture at low wind speeds and maintaining rated power output at higher wind speeds. This section will detail how this turbine level mindset leads to higher levels of fatigue in a wind turbine's components. For the research shown in this chapter the 5MW Supergen wind turbine model has been used. This model assumes that the tower is a beam fixed at the base, bending in the fore-aft and side side directions [85]. This is an area where the modelling could be refined further in the future as with turbines being increasingly situated offshore the impact of the choice of support structure should be considered. However, there is no reason that the steady state modelling approaches used to develop the intelligent tower dispatch strategy later in this chapter can't be applied to other turbines.

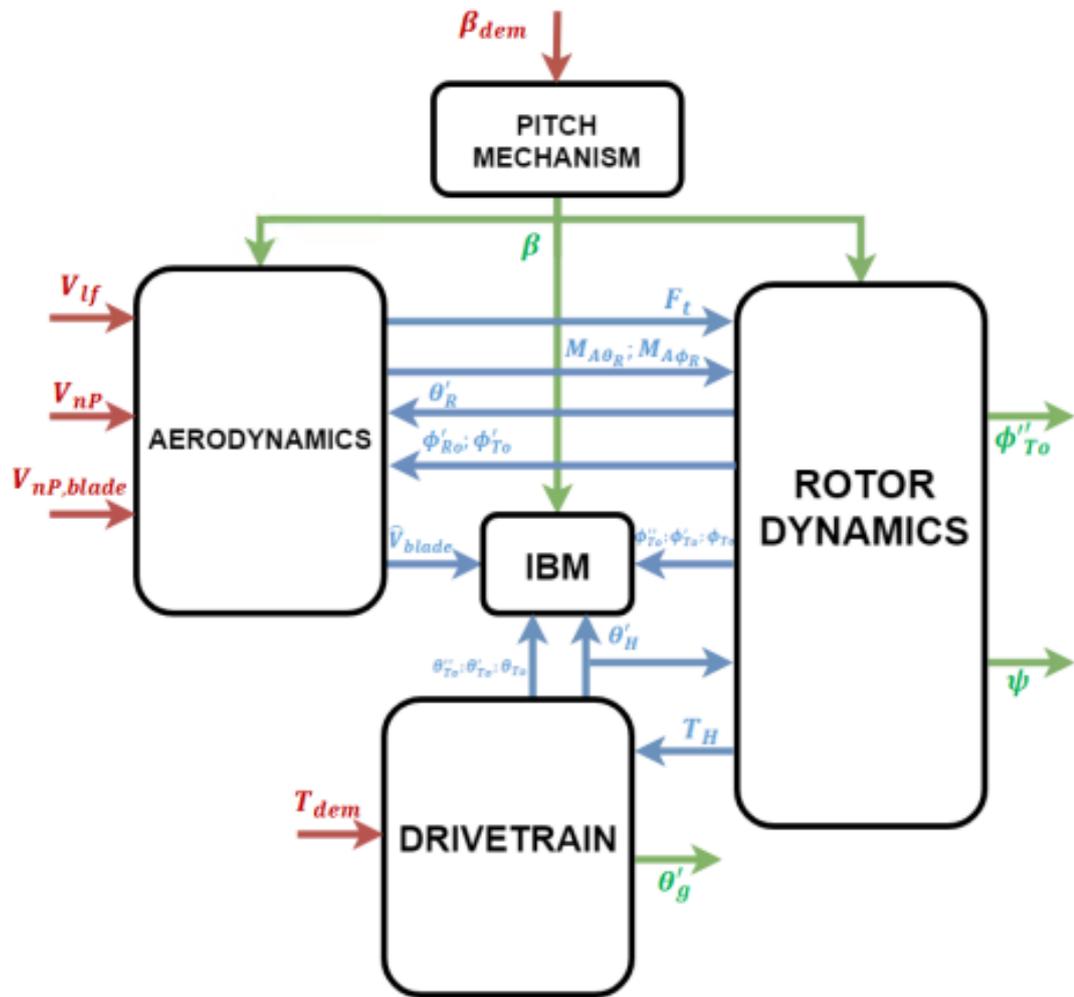


Figure 6.3: The interaction between different turbine components within the wind turbine model used in Strathfarm. From [11].

The wind turbine model used in Strathfarm includes the dynamics for the interaction of the different structural components within the wind turbine. This can be seen in Figure (6.3) with the model inputs given in red, the model outputs given in green and the component interactions given in blue. It should be noted that the tower dynamics are also included as they have been incorporated into the drivetrain and rotor dynamics.

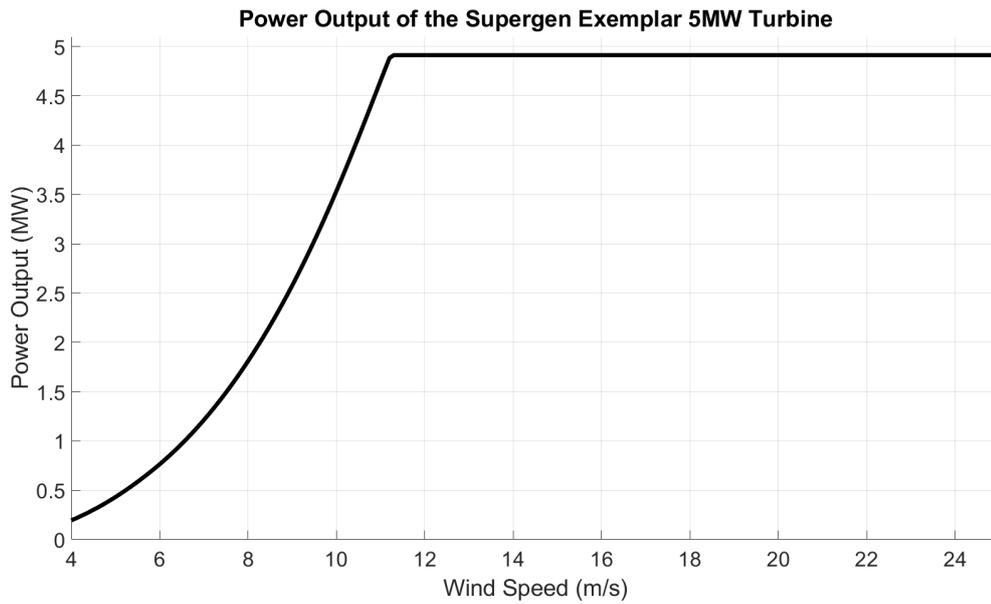


Figure 6.4: The power curve of the Supergen exemplar 5MW wind turbine over a range of wind speeds

Figure (6.4) shows the power output of the Supergen Exemplar 5MW wind turbine by wind speed.

When the rated power is reached the blades of the wind turbine are pitched to reduce the aerodynamic torque to maintain the power output at the rated power level.

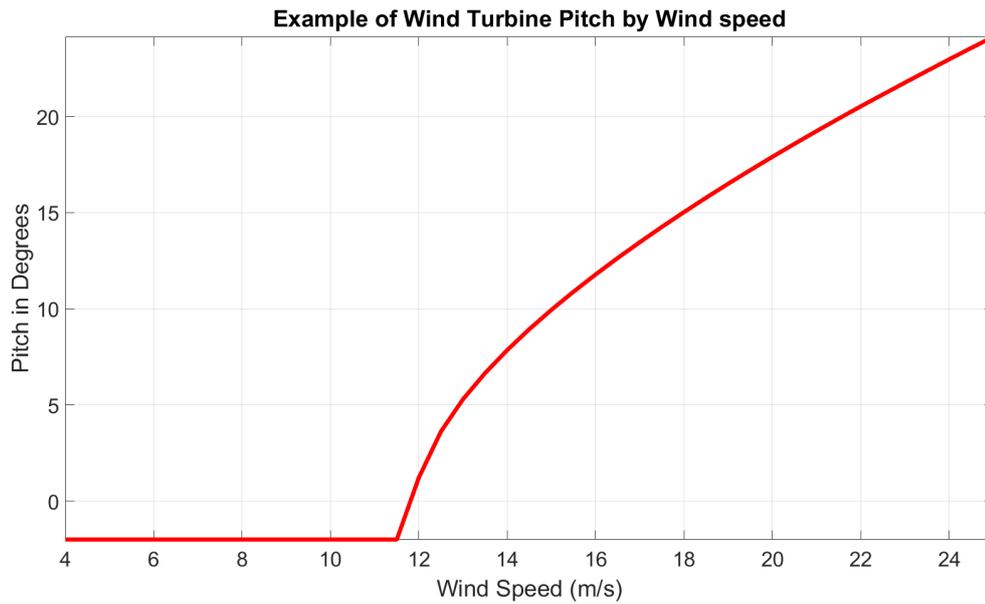


Figure 6.5: Pitch angle by wind speed from BLADED demo 5MW model.

Figure (6.5) shows an example of the pitching seen in a wind turbine by wind speed.

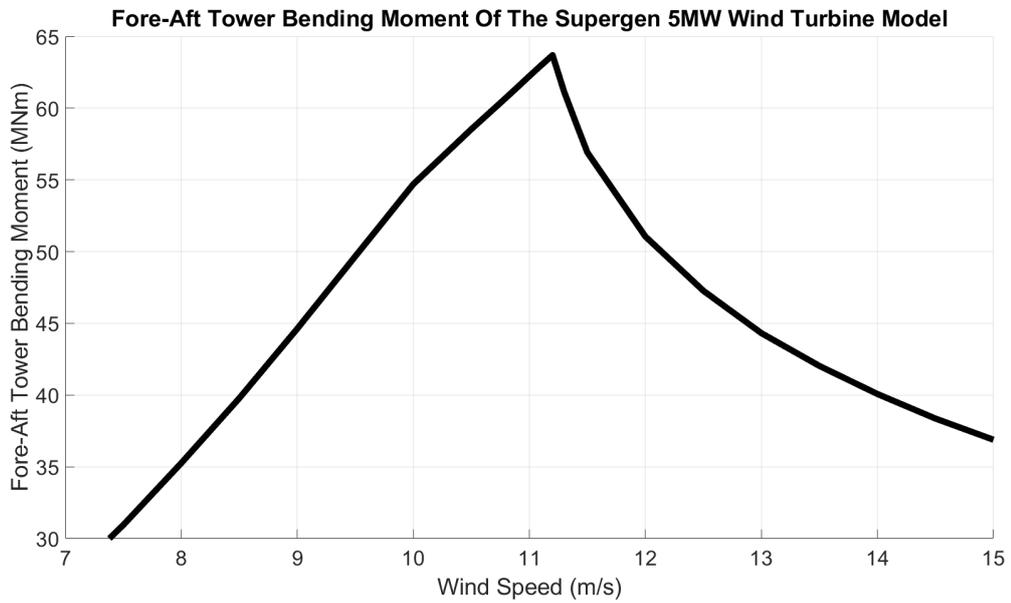


Figure 6.6: The fore-aft tower bending moment of the Supergen exemplar 5MW wind turbine over a range of wind speeds

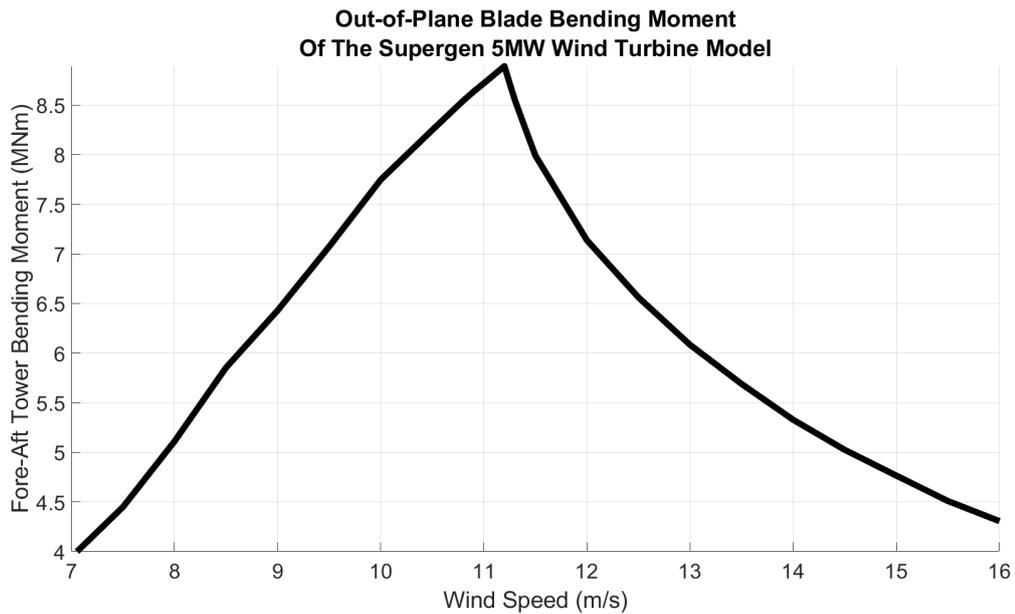


Figure 6.7: The out of plane blade bending moment of the Supergen exemplar 5MW wind turbine over a range of wind speeds

Figure (6.6) and Figure (6.7) show the bending moments of the Supergen exemplar 5MW wind turbine used in Strathfarm over a range of wind speeds. What can be seen in both profiles is that below rated the moments increase as the wind speed increases but above rated, when the blades are pitched to maintain rated power, there is a reduction in the moments as the wind speed increases. As this has shown the large impact which pitching the blades has on the moments of the structural components of the wind turbine, the obvious next step is to consider how curtailing the wind turbine using the PAC, which reduces power output through pitching, has on these moments.

6.2.1 Curtailment to Reduce Bending Moments

When a wind turbine operates normally the turbine sees a large variation in bending moments in the tower and the blades as the wind speed changes. This variation is the result of how a turbine's operational strategy is designed. In wind speeds below rated, the turbine's primary objective is to maximise power output. However, the operational strategy to maximize low wind speed energy capture results in the steady state bending moment profiles seen in Figure (6.8) and Figure (6.9) which show the variation in steady state fore-aft tower and out of plane blade bending moments for the Supergen 5MW wind turbine model as the wind speed changes. It can be seen that the bending moment increases until the turbine reaches the rated wind speed at 11.2 m/s, above which there is a large decrease in bending moment as the blades are pitched to maintain rated power. When the PAC is used to curtail the wind turbine by either the 200 or 400 kW, Figure (6.8) and Figure (6.9) show there is a decrease in the bending moments due to the increase in pitch angle used to reduce energy capture at all wind speeds.

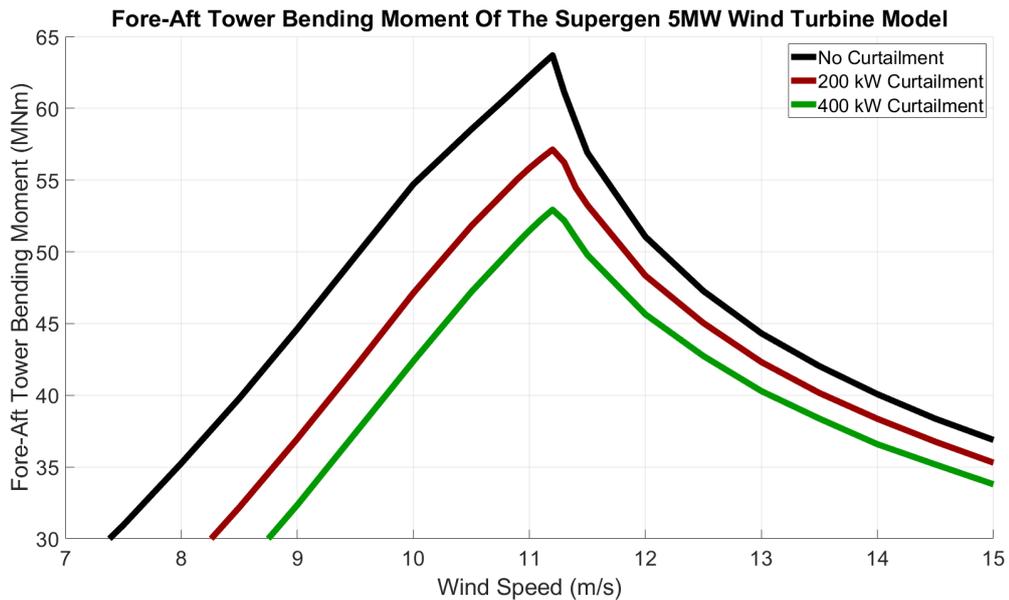


Figure 6.8: The fore-aft bending moment of the Supergen 5MW wind turbine tower by wind speed.

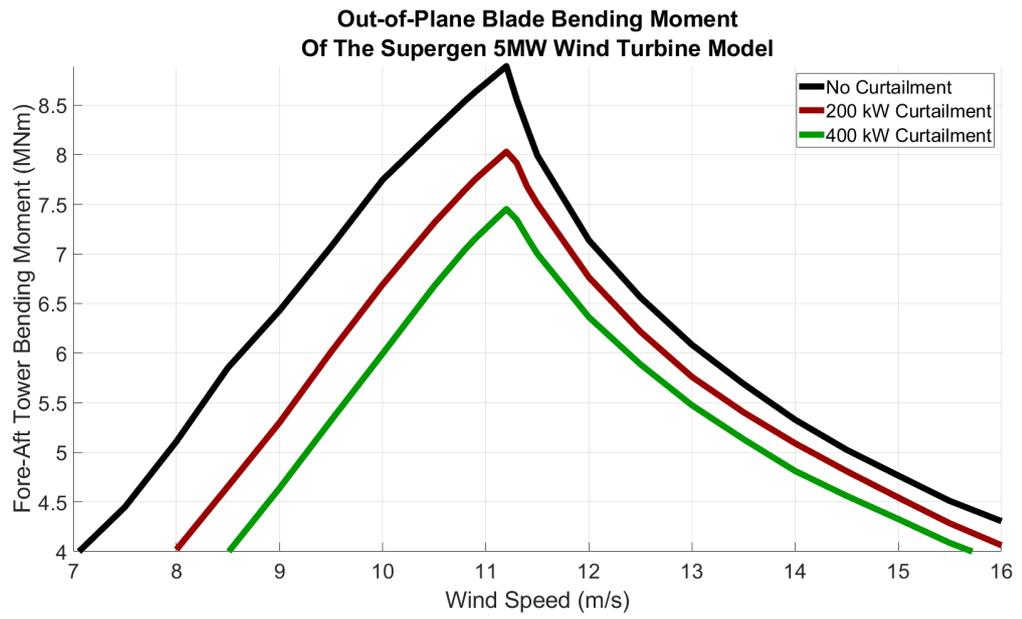


Figure 6.9: The out-of-plane bending moment of the Supergen 5MW wind turbine blade by wind speed.

6.3 Steady State Loads

This section of the chapter focuses on the steady state bending moments of the Supergen 5MW wind turbine model. Here steady state is defined as the wind speed being constant (i.e no turbulence). The 3D plots (Figure (6.10) and (6.10)) and the subsequent contour plots were generated by using a single turbine model in a constant wind field and very slowly curtailing the turbine through the PAC. This process was repeated over a range of wind speeds with a higher density of wind speeds used near the rated wind speed. These wind fields were at 0.5 m/s increments from 4m/s to 20m/s, around rated wind speed 0.1 m/s increments were used. In cases where the turbine was curtailed to the extent that it crossed a PAC limit the data is omitted due to its irrelevance. The data has been smoothed through an averaging process to show the steady state conditions more clearly.

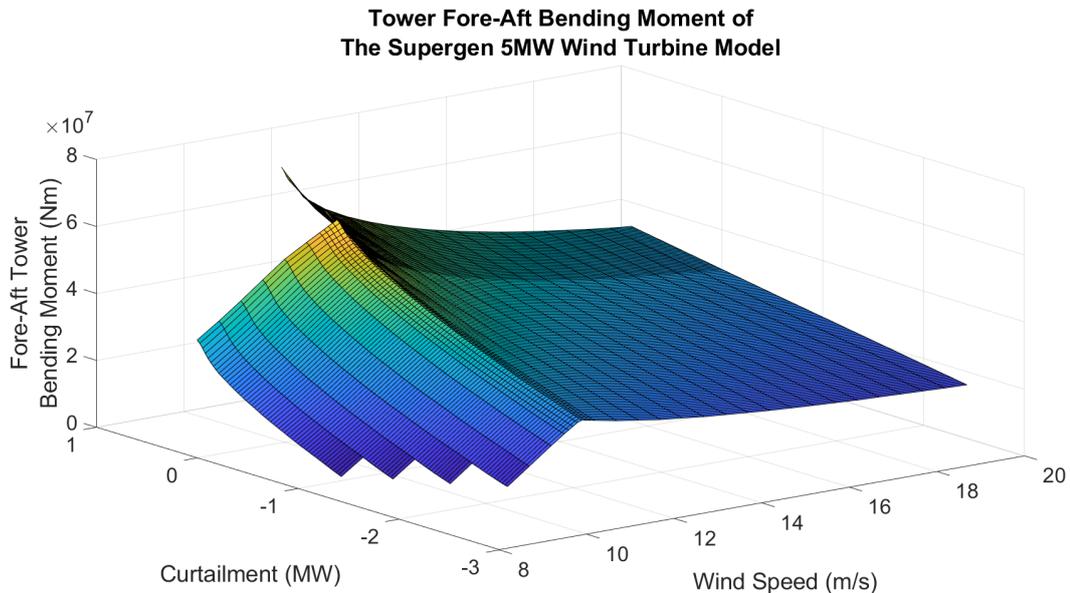


Figure 6.10: A 3D plot of the steady state fore-aft tower bending moment by wind speed and curtailment

Figure (6.10) shows the steady state tower fore-aft bending moment of the Supergen exemplar 5MW wind turbine over a range of wind speeds and curtailment levels.

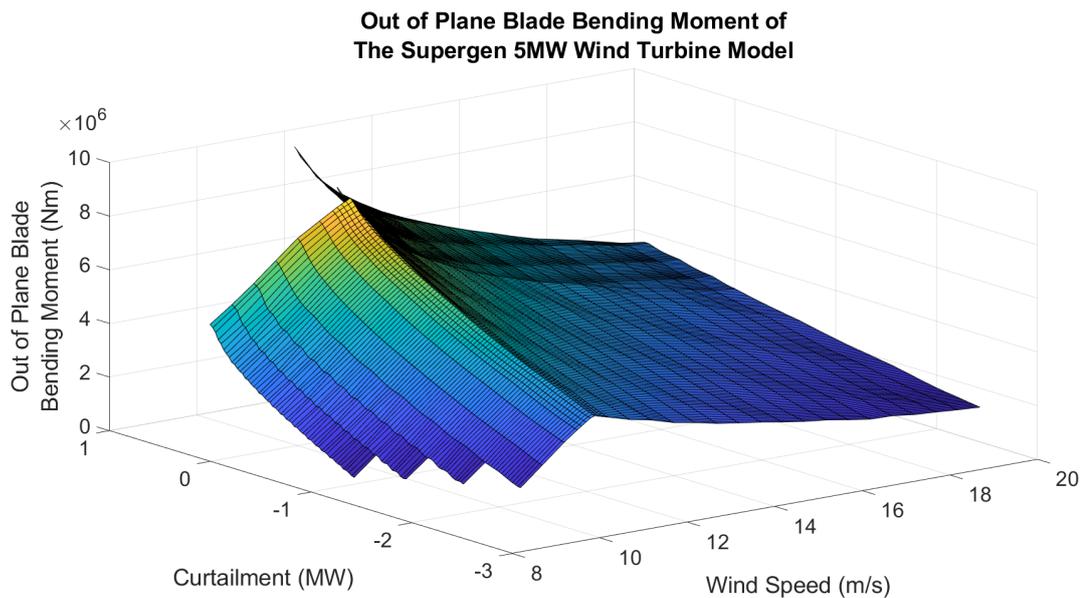


Figure 6.11: A 3D plot of the steady state out of plane bending moment by wind speed and curtailment

Figure (6.11) shows the steady state tower out of plane blade bending moment of the Supergen exemplar 5MW wind turbine over a range of wind speeds and curtailment levels.

It is worth noting the very similar shapes seen in Figure (6.10) and Figure (6.11) with a high peak in bending moment at the rated wind speed with zero curtailment and a general "Λ" shape that flattens out as the turbine's curtailment increases. It is also worth noting that there is not a large increase in the bending moments when the turbine is overrated at above rated wind speeds indicating that operating here in standard operation would not necessarily lead to component fatigue. What can also be seen is that as a turbine's pitch angle increases the reductions in steady state bending moment flatten out. This is the case above rated but also for when the PAC is used to curtail below rated as shown in Figures (6.10) and (6.11) which show the steady state load of the turbine's tower and blades with both level of curtailment and wind speed.

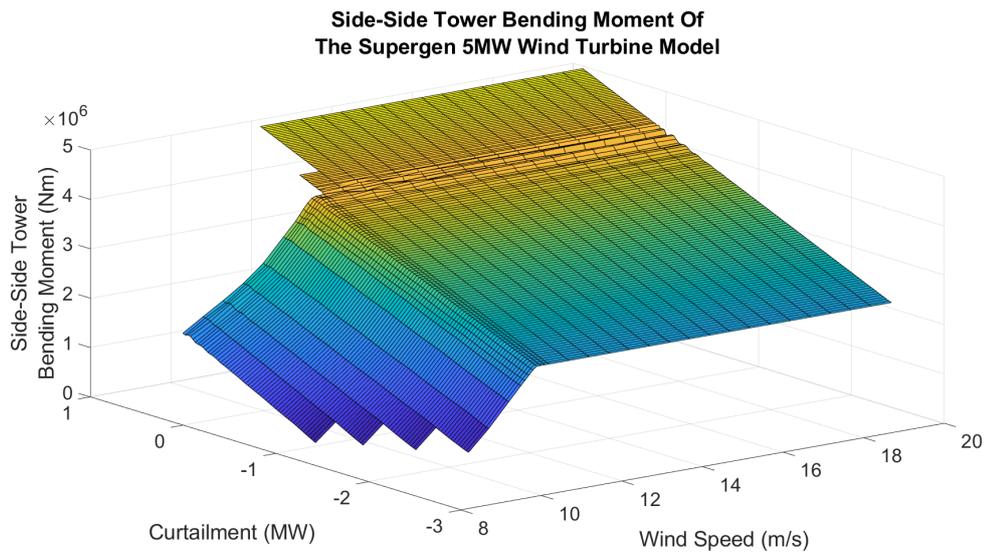


Figure 6.12: A 3D plot of the steady state side-side tower bending moment by wind speed and curtailment

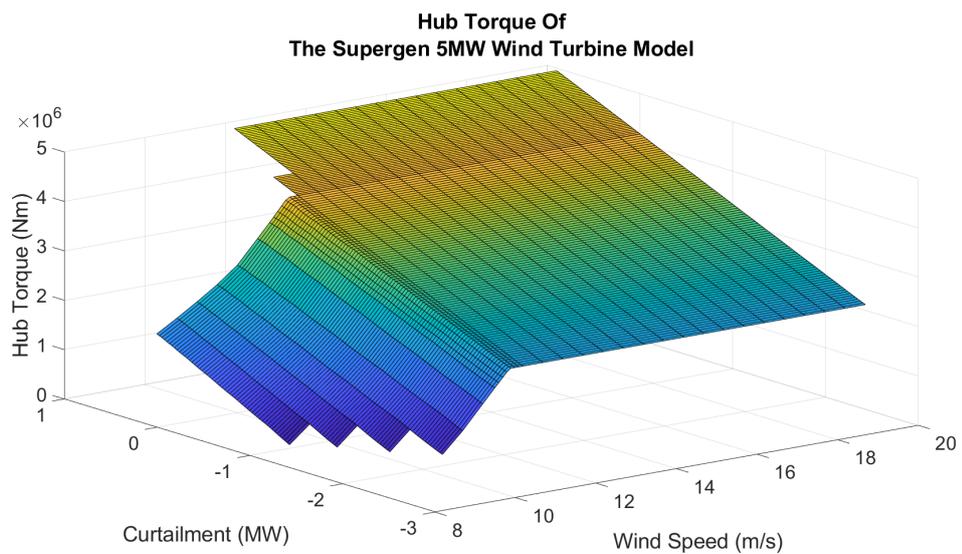


Figure 6.13: A 3D plot of the steady state hub torque moment by wind speed and curtailment

Figures (6.12) and (6.13) show the change in steady state tower side-side bending moment and hub torque as curtailment and wind speed vary respectively. It can be seen that they are almost identical, this is due to the side side bending moment being caused as a reaction torque to the hub torque through the drivetrain.

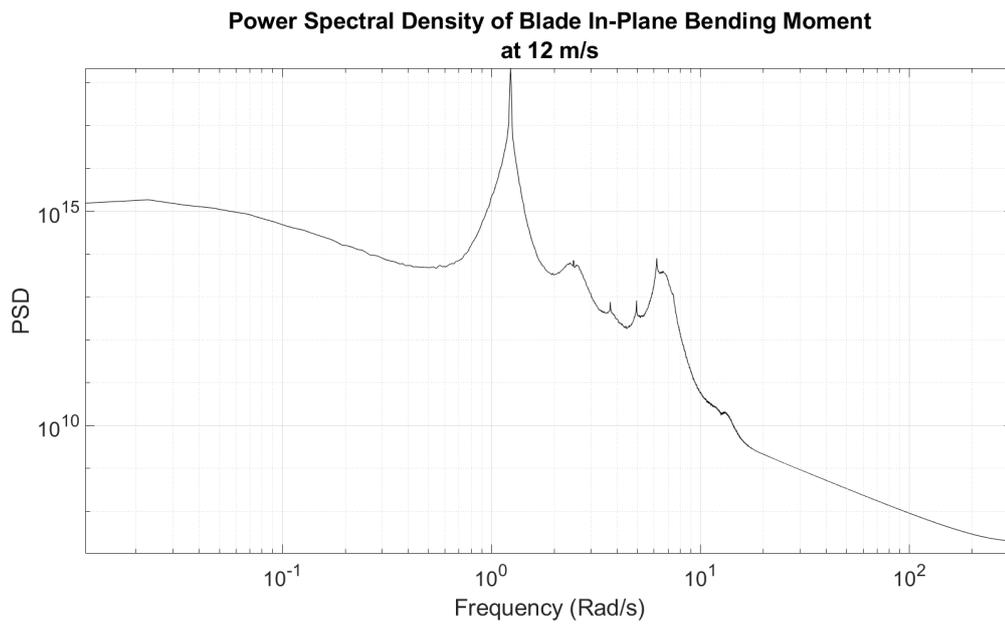


Figure 6.14: The Power spectral density of the in plane blade bending moment at 12 m/s

Figure (6.14) shows the power spectral density of the in plane blade bending moment at 12 m/s. It can be seen that the power spectral density (PSD) is dominated by the 1P rotational frequency of the turbine. As a result of this there is very little effect on the in plane blade DELs between any curtailment set-points or differences in control strategy.

6.4 The wind speed estimation from the PAC

The current wind farm controller in Strathfarm is the Strathclyde generic wind farm controller, so called since it can be used to implement almost all possible wind farm control strategies for all farms without requiring the turbine controllers to be modified except for a few gross parameters; that is, without compromising the existing turbine controller. It achieves this by only having the total power output of the wind farm fed back to the wind farm controller. If Individual turbine power or any other variable was fed back to the wind farm controller, it would in essence introduce additional feed back loops acting on individual turbines, with potentially damaging consequences for the turbine even to the extent of destabilising it. [100] However, the signal of the effective upstream wind speed does not have the same limitation as it has been calculated such that it does not depend on the turbine's power output or any other turbine variable [98]. Hence, the upstream wind speed estimate can be used for allocating curtailment within a wind farm.

The PAC's wind speed estimator uses the turbine's operation information to estimate the effective wind speed across the rotor, including induction lag, and other aerodynamic effects. This is done by reformulating BEM in a form local to the rotor disc, following this the dynamic inflow effects are included and then the equations are rearranged to give the effective wind speed estimate. [98]

6.5 Wind Farm Curtailment Allocation Strategies

This section presents three wind farm curtailment allocation strategies for distributing reductions in power across a wind farm while it is curtailed to a set-point power using the approach discussed in Chapter 5. The contours used in this section are the contours of the 3D plots: Figure (6.10) , Figure (6.11), Figure (6.12) and Figure (6.13).

6.5.1 Equal Curtailment

Equally curtailing the wind turbines allocates the total curtailment across the wind farm using the method proposed by [54] using only the traffic light flags from the turbine PACs for allocation. The curtailment request sent to each turbine in this strategy is given by:

$$\Delta P_i = \frac{\Delta P f_i}{\sum_{j=1}^N f_j} \quad (6.2)$$

where ΔP_i is the requested curtailment for the i th wind turbine, ΔP is the total farm level curtailment, f_i is the status flag of the i th wind turbine and $\sum_{j=1}^N f_j$ is the sum of all the status flags across the wind farm. The value of the status flag is determined from the traffic light flags sent from the PACs in a ratio of 3:1:0 for green, amber and red flags respectively.

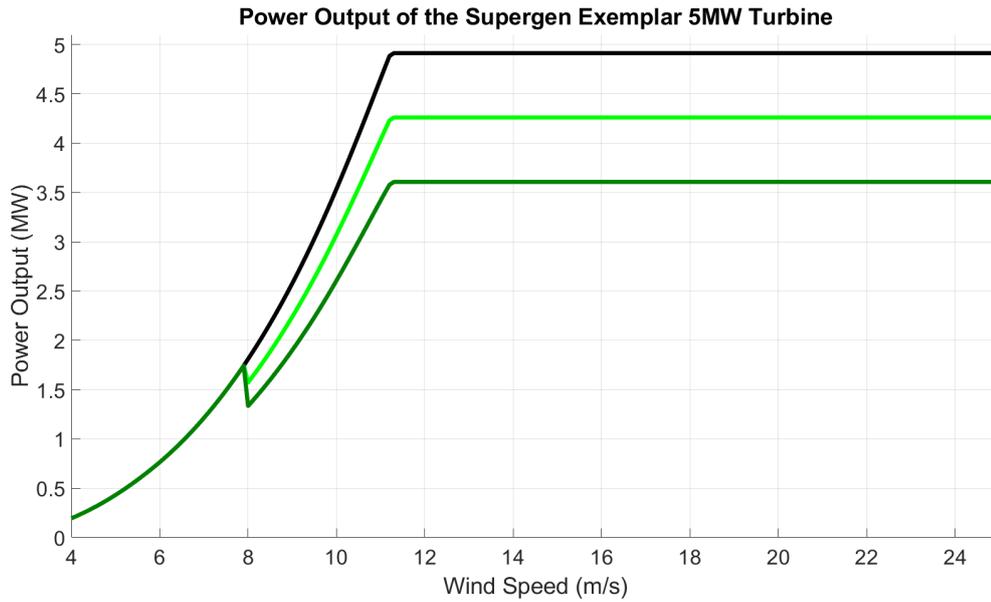


Figure 6.15: Power curve adjustments of the power estimation control strategy

6.5.2 Power Estimation Control

As one of the operational data streams from the PACs to the WFC is the upstream estimated wind speed this can be used as an input to the distribution algorithm in the wind farm distributed controller. This approach allocates the farm wide curtailment across the wind turbines in the ratio of the cubes of the estimated wind speed at each wind turbine. This was chosen as it is an approximation for the uncurtailed power of each turbine. The wind speed estimate can be used to approximate the power output of the wind turbines as at below rated wind speeds the power output of the wind turbine is approximately proportional to the cube of the wind speed. Hence, curtailment of the wind turbines can be distributed as:

$$\Delta P_i = \frac{\Delta P v_i^3}{\sum_{j=1}^N v_j^3} \quad (6.3)$$

$$v_i = \begin{cases} v_i & \text{if } v_i \leq 11.2 \\ 11.2 & \text{else} \end{cases} \quad (6.4)$$

where v_i is the estimated wind speed at the i th wind turbine.

An example of the power curves that result from this curtailment strategy can be seen in Figure (6.15).

Power Estimation Allocation Contour Plots

The impact of the Power Estimation allocation strategy on turbine components can be predicted by considering curtailment levels by wind speed superimposed over contour plots of each of the turbines major components.

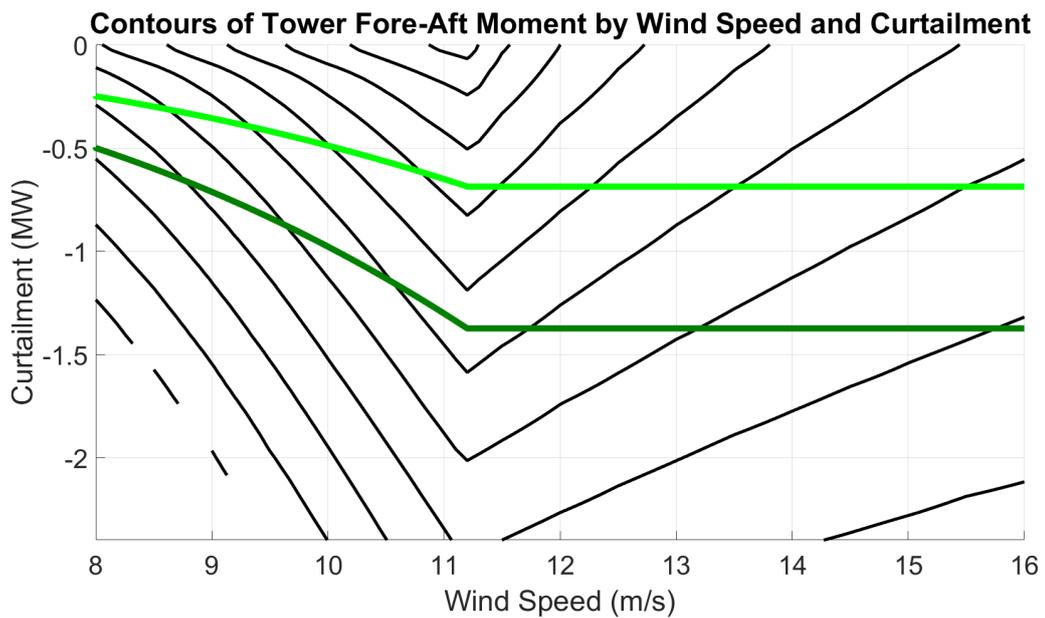


Figure 6.16: A contour plot of the steady state fore-aft bending moment of the tower with the power estimation control strategy at different curtailment levels.

Figure (6.16) shows a contour plot of the steady state fore-aft bending moment of the tower with the power estimation control strategy at different curtailment levels for the Supergen 5MW exemplar wind turbine. It can be seen that the strategy does cross contour lines at below rated wind speed but it is more aligned with them than the uncurtailed or equally curtailed strategies so should see lower levels of fatigue. It can also be seen that while the above rated curtailment is the same for all above rated wind speeds the curtailment means that the Power Estimation Allocation strategy crosses fewer contours than the uncurtailed strategy so the Power Estimation Allocation strategy will see lower levels of above rated fatigue compared to not curtailing.

It is also worth noting that this strategy avoids the peak of the fore-aft tower bending mo-

ment curve which should also lead to a reduction in fatigue.

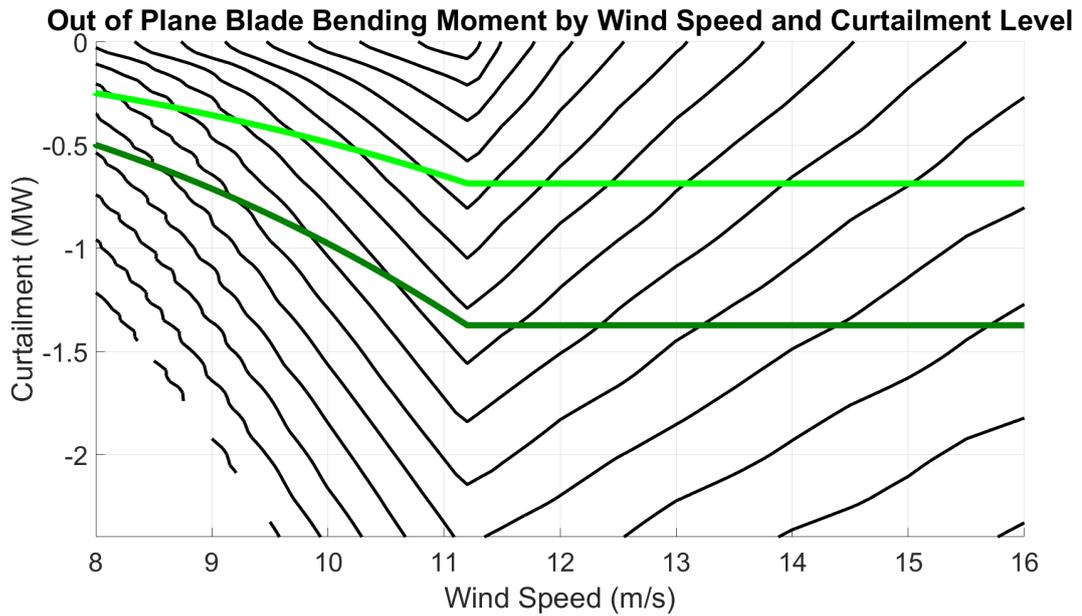


Figure 6.17: A contour plot of the steady state out of plane bending moment of the blade with the power estimation control strategy at different curtailment levels.

Figure (6.17) shows a contour plot of the steady state out of plane bending moment of the blade with the power estimation control strategy at different curtailment levels for the Supergen 5MW exemplar wind turbine. It can be seen that the strategy does cross contour lines at below rated wind speed it is more aligned with them than the uncurtailed or equally curtailed strategies so should see lower levels of fatigue. It can also be seen that at above rated wind speeds the Power Estimation Allocation strategy crosses fewer contours than the uncurtailed strategy but not to the same extent seen for the fore-aft bending moment.

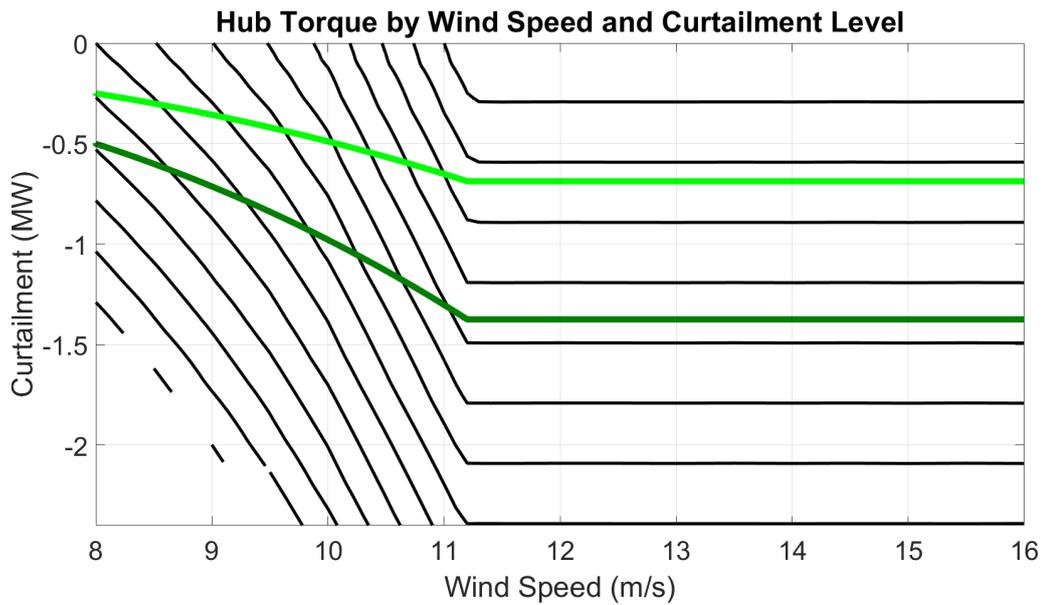


Figure 6.18: A contour plot of the steady state hub torques with the power estimation control strategy at different curtailment levels.

Figure (6.18) shows the hub torque contour plot of the steady state hub torques with the power estimation control strategy at different curtailment levels for the Supergen 5MW exemplar wind turbine. It can be seen that unlike in Figure (6.16) and Figure (6.17) this allocation strategy does not cross any contour lines above rated however this might not lead to a significant reduction in fatigue as the uncurtailed operation of the turbine also does not cross any contours. It is also worth noting that while this strategy does cross contour lines at below rated wind speeds it crosses fewer of them than not curtailing so should result in lower levels of fatigue at these wind speeds by comparison.

6.5.3 Intelligent Tower Control

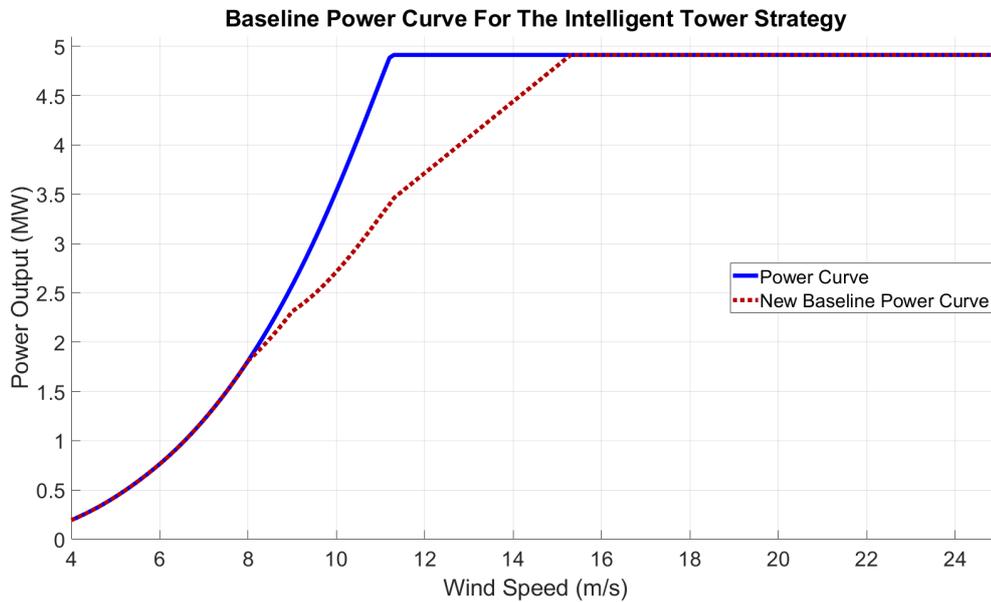


Figure 6.19: The “New Baseline” Power curve for the Intelligent Tower Control strategy

For this strategy the controller distributes curtailment by allocating a set amount of curtailment to each wind turbine based on its estimated wind speed so that the tower bending moment is maintained. Once this has been done, any remaining curtailment is then distributed across the wind farm using the traffic light approach. Where turbines are requested to achieve an unachievable level of curtailment the anti-windup mechanism redistributes any curtailment which would cause a turbine to experience windup. This reallocation is done by sorting the wind turbines by estimated wind speed from highest to lowest and requesting additional changes in power of each turbine up to a saturation limit until the farm level curtailment will be achieved. As a result of this set-point power is maintained. This strategy preallocates each wind turbine

with a level of curtailment:

$$\Delta P_{NB_i} = \begin{cases} 0 & \text{if } v \leq 8 \\ (v - 8) \times -28000 & \text{if } 8 \leq v \leq 9 \\ -28000 + (v - 9) \times -58400 & \text{if } 9 \leq v \leq 11.2 \\ (-15.3 + v) \times 382000 & \text{if } 11.2 \leq v \leq 15.3 \\ (-15.3 + v) \times 270000 & \text{if } 15.3 \leq v \leq 19 \\ 100000 & \text{if } 19 \leq v \end{cases} \quad (6.5)$$

The “New Baseline” power curve resulting from the curtailment in Equation (6.5) can be seen in Figure (6.19).

The turbines are allocated curtailment using this preallocation:

$$\Delta P_i = \Delta P_{NB_i} + \frac{\Delta P_{Farm} - \sum_{j=1}^N \Delta P_{NB_j}}{N} \quad (6.6)$$

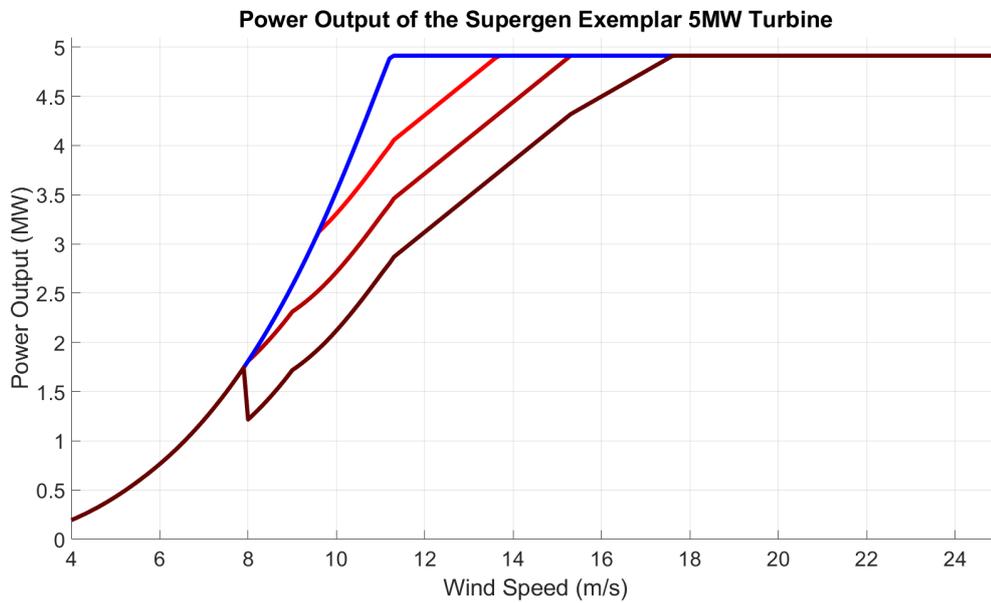


Figure 6.20: Power curve adjustments of the intelligent tower control strategy

Chapter 6. Designing a Wind Farm Controller to Reduce Wind Turbine Fatigue

An example of the power curve with the Intelligent Tower Control curtailment distribution can be seen in Figure (6.20).

It should be noted that any preallocation values are turbine specific so the numbers presented here would not be effective for another wind turbine.

Intelligent Tower Control Contour Plots

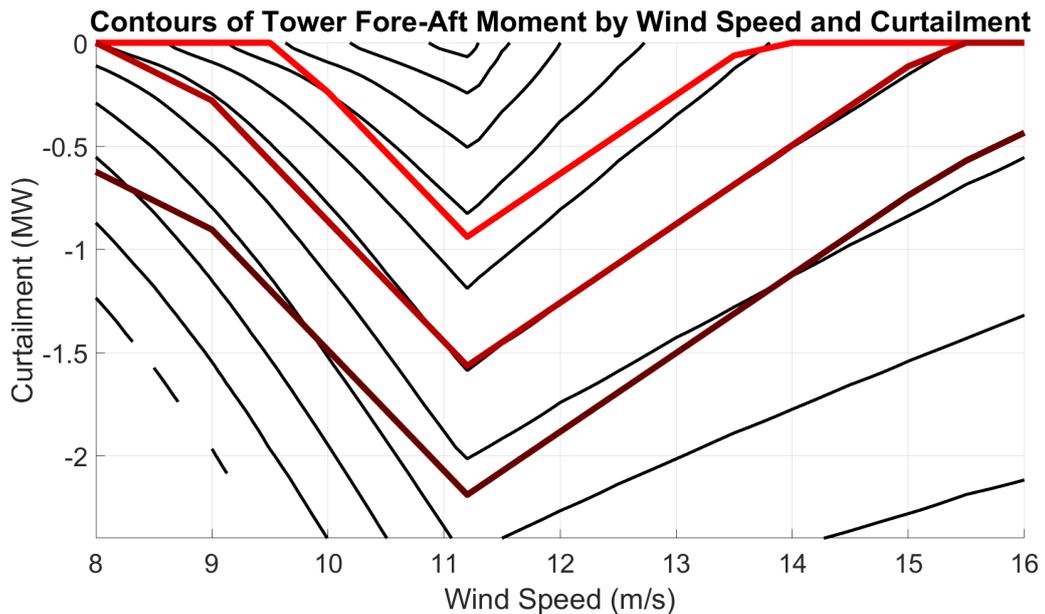


Figure 6.21: A contour plot of the steady state fore-aft bending moment of the tower with the intelligent tower control strategy at different curtailment levels.

Figure (6.21) shows the steady state fore-aft bending moment of the tower with the intelligent tower control strategy at different curtailment levels for the Supergen 5MW exemplar wind turbine. It can be seen that this strategy follows the contours for the fore-aft tower bending moment which should result in a reduction in tower fatigue.

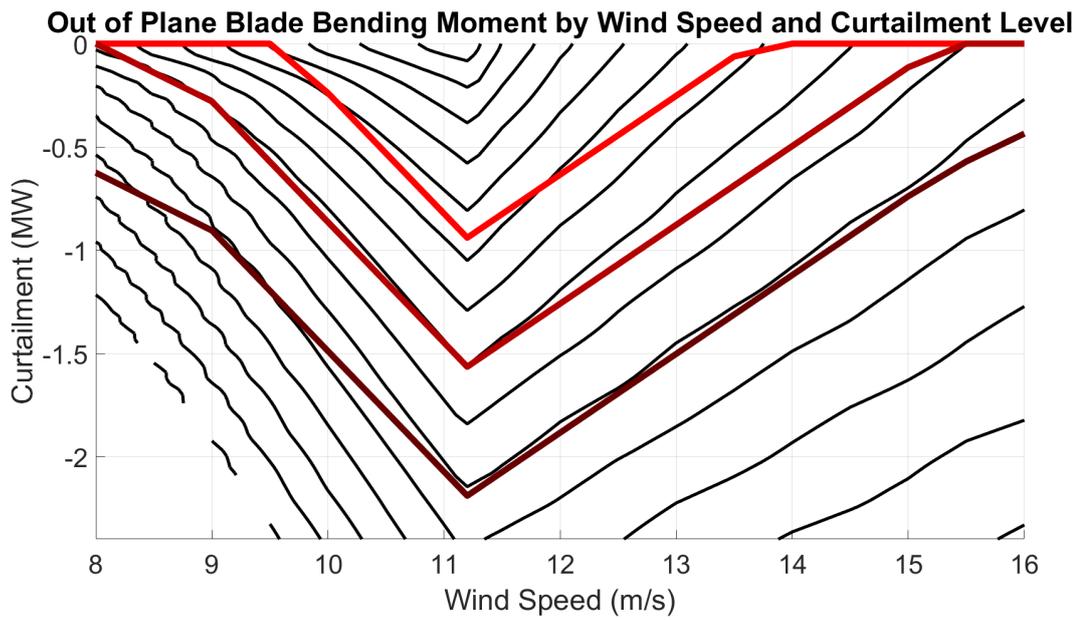


Figure 6.22: A contour plot of the steady state out of plane bending moment of the blade with the intelligent tower control strategy at different curtailment levels.

Figure (6.22) shows the steady state out of plane bending moment of the blade with the intelligent tower control strategy at different curtailment levels for the Supergen 5MW exemplar wind turbine. It can be seen that this strategy aligns with the contours for the out of plane bending moment which should result in a reduction in blade fatigue.

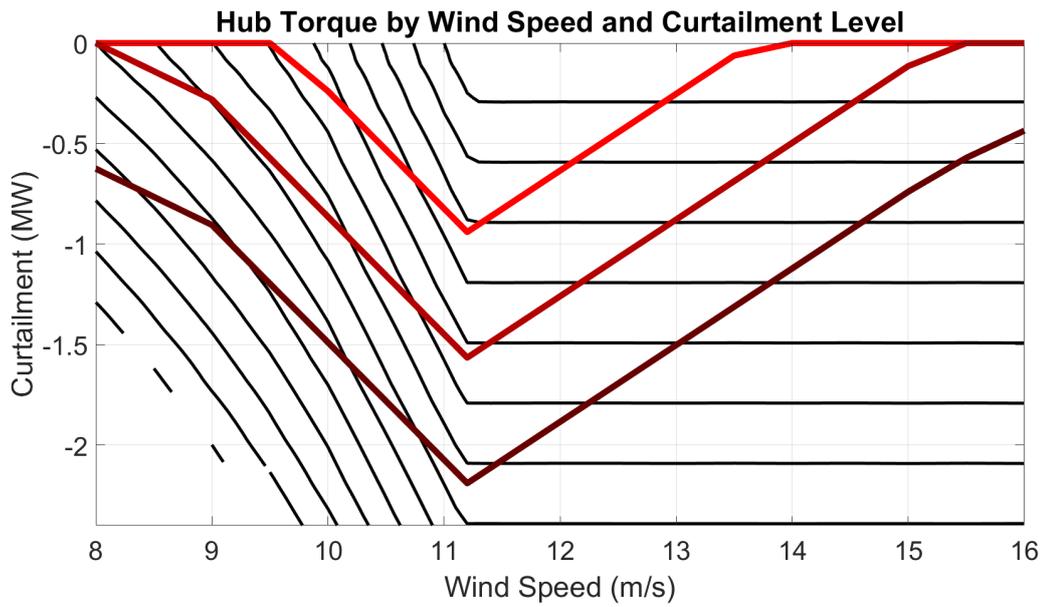


Figure 6.23: A contour plot of the steady state hub torques with the intelligent tower control strategy at different curtailment levels.

Figure (6.23) shows the hub torque contour plot of the steady state hub torques with the intelligent tower control strategy at different curtailment levels for the Supergen 5MW exemplar wind turbine. It can be seen that the allocation strategy does not align with the contours in the plot at above rated wind speeds.

Intelligent Tower Control Component Plots

The impact that this “New Baseline” has on the wind turbine over a steady state power curve is shown in Figures (6.21), (6.22) and (6.23). As the strategy has been designed around reducing variation of tower fore aft bending moment changes rather than for the out-of plane bending moment, the reductions in out-of plane blade bending moment DELs could be reduced further with a different approach. There is no reason that this strategy could not be adapted for prioritising reductions in blade DELs but as the tower is integral to the wind turbine’s structure it has been prioritised in this strategy.

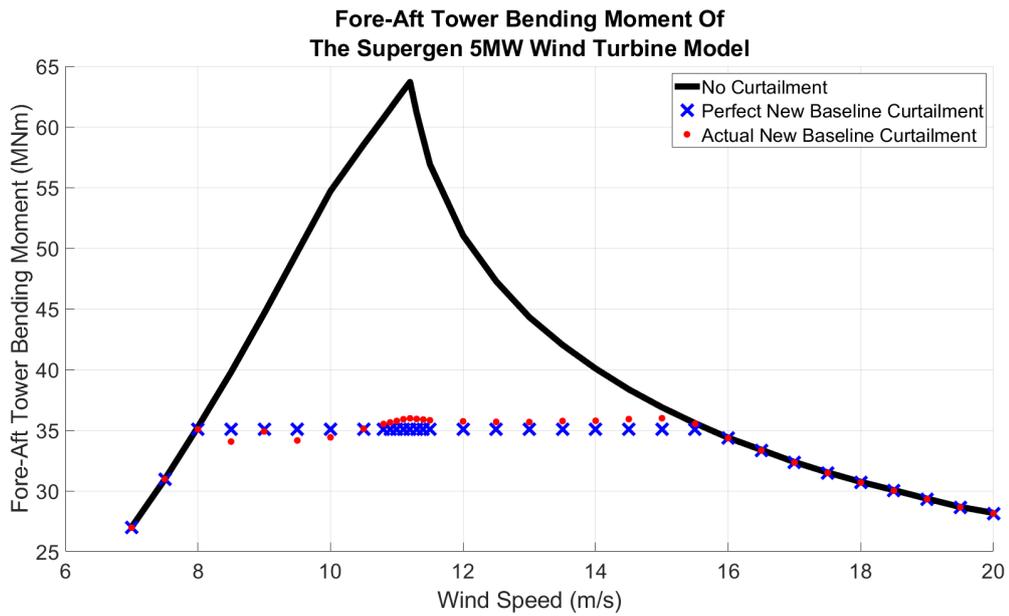


Figure 6.24: The idealized fore-aft tower bending moment with the intelligent tower control strategy

Figure (6.24) shows the tower fore-aft bending moment of the Supergen 5MW exemplar wind turbine in normal operation and when it is downrated to the baseline of the intelligent tower control strategy. It can be seen that when downrated to these values the tower bending moment is almost unchanging between 8 and 15 m/s. This is obviously idealised as the control strategy will need to sometimes allocated curtailment levels different to the baseline level to ensure set-point tracking but as these differences are from a near perfect baseline its implementation should lead to reductions in fatigue levels.

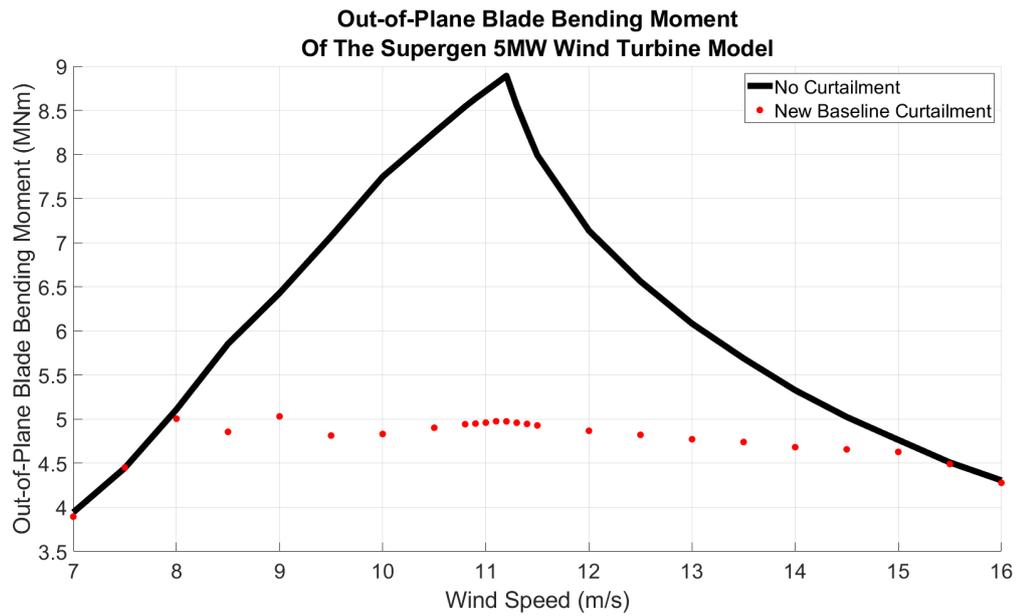


Figure 6.25: The idealized out of plane blade bending moment with the intelligent tower control strategy

Figures (6.25) shows the blade out of plane root bending moment of the Supergen 5MW exemplar wind turbine in normal operation and when it is downrated to the baseline of the intelligent tower control strategy. As should be expected due to the contour similarities this graph is very similar in shape to that presented in Figures (6.24). While it is again imperfect the new baseline level of power should result in flat bending moments between 8 and 15 m/s.

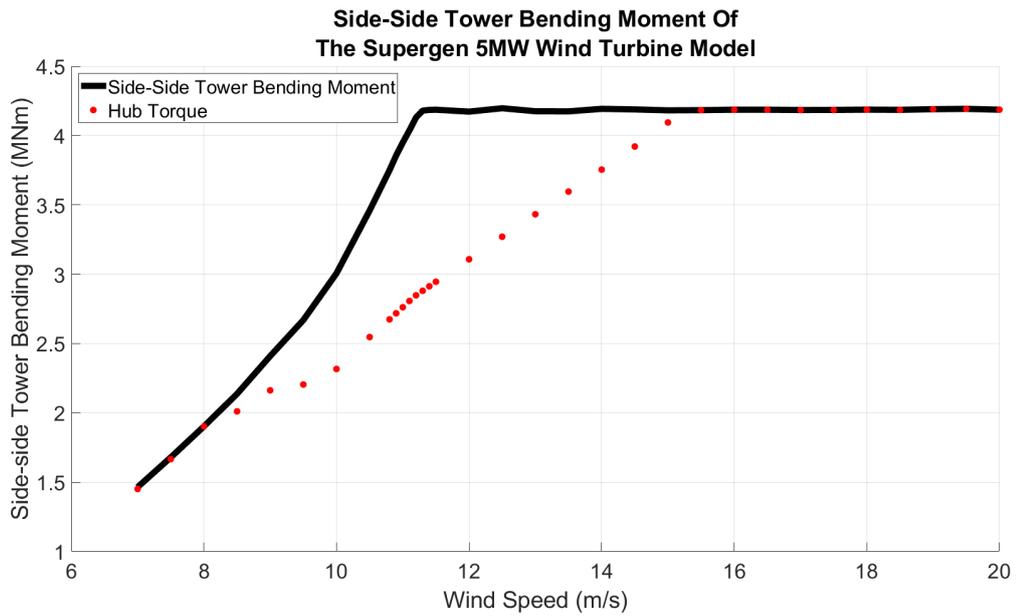


Figure 6.26: The idealized side-side tower bending moment with the intelligent tower control strategy.

Figure (6.26) shows the steady state side-side bending moment of the Supergen 5MW exemplar wind turbine in normal operation and when it is downrated to the baseline of the intelligent tower control strategy. It can be seen that the curtailment allocation results in a shallower slope in the tower side-side bending moment resulting in lower absolute loads. This could increase DELs however as there is a wider range of speeds where the moment has an underlying variation.

6.6 Validation Methodology

6.6.1 Validation

At present there are no official standards for how to validate changes in damage equivalent loads across a wind farm. Where appropriate and available the wind turbine standards [1] have been used. The approach used here is the one detailed in [1] with a wind farm of 16 turbines spaced 1000 meters apart in a four by four square. Simulations were then run at wind speeds 2m/s apart between 4 and 24 m/s with turbulence intensities of 12%, 14% and 16%, at wind direction angles of 0, 15, 30 and 45 degrees. For each combination of variables six different wind fields were used to increase the accuracy of the results with each simulation having the WFC active for 600 seconds. This created a total of 792 different wind fields. In each of these wind fields each of the three strategies were simulated with three different set-points listed in the table below. As the PACs turn off at low wind speeds the WFC was not activated for the simulations at 4m/s, 6m/s or 8m/s. The PACs were turned on when the estimated wind speed of a turbine exceeded 8m/s. To prevent the PACs oscillating between being turned on and off a hysteresis loop was implemented with the turbines being turned off when the estimated wind speed went below 6m/s.

6.6.2 Weibull Comparison

The results are weighted against 3 different Weibull distributions with means at 8, 9 and 10 m/s. As the results have been generated at 2m/s intervals the Weibull distributions have been binned into 11 wind speeds. The 4m/s bin and the 24 m/s bin also include probabilities for lower and higher wind speeds in the Weibull distribution. As the DELs at 4m/s are very low and the odds of exceeding 25m/s being also very low this assumption does not have a large impact on the total DELs but as they cannot be properly simulated in this model these assumptions contribute to the uncertainty.

6.6.3 Damage Equivalent Loads

Damage equivalent loads have been calculated using the approach detailed by [52] which decomposes a time series of bending moment data into a number of cycles and half cycles using

rainflow counting. The amplitude of these cycles is weighted on an S-N curve in comparison with the 1 Hz baseline to find the damage equivalent loads for each component. Five wind turbine components are considered in this research: fore-aft and side-side tower bending moment, in and out of plane blade bending moment, and hub torque. Hub torque is used as it is the best proxy available in the outputs from Strathfarm for examining any impacts on drivetrain loads. For the blade DELs a slope of 10 has been assumed for the S-N curve as they are assumed to be composed of some form of fiber glass composite [4]. The tower and hub DELS have been calculated with a slope of 3 has been assumed for the S-N curve as they are assumed to be primarily made of steel [4]. Hub torque has been used here as an indicator of drivetrain fatigue. Previous research looking at drivetrain DELs has considered each component individually [40]. However, here it is being assumed that the DELs calculated from the hub torque are a reasonable approximation for the drivetrain.

6.6.4 Choice of Farm Power Set-Points

For the simulations run for this chapter three sets of set-points have been used: an upper set with values around a farm wide curtailment of 10%, a middle set with values around a farm wide curtailment of 20% and a lower set with values around a farm wide curtailment of 30%. These values were chosen to demonstrate that the proposed curtailment allocation strategy shows improvements in damage equivalent loads across the full range of set-points that the WFC can deliver.

6.6.5 Turbine Level Results Methodology

Figure (6.27) shows the wind farm layout used in the validation process. For the results presented here the wind direction is assumed to be equally distributed in each of the 24 directions which can be simulated from the 15 degree incremental simulations. As a result of this the turbines only have 3 possible states in the wind farm due to the eight symmetries present in a square, these are that each turbine is either a corner, edge or center turbine.

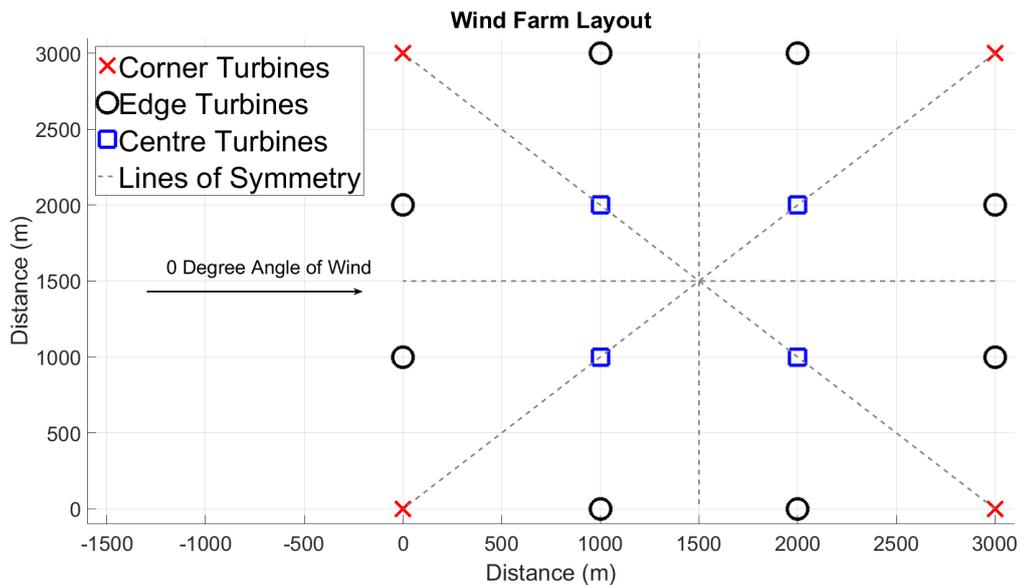


Figure 6.27: The wind farm layout used in the validation process.

6.6.6 Power Spectral Density Analysis

By combining PSDs of all of the turbines from all of the simulations of a control strategy at a given wind speed and farm level set-point a clearer plot can be generated. This is due to the noise present in each individual PSD cancelling out. Each PSD is therefore made up of approximately a week (633600 seconds) of simulated data.

6.6.7 The Impact of the Wind Rose on Turbine Tower Fatigue

It has been shown [51] that the wind direction distribution has an impact on fatigue seen in a wind turbine's tower.

However, as this analysis considers a uniform wind rose, the turbine level fatigue is uniform in each direction.

If this methodology was considered with a non uniform wind rose then the impact of the directionality of the wind would need to be included in the analysis.

In addition to this, the uniformity of the wind rose means that the DELs for the tower fore-aft and side-side directions are additive as tower fatigue is the same in each direction due to the weighting of the wind rose.

6.7 Unweighted Farm Level Simulation Results

This section focuses on the results at the wind farm level, considering the impact of the three curtailment allocation strategies over three curtailed wind farm power curves.

6.7.1 Upper Curtailment Level Power Curve Results

Upper Curtailment

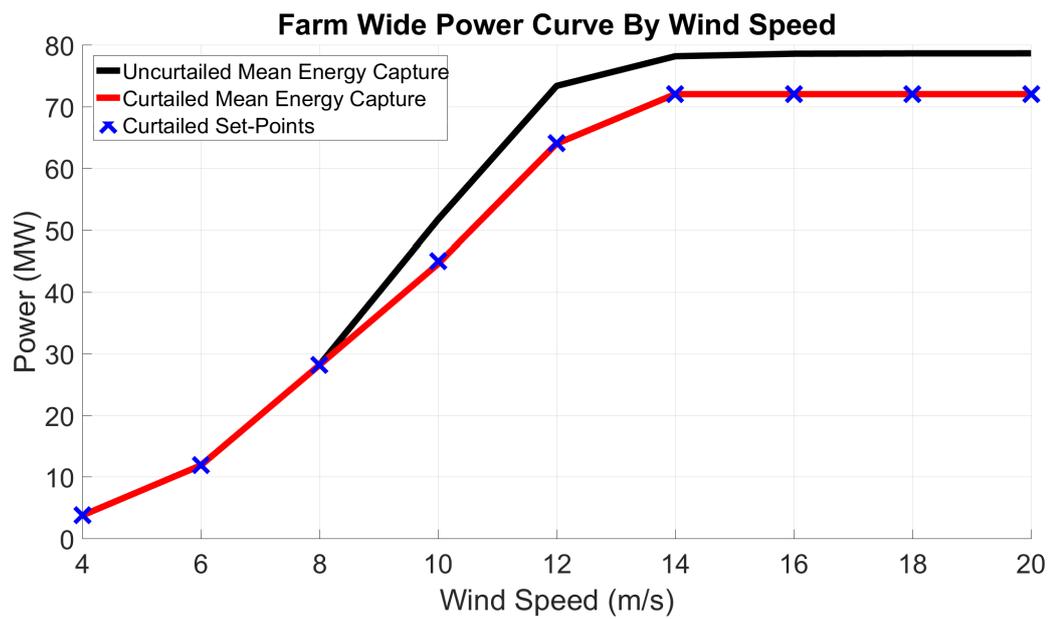


Figure 6.28: Wind farm power curve for the upper set-points

Figure (6.28) shows the upper level curtailment power curve for the wind farm level results.

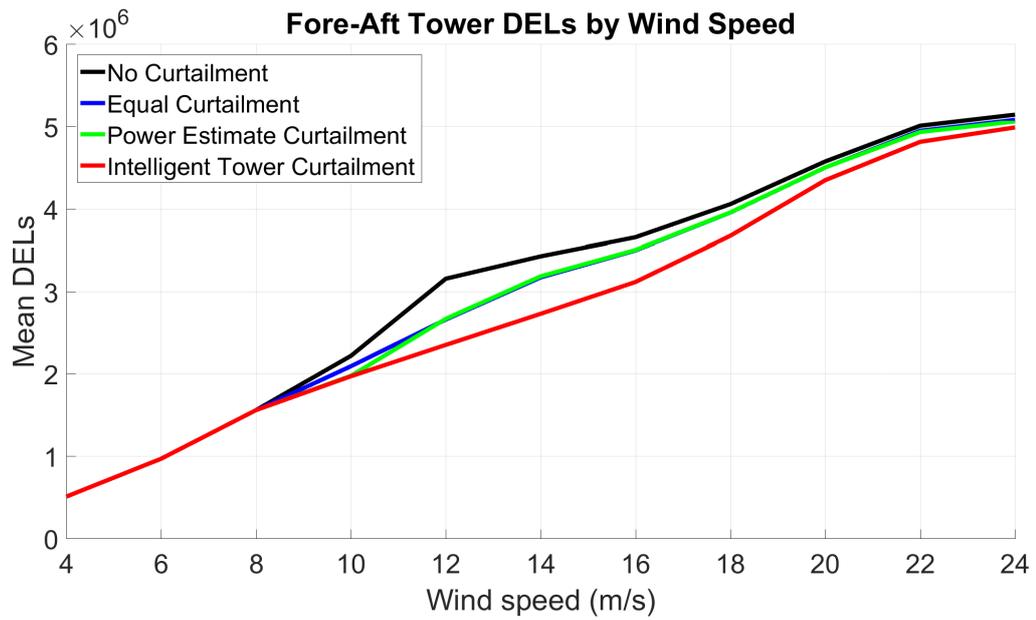


Figure 6.29: Mean fore-aft tower DELs by wind speed and curtailment strategy

Figure (6.29) shows the mean fore aft tower DELs for each of the three curtailment strategies and for not curtailing. It shows that the intelligent tower strategy has lower DELs than the other strategies across the wind speed where the controller is active but that the other two curtailment strategies also reduce fatigue compared with not curtailing. It can also be seen that as expected the intelligent control strategy sees the largest reduction in DELs near rated wind speed.

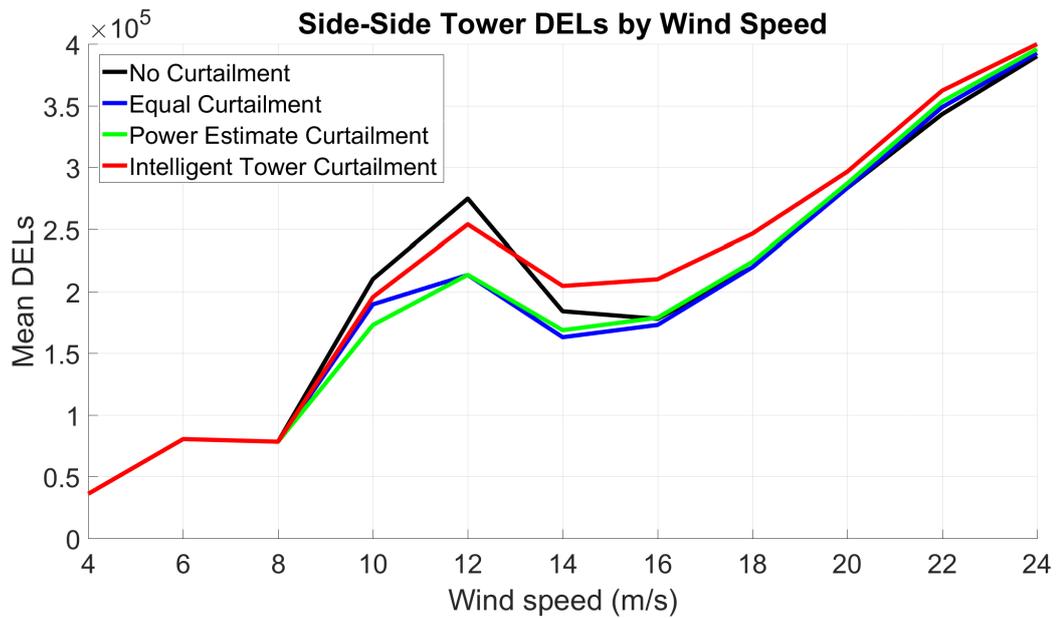


Figure 6.30: Mean side-side tower DELs by wind speed and curtailment strategy

Figure (6.30) shows the mean side-side tower DELs for each of the three curtailment strategies and for not curtailing. It can be seen that in contrast with Figure (6.29) that the intelligent control strategy leads to significantly higher DELs at near rated wind speeds than the other two curtailment strategies and at certain wind speeds results in higher DELs than not curtailing the wind farm. However, by comparing Figure (6.29) and Figure (6.30) it can be seen that the side-side DELs are an order of magnitude lower than the fore-aft DELs. This difference can be seen by considering a combined DEL for both the fore-aft and side side DEL added together, this can be seen in Figure (6.31).

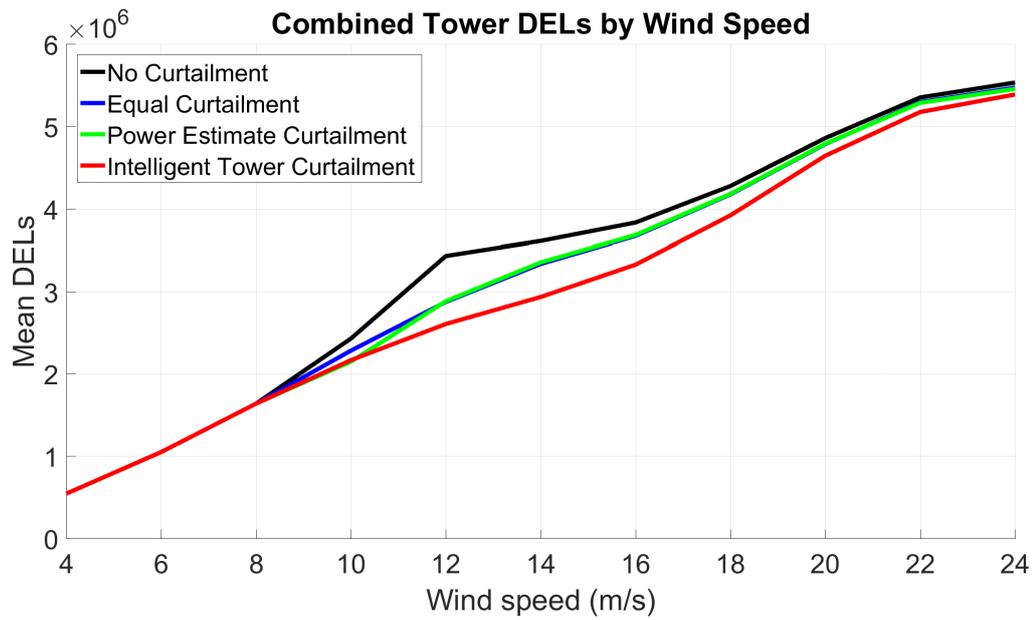


Figure 6.31: Mean combined tower DELs by wind speed and curtailment strategy for the upper set-points

Figure (6.31) shows the combined tower DELs for the curtailment allocation strategies for the upper farm level set-points. It can be seen that this plot is very similar to the fore-aft tower DELs seen in Figure (6.29) due to the fore-aft DELs dominating the tower side side DELs. For the remainder of this chapter tower damage equivalent loads will be discussed in terms of the combined side side and fore-aft value for simplicity.

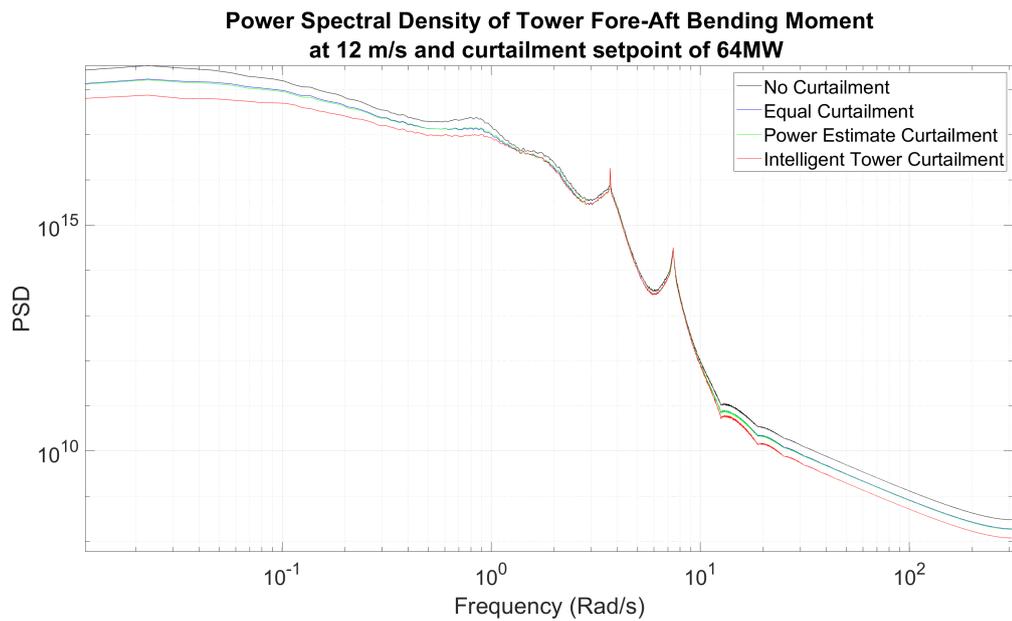


Figure 6.32: Fore-aft tower PSD at 12m/s with farm curtailment set-point of 64MW.

Figure (6.32) show the PSD of the fore-aft tower bending moments from the simulations at 12m/s with a 64 MW set-point for each of the three strategies. It can be seen in the PSD plot that at the lowest frequencies there is a lower power level for the intelligent control strategy which is where peak of the spectra occurs. While the differences do not look significant it should be noted that this plot is on a log scale so small differences at the upper end of the y axis represent very large differences on a linear scale.

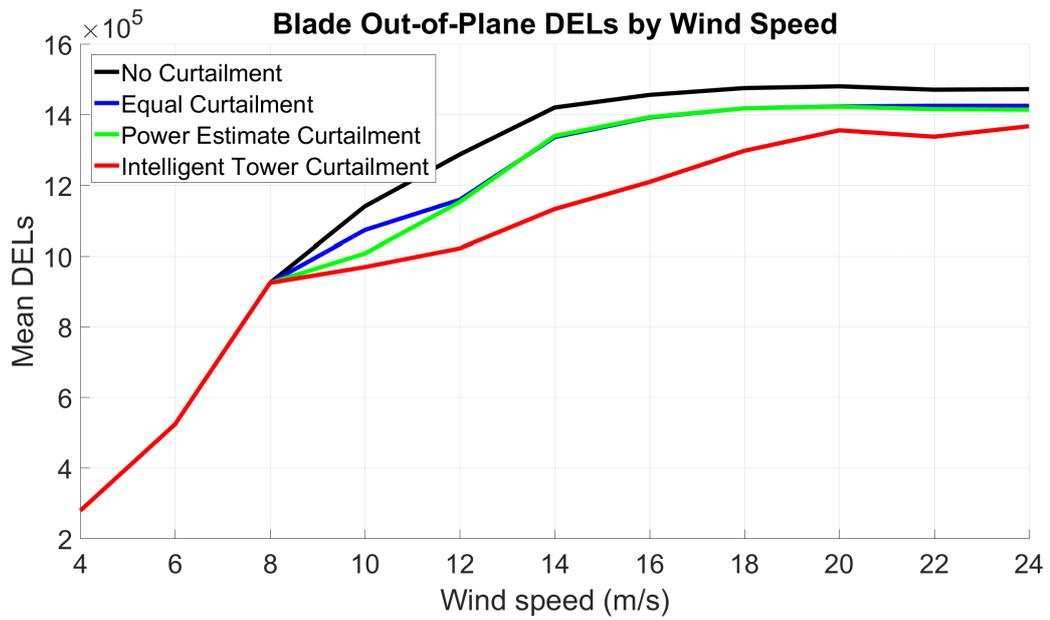


Figure 6.33: Mean out of plane blade DELs by wind speed and curtailment strategy for the upper set-points

Figure (6.33) shows the out of plane bending DELs for each curtailment allocation strategy by wind speed for the upper set-points of farm level curtailment. It can be seen that the intelligent tower control strategy has lower DELs than both the other curtailment allocation strategies and compared with not curtailing the wind farm.

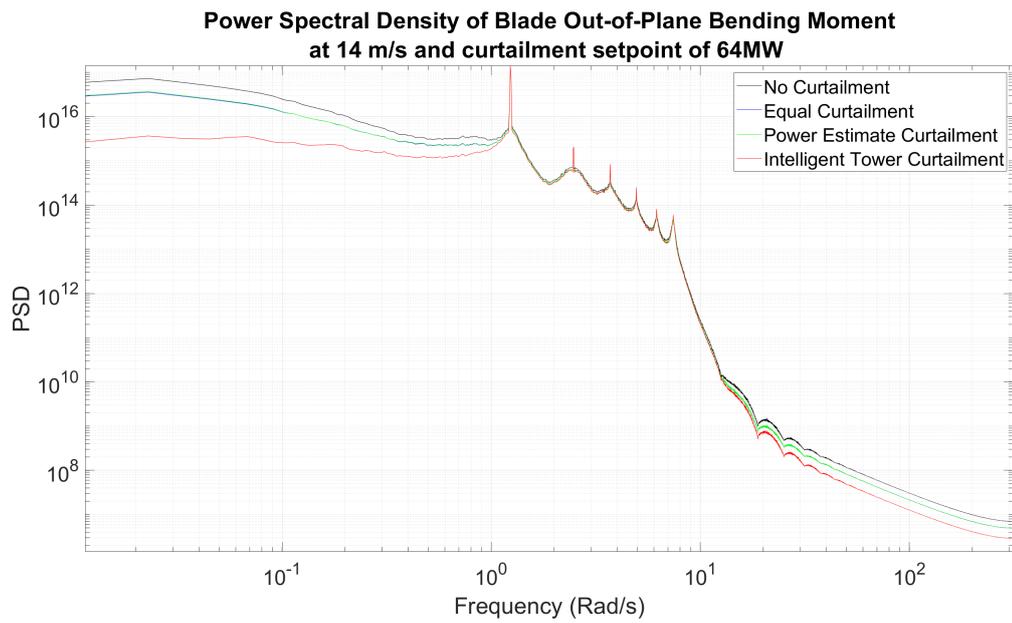


Figure 6.34: Out of plane blade PSD at 14m/s with farm curtailment set-point of 64MW.

Figure (6.34) show the PSD of the out of plane blade bending moments from the simulations at 14m/s with a 64 MW set-point for each of the three strategies. . As before there is a significant reduction in the magnitude of the spectra at the lowest frequencies where most of the power of the spectra are located hence the reduction in DELs.

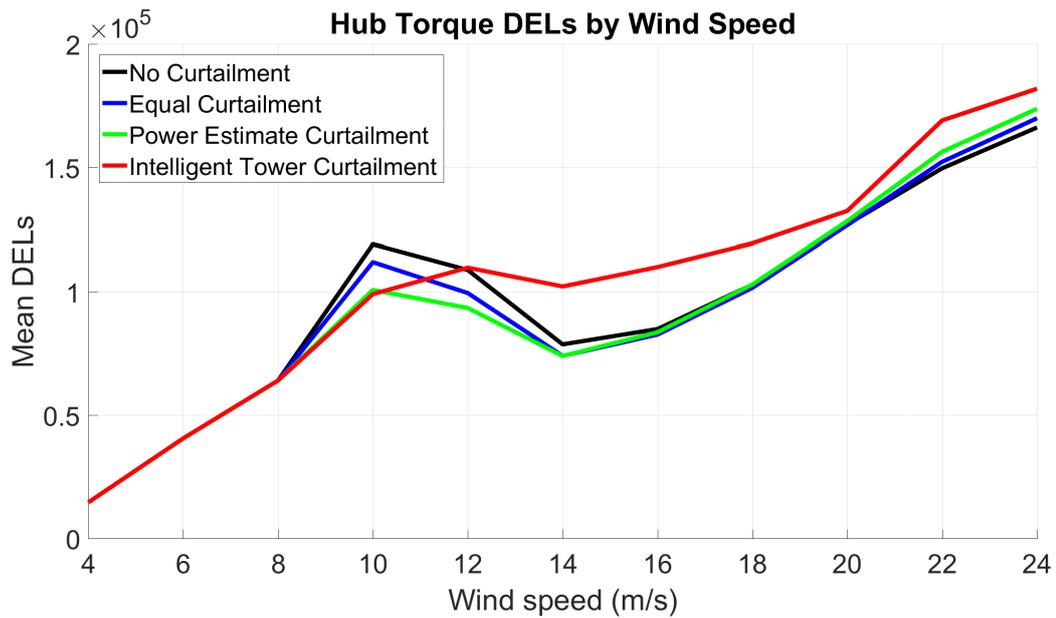


Figure 6.35: Mean hub torque DELs by wind speed and curtailment strategy for the upper set-points

Figure (6.35) shows the mean hub torque DELs for each strategy by wind speed. It is clear that the intelligent tower controller has higher DELs at above rated wind speeds. This is caused by the strategy varying the generator torque as the wind speed changes whereas the other strategies do not see such variation so have lower DELs.

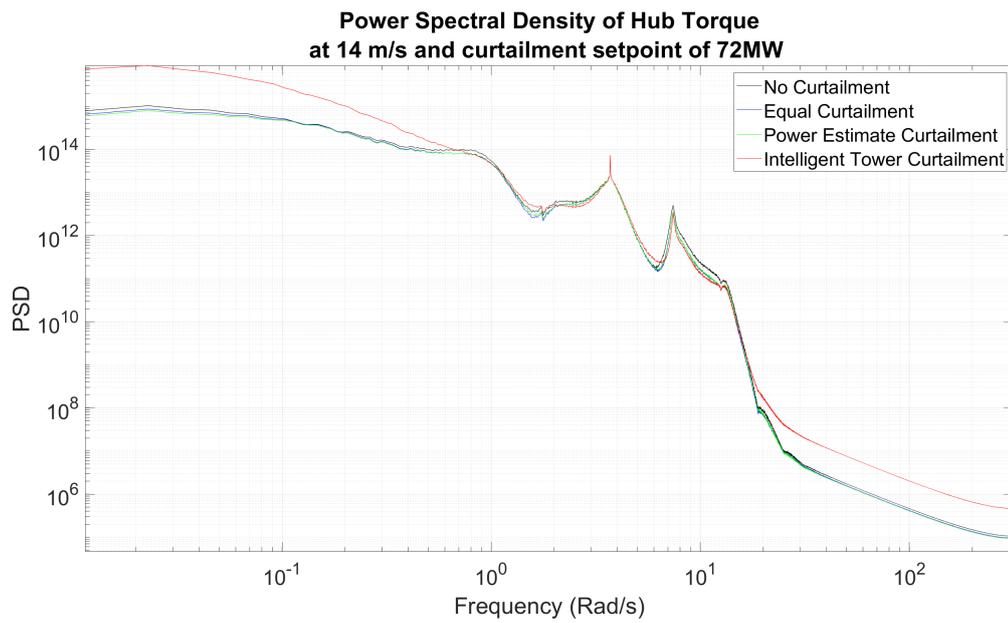


Figure 6.36: Hub torque PSD at 14m/s with farm curtailment set-point of 72MW.

This is also evidenced by Figure (6.36) which shows the PSD of the hub torque at 12m/s with a 64 MW set-point for each of the three strategies. It is clear that there is an increase in the hub torque at lower frequencies when the intelligent tower strategy is used compared to the other approaches. Hence the increase seen in DELs in Figure (6.35) .

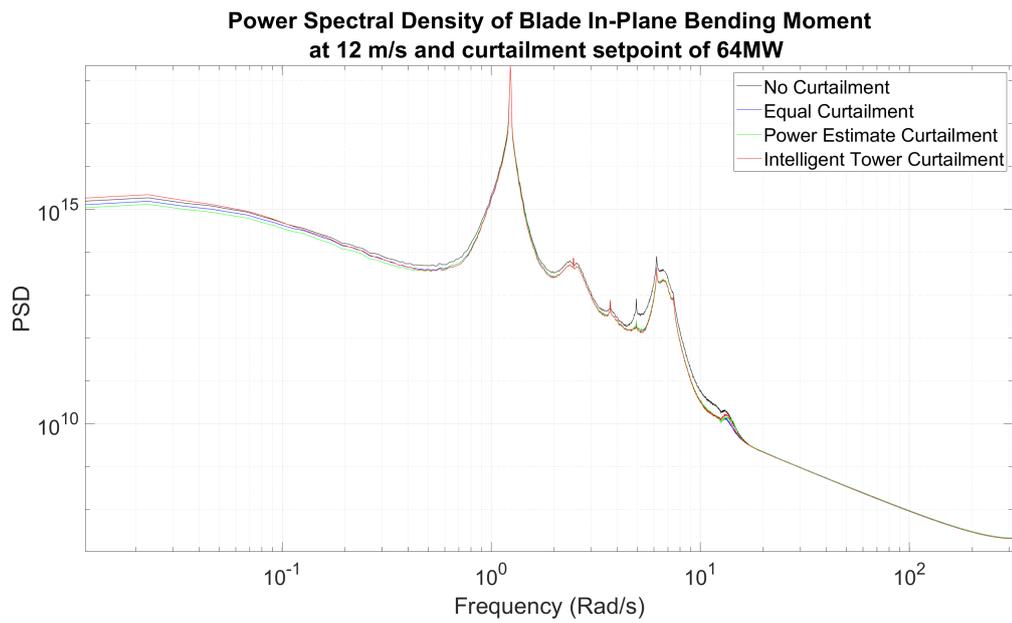


Figure 6.37: In plane blade PSD at 14m/s with farm curtailment set-point of 64MW.

Figure (6.37) shows the PSD of the in plane blade bending moments from the simulations at 12m/s with a 64 MW set-point for each of the three strategies. As would be expected all strategies see most of the power in the PSD at 1P.

6.7.2 Middle Curtailment Level Power Curve Results

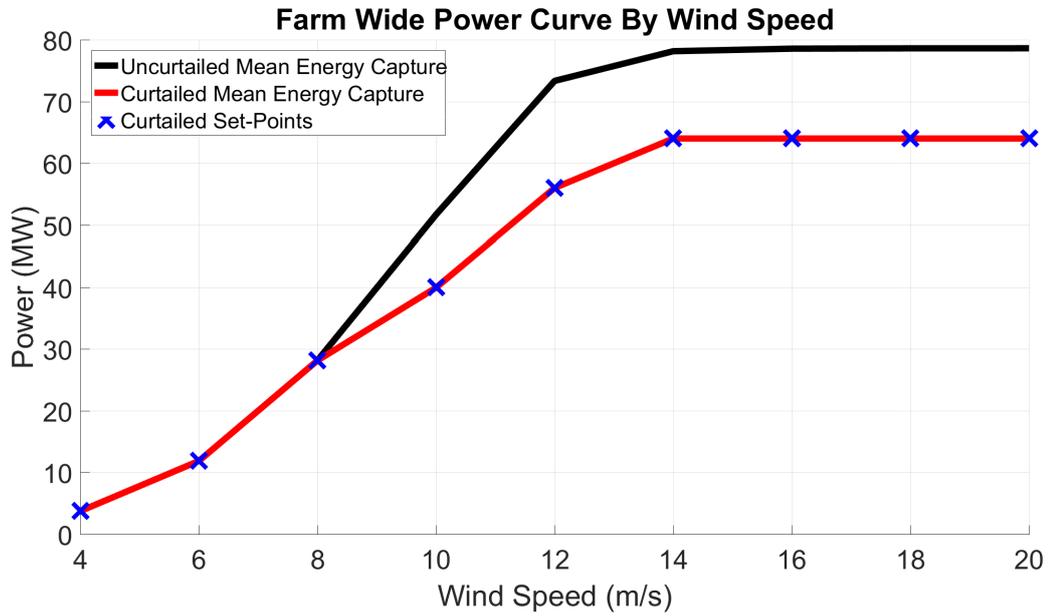


Figure 6.38: Wind farm power curve for the middle set-points

Figure (6.38) shows the middle set-points of the windfarm as a power curve.

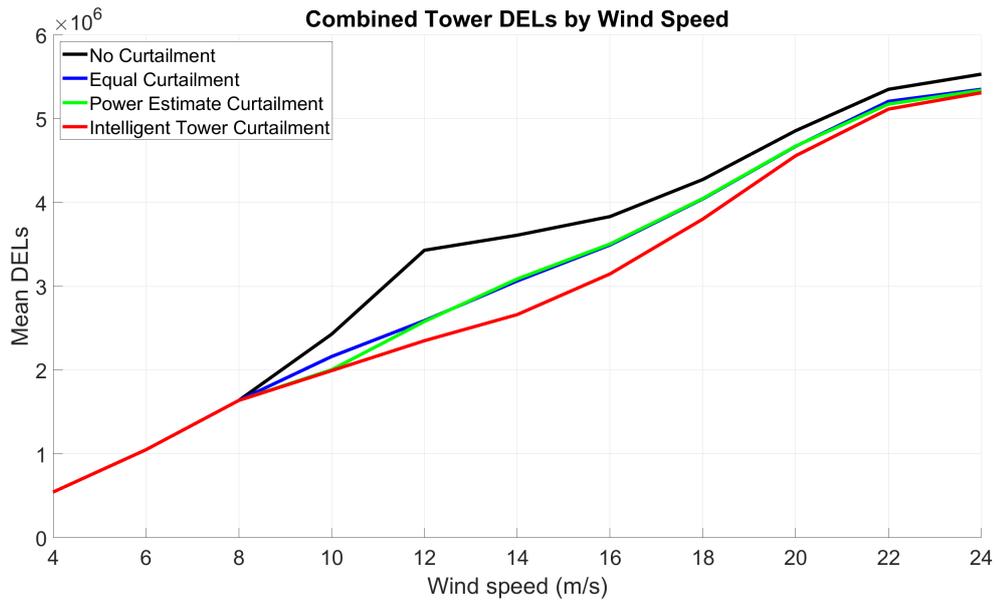


Figure 6.39: Mean combined tower DELs by wind speed and curtailment strategy for the middle set-points

Figure (6.39) shows the combined tower damage equivalent loads for the windfarm when the middle set of power set-points are used. It should be noted that the DELs here are much lower than the values seen for the upper set of set-point power outputs as would be expected due the increased level of curtailment.

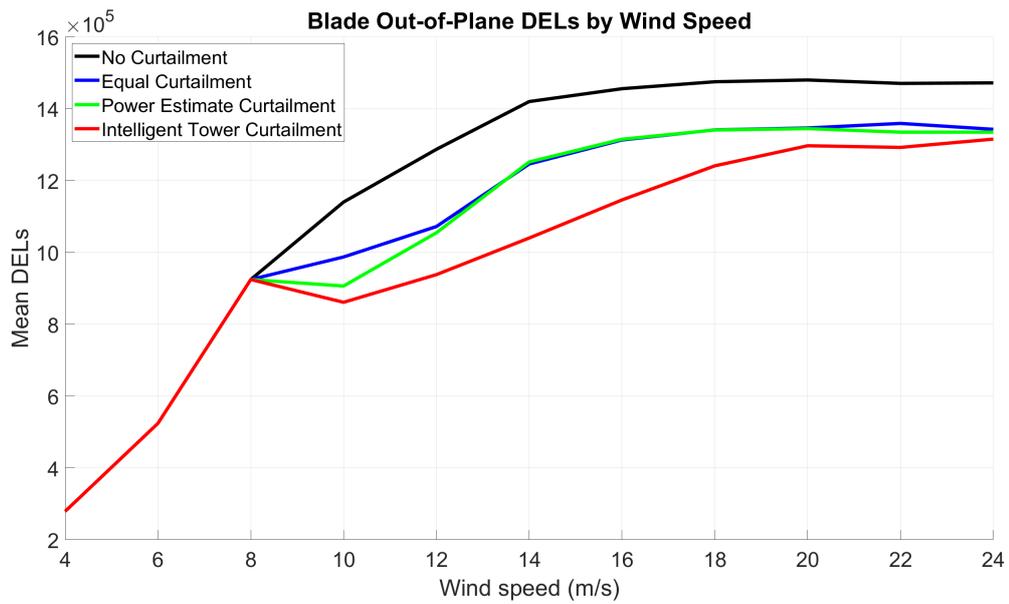


Figure 6.40: Mean out of plane blade DELs by wind speed and curtailment strategy for the middle set-points

Figure (6.40) shows the out-of plane blade damage equivalent loads for the windfarm when the middle set of power set-points are used. It should be noted that the DELs here are much lower than the values seen for the upper set of set-point power outputs as would be expected due the increased level of curtailment.

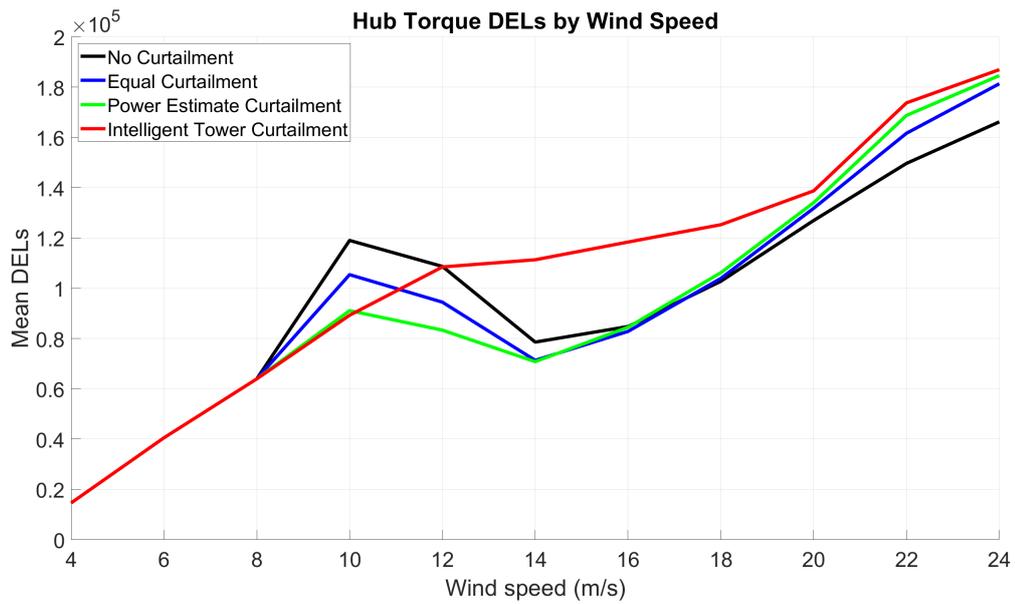


Figure 6.41: Mean hub torque DELs by wind speed and curtailment strategy for the middle set-points

Figure (6.41) shows the drive train damage equivalent loads (using hub torque as a proxy variable) for the windfarm when the middle set of power set-points are used. It should be noted that the DELs here are much lower than the values seen for the upper set of set-point power outputs as would be expected due to the increased level of curtailment.

6.7.3 Lower Curtailment Level Power Curve Results

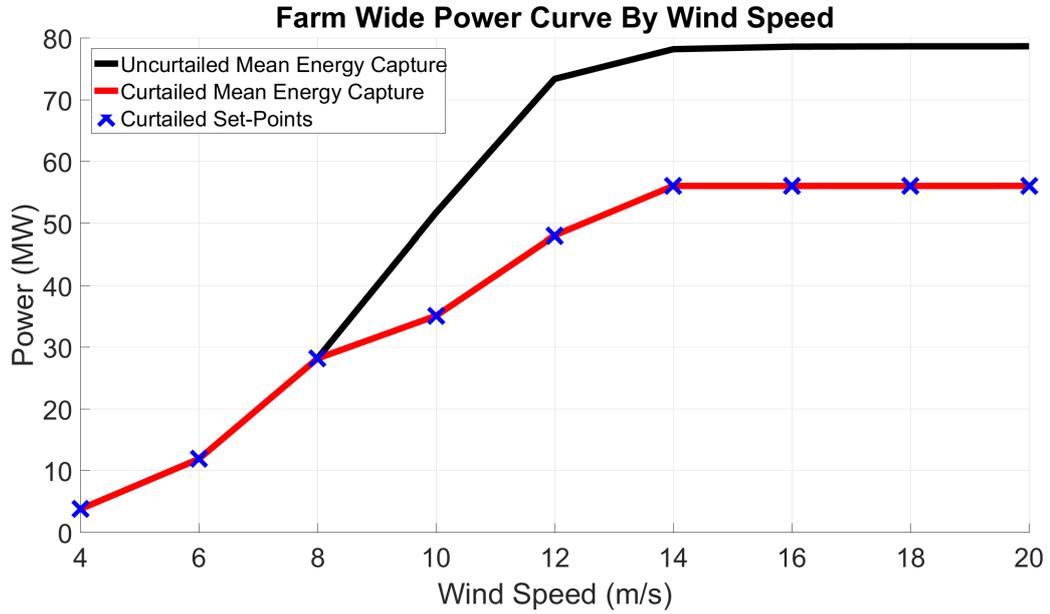


Figure 6.42: Wind farm power curve for the lower set-points

Figure (6.42) shows the lower set-points of the windfarm as a power curve.

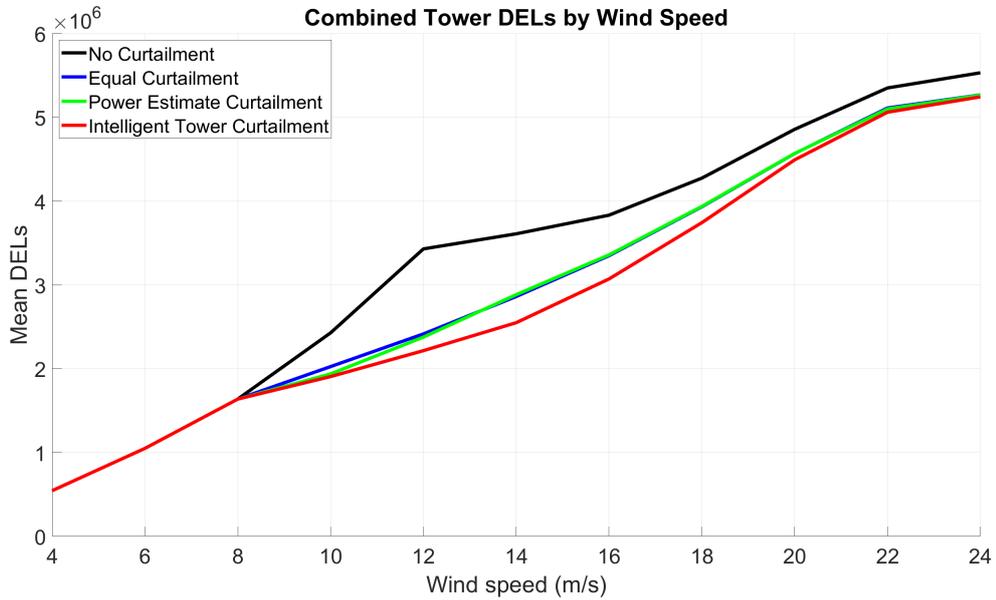


Figure 6.43: Mean combined tower DELs by wind speed and curtailment strategy for the lower set-points

Chapter 6. Designing a Wind Farm Controller to Reduce Wind Turbine Fatigue

Figure (6.43) shows the combined tower damage equivalent loads for the windfarm when the lower set of power set-points are used. It should be noted that the DELs here are much lower than the values seen for the upper and middle sets of set-point power outputs as would be expected due the increased level of curtailment.

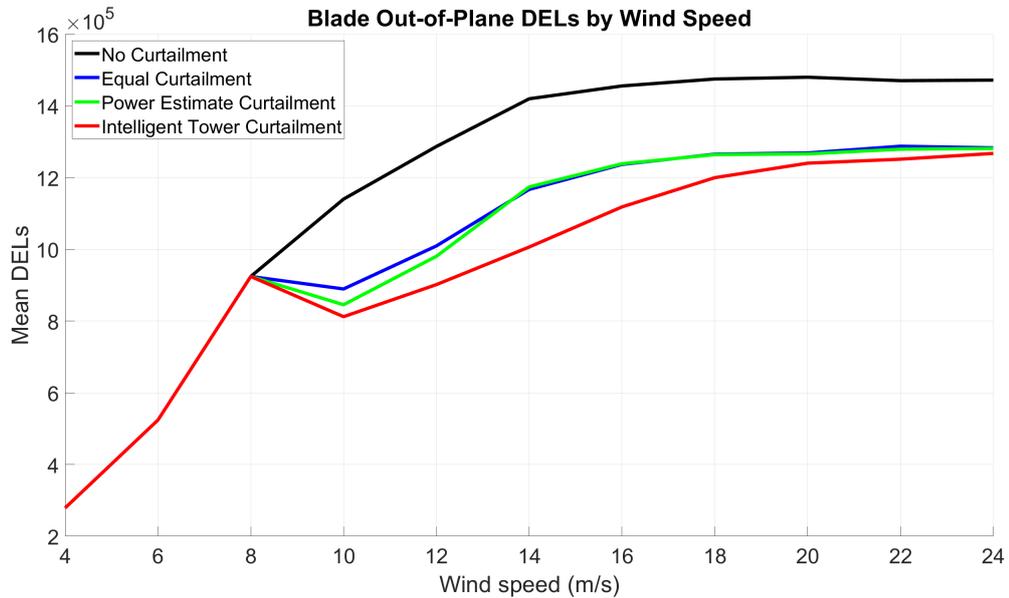


Figure 6.44: Mean out of plane blade DELs by wind speed and curtailment strategy for the lower set-points

Figure (6.44) shows the out-of plane blade damage equivalent loads for the windfarm when the lower set of power set-points are used. It should be noted that the DELs here are much lower than the values seen for the upper and middle sets of set-point power outputs as would be expected due the increased level of curtailment.

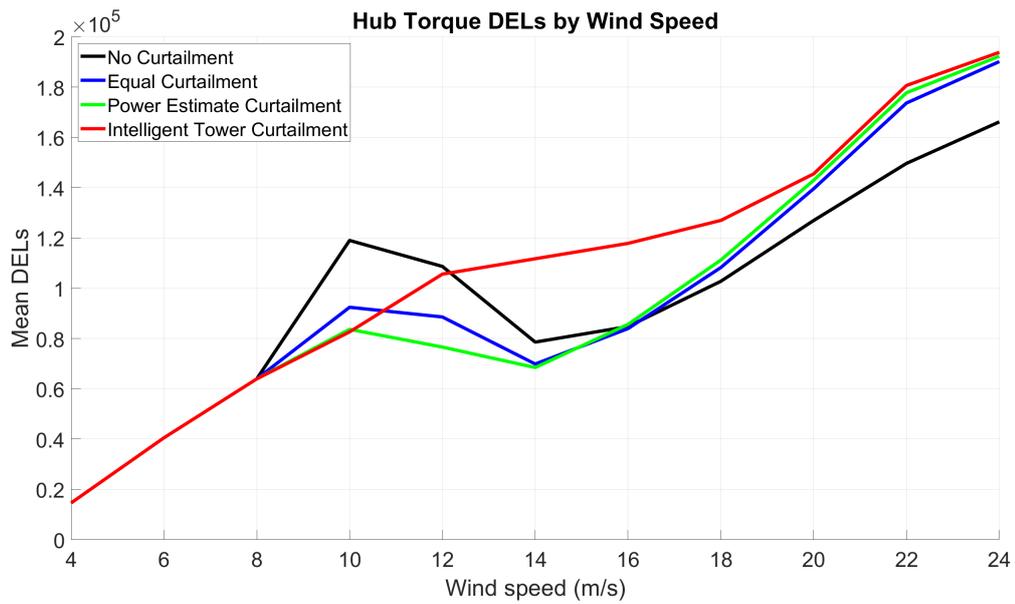


Figure 6.45: Mean hub torque DELs by wind speed and curtailment strategy for the lower set-points

Figure (6.45) shows the drive train damage equivalent loads (using hub torque as a proxy variable) for the windfarm when the lower set of power set-points are used. It should be noted that the DELs here are much lower than the values seen for the upper and middle sets of set-point power outputs as would be expected due the increased level of curtailment.

6.7.4 Weibull Weighted DEL Changes for Weibull with mean wind speeds of 8, 9 and 10 m/s

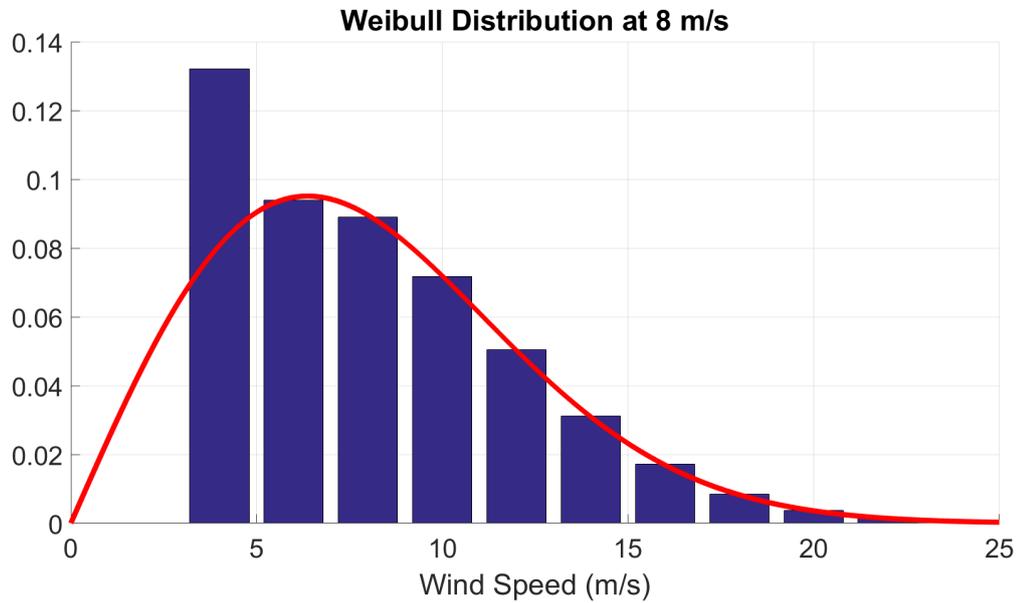


Figure 6.46: Bins used for the Weibull distribution used with a mean speed of 8m/s.

Figure (6.46) shows the binning of the Weibull distribution with a mean wind speed of 8 m/s which is used to weight the DELs so that a lifetime DEL comparison can be made between the three control strategies and not curtailing the wind turbines.

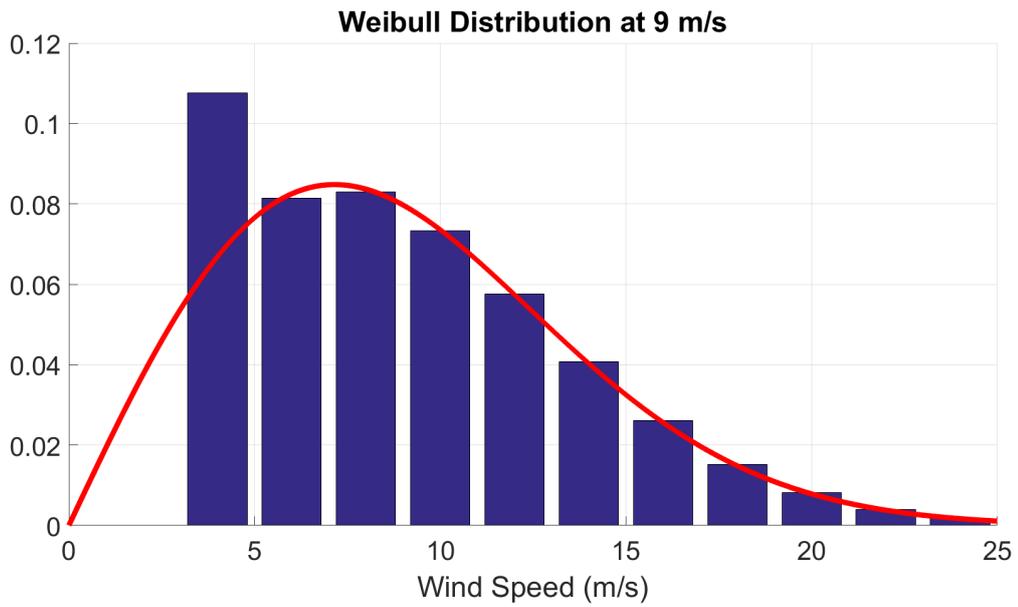


Figure 6.47: Bins used for the Weibull distribution used with a mean speed of 9m/s.

Figure (6.47) shows the binning of the Weibull distribution with a mean wind speed of 9 m/s which is used to weight the DELs so that a lifetime DEL comparison can be made between the three control strategies and not curtailing the wind turbines.

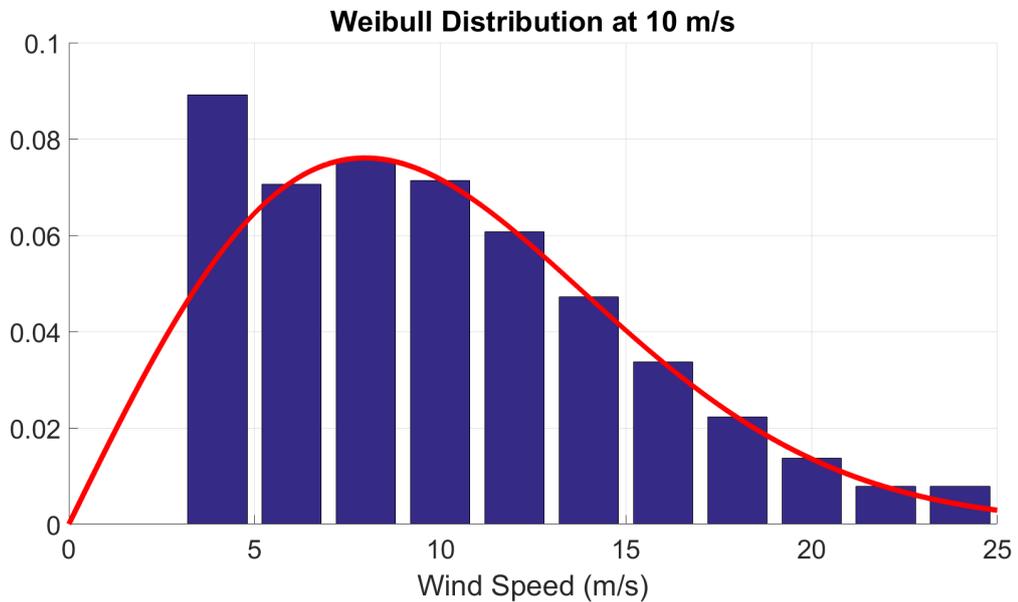


Figure 6.48: Bins used for the Weibull distribution used with a mean speed of 10m/s.

Chapter 6. Designing a Wind Farm Controller to Reduce Wind Turbine Fatigue

Figure (6.48) shows the binning of the Weibull distribution with a mean wind speed of 10 m/s which is used to weight the DELs so that a lifetime DEL comparison can be made between the three control strategies and not curtailing the wind turbines.

Set-Points	Curtailment Strategy	Weibull Mean		
		8m/s	9m/s	10m/s
Upper Set-Points	Equal Curtailment	- 5.914 %	- 6.0017 %	- 5.7747 %
	Power Estimation Curtailment	- 6.8314 %	- 6.7676 %	- 6.4038 %
	Intelligent Tower Curtailment	- 10.9467 %	- 11.4802 %	- 11.3728 %
Middle Set-Points	Equal Curtailment	- 10.2764 %	- 10.5482 %	- 10.2942 %
	Power Estimation Curtailment	- 11.5296 %	- 11.6196 %	- 11.1931 %
	Intelligent Tower Curtailment	- 15.5715 %	- 16.2072 %	- 15.9814 %
Lower Set-Points	Equal Curtailment	- 13.7286 %	- 14.1226 %	- 13.8391 %
	Power Estimation Curtailment	- 14.627 %	- 14.9035 %	- 14.4998 %
	Intelligent Tower Curtailment	- 17.9368 %	- 18.6098 %	- 18.3326 %

Table 6.1: Change in combined tower DELs by control strategy and wind speed compared to not curtailing.

Table (6.1) shows the change in combined tower DELs for each of the three strategies over Weibull distributions with mean wind speeds of 8m/s, 9m/s and 10m/s. The table shows that across all three mean wind speeds and curtailment levels that the intelligent tower controller has the lowest tower DELs.

Set-Points	Curtailment Strategy	Weibull Mean		
		8m/s	9m/s	10m/s
Upper Set-Points	Equal Curtailment	- 4.1176 %	- 4.4104 %	- 4.5418 %
	Power Estimation Curtailment	- 5.3488 %	- 5.5354 %	- 5.5569 %
	Intelligent Tower Curtailment	- 10.7521 %	- 11.7444 %	- 12.2452 %
Middle Set-Points	Equal Curtailment	- 8.4344 %	- 9.0576 %	- 9.4054 %
	Power Estimation Curtailment	- 10.1316 %	- 10.6239 %	- 10.8258 %
	Intelligent Tower Curtailment	- 15.4289 %	- 16.6184 %	- 17.1722 %
Lower Set-Points	Equal Curtailment	- 12.5041 %	- 13.4144 %	- 13.9395 %
	Power Estimation Curtailment	- 13.7486 %	- 14.5785 %	- 15.0004 %
	Intelligent Tower Curtailment	- 17.6304 %	- 18.9539 %	- 19.6125 %

Table 6.2: Change in out of plane blade DELs by control strategy and wind speed compared to not curtailing.

Table (6.2) shows the change in out of plane blade DELs for each of the three strategies over Weibull distributions with mean wind speeds of 8m/s, 9m/s and 10m/s. The table shows that across all three mean wind speeds and curtailment levels that the intelligent tower controller

has the lowest out of plane blade DELs.

Set-Points	Curtailment Strategy	Weibull Mean		
		8m/s	9m/s	10m/s
Upper Set-Points	Equal Curtailment	- 0.5508 %	- 0.5706 %	- 0.5549 %
	Power Estimation Curtailment	- 0.5621 %	- 0.5796 %	- 0.5589 %
	Intelligent Tower Curtailment	- 0.5698 %	- 0.584 %	- 0.5512 %
Middle Set-Points	Equal Curtailment	- 0.8126 %	- 0.8445 %	- 0.8109 %
	Power Estimation Curtailment	- 0.8296 %	- 0.8583 %	- 0.8182 %
	Intelligent Tower Curtailment	- 0.8249 %	- 0.8511 %	- 0.8066 %
Lower Set-Points	Equal Curtailment	- 1.0039 %	- 1.0372 %	- 0.9803 %
	Power Estimation Curtailment	- 1.0155 %	- 1.0473 %	- 0.9871 %
	Intelligent Tower Curtailment	- 1.0051 %	- 1.0351 %	- 0.9724 %

Table 6.3: Change in in plane blade DELs by control strategy and wind speed compared to not curtailing.

Table (6.3) shows the change in in plane blade DELs for each of the three strategies over Weibull distributions with mean wind speeds of 8m/s 9m/s and 10m/s. The table shows that there is little variation in the DELs between the different strategies and curtailment levels.

Set-Points	Curtailment Strategy	Weibull Mean		
		8m/s	9m/s	10m/s
Upper Set-Points	Equal Curtailment	- 3.9241 %	- 3.9416 %	- 3.6877 %
	Power Estimation Curtailment	- 7.3621 %	- 7.0766 %	- 6.3488 %
	Intelligent Tower Curtailment	+0.1501 %	+2.0525 %	+3.7923 %
Middle Set-Points	Equal Curtailment	- 6.4445 %	- 6.164 %	- 5.3326 %
	Power Estimation Curtailment	- 11.5756 %	- 10.8924 %	- 9.4108 %
	Intelligent Tower Curtailment	- 0.8991 %	+1.8028 %	+4.3119 %
Lower Set-Points	Equal Curtailment	- 10.3299 %	- 9.5185 %	- 7.8808 %
	Power Estimation Curtailment	- 14.5538 %	- 13.4547 %	- 11.3037 %
	Intelligent Tower Curtailment	- 2.9011 %	+0.1207 %	+3.1255 %

Table 6.4: Change in hub torque DELs by control strategy and wind speed compared to not curtailing.

Table (6.4) shows the change in in plane blade DELs for each of the three strategies over Weibull distributions with mean wind speeds of 8m/s 9m/s and 10m/s. The table shows that when the turbines are equally curtailed or when the power estimation curtailment is used there is a decrease in the hub torque DELs. However when the intelligent tower controller is used there a negligible change at 8m/s and small increases at 9m/s and 10m/s.

Chapter 6. Designing a Wind Farm Controller to Reduce Wind Turbine Fatigue

Set-Points	Curtailment Strategy	Weibull Mean		
		8m/s	9m/s	10m/s
Upper Set-Points	Equal Curtailment	- 8.1189 %	- 8.4297 %	- 8.5638 %
	Power Estimation Curtailment	- 8.1389 %	- 8.4764 %	- 8.6701 %
	Intelligent Tower Curtailment	- 8.2375 %	- 8.7011 %	- 9.1437 %
Middle Set-Points	Equal Curtailment	- 15.6286 %	- 16.5755 %	- 17.2068 %
	Power Estimation Curtailment	- 15.6635 %	- 16.6566 %	- 17.3647 %
	Intelligent Tower Curtailment	- 15.7018 %	- 16.747 %	- 17.55 %
Lower Set-Points	Equal Curtailment	- 23.2991 %	- 24.8493 %	- 25.8872 %
	Power Estimation Curtailment	- 23.3155 %	- 24.8849 %	- 25.9535 %
	Intelligent Tower Curtailment	- 23.3371 %	- 24.9408 %	- 26.0734 %

Table 6.5: Change in energy capture by control strategy and wind speed compared to not curtailing.

Table (6.5) shows the energy capture for each of the three strategies over Weibull distributions with mean wind speeds of 8m/s, 9m/s and 10m/s. It can be seen that there is a difference between the strategies in terms of energy capture. This is due to turbines entering recovery mode due to high wind speeds in the 22 m/s and 24 m/s simulations. Recovery mode is used in this way as the wind turbines models do not have a high speed cut out programmed into their control system so the turbulence flag is used for this instead. Because the turbulence flag is used the turbines are kept in recovery from when the turbulence flag is triggered until the end of the simulation. Due to how the turbines are curtailed the turbines trigger the turbulence flag at different times resulting in differences in energy capture in high wind speed simulations.

Set-Points	Curtailment Strategy	Weibull Mean		
		8m/s	9m/s	10m/s
Upper Set-Points	Equal Curtailment	- 8.1285 %	- 8.4598 %	- 8.6305 %
	Power Estimation Curtailment	- 8.13 %	- 8.4611 %	- 8.6316 %
	Intelligent Tower Curtailment	- 8.1388 %	- 8.4693 %	- 8.6391 %
Middle Set-Points	Equal Curtailment	- 15.6065 %	- 16.5388 %	- 17.1119 %
	Power Estimation Curtailment	- 15.6075 %	- 16.5398 %	- 17.1127 %
	Intelligent Tower Curtailment	- 15.6082 %	- 16.5404 %	- 17.1132 %
Lower Set-Points	Equal Curtailment	- 23.2523 %	- 24.7675 %	- 25.7268 %
	Power Estimation Curtailment	- 23.2543 %	- 24.7693 %	- 25.7284 %
	Intelligent Tower Curtailment	- 23.2516 %	- 24.7669 %	- 25.7262 %

Table 6.6: Change in energy capture by control strategy and wind speed compared to not curtailing with the 22m/s and 24m/s bins ignored.

Table (6.6) shows the energy capture for each of the three strategies over Weibull distribu-

Chapter 6. Designing a Wind Farm Controller to Reduce Wind Turbine Fatigue

tions with mean wind speeds of 8m/s, 9m/s and 10m/s with the 22m/s and 24 m/s bins ignored. The table shows that there is very little difference in energy capture between the strategies when the high wind speed simulations are not considered.

6.7.5 Turbine Level Results

For an equally distributed wind rose there is very little difference in the DELs between the different turbines in the wind farm. To demonstrate this the turbine level averages are presented here.

Set-Points	Curtailment Allocation	8m/s			9m/s			10m/s		
		Edge	Center	Corner	Edge	Center	Corner	Edge	Center	corner
Upper	Equal Curtailment	-6.08%	-6.28%	-6.02%	-6.15%	-6.35%	-6.07%	-5.90%	-6.10%	-5.81%
	Power Estimation Curtailment	-7.07%	-7.15%	-7.14%	-6.98%	-7.07%	-7.02%	-6.59%	-6.69%	-6.60%
	Intelligent Tower Curtailment	-11.09%	-11.53%	-11.34%	-11.61%	-12.04%	-11.80%	-11.49%	-11.90%	-11.61%
Middle	Equal Curtailment	-10.41%	-10.72%	-10.14%	-10.67%	-10.98%	-10.38%	-10.41%	-10.70%	-10.11%
	Power Estimation Curtailment	-11.71%	-11.90%	-11.56%	-11.78%	-11.99%	-11.60%	-11.34%	-11.54%	-11.14%
	Intelligent Tower Curtailment	-15.82%	-15.94%	-15.73%	-16.43%	-16.59%	-16.29%	-16.18%	-16.36%	-15.99%
Lower	Equal Curtailment	-13.82%	-14.26%	-13.57%	-14.21%	-14.65%	-13.92%	-13.92%	-14.34%	-13.61%
	Power Estimation Curtailment	-14.71%	-15.01%	-14.52%	-14.99%	-15.29%	-14.74%	-14.58%	-14.88%	-14.30%
	Intelligent Tower Curtailment	-18.06%	-18.25%	-17.87%	-18.73%	-18.95%	-18.48%	-18.44%	-18.68%	-18.15%

Table 6.7: Change in combined tower DELs by control strategy and wind speed at the turbine level.

Table (6.7) shows the change in combined tower DELs for each of the three strategies over Weibull distributions with mean wind speeds of 8m/s 9m/s and 10m/s at the turbine level. The table shows that across all three mean wind speeds and curtailment levels that there is little variation between the different turbines.

Set-Points	Curtailment Allocation	8m/s			9m/s			10m/s		
		Edge	Center	Corner	Edge	Center	Corner	Edge	Center	corner
Upper	Equal Curtailment	-4.34%	-4.37%	-4.04%	-4.61%	-4.66%	-4.33%	-4.73%	-4.77%	-4.46%
	Power Estimation Curtailment	-5.66%	-5.69%	-5.42%	-5.82%	-5.86%	-5.59%	-5.81%	-5.86%	-5.60%
	Intelligent Tower Curtailment	-11.08%	-11.29%	-10.99%	-12.05%	-12.27%	-11.95%	-12.52%	-12.74%	-12.42%
Middle	Equal Curtailment	-8.65%	-8.69%	-8.15%	-9.26%	-9.31%	-8.79%	-9.59%	-9.64%	-9.15%
	Power Estimation Curtailment	-10.35%	-10.45%	-10.04%	-10.82%	-10.93%	-10.53%	-11.01%	-11.12%	-10.73%
	Intelligent Tower Curtailment	-15.65%	-15.71%	-15.53%	-16.82%	-16.91%	-16.69%	-17.36%	-17.46%	-17.22%
Lower	Equal Curtailment	-12.64%	-12.76%	-12.31%	-13.55%	-13.66%	-13.23%	-14.07%	-14.18%	-13.77%
	Power Estimation Curtailment	-13.83%	-14.01%	-13.46%	-14.65%	-14.83%	-14.30%	-15.07%	-15.25%	-14.75%
	Intelligent Tower Curtailment	-17.73%	-17.76%	-17.43%	-19.06%	-19.09%	-18.76%	-19.71%	-19.76%	-19.42%

Table 6.8: Change in out of plane blade DELs by control strategy and wind speed at the turbine level.

Chapter 6. Designing a Wind Farm Controller to Reduce Wind Turbine Fatigue

Table (6.8) shows the change in the out of plane blade DELs for each of the three strategies over Weibull distributions with mean wind speeds of 8m/s 9m/s and 10m/s at the turbine level. The table shows that across all three mean wind speeds and curtailment levels that there is little variation between the different turbines.

Set-Points	Curtailment Allocation	8m/s			9m/s			10m/s		
		Edge	Center	Corner	Edge	Center	Corner	Edge	Center	corner
Upper	Equal Curtailment	-0.57%	-0.57%	-0.57%	-0.59%	-0.59%	-0.59%	-0.57%	-0.58%	-0.57%
	Power Estimation Curtailment	-0.58%	-0.58%	-0.59%	-0.60%	-0.60%	-0.60%	-0.58%	-0.58%	-0.58%
	Intelligent Tower Curtailment	-0.58%	-0.60%	-0.60%	-0.60%	-0.61%	-0.61%	-0.56%	-0.58%	-0.57%
Middle	Equal Curtailment	-0.83%	-0.83%	-0.83%	-0.86%	-0.86%	-0.86%	-0.82%	-0.83%	-0.82%
	Power Estimation Curtailment	-0.85%	-0.84%	-0.85%	-0.87%	-0.87%	-0.87%	-0.83%	-0.83%	-0.83%
	Intelligent Tower Curtailment	-0.84%	-0.84%	-0.84%	-0.86%	-0.87%	-0.87%	-0.82%	-0.82%	-0.82%
Lower	Equal Curtailment	-1.02%	-1.01%	-1.02%	-1.05%	-1.05%	-1.05%	-0.99%	-0.99%	-0.99%
	Power Estimation Curtailment	-1.03%	-1.02%	-1.03%	-1.06%	-1.06%	-1.06%	-1.00%	-1.00%	-0.99%
	Intelligent Tower Curtailment	-1.02%	-1.02%	-1.02%	-1.04%	-1.05%	-1.04%	-0.98%	-0.98%	-0.98%

Table 6.9: Change in in plane blade DELs by control strategy and wind speed at the turbine level.

Table (6.9) shows the change in the in plane blade DELs for each of the three strategies over Weibull distributions with mean wind speeds of 8m/s 9m/s and 10m/s at the turbine level. The table shows that across all three mean wind speeds and curtailment levels that there is little variation between the different turbines.

Set-Points	Curtailment Allocation	8m/s			9m/s			10m/s		
		Edge	Center	Corner	Edge	Center	Corner	Edge	Center	corner
Upper	Equal Curtailment	-4.07%	-4.30%	-3.70%	-4.07%	-4.31%	-3.71%	-3.80%	-4.04%	-3.46%
	Power Estimation Curtailment	-7.74%	-7.94%	-7.37%	-7.43%	-7.64%	-7.05%	-6.66%	-6.87%	-6.30%
	Intelligent Tower Curtailment	-0.10%	-0.59%	0.35%	1.84%	1.33%	2.31%	3.62%	3.13%	4.08%
Middle	Equal Curtailment	-6.61%	-6.85%	-6.00%	-6.32%	-6.58%	-5.71%	-5.47%	-5.73%	-4.88%
	Power Estimation Curtailment	-11.90%	-12.05%	-11.30%	-11.20%	-11.37%	-10.58%	-9.68%	-9.87%	-9.07%
	Intelligent Tower Curtailment	-0.92%	-1.46%	-0.33%	1.80%	1.21%	2.44%	4.32%	3.72%	4.98%
Lower	Equal Curtailment	-10.58%	-10.95%	-9.94%	-9.76%	-10.15%	-9.10%	-8.09%	-8.49%	-7.43%
	Power Estimation Curtailment	-14.71%	-15.03%	-14.14%	-13.60%	-13.96%	-13.00%	-11.43%	-11.81%	-10.82%
	Intelligent Tower Curtailment	-2.92%	-3.40%	-2.08%	0.11%	-0.42%	1.01%	3.12%	2.58%	4.03%

Table 6.10: Change in hub torque DELs by control strategy and wind speed

Table (6.10) shows the change in the Hub torque DELs for each of the three strategies over Weibull distributions with mean wind speeds of 8m/s 9m/s and 10m/s at the turbine level. The table shows that across all three mean wind speeds and curtailment levels there is little variation between the different turbines.

6.8 Concluding Remarks

This chapter has demonstrated the impact that wind farm control can have on reducing wind turbine fatigue through the consideration of wind farm control techniques. By considering the steady state bending moments of a turbine a curtailment allocation strategy has been developed which prioritizes curtailment near the rated wind speed where the turbine has the highest static tower and bending moments. Through a comprehensive series of simulations considering wind speed, wind direction and turbulence intensity as well as different levels of curtailment, the impact of the proposed intelligent tower curtailment strategy has been found. For comparison this analysis exercise was also performed for two other strategies (equal curtailment, power estimate curtailment).

It is unclear whether the power estimation control strategy or the intelligent tower control strategy is preferable to the other as the intelligent tower control strategy shows greater reductions in both tower and blade DELs which are normally the DELs which are prioritized but shows a no change in hub torque DELs so could result in no reduction to the fatigue of the turbine drive trains. The power estimation control strategy shows consistent reductions across both the tower and blade DELs but also shows reduction in the hub torque DELs.

A possible area of future work could be to test a control strategy similar to the intelligent tower strategy but with a shallower gradient above rated to reduce hub torque variations.

Chapter 7

Development of a dynamic power system model in Strathfarm

This chapter details the development of a power system model for Strathfarm beginning with the inclusion of the offshore power system model developed by [42]. This is developed through a Levenberg-Marquardt type nonlinear solver code for implementing the offshore power system into Strathfarm. The chapter continues with the development of a power system model for providing Strathfarm's wind farm controller with grid frequency data from a power system model to measure the efficacy of the droop controller and inertial response algorithms developed by [90].

7.1 The Model Development Process

Previous research in Strathfarm has looked at how a wind farm can provide droop and inertial frequency response [90] but has done so by feeding grid frequency data into the model. This chapter seeks to develop a power system model which will provide frequency data to the wind farm controller but will respond to changes in power output from the wind farm to demonstrate the efficacy of the strategy.

The power system model presented in this chapter has been developed so that it can accurately model the grid frequency for use in the wind farm controller for demonstrating the efficacy of a wind farm's frequency response.

Chapter 7. Development of a dynamic power system model in Strathfarm

The power system model developed in this chapter has large scope to be improved in future projects and has been designed in such a way that future improvements and repurposing should not require large redesigns of the model.

The model developed in this chapter has been designed in discrete time as it is computationally more efficient than continuous time. Using discrete time also future proofs the model as Strathfarm is planned on being entirely converted to being a discrete time model in C++.

The power system model is designed as a series of DLLs which are included into Simulink through S-functions and linking codes.

The model of the power system assumes a simple continuous grid demand with the supply modelled through a single generator.

The generator model used is a discretised version of a synchronous generator with a governor based on the 15MW model of a diesel generator present in the example Simulink model for the 24-hour Simulation of a Vehicle-to-Grid (V2G) System [80]. The diesel generator has been used as a starting point for the grid model as it can be scaled up to be a size of a whole power system's generation. The diesel generator model also contains the required dynamics for mimicking a grid frequency event for testing the wind farm controller's ancillary service dispatch algorithms. The power system model uses normalised values so it can be scaled by increasing the current output. This may need future development, but at present as only the grid frequency dynamics that impact wind farm control decisions are considered it is sufficient to assume linearity.

7.2 The Solver Code

The first section of the wind farm power system model that needed to be included within Strathfarm is a model of the wind farm power system. This is essential as the interaction between the different turbines in the windfarm can impact the aggregate power that is delivered to the grid. The model that was chosen for this task was the one developed in [42] which models a scalable wind farm connected to a power system represented by a voltage source through a long AC cable.

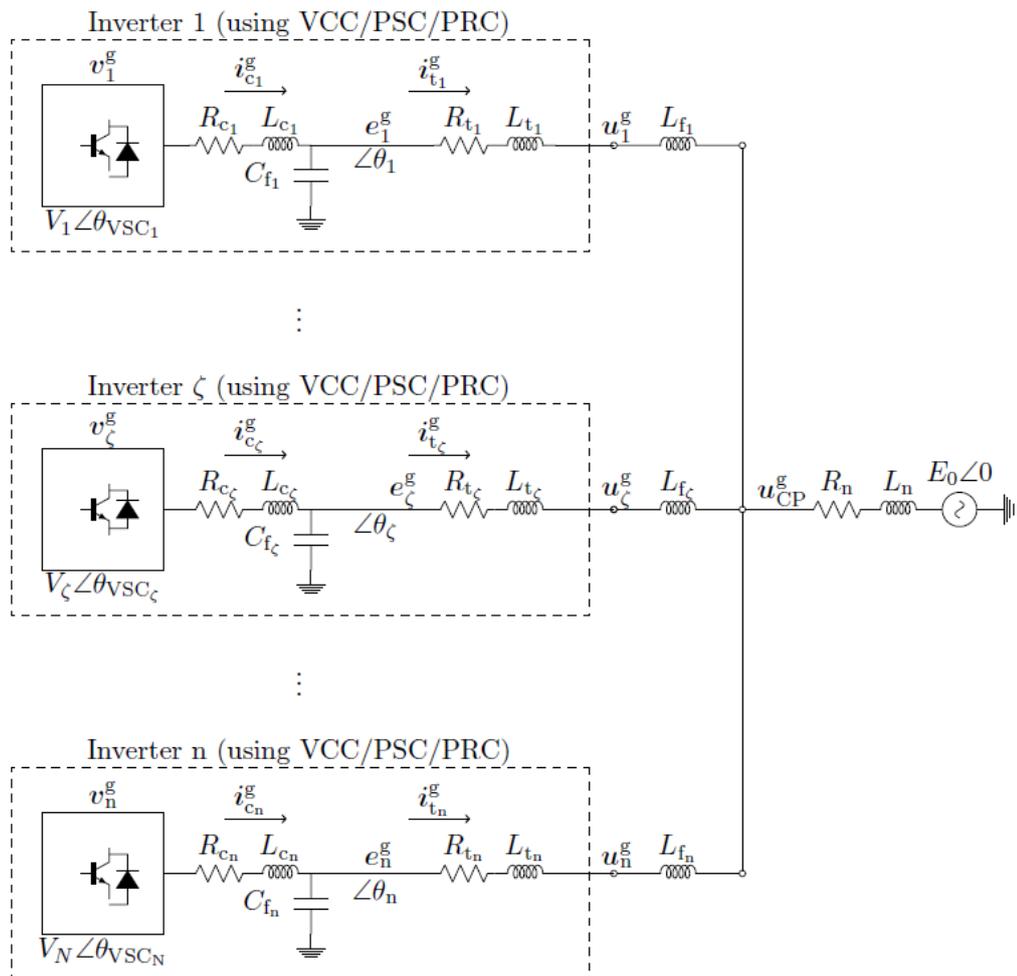


Figure 7.1: Offshore power system model diagram from [42].

Figure (7.1) shows a circuit diagram for an offshore wind farm array as detailed in [42].

Chapter 7. Development of a dynamic power system model in Strathfarm

This wind farm circuit can be represented as a scalable matrix equation by combining Equation 4.39, 4.44 and 4.37 from [42] into a single series of equations. For simplicity in this work the configuration of the converters for each of the wind turbines are identical. The steady state dynamics of an offshore wind farm array connected through an AC cable can be modelled as the following equations,

$$P_{\zeta} = e_{\zeta d}^g i_{\zeta d}^g + e_{\zeta q}^g i_{\zeta q}^g \quad (7.1)$$

where P_{ζ} is the power output from the ζ th turbine, $e_{\zeta d}^g$ and $e_{\zeta q}^g$ are the dq components of the filter bus voltage in the grid dq frame from the ζ th turbine and $i_{\zeta d}^g$ and $i_{\zeta q}^g$ are the dq components of the current flowing to the network for the ζ th turbine.

$$V_{\zeta} = \sqrt{(v_{\zeta d}^g)^2 + (v_{\zeta q}^g)^2} \quad (7.2)$$

where V_{ζ} is the magnitude of the bridge voltage of the ζ th turbine and $v_{\zeta d}^g$ and $v_{\zeta q}^g$ are the dq components of the bridge voltage in the grid dq frame from the ζ th turbine.

The next step is to consider Equation 4.39 from [42] which gives the steady state equations for the offshore power system.

$$0 = -\left(\frac{R_c}{L_c} + \begin{pmatrix} 0 & \omega \\ \omega & 0 \end{pmatrix}\right) \mathbf{i}_{c\zeta} + \frac{1}{L_c} \mathbf{e}_{\zeta}^g + \frac{1}{L_t} \mathbf{v}_{\zeta}^g \quad (7.3)$$

where $\mathbf{i}_{c\zeta}$ is a vector in the dq frame of the form $\begin{pmatrix} i_{c\zeta d}^g \\ i_{c\zeta q}^g \end{pmatrix}$ where $i_{c\zeta d}^g$ and $i_{c\zeta q}^g$ are the d and q components of the current flowing from the ζ th wind turbine, \mathbf{e}_{ζ}^g is a vector in the dq frame of the form $\begin{pmatrix} e_{\zeta d}^g \\ e_{\zeta q}^g \end{pmatrix}$, \mathbf{v}_{ζ}^g is a vector in the dq frame of the form $\begin{pmatrix} v_{\zeta d}^g \\ v_{\zeta q}^g \end{pmatrix}$, ω is the grid frequency, L_t inductance of the transmission line from a turbine to the PCC for the farm. L_c inductive component of the phase reactor of the converters. R_c is the resistive component of the phase reactor of the converters.

$$0 = \frac{1}{C_f} \mathbf{i}_{c\zeta} + \begin{pmatrix} 0 & \omega \\ -\omega & 0 \end{pmatrix} \mathbf{e}_{\zeta}^g - \frac{1}{C_f} \mathbf{i}_{t\zeta} \quad (7.4)$$

where C_f is the capacitance of the converter filters and $\mathbf{i}_{t\zeta}^g$ is a vector in the dq frame of the form $\begin{pmatrix} i_{t\zeta d}^g \\ i_{t\zeta q}^g \end{pmatrix}$,

$$0 = \frac{1}{L_t} \mathbf{e}_{\zeta}^g + \begin{pmatrix} 0 & \omega \\ -\omega & 0 \end{pmatrix} \mathbf{i}_{t\zeta} - \frac{R_t}{L_t} \mathbf{i}_{t\zeta} - \frac{E_0}{L_t} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \frac{R_n}{L_t} \mathbf{i}_{t\zeta} \sum_{\eta=1}^N (\mathbf{i}_{t\eta}) + \frac{L_n}{L_t} \begin{pmatrix} 0 & \omega \\ -\omega & 0 \end{pmatrix} \sum_{\eta=1}^N (\mathbf{i}_{t\eta}) \quad (7.5)$$

where E_0 is the grid voltage, L_n is the inductance of the AC connection from the farm to the grid R_t is the resistance of the transmission line from a turbine to the PCC for the farm and N is the number of wind turbines in the farm.

For ease of calculation the variables in the equation system can be considered as a vector of the form:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ \vdots \\ x_{8(\zeta-1)+1} \\ x_{8(\zeta-1)+2} \\ x_{8(\zeta-1)+3} \\ x_{8(\zeta-1)+4} \\ x_{8(\zeta-1)+5} \\ x_{8(\zeta-1)+6} \\ x_{8(\zeta-1)+7} \\ x_{8(\zeta-1)+8} \\ \vdots \\ x_{8(N-1)+1} \\ x_{8(N-1)+2} \\ x_{8(N-1)+3} \\ x_{8(N-1)+4} \\ x_{8(N-1)+5} \\ x_{8(N-1)+6} \\ x_{8(N-1)+7} \\ x_{8(N-1)+8} \end{pmatrix} = \begin{pmatrix} v_{g1d} \\ v_{g1q} \\ i_{c1d} \\ i_{c1q} \\ e_{1d} \\ e_{1q} \\ i_{t1d} \\ i_{t1q} \\ \vdots \\ v_{g\zeta d} \\ v_{g\zeta q} \\ i_{c\zeta d} \\ i_{c\zeta q} \\ e_{\zeta d} \\ e_{\zeta q} \\ i_{t\zeta d} \\ i_{t\zeta q} \\ \vdots \\ v_{gNd} \\ v_{gNq} \\ i_{cNd} \\ i_{cNq} \\ e_{Nd} \\ e_{Nq} \\ i_{tNd} \\ i_{tNq} \end{pmatrix} \quad (7.6)$$

where the x vector has $8N$ entries where N is the number of wind turbines in the wind farm.

Equations (7.1), (7.2), (7.3), (7.4), and (7.5) can be rearranged to all equal zero and rewritten using the x vector in Equation (7.6) to give:

$$0 = x_{8(\zeta-1)+4}x_{8(\zeta-1)+6} + x_{8(\zeta-1)+5}x_{8(\zeta-1)+7} - P_\zeta \quad (7.7)$$

$$0 = \sqrt{x_{8(\zeta-1)+0}^2 + x_{8(\zeta-1)+1}^2} - V_\zeta \quad (7.8)$$

$$0 = \frac{-R_c}{L_c}x_{8(\zeta-1)+2} - \omega x_{8(\zeta-1)+3} - \frac{x_{8(\zeta-1)+4}}{L_c} + \frac{x_{8(\zeta-1)+0}}{L_t} \quad (7.9)$$

$$0 = \frac{-R_c}{L_c}x_{8(\zeta-1)+3} - \omega x_{8(\zeta-1)+2} - \frac{x_{8(\zeta-1)+5}}{L_c} + \frac{x_{8(\zeta-1)+1}}{L_t} \quad (7.10)$$

$$0 = \frac{x_{8(\zeta-1)+2}}{C_f} + \omega x_{8(\zeta-1)+5} - \frac{x_{8(\zeta-1)+6}}{C_f} \quad (7.11)$$

$$0 = \frac{x_{8(\zeta-1)+3}}{C_f} - \omega x_{8(\zeta-1)+4} - \frac{x_{8(\zeta-1)+7}}{C_f} \quad (7.12)$$

$$0 = \frac{x_{8(\zeta-1)+4}}{L_t} - \frac{R_t}{L_t}x_{8(\zeta-1)+6} + \left(1 + \frac{L_f}{L_t}\right)\omega x_{8(\zeta-1)+7} - \frac{E_0}{L_t} - \frac{R_n}{L_t} \sum_{\eta=1}^N (x_{6+8(\eta-1)}) + \omega \frac{L_n}{L_t} \sum_{\eta=1}^N (x_{7+8(\eta-1)}) \quad (7.13)$$

$$0 = \frac{x_{8(\zeta-1)+5}}{L_t} - \frac{R_t}{L_t}x_{8(\zeta-1)+7} - \left(1 + \frac{L_f}{L_t}\right)\omega x_{8(\zeta-1)+6} - \frac{R_n}{L_t} \sum_{\eta=1}^N (x_{7+8(\eta-1)}) - \omega \frac{L_n}{L_t} \sum_{\eta=1}^N (x_{6+8(\eta-1)}) \quad (7.14)$$

Finally, each of these equations can be set equal to an entry in a vector $F(x) = 0$ where F can be used as a variable to be minimized in an iterative solver:

$$F_{8(\zeta-1)+0} = x_{8(\zeta-1)+4}x_{8(\zeta-1)+6} + x_{8(\zeta-1)+5}x_{8(\zeta-1)+7} - P_\zeta \quad (7.15)$$

$$F_{8(\zeta-1)+1} = \sqrt{x_{8(\zeta-1)+0}^2 + x_{8(\zeta-1)+1}^2} - V_\zeta \quad (7.16)$$

$$F_{8(\zeta-1)+2} = \frac{-R_c}{L_c}x_{8(\zeta-1)+2} - \omega x_{8(\zeta-1)+3} - \frac{x_{8(\zeta-1)+4}}{L_c} + \frac{x_{8(\zeta-1)+0}}{L_t} \quad (7.17)$$

$$F_{8(\zeta-1)+3} = \frac{-R_c}{L_c}x_{8(\zeta-1)+3} - \omega x_{8(\zeta-1)+2} - \frac{x_{8(\zeta-1)+5}}{L_c} + \frac{x_{8(\zeta-1)+1}}{L_t} \quad (7.18)$$

$$F_{8(\zeta-1)+4} = \frac{x_{8(\zeta-1)+2}}{C_f} + \omega x_{8(\zeta-1)+5} - \frac{x_{8(\zeta-1)+6}}{C_f} \quad (7.19)$$

$$F_{8(\zeta-1)+5} = \frac{x_{8(\zeta-1)+3}}{C_f} - \omega x_{8(\zeta-1)+4} - \frac{x_{8(\zeta-1)+7}}{C_f} \quad (7.20)$$

$$F_{8(\zeta-1)+6} = \frac{x_{8(\zeta-1)+4}}{L_t} - \frac{R_t}{L_t} x_{8(\zeta-1)+6} + \left(1 + \frac{L_f}{L_t}\right) \omega x_{8(\zeta-1)+7} - \frac{E_0}{L_t} - \frac{R_n}{L_t} \sum_{\eta=1}^N (x_{6+8(\eta-1)}) + \omega \frac{L_n}{L_t} \sum_{\eta=1}^N (x_{7+8(\eta-1)}) \quad (7.21)$$

$$F_{8(\zeta-1)+7} = \frac{x_{8(\zeta-1)+5}}{L_t} - \frac{R_t}{L_t} x_{8(\zeta-1)+7} - \left(1 + \frac{L_f}{L_t}\right) \omega x_{8(\zeta-1)+6} - \frac{R_n}{L_t} \sum_{\eta=1}^N (x_{7+8(\eta-1)}) - \omega \frac{L_n}{L_t} \sum_{\eta=1}^N (x_{6+8(\eta-1)}) \quad (7.22)$$

As these equations represent a non-linear system of equations a non linear solver algorithm is required for which the first step to developing any solution requires finding a linearised representation for of this system. Hence the Jacobian matrix of the F vector equation system needs to be found. In this version of the power system model the turbine level converter values are assumed to be identical for each turbine. This does not necessarily need to be the case in the model and can be changed in the code if a future project requires but has been chosen here for simplicity.

7.2.1 The Jacobian Matrix

The Jacobian matrix is found by taking the partial derivative of each entry of F vector with respect to an entry in the x vector. As many of the converter level sub matrices equations are only dependent on what is happening at that converter the Jacobian, like the F matrix, does not have many non zero terms away from the diagonal. The matrix is blockdiagonal for all terms except for the 7th and 8th row of the block where the interaction between the converters is included.

The submatrix of the Jacobian excluding cross coupled terms is given as an 8 by 8 matrix

of the form:

$$H_k = \begin{bmatrix} 0 & 0 & 0 & 0 & x_7 & x_8 & x_5 & x_6 \\ \frac{x_1}{\sqrt{x_1^2+x_2^2}} & \frac{x_2}{\sqrt{x_1^2+x_2^2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{L_t} & 0 & \frac{-R_c}{L_c} & \omega_1 & \frac{-1}{L_c} & 0 & 0 & 0 \\ 0 & \frac{1}{L_t} & -\omega_1 & \frac{-R_c}{L_c} & 0 & \frac{-1}{L_c} & 0 & 0 \\ 0 & 0 & \frac{1}{C_f} & 0 & 0 & \omega_1 & \frac{-1}{C_f} & 0 \\ 0 & 0 & 0 & \frac{1}{C_f} & -\omega_1 f & 0 & 0 & \frac{-1}{C_f} \\ 0 & 0 & 0 & 0 & \frac{1}{L_t} & 0 & -R_n & \omega_1 L_n \\ 0 & 0 & 0 & 0 & 0 & 1 & -\omega_1 L_n & -R_n \end{bmatrix} \quad (7.23)$$

The cross coupling terms between the converters are given as 8 by 8 matrices of the form:

$$H_{cc} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{-R_n}{L_t} & \omega_1 \frac{L_n}{L_t} \\ 0 & 0 & 0 & 0 & 0 & 0 & -\omega_1 \frac{-L_n}{L_t} & \frac{-R_n}{L_t} \end{bmatrix} \quad (7.24)$$

Hence the full Jacobian matrix $H_{k_{full}}$ can be written as a scalable 8N by 8N matrix in the form:

$$H_{k_{full}} = \begin{pmatrix} H_{k_1} & H_{cc_{1,2}} & H_{cc_{1,3}} & \dots & \dots & H_{cc_{1,N}} \\ H_{cc_{2,1}} & H_{k_2} & \dots & \dots & \dots & \vdots \\ \vdots & \vdots & \cdot & & & \vdots \\ \vdots & \vdots & & \cdot & & \vdots \\ \vdots & \vdots & & & \cdot & \vdots \\ H_{cc_{N,1}} & \dots & \dots & \dots & \dots & H_{k_N} \end{pmatrix} \quad (7.25)$$

where N is the number of wind turbines in the farm. With the Jacobian matrix of the F vector equation system found the non-linear solver algorithm can be developed.

7.2.2 The Levenberg-Marquardt Method

As this system of equations is non-linear, a non-linear solver algorithm needs to be used for a solution to be found. For the implementation of this solver in Strathfarm, Levenberg–Marquardt was chosen as the type of solver to be used. This was done for two reasons firstly as the prior work in this area had used the 'fsolve' command within MATLAB which uses Levenberg–Marquardt as its underlying calculation process. Secondly, as the behaviour of the system wasn't known Levenberg–Marquardt was chosen as it is a compromise between a Gauss-Newton approach and a gradient descent approach with how close it is to either approach determined by a tuning variable. However, through the testing of the model it was found that the optimal solution is to have the tuning variable μ to be as low as possible. Hence the algorithm behaviour is very similar to a Gauss-Newton style non-linear solver algorithm. A general form of the Levenberg–Marquardt method is detailed in [60] which gives the following equations which will be the form used here.

Firstly assuming a system of equations of the form:

$$F(x) = 0. \quad (7.26)$$

The Levenberg–Marquardt method next requires that an optimization function is considered of the form:

$$\min(f(x)) \text{ s : } t : x \in X \quad (7.27)$$

where:

$$f(x) := \|F(x)\|^2 \quad (7.28)$$

is the natural merit function corresponding to the mapping of F [60].

Assuming an iterative Levenberg–Marquardt method type solution, and optimization system of the form:

$$\min \theta^k(d) \text{ s : } t : (x^k + d) \in X \quad (7.29)$$

with the objective function

$$\theta^k(d) := \|F(x^k) + H_k d\|^2 + \mu_k \|d\|^2; \quad (7.30)$$

where x^k is the k th iteration of x , d is the incremental change between iterations of x^k (i.e. that $x^{k+1} = x^k + d$), $H_k d$ is the Jacobian of $F(x^k)$ and $\mu_k := \mu \|F(x_k)\|^2$ where $\mu > 0$.

As [60] state that Equation(7.30) is a strictly convex quadratic function it can be solved by differentiating with respect to d . This is inherently strictly convex as the gradient of the system will monotonically decrease as the solver approach the solution.

By taking the partial derivative of this system with respect to each entry in the x vector at the minimum of the function a linear series of equations can be found.

This set of linear equations can then be solved using a linear solver algorithm to find the solution for the iteration of the solver.

7.2.3 Non-linear Solver Algorithm

As discussed above the series of equations used in this part of the power system model are non-linear so a non-linear solver algorithm must be used. The Levenberg–Marquardt approach has been chosen for this.

The first term of Equation (7.30) can be considered as:

$$\|F(x^k) + H_k d\|^2 = \sum_{i=1}^N (F(i) + HD(i))^2 \quad (7.31)$$

As it is known that the derivative with respect to d is going to be taken of Equation (7.31) any zero order d terms can be ignored as the derivative will eliminate them. Therefore Equation (7.31) can be reduced by eliminating any zero order terms and written as

$$\|F(x^k) + H_k d\|^2 = \sum_{i=1}^N (2F(i)HD(i) + HD(i)^2) + \mathcal{O}(0) \quad (7.32)$$

For the second term of Equation (7.30) we can define μ_{all} to represent the derivative for calcu-

lations in the solver:

$$\mu_{all} = \mu \sum_{i=1}^N (F(x_i)^2) \quad (7.33)$$

These last two equations form the single equation for the objective function which is of the form of a single second order equation in terms of each entry with in d .

The next step is to consider the differential of this quadratic equation which can be considered as a series of equations in matrix form.

Here the terms can be split into:

$$T_{j,k} = \sum_{i=1}^N H_{i,k} H_{i,j} + 2\delta_{j,k} \mu_{all} \quad (7.34)$$

where $\delta_{j,k}$ is a Kronecker delta.

$$T2_i = \sum_{j=1}^N F(j) H_{i,j} \quad (7.35)$$

$Td = -T_2$ is a linear series of equations so can be solved for d through Gaussian elimination. Gaussian elimination has been chosen for this purpose as it is the most simple algorithm for solving a non-singular system of linear equations. This is a key area that the model could be made more computationally efficient in the future as more sophisticated approaches could be used that would require significantly fewer computational steps. This was unfortunately beyond the scope of this project to investigate.

Once the linear system of equations is solved the iteration is completed by finding the next iteration of x^k by solving for d and incrementing:

$$x^{k+1} = x^k + d \quad (7.36)$$

The final step of the code is a check on the error in the solution, if this exceeds the tolerance the code is run again until the max iter limit is reached or a solution is found.

The iterative solver presented here provides results almost identical to the pseudocode presented in [42] but is computationally faster due to not relying on the inbuilt Matlab function 'Fsolve'.

7.3 The Solver S-function in Strathfarm

For implementation in Strathfarm the power system is included into the Simulink model as a series of interacting S-Functions which call external DLLs written in C++. each of these external codes reads in external text files which allow for tuning without the need to recompile the DLLs with new hard coded variables.

This code calculates the combined current flow from a wind farm to the power grid through a long AC cable. Due to the interaction between the turbines in the wind farm this is not as simple as summing the current from each wind turbine. Due to this system being non linear this code requires a linearisation step before it can be solved using a linear solver method.

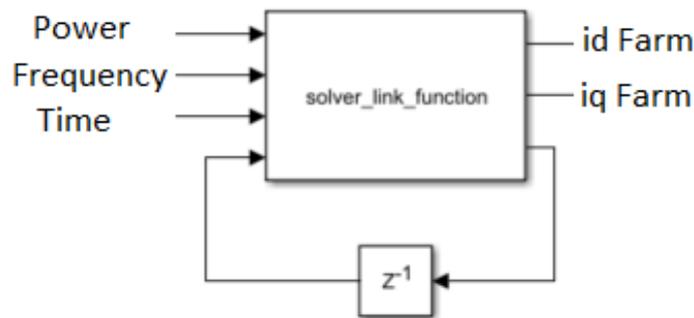


Figure 7.2: Solver block from the Power system model.

Figure (7.2) shows the S-function block containing the solver code in Strathfarm. The inputs of this S-Function block are:

Inputs for the solver	
Variable name	description
Power	The turbine level power for each turbine in the wind farm
Frequency	the normalized grid frequency from power system model
Time	The simulation time
previous timestep solution	The solution of the previous timestep

Table 7.1: The inputs of the solver S-Function block in Strathfarm

Table(7.1) lists and describes the inputs to the solver S-Function block in Strathfarm.

outputs from the solver	
Variable name	description
idfarm	the d component of the current from each wind turbine
iqfarm	the q component of the current from each wind turbine.
previous timestep solution	The solution of the previous timestep

Table 7.2: The outputs of the solver S-Function block in Strathfarm

Table(7.2) lists and describes the inputs of the solver S-Function block in Strathfarm. In this code the previous matrix solution is fed back into the block for the next iteration as a starting solution for the next iteration of the solver to reduce the number of iterations for the next time step. This normally means that the solver only has to perform a single iteration per model time step.

7.3.1 Example Solver Results

The graph below shows the error in the solver as it iterates to a solution for a single turbine model:

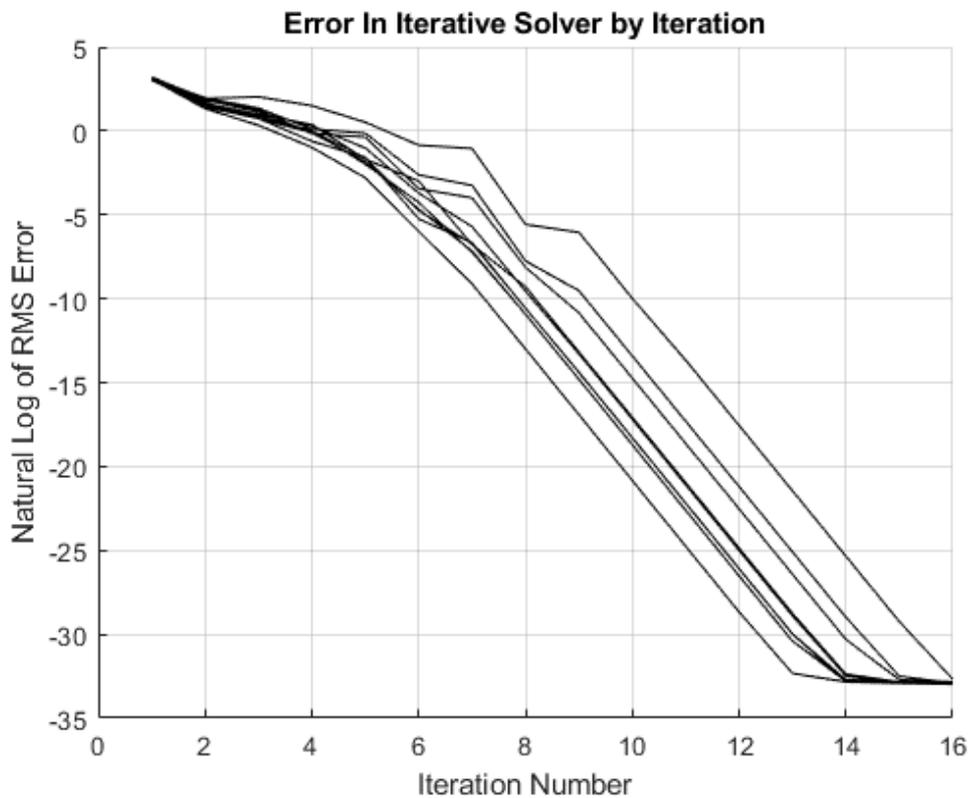


Figure 7.3: An example of the solver error by iteration for a single turbine system.

Figure (7.3) shows an example of the solver's solution error for a single turbine model for 10 randomly generated starting initial vectors.

It should be noted that the y axis is on a log scale. the solutions converge to the same error magnitude each time, this is due to the mathematical limit for the floating point values in MATLAB being reached in the solution. in practice a single iteration is required at most timesteps for a reasonably accurate solution to be found if the previous solution is used as the initial vales for the solver.

In practice as there is diminishing improvement in the level of the accuracy in most cases only a single iteration is required per timestep if the previous solution is used as the initial solution at the next time step.

The solver used in this chapter is guaranteed to converge as the initial solution used is the solution from the previous timestep. Because of this the solver usually only requires a single iteration per timestep to find a solution with a sufficiently low error. It should also be noted

Chapter 7. Development of a dynamic power system model in Strathfarm

that most of the governing equations are very weakly nonlinear so are closely approximated in the linearisation of the system in the Levenberg-Marquardt algorithm.

7.4 The Power System Model

This power system model has been created by discretizing the 15MW diesel generator model in Simulink found in an existing Simulink in-built model [80] and given its design, it has the capability to scale up in power retaining valid power system and governor dynamics. This model in particular was chosen as a starting point as the generator was designed for responding to frequency deviations in a power grid with high levels of variable generation and because the chosen design in the existing model uses a salient pole generator model which is the recommend model for studies in frequency stability [79]. The governor and actuator from the diesel model have been detuned to produce a slower frequency response to any generation change and have been designed to approximate a real grid deviation as seen when a generator disconnects.

The power system model that has been developed in this project is detailed in this section. The power system model has been designed for use in Strathfarm for the purpose of providing the wind farm controller with accurate frequency data so that the provision of providing inertial response and droop control on wind turbines can be studied in a dynamic model. The power system model is divided into three different sections: The mechanical model, the electrical model and the power system linking model with some simple calculations and functions between the blocks. The power system model is based on a single generator lumped approach where the whole grid is modelled as a single generator.

The power system model needs to operate at a higher rate than the rest of Strathfarm. This is because as a result of this the power system model operates with a time step of 0.00005 seconds compared to the rest of Strathfarm which runs with a time step of 0.01 seconds.

7.5 The Model Between the DLL Blocks

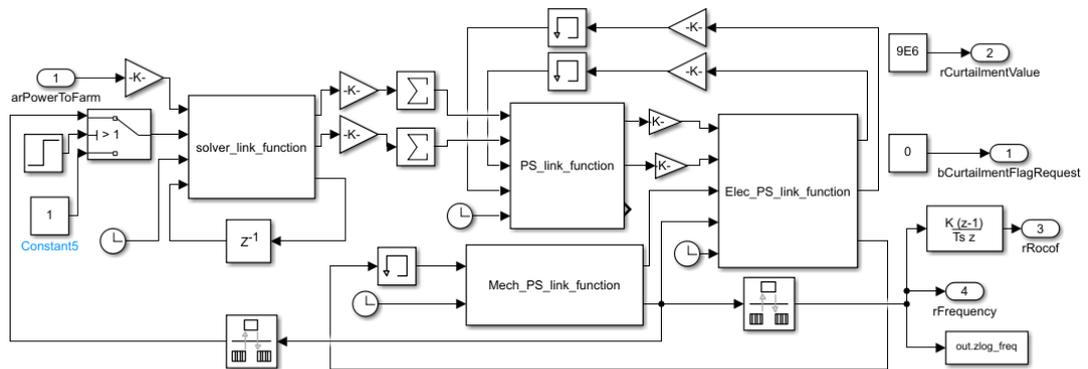


Figure 7.4: The full power system model as it appears in Strathfarm

Figure (7.4) shows the full power system model from Strathfarm as seen in Simulink.

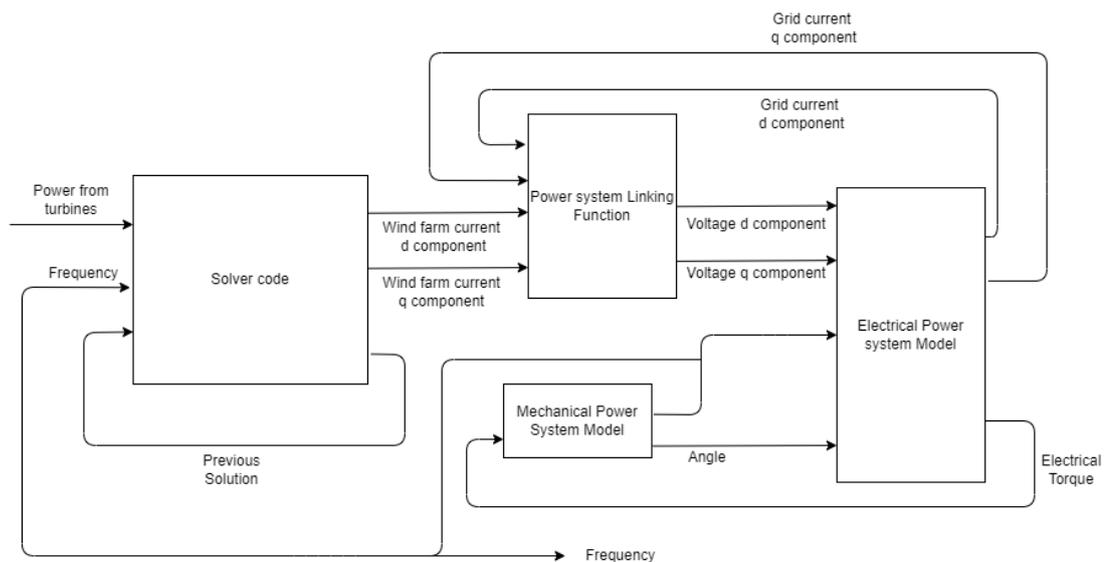


Figure 7.5: A schematic diagram of the power system in Strathfarm

Figure (7.5) shows a schematic of the power system model from Strathfarm .

Some parts of the model are calculated between the blocks in Simulink. These are just gains and sums that allow for scalability in wind farm size and of the size of the power generation from the grid. In future these values should be moved to a more user friendly place.

In order to scale the model the gain on the first two outputs of the power system electrical

Chapter 7. Development of a dynamic power system model in Strathfarm

block should be increases proportionately to scale the default output of a 15MW output.

The memory blocks (seen in the Simulink model in Figure (7.4)) are required to be included in the model so that Simulink's solver can have an start and end point within the loop that it can calculate the values for each iteration.

The power system model can be broken down into three subsections which correspond to the S-Function blocks seen in the Simulink diagram in Figure (7.4). The three sections are the mechanical generator model, the electrical generator model and the electrical model.

The contents of each of these codes is discussed in this chapter with the full discretization process being found in Appendix B.

7.5.1 The Mechanical Model

The mechanical section of the power system model considers the Swing equation and the speed governor of the generator model.

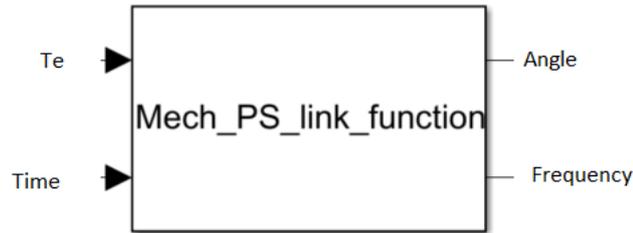


Figure 7.6: The Mechanical model block from the Power system model.

Figure (7.6) shows the S-function Simulink for the mechanical section of the power system model.

Inputs for the solver	
Variable name	description
Te	The electromechanical torque from the electrical model
Time	The simulation time

Table 7.3: The inputs to the mechanical S-Function

Table(7.3) lists and describes the inputs to the mechanical S-Function

outputs from the solver	
Variable name	description
Angle	The angle of the generator
Frequency	the normalized grid frequency from power system model

Table 7.4: The outputs of the mechanical S-Function

Table(7.4) lists and describes the outputs of the mechanical S-function

The Swing Equation

The mechanical part of the power system model is where the swing equation for the grid model is calculated. It takes the electrical torque and simulation time as in puts and outputs a power

angle and the grid's frequency. The swing equation is calculated by finding the incremental change each time step from the equations:

$$dw_1 = \left(\frac{P_m[n-1]}{\omega[n-1]} - T_e \right) \frac{1}{2H} \quad (7.37)$$

where P_m is the mechanical power of the generator, T_e is the electrical torque, H is the inertia of the power system and dw_1 is the incremental change to the rotational speed.

$$dw = \left(\frac{T_s * dw_1[n] + T_s * dw_1[n-1]}{2} \right) + dw[n-1] \quad (7.38)$$

where dw is the deviation of the speed from the synchronous speed.

$$\omega = dw + 1 \quad (7.39)$$

The code has two sections the first is the swing equation the second is the speed controller for the governor.

The Governor

The governor model used in the power system model is based on a diesel generator governor system of the form seen in [56]. This approach was chosen as a diesel generator governor system is very simple mathematically hence is computationally inexpensive while also providing a mechanical power to the generator that will make the generator return its operational speed to the reference level. If the parameters of the diesel generator governor system are changed this can result in the speed of the generator approximately behaving in a similar way to a power system response to an inertial frequency event.

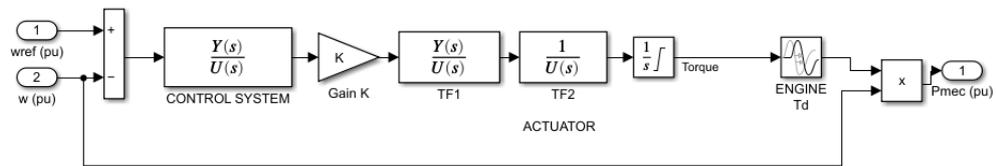


Figure 7.7: The contents of the governor block in the diesel generator system from [80].

Figure (7.7) shows the governor model in Simulink. It can be seen that this is a model of a governor system consisting of a controller and actuators. This has been discretized fully in appendix B. The only major deviation from the original approach is that the time delay has been removed from the discretized model as it was causing instability. When used for droop control and inertial response modelling within Strathfarm the response is detuned so that the frequency of the modelled grid follows a more realistic profile. This is done by modifying the values of the actuators in the governor controller so that it mimics the time delay of the unmodified governor model, resulting in a grid frequency deviation similar to those seen in power grids.

7.6 The Electrical Model

The Electrical model S-function consists of the section of the generator model that are electrical by nature such as the calculation of the electromagnetic torque and the excitation control system,

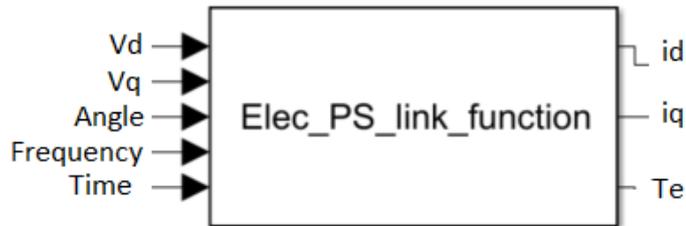


Figure 7.8: The electrical model block from the power system model.

Figure (7.8) shows the S-function block in Simulink for the electrical model code of the power system model

Inputs for the electrical S-Function	
Variable name	description
vd	The voltage of the power system
vq	The voltage of the power system
Angle	The angle of the generator
Frequency	the normalized grid frequency from power system model
Time	The simulation time

Table 7.5: The inputs to the electrical S-Function in the power system model.

Table (7.5) lists and describes the inputs to the electrical S-Function in the power system model in Strathfarm.

outputs from the electrical S-Function	
Variable name	description
id	the current of the power system
iq	the current of the power system
Te	The electromechanical Torque from the electrical model

Table 7.6: the outputs of the electrical S-Function in the power system model.

Table (7.6) lists and describes the outputs of the electrical S-Function in the power system model in Strathfarm.

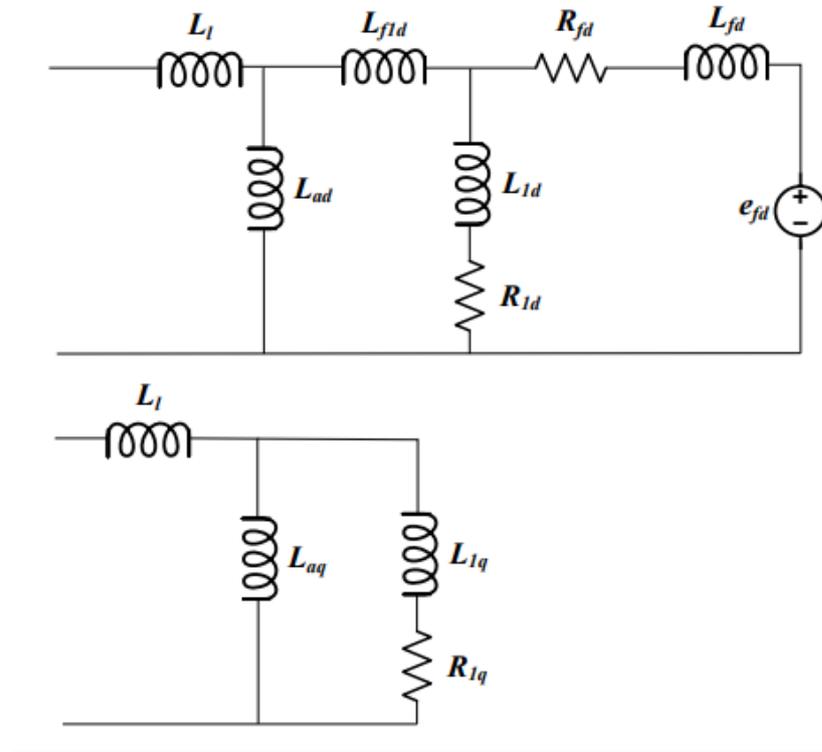


Figure 7.9: An example circuit from [79] showing the generator circuits

Figure (7.9) shows the circuit diagram of the generator model used in Strathfarm. This configuration has been chosen for this application as the standards [79] recommend it use for models containing a salient pole generator. A salient pole generator was used as the basis for this model as they have higher levels of inertia than round pole generators hence are better for modelling grid frequency.

The governing equations for the generator voltages in continuous time for the system in dq0 terms are:

$$V_d = -i_d R_s - \omega \psi_q + \frac{d\psi_d}{dt} \quad (7.40)$$

$$V_q = -i_q R_s - \omega \psi_d + \frac{d\psi_q}{dt} \quad (7.41)$$

$$V_{fd} = \frac{d\psi_{fd}}{dt} + R_{fd} i_{fd} \quad (7.42)$$

$$0 = \frac{d\psi_{kd}}{dt} + R_{kd} i_{kd} \quad (7.43)$$

$$0 = \frac{d\psi_{kq}}{dt} + R_{kq}i_{kq} \quad (7.44)$$

where:

- V_d is the d component of the voltage,
- V_q is the q component of the voltage,
- V_{fd} is the voltage of the field winding,
- ψ_d is the d component of the flux linkage,
- ψ_q is the q component of the flux linkage,
- ψ_{fd} is the flux linkage of the field winding,
- ψ_{fk} is the d component of the flux linkage of the damper winding,
- ψ_{kq} is the q component of the flux linkage of the damper winding,
- i_d is the d component of the current,
- i_q is the q component of the current,
- i_{fd} is the current of the field winding,
- i_{fk} is the d component of the current of the damper winding,
- i_{kq} is the q component of the current of the damper winding,
- R_s is the resistance of the stator,
- R_{fd} is the resistance of the field winding,
- R_{kd} is the d component of the resistance of the damper winding,
- R_{kq} is the q component of the resistance of the damper winding.

The governing equations for the flux linkages can be written in the matrix form:

$$\begin{pmatrix} \psi_q \\ \psi_d \\ \psi_{fd} \\ \psi_{kd} \\ \psi_{fq} \end{pmatrix} = \begin{pmatrix} L_q & 0 & 0 & 0 & L_{akq} \\ 0 & L_d & L_{afd} & L_{akd} & 0 \\ 0 & L_{afd} & L_{ffd} & L_{fkd} & 0 \\ 0 & L_{akd} & L_{fkd} & L_{kkd} & 0 \\ L_{akq} & 0 & 0 & 0 & L_{kkq} \end{pmatrix} \begin{pmatrix} -i_q \\ -i_d \\ i_{fd} \\ i_{kd} \\ i_{fq} \end{pmatrix} \quad (7.45)$$

For this subsection the following terms are used: ψ is the flux linkage and V is the voltage vector entries. The input to the block is the voltage from the node and the grid frequency from the mechanical model the outputs from the model are lumped generator's current in dq and the electrical torque. The model does not include saturation effects.

Modelling the generator in discrete time can be done through the following system of equations:

$$\psi[n] = A_d \psi[n-1] + B_d * \frac{1}{2} (\mathbf{V}[n] + \mathbf{V}[n-1]) \quad (7.46)$$

Where:

$$A_d = \left(I - \frac{\omega_{base} T_s A}{2} \right)^{-1} \left(I + \frac{\omega_{base} T_s A}{2} \right) \quad (7.47)$$

$$B_d = \left(I - \frac{\omega_{base} T_s A}{2} \right)^{-1} \left(\frac{\omega_{base} T_s}{2} \right) \quad (7.48)$$

Where I is a 5-by-5 identity matrix, ω_{base} is the set-point rotational frequency, T_s is the timestep,

$$V = \begin{pmatrix} V_q \\ V_d \\ V_{fd} \\ 0 \\ 0 \end{pmatrix},$$

$$A = RL^{-1} + \begin{pmatrix} 0 & -\omega & 0 & 0 & 0 \\ \omega & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$R = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & R_s & 0 & 0 & 0 \\ 0 & 0 & R_f & 0 & 0 \\ 0 & 0 & 0 & R_{kd} & 0 \\ 0 & 0 & 0 & 0 & R_{kq} \end{pmatrix} \text{ and}$$

$$L = \begin{pmatrix} L_q & 0 & 0 & 0 & L_{akq} \\ 0 & L_d & L_{afd} & L_{akd} & 0 \\ 0 & L_{afd} & L_{ffd} & L_{fkd} & 0 \\ 0 & L_{akd} & L_{fkd} & L_{kkd} & 0 \\ L_{akq} & 0 & 0 & 0 & L_{kkq} \end{pmatrix}.$$

The equations are then concluded with the calculation of the electrical torque T_e :

$$T_e = \psi_d i_q - \psi_q i_d \quad (7.49)$$

To improve the computational efficiency of this S-function some of the calculations are not run if the per unit frequency is approximately equal to 1. This is done as recalculating the A_d and B_d matrices requires inverting matrices which is computationally expensive. However as in most cases the grid frequency is approximately equal to 1 this solution is pre-calculated to skip rerunning these calculations.

7.6.1 The Excitation System

The excitation system is based on the MATLAB model [80] for which the exciter is on based IEEE standards [5]. In this case a type 1 excitation system has been used as it is the historical default for most generator modelling [92]. As with the other sections of the model it has been

7.7 Power System Linking Model

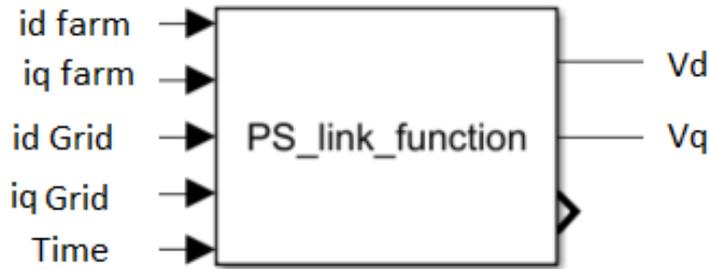


Figure 7.11: The power system block from the power system model

Figure (7.11) shows the S-function Simulink for the linking function section of the power system model.

Inputs for the solver	
Variable name	description
id farm	the current from the wind farm
iq farm	the current from the wind farm
id grid	the current of the power system
iq grid	the current of the power system
Time	The simulation time

Table 7.7: The inputs to the linking S-Function in Strathfarm.

Table (7.7) lists and describes the inputs to the linking function S-Function in Strathfarm.

outputs from the solver	
Variable name	description
vd	the voltage of the power system
vq	the voltage of the power system

Table 7.8: The outputs from the linking S-function in Strathfarm.

Table (7.7) lists and describes the outputs of the linking function S-function in Strathfarm.

This code is the remaining parts of the power system model in Strathfarm. it is a simple code compared with the others modelling the constant grid power demand as a series of simple electrical equations.

As [18] demonstrates a power system can be effectively modelled as a mass spring damper system using the dynamic node technique. The idea behind the dynamic node technique is that a connection point of three different lines can be modelled as a capacitive node with a very low capacitance. For implementation in this model a very small resistance has been added to the node to improve model stability particularly at startup.

Using the dynamic node technique the power system can be modelled as a capacitive node at the connection point of the wind farm, the generator and the grid's load and a load modelled as an impedance.

Reference [18] shows the that a node can be modelled in the form shown in Figure (7.12).

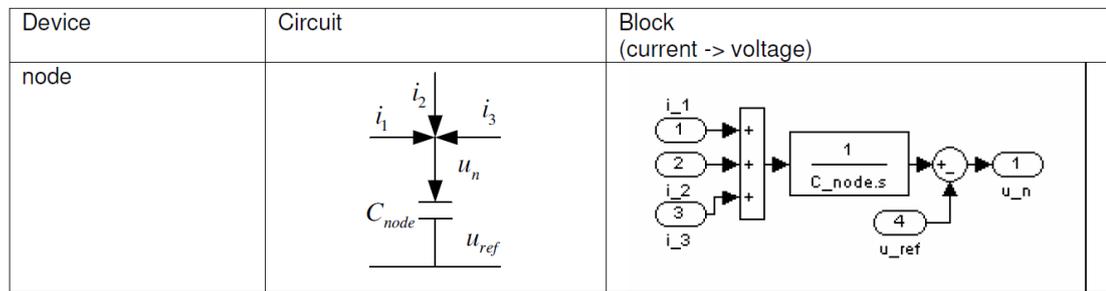


Figure 7.12: Diagram of Dynamic node from [18].

However, an extremely small parallel resistance, of order of 10^{-9} , is added into the system to improve stability. This acts as a damper in the system and is necessary to ensure the model runs as other wise transient behaviour can make the node unstable when discretized.

This essentially operates as a spring with a very small damping effect and is modelled by the equations:

$$v_d[m] = \frac{i_{d_{sum}} - R_{node} * i_{d_{sum}}[m - 1]}{(K_T * L_{node})} + v_d[m - 1] \quad (7.50)$$

$$v_q[m] = \frac{i_{q_{sum}} - R_{node} * i_{q_{sum}}[m - 1]}{(K_T * L_{node})} + v_q[m - 1] \quad (7.51)$$

where L_{node} is the capacitance of the node, R_{node} is the resistance of the node, $i_{d_{sum}}$ and $i_{q_{sum}}$ are the d and q components of the sum of the currents from the wind farm, the power system model and flowing to the ground and v_d and v_q are the d and q components of the voltage at the node.

The model assumes that a constant power is being taken out of the grid at all times and is

Chapter 7. Development of a dynamic power system model in Strathfarm

included in the model as an impedance of magnitude:

$$Z_{mag} = \frac{V_{base}^2}{P_{demand}} \quad (7.52)$$

where P_{demand} is the modelled power demand in the grid and V_{base} is the voltage of the power system. Hence the output of the demand section of the grid has the current equations:

$$i_{2d} = \frac{v_d}{Z_{mag}} \quad (7.53)$$

$$i_{q2} = \frac{v_q}{Z_{mag}} \quad (7.54)$$

where i_{2d} and i_{2q} are the d and q components of the current flowing to the ground.

This is equivalent to removing a set amount of power from the grid as a load at the angle that the grid is operating at. While this approach is very simplistic it is sufficient for the purposes of this model.

7.8 The Wind Farm Controller

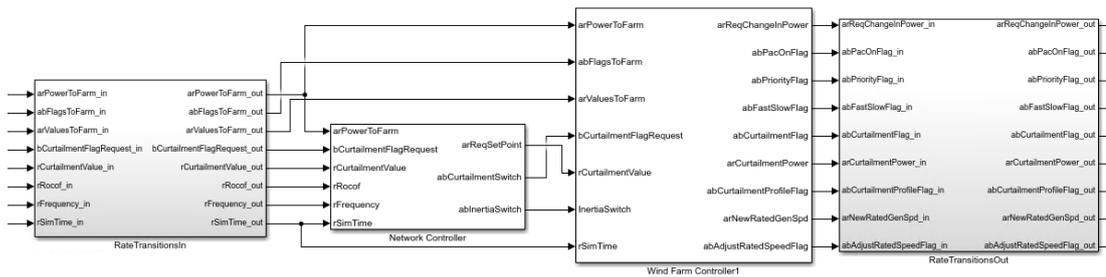


Figure 7.13: The wind farm controller blocks in Strathfarm.

Figure (7.13) shows the wind farm controller blocks within Strathfarm. The controller is split into two different codes: a network controller and a dispatch controller. The network controller takes in the grid frequency signals from the power systems and decides whether to change the set-point power for the farm or whether inertial response is required. This is based on the approach developed by [90]. The Network controller takes in the grid frequency and baseline power set-point and sends the distribution controller three pieces of information: A flag to say if curtailment is required, a flag to say if inertial response is require and a set-point power output to curtail the farm to. This set-point power output is not always the same as the baseline sent into the network controller as the network controller can modify it in order to provide a droop response. For the model in Strathfarm the frequency used in the controller is calculated in the swing equation of the power system. It should be noted that in the real world the frequency signal would need to be found from the grid and sent to the wind farm. This can be done in several ways such as using a phasor measurement unit which estimates the magnitude and phase angle of the grid or as discussed in chapter four’s review of generator response following a smaller generator could be used to act as a sensor to bypass any effects of the transmission line.

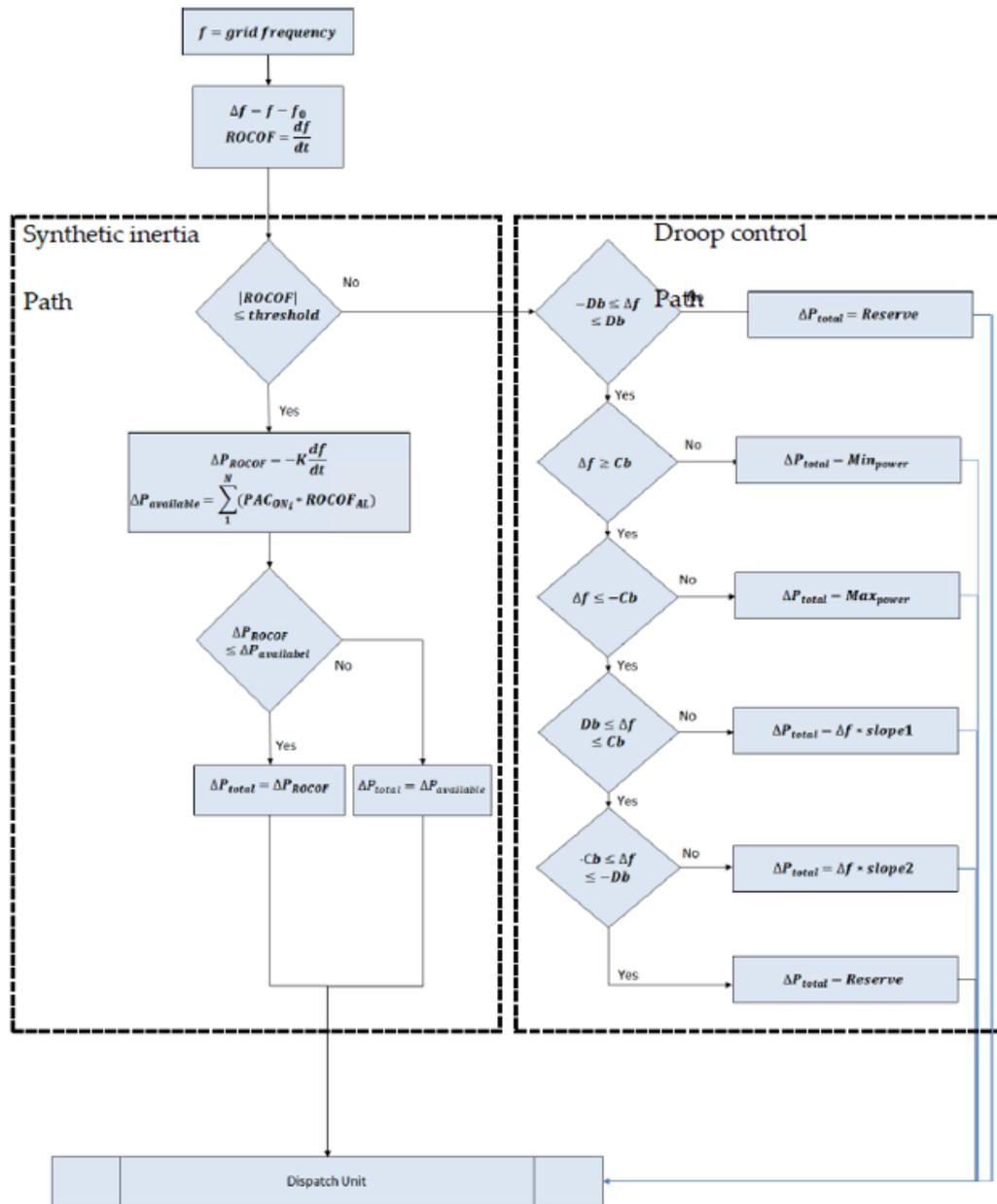


Figure 7.14: The dispatch algorithm used in the inertial response shown in this chapter from [90]

Figure (7.14) shows the dispatch algorithm used in the network controller developed in [90].

The distribution controller is used to allocate any changes in power across the wind turbines

Chapter 7. Development of a dynamic power system model in Strathfarm

in a wind farm. This can be done based on operation information from each wind turbine but for this project the distribution controller equally distributes curtailment to the turbines within the modelled wind farm in order to maintain a farm wide power set-point. Two other distribution approaches exist for the controller but have not been included in the testing for the controller so far. Considering these approaches in inertial and droop response situations is an area of interest for future research.

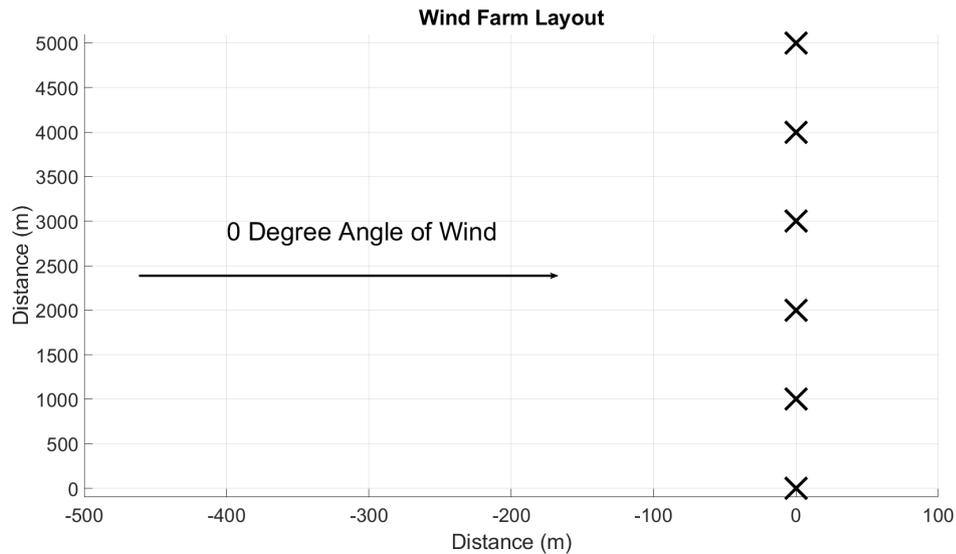


Figure 7.15: Wind Farm Layout for the inertial and droop response results

Figure (7.15) shows the farm layout for the droop controller and inertial response simulations. It has been chosen to be a six turbine farm with a single row of turbines as the power system model is slow to run and this layout removes wake effects from the modelling which also slow the simulation down. Six turbines have been used as this modelling is using wind farm control which requires five active turbines so six turbines have been used in case one of the turbines cannot be used due to either crossing a flag boundary or low wind speed.

7.8.1 Droop Controller Example Results

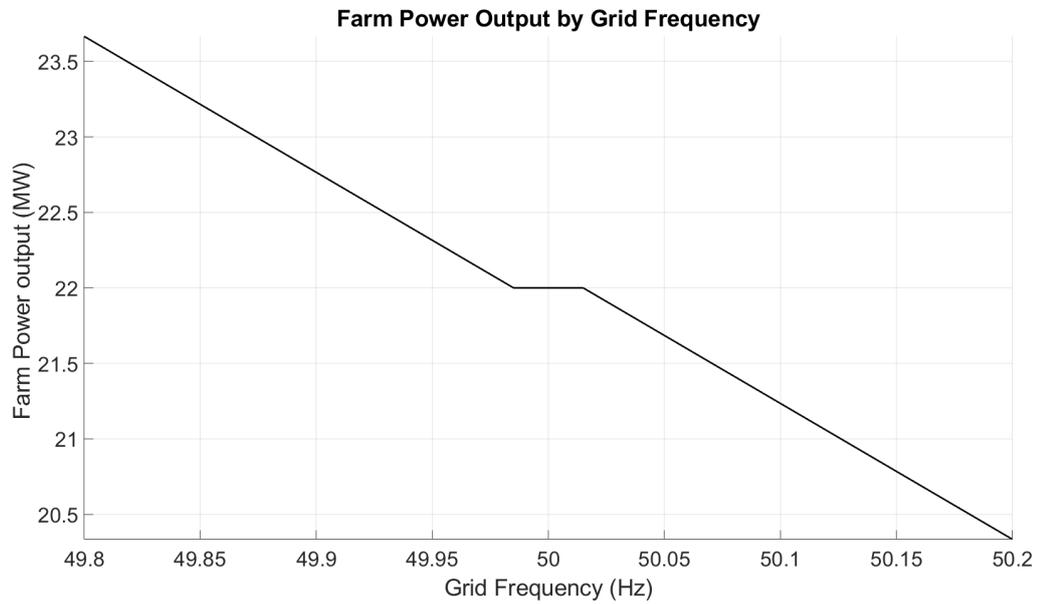


Figure 7.16: The droop curve for the wind farm showing the requested set-point power by the grid frequency.

Figure (7.16) shows the farm level change to the set-point power by level of grid frequency. for this simulation a ramp of 3 has been chosen with a baseline requested set-point power of 22MW.

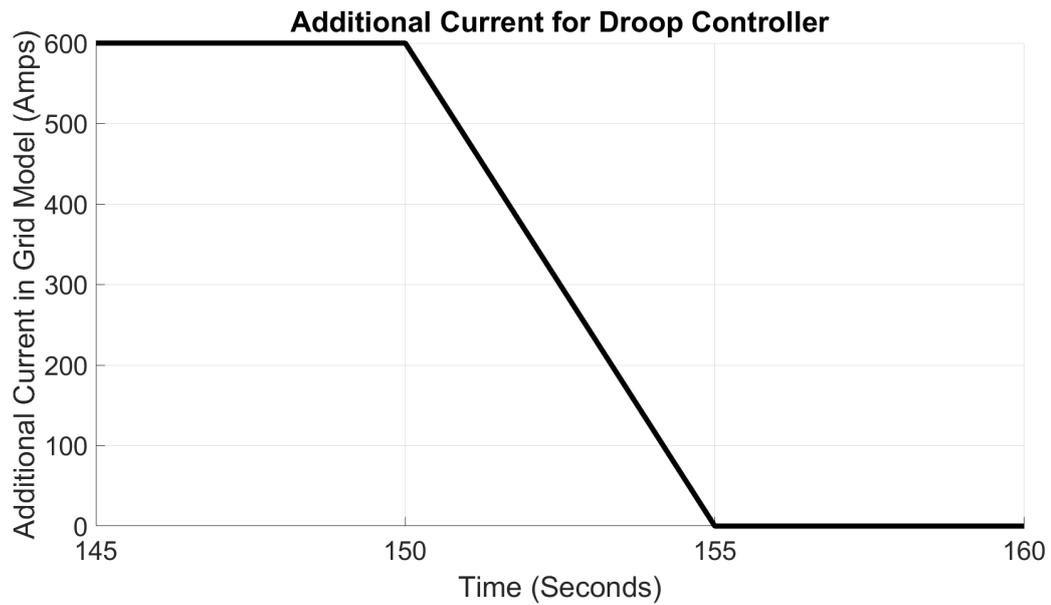


Figure 7.17: The additional current added to the power system model to create a frequency change to activate the droop controller.

Figure (7.17) shows the additional current added to the model in the power system code. A ramp has been used as it changes the frequency enough to demonstrate the efficacy of the droop controller without having a rate of change of frequency (RoCoF) large enough to activate the inertial controller.

The limited results in this chapter are a result of the power system model taking considerable computational resource to operate. This is primarily a result of the turbine converter level solver code which scales, in terms of computational operations, to the order of a cubic of the number of turbines in the farm and is an area where future development work in Strathfarm should be done.

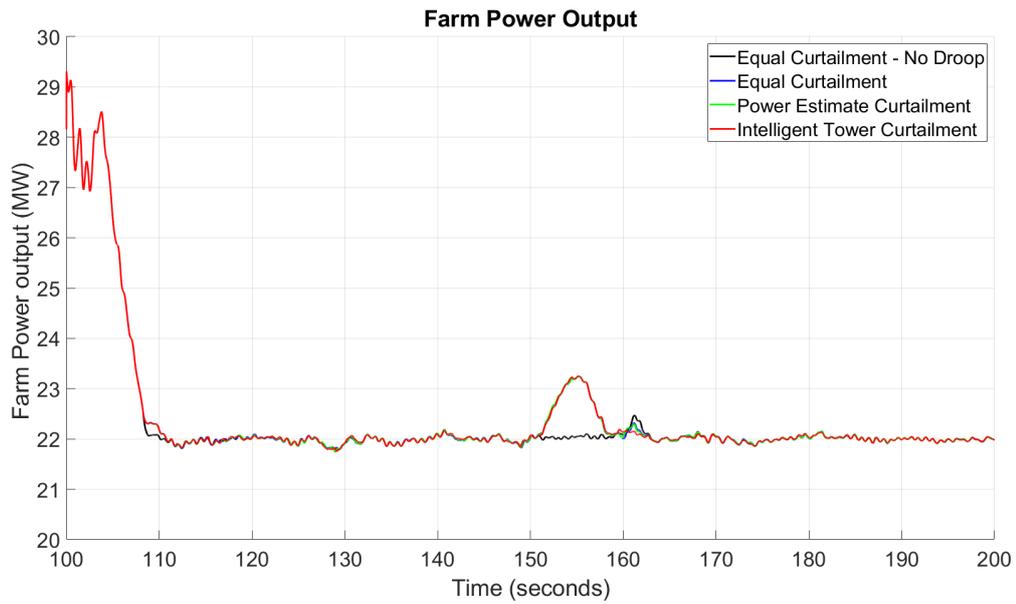


Figure 7.18: The farm wide power output of the wind farm in the droop control stimulations by distribution strategy.

Figure (7.18) shows the farmwide power output for the droop response of the three distribution strategies and when the droop controller is not active. It should be noted that 100 seconds are simulated before the controller is activated to allow transients to dissipate from the startup of the model.

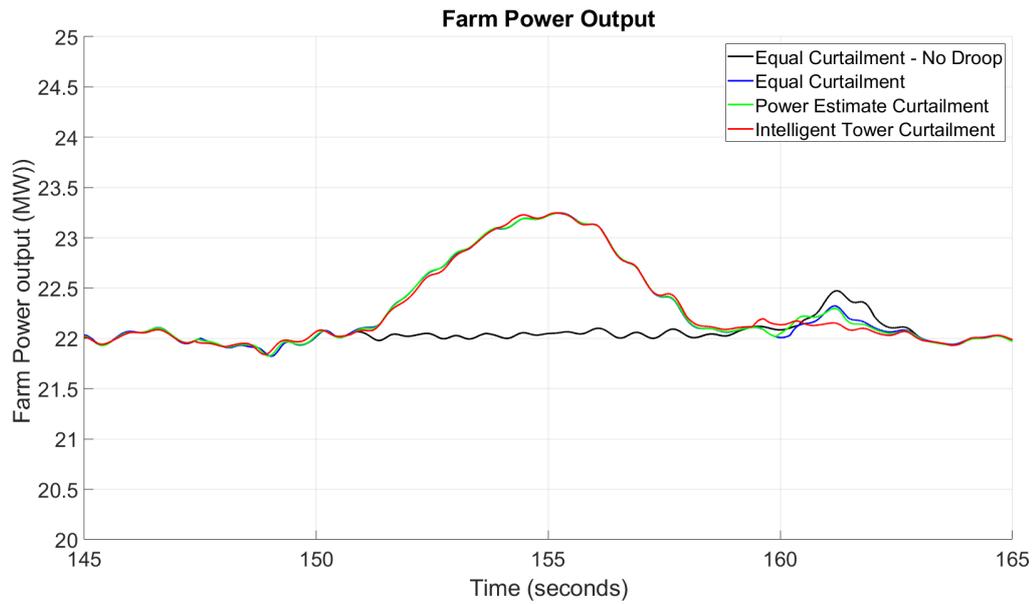


Figure 7.19: The farm wide power output of the wind farm in the droop control stimulations by distribution strategy. at the time of the droop event.

Figure (7.19) shows the farmwide power output for the droop response of the three distribution strategies and when the droop controller is not active at the time of when the droop controller is activated. It can be seen from Figure (7.18) and Figure (7.19) that there is clearly a response to the deviation of grid frequency in the wind farm control system for each of the three strategies.

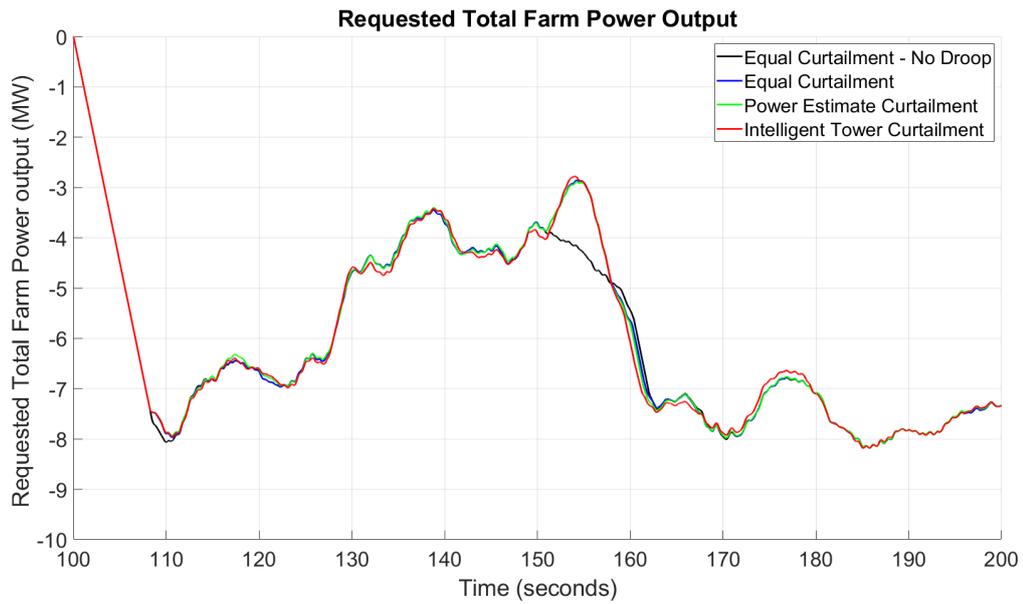


Figure 7.20: The requested change to the wind farm power in the droop simulations for three distribution strategies and when the droop controller is not active

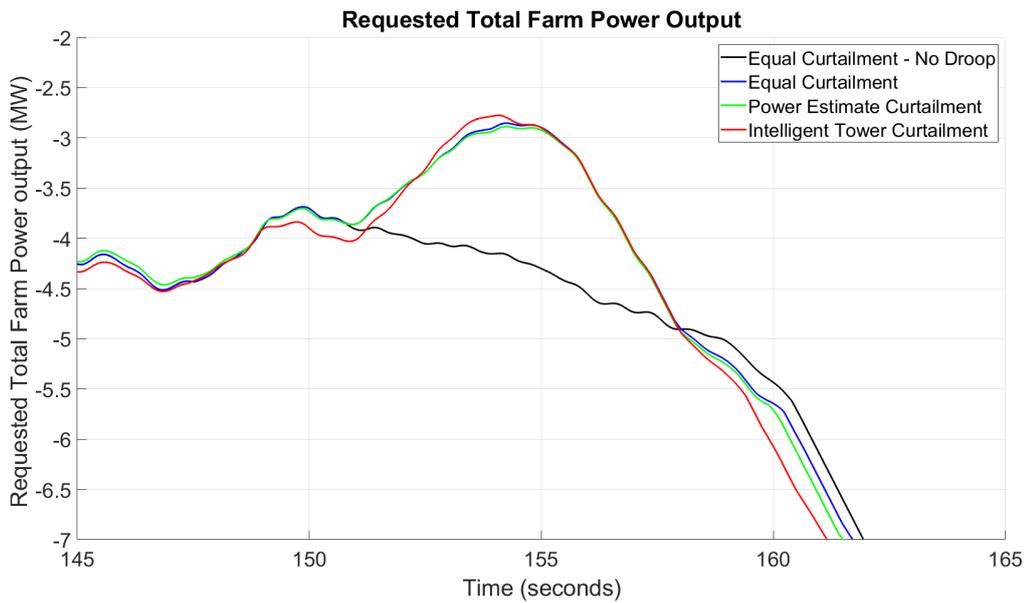


Figure 7.21: The requested change to the wind farm power in the droop simulations for three distribution strategies and when the droop controller is not active at the time of the droop event.

Figure (7.20) and (7.21) show the farmwide requested change to power output for the droop response of the three distribution strategies and the level of requested power adjustment without

Chapter 7. Development of a dynamic power system model in Strathfarm

the droop response. It can be seen that at the time where the grid deviates the requested change in power output of the farm responds as expected for all three strategies.

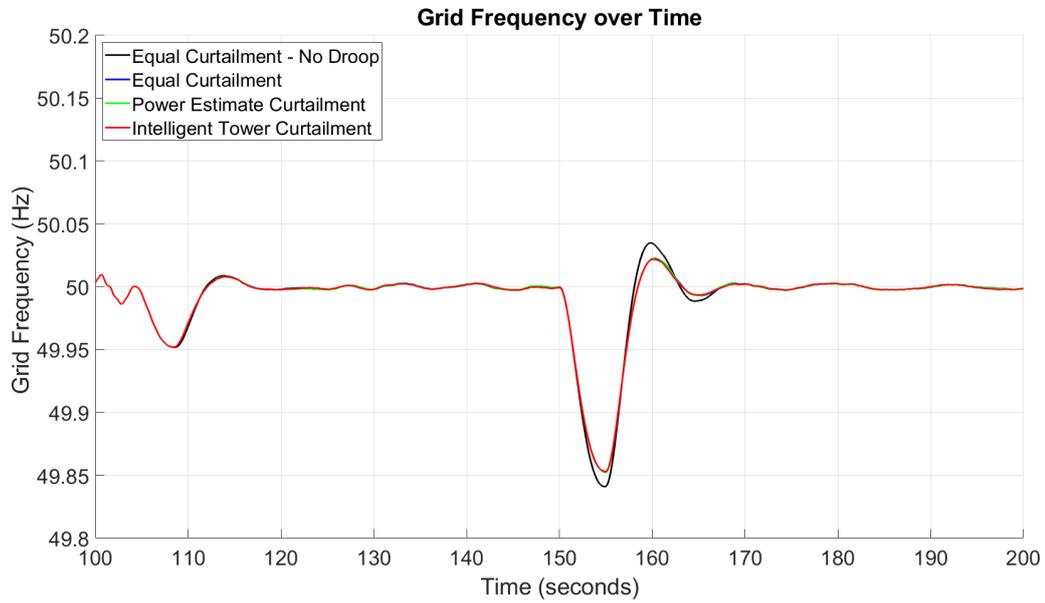


Figure 7.22: The grid frequency in the droop simulations for three distribution strategies and when the droop controller is not active

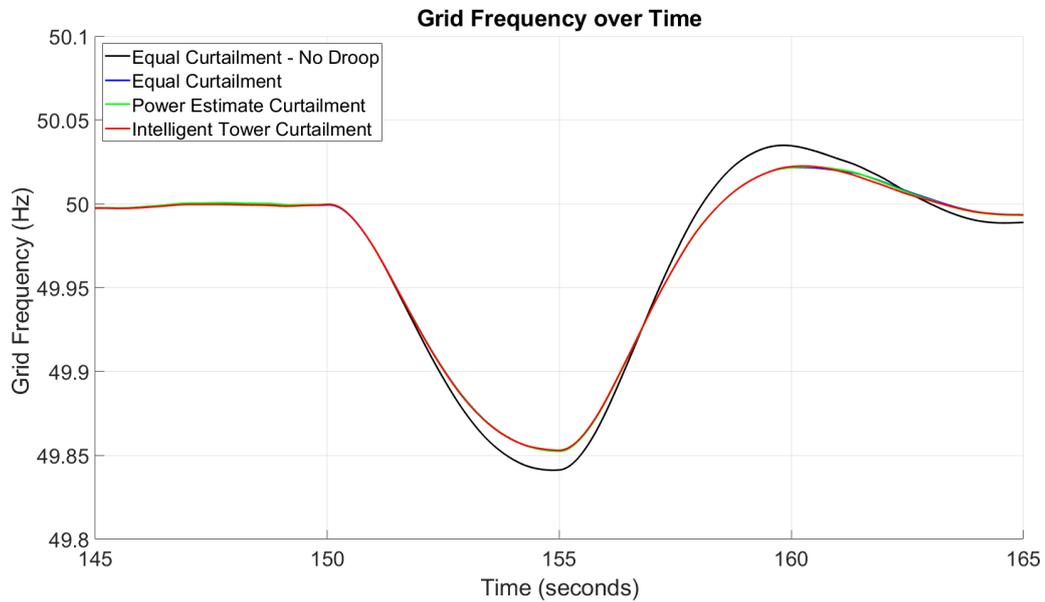


Figure 7.23: The grid frequency in the droop simulations for three distribution strategies and when the droop controller is not active at the time of the droop event.

Chapter 7. Development of a dynamic power system model in Strathfarm

Figure (7.22) and (7.23) show the grid frequency of the simulations using each of the three distribution algorithms and also the equal distribution strategy with no droop control.

It can be clearly seen that the three strategies of curtailment are almost identical and are all contributing to grid stability as they are reducing the magnitude of the frequency nadir and subsequent frequency overshoot seen when there is no droop controller active.

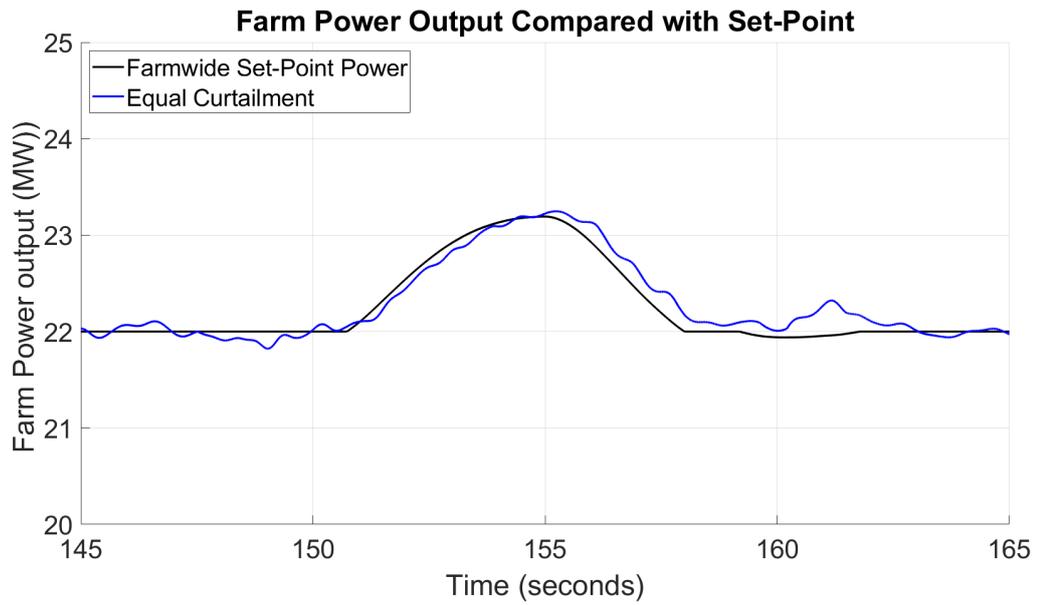


Figure 7.24: The power output and set-point from the droop controller in the equal distribution strategy droop response.

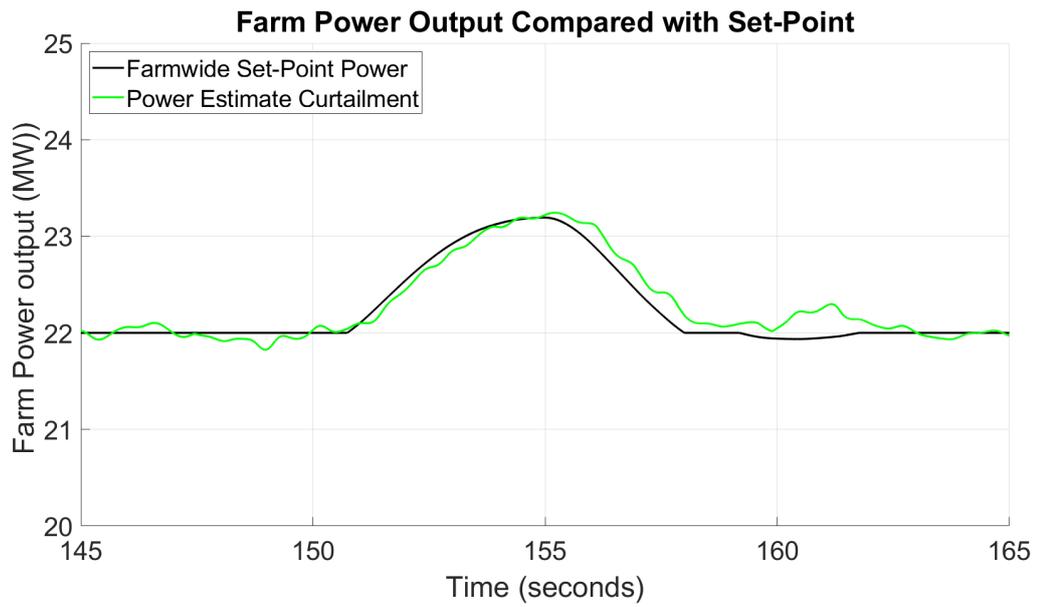


Figure 7.25: The power output and set-point from the droop controller in the power estimation distribution strategy droop response.

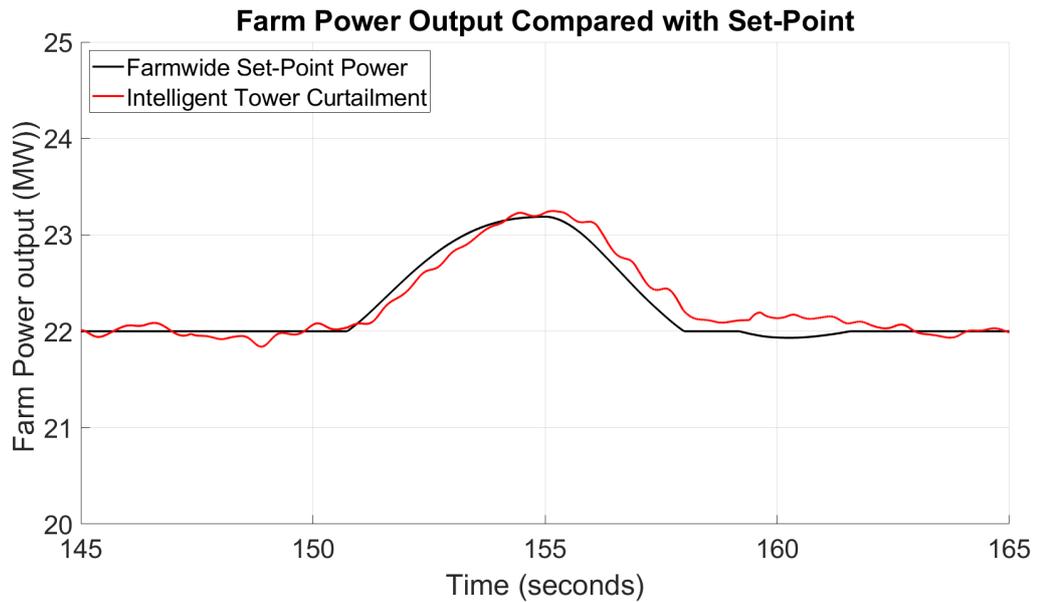


Figure 7.26: The power output and set-point from the droop controller in the intelligent tower distribution strategy droop response.

Figure (7.24), (7.25) and (7.26) show the power of the wind farm compared with the requested set-point of the network controller when the equal distribution, power estimation dis-

Chapter 7. Development of a dynamic power system model in Strathfarm

tribution and intelligent tower distribution strategies are used respectively. It can be seen that the wind farm power follows the change in set-point for the farm in each of the three cases but with a slight lag. This lagging effect is caused by the turbine's maximum rate of change in power being limited to $150 \frac{kW}{s}$ in non-inertial conditions.

7.8.2 Example Inertial Response

The simulations shown in these results use the same wind farm layout as the droop control results as seen in Figure (7.15) .

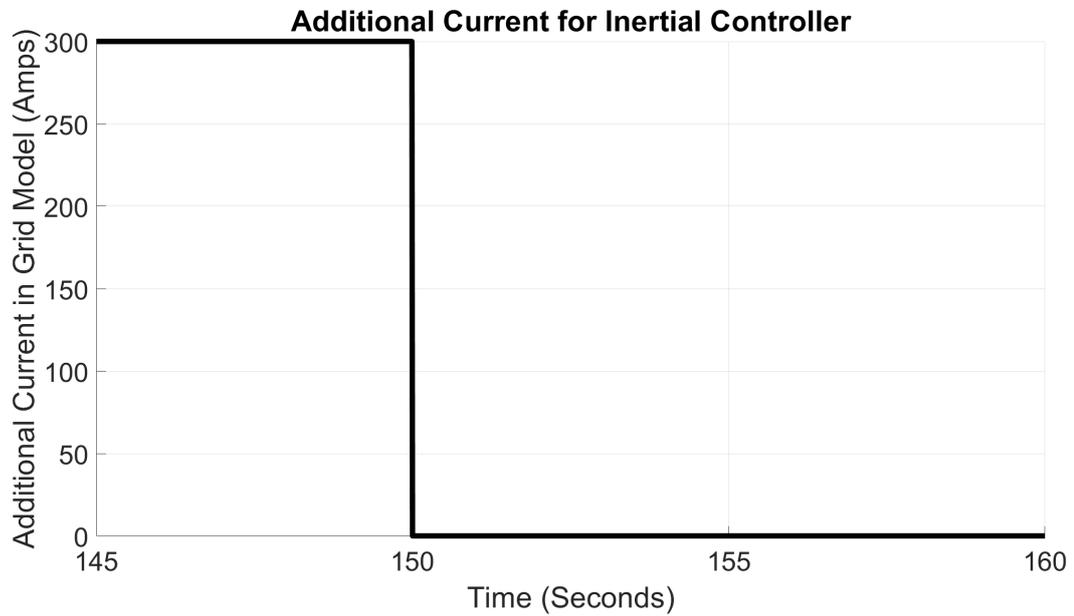


Figure 7.27: The additional current added to the power system to simulation an inertial event.

Figure (7.27) shows the additional current in the power system model used to activate the inertial dispatch in the wind farm controller.

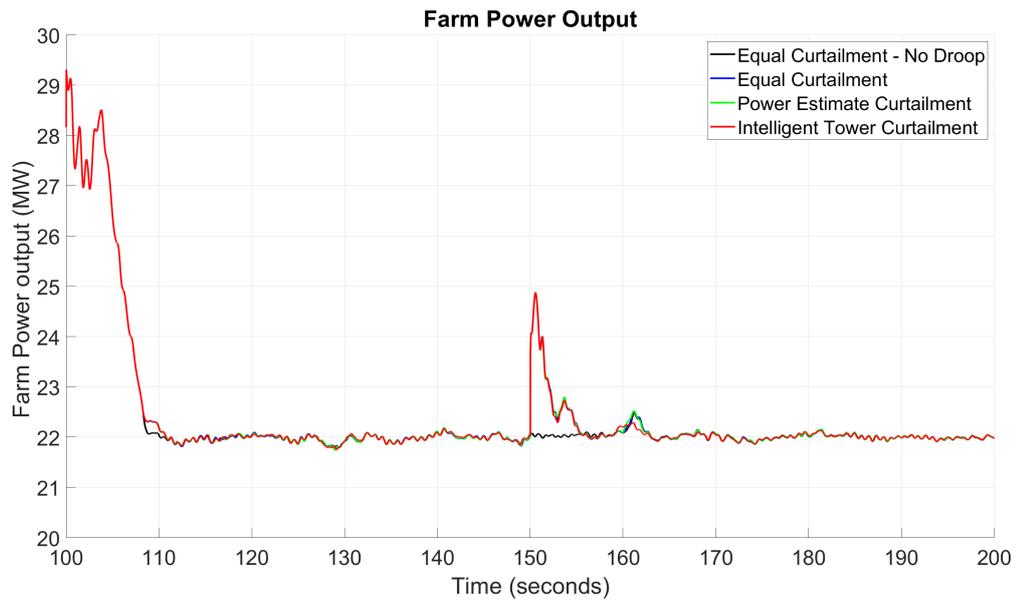


Figure 7.28: The power output of the wind farm in the inertial event for different wind farm algorithms

Figure (7.28) shows the power output of the wind farm in the inertial event for each of the three distribution strategies and when no inertial or droop response is delivered by the wind farm.

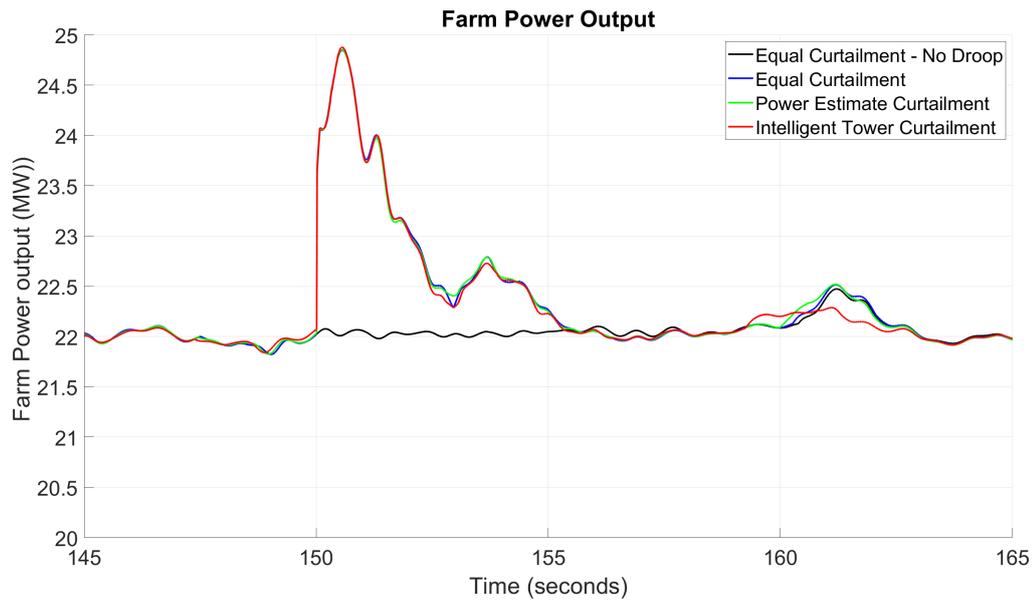


Figure 7.29: The power output of the wind farm in the inertial event for different wind farm algorithms at the time of the inertial event.

Figure (7.29) shows the power output of the wind farm in the inertial event at the time of the frequency disturbance.

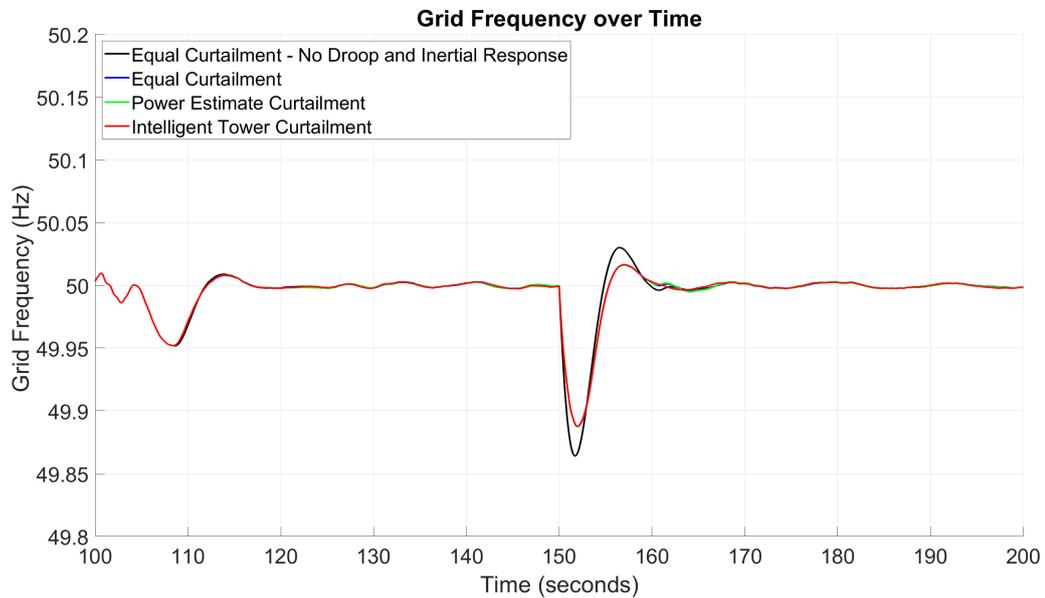


Figure 7.30: The power system frequency of the wind farm in the inertial event for different wind farm algorithms.

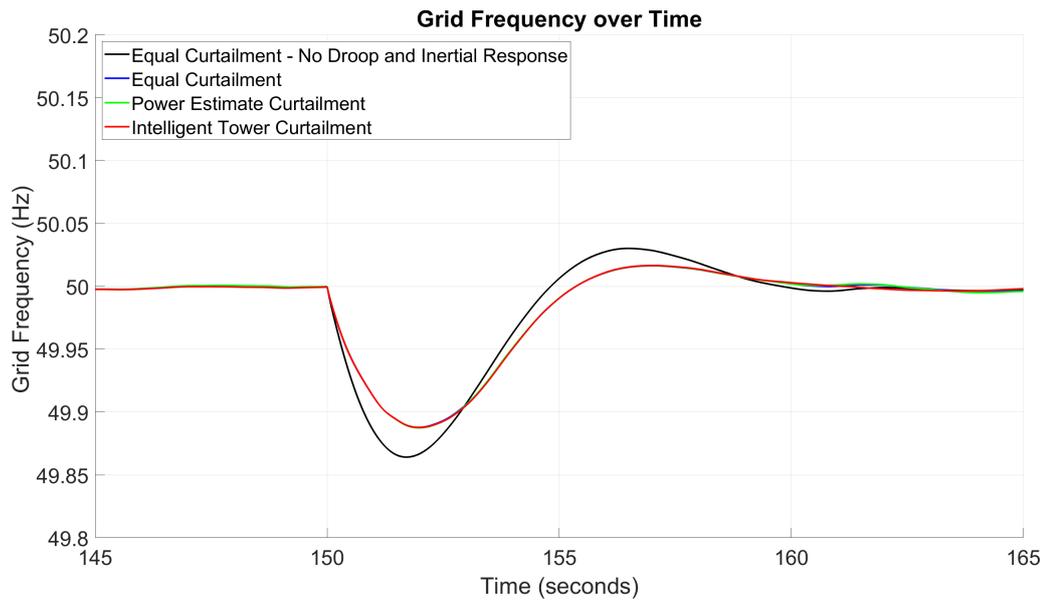


Figure 7.31: The power system frequency of the wind farm in the inertial event for different wind farm algorithms at the time of the inertial event

Figure (7.31) and (7.30) show the grid frequency of the power system in the inertial event for the three curtailment distribution algorithms and when no inertial or droop response is delivered by the wind farm. These graphs show that there is a significant reduction in the deviation of the grid frequency when the inertial response dispatch algorithm is used. This result confirms that the dispatch algorithm is effective in responding to inertial grid events.

It can also be seen that each of the three curtailment distribution approaches have very similar frequency plots, as would be expected given their similar power outputs.

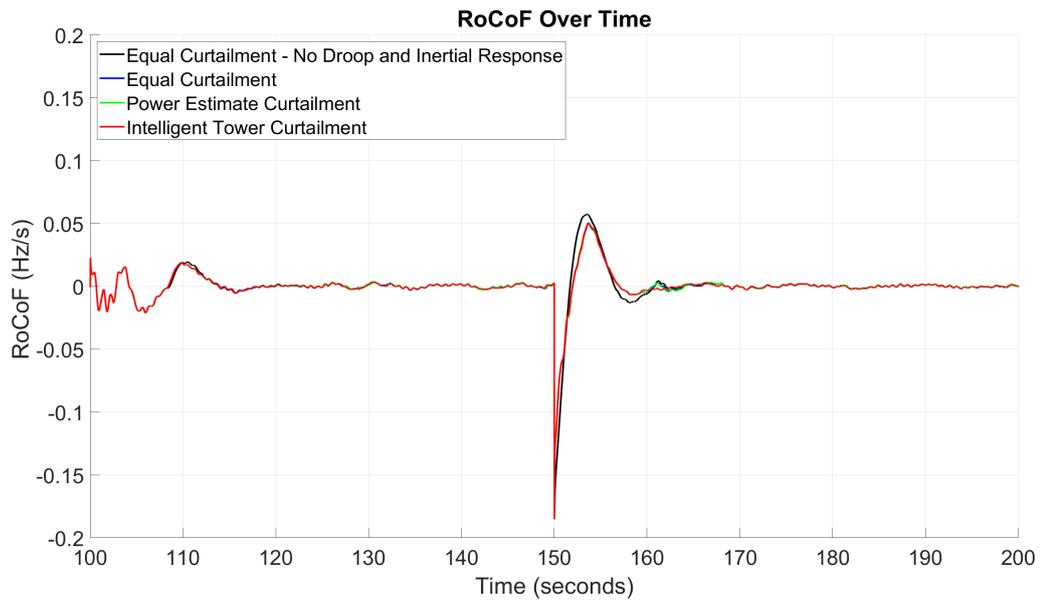


Figure 7.32: The rate of change of frequency of the power system frequency of the wind farm in the inertial event for different wind farm algorithms.

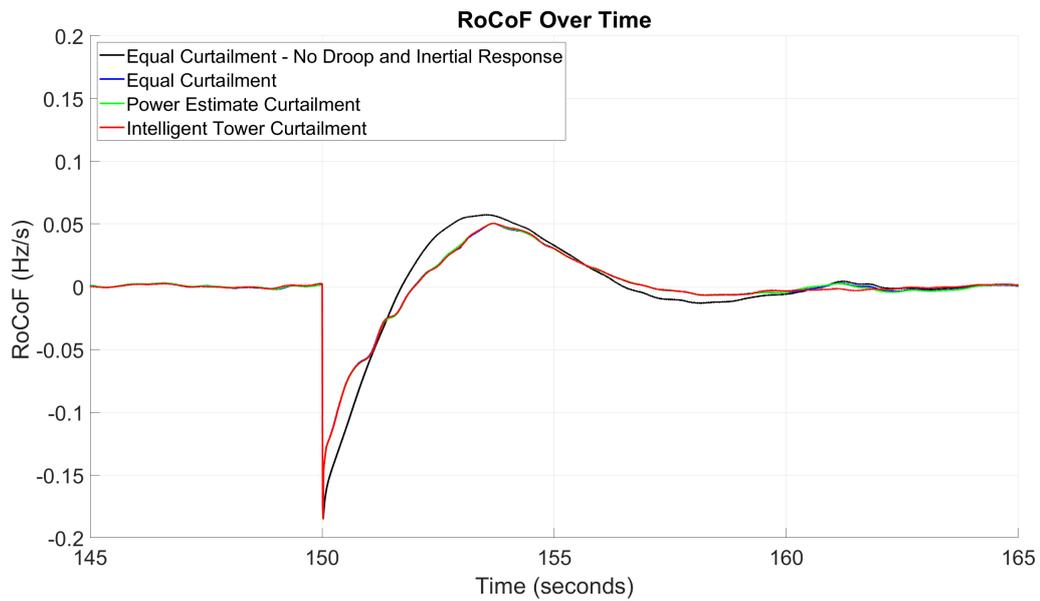


Figure 7.33: The rate of change of frequency of the power system frequency of the wind farm in the inertial event for different wind farm algorithms at the time of the inertial event.

Figure (7.32) shows the RoCoF of the power system in the inertial event for the three curtailment distribution algorithms and when no inertial or droop response is delivered by the

wind farm. Figure (7.33) shows the RoCoF of the power system in the inertial event at the time of the frequency disturbance. These figures show that in the cases where the inertial dispatch is active there is a significantly reduced RoCoF magnitude demonstrating that the dispatch strategy is effective. As with the other results there is not a significant difference between the curtailment distribution strategies.

7.9 Concluding Remarks

The results shown in this chapter demonstrate that the dispatch algorithms developed by [90] are effective in contributing to frequency response for both the inertial and droop dispatch cases. It can be seen in both the droop case and the inertial response case that the choice of curtailment distribution strategy does not impact the frequency response of the farm.

While the model presented here is imperfect it is a strong base for future development, with many areas that could be improved in different projects depending of the priority for that research area. The key area that requires the most attention is optimizing the codes, particularly the non-linear solver algorithm, to improve computational speed to allow for comprehensive assessments of different strategies to be simulated over a representative sample of wind speeds directions and turbulence intensities.

Chapter 8

Development and Inclusion of a Gaussian Wake Model in Strathfarm

This chapter reviews the history of engineering wake models used in computational wind farm simulations including the existing model used in Strathfarm. It then discusses the implementation of the Gaussian wake model proposed in [15] in Strathfarm. An alternative unused model for the implementation of the Gaussian wake is detailed as an Appendix. The motivation for the research shown in this chapter was to enable Strathfarm to be used in the Farmconners benchmarking collaboration with other wind farm modelling research software. For this benchmarking exercise Strathfarm needed to have the most up to date wake model in line with other wind farm modelling research software. The novel research contribution in this chapter is in how we model the wakes so that they can be included with Strathfarm's wind speed model which requires the wake to be considered as an adjustment to the effective wind speed as sampled by the rotor.

8.1 Review and History of Engineering Models of Wakes

8.1.1 The Jensen Wake Model

The earliest approach for modelling wakes was suggested by [57] where the wake deficit was found through considering the conservation of momentum of a linearly expanding wake. Jensen's approach assumes that for a wake x meters behind the rotor, the momentum of the

Chapter 8. Development and Inclusion of a Gaussian Wake Model in Strathfarm

flow through the radius of the wake must be equal to the momentum of flow through a radius of the same size immediately behind the rotor. The conservation of momentum can be expressed as:

$$\pi r_0^2 v_0 + \pi(r^2 - r_0^2)u = \pi r^2 v \quad (8.1)$$

where r_0 is the initial wake radius, v_0 is the wind speed behind the upwind turbine, r is the radius of the wake at a distance downstream of x , u is the unimpeded wind speed and v is the wind speed at a distance x downstream.

As it is assumed that the wake expands linearly as it propagates downstream the wake radius r can be defined as:

$$r = \alpha x + r_0 \quad (8.2)$$

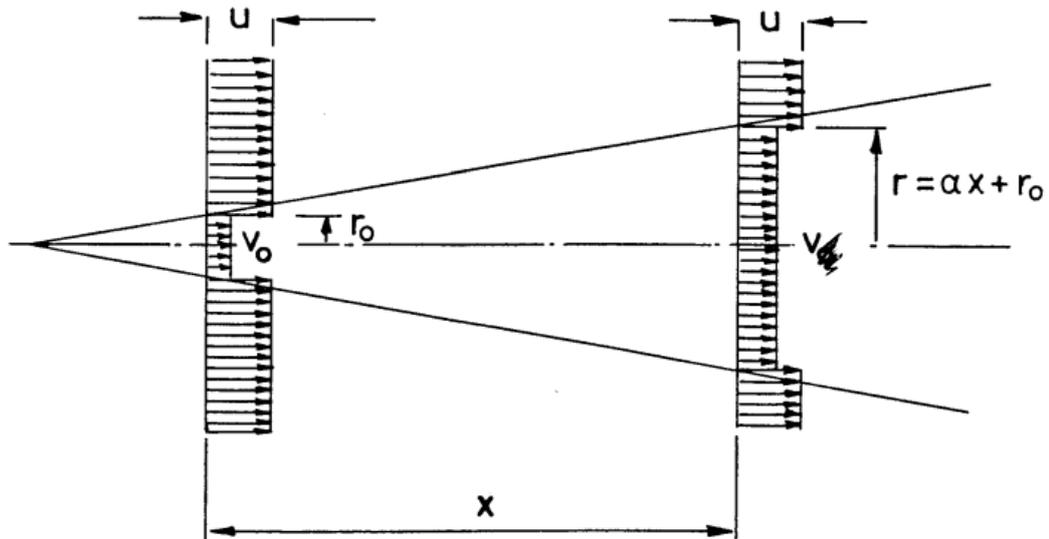


Figure 8.1: An image from [57] showing the wake expansion as the wake travels downwind.

In Reference [57] it is assumed from classical theory that $v_0 = u(1 - 2a)$ where a is the axial induction factor. Thus Equation (8.1) can be written as:

$$v = u \left(1 - 2a \frac{r_0^2}{(r_0 + \alpha x)^2} \right) \quad (8.3)$$

therefore using that $a = \frac{1}{2} - \frac{1}{2}\sqrt{1 - C_t}$ the wake deficit can be written as:

$$\frac{\Delta v}{u_\infty} = \frac{1 - \sqrt{1 - C_t}}{\left(1 + \frac{\alpha x}{r_0}\right)^2} \quad (8.4)$$

Reference [57] also contains an approach for modelling multiple wakes by considering a line of wind turbines in the direction of the wind flow. To find the impact of multiple wakes the wind speed at the third turbine is found by first using the momentum conservation equation in the form:

$$r_0^2(1 - 2av_1) + (r^2 - r_0^2)u = r^2v_2. \quad (8.5)$$

Equation (8.5) can be rearranged to give:

$$\frac{v_2}{u} = 1 - \left(1 - (1 - 2a)\frac{v_1}{u}\right)\left(\frac{r_0}{r_0 + \alpha x_0}\right)^2 \quad (8.6)$$

By repeating the calculations for Equation (8.6) for any downstream turbine it can be shown that the deficit will always be of the form:

$$\frac{v_N}{u} = 1 - \left(1 - (1 - 2a)\frac{v_{N-1}}{u}\right)\left(\frac{r_0}{r_0 + \alpha x_0}\right)^2 \quad (8.7)$$

8.1.2 The Frandsen Wake Model

Following on from [57], the research presented in [37] modelled wakes by first considering the forces acting on the momentum of the flow of a given volume of air through a circular surface area.

This gives the equation:

$$\int_X \rho \frac{\partial \vec{U}}{\partial t} dX + \int_{A_T} \rho \vec{U} (\vec{U} d\vec{A}) = - \int_{A_T} p d\vec{A} + \int_X \rho \vec{g} dX + \vec{T} + \int_{A_T} \vec{\tau} dA \quad (8.8)$$

However, the approach presented in [37] neglects the following terms: the acceleration term ($\int_X \rho \frac{\partial \vec{U}}{\partial t} dX$), the pressure term ($-\int_{A_T} p d\vec{A}$) and the gravity term ($\int_X \rho \vec{g} dX$) on the basis that neglecting these terms is commonly applied.

If the turbulent shear term is neglected and it is assumed that this is being considered significantly far down wind that the pressure can be neglected. Hence, Equation (8.8) can be simplified

to give:

$$T = \int_A \rho U (U_0 - U) dA \quad (8.9)$$

Next, the work in [38] assumes a wake self similarity as a function of the distance of the position from the center of the wake and the down wind distance as:

$$U = U_w(x) f\left(\frac{r}{R}\right) \quad (8.10)$$

[38] argues that by integrating this over a rotor and assuming that the wake can always be modelled as rectangular shape that the forces can be simplified to the equation:

$$T = \rho A U (U_0 - U) \quad (8.11)$$

also noting that the thrust can be defined as:

$$T = \frac{1}{2} \rho A_0 U_0^2 C_t \quad (8.12)$$

By defining the induction factor as $a = 1 - \frac{U}{U_0}$, [38] finds the wake flow speed to be:

$$\frac{U}{U_0} = \frac{1}{2} \pm \frac{1}{2} \sqrt{1 - 2 \frac{A_0}{A} C_t} \quad (8.13)$$

8.1.3 The Gaussian Wake Model

When the Jensen [57] and Frandsen [37, 38] models were developed, wind energy research was in its infancy. However, as wind turbine simulation models have become more sophisticated and computers much more powerful, it has been suggested that these models are no longer sufficiently accurate for use in computational wind farm models. [15] suggest that rather than the previous approach of a “top hat” shaped wake that a Gaussian shape is used instead. As with the previous approach they begin with assuming mass and momentum conservation:

$$\rho \int U_W (U_\infty - U_W) DA = T \quad (8.14)$$

Chapter 8. Development and Inclusion of a Gaussian Wake Model in Strathfarm

with T being found from:

$$T = \frac{1}{2}\rho A_0 U_0^2 C_t \quad (8.15)$$

Again, as before the self-similarity of the wake is defined in the form:

$$\frac{U}{U_\infty} = C(x) f\left(\frac{r}{\delta(x)}\right) \quad (8.16)$$

If the wake is always assumed to have a Gaussian shape then this can be represented by:

$$\frac{U}{U_\infty} = C(x) e^{-\frac{r^2}{2\sigma^2}}. \quad (8.17)$$

[15] shows that:

$$C(x) = 1 - \sqrt{1 - \frac{C_t}{8\left(\frac{\sigma}{d_0}\right)^2}} \quad (8.18)$$

Assuming a linear wake expansion means that σ can be found as a function of x :

$$\frac{\sigma}{d_0} = k^* \frac{x}{d_0} + \epsilon \quad (8.19)$$

where d_0 is the wind turbine diameter, $k^* = 0.3837 * TI + 0.003678$ where TI is the turbulence intensity and $\epsilon = 0.25\sqrt{\beta}$ where $\beta = \frac{1}{2} \frac{1 + \sqrt{1 - C_t}}{\sqrt{1 - C_t}}$. Therefore using these equations the wake deficit at the rotor can be defined as:

$$\frac{\Delta u}{u_\infty} = \left(1 - \sqrt{1 - \frac{C_T}{8\left(k^* \frac{x}{d_0} + 0.2\sqrt{\beta}\right)^2}}\right) \exp\left(-\frac{1}{2\left(k^* \frac{x}{d_0} + 0.2\sqrt{\beta}\right)^2} \left\{ \left(\frac{z - z_h}{d_0}\right)^2 + \left(\frac{y}{d_0}\right)^2 \right\}\right) \quad (8.20)$$

where z is the vertical position, z_h is the height of the centre of the rotor and y is the lateral distance from the centre of the rotor. [86] expands on this by considering the Gaussian wake at a wind farm level. This paper discusses the behaviour of multiple wakes interacting with a turbine stating that it is reasonable to assume that the wake deficits are additive. Hence, this is assumed in the implementation in Strathfarm.

8.2 The Existing Strathfarm Wake Model

The existing wake model within Strathfarm is based on the Frandsen model as discussed in [90].

This model has three key numerical components that require dynamic calculation: the wake deficit, the wake centre position and the wake diameter.

8.2.1 The Wake Centre Calculation

The wake centre is calculated assuming an axis-symmetric wake while also considering wake meandering. The wake meandering is calculated through the lateral turbulence components in the model which exists as a grid of turbulence values on a positional grid through the wind farm. This grid is generated in the preprocessing of the generation of the wind model. The wake position is found by summing the lateral turbulence at each node between two turbines where nodes are included if they fall within the wake's diameter.

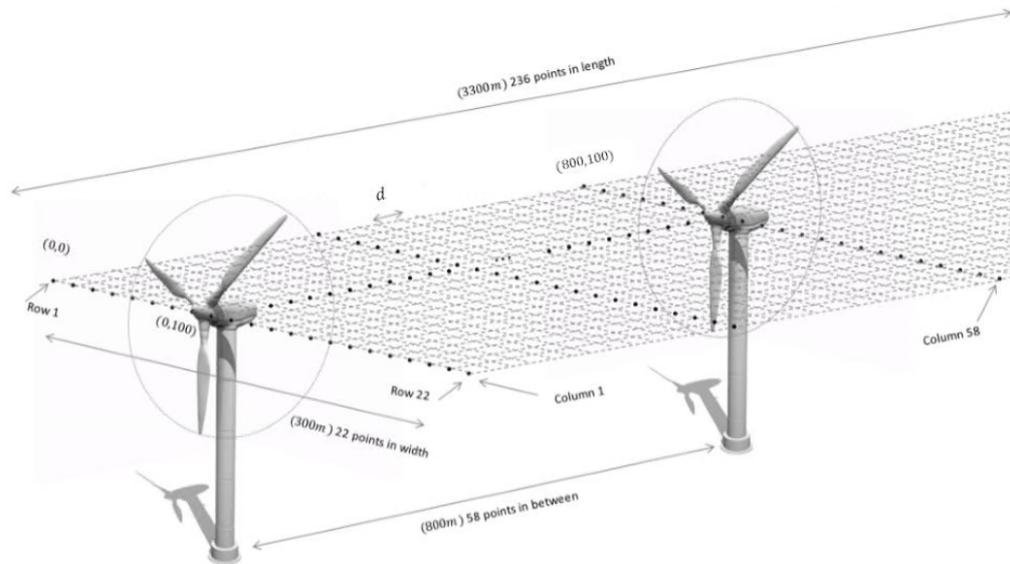


Figure 8.2: Diagram showing the grid used for node based turbulence. From [90].

Figure (8.2) is an image from [90] which shows the nodes of turbulence used in Strathfarm.

8.2.2 The Wake Diameter Calculation

The diameter of the wake is found using the Frandsen method. As the wake propagates downstream it expands increasing its diameter while reducing the wind speed deficit. The wake diameter is found through the equation:

$$WD_j(x_{i,j}, t) = (\beta(t)^{\frac{k}{2}} + \alpha \frac{x}{D_0})^{\frac{1}{k}} D_0 \quad (8.21)$$

where β is the wake factor given as:

$$\beta(t) = \frac{1}{2} \frac{1 + \sqrt{1 - C_T(t - \tau)}}{\sqrt{1 - C_T(t - \tau)}} \quad (8.22)$$

8.2.3 The Wake Deficit Calculation

The wake deficit is applied to the turbines as a binary reduction in the wind speed meaning that if the turbine is within the wake, the deficit is applied to its speed, and if the turbine is not within the wake diameter, the deficit is not applied.

The deficit is found through the equation:

$$Def_j(t) = 1 - \frac{CT_i(t - \tau) D_0^2}{WD_j(t)^2} \quad (8.23)$$

where j is the wind turbine experiencing the wake and i is the turbine the wake is from. This binary operation of the wake has the impact of effectively applying step changes in wind speed to the turbines when they move in and out of the wake's diameter.

8.3 Development of a Gaussian Wake Model for Strathfarm

Updating Strathfarm with a Gaussian wake model was not as straightforward as simply averaging the wind speed deficit over the rotor area as the approach doing this (seen in Appendix C) neglected components of the wind speed model from [39]. When the Frandsen model was applied the rotor saw a uniform change over the whole swept area resulting in the rotor loads not being affected by rotor sampling effects. The implementation developed in this chapter includes the wakes as part of the deterministic sampling of the wind field. This sampling occurs

Chapter 8. Development and Inclusion of a Gaussian Wake Model in Strathfarm

at frequencies of nP where P is the period of the one rotation of the turbine's rotor and n is a positive integer.

The Gaussian wake model in Strathfarm is found through using 4-dimensional lookup tables based on a 6th order polyfit of the Gaussian wake. The inputs to the lookup tables are the distance between the turbines, the upwind turbine CT , the lateral position of the wake compared to the centre of the downwind turbine and the turbulence intensity of the wind field.

The wake's change on the wind field has to be considered in these terms:

$$a_{nP} = -\frac{c+1}{\pi R^{c+1}} V_m \int_0^{2\pi} f(r_w, \theta) \cos n\theta d\theta \quad (8.24)$$

$$b_{nP} = -\frac{c+1}{\pi R^{c+1}} V_m \int_0^{2\pi} f(r_w, \theta) \sin n\theta d\theta \quad (8.25)$$

where a_{nP} and b_{nP} are Fourier coefficients, c is a weighting coefficient, θ is the azimuthal angle of rotation, r_w the radial distance from the wake centre to an arbitrary point in the swept area of the rotor, and,

$$f(r_w, \theta) = \int_0^R (K_3(l^2 - 2lr \sin \theta + r^2)^3 + K_2(l^2 - 2lr \sin \theta + r^2)^2 + K_1(l^2 - 2lr \sin \theta + r^2) + K_0)r^c dr \quad (8.26)$$

where K_3 , K_2 , K_1 and K_0 are functions of C_T , the turbulence intensity, the distance downstream between the turbines, and the lateral distance between the centre of the wind turbine's rotor and the centre of the wake, l .

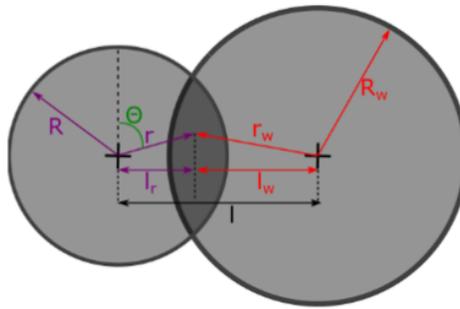


Figure 8.3: Diagram showing the wake interaction with a turbine's swept area.

Figure(8.3) shows the interaction of a wake across a turbine's swept area.

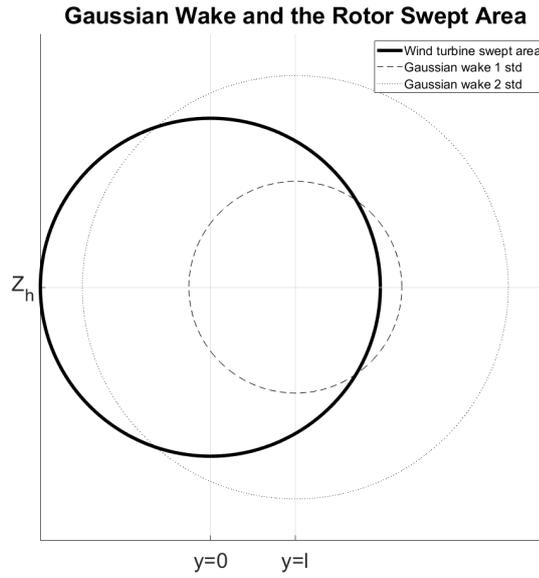


Figure 8.4: Diagram showing the wake interaction with a turbine's swept area.

Figure(8.4) shows the rotor swept area interacting with a Gaussian wake. In this diagram the wake is shown as outlines representing standard deviations of the Gaussian wake.

It should be noted that $l^2 - 2lr \sin \theta + r^2$ is used here to account for the offset in the Gaussian wake from the centre of the turbine requiring an offset in the y direction. $l^2 - 2lr \sin \theta + r^2$ can be found through considering the first principles of integrating in a polar reference frame with a y direction offset of l . Non-dimensionalising equation 8.26 using $\bar{r} = \frac{r}{R}$ and $\bar{l} = \frac{l}{R}$ gives:

$$f(r_w, \theta) = \int_0^1 (K_3 R^6 (\bar{l}^2 - 2\bar{l}\bar{r} \sin \theta + \bar{r}^2)^3 + K_2 R^4 (\bar{l}^2 - 2\bar{l}\bar{r} \sin \theta + \bar{r}^2)^2 + K_1 R^2 (\bar{l}^2 - 2\bar{l}\bar{r} \sin \theta + \bar{r}^2) + K_0) \bar{r}^c R^{c+1} d\bar{r} \quad (8.27)$$

With the K_n values scaled by

$$\bar{K}_n = K_n R^{2n+c} \quad (8.28)$$

Equation(8.27) simplifies to:

$$f(r_w, \theta) = \int_0^1 (\bar{K}_3(\bar{l}^2 - 2\bar{l}\bar{r} \sin \theta + \bar{r}^2)^3 + \bar{K}_2(\bar{l}^2 - 2\bar{l}\bar{r} \sin \theta + \bar{r}^2)^2 + \bar{K}_1(\bar{l}^2 - 2\bar{l}\bar{r} \sin \theta + \bar{r}^2) + \bar{K}_0)\bar{r}^c d\bar{r} \quad (8.29)$$

Hence, solving the integral gives expressions for the Fourier coefficients a_{nP} and b_{nP} of,

$$a_{nP} = -\frac{c+1}{\pi} V_m \int_0^{2\pi} (A + B + C + D) \cos nP d\theta \quad (8.30)$$

$$b_{nP} = -\frac{c+1}{\pi} V_m \int_0^{2\pi} (A + B + C + D) \sin nP d\theta \quad (8.31)$$

where,

$$A = \frac{\bar{K}_3 \bar{l}^6}{c+1} - \frac{6\bar{K}_3 \bar{l}^5 \sin \theta}{c+2} - \frac{6\bar{K}_3 \bar{l}^4 \cos 2\theta}{c+3} + \frac{9\bar{K}_3 \bar{l}^4}{c+3} - \frac{18\bar{K}_3 \bar{l}^3 \sin \theta}{c+4} + \frac{2\bar{K}_3 \bar{l}^3 \sin 3\theta}{c+4} - \frac{6\bar{K}_3 \bar{l}^2 \cos 2\theta}{c+5} + \frac{9\bar{K}_3 \bar{l}^2}{c+5} - \frac{6\bar{K}_3 \bar{l} \sin \theta}{c+6} + \frac{\bar{K}_3}{c+7} \quad (8.32)$$

$$B = \frac{\bar{K}_2 \bar{l}^4}{c+1} - \frac{4\bar{K}_2 \bar{l}^3 \sin \theta}{c+2} + \frac{4\bar{K}_2 \bar{l}^2 \sin^2 \theta}{c+3} + \frac{2\bar{K}_2 \bar{l}^2}{c+3} - \frac{4\bar{K}_2 \bar{l} \sin \theta}{c+4} + \frac{\bar{K}_2}{c+5} \quad (8.33)$$

$$C = \frac{\bar{K}_1 \bar{l}^2}{c+1} - \frac{2\bar{K}_1 \bar{l} \sin \theta}{c+2} + \frac{\bar{K}_1}{c+3} \quad (8.34)$$

$$D = \frac{\bar{K}_0}{c+1} \quad (8.35)$$

The additive nature of the polynomial means that each term in the integral can be considered separately, with the majority returning zero for most values of n . The non-zero Fourier coefficients are given below,

$$a_{0P} = -(c+1)V_m \left(\frac{2\bar{K}_3 \bar{l}^6}{c+1} + \frac{18\bar{K}_3 \bar{l}^4}{c+3} + \frac{18\bar{K}_3 \bar{l}^2}{c+5} + \frac{\bar{K}_3}{c+7} + \frac{2\bar{K}_2 \bar{l}^4}{c+1} + \frac{2\bar{K}_2}{c+5} + \frac{8\bar{K}_2 \bar{l}^2}{c+3} + \frac{2\bar{K}_1 \bar{l}^2}{c+1} + \frac{2\bar{K}_1}{c+3} + \frac{2\bar{K}_0}{c+1} \right) \quad (8.36)$$

$$b_{1P} = -(c+1)V_m \left(-\frac{6\bar{K}_3\bar{l}^5}{c+2} - \frac{18\bar{K}_3\bar{l}^3}{c+4} - \frac{6\bar{K}_3\bar{l}}{c+6} - \frac{4\bar{K}_2\bar{l}^3}{c+2} - \frac{4\bar{K}_2\bar{l}}{c+4} - \frac{2\bar{K}_1\bar{l}}{c+2} \right) \quad (8.37)$$

$$a_{2P} = -(c+1)V_m \left(-\frac{6\bar{K}_3\bar{l}^4}{c+3} - \frac{6\bar{K}_3\bar{l}^2}{c+5} - \frac{2\bar{K}_2\bar{l}^2}{c+3} \right) \quad (8.38)$$

$$b_{3P} = -(c+1) \left(\frac{2\bar{K}_3\bar{l}^3}{c+4} \right) \quad (8.39)$$

Hence an addition δv can be made to the wind speed local to the rotor where,

$$\delta v = \frac{a_{0P}}{2} + b_{1P} \sin \theta + a_{2P} \cos 2\theta + b_{3P} \sin 3\theta \quad (8.40)$$

8.4 Implementation in Strathfarm

Lookup tables are used in Strathfarm to find the K values which are then used in Function blocks based to find the a_{NP} and b_{NP} values for use in the structural sections of the wind turbine model.

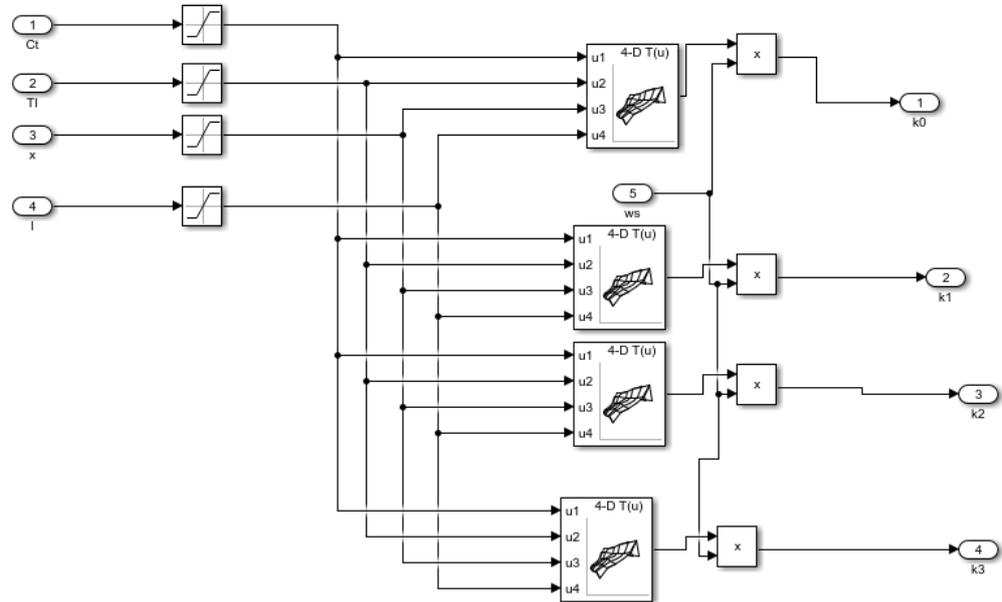


Figure 8.5: The turbine level implementation of the lookup tables in Strathfarm.

Figure (8.5) shows the turbine level implementation of the wake lookup tables in Strathfarm. The reason for the inclusion of the saturation blocks is that without them the 4-D lookup tables will extrapolate beyond the domain where the modelling is valid.

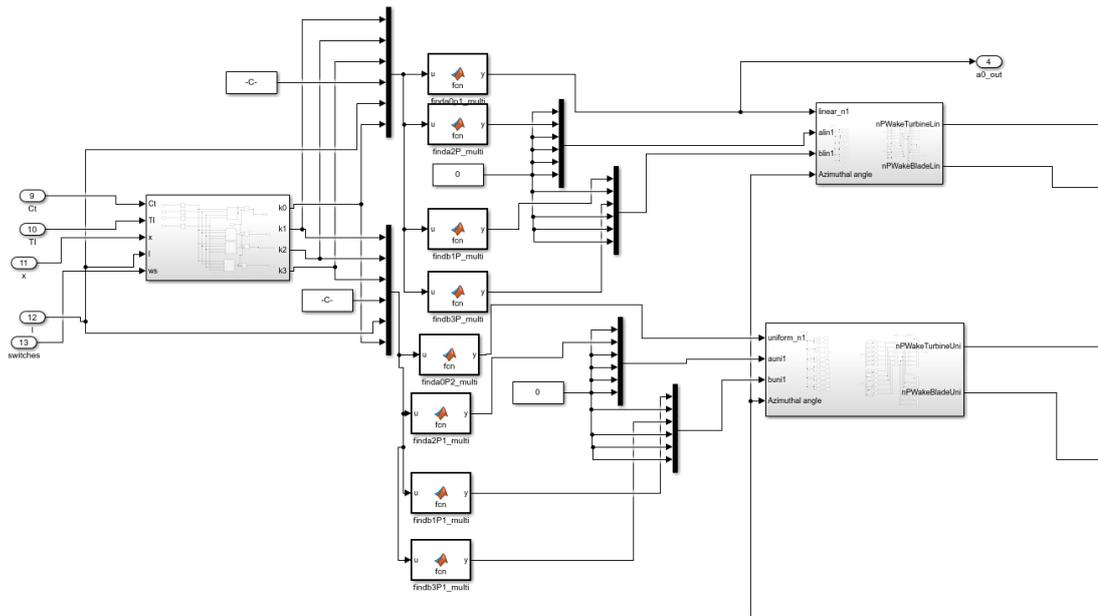


Figure 8.6: The turbine level implementation of the wake model in Strathfarm.

Figure (8.6) shows the turbine level implementation of the wake model in Strathfarm. Each of the in built functions sums all of the outputs of the lookup table for each of the upwind turbines based on the equations shown in the previous subsection. The outputted deficit coefficients are assumed to be additive based on the discussion presented in [86].

8.4.1 Example Plots

The figures in this subsection show a comparison of the Actual Deficit (the wake as given by the equations in [15]) and the estimated wake (the wake as calculated from the polyfit). Each of the four plots show that the polyfit wake is very similar to the numerical value except when there is a large lateral offset. However, when the wake is laterality offset the effective deficit on the down wind turbine is very low so the differences in result does not impact the accuracy of the model. The different C_T values at the upwind turbine, turbulence intensities and down wind distances and lateral offset distances are given in Table (8.1).

Chapter 8. Development and Inclusion of a Gaussian Wake Model in Strathfarm

Figure	C_T at the upwind turbine	Turbulence intensity	Downwind Distance in turbine Diameters	Lateral offset of the wake centre in Turbine Diameters
Figure (8.7)	0.8	0.1	4	0
Figure (8.8)	0.4	0.1	4	0
Figure (8.9)	0.8	0.065	4	0
Figure (8.10)	0.8	0.1	8	0
Figure (8.11)	0.8	0.1	4	0.4
Figure (8.12)	0.8	0.1	4	1
Figure (8.13)	0.4	0.065	4	0.4
Figure (8.14)	0.2	0.065	4	0.4

Table 8.1: The values used in the figures shown in this section

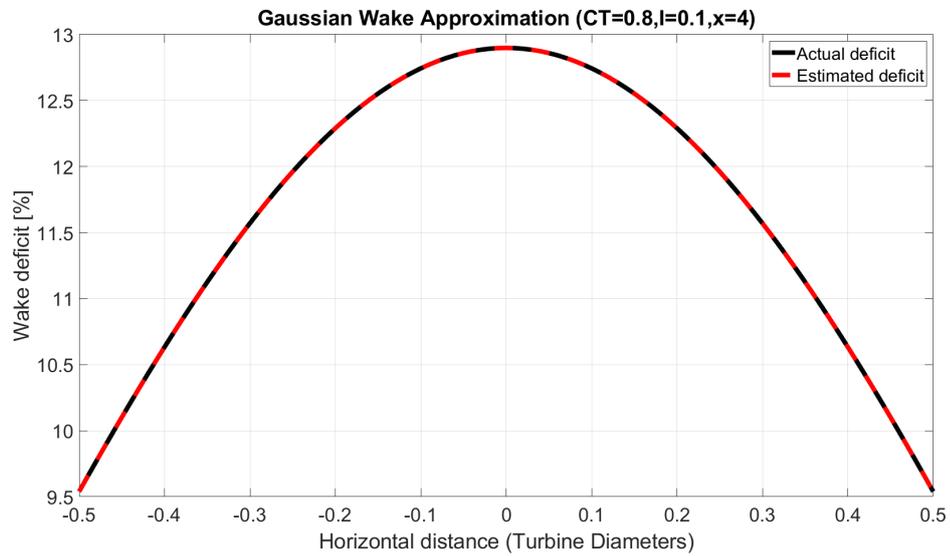


Figure 8.7: Gaussian wake with a C_T value of 0.8 at the upwind turbine, a turbulence intensity of 0.1 and a downwind distance of 4 turbine diameters.

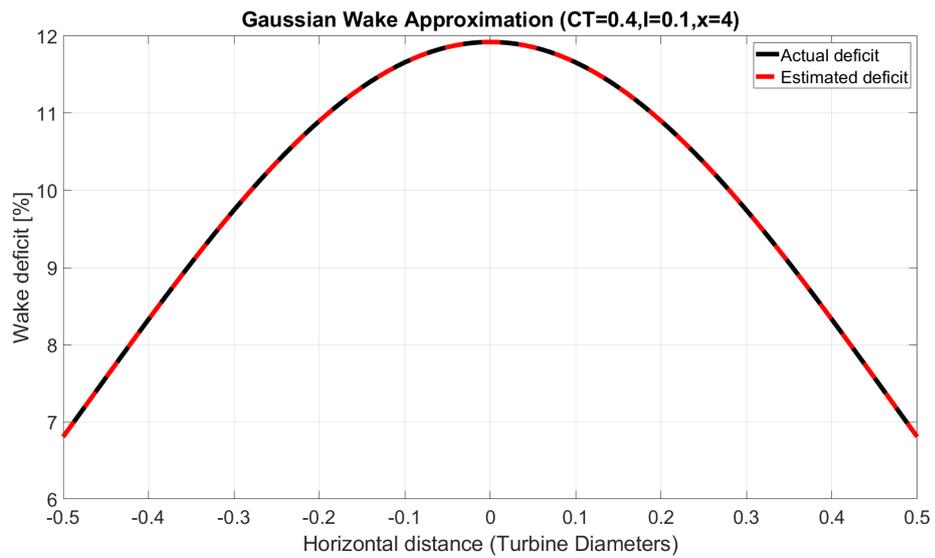


Figure 8.8: Gaussian wake with a CT value of 0.4 at the upwind turbine, a turbulence intensity of 0.1 and a downwind distance of 4 turbine diameters.

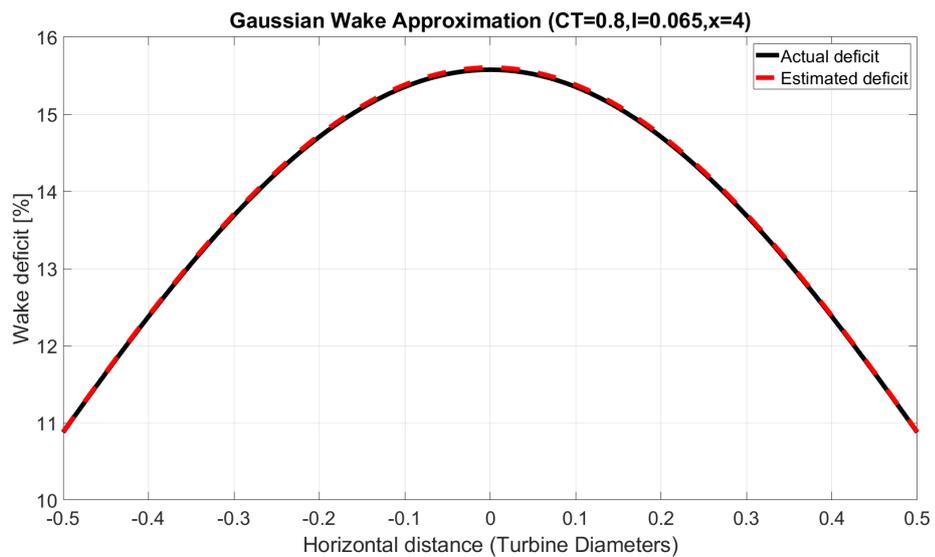


Figure 8.9: Gaussian wake with a CT value of 0.8 at the upwind turbine, a turbulence intensity of 0.065 and a downwind distance of 4 turbine diameters.

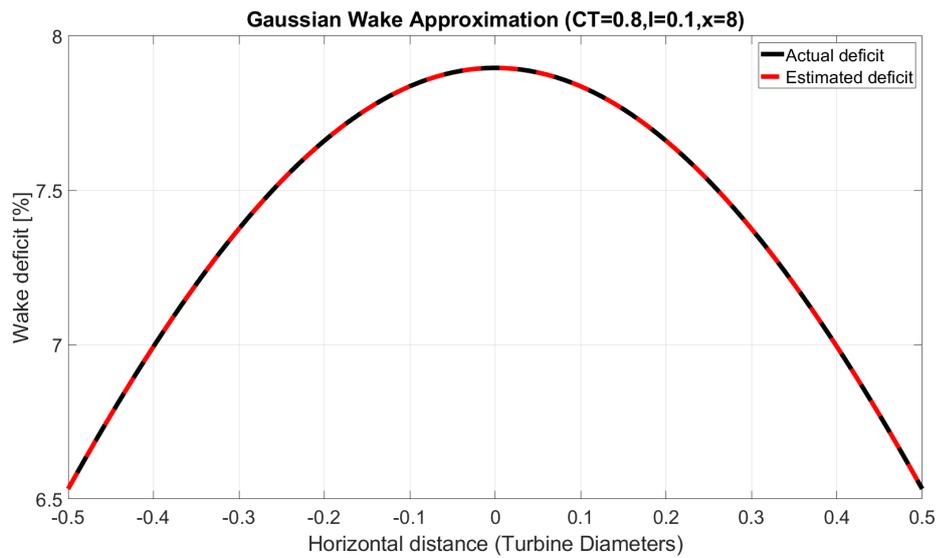


Figure 8.10: Gaussian wake with a CT value of 0.8 at the upwind turbine, a turbulence intensity of 0.1 and a downwind distance of 8 turbine diameters.

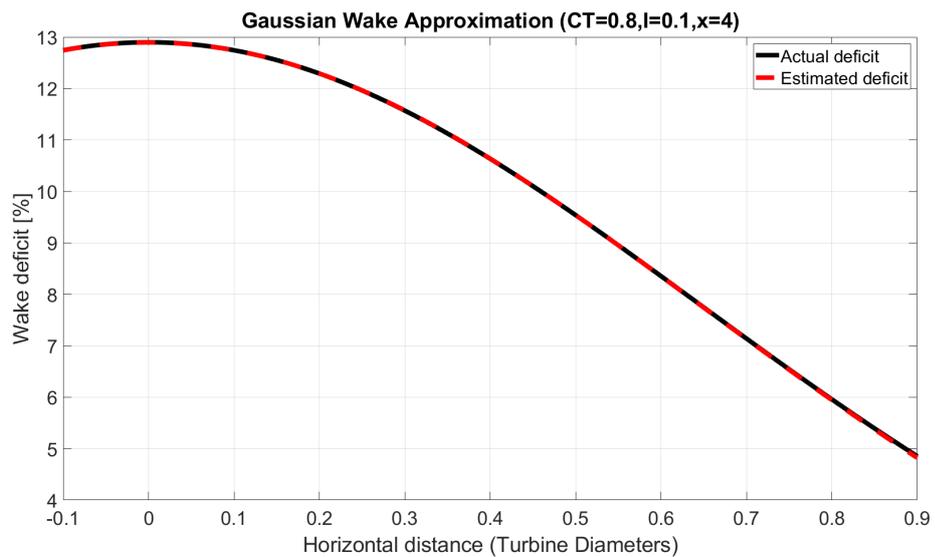


Figure 8.11: Gaussian wake with a CT value of 0.8 at the upwind turbine, a turbulence intensity of 0.1 and a downwind distance of 4 turbine diameters with a lateral offset of 0.4 turbine diameters.

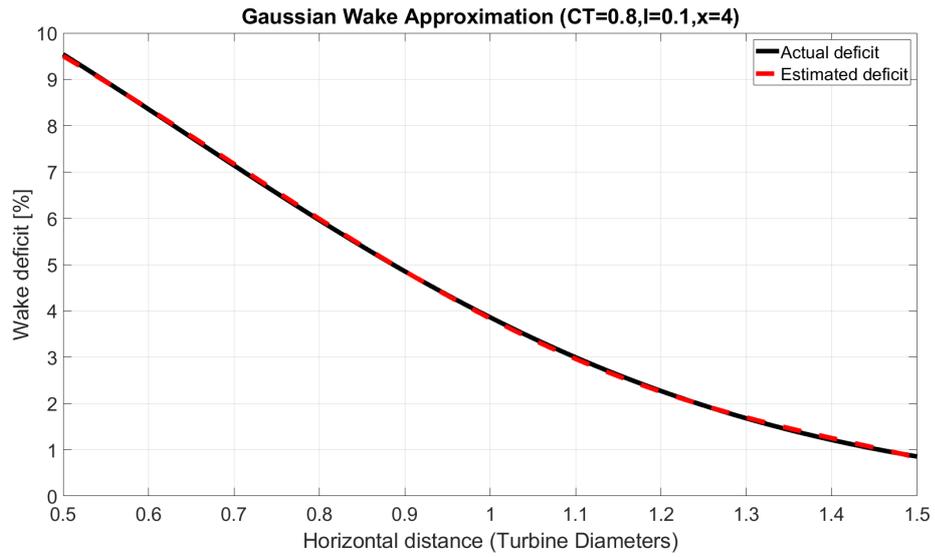


Figure 8.12: Gaussian wake with a CT value of 0.8 at the upwind turbine, a turbulence intensity of 0.1 and a downwind distance of 4 turbine diameters with a lateral offset of 1 turbine diameters.

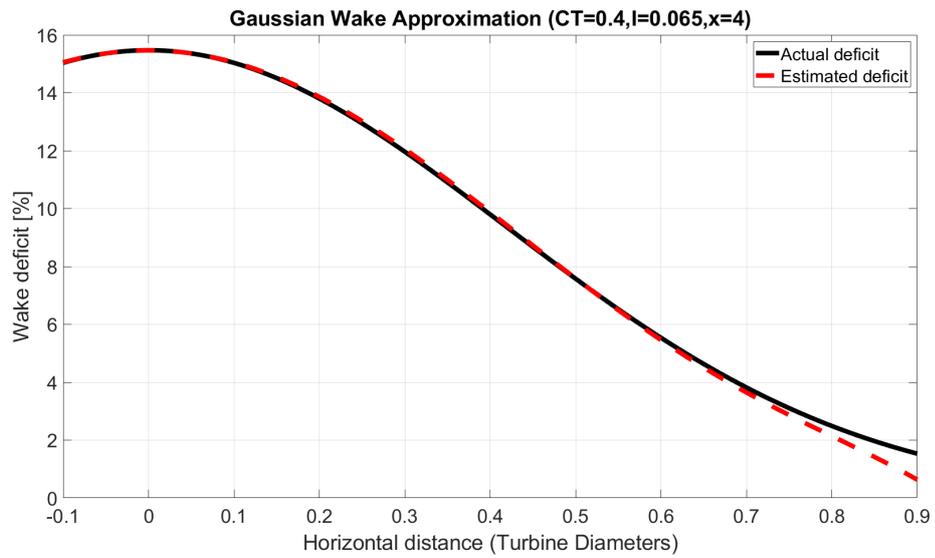


Figure 8.13: Gaussian wake with a CT value of 0.4 at the upwind turbine, a turbulence intensity of 0.065 and a downwind distance of 4 turbine diameters with a lateral offset of 0.4 turbine diameters.

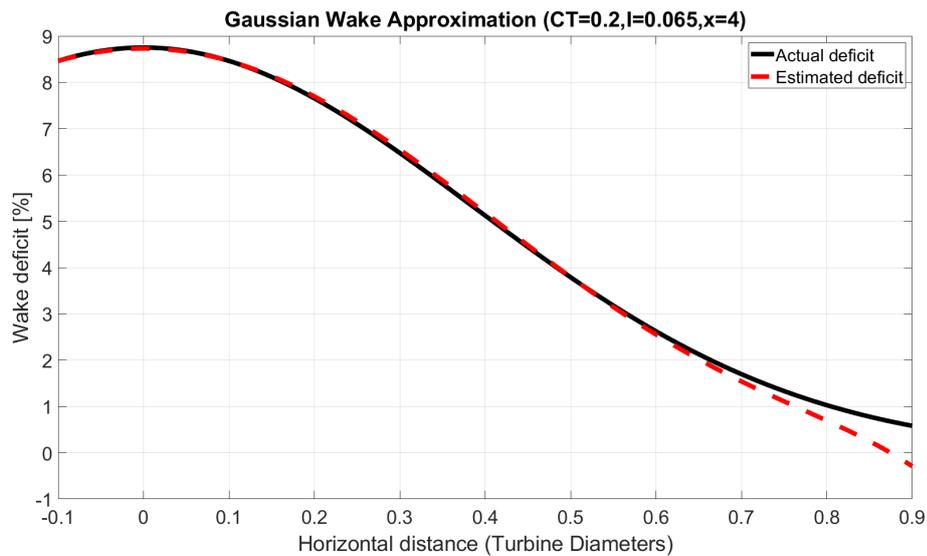


Figure 8.14: Gaussian wake with a CT value of 0.2 at the upwind turbine, a turbulence intensity of 0.065 and a downwind distance of 4 turbine diameters with a lateral offset of 0.4 turbine diameters.

8.5 Comparison of Wake Models

For these results a two turbine Strathfarm model has been used. This model contains two 5MW Supergen exemplar turbines separated by 1000 meters (8 rotor diameters) with one turbine downwind of the other. For this simulation the turbulent components of the wind field have been removed so that the wake effect can be seen more clearly. The wake from the upwind turbine is panned across the downwind turbine at two different speeds (1 m/s and 2m/s) to show the impact of the wake on the downwind turbine. Three wake models have been compared: The Gaussian wake model, the Frandsen wake model and the Frandsen wake model with a lensing adjustment [90]. For the Gaussian wake model's wake values the a_0 component of the wake has been used as it is approximately comparable to the total wake magnitude at the wake centre.

8.5.1 1 m/s wake movement

For these simulations the wake moves laterally at 1m/s.

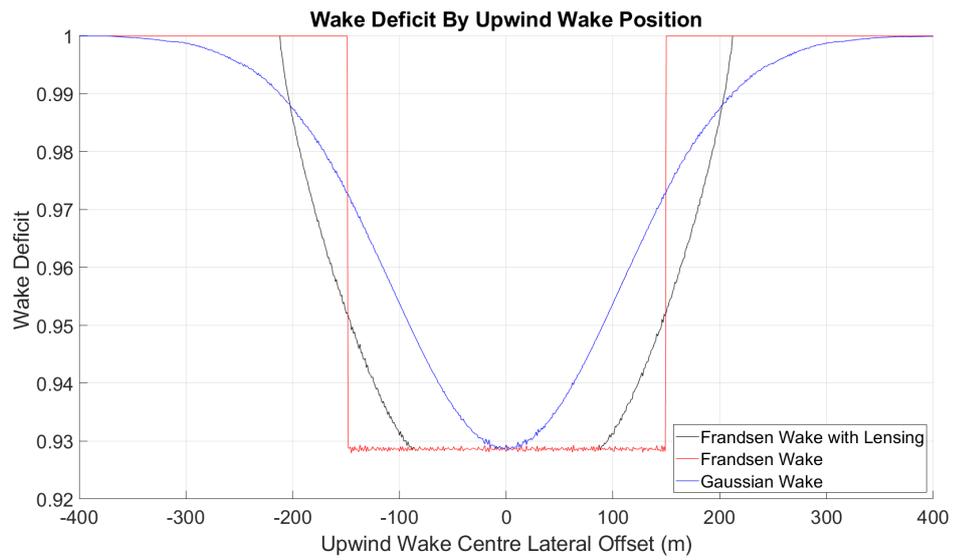


Figure 8.15: The wake deficit for the three wake models by lateral wake centre offset.

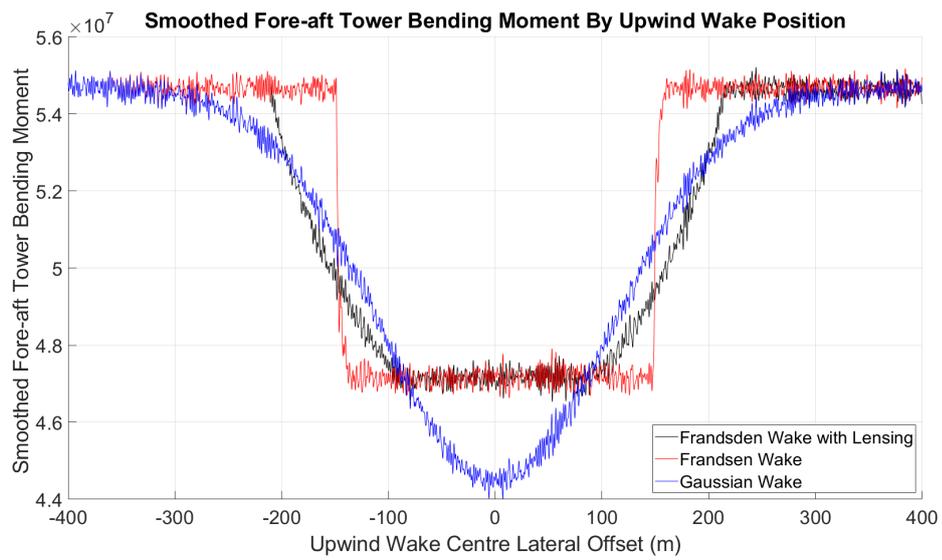


Figure 8.16: The fore-aft tower bending moment of the downwind turbine by lateral wake centre offset.

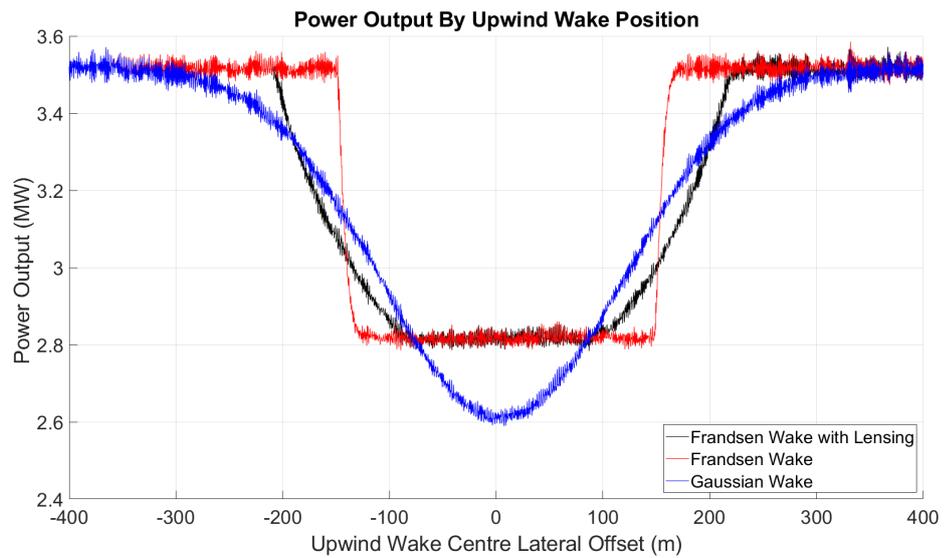


Figure 8.17: The power output of the downwind turbine by lateral wake centre offset.

8.5.2 2 m/s wake movement

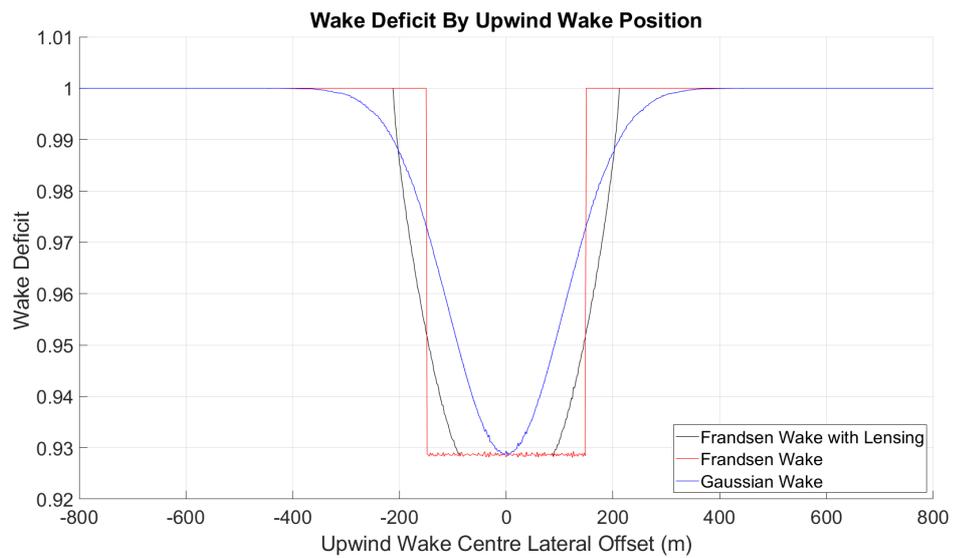


Figure 8.18: The wake deficit for the three wake models by lateral wake centre offset.

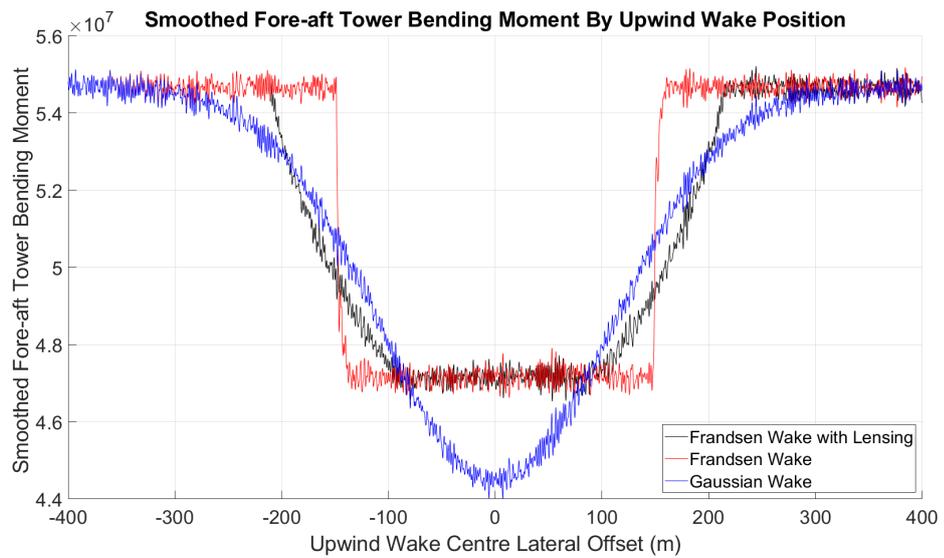


Figure 8.19: The fore-aft tower bending moment of the downwind turbine by lateral wake centre offset.

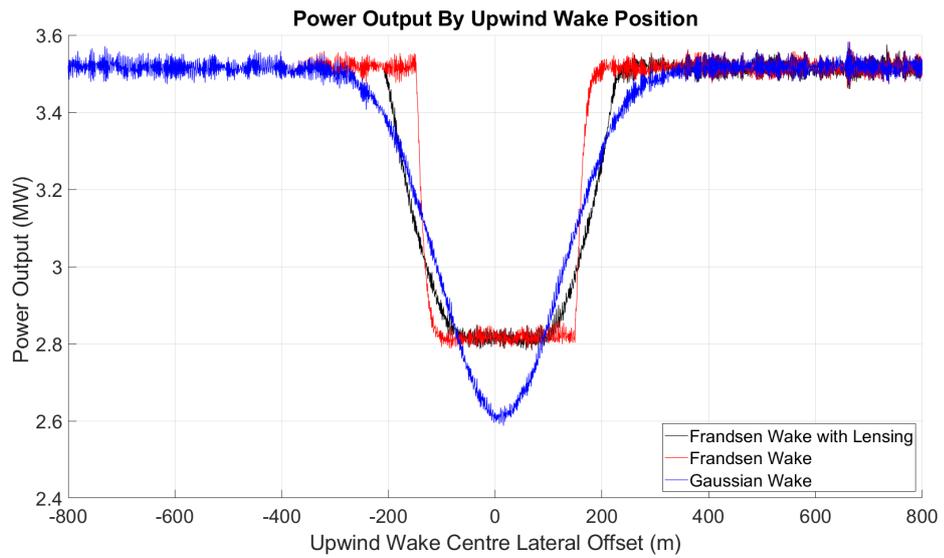


Figure 8.20: The power output of the downwind turbine by lateral wake centre offset.

It can be seen that in terms of power output and tower bending moment that the lensing model acts as a compromise between the Gaussian wake model and the Frandsen wake model. It would still, however, result in higher DELs for turbines in partial wakes.

Chapter 9

Conclusions

As wind generation increases in penetration in power systems around the world, wind farms will need to operate in a more flexible way. This thesis has set out ways in which wind farms can be operated and subsequent developed models for simulating the impact novel operational strategies will have. By first reviewing the academic literature chapter four found a gap in the modelling where there is limited wind farm modelling including both a high fidelity wind farm model with a representative grid model. However, before this gap could be exploited, Strathfarm's wind farm controllers needed to be developed. Chapter 5 considered how to create a wind farm set-point power tracking controller, with the novelty being that it is for a discrete time model. The discretisation of the wind farm controller is shown including the anti-windup developed at both the wind farm and turbine level.

Chapter 6 first considers the steady state bending moments of the Supergen 5MW wind turbine, which are then used in developing a novel wind farm control curtailment allocation strategy. This newly developed wind farm control algorithm is compared with two other approaches in a representative validation exercise finding that the newly developed strategies result in larger reductions in fatigue for both the tower and blades of the turbines.

Chapter 7 develops a novel power system model is developed to fill the research gap found in the literature review in chapter 4. This newly developed power system model is based on the wind farm circuit solver that was developed by [42]. This is then connected to a model which is based on a generator model which emulates the inertial behaviour of the grid frequency. Using this model the efficacy of any wind farm strategy to provide frequency support, in this work

Chapter 9. Conclusions

the approach developed by [90] has been shown to work as intended. The resultant model is a highly novel research tool as a power system model with this level of fidelity has not been included in a wind farm model with Strathfarm's level of fidelity.

In chapter 8 a Gaussian wake model is developed to update Strathfarm with the most up to date engineering model for wakes. The motivation for this chapter was so that Strathfarm could participate in the Farmconners Benchmarking exercise, a collaborative project between different research groups who have developed software for modelling wind farms. The result of this was that a Gaussian wake model was developed which could be included in the existing Strathfarm model.

9.1 Future Work

There is a large amount of research areas that could be further explored and expanded upon beyond what is show in this thesis.

9.1.1 Wind Farm Control

As the validation exercise shown in this thesis has shown that approaches for allocating curtailment can lead to significantly decreased DELs for both the blades and the tower in a wind farm. However, the intelligent control allocation saw no change to drivetrain fatigue when compared with other curtailment strategies.

It would also be useful to expand on the work done in [99] on these allocation strategies in a delta control setup. This could be expanded to a scope similar to the validation exercise presented in this thesis. Another area of future research in this area would be to consider a similar approach for another wind turbine, possibly the DTU 10MW wind turbine. As wind turbines are rapidly increasing in size it would be useful to know if the approach of considering a static bending moment of the tower is still a beneficial approach on a turbine with a more complicated torque speed curve.

The Development of an Improved Dispatch Algorithm

There is a clear case for developing the dispatch algorithms presented here to further refine them. The dispatch could be performed dynamically based on the gradient of the static contours of the wind turbine components. This could be done on the basis of using the tower as it is the most important component or on a composite of the tower, blades and drivetrain weighted by some kind of costing function. This dynamic algorithm would allocate changes in curtailment based on the gradient of this composite prioritizing changes to turbines where the gradient is lowest. This could be combined with O&M data to include a method of de-prioritizing curtailment variation on turbine's that are flagged for component damage from inspections. This could also be expanded to consider live price data because if the amount of low frequency fatigue accumulation is known then at times of low electricity prices it could be found to be beneficial to reduce power output, or switch off turbines, if they are accumulating a high level of fatigue at a given time.

Wind farm Set-Point Control Algorithms Allowing for Turbine Overrating

The modelling that was the precursor to this thesis chapter is shown in "Supergen Wind Hub: D3.3" [108]. In that modelling the turbines were not prevented from being allocated requested increases in power output when operating at higher wind speeds. This allowed the turbines to follow the static tower bending moment contour further at above rated wind speeds. By doing this the damage equivalent loads were reduced further than the results shown in this chapter. Allowing this overrating of wind turbines outside of a computational simulation has clear potential issues, such as increased wear on power electronics through over heating, potential risks in proximity to maximum allowable forces on the drivetrain, risks in operating a turbine outside of its design specification ect. . Despite this, these results should not be completely discounted, as the results show that if turbines are allowed this additional operational flexibility there is the potential for reductions in component fatigue

Wind Farm Controller With Price Data Input

Building on the work shown in chapter 6, pricing data could be used in the wind farm control algorithms to set the farm wide set-point using pricing data. This operation would likely not

Chapter 9. Conclusions

lead to increased costs for operators as the benefit in terms of life time extension can exceed the lost earnings from generation. This could also be more beneficial in the future as at times of high solar and wind resource electricity spot prices can go very low, sometimes even becoming negative, meaning that curtailment through strategies like those this thesis presents could actually be economically preferable.

9.1.2 Future PAC developments

The next generation of the PAC should be adapted to have a dynamic calculation of the traffic light flags based on wind turbine operational data. If the soft black limits are often breached sufficiently for the turbine at a given site this should trigger the PAC to automatically increase the buffer between the limits by having tighter soft black limits. This data would allow the PAC to automatically tune these limits for different wind turbines/sites based on their local conditions.

9.1.3 Wake Model

There is scope for a quantification exercise to study the impact of the Gaussian wake model compared with the previous wake model over a representative set of simulations using the batching process in Strathfarm. This could be useful for quantifying differences between older wind farm modelling research and modelling done with the Gaussian wakes as there are clear differences in power and tower bending moments between the models. While this has been investigated in Bladed it has not yet been looked at in a full batched validation in Strathfarm at the wind farm level.

9.1.4 Power System Model

A key area of development required on the power system model is that it needs to be optimized. At present the model is too computationally expensive to run a sufficient number of simulations to make a meaningful analysis of the impact of any ancillary service focussed wind farm control strategy on wind turbine fatigue. The goal of this project of creating a fast model for a power system for use of studying frequency data use in a wind farm controller has been achieved.

Chapter 9. Conclusions

This model is an excellent opportunity for use as an initial starting point for developing further in a variety of research areas.

While this model does work there are several areas that it could be improved in terms of computational efficiency and accuracy in the future.

Improved Usability

The current version of the model requires manual scaling of the 15MW model that this model is based on. This could be improved so that an new user would find the model scaling easier possibly through the GUI that is used for existing Strathfarm setups.

Code Optimization

There are several areas of this code that could be further optimized to improve performance.

The linear solver algorithm could potentially be significantly sped up by considering different mathematical techniques for the solver algorithms such as finding a preconditioning approach that could reduce the computational complexity of the linear matrix. Additionally, there is also scope to optimize the non-linear solver. Another approach to improving the operational speed of the power system model in Strathfarm is to fully discretize the model so that Strathfarm can be run with a discrete time solver approach in Simulink. While a discrete form of the turbine model does exist it is not reliably stable so cannot always be used. Discretizing the turbine model through a Huen's method could provide a more stable solution. However, none of these ideas can be easily implemented and would require dedicated research, development and implementation time so is beyond the scope of this project.

Controller Improvements

The turbine recovery process could be improved so that it does not activate after an inertial event. This could be improved further through the development of a 'release' algorithm which will chose which turbine to release form being overrated into recovery based on each of the turbine's operational state.

There are some issues with the power estimation at the turbine level in priority mode which are resulting in a diminished inertial response. This is likely due to the rapid change in power

Chapter 9. Conclusions

that the turbines are performing.

Comparison of Inertial Response from Different Curtailment Techniques

The results shown here and tested so far have only looked at a wind farm with equally curtailed wind turbines. Research has shown that there are more effective distribution algorithms that can result in lower levels of turbine fatigue which have been included in the wind farm controllers in this project. However testing the dispatch algorithm with those approaches was beyond the scope of this project. but should be looked at in future research.

Power System Model Accuracy

The model could be improved by including several different generators with different response times which would provide a more realistic frequency response trajectory in inertial response situations.

Droop control's impact on turbine component damage equivalent Loads

As is noted in the literature review of this thesis there is a clear gap in the academic literature in the combined modelling of a dynamic power system model and a wind farm model capable of accurately modelling component fatigue.

HVDC Connection

This model could in the future be adapted to include a HVDC link rather than a long AC connection to the power system. This would allow for additional modelling of ancillary service provision such as the approaches detailed in the Generator Response Following series of research papers and the [59] based approach of providing droop and inertial response to the grid.

Combining the Power System Codes

The three codes that comprise the power system model (The Mechanical, electrical generator codes and the power system linking code) could be combined into a single DLL. This would help improve the model's computational efficiency.

References

- [1] Bs en61400-1:2005+a1:2010 - wind turbines. Technical Report BS EN61400-1:2005+A1:2010, European Committee for Electrotechnical Standardization, Avenue Marnix 17, B-1000 Brussels, 2010.
- [2] *Wind power plants : fundamentals, design, construction and operation / [internet resource]*. Springer, Berlin ; Heidelberg, 2nd ed.. edition, 2012.
- [3] Guide to the supergen controllers. Technical Report BBS EN 61400-13-2016, University of Strathclyde, internal document., 2014. internal document.
- [4] Bbs en 61400-13-2016 - wind turbines. Technical Report BBS EN 61400-13-2016, European Committee for Electrotechnical Standardization, Avenue Marnix 17, B-1000 Brussels, 2016.
- [5] *1110-2019 - iee guide for synchronous generator modeling practices and parameter verification with applications in power system stability analyses*. IEEE, 2020.
- [6] Ahmad Shabir Ahmadyar and Gregor Verbič. Coordinated operation strategy of wind farms for frequency control by exploring wake interaction. *IEEE Transactions on Sustainable Energy*, 8(1):230–238, 2017.
- [7] Hasan Alrajhi Alsiraji and Ramadan El-Shatshat. Comprehensive assessment of virtual synchronous machine based voltage source converter controllers. *IET Generation, Transmission & Distribution*, 11:1762–1769(7), May 2017.
- [8] Müfit Altın, Ömer Göksu, Remus Teodorescu, Pedro Rodriguez, Birgitte-Bak Jensen, and Lars Helle. Overview of recent grid codes for wind power integration. In *2010*

References

- 12th International Conference on Optimization of Electrical and Electronic Equipment*, pages 1152–1160, 2010.
- [9] Lindsey Amos. Application of wind farm control and wind farm layout for the optimisation of wind farm power and loads. Technical report, University of Strathclyde, Glasgow, 2017. internal document.
- [10] Lindsey Amos. Application of wind farm control and wind farm layout for the optimisation of wind farm power and loads. Technical report, University of Strathclyde, Glasgow, 2018. internal document.
- [11] Lindsey Amos. Strathturb - university of strathclyde control-oriented wind turbine model, 2019.
- [12] P Argyle, S Watson, C Montavon, I Jones, and M Smith. Modelling turbulence intensity within a large offshore wind farm. *Wiley Wind Energy*, page 1329–1343, 06 2018.
- [13] A. B. Attya, O. Anaya-Lara, and W. E. Leithead. Novel concept of renewables association with synchronous generation for enhancing the provision of ancillary services. *Applied Energy*, 229:1035–1047, 11 2018.
- [14] A.B. Attya, J.L. Dominguez-Garcia, and O. Anaya-Lara. A review on frequency support provision by wind power plants: Current and future challenges. *Renewable and Sustainable Energy Reviews*, 81:2071–2087, 2018.
- [15] Majid Bastankhah and Fernando Porté-Agel. A new analytical model for wind-turbine wakes. *Renewable Energy*, 70(C):116–123, 2014.
- [16] M. P. Bhawalkar, N. Gopalakrishnan, and Y. P. Nerkar. A review of indian grid codes for wind power generation. *Journal of The Institution of Engineers (India): Series B*, 100(6):609–617, 2019.
- [17] S. Boersma, B. Doekemeijer, M. Vali, J. Meyers, and J.-W. van Wingerden. control-oriented dynamic wind farm model: Wfsim. *Wind Energy Science*, 3(1):75–95, 03 2018.
- [18] Sigrid Bolik. *Modelling and Analysis of Variable Speed Wind Turbines with Induction Generator during Grid Fault*. PhD thesis, Aalborg University, 2004.

References

- [19] BSI. *BS EN 61400-1:2005+A1:2010*. BSI, 2016.
- [20] David Campos-Gaona, Adam Stock, Olimpo Anaya-Lara, and William Leithead. Dynamic wind power plant control for system integration using the generator response following concept. *Energies*, 13(7), 2020.
- [21] P. Chatzopoulos. *Full Envelope Wind Turbine Controller Design for Power Regulation and Tower Load Reduction*. University of Strathclyde., Glasgow, 2011. PhD Thesis.
- [22] Zhenyu Chen, Jizhen Liu, Zhongwei Lin, and Zhenqing Duan. Closed-loop active power control of wind farm based on frequency domain analysis. *Electric Power Systems Research*, 170:13–24, 2019.
- [23] Zhenyu Chen, Jizhen Liu, Zhongwei Lin, and Chenzhi Qu. Variable-constrained model predictive control of coordinated active power distribution for wind-turbine cluster. *Applied Sciences*, 9(1):112, 2018.
- [24] Jingchun Chu, Ling Yuan, Zhongwei Lin, Wei Xu, and Zhenyu Chen. Receding horizon optimization of wind farm active power distribution with power tracking error and transmission loss. *Processes*, 6(12):259, 2018.
- [25] Matthew Cole, David Campos-Gaona, Adam Stock, and Marcel Nedd. A critical review of current and future options for wind farm participation in ancillary service provision. *Energies*, 16(3), 2023.
- [26] Matthew Cole, Adam Stock, David Campos-Gaona, and Michael Smailes. Development of a power system model for strathfarm. January 2023. EERA DeepWind 2023 ; Conference date: 18-01-2023 Through 20-01-2023.
- [27] Matthew Cole, Adam Stock, W.E. Leithead, and Lindsey Amos. Using rotor inertia as stored energy in below rated wind farms to provide primary frequency response. *Journal of Physics: Conference Series*, 1618(2), September 2020. Science of Making Torque from Wind 2020, TORQUE 2020 ; Conference date: 28-09-2020 Through 02-10-2020.

References

- [28] Aberdeenshire Council. Aberdeenshire her - no66ne0258 - blyth house, marykirk. <https://online.aberdeenshire.gov.uk/smrpub/master/detail.aspx?refno=NO66NE0258>, 2021. [Accessed 06-Nov-2022].
- [29] Francisco Díaz-González, Melanie Hau, Andreas Sumper, and Oriol Gomis-Bellmunt. Participation of wind power plants in system frequency control: Review of grid code requirements and control methods. *Renewable and Sustainable Energy Reviews*, 34:551–564, 2014.
- [30] Havang-Wikimedia Commons Public Domain. Sint hubert, heimolen. https://commons.wikimedia.org/wiki/File:St.Hubert,_Heimolen.JPG, 2007. [Accessed 10-Nov-2022].
- [31] Wikimedia Commons Public Domain. Wind turbine 1891 james blyth. <https://upload.wikimedia.org/wikipedia/commons/1/13/James`Blyth%27s`1891`windmill.jpg>, 1891. [Accessed 05-Nov-2022].
- [32] N. Dowling. *Mechanical behavior of materials*. Boston: Pearson., 2013.
- [33] EirGrid. *EirGrid Grid Code version 10*. EirGrid, 2021.
- [34] ENERGINET. *Technical regulation 3.2.5 for wind power plants above 11 kW*. ENERGINET, 2017.
- [35] I Erlich and M Wilch. Primary frequency control by wind turbines. In *IEEE PES General Meeting*, pages 1–8, 2010.
- [36] Boris Fischer and Martin Shan. A survey on control methods for the mitigation of tower loads. 09 2013.
- [37] S. Frandsen. On the wind speed reduction in the center of large clusters of wind turbines. *Journal of Wind Engineering and Industrial Aerodynamics*, 39:251–265, 1992.
- [38] Sten Frandsen, Rebecca Barthelmie, Sara Pryor, Ole Rathmann, Søren Larsen, Jørgen Højstrup, and Morten Thøgersen. Analytical modelling of wind speed deficit in large offshore wind farms. *Wind Energy*, 9(1-2):39–53, 2006.

References

- [39] Maria Lourdes Gala Santos. *Aerodynamics and wind-field models for wind turbine control*. University of Strathclyde, Glasgow, 2018. PhD Thesis.
- [40] Juan Gallego-Calderon and Anand Natarajan. Assessment of wind turbine drive-train fatigue loads under torsional excitation. *Engineering Structures*, 103(9):189–202, 2015.
- [41] Vahid Gholamrezaie, Mehdi Ghazavi Dozein, Hassan Monsef, and Bin Wu. An optimal frequency control method through a dynamic load frequency control (lfc) model incorporating wind farm. *IEEE Systems Journal*, 12(1):392–401, 2018.
- [42] A. Giles. *State-space modelling of multi-converter power systems*. University of Strathclyde, Glasgow, 2018. PhD Thesis.
- [43] Alexander Giles, Daan van der Hoek, Gabriele Bedon, Karl Merz, Martin Kühn, Mehdi Vali, Olimpo Anaya-Lara, Stoyan Kanev, Vlaho Petrović, Davide Trabucchi, and William Leithead. *Integrated Research Programme on Wind Energy: Offshore array control Work Package 6.3 - Deliverable number 63.4*. University of Strathclyde, February 2018.
- [44] Paul Gipe and Erik Möllerström. An overview of the history of wind turbine development: Part ii—the 1970s onward. *Wind Engineering*, 47(1):220–248, 2023.
- [45] Tuhfe Göçmen, Filippo Campagnolo, Thomas Duc, Irene Eguinoa, Søren Juhl Andersen, Vlaho Petrović, Lejla Imširović, Robert Braunbehrens, Ju Feng, Jaime Liew, Mads Baungaard, Maarten Paul van der Laan, Guowei Qian, Maria Aparicio-Sanchez, Rubén González-Lope, Vinit Dighe, Marcus Becker, Maarten van den Broek, Jan-Willem van Wingerden, Adam Stock, Matthew Cole, Renzo Ruisi, Ervin Bossanyi, Niklas Requate, Simon Strnad, Jonas Schmidt, Lukas Vollmer, Frédéric Blondel, Ishaan Sood, and Johan Meyers. Farmconners wind farm flow control benchmark - part 1: blind test results. *Wind Energy Science*, 7(5):1791–1825, September 2022.
- [46] I A Gowaid, A. El-Zawawi, and M. El-Gammal. Improved inertia and frequency support from grid-connected dfig wind farms: 2011 ieee/pes power systems conference and exposition. In *2011 IEEE/PES Power Systems Conference and Exposition*, pages 1–9. IEEE, mar 2011. 2011 IEEE PES Power Systems Conference and Exposition, PSCE 2011 ; Conference date: 20-03-2011 Through 23-03-2011.

References

- [47] National Grid. Black start from non-traditional generation technologies. Technical report, National Grid, 2019.
- [48] National Grid. *Guidance Notes for Power Park Modules*. National Grid, 2019.
- [49] National Grid. *THE GRID CODE, ISSUE 5, REVISION 38*. National Grid, 2019.
- [50] National Grid. *Frequency Risk and Control Report Security & Quality of Supply Standard*. National Grid, 2022.
- [51] Nicola Grieve, Abbas Mehrad Kazemi Amiri, and William E. Leithead. A straightforward approach to site-wide assessment of wind turbine tower lifetime extension potential. *Energies*, 15(9), May 2022.
- [52] E Hart, M Keegan, and D Mcmillan. A lookup table approach to determining wind turbine operational fatigue loading from wind field measurements. 2016.
- [53] Sheng Huang, Qiuwei Wu, Yifei Guo, and Zhongwei Lin. Bi-level decentralised active power control for large-scale wind farm cluster. *IET Renewable Power Generation*, 12(13):1486–1492, 2018.
- [54] S Hur and W.E. Leithead. Adjustment of wind farm power output through flexible turbine operation using wind farm control. *Wind Energy*, 19(9):1667–1686, 2016.
- [55] S H. Hur and W.E. Leithead. Control oriented modelling of a wind turbine and farm. *J. Phys.: Conf. Ser.* 1037 062020, 2018.
- [56] Halima Ikaouassen, Abderraouf Raddaoui, Miloud Rezkallah, and Hussein Ibrahim. Improved predictive current model control based on adaptive pr controller for standalone system based dg set. *International journal of electrical and computer engineering (Malacca, Malacca)*, 10(2):1905–1914, 2020.
- [57] N.O. Jensen. A note on wind generator interaction, 1983.
- [58] J M Jonkman, J Annoni, G Hayman, B Jonkman, and A Purkayastha. Development of fast.farm: A new multi-physics engineering tool for wind-farm design and analysis. *35th Wind Energy Symposium*, 2017.

References

- [59] A Junyent-Ferre, Y Pipelzadeh, and T C Green. Blending hvdc-link energy storage and offshore wind turbine inertia for fast frequency response. *Sustainable Energy*, vol. 6, no. 3, page 1059–1066, 07 2015.
- [60] Christian Kanzow, Nobuo Yamashita, and Masao Fukushima. Levenberg-marquardt methods with strong local convergence properties for solving nonlinear equations with convex constraints. *Journal of Computational and Applied Mathematics*, 172(2):375–397, 2004.
- [61] Hazem Karbouj, Zakir Hussain Rather, Damian Flynn, and Hassan W. Qazi. Non-synchronous fast frequency reserves in renewable energy integrated power systems: A critical review. *International Journal of Electrical Power & Energy Systems*, 106:488–501, 2019.
- [62] Tariq Khwaja and Azer Reza. Power transmittance of a laterally shifted gaussian beam through a circular aperture. 05 2016.
- [63] T. Knudsen, T. Bak, and M. Svenstrup. Survey of wind farm control-power and fatigue optimization. *Wind Energy*, 18(8):1333–1351, 2014.
- [64] Peng Kou, Deliang Liang, Linbo Yu, and Lin Gao. Nonlinear model predictive control of wind farm for system frequency support. *IEEE Transactions on Power Systems*, 34(5):3547–3561, 2019.
- [65] Velissarios Kourkoulis. *Wind farm control and modelling to provide grid primary response and meet OandM requirements*. University of Strathclyde, Glasgow, 2019. PhD Thesis.
- [66] P Kundur. *Power system stability and control*. McGraw Hill, 25 edition.
- [67] John Langdon and Martin Watts. Tower windmills in medieval england: A case of arrested development? *Technology and Culture*, 46(4):697–718, 2005.
- [68] W. E. Leithead, Victoria Neilson, and S. Dominguez. Alleviation of unbalanced rotor loads by single blade controllers. In *European Wind Energy Conference & Exhibition 2009*, pages 576–615. Curran Associates, Inc., Marseilles, France, March 2009.

References

- [69] Wei Li, Dean Kong, Qiang Xu, Xiaoyu Wang, Xiang Zhao, Yongji Li, Hongzhi Han, Wei Wang, and Zhenyu Chen. A wind farm active power dispatch strategy considering the wind turbine power-tracking characteristic via model predictive control. *Processes*, 7(8):530, 2019.
- [70] British Library. James blyth and the world's first wind-powered generator. <https://blogs.bl.uk/science/2017/08/james-blyth-and-the-worlds-first-wind-powered-generator.html>, 2017. [Accessed 06-Nov-2022].
- [71] Zhongwei Lin, Zhenyu Chen, Chenzhi Qu, Yifei Guo, Jizhen Liu, and Qiuwei Wu. A hierarchical clustering-based optimization strategy for active power dispatch of large-scale wind farm. *International Journal of Electrical Power & Energy Systems*, 121:106155, 2020.
- [72] Liu, Wang, Wang, Zhu, and Lio. Active power dispatch for supporting grid frequency regulation in wind farms considering fatigue load. *Energies*, 12(8):1508, 2019.
- [73] Tongchui Liu, Wenxia Pan, Rui Quan, and Mingyang Liu. A variable droop frequency control strategy for wind farms that considers optimal rotor kinetic energy. *IEEE Access*, 7:68636–68645, 2019.
- [74] L Lu and N A Cutululis. Virtual synchronous machine control for wind turbines: a review. *Journal of Physics: Conference Series*, 1356(1):012028, oct 2019.
- [75] Shaokang Ma, Hua Geng, Geng Yang, and Bikash C. Pal. Clustering-based coordinated control of large-scale wind farm for power system frequency support. *IEEE Transactions on Sustainable Energy*, 9(4):1555–1564, 2018.
- [76] D. Madjidian. Scalable minimum fatigue control of dispatchable wind farms. *Wind Energy*, 19(10):1933–1944, 01 2016.
- [77] H A Madsen, G. C. Larsen, T. J. Larsen, N. Troldborg, and R. Mikkelsen. Calibration and validation of the dynamic wake meandering model for implementation in an aeroelastic code. *Journal of Solar Energy Engineering*, 132(4):41014, 2010.

References

- [78] J. F Manwell. *Wind energy explained [internet resource] : theory, design and application*. 2nd ed.. edition, 2009.
- [79] MathWorks. <https://mathworks.com/help/phymod/sps/powersys/ref/synchronouismachinepustandard.html>, 2022.
- [80] MathWorks. <https://uk.mathworks.com/help/phymod/sps/ug/24-hour-simulation-of-a-vehicle-to-grid-v2g-system.html>, 2022.
- [81] Lucas Bauer & Silvio Matysik. Weg wind energygroup ls1. <https://en.wind-turbine-models.com/turbines/1020-weg-wind-energygroup-ls1>. [Accessed 06-May-2023].
- [82] R. McGill, R. Torres-Olguin, O. Anaya-Lara, and W.E. Leithead. Generator response following as a primary frequency response control strategy for vsc-hvdc connected offshore wind farms. *Energy Procedia*, 137:108–118, 10 2017.
- [83] Mansour Mohseni and Syed M. Islam. Review of international grid codes for wind power integration: Diversity, technology and a case for global standard. *Renewable and Sustainable Energy Reviews*, 16(6):3876–3890, 2012.
- [84] Marcel Nedd, Jethro Browell, Agusti Egea-Alvarez, Keith Bell, Robert Hamilton, Shuren Wang, and Susan Brush. Operating a zero carbon gb power system in 2025 : Frequency and fault current [annexes - review of system and network issues, frequency stability, power electronic devices and fault current, & market needs], December 2020.
- [85] V. Neilson. *Individual Blade Control for Fatigue Load Reduction of Large-scaled Wind Turbines: Theory and Modelling*. University of Strathclyde, Glasgow, 2010. Masters Thesis.
- [86] Amin Niayifar and Fernando Porté-Agel. Analytical modeling of wind farms: A new approach for power prediction. *Energies*, 9(9), 2016.
- [87] Elis Nycander and Lennart Söder. Review of european grid codes for wind farms and their implications for wind power curtailments. In *17th International Wind Integration Workshop Stockholm, Sweden — 17 – 19 October 2018* :, 2018. QC 20181023.

References

- [88] US Department of Energy. Wind turbines: the bigger, the better. <https://www.energy.gov/eere/articles/wind-turbines-bigger-better>, 2022. [Accessed 06-Nov-2022].
- [89] A Peña, P-E Réthoré, and M P van der Laan. On the application of the jensen wake model using a turbulence-dependent wake decay coefficient: the sexbierum case. *Wind Energy*, vol. 19, no. 4, page 763–776, 05 2015.
- [90] S Poushpas. *Wind Farm Simulation Modelling and Control*. University of Strathclyde, Glasgow, 2016. PhD Thesis.
- [91] Trevor J. Price. Uk large-scale wind power programme from 1970 to 1990: The carmarthen bay experiments and the musgrove vertical-axis turbines. *Wind Engineering*, 30(3):225–242, 2006.
- [92] IEEE Committee Report. Computer representation of excitation systems. *IEEE Transactions on Power Apparatus and Systems*, PAS-87(6):1460–1464, 1968.
- [93] A. Roscoe, T. Knueppel, Brogan Da Silva, R., I. P. and Gutierrez, D. Elliott, and J. Perez Champion. Response of a grid forming wind farm to system events, and the impact of external and internal damping. *IET Renewable Power Generation*, 14(19):3908–3917, 2020.
- [94] Jonas Schmutz and Msc Michael Koller. Primary frequency control provided by battery, 2013.
- [95] S Siniscalchi-Minn, F D Bianchia, M De-Prada-Gil, and C Ocampo-Martinez. A wind farm control strategy for power reserve maximization. *Elsevier Renewable energy* 131, pages 37–43, 01 2019.
- [96] Preeti Sonkar and OP Rahi. Contribution of wind power plants in grid frequency regulation: Current perspective and future challenges. *Wind Engineering*, 45(2):442–456, 2020.
- [97] A. Stock, W.E. Leithead, and Shona Pennock. Providing frequency droop control using variable speed wind turbines with augmented control. European Wind Energy Associ-

References

- ation Conference and Exhibition 2014, EWEA 2014. European Wind Energy Association, 2014.
- [98] Adam Stock. *Augmented Control for Flexible Operation of Wind Turbines*. University of Strathclyde, Glasgow, 2015. PhD Thesis.
- [99] Adam Stock, Matthew Cole, W E Leithead, and Lindsey Amos. Distributed control of wind farm power set points to minimise fatigue loads. American Control Conference (ACC) 2020. American Control Conference (ACC), 2020.
- [100] Adam Stock and William Leithead. A generic approach to wind farm control and the power adjusting controller. *Wind Energy*, 25(10):1735–1757, 2022.
- [101] Adam Stock, William Leithead, Olimpo Anaya-Lara, Lindsey Amos, Matthew Cole, Peter Taylor, Paul Pirrie, and David Campos-Gaona. Wind farm control work at the university of strathclyde, past, present and future. October 2019. 5th Wind Energy Systems Engineering Workshop, WESE2019 ; Conference date: 02-10-2019 Through 04-10-2019.
- [102] Khagendra Bahadur Thapa and Kishan Jayasawal. Pitch control scheme for rapid active power control of a pmsg-based wind power plant. *IEEE Transactions on Industry Applications*, 56(6):6756–6766, 2020.
- [103] David William Thompson. Mitigating size related limitations in wind turbine control, 2018. PhD Thesis.
- [104] J.-W. van Wingerden, L. Pao, J. Aho, and P. Fleming. ‘active power control of waked wind farms. *IFAC-PapersOnLine*, 50(1):4484–4491, 07 2017.
- [105] Richard P. Walker. *Wind energy essentials : societal, economic, and environmental impacts / [internet resource]*. 2015.
- [106] Zhen Wang, Kit Po Wong, S.S. Choi, Deqiang Gan, and Yi Zong. 1 - an overview of codes and control strategies for frequency regulation in wind power generation. In Qiang Yang, Ting Yang, and Wei Li, editors, *Smart Power Distribution Systems*, pages 3–20. Academic Press, 2019.

References

- [107] Wind Energy & Control Group, Sung Ho Hur, Saman Poushpas, Lindsey Amos, and Bill Leithead. *Supergen Wind Hub: D3.1 Report of Wind Farm Modelling*. University of Strathclyde, September 2019.
- [108] Wind Energy & Control Group, Romans Kazacoks, Matthew Cole, and Bill Leithead. *Supergen Wind Hub: D3.3 Report on Wind Farm Control Algorithms to Meet Asset Management Requirements*. University of Strathclyde, September 2019.
- [109] Qi Yao, Jizhen Liu, and Yang Hu. Optimized active power dispatching strategy considering fatigue load of wind turbines during de-loading operation. *IEEE Access*, 7:17439–17449, 2019.
- [110] Yong Zhang, Zhengang Duan, and Xuelian Liu. Comparison of grid code requirements with wind turbine in china and europe. In *2010 Asia-Pacific Power and Energy Engineering Conference*, pages 1–4, 2010.
- [111] H. Zhao, Q. Wu, S. Huang, M. Shahidehpour, Q. Guo, and H. Sun. Fatigue load sensitivity-based optimal active power dispatch for wind farms. *IEEE Transactions on Sustainable Energy*, 8(3):1247–1259, 07 2017.

Appendix A

Using Rotor Inertia as Stored Energy in Below Rated Wind Farms to Provide Primary Frequency Response

This Appendix contains a conference paper published in Journal of Physics from the Torque 2020 conference [27]. This work was based on a version of Strathfarm developed prior to the inclusion of the Gaussian wake model or the power system model.

A.1 Introduction

As wind power increases its share of grid penetration in power grids around the world, there is an increased requirement for wind power to not just generate as the wind speed dictates, but also to provide additional services such as frequency response. While primary and secondary frequency response are not currently mandatory for wind farms in the UK, wind farm operators can bid for provision of firm frequency response, which requires them to provide both primary and secondary responses. This may change in the future as the number of synchronous generators is reduced as the grid is decarbonised. The aim of this paper is to investigate a number of options for control strategies to provide primary response using the rotational stored energy in the rotors of a wind farm for solely primary response provision.

The requirement for primary response in Great Britain is to increase power output from

Appendix A. Using Rotor Inertia as Stored Energy in Below Rated Wind Farms to Provide Primary Frequency Response

a plant for 30 seconds to increase the grid frequency. The grid code states that at least a "frequency response of 10% of Registered Capacity achievable" [48] [49]. A reserve to enable this increase is normally held by derating the wind turbines, but the machines have some stored energy in the kinetic inertia of their rotors which is not normally considered.

Strathfarm, the University of Strathclyde's in-house wind farm modelling software, is used to simulate the wind farm and assess the performance of various farm level control strategies. The main advantage of Strathfarm is that it can be run on a standard desktop PC, simulating up to 100 wind turbines in real time. The control strategies are implemented in Strathfarm by means of a farm level controller the architecture of which allows each turbine to have a high level of operational autonomy.

This autonomy is provided by the addition of a Power Adjusting Controller (PAC) [98] to each wind turbine controller. The PAC acts as an augmentation to the control structure which allows a turbine's generated power to be adjusted continuously through a combination of changes to both the torque and pitch demand, without compromising the wind turbine's full envelope controller. The feedback action of the turbine's full envelope controller is unaffected by the presence of the PAC. The PAC jackets the full envelope controller as can be seen in Figure A.1. In addition, the PAC provides operational information to the wind farm controller (WFC), generally in the form of binary flags, e.g. providing information regarding the displacement of the turbine from its normal operational strategy, but sometimes in the form of a numeric value such as an estimate of wind speed.

Appendix A. Using Rotor Inertia as Stored Energy in Below Rated Wind Farms to Provide Primary Frequency Response

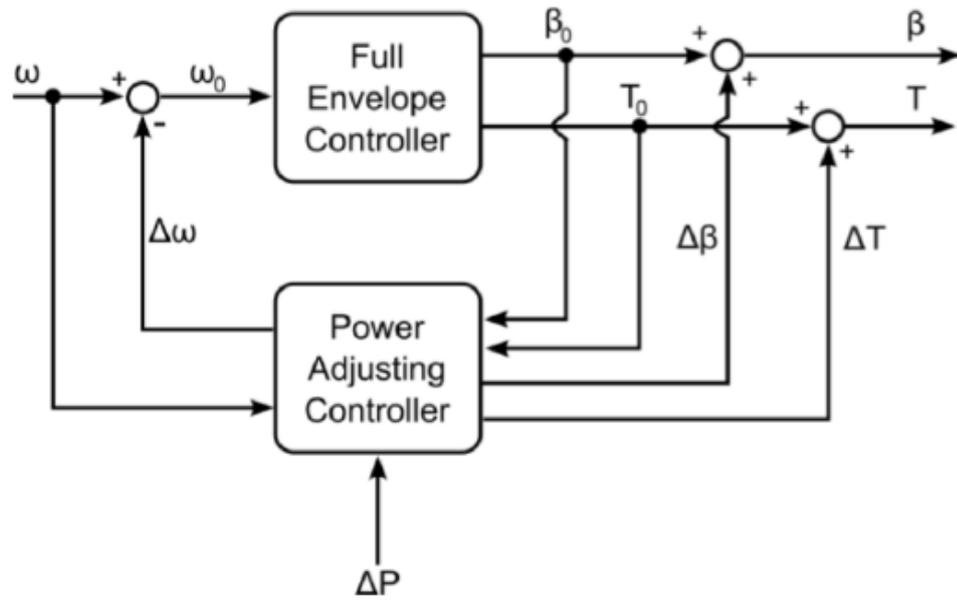


Figure A.1: The general form of the Power Adjusting Controller (PAC) [98]

The purpose of this work is to explore the provision of primary ancillary services to the grid. The first requirement is to minimise the reduction in power output from the turbines in a wind farm by utilising the stored energy in the turbine rotors. By beginning with a turbine level proof of concept and explanation of the control strategy for extracting energy from the turbine's rotor, a cost analysis demonstrates that this approach for holding spinning reserve is worth pursuing further. Subsequently, a farm level strategy can be compared to two turbine level approaches, comparing damage equivalent loads (DELs) of both turbine towers and blades and energy capture over a range of mean wind speeds, wind directions and turbulence intensities. Previous research has been done before in this area, particularly at the wind turbine level [35] [98] and for providing droop control [97], but the work presented here is focused more in the impact on DELs than the actual delivering of primary response.

A.2 Methodology

A.2.1 Wind Turbine level Approach

The first step in the development of this approach is to explore the limits of how much energy can be extracted from a turbine's rotor in the 30 second period when primary response is provided. The amount of stored energy in the rotor is given by:

$$E = \frac{1}{2}J(\omega_0^2 - \omega^2) \quad (\text{A.1})$$

Where:

- E is the energy stored in the rotor,
- J is the inertia of the rotor and drivetrain,
- ω_0 is the initial rotational speed of the generator.
- ω is the new rotational speed of the generator.

Equation 1 shows that the amount of energy which can be extracted from a turbine's rotor is not strictly governed by the wind speed, but instead by the difference in the square of the rotor speeds.

The PAC ensures safe operation of the turbine either by setting limits to any requested change to generated power from the farm level controller or by setting mandatory limits on the operating state of the turbine. The former limits are referred to as soft limits, as the turbine state can still cross them. The latter are referred to as hard limits, as the PAC acts to prevent the turbine state from crossing them. Information regarding which soft and hard limits apply at any given time is communicated to the farm level controller by binary flags. Examples of soft limits are traffic light boundaries, the green, amber and red boundaries in Figure 2, and an example of a hard limit is the black boundary, depicted on Figure 2. Within the green boundary, the turbine operating state is furthest from the unsafe region and the limits on requested changes to generated power are most relaxed.

By using steady state wind fields in the turbine model, the amount of energy which can be extracted for 30 seconds while remaining inside the PAC's green flag boundary has been found

Appendix A. Using Rotor Inertia as Stored Energy in Below Rated Wind Farms to Provide Primary Frequency Response

for wind speeds between 7 and 11 m/s. Following this, energy extraction from the rotor, the turbine enters a recovery mode which returns the turbine to its uncurtailed operational strategy.

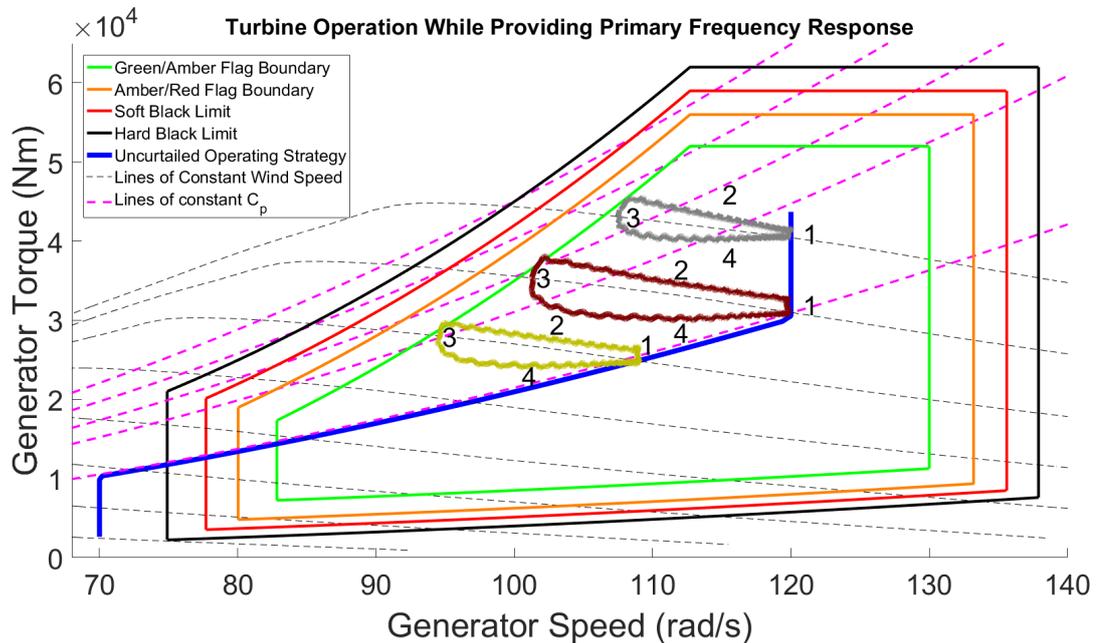


Figure A.2: A torque-speed diagram of a wind turbine extracting energy from its rotor for 30 seconds at 9, 10 and 11 m/s.

Figure A.2 shows the operation of the turbine in the torque-speed plane of the turbine as energy is extracted from the rotor with the PAC's traffic light flag boundaries, the turbine's lines of constant wind speed between 4m/s and 11m/s and the turbine's lines of constant C_p from 100% to 92% at 2% intervals all shown. The extraction process has been numbered for clarity, with 1 showing the initial position, 2 showing the 30 second extraction period, 3 showing the switching from extraction to recovery and 4 showing the recovery of the turbine to its normal operational strategy. The magenta dashed lines show efficiency curves at 2% intervals beginning at 100% of the maximum C_p .

For the three simulations shown the increase in power output for the 30 second period is 180kW, 240 kW and 100kW for the constant wind speeds of 9 m/s, 10m/s and 11 m/s respectively. When this approach is used the wind turbines operate in a less aerodynamically efficient location of the torque-speed plane; that is, there is a reduction in C_p and so energy capture. This reduction in C_p is particularly noticeable at 11 m/s, as is shown in Figure A.3

Appendix A. Using Rotor Inertia as Stored Energy in Below Rated Wind Farms to Provide Primary Frequency Response

and A.4. Even though the 11 m/s plot indicates a smaller reduction in tip speed ratio (λ) it has the largest relative reduction in its C_p value.

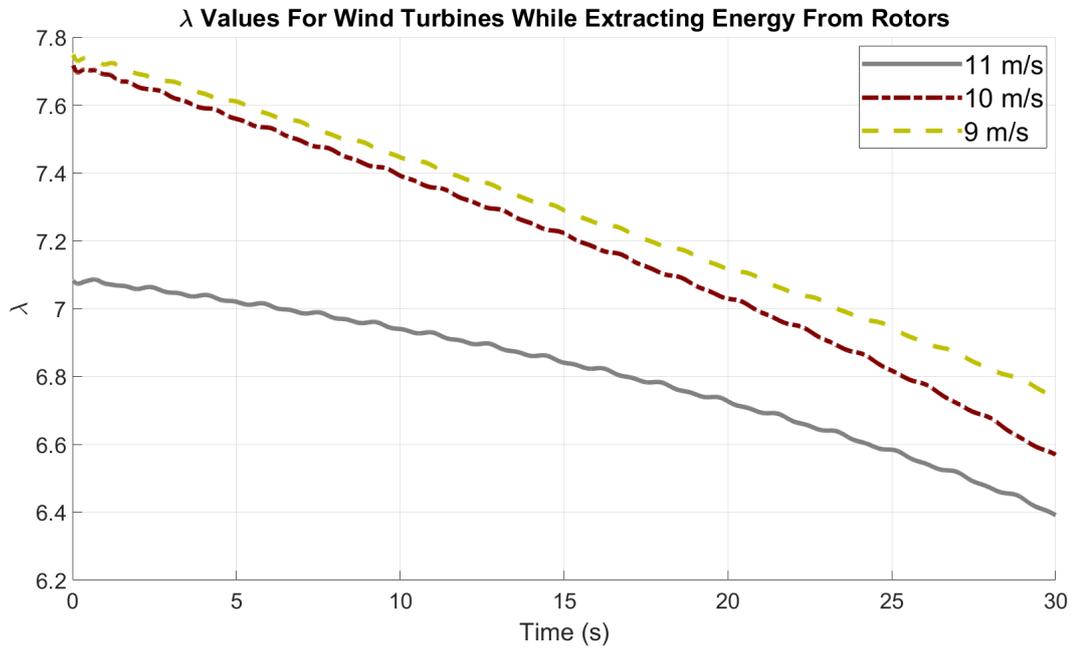


Figure A.3: The tip speed ratio of a wind turbine which is extracting energy from its rotor for 30 seconds at 9, 10 and 11 m/s.

Appendix A. Using Rotor Inertia as Stored Energy in Below Rated Wind Farms to Provide Primary Frequency Response

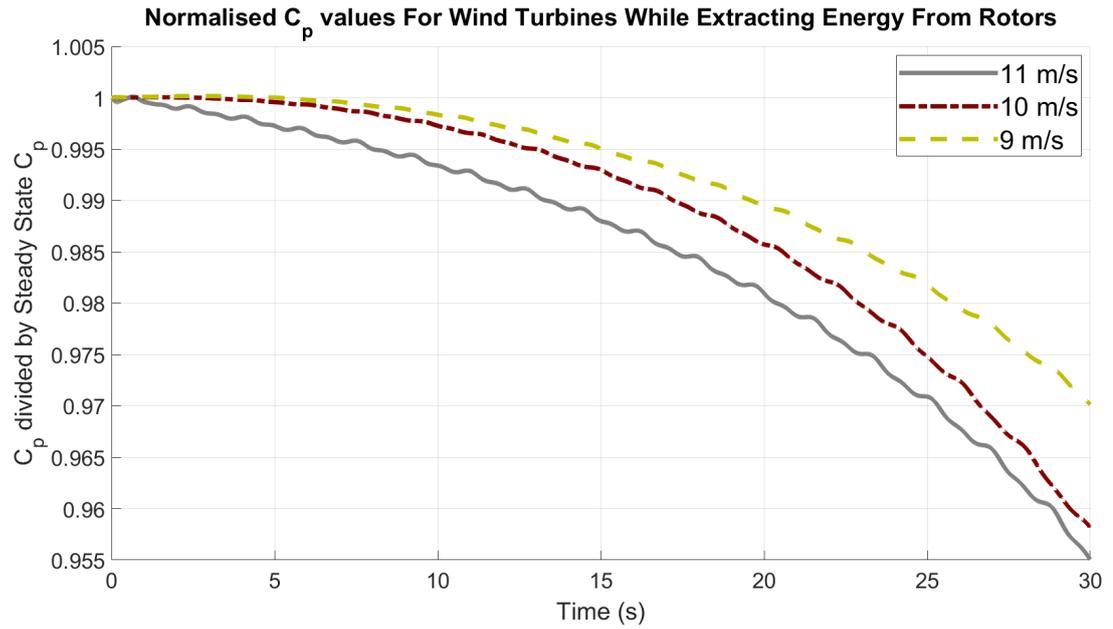


Figure A.4: The ratio of the turbine level C_p divided by the steady state C_p as energy is extracted from the turbine's rotor.

Figure A.4 shows the normalised C_p of the turbine as energy is extracted from its rotor. Due to the flatness of the peak of the $C_p - \lambda$ curve near its peak there is very little change in the C_p values initially at 9 or 10 m/s. This flatness can be seen in the efficiency curves shown in Figure (A.2) as the gap between the 100% and 98% curves is significantly larger than that of the other curves.

Wind Speed (m/s)	Rotor Inertia Reserve [kW]	Percent of Uncurtailed Power	Additional Curtailment Allocation [kW]	Total requested Reserve Power for Farm Level Approach [kW]
7	25	2.1%	0	25
7.5	75	5%	0	75
8	100	5.6%	0	100
8.5	140	6.5%	0	140
9	180	7.2%	0	180
9.5	220	7.3%	0	220
10	240	6.9%	120	360
10.5	170	3.8%	280	450
11	100	2.2%	370	470

Table A.1: Maximum sustainable power extraction from the rotor for 30 seconds and allocated turbine level curtailment by wind speed.

Appendix A. Using Rotor Inertia as Stored Energy in Below Rated Wind Farms to Provide Primary Frequency Response

Table A.1 shows the increase in turbine power output which can be sustained for 30 seconds by wind speed while staying within safe operational limits. When a farm level approach is considered an additional curtailment through pitching is added to the curtailment total as a way to avoid higher tower loads near to the rated wind speed and to allocate the power reduction to turbines which are more likely to be able to provide the curtailment request without crossing a flag boundary in the PAC.

By considering the costings of this strategy on an individual turbine level it can be seen that there is a possible case for this new approach. If the approach shown in Table A.1 is compared to curtailing a turbine by 10% of its power output on an annualised basis using a Weibull distributed wind speed, with a mean wind speed of 10 m/s, it will have an energy capture 402438 kWh higher than curtailing a turbine by 10% if this strategy were to be used at all times of operation. If a constant energy price of £50 per MWh is assumed this will mean that a turbine using this proposed approach will increase revenue by £20122 each year.

A.2.2 Farm Level implementation

Three strategies for holding a level of reserve power are tested here, two turbine level approaches and one farm level approach.

The reduction in power allocated to each turbine for the three strategies are given by:

$$\Delta p_i = -0.1 \times P_i \quad (\text{A.2})$$

for always curtailing by 10%,

$$\Delta p_i = -0.1 \times P_i + R_{WT}(v)_i \quad (\text{A.3})$$

for implementing the use of stored energy at the turbine level and

$$\Delta p_i = R_{WF}(v)_i + \frac{\sum_{i=1}^n (-0.1 \times P_i + R_{WF}(v)_i)}{n} \quad (\text{A.4})$$

for implementing the use of stored energy at the farm level, where:

- Δp_i - is the requested reduction of power for the i th turbine,

Appendix A. Using Rotor Inertia as Stored Energy in Below Rated Wind Farms to Provide Primary Frequency Response

- P_i - is the uncurtailed power of the i th turbine,
- $R_{WT}(v)_i$ - is the turbine level curtailment from Table 1 of the i th turbine as a function of the estimated wind speed v ,
- $R_{WF}(v)_i$ - is the total curtailment from both the energy in the rotors and additional curtailment from pitching shown in column 2 and 4 respectively in Table 1 of the i th turbine as a function of the estimated wind speed v ,
- n - is the total number of turbines in the farm.

To investigate the effectiveness of these approaches 240 wind fields have been created which each last 900 seconds, the strategies were implemented from 150 seconds to allow for the dissipation of transients in the turbine models. The DELs and energy capture were calculated from 350 seconds to allow for the propagation of wakes through the wind farm. The DELs were calculated using the approach described by [52].

For the simulations mean wind speeds of 8.5, 9, 9.5, 10 and 10.5 m/s were used, wind directions of 0, 15, 30 and 45 degrees and turbulence intensities (TI) of 5% and 10%. In addition to these each combination of parameters were used to generate wind fields six times.

The wind farm layout used can be seen in Figure A.5, the turbines are spaced 1000 meters (8 diameters apart) in a 4 by 4 square of 16 turbines. The wind angles of 0, 15, 30 and 45 degrees were chosen as the square layout means that one eighth of a rotation is sufficient due to a square having eight symmetries.

Appendix A. Using Rotor Inertia as Stored Energy in Below Rated Wind Farms to Provide Primary Frequency Response

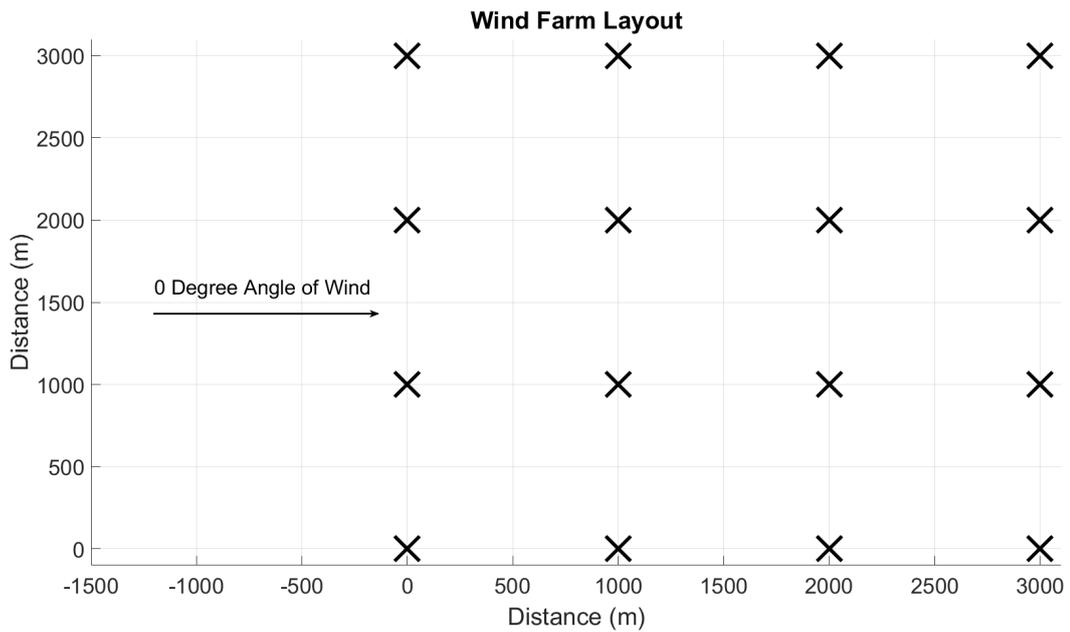


Figure A.5: The layout of the wind farm used for the simulations presented here.

A.3 Results

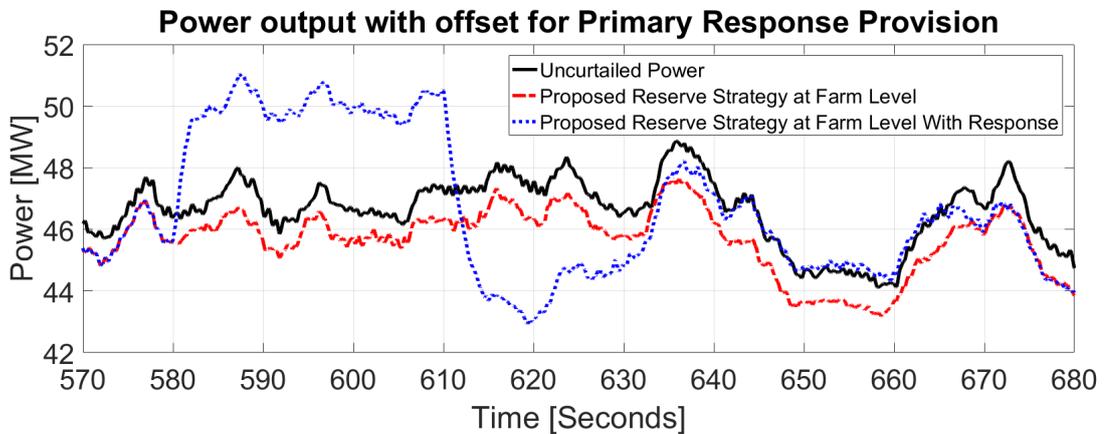


Figure A.6: An example of the Reserve Curtailment strategy in a 9.5 m/s windfield with 10% turbulence intensity in a 16 turbine wind farm.

Figure A.6 shows the farm level power with and without activation of the primary frequency response. The response rate is faster than is required by the grid code [49] as wind turbines can change their power outputs very quickly. This fast response has been demonstrated as a

Appendix A. Using Rotor Inertia as Stored Energy in Below Rated Wind Farms to Provide Primary Frequency Response

possible method of providing synthetic inertia [90]. Figure A.6 shows a large drop in power when the turbines enter into the PAC's recovery process to bring them back to their normal operational strategy. This large dip can be mitigated by staggering when the turbines enter into recovery [98]. Further research on implementing this is planned as part of the development of a power system model in Strathfarm which will be able to use information on a grid's frequency to decide when to begin the recovery process for each turbine to reduce the impact on grid stability.

For the results where the reserve power is held there is a significant difference between when there is a 5% TI and when there is a 10% TI so each will be discussed in turn. There is no statistically significant difference in the totals for energy capture, tower or blade DELs for when the angle of the wind field to the farm is varied. 5% and 10% TI are around the level of turbulence which would be expected in typical operation of an offshore wind farm as [12] shows with data from the Greater Gabbard wind farm.

The results presented here show the average changes of all 24 simulations performed at each wind speed-turbulence combination. While some of the changes are significant the number of simulations needed to prove a statistical improvement in Strathfarm is currently being investigated. As a result of this, definitive conclusions cannot currently be made for which strategy performs the best.

In the results presented here: 10% Curtailment refers to the strategy shown in Equation 2, Turbine Level Curtailment refers to the strategy shown in Equation 3 and Reserve Curtailment refers to the strategy shown in Equation 4.

A.3.1 5% Turbulence Intensity

Approach	Tower DELs				
	8.5 m/s	9 m/s	9.5 m/s	10 m/s	10.5 m/s
Reserve Curtailment	0.87%	2.45%	6.13%	4.64%	0.74%
Turbine Level Curtailment	-2.16%	-1.31%	2.82%	5.26%	4.82%
10% Curtailment	8.12 %	7.96%	8.43%	8.27%	8.17%

Table A.2: The average reduction in tower DELs of each of the three strategies by wind speed when compared to not curtailing the turbines at 5% TI.

Table A.2 shows that there are mixed results for the Reserve Curtailment and Turbine Level Curtailment approaches, peaking at 10 m/s and 9.5 m/s respectively but also with some results

Appendix A. Using Rotor Inertia as Stored Energy in Below Rated Wind Farms to Provide Primary Frequency Response

showing that the approaches are detrimental to tower DEL reductions. In contrast to this the 10% Curtailment approach shows a consistent reduction in tower DELs of approximately 8%.

Blade DELs					
Approach	8.5 m/s	9 m/s	9.5 m/s	10 m/s	10.5 m/s
Reserve Curtailment	2.95%	5.35%	13.48%	16.51%	13.91%
Turbine Level Curtailment	-1.22%	-0.26%	5.53%	10.48%	12.37%
10% Curtailment	11.97%	11.61%	11.76%	10.86%	9.26%

Table A.3: The average reduction in blade DELs of each of the three strategies by wind speed when compared to not curtailing the turbines at 5% TI.

Table A.3 shows the reduction in blade DELs compared with not curtailing the wind farm. The Reserve Curtailment and the Turbine Level Curtailment approaches both show very small improvements over not curtailing at lower wind speed but have very large reductions at 10 and 10.5 m/s. The 10% Curtailment approach shows improvements which vary much less with wind speed.

Energy Capture					
Approach	8.5 m/s	9 m/s	9.5 m/s	10 m/s	10.5 m/s
Reserve Curtailment	1.48%	1.20%	1.37%	2.29%	3.61%
Turbine Level Curtailment	1.46%	1.19%	1.38%	2.29%	3.62%
10% Curtailment	7.26%	7.24%	7.34 %	7.45%	7.59%

Table A.4: The average reduction in energy capture of each of the three strategies by wind speed when compared to not curtailing the turbines.

Table A.4 shows that the 10% Curtailment approach has a reduction in energy capture of between 7.24 and 7.59%, rather than the 10% that would be intuitively predicted. This higher energy capture is due to the wake effects within the farm. As the wind turbines are curtailed the strength of the wakes which propagate through the farm is reduced. This reduction leads to a small change in wind speed but due to the cubic relationship between wind speed and energy in the wind leads to a much larger increase in power generation. Approaches which use the effect are common in the literature but are normally researched in quasi-steady state models rather than the dynamic model used here [90] [95].

A.3.2 10% Turbulence Intensity

The simulations run at 10% turbulence intensity show larger improvements in tower and blade DELs. This is to be expected as when the turbulence intensity increases the DELs will become

Appendix A. Using Rotor Inertia as Stored Energy in Below Rated Wind Farms to Provide Primary Frequency Response

larger as they are caused by variation in bending moments so more variation in operation will make them increase.

Tower DELs					
Approach	8.5 m/s	9 m/s	9.5 m/s	10 m/s	10.5 m/s
Reserve Curtailment	1.35%	7.33%	10.00%	11.20%	13.49%
Turbine Level Curtailment	-3.63%	1.46%	4.61%	8.23%	10.31%
10% Curtailment	5.27%	9.05%	9.71%	10.55%	12.50%

Table A.5: The average reduction in tower DELs of each of the three strategies by wind speed when compared to not curtailing the turbines at 10% TI.

Table A.5 shows that the Reserve Curtailment strategy shows a much greater reduction than not only the Individual Turbine approach but also than the 10% Curtailment approach at wind speeds above 9 m/s.

Blade DELs					
Approach	8.5 m/s	9 m/s	9.5 m/s	10 m/s	10.5 m/s
Reserve Curtailment	7.03%	12.13%	14.83%	15.11%	14.08%
Turbine Level Curtailment	0.50%	4.60%	7.63%	10.50%	10.76%
10% Curtailment	10.93%	11.40%	11.35%	10.58%	9.86%

Table A.6: The average reduction in blade DELs of each of the three strategies by wind speed when compared to not curtailing the turbines at 10% TI.

Table A.6 shows that the improvements for the blade DELs show a similar pattern to the tower DELs with the Reserve Curtailment approach having larger reductions in DELs over both of the other strategies even though it has a higher energy capture.

Energy Capture					
Approach	8.5 m/s	9 m/s	9.5 m/s	10 m/s	10.5 m/s
Reserve Curtailment	1.54%	1.76%	2.17 %	3.27%	4.21%
Turbine Level Curtailment	1.46%	1.72%	2.13%	3.28%	4.27%
10% Curtailment	6.79%	7.33%	7.45%	7.69%	7.94%

Table A.7: The average reduction in energy capture of each of the three strategies by wind speed when compared to not curtailing the turbines at 10% TI.

Table A.7 shows that the Reserve Curtailment and the Individual Turbine approach have higher energy capture than the 10% Curtailment approach with the differences reducing as the wind speed increases.

A.4 Conclusions

By considering the stored rotational energy in each wind turbine's rotor and by curtailing turbines where this cannot be efficiently utilised, a more sophisticated method of holding reserve power than simply reducing turbine power by 10% is shown. The Reserve Curtailment control strategy has a higher energy capture than curtailing each turbine by 10% while not resulting in significantly higher DELs to the towers or the blades in the wind farm.

While the lower tower DELs for the 10% Curtailment strategy is possibly offset by the higher energy capture and lower blade DELs of the Reserve Curtailment approach it is most likely not enough to overcome the fact that the Reserve Curtailment approach not only cannot provide a complete secondary response but also requires a reduction in power output to return each of the wind turbines to their normal operational generator speed. Currently the UK Grid Code requires both primary and secondary response, so if this approach were to be implemented it may be worth considering in the future if, for example a wind farm were to provide a primary and inertial response and could create an arrangement with a slower acting generator to provide the secondary response a joint bid could be submitted in the market.

Another possible approach would be to extract less energy from the wind turbines' rotors, as Figure A.4 shows that 9 and 10 m/s the change in C_p is initially very small so by only extracting a small amount of energy there will be a smaller reduction in C_p so the turbine can be returned to its normal operational strategy over a longer time frame as it will not have to overcome as large of a deficit in efficiency.

Future work in this area would be to simulate the proposed strategies over more more wind speeds, particularly above rated so that the changes in DELs can be validated. Additional further research will be to test these approaches in Strathfarm using a power system model, which is currently under development. Using the grid frequency as an input to the WFC would allow it to stagger the start of the recovery periods across the farm, mitigating the dip in power output.

Appendix B

Power system Model Discretization

This appendix provides additional mathematical working for the discretization done in Chapter 7, showing the discretization of the governor and excitation system for the power system model.

B.1 Discretization of the Governor

The governor is given in continuous time form as:

$$TF_{Governorcontrol} = k_{gain} \frac{1 + T_3s}{1 + T_1 + T_1T_2s^2} \quad (\text{B.1})$$

where $TF_{Governorcontrol}$ is the transfer function of the governor, k_{gain} is the gain of the transfer function, and T_1 , T_2 and T_3 are time constants.

This can be discretized through the following steps:

$$G(s) = \frac{T_3s + 1}{T_1T_2s^2 + T_1s + 1} \quad (\text{B.2})$$

This equation can be written in the form:

$$G(s) = \frac{T_3s^{-1} + s^{-2}}{T_1T_2 + T_1s^{-1} + s^{-2}} \quad (\text{B.3})$$

using the inverse of the bilinear transform in the form:

$$s^{-1} = \frac{1}{K_T} \frac{1 + z^{-1}}{1 - z^{-1}} \quad (\text{B.4})$$

Appendix B. Power system Model Discretization

where $K_T = \frac{2}{T}$ gives:

$$G(s) = \frac{T_3 \frac{1}{K_T} \frac{1+z^{-1}}{1-z^{-1}} + \left(\frac{1}{K_T} \frac{1+z^{-1}}{1-z^{-1}}\right)^2}{T_1 T_2 + T_1 \frac{1}{K_T} \frac{1+z^{-1}}{1-z^{-1}} + \left(\frac{1}{K_T} \frac{1+z^{-1}}{1-z^{-1}}\right)^2} \quad (\text{B.5})$$

which is equal to:

$$G(s) = \frac{T_3 \frac{1}{K_T} (1+z^{-1})(1-z^{-1}) + \left(\frac{1}{K_T}\right)^2 (1+z^{-1})^2}{T_1 T_2 (1+z^{-1})^2 + T_1 \frac{1}{K_T} (1+z^{-1})(1-z^{-1}) + \left(\frac{1}{K_T}\right)^2 (1+z^{-1})^2} \quad (\text{B.6})$$

which can be expanded to:

$$G(s) = \frac{\frac{T_3}{K_T} (1-z^{-2}) + \left(\frac{1}{K_T}\right)^2 (1+2z^{-1}+z^{-2})}{T_1 T_2 (1+2z^{-1}+z^{-2}) + \frac{T_1}{K_T} (1-z^{-2}) + \left(\frac{1}{K_T}\right)^2 (1+2z^{-1}+z^{-2})} \quad (\text{B.7})$$

rearranging for coefficient terms gives:

$$G(s) = \frac{\left(\frac{1}{K_T^2} + \frac{T_3}{K_T}\right) + \frac{2}{K_T^2} z^{-1} + \left(\frac{1}{K_T^2} - \frac{T_3}{K_T}\right) z^{-2}}{\left(T_1 T_2 + \frac{T_1}{K_T} + \frac{1}{K_T^2}\right) + (-2T_1 T_2 + \frac{2}{K_T^2}) z^{-1} + \left(T_1 T_2 - \frac{T_1}{K_T} + \frac{1}{K_T^2}\right) z^{-2}} \quad (\text{B.8})$$

Next the difference equation can be found by considering $G(z) = \frac{Y(z)}{X(z)}$:

$$\begin{aligned} & \left(\left(T_1 T_2 + \frac{T_1}{K_T} + \frac{1}{K_T^2}\right) + (-2T_1 T_2 + \frac{2}{K_T^2}) z^{-1} + \left(T_1 T_2 - \frac{T_1}{K_T} + \frac{1}{K_T^2}\right) z^{-2}\right) Y(z) \\ &= \left(\left(\frac{1}{K_T^2} + \frac{T_3}{K_T}\right) + \left(\frac{2}{K_T^2}\right) z^{-1} + \left(\frac{1}{K_T^2} - \frac{T_3}{K_T}\right) z^{-2}\right) X(z) \end{aligned} \quad (\text{B.9})$$

Rearranging and multiplying through by K_T^2 to improve readability gives:

$$\begin{aligned} & \left(\left(T_1 T_2 K_T^2 + T_1 K_T + 1\right) + (-2T_1 T_2 K_T^2 + 2) z^{-1} + \left(T_1 T_2 K_T^2 - T_1 K_T + 1\right) z^{-2}\right) Y(z) \\ &= \left(1 + T_3 K_T\right) + (2) z^{-1} + (1 - T_3 K_T) z^{-2} X(z) \end{aligned} \quad (\text{B.10})$$

hence the difference equation is:

$$\begin{aligned} (T_1 T_2 K_T^2 + T_1 K_T + 1) y[n] &= (1 + T_3 K_T) x[n] + 2x[n-1] + (1 - T_3 K_T) x[n-2] \\ &\quad - (-2T_1 T_2 K_T^2 + 2) y[n-1] - (T_1 T_2 K_T^2 - T_1 K_T + 1) y[n-2] \end{aligned} \quad (\text{B.11})$$

Appendix B. Power system Model Discretization

Which can be rearranged to give the final form of the difference equation:

$$y[n] = \frac{1}{(T_1 T_2 K_T^2 + T_1 K_T + 1)} ((1 + T_3 K_T)x[n] + 2x[n-1] + (1 - T_3 K_T)x[n-2] - (-2T_1 T_2 K_T^2 + 2)y[n-1] - (T_1 T_2 K_T^2 - T_1 K_T + 1)y[n-2]) \quad (\text{B.12})$$

B.1.1 Actuation

The governor also includes an actuator of the form:

$$TF_{Governorcontrol2} = \frac{1 + T_4 s}{s(1 + T_5 s)(1 + T_6 s)} \quad (\text{B.13})$$

where and T_4 , T_5 and T_6 are time constants.

This equation can be split into two by considering each actuator independently.

The Governor's 1st Actuator

The transfer function of the first governor actuator is given as:

$$TF_{A1} = \frac{T_4 s + 1}{T_5 s + 1} \quad (\text{B.14})$$

which can be written as:

$$TF_{A1} = \frac{T_4 + s^{-1}}{T_5 + s^{-1}} \quad (\text{B.15})$$

As above the bilinear transform can be used:

$$TF_{A1} = \frac{T_4 + \frac{1}{K_T} \frac{1+z^{-1}}{1-z^{-1}}}{T_5 + \frac{1}{K_T} \frac{1+z^{-1}}{1-z^{-1}}} \quad (\text{B.16})$$

$$TF_{A1} = \frac{K_T T_4 (1 - z^{-1}) + (1 + z^{-1})}{K_T T_5 (1 - z^{-1}) + (1 + z^{-1})} \quad (\text{B.17})$$

To find the difference equation for this transfer function TF_{A1} can be considered as a transfer function of the form $\frac{Y(z)}{X(z)}$:

$$\frac{Y(z)}{X(z)} = \frac{K_T T_4 (1 - z^{-1}) + (1 + z^{-1})}{K_T T_5 (1 - z^{-1}) + (1 + z^{-1})} \quad (\text{B.18})$$

Appendix B. Power system Model Discretization

$$(K_T T_5(1 - z^{-1}) + (1 + z^{-1}))Y(z) = ((K_T T_4(1 - z^{-1}) + (1 + z^{-1}))X(z) \quad (\text{B.19})$$

Hence the difference equation is:

$$y[n] = \frac{(K_T T_4 + 1)x[n] + (1 - K_T T_4)x[n-1] - (1 - T_5 K_T)y[n-1]}{1 + T_5 K_T} \quad (\text{B.20})$$

B.1.2 The Governor's 2nd Actuator

The continuous time form of the transfer function for the 2nd actuator is given as:

$$TF_{A2} = \frac{1}{sT_6 + 1} \quad (\text{B.21})$$

This can be written as:

$$TF_{A2} = \frac{s^{-1}}{T_6 + s^{-1}} \quad (\text{B.22})$$

As above the bilinear transform can be used:

$$TF_{A2} = \frac{\frac{1}{K_T} \frac{1+z^{-1}}{1-z^{-1}}}{T_6 + \frac{1}{K_T} \frac{1+z^{-1}}{1-z^{-1}}} \quad (\text{B.23})$$

and can be subsequently simplified to:

$$TF_{A2} = \frac{1 + z^{-1}}{K_T(1 - z^{-1})T_6 + 1 + z^{-1}} \quad (\text{B.24})$$

which can be written as:

$$\frac{Y(z)}{X(z)} = \frac{1 + z^{-1}}{K_T(1 - z^{-1})T_6 + 1 + z^{-1}} \quad (\text{B.25})$$

rearranged this gives

$$Y(z)(K_T(1 - z^{-1})T_6 + 1 + z^{-1}) = X(z)(1 + z^{-1}) \quad (\text{B.26})$$

which can be written as:

$$Y(z)(z^{-1}(-K_T T_6 + 1) + K_T T_6 + 1) = X(z)(1 + z^{-1}) \quad (\text{B.27})$$

Appendix B. Power system Model Discretization

which can be written as a difference equation in the form:

$$y[n-1](-K_T T_6 + 1) + y[n](K_T T_6 + 1) = x[n] + x[n-1] \quad (\text{B.28})$$

Hence the difference equation is:

$$y[n] = \frac{x[n] + x[n-1] + (K_T T_6 - 1)y[n-1]}{K_T T_6 + 1} \quad (\text{B.29})$$

B.2 Discretization of the Generator Excitation System

B.2.1 Low Pass Filter

In continuous time the low pass filter for the generator excitation system is given by:

$$TF_{lpf} = \frac{1}{t_r s + 1} \quad (\text{B.30})$$

where TF_{lpf} is a transfer function and t_r is a time constant. In a discrete time implementation the transfer function can be written as:

$$y[n] = \frac{x[n] + x[n-1] + (K_T t_r - 1)y[n-1]}{K_T t_r + 1} \quad (\text{B.31})$$

This can be done by comparison to the transfer function of the 2nd generator actuator as the two transfer functions are of the same form.

B.2.2 Main Regulator

In continuous time the transfer function for the main regulator of the exciter can be written as:

$$TF_{mr} = \frac{k_a}{T_a s + 1} \quad (\text{B.32})$$

where TF_{mr} is a transfer function, k_a is the gain of the transfer function, and T_a is a time constant. ignoring k_a as it is just a gain then transfer function can be written in discrete time as:

$$y[n] = \frac{x[n] + x[n-1] + (K_T T_a - 1)y[n-1]}{K_T T_a + 1} \quad (\text{B.33})$$

Appendix B. Power system Model Discretization

This can be done by comparison to the transfer function of the 2nd generator actuator as the two transfer functions are of the same form.

B.2.3 Lead Lag Compensator

In continuous time the lead lag compensator can be written as:

$$TF_{llc} = \frac{T_c s + 1}{T_b s + 1} \quad (\text{B.34})$$

Which can be discretized to:

$$y[n] = \frac{(K_T T_c + 1)x[n] + (1 - K_T T_c)x[n - 1] - (1 - T_b K_T)y[n - 1]}{1 + T_b K_T} \quad (\text{B.35})$$

This can be done by comparison to the transfer function of the 1st generator actuator as the two transfer functions are of the same form.

B.2.4 Damping

The damping is given in continuous time as:

$$TF_{damp} = \frac{k_{fd} s}{t_{fd} s + 1} \quad (\text{B.36})$$

where TF_{damp} is a transfer function, k_{fd} is the gain and t_{fd} is a time constant. The transfer function can be written in the form:

$$TF_{damp} = \frac{k_{fd}}{t_{fd} + s^{-1}} \quad (\text{B.37})$$

which can be written in discrete time using a bilinear transform as:

$$TF_{damp} = \frac{k_{fd}}{t_{fd} + \frac{1}{K_T} \frac{1+z^{-1}}{1-z^{-1}}} \quad (\text{B.38})$$

which can be simplified:

$$TF_{damp} = \frac{K_T(1 - z^{-1})k_{fd}}{K_T(1 - z^{-1})t_{fd} + 1 + z^{-1}} \quad (\text{B.39})$$

Appendix B. Power system Model Discretization

$$\frac{Y(z)}{X(z)} = \frac{K_T(1 - z^{-1})k_{fd}}{K_T t_{fd} + 1 + z^{-1}(1 - K_T t_{fd})} \quad (\text{B.40})$$

which can be written as:

$$Y(z)(K_T t_{fd} + 1 + z^{-1}(1 - K_T t_{fd})) = X(z)(K_T(1 - z^{-1})k_{fd}) \quad (\text{B.41})$$

and rearranged to give:

$$y[n](K_T t_{fd} + 1) + y[n - 1](1 - K_T t_{fd}) = k_{fd} K_T x[n] - k_{fd} K_T x[n - 1] \quad (\text{B.42})$$

Hence the difference equation is:

$$y[n] = \frac{k_{fd} K_T x[n] - k_{fd} K_T x[n - 1] + (t_{fd} K_T - 1)y[n - 1]}{t_{fd} K_T + 1} \quad (\text{B.43})$$

B.2.5 Exciter

$$TF_{ex} = \frac{1}{T_{ex}s + k_e} \quad (\text{B.44})$$

where TF_{ex} is a transfer function, T_{ex} is a time constant and k_e is a constant. The transfer function be written in the form:

$$TF_{ex} = \frac{\frac{1}{k_e}}{\frac{T_{ex}}{k_e}s + 1} \quad (\text{B.45})$$

take $\frac{1}{k_e}$ as a gain and ignore then the transfer function is of form of same as governor's second actuator hence:

$$y[n] = \frac{x[n] + x[n - 1] + (K_T \frac{T_{ex}}{k_e} - 1)y[n - 1]}{K_T \frac{T_{ex}}{k_e} + 1} \quad (\text{B.46})$$

with a gain of the form:

$$y[n] = \frac{1}{k_e} x[n] \quad (\text{B.47})$$

Appendix C

Alternative model for the Gaussian wake

This appendix details an alternative model for including a Gaussian wake in Strathfarm. As discussed in chapter 8 this approach is less accurate than the implemented solution due to it averaging the wake within the rotor swept area. This cannot be neglected within the existing turbine model within Strathfarm which includes rotor shear forces and the imbalance that is seen when a turbine is in a partial wake.

The equations presented in [15] only provide an equation for the deficit at a given point. to find the average deficit across the swept area requires further calculation. The method used here is based on a similar problem in optics [62] which looks at finding the power of a Gaussian laser through an aperture. While this seems unrelated to wake modelling from a simply mathematical perspective they are very similar problems. Beginning with the wake deficit equation for the Gaussian model from [15] :

$$\frac{\Delta u}{u_\infty} = \left(1 - \sqrt{1 - \frac{C_T}{8(k^* \frac{x}{d_0} + 0.2\sqrt{\beta})^2}}\right) \exp\left(-\frac{1}{2(k^* \frac{x}{d_0} + 0.2\sqrt{\beta})^2} \left\{ \left(\frac{z - z_h}{d_0}\right)^2 + \left(\frac{y}{d_0}\right)^2 \right\}\right) \quad (\text{C.1})$$

for ease of calculation the following are used:

$$A(x) = \left(1 - \sqrt{1 - \frac{C_T}{8(k^* \frac{x}{d_0} + 0.2\sqrt{\beta})^2}}\right) \quad (\text{C.2})$$

Appendix C. Alternative model for the Gaussian wake

$$B(x) = \frac{1}{2(k^* \frac{x}{d_0} + 0.2\sqrt{\beta})^2} \quad (\text{C.3})$$

$$\frac{\Delta u}{u_\infty} = A(x)e^{-B(x)\left\{\left(\frac{z-z_h}{d_0}\right)^2 + \left(\frac{y}{d_0}\right)^2\right\}} \quad (\text{C.4})$$

z is redefined as: $z_{New} = z - z_h$ so that the frame of reference has its origin at the centre of the rotor. y is redefined as $y = y + d$ where d is an offset representing the meandering of a wake in the lateral dimension.

$$\frac{\Delta u}{u_\infty} = A(x)e^{-B(x)\left\{\left(\frac{z_{new}}{d_0}\right)^2 + \left(\frac{y+d}{d_0}\right)^2\right\}} \quad (\text{C.5})$$

$$\frac{\Delta u}{u_\infty} = A(x)e^{-B(x)\frac{1}{d_0^2}\{z_{new}^2 + y^2 + 2yd + d^2\}} \quad (\text{C.6})$$

converting this to cylindrical coordinates of the mapping $y, z, x \rightarrow r, \theta, z$ gives:

$$\frac{\Delta u}{u_\infty} = A(x)e^{-B(x)\frac{1}{d_0^2}\{r^2 + 2rd\cos(\theta) + d^2\}} \quad (\text{C.7})$$

To average the deficit over the rotor's swept area first the double integral must be found:

$$Q = \int_0^R \int_0^{2\pi} = A(x)e^{-B(x)\frac{1}{d_0^2}\{r^2 - 2rd\cos(\theta) + d^2\}} d\theta r dr \quad (\text{C.8})$$

which can be written as:

$$Q = A(x)e^{-B(x)\frac{d^2}{d_0^2}} \int_0^R r e^{-B(x)\frac{r^2}{d_0^2}} \int_0^{2\pi} e^{\frac{B(x)2rd\cos(\theta)}{d_0^2}} d\theta dr \quad (\text{C.9})$$

C.1 Bessel Function Derivation

The θ integral can be shown to be equal to a Modified Bessel Function of the first kind and order zero through the following steps:

$$D = \int_0^{2\pi} e^{\frac{2B(x)rd\cos(\theta)}{d_0^2}} \quad (\text{C.10})$$

Take the Maclaurin series of D :

Appendix C. Alternative model for the Gaussian wake

$$D = \int_0^{2\pi} \sum_{k=0}^{\infty} \frac{\left(\frac{2B(x)rd\cos(\theta)}{d_0^2}\right)^k}{k!} \quad (\text{C.11})$$

We now need to consider the odd and even terms in the sum

$$D = \int_0^{2\pi} \left(\sum_{k=0}^{\infty} \frac{\left(\frac{2B(x)rd\cos(\theta)}{d_0^2}\right)^{2k}}{(2k)!} + \sum_{k=0}^{\infty} \frac{\left(\frac{2B(x)rd\cos(\theta)}{d_0^2}\right)^{2k+1}}{(2k+1)!} \right) \quad (\text{C.12})$$

C.1.1 Even Terms

Only considering the cos component of the even sum through the use of a repeated iteration of integration by parts gives the equation:

$$\int_0^{2\pi} \cos^{2n}(\theta) d\theta = \left[\frac{(2n!)x}{4^n(n!)^2} + \sum_{k=0}^{\infty} \frac{(2n)!((k-1)!)^2 \sin(\theta) \cos^{2k-1}(\theta)}{4^{n-k+1}(n!)^2(2k-1)!} \right]_0^{2\pi} \quad (\text{C.13})$$

due to the sine terms within the summation all terms of the summation are equal to zero.

C.1.2 Odd Terms

The Odd terms are given by the equation:

$$\int_0^{2\pi} \cos^{2n+1}(\theta) d\theta = \int_0^{2\pi} \cos^{2n+1}(\theta) d\theta \quad (\text{C.14})$$

which can be seen to equal zero for all terms.

C.2 Continuation

Therefore the equation can be written as:

$$Q = A(x) e^{-B(x)\frac{d^2}{d_0^2}} 2\pi \int_0^R r e^{-B(x)\frac{r^2}{d_0^2}} \sum_{k=0}^{\infty} 2\pi \frac{\left(\frac{1}{4}\left(\frac{B(x)2rd}{d_0^2}\right)^2\right)^k}{(k!)^2} dr \quad (\text{C.15})$$

which can be rearranged and simplified as:

$$Q = A(x) e^{-B(x)\frac{d^2}{d_0^2}} 2\pi \sum_{k=0}^{\infty} \frac{\left(\frac{B(x)d}{d_0^2}\right)^{2k}}{(k!)^2} \int_0^R r^{2k+1} e^{-B(x)\frac{r^2}{d_0^2}} dr \quad (\text{C.16})$$

C.3 Gamma Substitution

$$\int_0^R r^{2k+1} e^{-B(x) \frac{r^2}{d_0^2}} dr \quad (\text{C.17})$$

$$\gamma(s, z) = \int_0^z t^{s-1} e^{-t} dt \quad (\text{C.18})$$

$$t = B(x) \frac{r^2}{d_0^2} \quad (\text{C.19})$$

$$\frac{\partial t}{\partial r} = \frac{2B(x)r}{d_0^2} \quad (\text{C.20})$$

$$\partial r = \frac{d_0^2 \partial t}{2B(x)r} \quad (\text{C.21})$$

our integration limits are therefore:

$$t_{upper} = \frac{B(x)R^2}{d_0^2} \quad (\text{C.22})$$

$$t_{lower} = 0 \quad (\text{C.23})$$

and

$$r = \pm \sqrt{\frac{d_0^2 t}{B(x)}} \quad (\text{C.24})$$

$$\int_0^R r^{2k+1} e^{-B(x) \frac{r^2}{d_0^2}} dr = \int_0^{\frac{B(x)R^2}{d_0^2}} r^{2k+1} e^{-t} \frac{d_0^2 dt}{2B(x)r} \quad (\text{C.25})$$

$$\int_0^R r^{2k+1} e^{-B(x) \frac{r^2}{d_0^2}} dr = \int_0^{\frac{B(x)R^2}{d_0^2}} r^{2k} e^{-t} \frac{d_0^2 dt}{2B(x)} \quad (\text{C.26})$$

$$\int_0^R r^{2k+1} e^{-B(x) \frac{r^2}{d_0^2}} dr = \frac{d_0^2}{2B(x)} \int_0^{\frac{B(x)R^2}{d_0^2}} r^{2k} e^{-t} dt \quad (\text{C.27})$$

$$\int_0^R r^{2k+1} e^{-B(x) \frac{r^2}{d_0^2}} dr = \frac{d_0^2}{2B(x)} \gamma(k+1, \frac{B(x)R^2}{d_0^2}) \quad (\text{C.28})$$

Appendix C. Alternative model for the Gaussian wake

as a lower incomplete gamma function can be written in the form:

$$\gamma(n, x) = (n - 1)! (1 - e^{-x}) \sum_{k=0}^{n-1} \frac{x^k}{k!} \quad (\text{C.29})$$

$$\gamma(k + 1, \frac{B(x)R^2}{d_0^2}) = (k)! (1 - e^{-\frac{B(x)R^2}{d_0^2}}) \sum_{l=0}^k \frac{(\frac{B(x)R^2}{d_0^2})^l}{l!} \quad (\text{C.30})$$

C.4 Final Equations

Therefore the deficit of the wake can be calculated by:

$$Q = A(x) e^{-B(x) \frac{d^2}{d_0^2}} 2\pi \sum_{k=0}^{\infty} \frac{(\frac{B(x)d}{d_0^2})^{2k}}{(k!)^2} (k)! (1 - e^{-\frac{B(x)R^2}{d_0^2}}) \sum_{l=0}^k \frac{(\frac{B(x)R^2}{d_0^2})^l}{l!} \quad (\text{C.31})$$

While this looks like it would be computationally expensive due to the infinite series sum in a engineering model only the first term (or few terms for high accuracy) are required to find the wake deficit.

C.5 Example Results

The plots in this subsection show the difference between using the approach detailed above compared with only using the point value of the wake deficit.

These plots assume that the turbulence intensity is 10% and turbines with a wake diameter of 178.3320 meters. The averaged Gaussian wake from the proof given above is labelled as averaged deficit and the point deficit from equation (C.1) is given as ‘point deficit’. Figure (C.1), (C.2), (C.3) and (C.4) show a comparison of the averaged Gaussian wake deficit of with the point deficit with different C_T and downwind distances.

Appendix C. Alternative model for the Gaussian wake

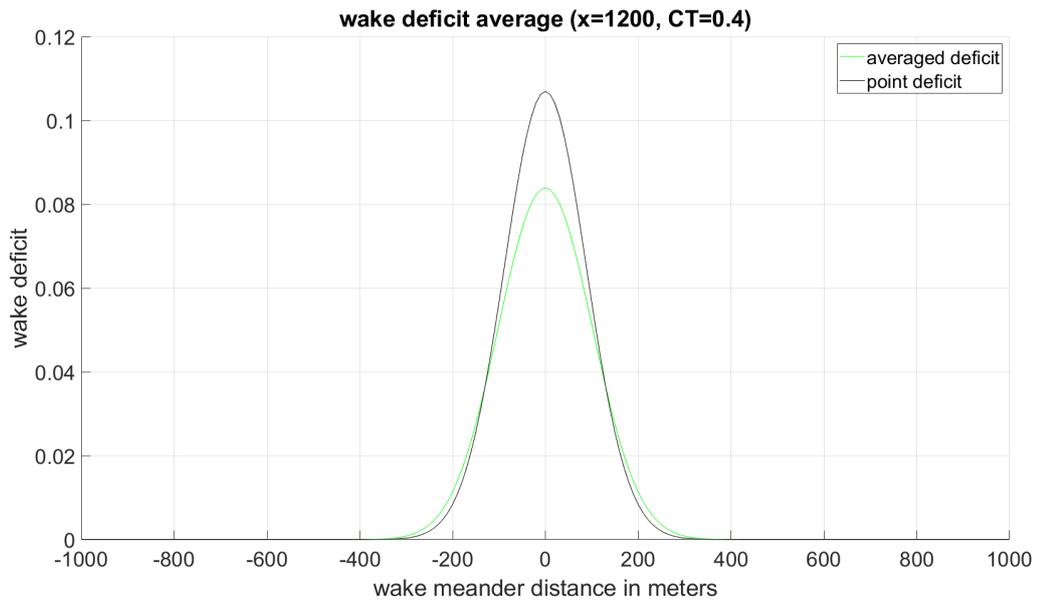


Figure C.1: Gaussian wake 1200 meters down wind with an upwind turbine C_T of 0.4

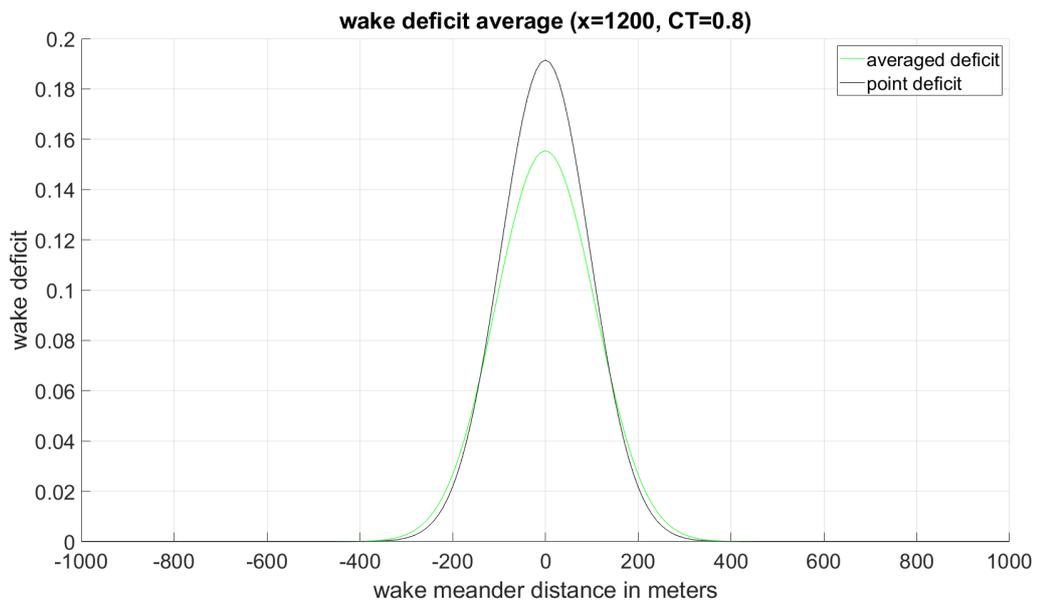


Figure C.2: Gaussian wake 1200 meters down wind with an upwind turbine C_T of 0.8

Appendix C. Alternative model for the Gaussian wake

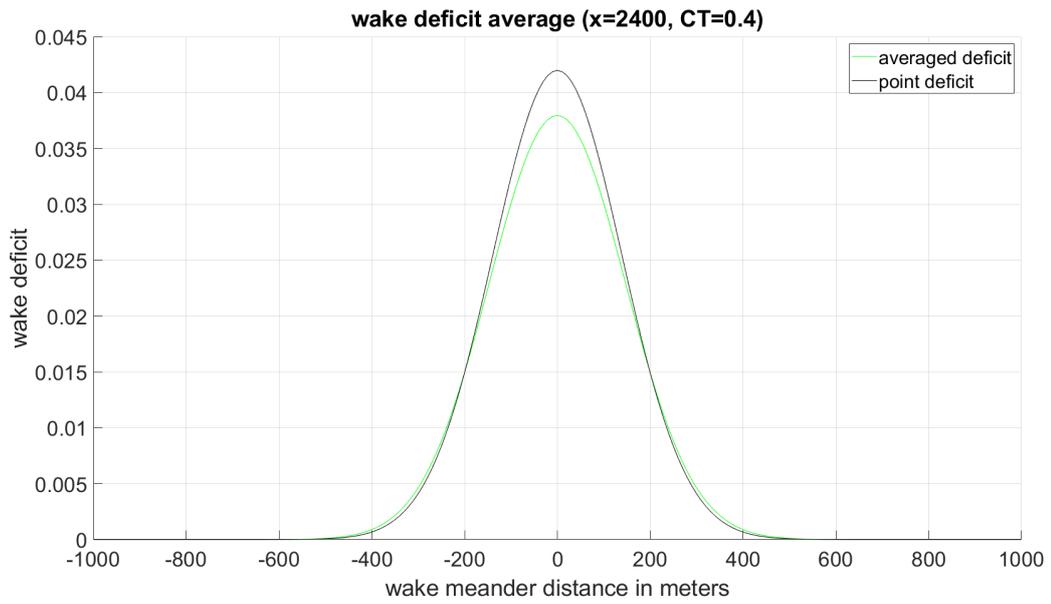


Figure C.3: Gaussian wake 2400 meters down wind with an upwind turbine C_T of 0.4

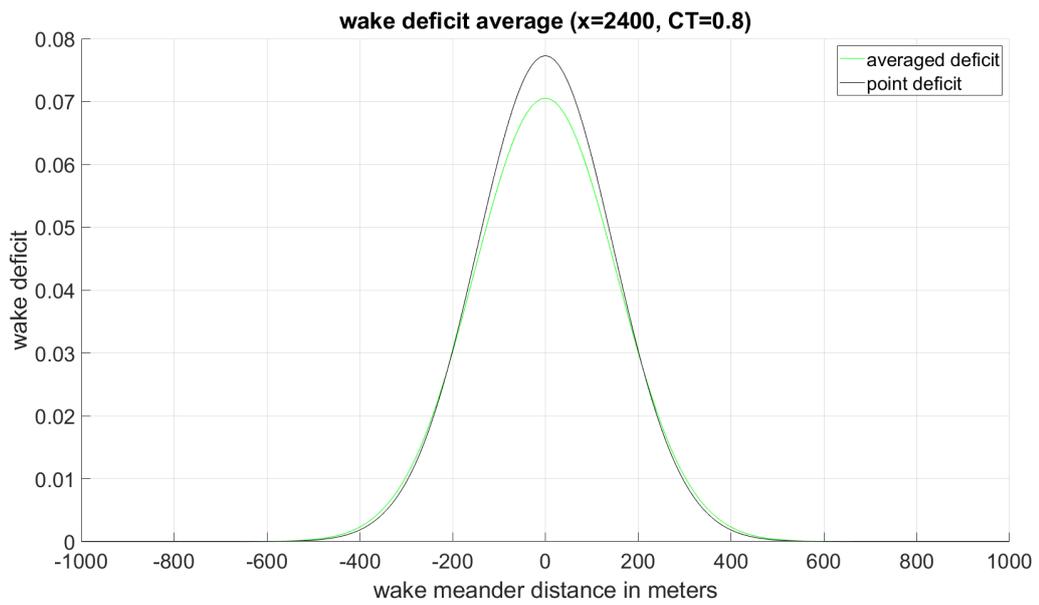


Figure C.4: Gaussian wake 2400 meters down wind with an upwind turbine C_T of 0.8

Appendix D

Variables For Power System Model

S-function External Tuning

This appendix lists the external tuning variables for the four S-functions that make up the power system model. Each of the four codes reads an external text file to import the variables in the code. This is done so that the model can be tuned or tested without the need to recompile the DLLs each time.

project name:multi_solver	
External variable file:multi_solver_parameters.txt	
Variable name	description
Rc	resistive component of the phase reactor of converter
Lc	inductive component of the phase reactor of converter
Rn	AC cable resistance
Ln	AC cable Inductance
Cf	capacitance of the filter of converter
Lf	Additional model connection point inductance per line
V0	converter voltage
E0	grid voltage
Lt	inductance of the transmission line for converter
Rt	resistance of the transmission line for converter
iter_max	The maximum number of iterations that the solver will perform per time step
tol	a tolerance limit for the solver to decide if the solution is close enough each time step.

Table D.1: The external tuning parameters of the solver S-Function

Appendix D. Variables For Power System Model S-function External Tuning

Table(D.1) lists and describes the external parameters for the solver S-Function in Strath-farm.

Please note that the external variables are normalized as discussed in [42].

project name:PS_mech_model_dll	
external variables:PS_mech_parameters.txt	
Variable name	description
Ts	time step for model
H	inertia constant of the model
w_ref	reference grid frequency (PU)
T1	time constant for governor controller
T2	time constant for governor controller
T3	time constant for governor controller
T4	time constant for governor 1st actuator
T5	time constant for governor 1st actuator
T6	time constant for governor 2nd actuator
K_gain	gain for governor controller

Table D.2: The external tuning of the Mechanical model S-function.

Table(D.2) shows the external tuning of the Mechanical model S-function.

Appendix D. Variables For Power System Model S-function External Tuning

project name:PS_elec_model_dll	
external variable tuning:PS_elec_parameters.txt	
Variable name	description
Vref	reference voltage
Vf0	main regulator
Ka	main regulator gain
Tr	time constant of the low pass filter if set to zero low pass filter removed.
Kt	2 divided by the model sample time should be 40000 on current setup.
Ta	main regulator time constant
Tfd	damping
Kfd	damping
Rs	Stator resistance
Rf	field winding resistance
Rkd	damper resistance
Rkq	damper resistance
Lq	mutual inductance between stator and rotor windings
Lakq	mutual inductance between stator and rotor windings
Lakd	mutual inductance between stator and rotor windings
Lafd	mutual inductance between stator and rotor windings
Lfkd	mutual inductance between stator and rotor windings
Ld	mutual inductance between stator and rotor windings
Lffd	self inductance of rotor circuit
Lkkd	self inductance of rotor circuit
Lkkq	self inductance of rotor circuit
constant_freq_switch	switch for speed up mode when the grid frequency is very close to 1.

Table D.3: The external tuning parameters for the electrical S-Function for the power system model.

Table(D.3) lists and describes the external tuning parameters for the electrical S-Function for the power system model in Strathfarm. These values have been normalised as described in [66] to allow for the model to be scaled as required.

Appendix D. Variables For Power System Model S-function External Tuning

project name:PS_model_dll	
external variables:PS_parameters.txt	
Variable name	description
Timestep	model timestep
L_node	node capacitance in power model
Vbase	base grid voltage (default=25000)
Power	The power demand of the grid
R_node	stray resistance of the node

Table D.4: The external tuning variables for the power system linking section of the power system model in Strathfarm

Table (D.4) lists and describes the external variables used in the S-Function for the linking code in the power system model in Strathfarm.

Appendix E

Wind Farm Distribution Controller

C++ Code

This Appendix contains the C++ code used in the Wind Farm Distribution Controller S-function in Strathfarm. It should be noted that this code will not work without the linking function which cannot be included in this thesis due to intellectual property restrictions.

```
1 // Common libraries
2 #include <windows.h>
3 #include <stdio.h>
4 #include <string.h>
5 #include <math.h>
6 #include <conio.h>
7 #include <iostream>
8 using namespace std;
9 #include <stdlib.h>
10 #include <string>
11 #include <iomanip>
12 #include <fstream>
13 #include <cmath>
14 #include <vector>
15 #include <algorithm> // std::sort
16
17 #define INUM_PAC_FLAGS 20
18 #define INUM_PAC_VALUES 5
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
19
20
21 //***** Required at the beginning (from Bladed Manual)
22 extern "C" //avoid mangled names
23 { void __declspec(dllexport) __cdecl windFarmControl(float *parPowerToFarm,
24             int *pabFlagsToFarm, float *parValuesToFarm,
25             int *pbCurtailmentFlagRequest, float *prCurtailmentValue,
26             int *prInertiaSwitch, float *prSimTime, int iNUM_TURBINES, float *
27             parReqChangeInPower,
28             int *pabPacOnFlag, int *pabPriorityFlag, int *pabFastSlowFlag, int *
29             pabCurtailmentFlag,
30             float *parCurtailmentPower, int *pabCurtailmentProfileFlag, float *
31             parNewRatedGenSpd,
32             int *pabAdjustRatedSpeedFlag);
33 }
34 int getFlag(char *achRequestedFlag, int abFlags[], int iNUM_TURBINES,
35             int *abOutputFlag);
36 int getValue(char *achRequestedValue, float arValues[], int iNUM_TURBINES,
37             float *arOutputValue);
38 int getValue2(char *achRequestedValue, int arValues[], int iNUM_TURBINES,
39             float *arOutputValue);
40 int getTurbineFlags(int iReqTurbine, int abFlags[], int iNUM_TURBINES,
41                     int *abOutputFlags);
42
43 //***** Main DLL routine *****//
44 void __declspec(dllexport) __cdecl windFarmControl(float *parPowerToFarm,
45             int *pabFlagsToFarm, float *parValuesToFarm,
46             int *pbCurtailmentFlagRequest, float *prCurtailmentValue,
47             int *prInertiaSwitch, float *prSimTime, int iNUM_TURBINES, float *
48             parReqChangeInPower,
49             int *pabPacOnFlag, int *pabPriorityFlag, int *pabFastSlowFlag, int *
50             pabCurtailmentFlag,
51             float *parCurtailmentPower, int *pabCurtailmentProfileFlag, float *
52             parNewRatedGenSpd,
53             int *pabAdjustRatedSpeedFlag)
54 {
55 }
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
48  /* The inputs are:
49  parPowerToFarm - The power output of the wind farm - each element of the
      array is one turbine's
50  power output [W]
51  pabFlagsToFarm - The 20 flags of each turbine, elements 0-19 are turbine
      1, 20-39 are turbine 2
52  etc. [Booleans]
53  parValuesToFarm - The 4 PAC values of each turbine [Floats]
54  pbCurtailmentFlagRequest - A flag that denotes that curtailment is
      requested [Boolean]
55  prCurtailmentValue - The value of the curtailment [W]
56  prRocof - "Rate of change of frequency". The rate of change of the grid
      frequency [Hz/s]
57  prFrequency - The grid frequency [Hz]
58  prSimTime - The simulation time [s]
59  piNUM_TURBINES - The number of turbines in the farm [Turbines]
60  */
61  /* The outputs are:
62  parReqChangeInPower - The requested change in power output for each
      turbine. Each element is
63  a turbine's requested change in power [W]
64  pabPacOnFlag - The flags to turn the PACs on. Each element represents a
      turbine's flag
65  [Boolean]
66  pabPriorityFlag - The flags to remove rate of change of power limits.
      Each element represents
67  a turbine's flag [Boolean]
68  pabFastSlowFlag - The flag to select the recovery speed (1 = fast, 0 =
      slow) Each element
69  represents a turbine's flag [Boolean]
70  pabCurtailmentFlag - The flag to use Curtailment mode of PAC. Each element
      represents a turbine's flag [Boolean]
71  parCurtailmentPower - The requested curtailment power value. Each element
      represents a turbine's flag [W]
72  pabCurtailmentProfileFlag - The flag for curtailment profile of PAC. Each
      element represents a turbine's flag [Boolean]
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
73  parNewRatedGenSpd - The requested new rated generator speed. Each element
    represents a turbine's flag [rad/s]
74  pabAdjustRatedSpeedFlag - The flag to use new rated generator speed. Each
    element represents a turbine's flag [Boolean]
75  */
76
77  int iCntTurbines; // Counting turbines [turbines]
78  int iTurbineFlags[iNUM_PAC_FLAGS];
79  int rTurbineValues[iNUM_PAC_VALUES];
80
81  // Examples of using the "getValue" function used for the 4 value inputs:
82  // Get the estimated wind speeds from the PACs:
83  float arEWS[' '];
84  int itest = getValue("rUpStreamWindSpd", &(parValuesToFarm[0]),
    iNUM_TURBINES, arEWS);
85  // Get the estimate of power without PAC from the PAC:
86  float arPowNoPac[' '];
87  int itest1 = getValue("rPowEstNoPac", &(parValuesToFarm[0]), iNUM_TURBINES
    , arPowNoPac);
88  float rAvailPow = 0;
89  int rate_flag[' '];
90  int rflag = getFlag("bSubFlagPowerRate", &(pabFlagsToFarm[0]),
    iNUM_TURBINES, rate_flag);
91  // Examples of using the "getFlag" function used for the 20 flag inputs:
92  // Get the green flags:
93  int abGreenFlag[' '];
94  int gFlag = getFlag("bGreenFlag", &(pabFlagsToFarm[0]), iNUM_TURBINES,
    abGreenFlag);
95  // Get the amber flags:
96  int abAmberFlag[' '];
97  int aFlag = getFlag("bAmberFlag", &(pabFlagsToFarm[0]), iNUM_TURBINES,
    abAmberFlag);
98  //Get the red flags:
99  int abRedFlag[' '];
100 int rFlag = getFlag("bRedFlag", &(pabFlagsToFarm[0]), iNUM_TURBINES,
    abRedFlag);
101 //get recovery flags
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
102 int abRecoveryFlag[' '];
103 int recFlag = getFlag("bRecoveryFlag", &(pabFlagsToFarm[0]), iNUM_TURBINES
, abRecoveryFlag);
104 int abSubFlagRecoveryComplete[' '];
105 int rec1Flag = getFlag("bSubFlagRecoveryComplete", &(pabFlagsToFarm[0]),
iNUM_TURBINES, abSubFlagRecoveryComplete);
106 int abRejectionFlag[' '];
107 int rejFlag = getFlag("bRejectionFlag", &(pabFlagsToFarm[0]),
iNUM_TURBINES, abRejectionFlag);
108
109 float DiffPow[' '];
110 int itest2 = getValue("rDiffPower", &(parValuesToFarm[0]), iNUM_TURBINES,
DiffPow);
111
112 int abDivergentFlag[' '];
113 int divFlag = getFlag("bDivergentFlag", &(pabFlagsToFarm[0]),
iNUM_TURBINES, abDivergentFlag);
114
115
116 //-----External Parameter section-----//
117
118
119 static float Ki; //Imports WFC integral constant for PI controller
120 static float Kp; //Imports WFC proprtional constant for PI controller
121 static float start_time; // Imports start time for curtailment. Must be
greater than or equal to transient time in pactuning
122 static float rate_limit; // Imports rate limit for curtailment. Must be
less than or equal to rate limits in pactuning
123 //This code supercedes the limits in the PACs but instead controls the
level of curtailment to prevent windup. so limits for green, amber and
red flags must be set in pac tuning to all be to the maximum range(e.g
those of the green flag)
124 static float green_flag_upper_power_limit[50]; // Imports upper limit for
curtailment when a turbines flag is green.
125 static float green_flag_lower_power_limit[50]; // Imports lower limit for
curtailment when a turbines flag is green.
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
126 static float amber_flag_upper_power_limit[50]; // Imports upper limit for
      curtailment when a turbines flag is amber.
127 static float amber_flag_lower_power_limit[50]; // Imports lower limit for
      curtailment when a turbines flag is amber.
128 static float red_flag_upper_power_limit; // Imports upper limit for
      curtailment when a turbines flag is red.
129 static float red_flag_lower_power_limit; // Imports lower limit for
      curtailment when a turbines flag is red.
130 static int strat; // 1-curtailment is equally distributed, 2-curtailment
      is allocated by the uncurtailed power estimated by the cube of the wind
      speed, 3- curtailment is allocated to maintain the tower moment and to
      the proportion of the square of the wind speed so that turbines with
      higher wind speeds get greater changes in curtailment allocation.
131 //
132 static float green_flag_upper_power_limit_val;
133 static float green_flag_lower_power_limit_val;
134 static float amber_flag_upper_power_limit_val;
135 static float amber_flag_lower_power_limit_val;
136
137 if (*prSimTime < 0.02) {
138     //Calls the text parameter file for external tuning:
139
140     if (!myfile.is_open()) //checks if file is properly opened
141     {
142         cout << "Unable to open file"; //Displays "Unable to open file"
      when above value is false
143         //?????
144     }
145     float wfc_parameters[11]; //creates array to store text file data.
146     int i = 1;
147     string words;
148     for (i = 1; i < 22; i += 1){
149         myfile >> words; //used as a means of ignoring the parameter titles
      within the for loop.
150         myfile >> wfc_parameters[i]; //reads line of text from myfile and
      puts it into array earnings with i being line number
151     }
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
152
153
154
155
156     Ki = wfc_parameters[1]; //Imports WFC integral constant for PI
        controller
157     Kp = wfc_parameters[2]; //Imports WFC proportional constant for PI
        controller
158     start_time = wfc_parameters[3]; // Imports start time for curtailment.
        Must be greater than or equal to transient time in pactuning
159     rate_limit = wfc_parameters[4]; // Imports rate limit for curtailment.
        Must be less than or equal to rate limits in pactuning
160     //This code supercedes the limits in the PACs but instead controls the
        level of curtailment to prevent windup. so limits for green, amber and
        red flags must be set in pac tuning to all be to the maximum range(e.g
        those of the green flag)
161     green_flag_upper_power_limit_val = wfc_parameters[5]; // Imports upper
        limit for curtailment when a turbines flag is green.
162     green_flag_lower_power_limit_val = wfc_parameters[6]; // Imports lower
        limit for curtailment when a turbines flag is green.
163     amber_flag_upper_power_limit_val = wfc_parameters[7]; // Imports upper
        limit for curtailment when a turbines flag is amber.
164     amber_flag_lower_power_limit_val = wfc_parameters[8]; // Imports lower
        limit for curtailment when a turbines flag is amber.
165     red_flag_upper_power_limit = wfc_parameters[9]; // Imports upper limit
        for curtailment when a turbines flag is red.
166     red_flag_lower_power_limit = wfc_parameters[10]; // Imports lower limit
        for curtailment when a turbines flag is red.
167     strat = wfc_parameters[11]; // 1-curtailment is equally distributed, 2-
        curtailment is allocated by the uncurtailed power estimated by the cube
        of the wind speed, 3- curtailment is allocated to maintain the tower
        moment and to the proportion of the square of the wind speed so that
        turbines with higher wind speeds get greater changes in curtailment
        allocation.
168     //
169     myfile.close();
170 }
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
171 //-----Internal declaratios-----//
172 float delP_new, Pdem_new, A, B, Ptot, dt;
173 static float delP_old, Pdem_old;
174 static float turbinecurtreq_old[50];
175 static float turbinecurtreq_new[50];
176 static float fi[50]; //fi is used to allocate curtailment ratios to each
    wind turbine.
177 static int reccheck[50]; // reccheck is used to check whether a wind
    turbine has gone into recovery. This is used to prevent the farm level
    controller winding up.
178 float reffset = 0; //Allows for an offset in the farm level anti windup
    when a turbine goes into recovery.
179 Ptot = 0;
180 dt = 0.01; // hard coded step size
181 //calculate power out from farm total by summing each turbine
182 for (iCntTurbines = 0; iCntTurbines < iNUM_TURBINES; iCntTurbines++)
183 {
184     Ptot += parPowerToFarm[iCntTurbines]; // calcualte power output from
    farm by summing power from each turbine
185     rAvailPow += arPowNoPac[iCntTurbines]; //gives the estimated power
    of the wind farm from PAC
186 }
187
188
189 int gsum[50]; //gsum is used for the total number of turbines which are
    not in recovery(needed in the anti-windup section)
190 // do a sum of flag states for the farm - FLAGSUM used in fraction to
    divide up the delP accross turbines accounting for their flag status
191 float FLAGSUM; //FLAGSUM is the total of all of the power allocation flags
    in the windfarm.
192
193 if (*prSimTime > -10) {
194     for (iCntTurbines = 0; iCntTurbines < iNUM_TURBINES; iCntTurbines++) {
195         //cycle through turbines
196         if (abRecoveryFlag[iCntTurbines] == 1) {
197             gsum[iCntTurbines] = 0; // If the turbine is in recovery it is
            treated as a red flag turbine as it cannot be curtailed.
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
197
198     }
199     else{
200         if (abRedFlag[iCntTurbines] == 1) {
201             gsum[iCntTurbines] = 0;
202         }
203         else {
204             gsum[iCntTurbines] = 1;
205         }
206     }
207 }
208 }
209
210 float MAXDP_low = 0;
211 float MAXDP_high = 0;
212
213 if (*prSimTime < 0.02) {
214     delP_old = 0;
215     Pdem_old = 0;
216 }
217
218
219
220 static int inertia_rec_flag_m1;
221 static float inertia_time;
222 int inertia_rec_flag=0;
223
224 if (*prInertiaSwitch == 0 && inertia_rec_flag_m1 == 1){
225
226     inertia_time = *prSimTime ;
227 }
228
229
230
231 if (*prSimTime<inertia_time)
232 {
233     inertia_rec_flag = 1;
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
234 }
235
236
237
238
239 // Anti Wind up section
240 // need to know max allowable power change per time step
241
242
243 for (iCntTurbines = 0; iCntTurbines < iNUM_TURBINES; iCntTurbines++) {
244     green_flag_upper_power_limit[iCntTurbines] =
245     green_flag_upper_power_limit_val;
246     amber_flag_upper_power_limit[iCntTurbines] =
247     amber_flag_upper_power_limit_val;
248
249     green_flag_lower_power_limit[iCntTurbines] =
250     green_flag_lower_power_limit_val;
251     amber_flag_lower_power_limit[iCntTurbines] =
252     amber_flag_lower_power_limit_val;
253
254     if (abGreenFlag[iCntTurbines] == 1 && (abRecoveryFlag[iCntTurbines] == 0
255     || abSubFlagRecoveryComplete[iCntTurbines] == 1)) {
256         // Green flag can change power by X MW
257         // Recovery Flag Can hinder power change
258         MAXDP_low += green_flag_lower_power_limit[iCntTurbines];
259
260         MAXDP_high += green_flag_upper_power_limit[iCntTurbines];
261     }
262     else {
263         if (abAmberFlag[iCntTurbines] == 1 && (abRecoveryFlag[iCntTurbines] ==
264         0 || abSubFlagRecoveryComplete[iCntTurbines] == 1)) {
265             // Amber Flag Can Change Power by Y MW
266             // Recovery Flag Can hinder power change
267             MAXDP_low += amber_flag_lower_power_limit[iCntTurbines];
268             MAXDP_high += amber_flag_upper_power_limit[iCntTurbines];
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
265     }
266     else {
267         // Red Flag Can't Change Power
268         MAXDP_low += red_flag_lower_power_limit;
269         MAXDP_high += red_flag_upper_power_limit;
270     }
271 }
272 }
273
274 //-----
275 FLAGSUM = 0; //gsum set initially to zero
276 float vel_limit[50];
277 float vel_limit_sum = 0;
278 float gsum_sum = 0; // total velocity limit of the farm
279 float gsum_count = 0;
280
281 for (iCntTurbines = 0; iCntTurbines < iNUM_TURBINES; iCntTurbines++) {
282     if (*prInertiaSwitch == 1 || inertia_rec_flag==1){
283         vel_limit[iCntTurbines] = rate_limit * 10;
284     }
285     else{
286         vel_limit[iCntTurbines] = rate_limit; // set up here as a for loop so
287         that work can be done on possibly varying rate limits.
288     }
289 }
290 for (iCntTurbines = 0; iCntTurbines < iNUM_TURBINES; iCntTurbines++) {
291     vel_limit_sum += vel_limit[iCntTurbines];
292     gsum_sum += vel_limit[iCntTurbines] * gsum[iCntTurbines];
293     gsum_count += gsum[iCntTurbines];
294 }
295
296
297 for (iCntTurbines = 0; iCntTurbines < iNUM_TURBINES; iCntTurbines++) {
298     if (reccheck[iCntTurbines] < abRecoveryFlag[iCntTurbines]){//changed
299         from >#####
300         recoffset += turbinecurtreq_old[iCntTurbines];
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
300     }
301 }
302
303 for (iCntTurbines = 0; iCntTurbines < iNUM_TURBINES; iCntTurbines++){
304     //cycle through farm
305     if (abRecoveryFlag[iCntTurbines] == 1){
306         // turbinecurtreq_old[iCntTurbines] = 0; //????possibly defunct
307         #####
308     }
309 }
310
311 //A discretised PI controller is used for set-point tracking.
312 A = Kp + ((dt / 2)*Ki);
313 B = ((dt / 2)*Ki) - Kp;
314
315 if (*prInertiaSwitch == 1){
316     Pdem_new = 25000000 - Ptot;//          //demanded curtailment each time
317     step
318     delP_new = delP_old + A * Pdem_new + B * Pdem_old;//          //calcualte
319     the new delta P
320 }
321 else{
322     if (*prSimTime > start_time){
323         Pdem_new = *prCurtailmentValue - Ptot;//          //demanded curtailment
324         each time step
325         delP_new = delP_old + A * Pdem_new + B * Pdem_old;//          //
326         calcualte the new delta P
327     }
328 }
329
330 //-----Controller level Antiwindup-----//
331 // Anti-windup section
332 //Total DelP cannot go above or below the MAXDP_high or the MAXDP_low
333 determined earlier in the code
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
330
331  if (delP_new > MAXDP_high)
332  {
333      delP_new = MAXDP_high;
334      Pdem_new = (delP_new - delP_old - B * Pdem_old) / A;
335  }
336
337  if (delP_new < MAXDP_low)
338  {
339      delP_new = MAXDP_low;
340      Pdem_new = (delP_new - delP_old - B * Pdem_old) / A;
341  }
342
343  //As there are limits in the PAC on how quickly curtailment levels can
344  // change for each turbine there is also antiwindup limits on the rate of
345  // change of curtailment.
346  // These limits must remain smooth except in the case where a turbine goes
347  // into recovery where a large discontinuity may be included as the
348  // recovering turbine is no longer being assigned any curtailment.
349
350  if (delP_new - delP_old < -gsum_sum + reffset){
351
352      delP_new = delP_old - gsum_sum + reffset;
353      Pdem_new = (delP_new - delP_old - B * Pdem_old) / A;
354  }
355
356  if (delP_new - delP_old > gsum_sum + reffset){
357      delP_new = delP_old + gsum_sum + reffset;
358      Pdem_new = (delP_new - delP_old - B * Pdem_old) / A;
359  }
360
361  // This section allocates the flags across the wind turbines for the
362  // allocation of curatiltment. The strat flag is used to decide which of the
363  // strategies is used. The strat flag also acts as a switch later for
364  // whether reallocation is used to avoid high tower moments.
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
359 for (iCntTurbines = 0; iCntTurbines < iNUM_TURBINES; iCntTurbines++) {
360     //cycle through turbines
361     if (abRecoveryFlag[iCntTurbines] == 0 || abSubFlagRecoveryComplete[
362         iCntTurbines] == 1){
363         if (arEWS[iCntTurbines] >= 8){
364             if (strat == 1)
365             {
366                 fi[iCntTurbines] = 3 * abGreenFlag[iCntTurbines] + 1 * abAmberFlag
367                 [iCntTurbines] + 0 * abRedFlag[iCntTurbines];
368                 // Allocation based on traffic light flags from sung-ho Hurr
369             }
370             if (strat == 2)
371             { //Allocates based on uncurtailed power.
372                 if (arEWS[iCntTurbines] < 11.2){
373                     fi[iCntTurbines] = arEWS[iCntTurbines] * arEWS[iCntTurbines] *
374                     arEWS[iCntTurbines] * abGreenFlag[iCntTurbines] + 1 * abAmberFlag[
375                     iCntTurbines] + 0 * abRedFlag[iCntTurbines];
376                 }
377                 else{
378                     fi[iCntTurbines] = 11.2*11.2*11.2 * abGreenFlag[iCntTurbines] +
379                     1 * abAmberFlag[iCntTurbines] + 0 * abRedFlag[iCntTurbines];
380                 }
381             }
382             if (strat == 3)
383             {
384                 fi[iCntTurbines] = 3 * abGreenFlag[iCntTurbines] + 1 * abAmberFlag
385                 [iCntTurbines] + 0 * abRedFlag[iCntTurbines];
386             }
387         }
388     }
389     else{
390         fi[iCntTurbines] = 0;
391     }
392 }
393 }
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
389 for (iCntTurbines = 0; iCntTurbines < iNUM_TURBINES; iCntTurbines++) {
390     FLAGSUM += fi[iCntTurbines];
391 }
392
393 //-----Turbine Level Anti-Windup
394 //-----//
395 //This section is required so that the wind farm controller does not
396 //allocate an unachivable amount of curtailment to a turbine.
397 float recurt = 0;
398 int ambsum = 0;
399 int satup = 0;
400 int satdown = 0;
401 float sat[50];
402 int a;
403 float gi[50];
404 float satsum = 0;
405
406 for (iCntTurbines = 0; iCntTurbines < iNUM_TURBINES; iCntTurbines++){
407     //cycle through farm to flag turbines in recovery
408     if (abRecoveryFlag[iCntTurbines] == 0 || abSubFlagRecoveryComplete[
409         iCntTurbines] == 1){
410         sat[iCntTurbines] = 0;
411     }
412     else{ sat[iCntTurbines] = 5; }
413 }
414 //The section below checks whether each turbine is being allocated a level
415 //of curtailment that it can achive with experiancing windup.
416 //It also attempts to reallocate the curtailment that cannot be achived by
417 //a certain turbine to other turbines.
418
419 float highcurt = 0;
420 float abovecurt[50];
421
422 for (iCntTurbines = 0; iCntTurbines < iNUM_TURBINES; iCntTurbines++){
423     abovecurt[iCntTurbines] = 0;
424 }
425
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
421 // This section preallocates the curtailment across the wind farm to
    // priorities curtailment where wind speeds are near the rated wind speed
    // where the tower bending moment is at its highest.
422 if (strat == 3){
423     for (iCntTurbines = 0; iCntTurbines < iNUM_TURBINES; iCntTurbines++){
424         if (abRecoveryFlag[iCntTurbines] == 0 || abSubFlagRecoveryComplete[
            iCntTurbines] == 1){
425             if (abRedFlag[iCntTurbines] == 0){
426                 if (arEWS[iCntTurbines] > 19){
427                     highcurt += 1000000;
428                     abovecurt[iCntTurbines] = 1000000;
429                 }
430             else{
431                 if (arEWS[iCntTurbines] > 15.3){
432                     highcurt += ((-15.3 + arEWS[iCntTurbines]) * 1000000) / 3.7;
433                     abovecurt[iCntTurbines] = ((-15.3 + arEWS[iCntTurbines]) *
            1000000) / 3.7;
434                 }
435             else{
436                 if (arEWS[iCntTurbines] > 11.2){
437                     highcurt += ((-15.3 + arEWS[iCntTurbines]) * 15.65 * 100000)
            / 4.1;
438                     abovecurt[iCntTurbines] = ((-15.3 + arEWS[iCntTurbines]) *
            15.65 * 100000) / 4.1;
439                 }
440             else{
441                 if (arEWS[iCntTurbines] > 9){
442                     highcurt += -2.8 * 100000 + ((-9 + arEWS[iCntTurbines]) *
            -12.85 * 100000) / 2.2;
443                     abovecurt[iCntTurbines] = -2.8 * 100000 + ((-9 + arEWS[
            iCntTurbines]) * -12.85 * 100000) / 2.2;
444                 }
445             else{
446                 if (arEWS[iCntTurbines] > 8){
447                     highcurt += ((-8 + arEWS[iCntTurbines]) * -2.8 * 100000)
            / 1;
448                 }
            }
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
449         abovecurt[iCntTurbines] = ((-8 + arEWS[iCntTurbines]) *
450         -2.8 * 100000) / 1;
451     }
452 }
453 }
454 }
455 }
456 }
457 }
458 }
459 else{
460     for (iCntTurbines = 0; iCntTurbines < iNUM_TURBINES; iCntTurbines++){
461         highcurt += 0;
462         abovecurt[iCntTurbines] = 0;
463     }
464 }
465
466
467 //-----Turbine level allocation and antiwindup-----//
468
469 //This section of the code checks each turbine to ensure that the
470 //allocation of curtailment doesn't cause any turbine to be allocated an
471 //undeliverable level of curtailment. It then reallocates the curtailment
472 //to turbines which are not saturated to reduce the deviations from the
473 //allocated set-point at the farm level.
474 //This could use some refinement when using different curtailment
475 //stratergies.
476
477 if (*prSimTime > start_time && FLAGSUM > 0) {
478     if (*pbCurtailmentFlagRequest == 1 && delP_new != 0) {
479         for (iCntTurbines = 0; iCntTurbines < iNUM_TURBINES; iCntTurbines++){
480             if (abRecoveryFlag[iCntTurbines] == 0 || abSubFlagRecoveryComplete[
481             iCntTurbines] == 1){
482
483                 turbinecurtreq_new[iCntTurbines] = abovecurt[iCntTurbines] + fi[
484                 iCntTurbines] * (delP_new - highcurt) / FLAGSUM; //Allocates the
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
potentially woundup level of curtailment to each turbine.
478
479
480     if (abGreenFlag[iCntTurbines] == 1){
481         if (turbinecurtreq_new[iCntTurbines] <=
green_flag_lower_power_limit[iCntTurbines]){
482
483             recurt += turbinecurtreq_new[iCntTurbines] -
green_flag_lower_power_limit[iCntTurbines];
484             turbinecurtreq_new[iCntTurbines] =
green_flag_lower_power_limit[iCntTurbines];
485             sat[iCntTurbines] = -1;
486             if (turbinecurtreq_new[iCntTurbines] - turbinecurtreq_old[
iCntTurbines] <= -vel_limit[iCntTurbines]){
487                 recurt += turbinecurtreq_new[iCntTurbines] -
turbinecurtreq_old[iCntTurbines] + vel_limit[iCntTurbines];
488                 turbinecurtreq_new[iCntTurbines] = turbinecurtreq_old[
iCntTurbines] - vel_limit[iCntTurbines];
489                 sat[iCntTurbines] = -1;
490             }
491             if (turbinecurtreq_new[iCntTurbines] - turbinecurtreq_old[
iCntTurbines] >= vel_limit[iCntTurbines]){
492                 recurt += turbinecurtreq_new[iCntTurbines] -
turbinecurtreq_old[iCntTurbines] - vel_limit[iCntTurbines];
493                 turbinecurtreq_new[iCntTurbines] = turbinecurtreq_old[
iCntTurbines] + vel_limit[iCntTurbines];
494                 sat[iCntTurbines] = 1;
495             }
496         }
497     else{
498         if (turbinecurtreq_new[iCntTurbines] >=
green_flag_upper_power_limit[iCntTurbines]){
499
500             recurt += turbinecurtreq_new[iCntTurbines] -
green_flag_upper_power_limit[iCntTurbines];
501             turbinecurtreq_new[iCntTurbines] =
green_flag_upper_power_limit[iCntTurbines];
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
502
503         sat[iCntTurbines] = 1;
504         if (turbinecurtreq_new[iCntTurbines] - turbinecurtreq_old[
iCntTurbines] <= -vel_limit[iCntTurbines]){
505             recurt += turbinecurtreq_new[iCntTurbines] -
turbinecurtreq_old[iCntTurbines] + vel_limit[iCntTurbines];
506             turbinecurtreq_new[iCntTurbines] = turbinecurtreq_old[
iCntTurbines] - vel_limit[iCntTurbines];
507             sat[iCntTurbines] = -1;
508         }
509         if (turbinecurtreq_new[iCntTurbines] - turbinecurtreq_old[
iCntTurbines] >= vel_limit[iCntTurbines]){
510             recurt += turbinecurtreq_new[iCntTurbines] -
turbinecurtreq_old[iCntTurbines] - vel_limit[iCntTurbines];
511             turbinecurtreq_new[iCntTurbines] = turbinecurtreq_old[
iCntTurbines] + vel_limit[iCntTurbines];
512             sat[iCntTurbines] = 1;
513         }
514     }
515     else{
516         if (turbinecurtreq_new[iCntTurbines] - turbinecurtreq_old[
iCntTurbines] <= -vel_limit[iCntTurbines]){
517             recurt += turbinecurtreq_new[iCntTurbines] -
turbinecurtreq_old[iCntTurbines] + vel_limit[iCntTurbines];
518             turbinecurtreq_new[iCntTurbines] = turbinecurtreq_old[
iCntTurbines] - vel_limit[iCntTurbines];
519             sat[iCntTurbines] = -1;
520         }
521         if (turbinecurtreq_new[iCntTurbines] - turbinecurtreq_old[
iCntTurbines] >= vel_limit[iCntTurbines]){
522             recurt += turbinecurtreq_new[iCntTurbines] -
turbinecurtreq_old[iCntTurbines] - vel_limit[iCntTurbines];
523             turbinecurtreq_new[iCntTurbines] = turbinecurtreq_old[
iCntTurbines] + vel_limit[iCntTurbines];
524             sat[iCntTurbines] = 1;
525         }
526     }
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
527     }
528 }
529     if (abAmberFlag[iCntTurbines] == 1){
530         if (turbinecurtreq_new[iCntTurbines] <=
531         amber_flag_lower_power_limit[iCntTurbines]){
532             recurt += turbinecurtreq_new[iCntTurbines] -
533             amber_flag_lower_power_limit[iCntTurbines];
534             turbinecurtreq_new[iCntTurbines] =
535             amber_flag_lower_power_limit[iCntTurbines];
536             sat[iCntTurbines] = -1;
537             if (turbinecurtreq_new[iCntTurbines] - turbinecurtreq_old[
538             iCntTurbines] <= -vel_limit[iCntTurbines]){
539                 recurt += turbinecurtreq_new[iCntTurbines] -
540                 turbinecurtreq_old[iCntTurbines] + vel_limit[iCntTurbines];
541                 turbinecurtreq_new[iCntTurbines] = turbinecurtreq_old[
542                 iCntTurbines] - vel_limit[iCntTurbines];
543                 sat[iCntTurbines] = -1;
544             }
545             if (turbinecurtreq_new[iCntTurbines] - turbinecurtreq_old[
546             iCntTurbines] >= vel_limit[iCntTurbines]){
547                 recurt += turbinecurtreq_new[iCntTurbines] -
548                 turbinecurtreq_old[iCntTurbines] - vel_limit[iCntTurbines];
549                 turbinecurtreq_new[iCntTurbines] = turbinecurtreq_old[
550                 iCntTurbines] + vel_limit[iCntTurbines];
551                 sat[iCntTurbines] = 1;
552             }
553         }
554     }
555     else{
556         if (turbinecurtreq_new[iCntTurbines] >=
557         amber_flag_upper_power_limit[iCntTurbines]){
558             recurt += turbinecurtreq_new[iCntTurbines] -
559             amber_flag_upper_power_limit[iCntTurbines];
560             turbinecurtreq_new[iCntTurbines] =
561             amber_flag_upper_power_limit[iCntTurbines];
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
552         sat[iCntTurbines] = 1;
553         if (turbinecurtreq_new[iCntTurbines] - turbinecurtreq_old[
iCntTurbines] < -vel_limit[iCntTurbines]){
554             recurt += turbinecurtreq_new[iCntTurbines] -
turbinecurtreq_old[iCntTurbines] + vel_limit[iCntTurbines];
555             turbinecurtreq_new[iCntTurbines] = turbinecurtreq_old[
iCntTurbines] - vel_limit[iCntTurbines];
556             sat[iCntTurbines] = -1;
557         }
558         if (turbinecurtreq_new[iCntTurbines] - turbinecurtreq_old[
iCntTurbines] > vel_limit[iCntTurbines]){
559             recurt += turbinecurtreq_new[iCntTurbines] -
turbinecurtreq_old[iCntTurbines] - vel_limit[iCntTurbines];
560             turbinecurtreq_new[iCntTurbines] = turbinecurtreq_old[
iCntTurbines] + vel_limit[iCntTurbines];
561             sat[iCntTurbines] = 1;
562         }
563     }
564     else{
565         if (turbinecurtreq_new[iCntTurbines] - turbinecurtreq_old[
iCntTurbines] < -vel_limit[iCntTurbines]){
566             recurt += turbinecurtreq_new[iCntTurbines] -
turbinecurtreq_old[iCntTurbines] + vel_limit[iCntTurbines];
567             turbinecurtreq_new[iCntTurbines] = turbinecurtreq_old[
iCntTurbines] - vel_limit[iCntTurbines];
568             sat[iCntTurbines] = -1;
569         }
570
571         if (turbinecurtreq_new[iCntTurbines] - turbinecurtreq_old[
iCntTurbines] > vel_limit[iCntTurbines]){
572             recurt += turbinecurtreq_new[iCntTurbines] -
turbinecurtreq_old[iCntTurbines] - vel_limit[iCntTurbines];
573             turbinecurtreq_new[iCntTurbines] = turbinecurtreq_old[
iCntTurbines] + vel_limit[iCntTurbines];
574             sat[iCntTurbines] = 1;
575         }
576     }
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
577         }
578
579
580     }
581 }
582     if (abRedFlag[iCntTurbines] == 1){
583         sat[iCntTurbines] = 2;
584         if (turbinecurtreq_new[iCntTurbines] - turbinecurtreq_old[
585 iCntTurbines] < -vel_limit[iCntTurbines]){
586             recurt += turbinecurtreq_new[iCntTurbines] -
587 turbinecurtreq_old[iCntTurbines] + vel_limit[iCntTurbines];
588             turbinecurtreq_new[iCntTurbines] = turbinecurtreq_old[
589 iCntTurbines] - vel_limit[iCntTurbines];
590             sat[iCntTurbines] = 2;
591         }
592         if (turbinecurtreq_new[iCntTurbines] - turbinecurtreq_old[
593 iCntTurbines] >= vel_limit[iCntTurbines]){
594             recurt += turbinecurtreq_new[iCntTurbines] -
595 turbinecurtreq_old[iCntTurbines] - vel_limit[iCntTurbines];
596             turbinecurtreq_new[iCntTurbines] = turbinecurtreq_old[
597 iCntTurbines] + vel_limit[iCntTurbines];
598             sat[iCntTurbines] = 2;
599         }
600     }
601 }
602     else{
603         turbinecurtreq_new[iCntTurbines] = 0; // fi[iCntTurbines] * (
604 delP_new) / FLAGSUM; //Allocates the potentially woundup level of
605 curtailment to each turbine.)
606     }
607
608     satup = 0;
609     satdown = 0;
610 }
611 for (iCntTurbines = 0; iCntTurbines < iNUM_TURBINES; iCntTurbines++){
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
606     if (abRecoveryFlag[iCntTurbines] == 1 && abSubFlagRecoveryComplete[
iCntTurbines] == 0){
607         sat[iCntTurbines] = 5;
608     }
609     if (sat[iCntTurbines] == 1){
610         satup += 1;
611     }
612     if (sat[iCntTurbines] == -1){
613         satdown += 1;
614     }
615 }
616
617 //-----Sorting -----//
618
619 std::vector<double> wind_speeds_for_sort{};
620
621 for (int i = 0; i < iNUM_TURBINES; i++){
622     wind_speeds_for_sort.insert(wind_speeds_for_sort.end(), arEWS[i]);
623 }
624
625
626 std::vector<int> index = { 0 };
627
628 for (int i = 1; i < iNUM_TURBINES; i++){
629     index.insert(index.end(), i);
630 }
631 // initialize original index locations
632
633 //iota(idx.begin(), idx.end(), 0);
634
635 // sort indexes based on comparing values in v
636 std::sort(index.begin(), index.end(), [&wind_speeds_for_sort](size_t
i1, size_t i2) {return wind_speeds_for_sort[i1] < wind_speeds_for_sort[
i2]; });
637
638 //return index;
639 int while_iter = iNUM_TURBINES - 1;
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
640
641     while (recurt != 0 && while_iter >= 0){
642         iCntTurbines = index[while_iter];
643         if (abRecoveryFlag[iCntTurbines] == 0){
644             //turbinecurtreq_new[iCntTurbines] += recurt;
645             if (abGreenFlag[iCntTurbines] == 1){
646                 if (turbinecurtreq_new[iCntTurbines] + recurt >
647 green_flag_upper_power_limit[iCntTurbines]){
648                     if (green_flag_upper_power_limit[iCntTurbines] -
649 turbinecurtreq_old[iCntTurbines] < -vel_limit[iCntTurbines]){
650                         recurt += turbinecurtreq_new[iCntTurbines] -
651 turbinecurtreq_old[iCntTurbines] + vel_limit[iCntTurbines];
652                         turbinecurtreq_new[iCntTurbines] = turbinecurtreq_old[
653 iCntTurbines] - vel_limit[iCntTurbines];
654                     }
655                 }
656                 else{
657                     if (green_flag_upper_power_limit[iCntTurbines] -
658 turbinecurtreq_old[iCntTurbines] > vel_limit[iCntTurbines]){
659                         recurt += turbinecurtreq_new[iCntTurbines] -
660 turbinecurtreq_old[iCntTurbines] - vel_limit[iCntTurbines];
661                         turbinecurtreq_new[iCntTurbines] = turbinecurtreq_old[
662 iCntTurbines] + vel_limit[iCntTurbines];
663                     }
664                 }
665                 else{
666                     recurt += turbinecurtreq_new[iCntTurbines] -
667 green_flag_upper_power_limit[iCntTurbines];
668                     turbinecurtreq_new[iCntTurbines] =
669 green_flag_upper_power_limit[iCntTurbines];
670                 }
671             }
672         }
673     }
674     else{
675         if (turbinecurtreq_new[iCntTurbines] + recurt <
676 green_flag_lower_power_limit[iCntTurbines]){
677             if (green_flag_lower_power_limit[iCntTurbines] -
678 turbinecurtreq_old[iCntTurbines] < -vel_limit[iCntTurbines]){
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
666         recurt += turbinecurtreq_new[iCntTurbines] -
turbinecurtreq_old[iCntTurbines] + vel_limit[iCntTurbines];
667         turbinecurtreq_new[iCntTurbines] = turbinecurtreq_old[
iCntTurbines] - vel_limit[iCntTurbines];
668     }
669     else{
670         if (green_flag_lower_power_limit[iCntTurbines] -
turbinecurtreq_old[iCntTurbines] > vel_limit[iCntTurbines]){
671             recurt += turbinecurtreq_new[iCntTurbines] -
turbinecurtreq_old[iCntTurbines] - vel_limit[iCntTurbines];
672             turbinecurtreq_new[iCntTurbines] = turbinecurtreq_old[
iCntTurbines] + vel_limit[iCntTurbines];
673         }
674         else{
675
676             recurt += turbinecurtreq_new[iCntTurbines] -
green_flag_lower_power_limit[iCntTurbines];
677             turbinecurtreq_new[iCntTurbines] =
green_flag_lower_power_limit[iCntTurbines];
678         }
679     }
680 }
681 else{
682     if (turbinecurtreq_new[iCntTurbines] + recurt -
turbinecurtreq_old[iCntTurbines] < -vel_limit[iCntTurbines]){
683         recurt += turbinecurtreq_new[iCntTurbines] -
turbinecurtreq_old[iCntTurbines] + vel_limit[iCntTurbines];
684         turbinecurtreq_new[iCntTurbines] = turbinecurtreq_old[
iCntTurbines] - vel_limit[iCntTurbines];
685     }
686     else{
687         if (turbinecurtreq_new[iCntTurbines] + recurt -
turbinecurtreq_old[iCntTurbines] > vel_limit[iCntTurbines]){
688             recurt += turbinecurtreq_new[iCntTurbines] -
turbinecurtreq_old[iCntTurbines] - vel_limit[iCntTurbines];
689             turbinecurtreq_new[iCntTurbines] = turbinecurtreq_old[
iCntTurbines] + vel_limit[iCntTurbines];
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
690         }
691         else{
692             turbinecurtreq_new[iCntTurbines] += recurt;
693             recurt = 0;
694         }
695     }
696
697     }
698 }
699 }
700
701 //---
702 if (abAmberFlag[iCntTurbines] == 1){
703     if (turbinecurtreq_new[iCntTurbines] + recurt >
704         amber_flag_upper_power_limit[iCntTurbines]){
705         if (amber_flag_upper_power_limit[iCntTurbines] -
706             turbinecurtreq_old[iCntTurbines] < -vel_limit[iCntTurbines]){
707             recurt += turbinecurtreq_new[iCntTurbines] -
708                 turbinecurtreq_old[iCntTurbines] + vel_limit[iCntTurbines];
709             turbinecurtreq_new[iCntTurbines] = turbinecurtreq_old[
710                 iCntTurbines] - vel_limit[iCntTurbines];
711         }
712         else{
713             if (amber_flag_upper_power_limit[iCntTurbines] -
714                 turbinecurtreq_old[iCntTurbines] > vel_limit[iCntTurbines]){
715                 recurt += turbinecurtreq_new[iCntTurbines] -
716                     turbinecurtreq_old[iCntTurbines] - vel_limit[iCntTurbines];
717                 turbinecurtreq_new[iCntTurbines] = turbinecurtreq_old[
718                     iCntTurbines] + vel_limit[iCntTurbines];
719             }
720             else{
721                 recurt += turbinecurtreq_new[iCntTurbines] -
722                     amber_flag_upper_power_limit[iCntTurbines];
723                 turbinecurtreq_new[iCntTurbines] =
724                     amber_flag_upper_power_limit[iCntTurbines];
725             }
726         }
727     }
728 }
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
718         }
719         else{
720
721             if (turbinecurtreq_new[iCntTurbines] + recurt <
amber_flag_lower_power_limit[iCntTurbines]){
722                 if (amber_flag_lower_power_limit[iCntTurbines] -
turbinecurtreq_old[iCntTurbines] < -vel_limit[iCntTurbines]){
723                     recurt += turbinecurtreq_new[iCntTurbines] -
turbinecurtreq_old[iCntTurbines] + vel_limit[iCntTurbines];
724                     turbinecurtreq_new[iCntTurbines] = turbinecurtreq_old[
iCntTurbines] - vel_limit[iCntTurbines];
725                 }
726                 else{
727                     if (amber_flag_lower_power_limit[iCntTurbines] -
turbinecurtreq_old[iCntTurbines] > vel_limit[iCntTurbines]){
728                         recurt += turbinecurtreq_new[iCntTurbines] -
turbinecurtreq_old[iCntTurbines] - vel_limit[iCntTurbines];
729                         turbinecurtreq_new[iCntTurbines] = turbinecurtreq_old[
iCntTurbines] + vel_limit[iCntTurbines];
730                     }
731                     else{
732
733                         recurt += turbinecurtreq_new[iCntTurbines] -
amber_flag_lower_power_limit[iCntTurbines];
734                         turbinecurtreq_new[iCntTurbines] =
amber_flag_lower_power_limit[iCntTurbines];
735                     }
736                 }
737             }
738             else{
739                 if (turbinecurtreq_new[iCntTurbines] + recurt -
turbinecurtreq_old[iCntTurbines] < -vel_limit[iCntTurbines]){
740                     recurt += turbinecurtreq_new[iCntTurbines] -
turbinecurtreq_old[iCntTurbines] + vel_limit[iCntTurbines];
741                     turbinecurtreq_new[iCntTurbines] = turbinecurtreq_old[
iCntTurbines] - vel_limit[iCntTurbines];
742                 }

```

Appendix E. Wind Farm Distribution Controller C++ Code

```
743         else{
744             if (turbinecurtreq_new[iCntTurbines] + recurt -
turbinecurtreq_old[iCntTurbines] > vel_limit[iCntTurbines]){
745                 recurt += turbinecurtreq_new[iCntTurbines] -
turbinecurtreq_old[iCntTurbines] - vel_limit[iCntTurbines];
746                 turbinecurtreq_new[iCntTurbines] = turbinecurtreq_old[
iCntTurbines] + vel_limit[iCntTurbines];
747             }
748             else{
749                 turbinecurtreq_new[iCntTurbines] += recurt;
750                 recurt = 0;
751             }
752         }
753
754     }
755 }
756 }
757 //---
758 if (abRedFlag[iCntTurbines] == 1){
759     if (turbinecurtreq_new[iCntTurbines] + recurt >
red_flag_upper_power_limit){
760         if (red_flag_upper_power_limit - turbinecurtreq_old[
iCntTurbines] < -vel_limit[iCntTurbines]){
761             recurt += turbinecurtreq_new[iCntTurbines] -
turbinecurtreq_old[iCntTurbines] + vel_limit[iCntTurbines];
762             turbinecurtreq_new[iCntTurbines] = turbinecurtreq_old[
iCntTurbines] - vel_limit[iCntTurbines];
763         }
764         else{
765             if (red_flag_upper_power_limit - turbinecurtreq_old[
iCntTurbines] > vel_limit[iCntTurbines]){
766                 recurt += turbinecurtreq_new[iCntTurbines] -
turbinecurtreq_old[iCntTurbines] - vel_limit[iCntTurbines];
767                 turbinecurtreq_new[iCntTurbines] = turbinecurtreq_old[
iCntTurbines] + vel_limit[iCntTurbines];
768             }
769             else{
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
770         recurt += turbinecurtreq_new[iCntTurbines] -
red_flag_upper_power_limit;
771         turbinecurtreq_new[iCntTurbines] =
red_flag_upper_power_limit;
772     }
773 }
774 }
775 else{
776
777     if (turbinecurtreq_new[iCntTurbines] + recurt <
red_flag_lower_power_limit){
778         if (red_flag_lower_power_limit - turbinecurtreq_old[
iCntTurbines] < -vel_limit[iCntTurbines]){
779             recurt += turbinecurtreq_new[iCntTurbines] -
turbinecurtreq_old[iCntTurbines] + vel_limit[iCntTurbines];
780             turbinecurtreq_new[iCntTurbines] = turbinecurtreq_old[
iCntTurbines] - vel_limit[iCntTurbines];
781         }
782         else{
783             if (red_flag_lower_power_limit - turbinecurtreq_old[
iCntTurbines] > vel_limit[iCntTurbines]){
784                 recurt += turbinecurtreq_new[iCntTurbines] -
turbinecurtreq_old[iCntTurbines] - vel_limit[iCntTurbines];
785                 turbinecurtreq_new[iCntTurbines] = turbinecurtreq_old[
iCntTurbines] + vel_limit[iCntTurbines];
786             }
787             else{
788
789                 recurt += turbinecurtreq_new[iCntTurbines] -
red_flag_lower_power_limit;
790                 turbinecurtreq_new[iCntTurbines] =
red_flag_lower_power_limit;
791             }
792         }
793     }
794     else{
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
795         if (turbinecurtreq_new[iCntTurbines] + recurt -
796             turbinecurtreq_old[iCntTurbines] < -vel_limit[iCntTurbines]){
797             recurt += turbinecurtreq_new[iCntTurbines] -
798                 turbinecurtreq_old[iCntTurbines] + vel_limit[iCntTurbines];
799             turbinecurtreq_new[iCntTurbines] = turbinecurtreq_old[
800 iCntTurbines] - vel_limit[iCntTurbines];
801         }
802     else{
803         if (turbinecurtreq_new[iCntTurbines] + recurt -
804             turbinecurtreq_old[iCntTurbines] > vel_limit[iCntTurbines]){
805             recurt += turbinecurtreq_new[iCntTurbines] -
806                 turbinecurtreq_old[iCntTurbines] - vel_limit[iCntTurbines];
807             turbinecurtreq_new[iCntTurbines] = turbinecurtreq_old[
808 iCntTurbines] + vel_limit[iCntTurbines];
809         }
810     else{
811         turbinecurtreq_new[iCntTurbines] += recurt;
812         recurt = 0;
813     }
814 }
815 }
816 }
817 }
818 }
819 }
820 }
821 }
822 }
823 }
824 }
825 }
826 }
827 }
828 }
829 }
830 }
831 }
832 }
833 }
834 }
835 }
836 }
837 }
838 }
839 }
840 }
841 }
842 }
843 }
844 }
845 }
846 }
847 }
848 }
849 }
850 }
851 }
852 }
853 }
854 }
855 }
856 }
857 }
858 }
859 }
860 }
861 }
862 }
863 }
864 }
865 }
866 }
867 }
868 }
869 }
870 }
871 }
872 }
873 }
874 }
875 }
876 }
877 }
878 }
879 }
880 }
881 }
882 }
883 }
884 }
885 }
886 }
887 }
888 }
889 }
890 }
891 }
892 }
893 }
894 }
895 }
896 }
897 }
898 }
899 }
900 }
901 }
902 }
903 }
904 }
905 }
906 }
907 }
908 }
909 }
910 }
911 }
912 }
913 }
914 }
915 }
916 }
917 }
918 }
919 }
920 }
921 }
922 }
923 }
924 }
925 }
926 }
927 }
928 }
929 }
930 }
931 }
932 }
933 }
934 }
935 }
936 }
937 }
938 }
939 }
940 }
941 }
942 }
943 }
944 }
945 }
946 }
947 }
948 }
949 }
950 }
951 }
952 }
953 }
954 }
955 }
956 }
957 }
958 }
959 }
960 }
961 }
962 }
963 }
964 }
965 }
966 }
967 }
968 }
969 }
970 }
971 }
972 }
973 }
974 }
975 }
976 }
977 }
978 }
979 }
980 }
981 }
982 }
983 }
984 }
985 }
986 }
987 }
988 }
989 }
990 }
991 }
992 }
993 }
994 }
995 }
996 }
997 }
998 }
999 }
1000 }
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
826 for (iCntTurbines = 0; iCntTurbines < iNUM_TURBINES; iCntTurbines++){
      //cycle through farm
827 if (*prSimTime < 100){
828     pac_on_hyst[iCntTurbines] = 0;
829 }
830 else{
831     if (pac_on_hyst[iCntTurbines] == 1){
832         if (arEWS[iCntTurbines]<6){ pac_on_hyst[iCntTurbines] = 0; }
833         else{ pac_on_hyst[iCntTurbines] = 1; }
834     }
835     else{
836         if (arEWS[iCntTurbines]>8){ pac_on_hyst[iCntTurbines] = 1; }
837         else{ pac_on_hyst[iCntTurbines] = 0; }
838     }
839 }
840
841 }
842 //}
843
844 float testsum = 0;
845
846 if (*prSimTime > start_time) {
847     if (*prInertiaSwitch == 1 || inertia_rec_flag == 1){
848         for (iCntTurbines = 0; iCntTurbines < iNUM_TURBINES; iCntTurbines++){
849             parReqChangeInPower[iCntTurbines] = turbinecurtreq_new[iCntTurbines
850 ]+250000;//turbinecurtreq_new[iCntTurbines];
851             pabPacOnFlag[iCntTurbines] = 1;// pac_on_hyst[iCntTurbines];
852             pabPriorityFlag[iCntTurbines] = 1;
853             pabFastSlowFlag[iCntTurbines] = 0;
854             turbinecurtreq_old[iCntTurbines] = turbinecurtreq_new[iCntTurbines];
855             testsum += turbinecurtreq_new[iCntTurbines];
856         }
857         delP_old = testsum; //set delta P old for next
858         iteration
859         //delP_old = delP_new;
860         Pdem_old = Pdem_new;
861         for (iCntTurbines = 0; iCntTurbines < iNUM_TURBINES; iCntTurbines++) {
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
860     reccheck[iCntTurbines] = abRecoveryFlag[iCntTurbines];
861   }
862 }
863 else{
864   if (*pbCurtailmentFlagRequest == 1){
865     for (iCntTurbines = 0; iCntTurbines < iNUM_TURBINES; iCntTurbines++)
866     {
867       pabPacOnFlag[iCntTurbines] = pac_on_hyst[iCntTurbines];
868       pabPriorityFlag[iCntTurbines] = 0;
869     }
870   }
871   else{
872     for (iCntTurbines = 0; iCntTurbines < iNUM_TURBINES; iCntTurbines++)
873     {
874       pabPacOnFlag[iCntTurbines] = 0;
875     }
876   }
877   if (*pbCurtailmentFlagRequest == 1) { //operate when
878     curtailment requested
879     if (strat == 4){
880       for (iCntTurbines = 0; iCntTurbines < iNUM_TURBINES; iCntTurbines++)
881       { //cycle through farm
882         //if (abRecoveryFlag[iCntTurbines] == 0){
883           parReqChangeInPower[iCntTurbines] = *prCurtailmentValue /
884           iNUM_TURBINES;
885           parCurtailmentPower[iCntTurbines] = *prCurtailmentValue /
886           iNUM_TURBINES;
887           pabCurtailmentFlag[iCntTurbines] = 1;
888           //parReqChangeInPower[iCntTurbines] = turbinecurtreq_new[
889           iCntTurbines];
890           pabPriorityFlag[iCntTurbines] = 0;
891           turbinecurtreq_old[iCntTurbines] = turbinecurtreq_new[iCntTurbines
892           ];
893           testsum += turbinecurtreq_new[iCntTurbines];
894         }
895       }
896     }
897   }
898 }
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
889     }
890     else{
891         for (iCntTurbines = 0; iCntTurbines < iNUM_TURBINES; iCntTurbines++)
892         {           //cycle through farm
893             //if (abRecoveryFlag[iCntTurbines] == 0){
894             //turbinecurtreq_new[iCntTurbines] = delP_new *fi[iCntTurbines] /
895             FLAGSUM;
896             parReqChangeInPower[iCntTurbines] = turbinecurtreq_new[
897             iCntTurbines];
898             pabPriorityFlag[iCntTurbines] = 0;
899             pabFastSlowFlag[iCntTurbines] = 0;
900             turbinecurtreq_old[iCntTurbines] = turbinecurtreq_new[iCntTurbines
901             ];
902             testsum += turbinecurtreq_new[iCntTurbines];
903         }
904     }
905
906     delP_old = testsum;           //set delta P old for next
907     iteration
908     //delP_old = delP_new;
909     Pdem_old = Pdem_new;           //set demanded P for next
910     iteration
911     for (iCntTurbines = 0; iCntTurbines < iNUM_TURBINES; iCntTurbines++) {
912         reccheck[iCntTurbines] = abRecoveryFlag[iCntTurbines];
913     }
914 }
915 inertia_rec_flag_m1 = *prInertiaSwitch;
916 }
917
918
919 // End of main function
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
920
921
922
923 // HELPER FUNCTIONS:
924
925 int getTurbineFlags(int iReqTurbine, int abFlags[], int iNUM_TURBINES, int *
    abOutputFlags)
926 {
927     int iCntFlags;
928     if (iReqTurbine <= iNUM_TURBINES)
929     {
930         for (iCntFlags = 0; iCntFlags < iNUM_PAC_FLAGS; iCntFlags++)
931             abOutputFlags[iCntFlags] = abFlags[iCntFlags + (iReqTurbine - 1)*
    iNUM_PAC_FLAGS];
932         return 1;
933     }
934     else
935     {
936         for (iCntFlags = 0; iCntFlags < iNUM_PAC_FLAGS; iCntFlags++)
937             abOutputFlags[iCntFlags] = -1;
938         return 1;
939         return 0;
940     }
941
942 }
943
944 int getFlag(char *achRequestedFlag, int abFlags[], int iNUM_TURBINES,
945 int *abOutputFlag)
946 {
947     /*It is probably easiest to split the flags out into separate arrays as
    required.
948     This can be done in a separate function*/
949     /*
950     Inputs: Name of flag wanted (use the standard names),
951     the big array of flags from the Simulink simulation, and the number of
    turbines
952     Output: an array of just that flag (turbine 1 to N)
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
953
954 */
955 char achPAC_ON_FLAG[13] = "bPacOnFlag";
956 char achACTUATOR_LIM_SUB_FLAG[20] = "bSubFlagActuatorLim";
957 char achSUB_FLAG_WIND_SPEED[18] = "bSubFlagWindSpeed";
958 char achSUB_FLAG_RECOVERY[17] = "bSubFlagRecovery";
959 char achREJECTION_FLAG[15] = "bRejectionFlag";
960 char achSUB_FLAG_BLACK_LIMIT[20] = "bSubFlagBlackLimit";
961 char achSUB_FLAG_POWER[14] = "bSubFlagPower";
962 char achSUB_FLAG_POWER_RATE[18] = "bSubFlagPowerRate";
963 char achRECOVERY_FLAG[14] = "bRecoveryFlag";
964 char achSUB_FLAG_FAST_SLOW[17] = "bSubFlagFastSlow";
965 char achSUB_FLAG_REC_COMPLETE[25] = "bSubFlagRecoveryComplete";
966 char achRED_FLAG[9] = "bRedFlag";
967 char achAMBER_FLAG[11] = "bAmberFlag";
968 char achGREEN_FLAG[11] = "bGreenFlag";
969 char achPRIORITY_FLAG[14] = "bPriorityFlag";
970 char achDIVERGENT_FLAG[15] = "bDivergentFlag";
971 char achTURBULENCE_FLAG[16] = "bTurbulenceFlag";
972 char achCURTAILMENT_FLAG[17] = "bCurtailmentFlag";
973 char achCURTAILMENT_PROFILE_FLAG[24] = "bCurtailmentProfileFlag";
974 char achADJUST_RATED_SPD_FLAG[22] = "bAdjustRatedSpeedFlag";
975 int iCntTurbines;
976 int iFlag = 99;
977
978 if (!strcmp(achRequestedFlag, achPAC_ON_FLAG))
979 {
980     iFlag = 0;
981 }
982 if (!strcmp(achRequestedFlag, achACTUATOR_LIM_SUB_FLAG))
983 {
984     iFlag = 1;
985 }
986 if (!strcmp(achRequestedFlag, achSUB_FLAG_WIND_SPEED))
987 {
988     iFlag = 2;
989 }
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
990  if (!strcmp(achRequestedFlag, achSUB_FLAG_RECOVERY))
991  {
992      iFlag = 3;
993  }
994  if (!strcmp(achRequestedFlag, achREJECTION_FLAG))
995  {
996      iFlag = 4;
997  }
998  if (!strcmp(achRequestedFlag, achSUB_FLAG_BLACK_LIMIT))
999  {
1000     iFlag = 5;
1001 }
1002 if (!strcmp(achRequestedFlag, achSUB_FLAG_POWER))
1003 {
1004     iFlag = 6;
1005 }
1006 if (!strcmp(achRequestedFlag, achSUB_FLAG_POWER_RATE))
1007 {
1008     iFlag = 7;
1009 }
1010 if (!strcmp(achRequestedFlag, achRECOVERY_FLAG))
1011 {
1012     iFlag = 8;
1013 }
1014 if (!strcmp(achRequestedFlag, achSUB_FLAG_FAST_SLOW))
1015 {
1016     iFlag = 9;
1017 }
1018 if (!strcmp(achRequestedFlag, achSUB_FLAG_REC_COMPLETE))
1019 {
1020     iFlag = 10;
1021 }
1022 if (!strcmp(achRequestedFlag, achRED_FLAG))
1023 {
1024     iFlag = 11;
1025 }
1026 if (!strcmp(achRequestedFlag, achAMBER_FLAG))
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
1027 {
1028     iFlag = 12;
1029 }
1030 if (!strcmp(achRequestedFlag, achGREEN_FLAG))
1031 {
1032     iFlag = 13;
1033 }
1034 if (!strcmp(achRequestedFlag, achPRIORITY_FLAG))
1035 {
1036     iFlag = 14;
1037 }
1038 if (!strcmp(achRequestedFlag, achDIVERGENT_FLAG))
1039 {
1040     iFlag = 15;
1041 }
1042 if (!strcmp(achRequestedFlag, achTURBULENCE_FLAG))
1043 {
1044     iFlag = 16;
1045 }
1046 if (!strcmp(achRequestedFlag, achCURTAILMENT_FLAG))
1047 {
1048     iFlag = 17;
1049 }
1050 if (!strcmp(achRequestedFlag, achCURTAILMENT_PROFILE_FLAG))
1051 {
1052     iFlag = 18;
1053 }
1054 if (!strcmp(achRequestedFlag, achADJUST_RATED_SPD_FLAG))
1055 {
1056     iFlag = 19;
1057 }
1058
1059
1060 for (iCntTurbines = 0; iCntTurbines<iNUM_TURBINES; iCntTurbines++)
1061 {
1062     abOutputFlag[iCntTurbines] = abFlags[iFlag*iNUM_TURBINES + iCntTurbines
];
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
1063 }
1064 // If iFlag is still 99 then no match was found
1065 if (iFlag == 99)
1066     return 0;
1067 else
1068     return 1;
1069 }
1070
1071 int getValue(char *achRequestedValue, float arValues[], int iNUM_TURBINES,
1072 float *arOutputValue)
1073 {
1074     /*It is probably easiest to split the PAC values out into separate arrays
1075     as required.
1076     This can be done in a separate function*/
1077     /*
1078     Inputs: Name of value wanted (use the standard names),
1079     the array of PAC values from the Simulink simulation, and the number of
1080     turbines
1081     Output: an array of just that value (turbine 1 to N)
1082     */
1083     char achCURTAILMENT_VALUE[18] = "rCurtailmentValue";
1084     char achNEW_RATED_GEN_SPD[16] = "rNewRatedGenSpd";
1085     char achUPSTREAM_WIND_SPD[17] = "rUpStreamWindSpd";
1086     char achDIFF_POWER[11] = "rDiffPower";
1087     char achPOW_EST_NO_PAC[13] = "rPowEstNoPac";
1088     char achAREO_TORQUE_EST[22] = "rTotalAeroTrqEstimate";
1089     int iCntTurbines;
1090     int iFlag = 99;
1091
1092     if (!strcmp(achRequestedValue, achUPSTREAM_WIND_SPD))
1093     {
1094         iFlag = 0;
1095     }
1096     if (!strcmp(achRequestedValue, achAREO_TORQUE_EST))
1097     {
```

Appendix E. Wind Farm Distribution Controller C++ Code

```
1098     iFlag = 1;
1099 }
1100
1101 if (!strcmp(achRequestedValue, achCURTAILMENT_VALUE))
1102 {
1103     iFlag = 2;
1104 }
1105 if (!strcmp(achRequestedValue, achNEW_RATED_GEN_SPD))
1106 {
1107     iFlag = 3;
1108 }
1109
1110 if (!strcmp(achRequestedValue, achDIFF_POWER))
1111 {
1112     iFlag = 4;
1113 }
1114 if (!strcmp(achRequestedValue, achPOW_EST_NO_PAC))
1115 {
1116     iFlag = 5;
1117 }
1118
1119 for (iCntTurbines = 0; iCntTurbines<iNUM_TURBINES; iCntTurbines++)
1120 {
1121     arOutputValue[iCntTurbines] = arValues[iFlag*iNUM_TURBINES +
1122     iCntTurbines];
1123 }
1124 // If iFlag is still 99 then no match was found
1125 if (iFlag == 99)
1126     return 0;
1127 else
1128     return 1;
1129 }
```

Appendix F

Wind Farm Network Controller C++ Code

This Appendix contains the C++ code used in the Wind Farm Network Controller S-function in Strathfarm. It should be noted that this code will not work without the linking function which cannot be included in this thesis due to intellectual property restrictions.

```
1
2
3 // Common libraries
4 #include <windows.h>
5 #include <stdio.h>
6 #include <string.h>
7 #include <math.h>
8 #include <conio.h>
9 #include <iostream>
10 using namespace std;
11 #include <stdlib.h>
12
13 #include <iomanip>
14 #include <fstream>
15 #include <cmath>
16
17 #define INUM_PAC_FLAGS 20
18 #define INUM_PAC_VALUES 5
```

Appendix F. Wind Farm Network Controller C++ Code

```
19
20
21 //***** Required at the beginning (from Bladed Manual)
22 extern "C" //avoid mangled names
23 { void __declspec(dllexport) __cdecl windFarmControl(float *parPowerToFarm,
24 int *pbCurtailmentFlagRequest, float *prCurtailmentValue,
25 float *prRocof, float *prFrequency, float *prSimTime, int iNUM_TURBINES,
    float *parReqSetPoint,
26 int *pabCurtailmentSwitch, int *pabInertiaSwitch);
27 }
28 //int getFlag(char *achRequestedFlag, int abFlags[], int iNUM_TURBINES,
29 // int *abOutputFlag);
30 //int getValue(char *achRequestedValue, float arValues[], int iNUM_TURBINES,
31 // float *arOutputValue);
32 //int getValue2(char *achRequestedValue, int arValues[], int iNUM_TURBINES,
33 // float *arOutputValue);
34 //int getTurbineFlags(int iReqTurbine, int abFlags[], int iNUM_TURBINES,
35 // int *abOutputFlags);
36
37 //***** Main DLL routine *****//
38 void __declspec(dllexport) __cdecl windFarmControl(float *parPowerToFarm,
39 int *pbCurtailmentFlagRequest, float *prCurtailmentValue,
40 float *prRocof, float *prFrequency, float *prSimTime, int iNUM_TURBINES,
    float *parReqSetPoint,
41 int *pabCurtailmentSwitch, int *pabInertiaSwitch)
42 {
43
44 /* The inputs are:
45 parPowerToFarm - The power output of the wind farm - each element of the
    array is one turbine's
46 power output [W]
47 pabFlagsToFarm - The 20 flags of each turbine, elements 0-19 are turbine
    1, 20-39 are turbine 2
48 etc. [Booleans]
49 parValuesToFarm - The 4 PAC values of eah turbine [Floats]
50 pbCurtailmentFlagRequest - A flag that denotes that curtailment is
    requested [Boolean]
```

Appendix F. Wind Farm Network Controller C++ Code

```
51 prCurtailmentValue - The value of the curtailment [W]
52 prRocof - "Rate of change of frequency". The rate of change of the grid
    frequency [Hz/s]
53 prFrequency - The grid frequency [Hz]
54 prSimTime - The simulation time [s]
55 piNUM_TURBINES - The number of turbines in the farm [Turbines]
56 */
57 /* The outputs are:
58 parReqChangeInPower - The requested change in power output for each
    turbine. Each element is
59 a turbine's requested change in power [W]
60 pabPacOnFlag - The flags to turn the PACs on. Each element represents a
    turbine's flag
61 [Boolean]
62 pabPriorityFlag - The flags to remove rate of change of power limits.
    Each element represents
63 a turbine's flag [Boolean]
64 pabFastSlowFlag - The flag to select the recovery speed (1 = fast, 0 =
    slow) Each element
65 represents a turbine's flag [Boolean]
66 pabCurtailmentFlag - The flag to use Curtailment mode of PAC. Each element
    represents a turbine's flag [Boolean]
67 parCurtailmentPower - The requested curtailment power value. Each element
    represents a turbine's flag [W]
68 pabCurtailmentProfileFlag - The flag for curtailment profile of PAC. Each
    element represents a turbine's flag [Boolean]
69 parNewRatedGenSpd - The requested new rated generator speed. Each element
    represents a turbine's flag [rad/s]
70 pabAdjustRatedSpeedFlag - The flag to use new rated generator speed. Each
    element represents a turbine's flag [Boolean]
71 */
72
73 static float Frequency_m1; //used for hold previous frequency value
74
75
76
```

Appendix F. Wind Farm Network Controller C++ Code

```
77 float K = iNUM_TURBINES*43754224;//from equations in SP thesis assuming 5
    mw wind turbine
78 static float m1;
79 static float m2;
80 float reserve_pct=0.1;
81 float controlband_lower = 49 ;
82 float controlband_upper = 51 ;
83 float deadband_lower = 49.985;
84 float deadband_upper = 50.015;
85 float ROCOF_upper = 0.1;
86 float ROCOF_lower = -0.1;
87
88
89 m1 = 3;// (1 + reserve_pct) / ((50 - controlband_lower) - (50 -
    deadband_lower));// steepness of droop slope based on SP thesis. based
    on delta reserve so may need changing for set-point tracking
90 m2 = 3;// (1 - reserve_pct) / ((50 + controlband_upper) - (50 +
    deadband_upper));// steepness of droop slope based on SP thesis. based
    on delta reserve so may need changing for set-point tracking
91
92 //-----
93 // controller design section
94
95
96 int Outputflags[1];
97 Outputflags[1] = 0; //A vector of size one is used here otherwise MATLAB
    crashes
98
99 if (*prSimTime > 100) {
100     if ((*prRocof * 50 > ROCOF_upper)||(*prRocof * 50 < ROCOF_lower))
101     {
102
103         Outputflags[1] = 1;
104
105         *pabCurtailmentSwitch = *pbCurtailmentFlagRequest;
106         *pabInertiaSwitch = Outputflags[1];//tells dist controller to switch
            to inertia distribution/pac flags
```

Appendix F. Wind Farm Network Controller C++ Code

```
107     // *parReqSetPoint = *prCurtailmentValue - *prRocof * 50 * K; // from SP
thesis.
108
109     }
110
111     if (*prFrequency * 50 < deadband_upper && *prFrequency * 50 >
deadband_lower) {
112         *parReqSetPoint = *prCurtailmentValue;
113
114     }
115     else {
116         if (*prFrequency * 50 >= deadband_upper) {
117             *parReqSetPoint = *prCurtailmentValue - ((*prFrequency * 50 - (
deadband_upper)) * m1 * 0.1 * iNUM_TURBINES * 5000000);
118         }
119         else {
120             *parReqSetPoint = *prCurtailmentValue - ((*prFrequency * 50 - (
deadband_lower)) * m2 * 0.1 * iNUM_TURBINES * 5000000);
121         }
122     }
123     // *parReqSetPoint = *prCurtailmentValue;
124     *pabCurtailmentSwitch = *pbCurtailmentFlagRequest;
125     *pabInertiaSwitch = Outputflags[1];
126
127     }
128 }
129
130 // End of main function
131 }
```