

THE UNIVERSITY *of* EDINBURGH

This thesis has been submitted in fulfilment of the requirements for a postgraduate degree (e. g. PhD, MPhil, DClinPsychol) at the University of Edinburgh. Please note the following terms and conditions of use:

- This work is protected by copyright and other intellectual property rights, which are retained by the thesis author, unless otherwise stated.
- A copy can be downloaded for personal non-commercial research or study, without prior permission or charge.
- This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the author.
- The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the author.
- When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given.

The Techno-Economic Benefits of LIDAR-Assisted Pitch Control of Floating Offshore Wind Turbines

Andrew James Russell

Supervisors: M. Collu, P.R. Thies, A.S. McDonald, A. Keane and A.R. Quayle















Doctor of Engineering

THE UNIVERSITY OF EDINBURGH

2024

IDCORE Programme

The Industrial Centre for Doctoral Training for Offshore Renewable Energy (IDCORE) is an Institute of Marine Engineering, Science and Technology (IMarEST) accredited doctoral training programme funded by the UK Research and Innovation's (UKRI) Engineering and Physical Sciences (EPSRC) and Natural Environment (NERC) research councils under grant number EP/S023933/1. IDCORE is a collaborative partnership between the Universities of Edinburgh, Strathclyde and Exeter, as well as the Scottish Association for Marine Science (SAMS). Students undertake offshore renewable energy-based doctoral research alongside sponsor companies, with the goal of addressing industry-focused research aims relevant to the sponsors. This thesis aims to meet the criteria for the award of an Engineering Doctorate (EngD) degree.

Abstract

Floating offshore wind (FOW) is an emerging technology capable of harnessing wind energy in waters too deep for bottom-fixed wind (BFW) turbines. However, this presents additional design, construction, installation and maintenance challenges, leading to higher costs of FOW than BFW. Therefore, strategies capable of reducing these costs are highly desirable.

One such strategy is LIDAR-assisted pitch control (LAC), which utilises nacelle-mounted LIDAR to measure the velocity of the incoming wind, allowing the turbine to use feedforward (FF) control to actuate its blade pitch systems in advance of the wind's impact. Individual blade pitch control (IPC) can also be employed to mitigate variations in structural loads.

This thesis presents a novel LIDAR-assisted FF IPC (FFIPC) approach, which was combined with an FF-feedback collective pitch controller (FFCPC). The source code of OpenFAST wind turbine modelling software was modified to enable LIDAR simulation and LAC. Simulations of a 15 MW wind turbine mounted on two different floating platforms, a semi-submersible and a spar, were performed. For both FOW turbines (FOWTs), the FF controllers delivered standard deviation reductions of up to 75% in the rotor speed and power, 58% in the platform pitch and surge and 20% in the tower base fore-aft bending moment compared to the baseline feedback (FB)-only controller. The semi-submersible benefitted more from the combination of FFCPC and FFIPC (FFCPC+FFIPC), with up to 12% further reductions to the standard deviations compared to the FFCPC on its own. The spar did not benefit as significantly from the addition of FFIPC, with FB IPC (FBIPC) instead better complimenting the FFCPC.

The implementation of FFCPC+FFIPC within the 15 MW semi-submersible was applied to three case studies aiming to quantify the cost benefits of LAC for FOW. Firstly, the loading reductions were converted to reductions of lifetime component failure rates before being applied within an operations and maintenance (O&M) model. The decreased failure rates brought by LAC delivered reductions to the operational expenditure of FOW farms, by up to 11%. The second study applied modifications to the FOWT's tower. The results indicated the ability to reduce the tower thickness by up to 20% when using LAC while achieving similar levels of stress to the baseline tower design using FB-only control. Finally, simulations were extended to the array level using FAST.Farm. The addition of LAC was found to deliver improved wake recovery of the FOWTs. This allowed for reduced mean power deficits of downstream turbines at low above-rated wind speeds, which translated to an increase in their annual energy production by up to 0.8%, relative to when they used FB-only control.

In summary, this thesis provides strong evidence for the large-scale adoption of LAC within FOWTs, presenting multiple avenues for reducing the levelised cost of FOW energy, which can ultimately increase the commercial viability of FOW projects.

Lay Summary

Floating offshore wind (FOW) is a developing technology that can capture wind energy in deeper waters where traditional bottom-fixed wind turbines (BFWTs) cannot be installed. However, designing, building, installing and maintaining floating offshore wind turbines (FOWTs) is more challenging and expensive than for BFWTs. Therefore, it is important to find ways to reduce these costs.

One way is to use an emerging technology called LIDAR-assisted pitch control (LAC). This uses a device placed on top of the turbine, which is used to measure the speed of the incoming wind. This allows the turbine to see what is coming, which allows it to change its control settings in preparation for the wind's impact. Another technique called individual blade pitch control (IPC) can be used to balance the forces on the turbine's blades.

This work combines LAC with IPC and studies its effects on different FOWTs. A popular wind turbine simulation tool was enhanced so that it could simulate measurements of the incoming wind speed, which then allowed for LAC to be used within simulations. Simulations of a large wind turbine mounted on two different floating platforms were performed. For both FOWTs, the results showed that the combination of LAC and IPC could decrease the changes in the speed of the rotor's rotation, the platform's movements and the structural loadings compared to the traditional control methods.

The results were carried forward and used within three studies in this work. These intended to work out how LAC can reduce the high costs of FOW. The first study found that the reduced variations of the structural loads could extend the lifetimes of parts of the wind turbines, meaning that they do not need to be repaired as often. This meant that the costs of maintaining FOW farms across their lifetimes could be reduced, by up to 11%. The second study found that using LAC could allow for lighter tower designs without sacrificing structural integrity, potentially reducing material costs by up to 20%. Finally, simulations of multiple FOWTs within a wind farm were performed. This study found that LAC improved how the wind recovered after it passed through the turbines. This led to increased energy production from the turbines positioned behind the turbine at the front, compared to when the traditional control methods were used. This means that more electricity can be generated overall, making FOW farms more productive and cost-effective in the long run.

In summary, the findings of this work provide strong evidence for the large-scale use of LAC on FOWTs, showing multiple ways that it can reduce the lifetime cost of FOW energy and, therefore, make FOW projects more profitable.

Acknowledgements

There are numerous individuals that I would like to thank for guiding, inspiring and supporting me on this journey. Firstly, I would like to thank my esteemed supervisory team, consisting of Prof. Maurizio Collu (University of Strathclyde), Prof. Philipp Thies (University of Exeter), Prof. Alasdair McDonald (University of Edinburgh), Dr. Aidan Keane (Wood Renewables) and Dr. Alexander Quayle (Flotation Energy Ltd.), who provided expert supervision, guidance and inspiration throughout the research project.

I am deeply grateful to Dr. Jade McMorland for providing valuable contributions to the research by undertaking the O&M modelling simulations, sharing her knowledge and providing some of the results reported in Chapter 6 of this thesis. I would also like to thank Andy Platt and Derek Slaughter at NREL for facilitating the integration of the source code modifications that I made within OpenFAST. My thanks also go to Prof. David Schlipf and Dr. Feng Guo for sharing their knowledge of LIDAR-assisted control modelling.

I would also like to thank my colleagues at Wood Renewables and Flotation Energy Ltd. for their support and for the incredible opportunities and experiences that I have been privy to while working with them. My thanks also extend to the IDCORE management team, namely Prof. David Ingram, Dr. Katrina Tait and Andrew Aveyard for their perpetual support and organisation throughout my time at IDCORE.

Finally, my heartfelt thanks go to my caring family and my loving partner, Anusha. Their unwavering support and encouragement kept me motivated to succeed and to do them proud. They were always right beside me throughout all of the highs and lows of this journey and for that, I am, and always will be, immeasurably grateful.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

Andrew James Russell

Contents

IDCORE Programme ii

Abstract iii

Lay Summary iv

Acknowledgements v

Declaration vi

Figures and Tables xii

1 Introduction 1

1.1 Research context . . . 1

1.1.1 The growth of the offshore wind energy industry . . . 1

1.1.2 The case for floating offshore wind . . . 2

1.1.3 Floating offshore wind challenges . . . 3

1.1.4 The LIDAR-assisted control solution . . . 4

1.2 Project context and industrial applicability . . . 4

1.3 Fundamental concepts . . . 5

1.3.1 Wind modelling . . . 6

1.3.2 LIDAR . . . 7

1.3.3 Floating offshore wind turbine platform designs . . . 10

1.3.4 Turbine operating regions . . . 12

1.3.5 Blade pitch control . . . 12

1.3.6 Feedforward pitch control with LIDAR . . . 19

1.3.7 O&M of floating offshore wind . . . 21

1.3.8 Floating offshore wind turbine tower design . . . 23

1.3.9 Wind farms and wake effects . . . 25

1.4 Aim and objectives . . . 26

1.4.1 Feedforward individual pitch control of large floating offshore wind turbines . . . 27

1.4.2 O&M case study . . . 28

1.4.3 Tower design modifications . . . 28

1.4.4 Array-level benefits of LIDAR-assisted control . . . 29

1.4.5 Implementation of LIDAR simulation and LIDAR-assisted control within OpenFAST . . . 29

1.5 Simulation tools . . . 30
1.6 Thesis flowchart . . . 31
1.7 Contributions to knowledge . . . 32
1.7.1 Publications . . . 32
1.8 Thesis structure . . . 33

2 Literature review **34**
2.1 LIDAR measurement coherence . . . 34
2.2 Review of LIDAR-assisted feedforward collective pitch control . . . 35
2.2.1 Bottom-fixed wind turbines . . . 35
2.2.2 Floating offshore wind turbines . . . 39
2.3 Review of feedback and feedforward individual pitch control . . . 42
2.4 O&M . . . 47
2.5 Floating offshore wind turbine tower design . . . 49
2.6 Array-level simulation . . . 49

3 OpenFAST, ROSCO and WEIS source code modifications **51**
3.1 InflowWind source code architecture . . . 51
3.2 Implementation of LIDAR simulation within InflowWind . . . 53
3.2.1 Specification of LIDAR inputs within the InflowWind input file . . . 53
3.2.2 Parsing of LIDAR inputs . . . 54
3.2.3 Interface of parameters to LIDAR sub-module . . . 54
3.2.4 Enhancement of LIDAR calculations . . . 54
3.2.5 Wind speed determination . . . 56
3.2.6 Export of LIDAR outputs . . . 58
3.3 LIDAR simulation capabilities . . . 58
3.4 ROSCO source code architecture . . . 60
3.5 Integration of LIDAR-assisted control within ROSCO . . . 61
3.5.1 Parsing of LIDAR parameters from the avrSWAP array . . . 61
3.5.2 Specification of feedforward control inputs . . . 61
3.5.3 Calculation of LIDAR beam azimuth angles . . . 61
3.5.4 Development of feedforward collective and individual pitch control sub-routines . . . 63
3.6 WEIS source code modifications . . . 63

4 Pitch control theory **65**
4.1 Feedback collective pitch controller . . . 65
4.2 Feedforward collective pitch controller . . . 66
4.3 Feedback individual pitch controller . . . 69
4.4 Feedforward individual pitch controller . . . 70

4.5 Filter design and time delays . . . 74
4.5.1 LIDAR rotor-effective wind speed low-pass filter . . . 75
4.5.2 Feedforward individual pitch controller low-pass filter . . . 78

5 OpenFAST floating offshore wind turbine simulations 80
5.1 Simulation set-up . . . 80
5.2 Initial floating offshore wind turbine comparisons . . . 81
5.2.1 Feedback-only control . . . 81
5.2.2 Feedforward control restriction requirements . . . 84
5.3 Key statistical metrics . . . 86
5.4 VolturnUS-S simulation results . . . 88
5.4.1 Single wind field . . . 89
5.4.2 Above-rated wind spectrum . . . 94
5.4.3 Comparison of beam span locations . . . 96
5.5 WindCrete simulation results . . . 97
5.5.1 Single wind field . . . 97
5.5.2 Above-rated wind spectrum . . . 102
5.6 Floating offshore wind turbine feedforward control results comparison . . . 106
5.6.1 Single wind field . . . 106
5.6.2 Above-rated wind spectrum . . . 108
5.7 Estimation of feedforward individual pitch controller beam assignment errors . 109

6 O&M case study 113
6.1 Statement of collaboration . . . 113
6.2 ScotWind case study . . . 114
6.3 O&M methodology . . . 116
6.3.1 Maintenance strategy . . . 117
6.3.2 Failure rate determination . . . 118
6.3.3 Assessed parameters . . . 120
6.4 Results . . . 122
6.4.1 LIDAR-assisted control failure rate reductions . . . 122
6.4.2 Baseline comparison . . . 123
6.4.3 Workability impact . . . 125

7 Tower design case study 128
7.1 Methodologies . . . 130
7.1.1 Design variables . . . 130
7.2 Simulation set-up . . . 131
7.2.1 Stress analysis . . . 133
7.3 Feedback-only control results . . . 134

7.3.1 Outer diameter . . . 134
7.3.2 Shell thickness . . . 136
7.3.3 Analysis of other parameters . . . 138
7.3.4 Equivalent design stresses . . . 140
7.4 LIDAR-assisted control results . . . 141
7.5 Further analysis - Extreme conditions . . . 143
7.5.1 Responses to extreme conditions . . . 144
7.5.2 Strength analysis . . . 145
7.5.3 Stress envelope . . . 147
7.6 Further analysis - Wind-wave misalignment . . . 148
7.6.1 Loading and stress changes with increasing wind speeds . . . 148
7.6.2 Loads and stress on the lightweight tower . . . 150
7.7 Further analysis - Joint probability density function . . . 153
7.7.1 Probability density functions . . . 153
7.7.2 Results . . . 156

8 Array-level benefits of LIDAR-assisted pitch control 160
8.1 Simulation set-up . . . 160
8.2 Methodologies . . . 161
8.3 Results . . . 162
8.3.1 Feedback-only control . . . 163
8.3.2 LIDAR-assisted control . . . 165

9 Discussion, conclusions and opportunities for future work 173
9.1 Discussion . . . 173
9.1.1 LIDAR simulator, control strategies and floating offshore wind turbine platform configuration results . . . 173
9.1.2 O&M case study . . . 176
9.1.3 Tower design case study . . . 178
9.1.4 Array-level case study . . . 181
9.2 Conclusions . . . 183
9.2.1 OpenFAST simulation results . . . 184
9.2.2 O&M case study . . . 185
9.2.3 Tower design case study . . . 186
9.2.4 Array-level case study . . . 189
9.3 Opportunities for future work . . . 190
9.3.1 LIDAR simulator and feedforward individual pitch controller enhancements . . . 190
9.3.2 Fatigue estimations . . . 192
9.3.3 Structural design optimisation . . . 193

9.3.4 Array-level simulation 194
9.3.5 Field testing of LIDAR-assisted control 195

Bibliography **196**

Appendices

A Simulation tools **211**
A.1 OpenFAST 211
A.1.1 Key modules 211
A.1.2 ROSCO 214
A.2 Turbine and substructure models 214
A.2.1 Turbine 214
A.2.2 Floating platforms 215
A.3 StrathO&M 216
A.4 WEIS 217
A.5 FAST.Farm 217

B Relevant OpenFAST, ROSCO and WEIS source code **219**
B.1 OpenFAST 219
B.2 ROSCO 236
B.3 WEIS 245

C Dataset repository access **247**

D Simulation input files **248**
D.1 OpenFAST 248
D.2 MLife 257
D.3 FAST.Farm 260

E Feedback-only standard deviation changes with increasing wind speeds **262**

F FAST.Farm time series data alignment **263**

G Journal publications **264**

Figures and Tables

Figures

1.1 Nacelle-mounted single-beam LIDAR configurations. 7
1.2 Weighting function of a CW LIDAR beam, adapted from Simley and Scholbrock (2020). 8
1.3 Floating offshore wind platform configurations, from ACTEON (2020). 11
1.4 Power curve and operation regions of the IEA 15 MW reference turbine, adapted from Gaertner et al. (2020). 13
1.5 Forces acting on a blade airfoil. 14
1.6 Illustrating the general trend of increasing angle of attack on C_L and C_D. 15
1.7 Flowchart illustrating an example LIDAR-based FF pitch controller. 19
1.8 Frequencies pertinent to the tower design of the VolturnUS-S, from Allen et al. (2020). 24
1.9 An energy extracting actuator disk and stream-tube, adapted from Burton et al. (2011). 25
1.10 Thesis flowchart. 31

2.1 Findings from Schlipf et al. (2015), simulating FFCPC assuming perfect LIDAR wind preview of a wind speed step change. 40
2.2 Findings from Schlipf et al. (2015), simulating FFCPC with realistic LIDAR wind preview of a turbulent wind field. 41
2.3 Response of the FIR FF controller designed by Dunne, Pao, et al. (2011). 45
2.4 Tower top fore-aft pitching moment and blade root bending moment as a function of frequency, from Dunne et al. (2012). 46

3.1 InflowWind module source code architecture. 52
3.2 Time series of LIDAR-measured hub height wind speeds obtained from simulations with the modified OpenFAST code. 59
3.3 ROSCO source code architecture. 60

4.1 Illustration of a side view of the nacelle-mounted LIDAR beam trajectories. 66
4.2 Wind-speed-to-blade-pitch-angle curve as used in the modified ROSCO input file. 68
4.3 Azimuth positions of the LIDAR beam focal positions within the rotor plane. . . . 71
4.4 Illustrating an example scenario of the rotation of the turbine blades during the preview time provided for the FF individual pitch commands. 72

4.5 Excerpt of time series individual blade pitch commands pre- and post-low-pass filtering. 73
4.6 Excerpt of time series individual blade pitch variation when using FFIPC. 74
4.7 Flowchart illustrating the FFCPC and FFIPC control loops added to the FB pitch and nacelle FB loops. 75
4.8 Ten-minute time series of LIDAR-measured and turbine-experienced REWSs used in the coherence analysis. 76
4.9 Transfer function between the LIDAR-measured and turbine-experienced REWS simulation data as a function of wave number for the two FOWTs. 76
4.10 f_{cutoff} at different mean wind speeds at 50% radial span for the two FOWTs. . . 77
4.11 f_{cutoff} at different mean wind speeds for each of the radial span measurement locations of the VolturnUS-S. 78
4.12 T_{buffer} at different reference wind speeds applied to the VolturnUS-S. 79

5.1 Ten-minute excerpt of time series results comparing the FB-only pitch controller performance of the two FOWTs. 82
5.2 PSD plots generated from one-hour simulations of the two FOWTs using FB-only control under turbulent wind and irregular wave conditions. 84
5.3 Time series plots showing the impact of applying or not applying limits to the FFCPC commands in measured wind speeds below 13 ms^{-1} for both FOWTs. . . 85
5.4 Excerpt of time series results comparing the FB-only pitch controller to the FF controllers of the VolturnUS-S. 89
5.5 Spider-web plots showing the normalised standard deviations of various parameters of the VolturnUS-S when using the different FB and FF control strategies under turbulent wind and irregular wave conditions. 90
5.6 PSD plots of the turbine and substructure parameters generated from one-hour simulations of the VolturnUS-S under turbulent wind and irregular wave conditions. 93
5.7 Normalised SD or RMSE values delivered by the FF controllers compared to the FB-only controller of the VolturnUS-S across above-rated wind speeds. 94
5.8 Normalised SD or RMSE values delivered by the FFCPC+FFIPC controller with beams positioned at 30%, 50% and 70% rotor radial span compared to the FB-only controller of the VolturnUS-S across above-rated wind speeds. 96
5.9 Excerpt of time series results comparing the FB-only controller to the FF controllers on the WindCrete under turbulent wind and irregular wave conditions. 98
5.10 Spider-web plots showing the normalised standard deviations of various parameters of the WindCrete using the different FF control strategies under turbulent wind and irregular wave conditions. 99

5.11 PSD plots of the tower base fore-aft bending moment of the VolturnUS-S and WindCrete using FFCPC and FFCPC+FFIPC under turbulent wind and irregular wave conditions. 100
5.12 PSD plots of the turbine and substructure parameters of the WindCrete generated from one-hour turbulent wind and irregular wave simulations. 101
5.13 Normalised SD or RMSE values delivered by the FF controllers compared to the FB-only pitch controller of the WindCrete across above-rated wind speeds. . . . 104
5.14 PSD plots of the tower base fore-aft bending moment of the WindCrete using FB-only control, FFCPC and FFCPC+FFIPC under turbulent wind fields with different mean wind speeds. 105
5.15 Spider-web plots showing the normalised standard deviations of various parameters of the VolturnUS-S and WindCrete using the selected FF control strategies under turbulent wind and irregular wave conditions. 107
5.16 Normalised SD or RMSE values delivered by the FF controllers of the VolturnUS-S and WindCrete compared to their respective FB-only controllers across above-rated wind speeds. 108
5.17 Rotor azimuth quadrants associated with the outer LIDAR beams. 111
5.18 Estimated percentage misassignment of the LIDAR beams to the rotor blades when issuing the FFIPC commands. 112

6.1 Map, adapted from Clive et al. (2022), showing the locations of the ScotWind leasing round sites studied within this case study. 115
6.2 Limitations imposed on the accessibility of Design A for specific H_s-T_p combinations. Adapted from Scheu et al. (2018) 116
6.3 Relationship between average normalised failure rate reduction of the RNA and OpEx reductions. 125
6.4 OpEx savings resulting from reduced failure rates for FOW sites with imposed WI limitations. 126

7.1 Rotor 1P and 3P design and constraint frequency ranges of the VolturnUS-S when using FB-only control or LAC. 129
7.2 Impacts on the standard deviations of the tower base loads with increasing reductions in the tower outer diameter. 135
7.3 Vertical lines imposed on the rotor 1P and 3P frequency ranges of the VolturnUS-S, indicating the calculated tower first fore-aft natural frequencies of the tower designs with reduced outer diameter. 135
7.4 Time series of the tower base side-to-side bending moment with reduced tower outer diameter. 136

7.5 PSD plots of the tower base side-to-side and fore-aft bending moments of the modified tower outer diameter designs at different mean wind speeds within DLC 1.2. 137
7.6 Impacts on tower loads and DEL with increasing reductions in the tower shell thickness. 138
7.7 Vertical lines imposed on the rotor 1P and 3P frequency ranges of the VolturnUS-S when using FB-only control or LAC, indicating the calculated tower first fore-aft natural frequencies of the tower designs with reduced shell thickness. 138
7.8 PSD plots of the tower base side-to-side and fore-aft bending moments of the modified tower shell thickness designs at different mean wind speeds within DLC 1.2. 139
7.9 PSD plots of three parameters when using the modified tower designs with reduced outer diameter or shell thickness. 139
7.10 Normalised average standard deviations of equivalent design stress on the tower base of the modified tower designs compared to the baseline tower design. . . . 140
7.11 Comparing the tower base equivalent design stresses applied to towers with different percentage reductions in the outer diameter and shell thickness when supporting a LIDAR-assisted-controlled FOWT under DLC 1.2. 142
7.12 PSD plots of five parameters of the lightweight tower supporting a LIDAR-assisted-controlled FOWT, compared to the baseline tower supporting an FB-only-controlled FOWT. 143
7.13 Strength analysis of an FB-only-controlled FOWT supported by the baseline or lightweight tower under DLC 1.6. 146
7.14 Strength analysis of the baseline and lightweight tower designs supporting a LIDAR-assisted-controlled an FB-only controlled FOWT under DLC 1.6. 146
7.15 Stress envelope of the baseline and lightweight tower designs supporting an FB-only controlled or LIDAR-assisted controlled FOWT under DLC 1.6. 147
7.16 Spider-web plots of the normalised standard deviations of the tower base fore-aft and side-to-side bending moments and the equivalent design stress acting on the base of the baseline tower supporting an FB-only-controlled FOWT at different mean wind speeds. 149
7.17 Spider-web plots of the standard deviations of the tower base fore-aft and side-to-side bending moments of the baseline tower supporting an FB-only controlled FOWT. 150
7.18 Spider-web plots of the normalised standard deviations of the tower base fore-aft and side-to-side bending moments and the equivalent design stress acting on the base of the lightweight tower supporting a LIDAR-assisted-controlled FOWT and the baseline tower supporting an FB-only controlled FOWT. 151

7.19 Wind speed and wind-wave misalignment probability and probability density functions. 154
7.20 JPD function of mean wind speed and wind-wave misalignment angle. 155
7.21 Total JPD-weighted equivalent design stress at each wind speed and wind-wave misalignment angle. 156
7.22 Total JPD-weighted equivalent design stress of the baseline tower supporting an FB-only controlled FOWT and the lightweight tower supporting an FB-only controlled FOW at each wind speed and wind-wave misalignment angle and when using different normal distribution function standard deviations. 158

8.1 Layout of the three FOWTs, separated by multiples of the turbine's rotor diameter, as simulated within FAST.Farm. 161
8.2 Flowchart illustrating the steps undertaken to perform the array-level LAC study. . 162
8.3 Mean values of turbine parameters, normalised to that of T1, when using FB-only control across above-rated wind speeds and with a turbine spacing of 8 RD. . . . 164
8.4 Mean wake deficit at 0 m radial distance and 1000 m downstream distance of T1 across above-rated mean ambient wind speeds 165
8.5 Mean values of T2 parameters, normalised to that of T1, when using FB-only control or LAC across above-rated wind speeds and with a turbine spacing of 8 RD.165
8.6 Standard deviations of properties associated with the three turbines when using LAC, normalised to that when using FB-only control for each turbine at different mean wind speeds. 166
8.7 Standard deviations of parameters of T2 when using FB-only control or LAC, normalised to those of T1 FB under an ambient wind speed of 15 ms^{-1}. 167
8.8 Time series excerpt of the wake deficit recorded at 0 m radial position and 800 m downstream of T1 under turbulent wind and irregular wave conditions. 168
8.9 Range of the wake deficit recorded at various radial and downstream distances of T1 under turbulent wind field and irregular wave conditions. 169
8.10 Maximum wake deficit recorded at various radial and downstream distances of T1 under turbulent wind and irregular wave conditions. 170
8.11 Standard deviation of the wake deficit, recorded at various radial and downstream distances of T1 under turbulent wind and irregular wave conditions. 171
8.12 Mean power of T2 and T3 when using FB-only control or LAC, normalised to the mean power of T1 using FB-only control under above-rated wind speeds and with a turbine spacing of 8 RD. 171

A.1 Flowchart showing the OpenFAST modules and the interactions between them, from J. Jonkman (2022). 212

B.1 LIDAR.txt file (part 1) following modifications made to enable LIDAR simulation within the InflowWind module of OpenFAST. 219
B.2 LIDAR.txt file (part 2) following modifications made to enable LIDAR simulation within the InflowWind module of OpenFAST. 220
B.3 InflowWind.txt file (part 1) following the modifications made. 220
B.4 InflowWind.txt file (part 2) following the modifications made. 221
B.5 Code added within the InflowWind_Subs.f90 script to allow for the parsing of LIDAR configuration inputs from the InflowWind input file. 222
B.6 Code added within the InflowWind_Subs.f90 script to allow for the allocation of LIDAR measurement outputs. 223
B.7 Modified InflowWind input file containing a new section for the specification of LIDAR simulator parameters. 223
B.8 Code added within the InflowWind.f90 script to allow for the saving of LIDAR inputs as parameters as well as to initiate and calculate LIDAR outputs. 224
B.9 Complete modified LIDAR.f90 script (part 1). 225
B.10 Complete modified LIDAR.f90 script (part 2). 226
B.11 Complete modified LIDAR.f90 script (part 3). 227
B.12 Complete modified LIDAR.f90 script (part 4). 228
B.13 Complete modified LIDAR.f90 script (part 5). 229
B.14 Complete modified LIDAR.f90 script (part 6). 230
B.15 Complete modified LIDAR.f90 script (part 7). 231
B.16 Complete modified LIDAR.f90 script (part 8). 232
B.17 Complete modified LIDAR.f90 script (part 9). 233
B.18 Code added within the FAST_Subs.f90 script to initiate the transfer of the hub position from ElastoDyn to InflowWind. 233
B.19 Code added within the FAST_Solver.f90 script to set hub position outputs from ElastoDyn to InflowWind LIDAR inputs. 233
B.20 Code added within the FAST_Subs.f90 script to initiate the transfer of LIDAR outputs as ServoDyn inputs. 234
B.21 Code added within the FAST_Solver.f90 script to set LIDAR outputs as ServoDyn inputs. 234
B.22 Code added within the BladedInterface_EX.f90 script to enable the allocation of LIDAR outputs to the avrSWAP array. 235
B.23 Code added within the ReadSetParameters.f90 script to enable the reading of LIDAR data from the avrSWAP array. 236
B.24 Code added within the ReadSetParameters.f90 script to enable the reading of LIDAR-assisted control inputs from the ROSCO input file. 236

B.25 Variables added to the `ControlParameters` type within the `ROSCO_Types.f90` script to enable the specification of the LAC parameters added to the ROSCO input file. 237
B.26 Variables added to the `LocalVariables` type within the `ROSCO_Types.f90` script for use within the feedforward control subroutines. 237
B.27 Updated ROSCO input file (part 1) with the new LIDAR-assisted control flag. . . 238
B.28 Updated ROSCO input file (part 2) with a new section added for specification of LIDAR-assisted control parameters. 239
B.29 LIDAR configuration subroutines created to determine the rotor azimuth angles of the LIDAR beams and to determine the LIDAR-measured rotor-effective wind speed.240
B.30 FFCPC subroutine added within the `Controllers.f90` script. 241
B.31 FFCPC PI controller function added within the `Functions.f90` script. 241
B.32 FFIPC subroutine added within the `Controllers.f90` script. 242
B.33 Complete pitch control subroutine (part 1), which calls to the LIDAR configuration, FFCPC and FFIPC subroutines. 243
B.34 Complete pitch control subroutine (part 2), which calls to the LIDAR configuration, FFCPC and FFIPC subroutines. 244
B.35 Complete pitch control subroutine (part 3), which calls to the LIDAR configuration, FFCPC and FFIPC subroutines. 245
B.36 Code added within the `FAST_Reader.py` script to enable parsing of LIDAR simulator inputs the `modelling_schema.yaml` input file. 245
B.37 Code added within the `FAST_Writing.py` script to enable writing of LIDAR simulator inputs to the InflowWind input file. 245
B.38 Enabling of LIDAR simulation wind measurement outputs within the `FAST_Vars_Out.py` script to be output within the `.out` file of OpenFAST. 246
B.39 Code added within the `Utilities.py` script to enable writing of LIDAR-assisted control inputs to the ROSCO input file. 246
B.40 Code added within the `Controllers.py` script to enable LIDAR control flag within the ROSCO input file. 246

D.1 Example of a master .fst OpenFAST input file used when performing the OpenFAST simulations. 248
D.2 Example of an InflowWind input file used when performing the OpenFAST simulations. 249
D.3 Example of an ElastoDyn input file used when performing the OpenFAST simulations (part 1). 250
D.4 Example of an ElastoDyn input file used when performing the OpenFAST simulations (part 2). 251

D.5 Example of an ElastoDyn input file used when performing the OpenFAST simulations (part 3). 252
D.6 Example of an ElastoDyn tower input file used when performing the OpenFAST simulations. 252
D.7 Example of a ServoDyn input file used when performing the OpenFAST simulations 253
D.8 Example of an AeroDyn input file used when performing the OpenFAST simulations (part 1). 254
D.9 Example of an AeroDyn input file used when performing the OpenFAST simulations (part 2). 255
D.10 Example of a HydroDyn input file used when performing the OpenFAST simulations 256
D.11 Example of an MLife input file used when undertaking the OpenFAST simulation post-processing (part 1). 257
D.12 Example of an MLife input file used when undertaking the OpenFAST simulation post-processing (part 2). 258
D.13 Example of an MLife input file used when undertaking the OpenFAST simulation post-processing (part 3). 259
D.14 Example of a FAST.Farm .fstf master input file used when performing the FAST.Farm simulations (part 1). 260
D.15 Example of a FAST.Farm .fstf master input file used when performing the FAST.Farm simulations (part 2). 261

Tables

1.1 Summary of the published and under-review conference proceedings papers and academic journal articles that were prepared within this research project. 33

2.1 Summary of the reviewed LIDAR-assisted FFCPC papers and their key findings. 36

3.1 Functions of key source code `Types` used within the OpenFAST source code. . . 52
3.2 LIDAR input variables added to the InflowWind module source code and included within the InflowWind input file. 53
3.3 Functions of key source code `Types` used within the ROSCO source code. . . . 60
3.4 LAC inputs added to the ROSCO input file. 62
3.5 Determination of beam rotor azimuth angular positions. 62
3.6 Relevant WEIS/WISDEM Python scripts and `.yaml` files and their functions. . . . 63

5.1 IEC DLC 1.2 NTM wind and wave conditions, from Allen et al. (2020). 81

5.2 Ranges recorded during the ten-minute excerpts shown in Fig. 5.1 of each parameter of each FOWT. 83
5.3 Key performance indicating metrics and their significance to LAC performance assessment. 88
5.4 Average normalised metric values of various parameters of the VolturnUS-S when using the different FB and FF control strategies under turbulent wind. 91
5.5 Normalised average metric values of various parameters of the WindCrete using the different FB and FF control strategies under turbulent wind. 100
5.6 Four-seed-average mean and maximum blade azimuth prediction errors and mean rotor speed errors. 110

6.1 ScotWind FOW regions examined in the O&M case study. 114
6.2 Mean wind speeds and Weibull parameters of the assessed locations. 119
6.3 Turbine sub-systems considered in the O&M case and the OpenFAST outputs associated with them. 121
6.4 IEC DLC 1.2 NTM wind and wave conditions, from Allen et al. (2020). 122
6.5 Normalised values of failure rates for turbine sub-systems resulting from the loads recorded by the OpenFAST simulation outputs when using FFCPC+FFIPC relative to the FB-only controller at each ScotWind site. 123
6.6 OpEx savings breakdown based on the reduction of failure rates. 124
6.7 Percentage savings in OpEx from FOW with WI limits and LAC failure rate reductions, compared to the equivalent BFW baseline. 126

7.1 Outer diameter and shell thickness properties at different heights of the baseline tower of the VolturnUS-S. 131
7.2 Percentage reductions applied to the baseline tower outer diameter or shell thickness for each simulation experiment number. 132
7.3 Mean and maximum platform pitch angle and maximum tower-top acceleration of an FB-only-controlled FOWT supported by the baseline or lightweight tower recorded during six ten-minute simulations of DLCs 1.6 and 6.1. 144

8.1 Mean power of each turbine, normalised to that of T1, for different turbine spacings at a mean ambient wind speed of 11 ms^{-1}. 163

A.1 Parameters of the IEA 15 MW reference turbine, from Gaertner et al. (2020). . . 215
A.2 Parameters of the VolturnUS-S and WindCrete FOWT models. 215
A.3 Rigid body natural frequencies of the VolturnUS-S and WindCrete FOWT models. 216

E.1 Normalised standard deviations of key turbine and substructure parameters of the VolturnUS-S using FB-only control at increasing above-rated wind speeds. 262

E.2 Normalised standard deviations of key turbine and substructure parameters of the WindCrete using FB-only control at increasing above-rated wind speeds. 262

F.1 Data excerpt beginning and total simulation times used within the FAST.Farm simulations. 263

Chapter 1

Introduction

1.1 Research context

1.1.1 The growth of the offshore wind energy industry

The offshore wind energy industry is growing rapidly, accelerated by the demand for global decarbonisation amid the rising threats of climate change. Countries worldwide are recognising the need for cost-effective, socially acceptable and scalable renewable energy, such as offshore wind, to reduce their dependence on oil and gas and to increase energy security.

Offshore wind energy provides a renewable alternative to fossil fuels and generates electricity without emitting harmful greenhouse gases, meaning that it is set to play a leading role within the energy sector as countries seek to meet their net-zero emissions targets. The UK's government, for example, has targeted a domestic offshore wind power capacity of 50 GW by 2030 (see HM Government, 2023), up from the 11 GW that had been installed by 2021, highlighting an appetite to accelerate the expansion of their offshore wind energy portfolio as they aim to reach net-zero by 2050 (see Department for Energy Security and Net Zero, 2023b).

The power ratings of offshore wind turbines also continue to increase, with models of up to 22 MW set for development by 2025 (see Durakovic, 2023). This represents a significant increase in turbine ratings, which had a maximum capacity of only 8 MW in 2016. The drive for larger turbines is motivated by the fact that larger rotor diameters allow wind turbines to sweep more area, capture more wind and, therefore, generate more electricity. This ultimately leads to a lower levelised cost of energy (LCOE) (indicating the lifetime cost per unit of electricity generated) of offshore wind when using larger turbines. The LCOE is reduced by increasing the turbine capacity not only because the rated power can be increased, thereby generating more electricity and revenue per turbine, but also because the *number* of turbines can be reduced for the same wind farm total capacity, meaning that the overall volume of materials as well as the number of manufacturing, deployment and installation operations, and the costs associated with them, can also be reduced.

However, the caveat of increasing the size of wind turbines is that their larger rotor diameters ultimately render their longer blades subject to greater deflections and bending moments. These greater loadings extend to other turbine components, meaning that the turbines must be structurally reinforced to withstand them, incurring greater material usage and cost per turbine. Another concern is that the demand for fast upscaling to larger turbines applies significant pressure and expense to the wind turbine original equipment manufacturers (OEMs), who are required to continually develop their turbine portfolios and manufacturing capacities. This has led to claims that the continued increases in turbine capacity at the current rate are unsustainable (see Blackburne, 2023). The fast growth of turbines is also greatly outpacing the growth in the capability and availability of the vessels and cranes used for construction, deployment and maintenance, where there is a profound scarcity of those that can handle the significant heights, blade lengths and masses of these next-generation turbines. Ultimately, these challenges raise concerns over the practicality and economics of further increasing wind turbine capacities at the current rate, despite the energy generation benefits that larger turbines can offer.

1.1.2 The case for floating offshore wind

Bottom-fixed offshore wind (BFW) turbines, consisting of offshore wind turbines mounted on foundations that extend to the seabed, can only be deployed to a maximum water depth of approximately 60 m (see Paya & Zigeng, 2020). This presents a further issue in that only 7% of the world's oceans have a depth of less than 200 m (see WHOI, 2023), and Wind Europe (2017) estimates that around 80% of Europe's available wind resource lies within waters exceeding 60 m depth. This means that an overwhelming proportion of the world's offshore wind energy resource cannot be extracted by BFW turbines. Floating offshore wind turbines (FOWTs) provide a solution to this problem because they can capture wind power in waters that are too deep for the use of BFW turbines. A FOWT consists of an offshore wind turbine mounted upon a floating platform that is connected to anchors on the seabed with mooring lines composed of steel chains and/or synthetic fibre ropes. Dynamic electrical cables, which are designed to be suspended within the water and connect to the cabling on the seabed, are also required to enable the export of electricity from each FOWT.

Typically, the greater water depths within which FOWTs are deployed are found in locations that are greater distances offshore, where winds are strongest. However, this is not always the case, as many countries, such as Japan, for example, do not benefit from a continental shelf, meaning that water depths are too great for BFW even in close proximity to the shore. In these instances, floating offshore wind (FOW) is the only viable option for these countries to exploit their offshore wind resource and so FOW presents a practical solution to BFW for many locations around the world. The stronger winds found at the further offshore FOW sites can lead to the potential for FOWTs to achieve higher capacity factors (the ratio of actual

power produced in a given time period to the hypothetical maximum possible) than BFW sites, allowing FOWTs to capture more power and generate more revenue. Within the UK, Energy Numbers (2022) reported that the offshore wind farm with the highest recorded lifetime capacity factor is Hywind Scotland, a 30 MW (5 x 6 MW turbines) floating wind pilot project in the North Sea, at 52.6%.

1.1.3 Floating offshore wind challenges

Despite the benefits that they bring in terms of power capture and site location that is not limited by the water depth, FOWTs are subject to heightened challenges in numerous areas compared to BFW, leading to a higher LCOE. Firstly, the floating offshore wind sector is in its relative infancy compared to BFW, with less than 0.3 GW installed globally (see Quest FWE, 2024) compared to approximately 72.5 GW of BFW (see statista, 2024), at the time of writing. This means that costs are inevitably higher from the outset, as technology standardisation (of the floating platforms, moorings and cabling) has not yet been established, nor have the cost savings that can be expected from its learning rate. Standardisation refers to the use of the same or similar technology within each project. This means that supply, manufacturing, construction and installation processes can be streamlined and up-scaled, becoming more efficient and thereby reducing costs. An embodiment of this can be found within the wind energy industry, where the use of three-bladed horizontal-axis wind turbines has allowed for standardised mass production to be achieved. Learning rates refer to the fractional reduction in the cost for each doubling of cumulative capacity of energy technologies (see Rubin et al., 2015), as a result of the experience gained from the repetition of the same processes undertaken within different projects (see McDonald & Schrattenholzer, 2001). As the FOW industry matures and its installed capacity increases, its LCOE is expected to reduce due to its learning rate and potential convergence towards technology standardisation.

FOWTs are also subject to further design, construction and installation challenges that do not apply to BFW turbines. This in large part stems from the use of the floating platforms and the additional raw materials that they (and the towers of FOWTs) require, with each FOWT requiring at least 4-10 kilotonnes of additional raw material compared to BFW turbines (see Allen et al., 2020; Mahfouz et al., 2020). This incurs significant additional capital cost in addition to the concerns over the cost volatility and availability of supply of the required steel (see CRU Group, 2021), as well as the degree of CO_2 emissions (see Sustainable Ships, 2023) and water consumption (see Walling & Otts, 1967) of its manufacturing process.

In addition to the volume and cost of the raw material, the limited availability of appropriate port and grid infrastructure also hinders the wider adoption of FOW. Many European ports and harbours lack the necessary space, quayside depth and capacity to handle the storage and construction demands of such assets, which is a growing concern given the rapid pace of BFW and FOW site deployment. Further issues arise when considering the mating of the

rotor-nacelle assembly (RNA), tower and floating substructure, as well as the deployment and transport of the assets to site. FOWTs are also subject to additional sub-sea design and installation challenges (which do not apply to BFW turbines) with respect to the moorings, anchors and dynamic power cables.

Furthermore, while it is widely cited that operations and maintenance (O&M) costs can account for up to 30% of the total cost of energy within bottom-fixed offshore wind (see Feng et al., 2010), recent studies have estimated that, for FOW technology, this figure could increase to almost 40% (see BVG, 2023). This is in part due to their greater distances offshore and placement in harsher environments, meaning that heightened accessibility challenges and costs emerge when seeking to perform maintenance of FOWTs.

1.1.4 The LIDAR-assisted control solution

The additional challenges and subsequent higher costs of FOW mean that methods to reduce these costs are highly desirable. The application of Light Detection and Ranging (LIDAR)-assisted control is one such method that shows promise for reducing the LCOE of FOW. Nacelle-mounted, forward-facing LIDAR can provide measurements of the incoming wind field, the characteristics of which can be used for feedforward (FF) turbine control. Traditionally, wind turbine control strategies have utilised feedback (FB) control, which updates the controller settings following the impact of the incoming disturbance. However, the drawback of this type of control strategy is that the wind turbine reacts to the wind once it has already impacted upon it. FF control overcomes this drawback by enabling turbines to actuate their control systems in advance of the wind's impact upon the rotor, and has consistently shown benefits for improving the performance and reducing the loadings and motions of FOWTs when assisting with blade pitch control in above-rated wind speed conditions.

This thesis will develop, simulate and assess applications of LIDAR-assisted FF pitch control (LAC), with the key aim of quantifying the cost benefits achievable through the implementation of the technology. These cost benefits can ultimately aid the FOW industry by leading to reductions in the LCOE of FOW, making projects more commercially viable. These benefits may take the form of operational/capital cost reductions or increased revenue.

1.2 Project context and industrial applicability

This project was supported by Wood Renewables and Flotation Energy, an engineering consultancy and offshore wind farm developer, respectively, both based in the UK and internationally. The findings of this research are directly applicable to both companies' respective FOW industry sectors. The project has supported the enhancement of wind turbine modelling tools developed by Wood while demonstrating the opportunity to reap the commercial benefits

of LAC as a product within their LIDAR services arm. The research findings also highlight to developers, such as Flotation Energy, the potential for LAC to deliver cost reductions to their projects, with Flotation Energy's (in-progress) Green Volt (set to become Europe's first commercial-scale floating wind farm with a capacity of up to 560 MW, see Flotation Energy and Vårgrønn, 2024) and White Cross (a floating wind pilot project of up to 100 MW capacity) projects particularly relevant to the findings of this research. The knowledge of the potential cost benefits of LAC can allow developers to lead collaborations with OEMs to deliver better performance of turbines and floating platforms, which can ultimately lead to greater return on investment of projects.

Within industry, LAC has been recognised by wind turbine OEMs such as Goldwind, who received the world's first certification for LAC technology in 2019 (see DNV GL, 2019), as an opportunity to reduce the loads and costs of wind turbines, with 1,000 of their turbines fitted with LAC by the end of 2020 (see sowento, 2021). International Energy Agency (IEA)-Wind (2022) also have an ongoing project task (number 52) focused on accelerating the large-scale deployment of wind LIDAR for applications such as LAC. These highlight an appetite within the industry to utilise LAC to improve the performance and reduce the loads acting on wind turbines.

Research focused on determining the cost benefits of LAC is critical to the enhancement of LAC's commercial acceptability for large-scale use on FOWTs, and so this was a primary focus throughout this research project. The cost savings achievable from mass deployment of LAC on FOWTs are not well documented, at least within the public domain. Therefore, this work aimed to highlight the potential cost savings that can be delivered by LAC, in addition to assessing its applicability to large-scale 15 MW turbines mounted on different floating platform configurations.

1.3 Fundamental concepts

Following the overview of the context of the research and its industrial applicability, greater detail will now be provided on the fundamental technical themes and concepts pertinent to this thesis and their relevance to LAC. These include wind modelling, LIDAR, FOWT platforms, blade pitch control, FF control, O&M, FOWT tower design and wake effects.

1.3.1 Wind modelling

The wind velocity at a particular location can be characterised by a vector encompassing the longitudinal (u, along the x-axis direction of the mean wind velocity), transverse (v, along the y-axis) and vertical (w, along the z-axis) components of the wind speed. The wind speed also varies with height, a phenomenon termed wind shear. Wind speeds tend to be lower near the surface of the Earth or the ocean due to friction. The relationship of wind speed with height can be approximated by a power law relationship (see IEC 61400-1:2019, 2019).

$$V(z) = V_{hub}\left(1 + \frac{z}{z_{hub}}\right)^{a} \tag{1.1}$$

where z is the height above the mean sea level, z_{hub} is the hub height, V_{hub} is the wind speed at the hub height of a wind turbine and a is the wind shear exponent. The IEC 61400-1:2019 (2019) design requirements for offshore wind turbines recommend design to a = 0.2 for standard wind turbine classes.

Wind is also subject to turbulence (rapid fluctuations in the wind velocity) causing variable loads on wind turbine components. A typical way to characterise the degree of turbulence is the turbulence intensity (TI)

$$TI = \frac{V_{SD}}{\bar{V}} \tag{1.2}$$

where V_{SD} is the standard deviation (SD) of the wind speed, and $\bar{V}$ is the mean wind speed. Both V_{SD} and $\bar{V}$ are determined over a predefined time period, typically 10 minutes or 1 hour. These time frames are suitable for turbulence assessment because Van der Hoven (1957)'s wind spectrum strongly suggests that turbulence dominates wind variation within these periods. Turbulent wind fields can be generated for use in wind turbine modelling tools by employing the Mann (1994) or Kaimal et al. (1972) turbulence models, as recommended in the IEC 61400-1:2019 (2019) standards. These models are used to provide mathematical representations of the turbulent structure of the wind within the atmospheric boundary layer. The IEC 61400-1:2019 (2019) standards specify various turbulence categories (A, B or C), signifying the degree of the turbulence intensity within a wind field, which can be applied when performing wind turbine design simulations. These categories are also linked to wind field turbulence representations of normal (Normal Turbulence Model [NTM]) or extreme (Extreme Turbulence Model [ETM]) conditions, as defined by IEC 61400-1:2019 (2019), which are used within the design load cases (DLCs) specified within the standards (see IEC 61400-3-1:2019, 2019). The DLCs are critical aspects of wind turbine design and verification because they provide a suite of wind, wave and current condition scenarios (such as co-directional or misaligned wind and wave directions) that a wind turbine is likely to be subjected to over its lifetime, and which are recommended to be simulated in order to predict the responses of wind turbines, estimate their fatigue life and verify their structural survivability.

1.3.2 LIDAR

LIDAR operates by firing high-speed infrared laser pulses, which are reflected by aerosols (solid particles or liquid droplets suspended in the air) moving with the wind. The sensor can then measure the time taken for the pulses to return. The sent and reflected wavelengths of the laser are compared and the Doppler effect is used to analyse the frequency shift of laser light scattered by the atmospheric particles carried by the wind to determine the wind speed (see Mikkelsen, 2014; Simley & Scholbrock, 2020).

There are two main LIDAR beam configurations, continuous-wave (CW) and pulsed, illustrated in Fig. 1.1.

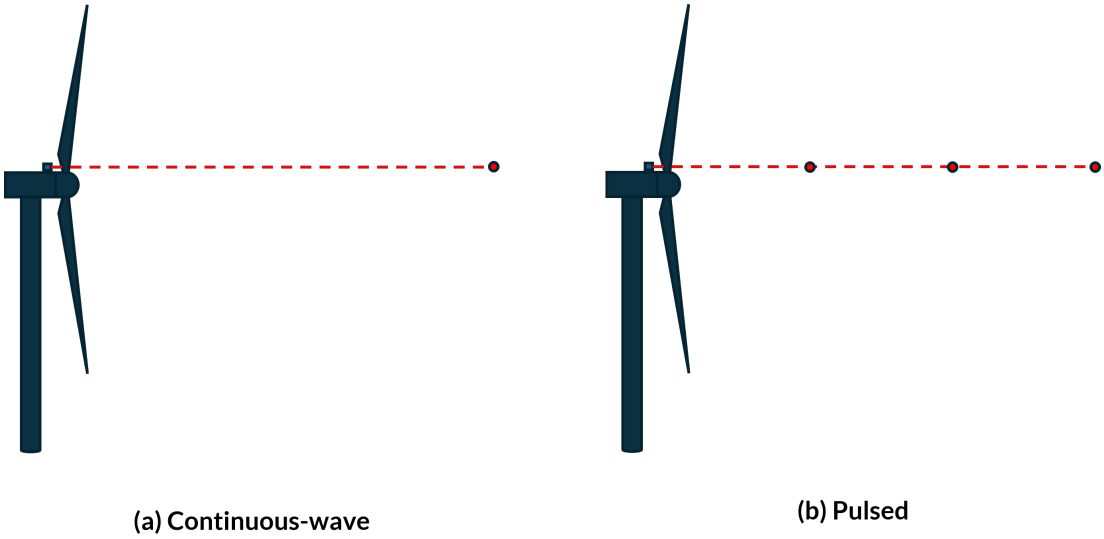


Figure 1.1: Nacelle-mounted single-beam LIDAR configurations.

As the names suggest, CW LIDAR emits laser light continuously, leading to higher temporal resolution compared to pulsed LIDAR, which instead emits short laser pulses in intervals. However, the spatial resolution of pulsed LIDAR is superior to that of CW LIDAR because it can be configured to measure at multiple points along the beam(s). CW LIDAR (Fig. 1.1a) uses beams that are focused at the desired distance ahead of the LIDAR, termed the focal distance. CW LIDAR beams are only capable of recording measurements at one specified focal distance at a time, though this can be adjusted. More information on LIDAR's mode of operation and these beam configurations can be found via a webinar delivered by Simley and Scholbrock (2020).

In both CW and pulsed configurations, the beam is modelled as a Gaussian or Lorentzian beam, whereby a range weighting function defines the probe length at the focal point (see Simley & Scholbrock, 2020). These have a characteristic intensity profile that follows a curve similar to that shown in Fig. 1.2. The intensity distribution across the longitudinal plane of the beam is often modelled using a weighting function to represent the significance of different

regions within the beam (see Fig. 1.2). The probe length refers to the axial region along the beam where the wind velocity is measured whereas the probe volume defines the total spatial region contributing to the LIDAR measurement. The probe volume, and by extension the uncertainty of the wind velocity measurement of CW LIDAR's increases with the square of the focal distance (see Fu et al., 2022), whereas the probe volume of pulsed LIDAR does not depend on the focal distance.

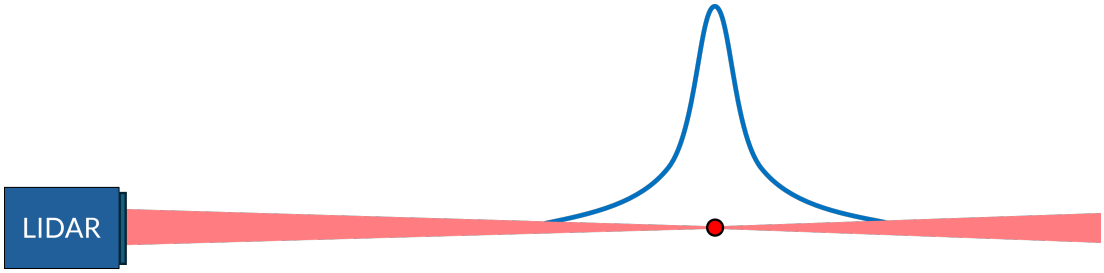


Figure 1.2: Weighting function used to determine the spatially averaged wind speed along the beam of a CW LIDAR. Adapted from Simley and Scholbrock (2020). Reprinted with permission from the National Renewable Energy Laboratory (NREL).

A LIDAR will typically be configured with several laser beams oriented in appropriate directions in order to probe the chosen volume. This configuration is referred to as a scan geometry. The line-of-sight (LOS) wind velocity is the component of the wind velocity that is projected onto the direction of a particular LIDAR beam. Since the LIDAR is only capable of measuring LOS wind velocities, a wind field reconstruction (WFR) method is invoked in order to determine the wind velocity field within the measurement volume. For a LIDAR positioned on, or at, a wind turbine, the WFR method provides reconstruction of wind speed and direction across the rotor-swept area from the LOS wind velocity. For a nacelle-mounted LIDAR, the WFR method provides estimation and isolation of the longitudinal (u) wind component, which dominates the aerodynamic loading of wind turbines over the lateral (v) and vertical components (w) and is used by LAC to formulate FF commands. When using multiple beams focused at multiple points within the rotor-swept area of a wind turbine, the data from them can be used to estimate the rotor-effective wind speed (REWS), representing the mean longitudinal wind speed across the rotor-swept area.

Measurement uncertainties

There are a number of sources of uncertainty associated with LIDAR measurement. These typically stem from the position of the LIDAR device, the wind profile and the configuration of the LIDAR beams.

Firstly, there are uncertainties stemming from the pointing accuracy of the LIDAR beams. Fundamentally, LIDAR beams deliver their most accurate wind speed measurements when they are perfectly aligned with the incoming wind direction. If a beam is not perfectly aligned with the wind direction when it is supposed to be, the measured LOS velocity will not represent the true longitudinal wind speed (u), as it will include additional contributions from the v and w components of the wind velocity vector (see Simley et al., 2012). If the beams are projected at angles, in terms of elevation and azimuth, then corrections have to be made to ensure that the u velocity component is isolated from the v and w components, given that LIDAR can only measure the wind velocity along its LOS direction. Deviations from the intended elevation and azimuth angles of the beams can also cause errors in LIDAR measurements, especially when long-range measurements are taken, as this results in measurements being taken at positions different from the intended positions (see Rott et al., 2022). Furthermore, the wind profile, particularly the wind shear, can lead to underestimation or overestimation of the wind speed due to the spatial averaging of the wind speed, which acts as a spatial filter along the LIDAR beam (see Simley et al., 2012).

The uncertainty of LIDAR measurement is further exacerbated when the device is mounted on the nacelle of a FOWT, as the platform motions render the movements of the LIDAR device to be much greater than when mounted on the nacelle of a bottom-fixed turbine. Not only does this amplify the aforementioned uncertainties and errors relating to the positions and angles of the LIDAR beam(s), but the quality of the LIDAR measurement can be affected by the linear velocity of the LIDAR. As the Doppler shift of LIDAR measurement is assumed to be taken between the positions of a stationary observer (the LIDAR) and the focal distance, this means that the motions of the LIDAR on the FOWT will be included within the Doppler effect, leading to errors in the measurements. Studies of LAC of FOWTs have typically applied a motion compensation algorithm to remove the contribution of the LIDAR velocity from the LOS wind speed measurement (see Guo et al., 2022; Schlipf et al., 2009) and mitigate the errors caused by the motions of the LIDAR device on FOWTs.

Moreover, measurement uncertainties emerge when a WFR algorithm is invoked, primarily due to the assumptions and approximations that are applied. For example, the WFR method is typically based on a wind field model that assumes shear and horizontal homogeneity of the wind velocity field. Consequently, this assumes that the wind field is homogeneous across the probe volume, when, in reality, it varies due to phenomena such as turbulence and wind shear, and these effects may not be captured when using spatially-averaged measurements (see Clifton et al., 2015). Another assumption is that of Taylor's Frozen hypothesis (see Taylor, 1938), which assumes that the turbulence measured by the LIDAR propagates unchanged from the point of measurement to the rotor, though turbulent eddies in fact evolve as they move.

Finally, an important source of error and uncertainty in LIDAR measurement is the "Cyclops" dilemma. This refers to the fundamental limitation of single-beam LIDAR measurements, whereby a single beam LIDAR cannot directly distinguish between horizontal wind shear and wind direction changes (see Towers & Jones, 2016). This is important because changes in wind speed across beams can result from either horizontal shear or changes in direction, but both can result in the same LOS measurements. This leads to the requirement for multiple beams pointed at different angles to allow for the spatial variation of the wind field to be better captured and for shear to be distinguished from directional changes.

From a practical perspective, LIDAR availability and reliability is important to consider and maximise for applications such as LIDAR-assisted control. For example, LIDAR relies on airborne particulates to backscatter the emitted light, but a low concentration of these particles can reduce the signal strength and measurement accuracy (see P. B. Russell et al., 1979). Furthermore, contaminants such as dirt, salt and ice can obstruct the lenses and limit the signal strength (see Schlager et al., 2022). Other issues may arise due to hardware malfunctions or problems with the transmission of the LIDAR measurement to the controller.

1.3.3 Floating offshore wind turbine platform designs

FOWTs are required to be mounted on floating platforms to support the wind turbines and allow them to capture wind energy in deep water locations. These are typically connected to anchors on the seabed using mooring lines to ensure station-keeping. Generally, FOWT platform designs are categorised into four distinct configurations, shown in Fig. 1.3 (from ACTEON, 2020). The principle design driver that forms the differences between these configurations is the way in which they counteract the overturning moment generated by the wind turbine rotor thrust force in order to achieve stability (see A. Henderson et al., 2016). This is accomplished either through buoyancy (semi-submersibles and barges), gravity (spars) or moorings design (tension leg platforms [TLPs]), which ultimately results in the geometrical differences seen in Fig. 1.3.

Semi-submersibles have begun to emerge as the most common FOWT platform design configuration (see Quest FWE, 2024) due to their relatively low draft (the distance between the still water level and the bottom of the platform), making them suitable for sites with lower water depths (such as in the North Sea) and for ports which have limited water depth at the quayside. Semi-submersibles also benefit from ease of deployment and installation, as the turbine can be mated with the platform at the quayside and the complete FOWT can be towed to site using standard vessels. However, the disadvantages of semi-submersibles are their relatively inferior stability, their high material usage and their size, demanding large areas for storage and assembly at the port.

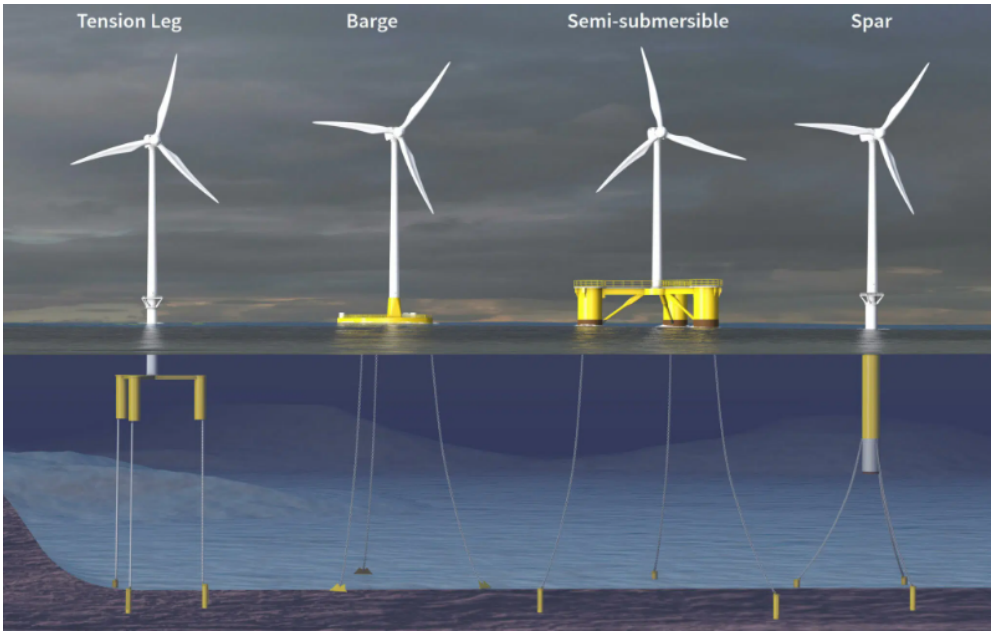


Figure 1.3: Floating offshore wind platform configurations, from ACTEON (2020).

Barges follow a similar principle to semi-submersibles in that their stability comes from their buoyancy as a result of their large water-plane area. However, despite their superior stability and scalability to semi-submersibles, they suffer from more challenging construction and installation, higher draft requirements at port and greater material usage.

The stability of TLPs is driven by the design of their mooring system, whose lines are required to be under constant tension. This constrains the movement of the platform, resulting in TLPs offering the greatest stability of all of the design configurations whilst demanding the lowest volume (and therefore cost) of materials. However, the challenge of TLPs stems from their difficulty of installation. Specialised vessels are often required when installing TLPs because they are unstable when being towed. Additionally, the mooring line connection to the TLP during installation is challenging due to the need to ensure that the correct pre-tensioning is achieved. TLPs are also required to be deployed in deeper waters than barges and semi-submersibles due to their greater drafts.

Finally, the stability of spars is achieved because they are designed such that their centres of gravity are situated below their centres of buoyancy. This requires the weight of the spar to be concentrated at its base, which entails the use of large quantities of ballast. Spars have a simple geometry and so are relatively easy to design and construct, though challenges emerge when deploying and installing them due to their large draft requirement, meaning that they are only suitable for use at deep water sites.

Overall, the suitability of a platform configuration for a given site is governed by several factors including the water depth, the met-ocean conditions, the seabed conditions (which inform the mooring and anchoring design), the supply/manufacturing capacity (domestically and internationally) and the port/vessel infrastructure capabilities for construction, deployment and installation. Therefore, it is probable that different FOW markets and locations around the world will be better suited for the use of different floating platform technologies depending on these factors. Consequently, a one-size-fits-all approach with regard to the floating platform design is unlikely to be achieved for FOW. Therefore, it is pertinent to investigate the benefits of LAC for different FOWT platform designs in order to assess its applicability (or lack thereof) to them and, by extension, to the different FOW markets and locations that may utilise them.

1.3.4 Turbine operating regions

Wind turbines are typically categorised into three operating regions, per J. Jonkman et al. (2009) and Gaertner et al. (2020), as illustrated in Fig. 1.4:

- Region 1, when the wind speed is below the cut-in speed of the turbine and it does not operate.
- Region 2, between cut-in and rated wind speed, when the rotor speed is varied through generator torque control and with a fixed blade pitch, to maximise the rotor efficiency.
- Region 3, when the wind speed is above-rated, but below the cut-out speed. In this region, there is more power in the wind than the turbine is rated to capture. Therefore, the generator torque is fixed and blade pitch control is used to shed power to maintain the rotor at its rated speed and, by extension, maintain the power at the turbine's rated capacity.

There also exists a transition region termed Region 2.5, where the rotor speed reaches its rated value before the generator torque reaches its rated value (see J. Jonkman et al., 2009). However, as a result of its design optimisation, the IEA 15 MW turbine (whose power curve is shown in Fig. 1.4) has a negligible Region 2.5, since the rated rotor speed is reached at rated torque (see Gaertner et al., 2020). When the wind speed is above the cut-out speed, the turbine does not operate in order to protect it from extreme loads and damage.

1.3.5 Blade pitch control

The greatest benefits of LAC have been delivered when assisting with blade pitch control in above-rated wind speed conditions (as opposed to torque control in below-rated conditions, see Scholbrock et al., 2016), and so this will also be the area of focus of this thesis. Therefore, this Section will provide details on the fundamental terms and principles pertinent to blade pitch control, as these are actively influenced by LAC.

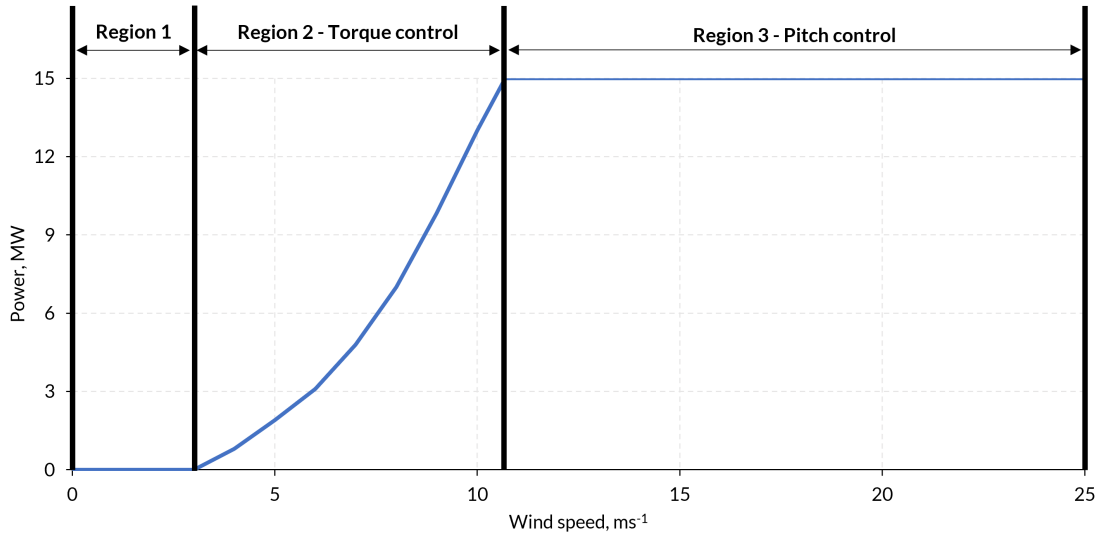


Figure 1.4: Power curve of the IEA 15 MW reference turbine, adapted from Gaertner et al. (2020), indicating the key operating regions and their control modes. Reprinted with permission from NREL.

Blade aerodynamics

When using active pitch control in above-rated wind speed conditions, the aerodynamics of wind turbine blades are managed in order to maintain the rotor rotational speed at its setpoint. The following defines the terminologies pertinent to wind turbine blade aerodynamics (see Hansen, 2015). These terms are illustrated in Fig. 1.5.

- An **airfoil** refers to the cross-sectional shape of the blade that is designed to efficiently harness energy from the wind, shown in Fig. 1.5. Airfoils are designed such that their shape creates a pressure difference between the upper and lower surfaces when wind flows over the blade, generating lift and causing the blades to rotate.
- The **blade pitch angle ($\boldsymbol{\beta}$)** refers to the angle between the chord line (a straight line connecting the leading and trailing edges of the airfoil) and the rotational plane. β is modified by the pitch controller in above-rated wind speeds.
- The **angle of attack (AoA)** of an airfoil is the angle between the chord line and the relative wind direction. The AoA determines how the air flows over the airfoil and affects the lift and drag forces, which determine the speed of the rotor's rotation.
- The **lift force ($\boldsymbol{F_L}$)** is the upward force generated by the airfoil as the wind flows over it. It is perpendicular to the direction of the wind and is responsible for generating the lift required to rotate the turbine's rotor. F_L can be calculated by

$$F_L = \frac{1}{2}\rho V^2 A_S C_L \tag{1.3}$$

where ρ is the air density, A_S is the blade span area and C_L is the coefficient of lift. C_L varies depending on the airfoil shape as well as the AoA.

- The **drag force (F_D)** is the resistance force experienced by the airfoil due to the airflow. It acts parallel to the direction of the wind and opposes the motion of the blade. F_D contributes to the overall aerodynamic load on the blade and can be calculated by

$$F_D = \frac{1}{2}\rho V^2 A_S C_D \tag{1.4}$$

where C_D is the coefficient of drag. C_D also varies depending on the airfoil shape as well as the AoA.

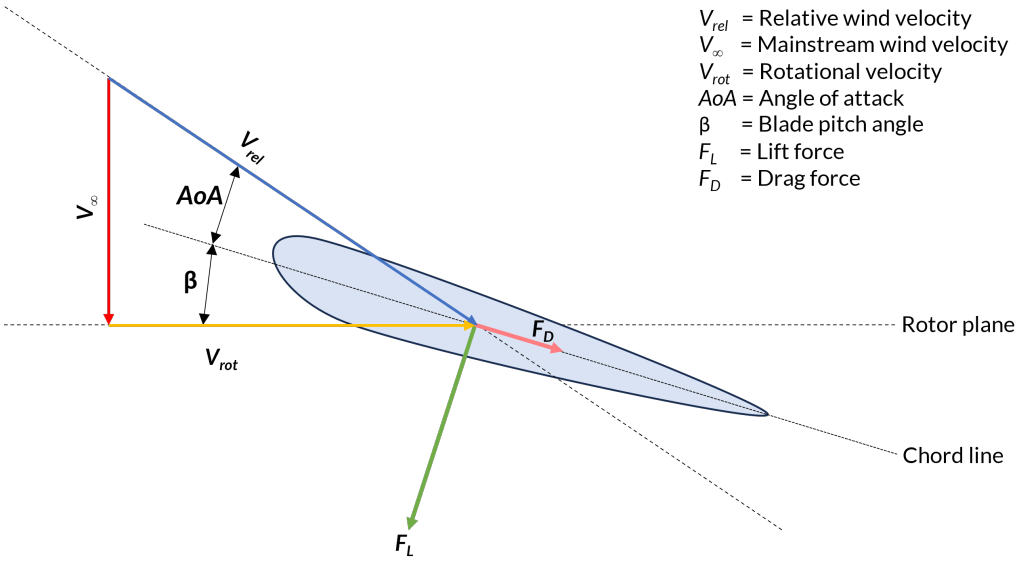


Figure 1.5: Forces acting on a blade airfoil.

Changes in β by the wind turbine controller impact upon the AoA and, by extension, C_L and C_D, which ultimately lead to changes in the rotor's rotational speed. The AoA is linked to C_L and C_D of an airfoil by the relationships illustrated in Fig. 1.6.

At low angles of attack, C_L increases linearly with the AoA. The critical AoA (indicated in Fig. 1.6), or the stall angle, occurs when C_L reaches its maximum value. This is where the airfoil is most efficient in generating lift. If the angle of attack is increased beyond the critical angle, C_L decreases sharply. The reverse is true for C_D, which increases significantly once the angle of attack rises beyond the critical angle. This results in stalling of the wind turbine's rotor beyond the critical angle.

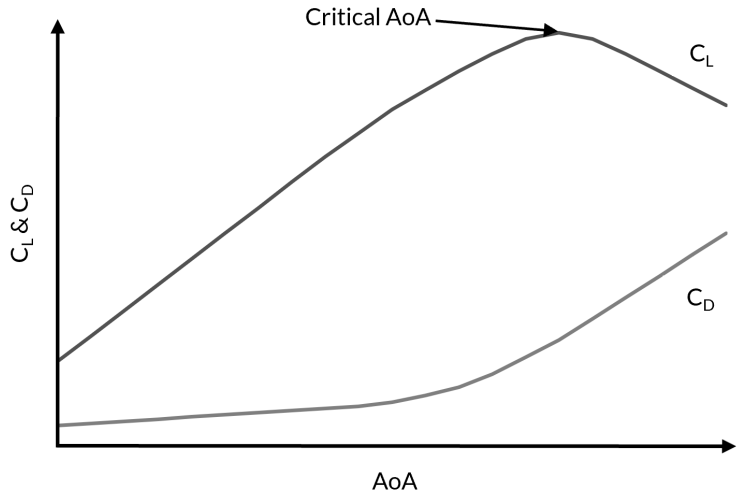


Figure 1.6: Illustrating the general trend of increasing angle of attack on C_L and C_D.

As the wind speed increases, the relative wind direction changes, increasing the AoA. Conversely, increasing β decreases the AoA, and vice versa. Therefore, the adjustment of β by the wind turbine controller helps to maintain the optimal AoA across varying wind speeds, ensuring efficient energy capture. The opposing relationships mean that higher values of β are required at greater wind speeds in order to reduce the AoA to within its optimal range. This ensures that C_L and C_D and, by extension, F_L and F_D are managed so that the rotor's rotational speed is maintained at its setpoint.

Pitch controller fundamentals

In wind turbine blade pitch control systems, the generator's rotational speed (Ω_g) is fed back and the error (e_{FB}) between it and the rated generator speed $(\Omega_{g,rated})$ is computed (see J. Jonkman et al., 2009)

$$e_{FB} = \Omega_{g,rated} - \Omega_g \tag{1.5}$$

The objective of the pitch controller is to minimise this error to ensure that the generator speed and hence the rotor speed is maintained at its rated value by adjustment of β and, by extension, the AoA, F_L and F_D. Having computed the error, this is typically input to a proportional-integral (PI) controller (see J. Jonkman et al., 2009). PI controllers aim to formulate commands to the actuation mechanisms in order to minimise the error between the FB variable's value and its prescribed setpoint. The proportional component of the FB pitch controller (P_{FB}) determines the ratio of the output response to the error signal, e_{FB},

$$P_{FB} = K_p \, e_{FB} \tag{1.6}$$

where K_p is the proportional gain. Increasing the proportional gain increases the speed of the response, but also increases the overshoot.

The integral component of the FB pitch controller, I_{FB}, determines the cumulative error over time. It can expedite the movement towards the setpoint and eliminate steady-state error

$$I_{FB} = I_{FB,prev} + \Delta T(K_i\, e_{FB}) \tag{1.7}$$

where $I_{FB,prev}$ is the integral component from the previous time step, ΔT is the time step and K_i is the integral gain. The proportional and integral components are then combined to give the FB pitch command (β_{FB})

$$\beta_{FB} = P_{FB} + I_{FB} \tag{1.8}$$

K_p and K_i are adjusted depending on the operating conditions. This is termed a gain schedule and is implemented to allow the controller to better match the non-linearities in the system and to improve its performance under different operating conditions (see Bao et al., 2016). As outlined by Abbas et al. (2020), the proportional and integral gains of a wind turbine's blade pitch controller at certain operating points are tuned by consideration of the desired damping ratio (ζ_{des}) and natural frequency (ω_{des}) of the controller

$$K_p(\upsilon) = \frac{1}{B}(2\zeta_{des}\omega_{des} + A_r(\upsilon)) \tag{1.9a}$$

$$K_i(\upsilon) = \frac{\omega_{des}^2}{B} \tag{1.9b}$$

where A_r is the rotor-swept area, υ is the REWS and B is the input matrix of the state-space model of the turbine, representing the sensitivity of the aerodynamic torque to changes in the blade pitch angle. The closed-loop response is characterised by ω_{des} and ζ_{des}. When assuming constant generator torque, these are the only two parameters required to be defined in order to tune the blade pitch controller, per Abbas et al. (2020).

It is possible to find a relationship between β and υ, as was demonstrated by Abbas et al. (2020). This allows for A_r to instead be expressed as a function of β ($A_r(\beta)$) and so Eqs. 1.9a and 1.9b can be redefined for the blade pitch controller as $K_p(\beta)$ and $K_i(\beta)$, per Abbas et al. (2020). This results in a gain schedule that varies as a function of β. This is more favourable than gain scheduling as a function of υ because β is the direct control variable that influences the AoA and aerodynamic forces. The use of β within the gain schedule is also simpler to implement within wind turbine control systems because υ is instead required to be measured or estimated (see Leithead et al., 2014). Additionally, as the gain schedule operates

by interpolation of several linear controllers designed for specific operating points, stability is only achieved with slow scheduling parameter variations (see Lescher et al., 2005). As υ is highly variable, this means that it is unsuitable for use as the gain scheduling parameter, whereas the relative stability of β makes it a more reliable parameter.

Collective pitch control

Collective pitch control delivers the same command to each blade, meaning that the pitch angles of all of the blades are equal at all times. This is a simple method to design and tune as only one input (the generator speed) and one output (the blade pitch command) are required by the controller.

Individual pitch control

Due to phenomena such as wind shear, wind veer (variation of wind direction) and turbulence, wind speeds can be highly variable across the rotor-swept area. This impact is amplified for larger turbines, which have greater rotor diameters, approaching 300 m, and thus the wind speed is even more variable across the rotor-swept areas of these turbines. As mentioned previously, the greater lengths and deflections of the larger blades also incur greater loads upon themselves and other turbine components. Individual pitch control (IPC) can be employed as a means of adjusting the pitch angles of blades independently, as they each experience different wind speeds and, by extension, loadings due to their different rotor azimuth positions. Therefore, it is anticipated that IPC will be an effective strategy for overcoming the greater asymmetric loads subjected to large turbines resulting from their large rotor diameters and blade deflections.

FB individual pitch control (FBIPC) requires the use of an additional control loop consisting of feedback of the three out-of-plane blade root bending moments and the rotor azimuth to be used as additional inputs. The operation of FBIPC is explained in more detail in Sect. 4.3. When discussing the use of IPC, it is important to consider that, while it has benefits for reducing the asymmetric loads on wind turbines, it, however, presents the drawback of increased blade pitch actuation, which can lead to a greater likelihood of wear and failure of the pitch control actuation mechanisms.

Bottom-fixed vs. floating offshore wind turbine dynamics

A commonly reported issue for FOWTs is that the pitch controller can cause instability in above-rated wind speeds (see Fleming et al., 2014; T. Larsen & Hanson, 2007; Yu et al., 2018). This is because the turbines are subject to softer foundation properties, leading to lower natural frequencies (defining how quickly a system oscillates when disturbed from its equilibrium position) and unfavourable coupling between platform motion and blade pitch control (see T. Larsen & Hanson, 2007; Yu et al., 2018), which can lead to negative aerodynamic damping, whereby oscillations increase in magnitude.

The primary issue is that, when using controllers designed for BFW turbines, the FOWT's pitch controller adjusts the blade pitch angles as the nacelle translates forwards and backwards relative to the wind, reducing the thrust when the motion is towards the wind and increasing the thrust when the motion is away from the wind. As the nacelle moves forward, its velocity relative to the wind increases. If the blade pitch control actuation frequency is similar to that of the floating platform rigid-body pitch oscillation natural frequency, the pitch controller acts to feather (increasing β to reduce F_L) the blades during the motion to slightly reduce the generator speed. The thrust force on the turbine is thereby reduced, further accelerating the nacelle forward and causing instability of the turbine.

The method most commonly used to overcome these instabilities is detuning of the blade pitch controller. Detuning is the method of reducing the proportional and integral gains of the pitch controller such that its bandwidth does not include the pitch mode of the platform (see Fleming et al., 2014). Therefore, the level of detuning is dependent upon the natural frequencies of motion of the floating platform, as these vary for the different configurations outlined in Sect. 1.3.3. Consequently, the pitch controller of a FOWT with a platform that has a lower pitching natural frequency must be detuned to a greater degree than a platform with a higher pitch natural frequency. Generally, detuning incurs a trade-off, whereby greater amounts of detuning lead to superior stability of the FOWT at the expense of inferior rotor speed regulation. Consequently, it has been reported that the maximum rotor speed of FOWTs is up to 30% higher than the rated rotor speed, compared to only 10% in onshore turbines (see Fischer, 2013; T. Larsen & Hanson, 2007), as a result of detuning of the pitch controller.

Another common strategy used to overcome the negative damping issue is the nacelle FB damping loop, first defined by van der Veen et al. (2012). This is also commonly referred to as the "parallel compensation method". Here, the nacelle (otherwise referred to as the tower-top) velocity ($\dot{x}_{tt}$) is fed back and multiplied by a proportional gain term ($K_{p,TFB}$)

$$\beta_{TFB} = K_{p,TFB}\,\dot{x}_{tt} \tag{1.10}$$

where β_{TFB} is the tower-top FB pitch command. This is combined with that of the FB pitch controller, as illustrated in Fig. 1.7. Implementing this additional control loop results in improved stability, as well as reductions in surge motion and tower loading (see Fleming et al., 2014).

1.3.6 Feedforward pitch control with LIDAR

The drawback of traditional FB control is that the wind turbine controller reacts to the incoming wind once it has already impacted upon the turbine. Nacelle-mounted, forward-facing LIDAR can provide measurements of the incoming wind field, the characteristics of which can be used for FF turbine control. FF pitch control overcomes the pitfall of FB control by enabling turbines to actuate their pitch systems in advance of the wind's impact upon the rotor. LAC can enhance turbine performance through improved rotor speed regulation, and therefore more stable power capture (see Pitter et al., 2023; Schlipf & Kühn, 2008; N. Wang et al., 2011), thereby demonstrating the potential to overcome the increased rotor speed variation of FOWTs brought by detuning of the FB pitch controller gains, as discussed in Sect. 1.3.5. It is also capable of reducing the variation in the structural loadings and platform motions of FOWTs (see Guo & Schlipf, 2023; Schlipf et al., 2023, 2015). This work seeks to further quantify and explore the performance and load reduction benefits by investigating how they can lead to cost savings for FOW.

Implementing LIDAR measurements into FOWT pitch control typically entails the inclusion of an FF control loop, in combination with the existing generator speed and nacelle velocity FB control loops. An example block diagram of LIDAR-based FF-FB pitch control is given in Fig. 1.7. This is a simplified version of the block diagram shown later in Fig. 4.7, but illustrates the key additions pertinent to FF pitch control.

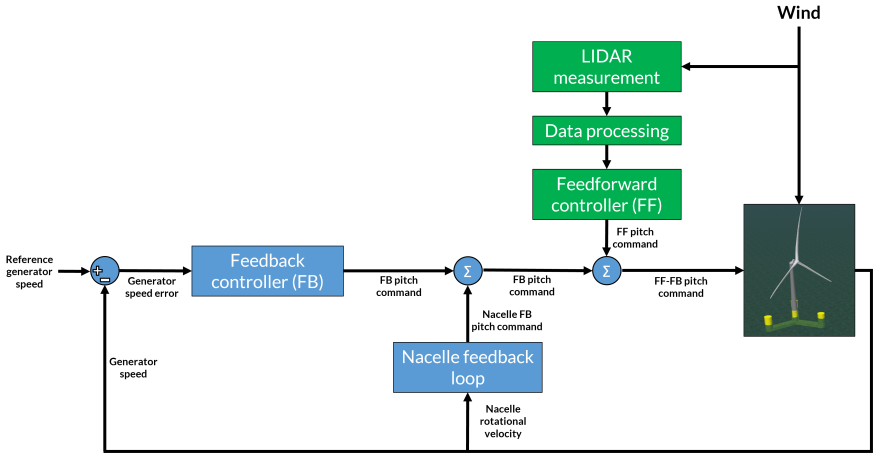


Figure 1.7: Flowchart illustrating an example LIDAR-based FF pitch controller. The green blocks denote those required for the LIDAR-assisted FF control loop, which add to the existing FB control loops.

As shown in Fig. 1.7 the FF control loop addition consists of three parts:

- LIDAR measurement, where the LIDAR measures the incoming wind at the desired location(s) ahead of the turbine.
- Data processing, which performs wind field reconstruction, filtering, and predicts how the wind field will evolve in the space between the point of measurement and the rotor, to ensure coherence between the measured and actual wind speed.
- The FF controller, which converts the wind speed measurement into a pitch command that is combined with the FB pitch commands.

Furthermore, additional loops can be incorporated for the use of FB or FF IPC, which can provide individual pitch commands to the blades.

Data processing

Key to the effective operation of FF control with LIDAR is the evaluation of the wind preview quality and its coherence to the actual wind speed. Coherence represents the strength of the correlation between the LIDAR-measured wind and the actual wind experienced by the turbine. Coherence analysis enables the determination of the cutoff frequency of the low-pass filter used to remove uncorrelated frequencies.

The purpose of a low-pass filter is to allow signals with a frequency lower than a certain cutoff frequency to pass through while attenuating signals with frequencies higher than the cutoff frequency. This is commonly used within LAC applications because the correlation of LIDAR-measured to actual wind speeds diminishes as signal frequency increases (see Schlipf, Schuler, et al., 2010). Therefore, low-pass filters are employed to remove noise and ensure that only data at frequencies with a strong correlation to the actual wind speed are considered for the formulation of FF control commands in order to prevent incorrect control action, which can be detrimental to the turbine's performance, loadings and motions.

The correlation of measured to actual wind can be quantified through assessment of a transfer function or magnitude squared coherence between the LIDAR-measured and turbine-based REWSs - ranging from 0 to 1. These provide a measure of the relationship between two signals in the frequency domain. They quantify the similarity in the frequency content of the signals and provide information about how well they are correlated at different frequencies. A value of 0 of the transfer function or magnitude squared coherence signifies no correlation and a value of 1 signifies that the LIDAR measurements perfectly capture the incoming REWS, meaning that disturbances can be rejected completely and optimal rotor speed control can be achieved. More detail on the literature findings of the importance of LIDAR measurement coherence is provided in Chapter 2, with detail on the determination of the cutoff frequency used within this work provided in Chapter 4.

Feedforward individual pitch control

LIDAR-assisted FF IPC (FFIPC) can be used to deliver individual pitch commands to the blades in advance of the wind's impact in order to better prepare them for the different wind profiles and loadings that they will be subjected to.

FFIPC has been studied using various implementations (see Dunne et al., 2010, 2012; Raach et al., 2014; Schlipf, Schuler, et al., 2010). Generally, these methods seek to overcome the variable loads on the rotor blades by separating the control problem into individual ones for each blade rather than one for the rotor as a whole. To achieve this, the blades can be controlled individually using independent identical FF controllers, as was shown by Dunne et al. (2012). FFIPC typically requires the use of multi-beam LIDAR, which can measure the incoming wind velocity at various points within the rotor-swept area. This is because, in order to provide tailored commands, the separated pitch controllers must only use the measured preview information that is relevant to the parts of the incoming wind field that will directly impact upon their respective blades. A systematic review of the previous studies focused on FFIPC is provided in Sect. 2.3.

1.3.7 O&M of floating offshore wind

O&M in the context of FOW concerns the management and maintenance of the FOWTs and their ancillaries during the operational phase of a project. The operational expenditure (OpEx) is defined as the costs associated with this phase of a project. These costs are split into direct and indirect costs. Direct costs are those which are directly impacted by the operation of the site and include aspects such as the cost of repair, transport, staff and other fixed costs such as insurance. Indirect costs include lost revenue resulting from downtime of the turbines, when they are not generating electricity.

Maintenance is required to repair and/or preserve the performance of wind turbines. Maintenance operations are typically categorised into planned or unplanned maintenance. Planned maintenance is the pre-determined number of hours or visits that will be taken to inspect and perform routine maintenance of the turbines each year. Unplanned maintenance is undertaken to address unexpected failures as they emerge throughout the lifetime of the wind farm.

The accessibility of wind farm sites is determined based upon both the weather conditions and the duration of suitable weather windows. The travel time to a site and the required weather window increase with the distance from shore. The weather window is the duration required to safely attend the site, complete the repair and then return to shore. As this duration increases, the access to the site decreases because it becomes more likely that weather conditions that render the vessels unsafe to be at sea will occur. This is likely to be a persistent issue for FOW farms that are situated far offshore.

The motions of FOWTs impact further upon the safe accessibility of FOW sites. The concept of workability has been explored within the literature by CoreWind (2020) and Scheu et al. (2018) as a means of qualifying how met-ocean conditions affect the accessibility of FOWTs. Workability is a measure of how safe it is for technicians to perform maintenance on an asset (such as a FOWT) in a number of sea states, which are a combination of significant wave height (H_s), $\bar{V}$ and peak spectral wave period (T_p) (see Jenkins et al., 2021). Certain sea states can render the motions of FOW platforms too significant for them to be accessed safely. Consequently, this can limit the access of personnel to FOWTs even in conditions in which the vessel can otherwise safely make the trip to site. Therefore, FOW platforms present further restrictions that compound the accessibility challenges of FOWTs for maintenance.

Maintenance strategies are typically governed by the failure rates of wind turbine components. The failure rate is defined as the number of failures of a component or subsystem per turbine per year. A failure of a turbine component or sub-system is defined as an event which impairs parts of or all of the normal operation of the wind turbine, and which results in the requirement for a repair or replacement of the component(s). Repair operations of wind turbines are typically characterised into minor repairs, major repairs and major component replacements, with (typically) decreasing frequencies of occurrence but increasing levels of cost and operational complexity, respectively (see Carroll et al., 2016).

Major component replacements (typically of the generators or blades) of FOWTs are particularly challenging because the deep depths of FOW sites are inaccessible to conventional major component replacement vessels (see Ahn et al., 2017), which are limited to 60 m depth. This gives rise to the need for alternative in situ methods such as floating-to-floating transfer, floating cranes and self-hoisting equipment, though these technologies are not yet at the maturity to perform major component replacements of large FOWTs. This leaves the only currently viable solution for major component replacements of FOWTs to be the tow-to-shore strategy. This requires disconnecting the FOWT from its mooring and cabling and towing it back to port, where the replacement operation is performed at the quayside. This is not only an expensive process because of the vessel hire and replacement costs, but also because of the significant lost revenue resulting from the typically extensive downtime of the turbine. This is further exacerbated by the aforementioned accessibility limits of the FOW sites and the large weather windows required to allow for the safe transport of the FOWT back to shore, which, in the worst-case scenario, can lead to the turbine being inactive for multiple months, such as in the winter, because the required weather window is not available within these periods. Therefore, the cost implications of major component replacements of FOWTs using the tow-to-shore strategy are likely to be particularly detrimental to the LCOE of FOW, at least until the alternative in situ solutions become viable.

Strategies such as LAC directly benefit the lifetime failure rates of components by reducing the loadings on the subsystems of FOWTs. This can result in lifetime extension and reduce the frequency and/or intensity of repairs. The decreased failure rates from LAC can also result in a lesser frequency of downtime events. This can allow for lower OpEx due to reducing the lost revenue and the number of transfers (and the associated vessel hire and personal costs) to the turbines. These failure rate reductions can also indirectly alleviate the impacts of the heightened accessibility challenges of FOW, as a decreased number of downtime events means that fewer transfers are required and so there are fewer opportunities for trips to be limited by the weather and floating platform accessibility constraints. Therefore, the use of LAC may ultimately result in reduced total length of downtime and lost revenue over the turbines' lifetimes.

1.3.8 Floating offshore wind turbine tower design

An important design aspect of FOWTs is that of the tower's first natural frequencies. For all wind turbines, these have to be carefully placed outside of the rotor rotational (1P) and blade passing (3P) frequency ranges to avoid resonance of the tower modes with the rotor's rotation. The towers of BFW turbines can be designed such that their natural frequencies lie within the interval between the 1P and 3P ranges (see Gaertner et al., 2020). This is often termed a soft-stiff tower design, as opposed to stiff-stiff design for towers whose frequencies lie above the 3P range. Soft-stiff towers are lighter than stiff-stiff towers and therefore benefit from lesser material requirements and capital expenditure (CAPEX).

Floating turbines tend to suffer from increased variability in the rotor rotational speed due to the modifications to the FB pitch controller, detailed in Sect. 1.3.5. This has impacts upon the tower design, as it broadens the rotor 1P and 3P constraint bands. The constraint bands are added to the 1P and 3P design bands (which are bands corresponding to the frequencies between the minimum and rated rotor rotational speeds of the turbine), as shown in Fig. 1.8, such that

$$\Omega_{r,const,min} = min[\bar{\Omega}_r(V) - 2\Omega_{r,SD}(V)] \tag{1.11a}$$

$$\Omega_{r,const,max} = max[\bar{\Omega}_r(V) + 2\Omega_{r,SD}(V)] \tag{1.11b}$$

where $\Omega_{r,const}$ is the design constraint rotor rotational frequency, $\bar{\Omega}_r(V)$ is the average rotor rotational speed at a wind speed and $\Omega_{r,SD}(V)$ is the standard deviation of the rotor rotational speed at a wind speed. This makes it challenging to design a soft-stiff tower with a natural frequency lying within the 1P and 3P interval of FOWTs, where wave loading is also apparent. This is further compounded by the fact that, as the turbines' rotor diameters increase, they are required to have a lower rated rotor speed to limit the tangential speed of the blade tip (V_{tip})

because they are inextricably linked by

$$V_{tip} = \frac{\pi D_r \Omega_r}{60} \tag{1.12}$$

where D_r is the rotor diameter and Ω_r is the rotor rotational speed. The blade tip speed is restricted in order to prevent erosion of the blades due to collisions with raindrops and sea spray (see UK Energy Research Centre, 2023).

The slower rotor speeds of larger turbines ultimately result in lower 1P and 3P frequency bands, meaning that the interval between them can overlap with frequencies (when modelled using the Joint North Sea Wave Observation Project [JONSWAP] spectrum) common for irregular ocean waves, as shown in Fig. 1.8. Additionally, due to the motions of the platform, FOWT towers are also subject to greater gravitational and inertial loads (see J. Jonkman & Matha, 2011), than those for BFW turbines, and so must be designed to withstand these. Therefore, the towers of the IEA 15 MW FOWTs have been designed such that their first natural frequencies lie above the 3P range (as in Allen et al., 2020), rendering them stiff-stiff and, consequently, more rigid, heavy and expensive than the towers of BFW turbines. These challenges were detailed within the reference documentation of the VolturnUS-S by Allen et al. (2020), where Fig. 1.8 indicates the aforementioned 1P and 3P design (frequencies between the minimum [5 rpm] and rated [7.56 rpm] rotational speeds of the rotor) and constraint bands, as well as the tower first natural frequencies.

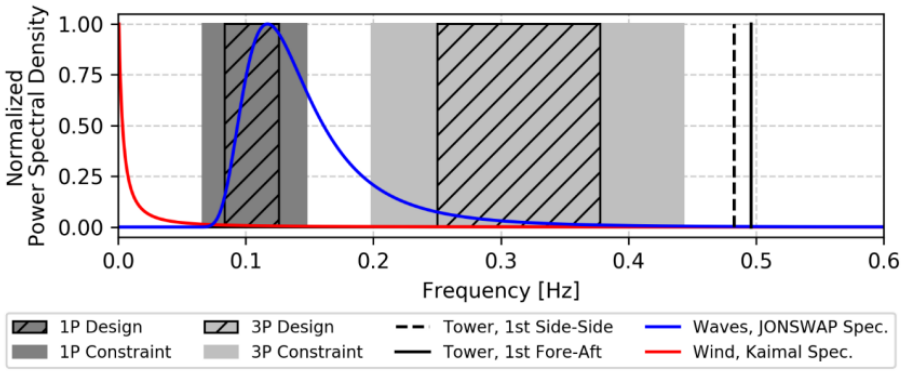


Figure 1.8: Rotor 1P and 3P design and constraint frequency ranges and tower first fore-aft and side-to-side natural frequencies of the VolturnUS-S, from Allen et al. (2020). Reprinted with permission from NREL.

With these considerations, it is possible to hypothesise that, given the tower loading reductions and the improved rotor speed regulation (thereby decreasing the size of the constraint bands shown in Fig. 1.8) delivered by LAC, it may be possible to design a lightweight tower whose first natural frequencies either remain above 3P or within the interval between 1P and 3P whilst experiencing similar levels of stress to the baseline design with FB-only control. This presents the opportunity to reduce the mass and cost of FOWT towers.

1.3.9 Wind farms and wake effects

Wind farms are clusters of wind turbines situated within a defined location. The arrangement of turbines within a wind farm and the separations between them are largely driven by a motivation to minimise wake effects and maximise power production. A wake refers to the region of disturbed airflow that forms downstream of a wind turbine as a result of the turbine's interaction with the incoming wind. The extraction of kinetic energy from the wind by the wind turbine causes a reduction in the wind velocity and a subsequent decrease in the air pressure immediately following the rotor. This results in the formation of a wake downstream of the turbine. The decrease in air pressure (p) following the decrease in wind velocity is due to the law of the conservation of energy, as governed by Bernoulli's principle, and represented by the actuator disk theory (Fig. 1.9, from Burton et al., 2011). The velocity and pressure gradually return to their ambient levels as the wake propagates downstream of the turbine and mixes with the surrounding air.

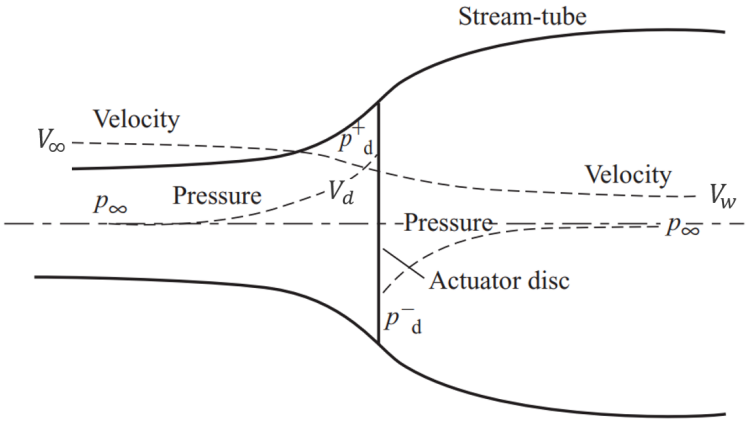


Figure 1.9: An energy extracting actuator disk and stream-tube, adapted from Burton et al. (2011). ∞ = free-stream, d = actuator disk and w = wake. Permission to reprint was granted by the Copyright Clearance Centre (CCC).

Wakes from upstream turbines are highly detrimental to downstream turbines because they result in higher turbulence intensities, which can lead to increased loads on turbine components. The increased turbulence of the wake occurs due to numerous aspects pertaining to the interaction of the rotor with the wind (see Vermeer et al., 2003). This includes the interaction of the lift and drag forces of the blades with the wind, as well as the vortices that propagate from the blade tips. Furthermore, turbulent eddies and vortices resulting from boundary layer separation at the blade surface at high angles of attack can also contribute to turbulence. Finally, the rotation of the blades introduces a rotation to the wind flow, which causes the air to spiral as it moves downstream of the turbine and which further adds to the complexity and turbulence of the wake.

As the upstream turbines extract power from the wind, this reduces the wind speed, resulting in a wake deficit - the difference between the ambient wind speed and the disturbed wind speed of the wake. Turbines are required to be separated appropriately in order to allow the wake deficit to reduce, such that the wind speed, pressure and turbulence return close to ambient before impacting the downstream turbine(s). The distance between turbines in the prevailing wind direction is recommended by Patel and Beik (2021) to be between 8 and 12 rotor diameters (RD). Should insufficient spacing be afforded, then downstream turbines will experience a lower longitudinal wind speed, which, if the ambient wind speed is close to the rated wind speed of the wind turbine, will result in reduced power capture of the downstream turbine as it will experience a below-rated wind speed.

A turbine with a stable rotor speed and power capture (such as one controlled with LAC) will benefit from more consistent lift and drag forces at its blades and more consistent wind velocity deficits, pressure drops and rotations of airflow within its wake than a turbine with more unstable rotor speed and power capture. This more uniform wind disturbance may lead to faster recovery of the wake and mean that downstream turbines experience a less significantly reduced wind speed, allowing them to capture more power and generate more revenue, thereby benefiting the LCOE of FOW.

1.4 Aim and objectives

Although previous studies have demonstrated LIDAR's promising potential for enhancing the performance of wind turbines with FF control (summarised within Chapter 2), further research opportunities were identified that sought to enhance these benefits and expand the field of LAC research. Primarily, the overarching aim became one of ascertaining the cost benefits of the implementation of LAC, which can ultimately lead to LCOE reductions of FOW.

This Section introduces the key objectives involved in achieving the research aim and how each was addressed. These include the implementation of FFIPC into large FOWTs, and quantifying the cost-saving benefits from an OpEx, via reduced component failure rates, CAPEX, via reduced tower mass, and revenue, via increased power capture at the array level. Therefore, the following aim and objectives of the research project were defined:

- **Aim:** To determine the techno-economic benefits of LIDAR-assisted feedforward pitch control (LAC) of floating offshore wind turbines (FOWTs).
- **Objectives:**
  - Assess the benefits of a novel LIDAR-assisted feedforward individual pitch control (FFIPC) strategy applied to large FOWTs.
  - Determine the floating offshore wind (FOW) operational expenditure (OpEx) reductions deliverable by component failure rate reductions brought by the implementation of LAC.

- Investigate the potential for lighter FOWT towers enabled by LAC, which can lead to capital expenditure (CAPEX) reductions.
- Assess the array-level benefits of LAC and how they can improve the power capture and revenue of downstream turbines via improved consistency and recovery of the upstream turbines' wakes.

The below provides more detail on the motivations and applied methodologies relevant to each of the objectives.

1.4.1 Feedforward individual pitch control of large floating offshore wind turbines

Although studies have reported the benefits of LIDAR-assisted FFIPC of smaller-scale FOWTs, namely Raach et al. (2014), research has not yet investigated the benefits that FFIPC can bring to large (15 MW+) FOWTs. Due to their significantly larger rotor-swept areas, with diameters approaching 300 m, these turbines are subject to greater variability in the wind profile acting upon them due to phenomena such as wind shear and turbulence. These variations can result in increased fluctuations in the loads experienced by the blades and, by extension, the rotor and turbine, as they rotate. This is further compounded by the greater deflections and bending moments that the blades are subjected to due to their greater lengths.

Given these concerns and IPC's ability to alleviate the cyclic loads on smaller turbines, it was deemed especially pertinent to examine the benefits that IPC can bring to larger (15 MW+) FOWTs. There was scope for a novel FFIPC approach to be curated which sought to be simple to execute, requiring fewer additional inputs from sensors than the traditional FBIPC strategy, while delivering improved benefits to the rotor speed regulation, the turbine loads and the platform motions of FOWTs. This curated strategy used data from a simulated multi-beam LIDAR to derive individual commands that were issued to the blades depending on their rotor azimuth angular positions. The FFIPC provided an additional refinement to the command delivered by the FF-FB collective pitch controller (FFCPC). The theory of the implemented approach is detailed in Sect. 4.4.

The work aimed to investigate the benefits of FF collective and individual pitch control of two 15 MW FOWTs - the University of Maine (UMaine)'s VolturnUS-S semi-submersible (see Allen et al., 2020) and CoreWind's WindCrete spar (see Mahfouz et al., 2020). This was because studies were yet to investigate the differences in the benefits brought by LAC to different 15 MW FOWT substructure designs. As previously explained in Sect. 1.3.3, the differences in the properties of these platform configurations (in terms of draft and ease of installation, for example) are expected to render each more suitable for different FOW farm sites. Therefore, it was deemed pertinent to predict the benefits and applicability (or lack thereof) of LAC for FOWTs using these different platform designs and, by extension, the FOW markets around the world that stand to benefit most from the implementation of LAC within FOWT control.

Simulations of FB, FFCPC, FFCPC+FBIPC and FFCPC+FFIPC of both FOWTs were performed, primarily under DLC 1.2 in above-rated wind speed conditions, as specified in IEC 61400-3-1:2019 (2019), because these encompass the normal conditions that the turbines are subjected to during operation and those which contribute the most to the fatigue of turbine components over their lifetimes, with results documented in Chapter 5. Differences in their baseline FB pitch control behaviour and platform motion behaviours and how these impacted upon the results deliverable from the LIDAR-assisted pitch control addition across the full above-rated wind spectrum was a primary focus.

1.4.2 O&M case study

As discussed in Sect. 1.3.7, the application of LAC shows promise for reducing the failure rates of components, which govern the number of downtime events and transfers to the turbines over their lifetimes. The loading reduction benefits recorded as a result of the LIDAR-assisted FFCPC+FFIPC implementation were converted to lifetime failure rate and damage equivalent load (DEL) reductions. Subsequently, these reduced failure rates of turbine subsystems were used within an O&M modelling tool to determine the OpEx cost savings via a case study of prospective FOW sites, with results documented in Chapter 6. The case study aimed to use the O&M model to determine where cost savings resulting from lifetime extension of components may stem from. These were expected to include reduced frequency and/or length of downtime and, therefore, reduced personnel and/or vessel requirement and/or reduced frequency or intensity of repairs. Furthermore, the study investigated how different wind farm distances from shore, wind speed distributions and the accessibility constraints imposed by the motions of FOWTs affected the reductions to the OpEx brought by LAC. The study was completed in collaboration with Dr. Jade McMorland, a (now graduated) PhD student (see McMorland, 2024) at the University of Strathclyde, who performed, and provided the results of, the O&M modelling simulations.

1.4.3 Tower design modifications

An investigation into the possibility for lightweight tower designs of 15 MW turbines employing LAC, leveraging the improved rotor speed regulation and reduced tower loads, is documented in Chapter 7. These benefits are especially relevant to large turbines due to the greater loadings applied to them as a result of their longer blade lengths, which incur requirements for greater structural robustness, and, therefore, material cost to withstand them. Consequently, it was deemed important to investigate the potential for structural optimisation of large FOWTs when using load-alleviating control strategies such as LAC, with the aim of reducing the material requirement and associated costs. Simulations of FB-only control and FFCPC+FFIPC of the VolturnUS-S FOWTs with modified tower geometry were performed within a FOWT structural design tool. A subset of DLCs (1.2, 1.6 and 6.1) were investigated to ensure that

the modified designs remained within operational and structural strength limits. The goal was to achieve a lightweight tower design that experienced similar loadings, stress and DELs when the FOWT was controlled using FFCPC+FFIPC compared to when the FOWT was controlled using the baseline FB-only controller with the baseline tower design. This could ultimately lead to CAPEX reductions via reduced steel mass of the tower.

1.4.4 Array-level benefits of LIDAR-assisted control

LAC has been shown to be able to reduce the variation in the rotor speed and power capture of FOWTs in above-rated wind speed conditions. These improvements could be leveraged to determine the benefits that they could lead to for the wake behaviour. It was hypothesised that the reduced variation in the energy extracted from the wind when using LAC, as well as the more consistent rotor rotational speed, may lead to a more consistent wake, which may expedite wake recovery. This may thereby reduce the turbulence in the wind impacting the downstream turbines, and lead to lesser wind velocity deficits such that the downstream turbines can benefit from improved power capture and, by extension, improved annual energy production (AEP) and revenue. Therefore, within Chapter 8, simulations of multiple turbines within an array were performed and the performance of upstream and downstream turbines using LAC were compared to when the same arrangement of turbines used FB-only control. This study investigated what, if any, benefits were brought to the wake behaviour and subsequent performance of downstream turbines as a result of LAC implementation within a wind farm.

1.4.5 Implementation of LIDAR simulation and LIDAR-assisted control within OpenFAST

In undertaking this research project, the need was identified to integrate LIDAR simulation and LAC within NREL's OpenFAST aero-elastic-servo-hydro wind turbine modelling software (NREL, 2024a, outlined in Appendix A.1) in order to address the aforementioned aim and objectives. Although similar work has been conducted by Schlipf et al. (2009) and Guo et al. (2022), their implementation involved the integration of a separate LIDAR simulation module to OpenFAST and an accompanying dynamic linked library (`.dll`) chain to interface with the Reference Open Source Controller (ROSCO, outlined in Appendix A.1.2) for LAC. However, for the purposes of this research, LIDAR simulation and LAC were required to be directly interfaced within the existing OpenFAST and ROSCO source code architecture to enable compatibility with the other simulation tools that were used to address the research objectives. Therefore, the source code of OpenFAST and ROSCO was modified to allow for the modelling of LIDAR simulation and LAC. The modifications made to OpenFAST to enable the simulation

of LIDAR measurement were approved by NREL and are present from v3.5 onwards (see NREL, 2024b). This marked a significant contribution to the community from this research project as it was the first time that a bespoke LIDAR simulator was made available within an official OpenFAST release. The source code modifications made will be outlined in Chapter 3.

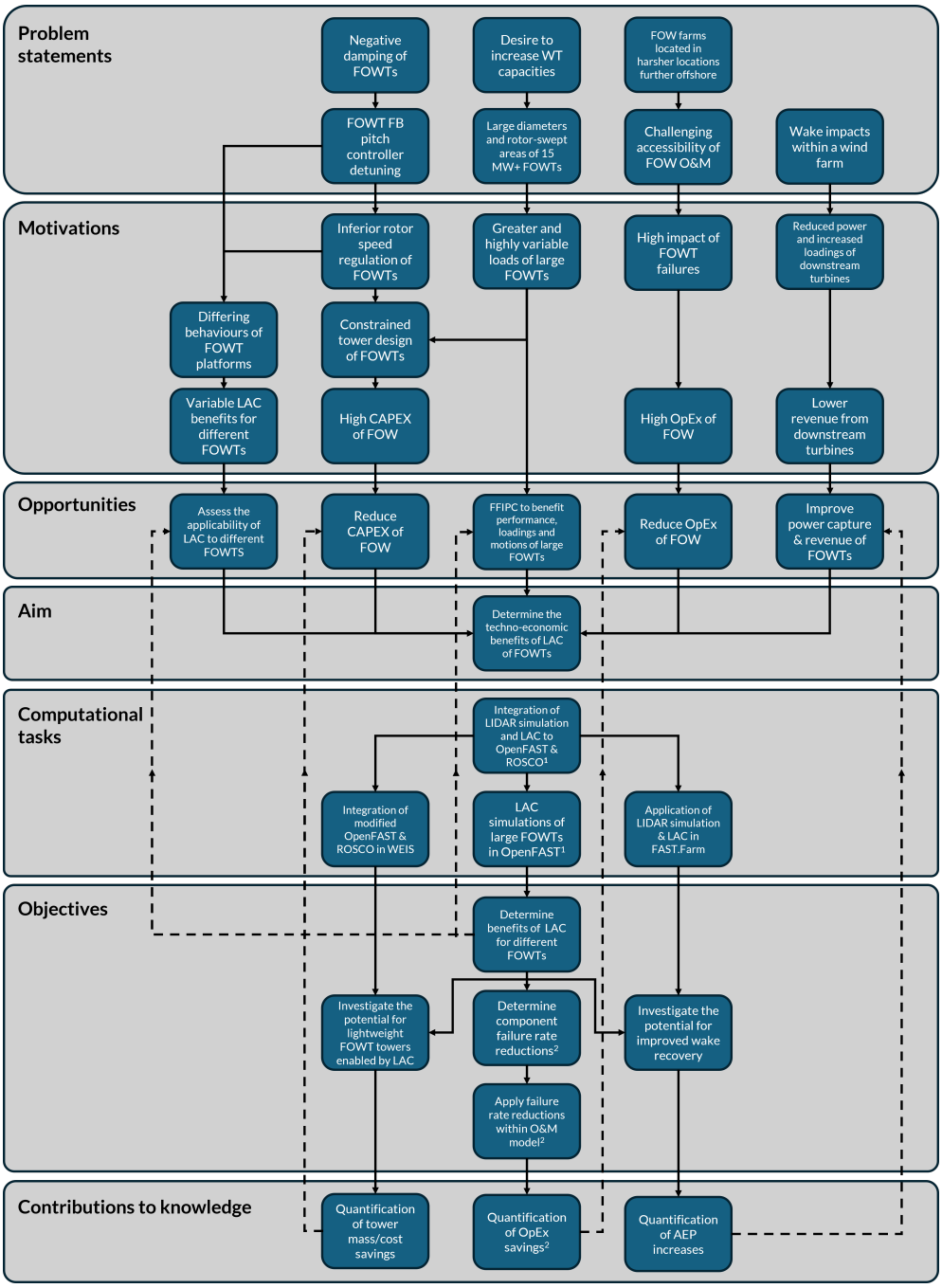
1.5 Simulation tools

Within the work performed in undertaking this research project, a number of simulation tools were utilised to address the aforementioned aim and objectives, which are fully introduced in Appendix A:

- NREL (2024b)'s OpenFAST aero-elastic-servo-hydro wind turbine modelling software was used to perform the FOWT simulations, with results outlined in Chapter 5. The turbine modelled was the IEA 15 MW reference turbine, defined by Gaertner et al. (2020), mounted on the University of Maine's (UMaine) VolturnUS-S semi-submersible, defined by Allen et al. (2020), and the WindCrete spar, defined by Mahfouz et al. (2020). It should be noted that subsequent mentions of the two floating platform designs (i.e. "VolturnUS-S" or "WindCrete") within this thesis refer to the complete FOWTs, including the IEA 15 MW reference turbine mounted upon them.
- StrathO&M was the O&M modelling tool employed by Dr. McMorland. This was used to translate the failure rate reductions of turbine components brought by the addition of LAC and obtained from the OpenFAST simulations to operational expenditure savings.
- NREL (2024c)'s Wind Energy and Integrated Servo-Control (WEIS) tool, which also employs NREL (2024d)'s Wind-Plant Integrated System Design and Engineering Model [WISDEM]), was used to perform simulations of the VolturnUS-S with towers of modified structural properties using LAC. The LIDAR simulation and LAC modifications made to OpenFAST and ROSCO were integrated within WEIS, as will be detailed in Chapter 3.
- NREL's FAST.Farm was utilised to investigate the array-level benefits of LAC on the performance of multiple VolturnUS-S FOWTs. This incorporated the modified OpenFAST and ROSCO versions within each turbine within the array.

1.6 Thesis flowchart

Figure 1.10 provides a synoptic breakdown of this thesis, highlighting the key problem statements, motivations, aim, objectives and contributions to knowledge.



[1]A. Russell, Collu, et al. (2024), [2]A. Russell, McMorland, et al. (2024)

Figure 1.10: Flowchart providing a synopsis of the key motivators, opportunities, objectives and contributions to knowledge of this thesis. The notes provide reference to the applicable academic journal articles that have been published or are under review.

1.7 Contributions to knowledge

The thesis aims to demonstrate the contributions to knowledge through the application of the LIDAR simulator and LIDAR-assisted controllers implemented within OpenFAST, ROSCO, WEIS and FAST.Farm. These applications aimed to quantify the potential cost savings and revenue increases achievable through large-scale implementation of LAC of FOWTs. This thesis presents the following points of novel and original work:

- A comparison of the benefits of a novel LIDAR-assisted FFIPC approach applied to two different 15 MW FOWT designs (Chapter 5).
- The application of reduced failure rates as a result of structural loading reductions delivered by LAC within an O&M modelling tool (Chapter 6) (courtesy of Dr. McMorland).
- The quantification of the potential OpEx reductions achievable as a result of the implementation of LAC within FOWTs (Chapter 6) (courtesy of Dr.McMorland).
- An assessment of how these cost benefits vary depending on the floating wind farm location and platform design due to differences in accessibility and weather conditions (Chapter 6) (courtesy of Dr. McMorland).
- Investigation of the benefit of LAC rotor speed regulation improvements and tower loading reductions to structural redesign of FOWT towers, quantifying the capacity for steel mass, cost and carbon footprint reductions without sacrificing the integrity of the tower (Chapter 7).
- The application and evaluation of LAC at the array level (Chapter 8).
- Assessment of the wake behaviour benefits brought by the addition of LAC and how this can lead to power capture improvements of downstream turbines (Chapter 8).

It is noteworthy that these contributions were delivered using the specific methods detailed in this work and the results may vary when simulating different wind farm sites, turbines or conditions. Nevertheless, the tools developed and methods outlined in this work enable the impacts of these variables to be ascertained.

1.7.1 Publications

Throughout the research project, paper publications were pursued in order to affirm the novelty of the research and the original contributions made within the project. Table 1.1 presents a summary of the conference proceedings papers and academic journal articles that have either been published or are under review.

Table 1.1: Summary of the published and under-review conference proceedings papers and academic journal articles that were prepared within this research project.

Ref.	Title	Year	Conference or Journal?	Status	Overview
A. Russell et al. (2022)	Review of LIDAR-assisted control for offshore wind turbine applications	2022	Conference	Published	A literature review of the published LAC papers of BFW and FOW turbines. This informed the literature review presented in Chapter 2.
A. Russell et al. (2023)	LIDAR-assisted feed-forward pitch control of 15 MW floating offshore wind turbines	2023	Conference	Published	A study focused on the implementation of LAC into two FOWTs, forming a foundation for the work presented in Chapter 5.
A. Russell, Collu, et al. (2024)	LIDAR-assisted feed-forward individual pitch control of a 15 MW floating offshore wind turbine	2024	Journal	Published	This paper documents the OpenFAST and ROSCO source code modifications outlined in Chapter 3, the novel FFIPC strategy presented in Chapter 4 and the results of the VolturnUS-S OpenFAST simulations documented in Chapter 5. The published paper can be found in Appendix G.
A. Russell, McMorland, et al. (2024)	The impact of LIDAR-assisted pitch control on floating offshore wind operational expenditure	2024	Journal	Published	The results of the O&M case study presented in Chapter 6 are documented in this paper.

1.8 Thesis structure

This thesis is structured as follows:

- Chapter 2 provides a literature review of topics pertinent to this work, including LIDAR-assisted FF collective and individual pitch control, O&M, FOWT tower design and array-level impacts.
- Chapter 3 details the modifications made to the source code of OpenFAST and ROSCO to enable LIDAR simulation and LAC, and subsequently to WEIS to enable simulations of FOWTs with modified tower properties.
- Chapter 4 outlines the FB and FF pitch control theory, including a novel FFIPC approach that was curated in this project.
- Chapter 5 presents the FF control simulation results of the two 15 MW FOWTs. The results of the LAC implementation within the VolturnUS-S were carried forward to the O&M case study (Chapter 6), the tower design case study (Chapter 7) and the array-level case study (Chapter 8).
- Chapter 9 provides a discussion of the results, offers conclusions drawn from the research and details opportunities for future work.

Chapter 2

Literature review

This chapter will outline the existing literature relevant to this research project. The findings of the literature review informed the formation of the aim and objectives introduced in Sect. 1.4 by enabling the identification of further areas of exploration, particularly from a FOW LCOE perspective. This review will outline the findings of previous works within the fields of LIDAR-assisted feedforward collective and individual pitch control of BFW and FOW turbines, O&M, FOWT tower design and array-level design and wake impacts.

2.1 LIDAR measurement coherence

The role of data processing of the nacelle-mounted, forward-facing LIDAR measurement data was introduced in Sect. 1.3.6 and this Section will discuss the literature findings on the subject. The importance of high correlation between the actual and LIDAR-measured wind for LAC was reported by Schlipf et al. (2014). In the event of low correlation, it was observed that the FF controller was not beneficial to the turbine. However, when high correlation occurred, the standard deviation of the rotor speed was reduced by 30% and 10% lower structural loads on the tower base were observed. A strong correlation was observed for frequencies below 0.1 Hz. Therefore, the FF control updates were required to be filtered by a low-pass filter with a cutoff frequency of 0.1 Hz to avoid incorrect control action from incorrect predictions.

The coherence between the actual and measured wind tends to decrease at frequencies above 0.1 Hz because of errors due to LIDAR measurement, wind evolution, and the turbine induction zone, as reported by Dunne and Pao (2013). Measurement quality has been found to strongly depend on the amount of wind evolution, whereas the induction zone has a lesser effect (see Simley et al., 2014). Other key shortcomings of LIDAR systems were discussed by Simley et al. (2014) and Held and Mann (2019), such as the spatial averaging of wind speeds along the LIDAR beam(s) and errors caused by the misalignment of the LIDAR's LOS with the mean wind direction.

To account for wind evolution, studies have often utilised Taylor's frozen turbulence hypothesis (see Held & Mann, 2019; Schlipf, Fischer, et al., 2010), which assumes that, under certain conditions, the turbulent structures move as frozen entities that are transported at the mean wind speed (see Taylor, 1938). In other words, it assumes that the wind velocity profile does not change in the interval between the wind's point of measurement by the LIDAR and its impact on the turbine. Its applicability for wind turbines was investigated through the use of LIDAR by Schlipf, Trabucchi, et al. (2011) and was found to be valid with a 90% accuracy for eddy length scales in the order of two rotor diameters. All of the error sources discussed are strongest at high frequencies, resulting in a measurement coherence function that drops from close to one at low frequencies (< 0.1 Hz) to close to zero at high frequencies, as reported by Schlipf, Schuler, et al. (2010). For FOWTs, the nacelle-mounted LIDAR is further impacted by the motions of the nacelle due to the platform's motions, leading to further errors in the measurements. Therefore, when assessing the coherence for FF control of FOWTs, Guo and Schlipf (2023) found that the coherence deteriorated at lower frequencies than for BFW turbines due to the additional motions brought by the floating platform, thereby requiring a lower cutoff frequency of the low-pass filter, which further restricted the FF control action.

2.2 Review of LIDAR-assisted feedforward collective pitch control

There have been numerous literary works focused on LIDAR-assisted collective pitch control for both BFW and FOW turbines. A summary table of the key LIDAR-assisted collective pitch control findings is shown in Table 2.1.

2.2.1 Bottom-fixed wind turbines

The majority of the studies that have been published in the field of LIDAR-assisted collective pitch control of wind turbines have been focused on BFW turbines. These studies underpin the work that authors have progressed on to publish for LAC of FOWTs. This Section will outline the findings of the key LIDAR-assisted collective pitch control studies that have been performed on BFW turbines, with details provided on the various control philosophies that have been employed.

An initial computational study of LAC was performed by Schlipf and Kühn (2008), who found that the standard deviation of the rotor speed could be reduced by 70-80% compared to FB-only control. This led to reductions in the fatigue and extreme loads on the tower, drive train and blades, whilst reducing the pitch rate (see Schlipf & Kühn, 2008). Their subsequent

Table 2.1: Summary of the reviewed LIDAR-assisted FFCPC papers and their key findings.

Ref.	Focus	Year	Turbine Rating	Sim. or Field-Test?	Key Findings
Bottom-Fixed Turbine					
Schlipf and Kühn (2008)	Initial FFCPC study	2008	5 MW	Simulation	FFCPC control reduced rotor speed standard deviation by 70-80% compared to FB-only, leading to reductions in the fatigue and extreme loads on the tower, drive train and blades, whilst reducing the pitch rate.
Schlipf, Bossanyi, et al. (2011)	FFCPC under extreme loads	2011	5 MW	Simulation	Compared to FB-only control, FFCPC led to an 80% reduction in the deviation of the tower base fore-aft bending moment for an extreme operating gust.
Schlipf et al. (2012)	FFCPC vs. NMPC	2012	5 MW	Simulation	The FFCPC was less computationally complex whilst yielding similar improvements to the NMPC.
N. Wang et al. (2011)	FX-RLS (Adaptive)	2011	600 kW	Simulation	FX-RLS gave improvements in tower and blade bending moments, rotor speed regulation and pitch rate.
Fu et al. (2023)	FFCPC using a spinner mounted LIDAR	2023	15 MW	Simulation	The benefits of using the optimised spinner-based configurations of both CW and pulsed LIDARs were much greater than the single-beam nacelle LIDAR, and similar to those of the four-beam nacelle LIDAR.
Guo, Schlipf, and Cheng (2023)	LAC under various turbulence characteristics	2023	5 MW	Simulation	Load reduction of the tower bending moment was found in all stability classes for wind speeds above 16 ms^{-1}, varying from 3.6% to 13.4%. The benefits of LAC by the Kaimal model were slightly different from the results obtained using the Mann model.
Scholbrock et al. (2013)	Field-testing of LAC	2013	600 kW	Field-test	Greater rejection of the wind disturbance at low frequencies compared to FB-only control. Reductions in the tower fore-aft bending, rotor torque, and collective flap bending from FF compared to FB-only control.
Schlipf et al. (2014)	Field-testing of LAC	2014	600 kW	Field-test	FFCPC control gave positive impacts upon rotor speed regulation as well as on tower, blade and shaft loads compared to FB-only control.
Pitter et al. (2023)	Design and retrofitting of LAC	2023	1.3 MW	Field-test	The FF controller delivered reductions to the DELs of the tower base-fore aft and blade root bending moments by 8.6% and 4.4%, respectively vs. FB.
Floating Turbine					
Navalkar et al. (2015)	FFCPC of a TLP FOWT	2015	5 MW	Simulation	Reduction of rotor speed variation by 45% and tower displacement by 40% vs. FB control under a wind speed step change. Under turbulent wind, the standard deviations of the generator speed and loads were reduced by 44% and 24%, respectively, vs. FB control.
Schlipf, Sandner, et al. (2013)	NMPC of a spar FOWT	2013	5 MW	Simulation	Compared to FB control, NMPC reduced tower loads and the rotor speed standard deviation by up to 90%.
Schlipf et al. (2015)	FFCPC of a spar FOWT	2015	5 MW	Simulation	Perfect wind preview - overshoot of the rotor speed reduced by 98.9%, the max. deviation from the static platform pitch angle by 93.7%, and the max. tower base fore-aft bending moment by 37.8%. Realistic wind preview - reductions in the loads on the tower base by 20%, shaft by 7% and blade roots by 9%, respectively.
Guo, Schlipf, Lemmer, et al. (2023)	FFCPC and multivariable (MV) FB control	2023	2 MW	Simulation	Combining MV FB and FFCPC reduced the standard deviations of the rotor speed, shaft torque, and blade root moments by 36%, 30%, and 15%, respectively.
Schlipf et al. (2023)	Tutorial of LAC of FOWTs	2023	15 MW	Simulation	Reduced rotor speed variation and platform pitch motion was shown. For large FOWTs, the wind-induced aerodynamic forces dominate the system motions.
Guo and Schlipf (2023)	LAC and MV FB control of large FOWTs	2023	15 MW	Simulation	For the large FOWT, it was found that a LIDAR focusing at a further distance can better estimate the REWS. With LAC, the tower lifetime was extended by 24.3 years, and the fairlead lifetime by 4.2 years.

modelling research into FFCPC focused on extreme loads (see Schlipf, Bossanyi, et al., 2011). For an extreme operating gust, when using FFCPC, a reduction of 80% (compared to FB-only control) in the deviation of the tower base fore-aft bending moment from the static value was achieved.

Model predictive control

Controller implementations have often utilised model predictive control (MPC) (see Laks et al., 2011; Schlipf, Schlipf, & Kühn, 2013). MPC primarily operates using FB control so does not require (although can still utilise) LIDAR, though is comparative to FF control because it uses a model to determine predictions of future process outputs. It is a holistic approach that can handle multiple-input, multiple-output (MIMO) systems and constraints. Its preview capability is analogous to that of FF control, and uses a "prediction horizon" to define the number of future time steps that it will consider (see MathWorks, 2018). However, the complexity of MPC is that it iteratively solves at each individual time step, optimising to minimise the error. Schlipf, Schlipf, and Kühn (2013) found that a non-linear model predictive controller (NMPC) showed load reductions of up to 50% for extreme gusts and 30% for lifetime fatigue loads compared to the FB-only controller without negative impacts on the overall energy production.

Although MPC/NMPC have been shown to be effective in controlling wind turbines and reducing loads, their significant drawback is their computational demand. Due to solving the model at each time step, the method requires a large amount of memory and a fast processor. Therefore, in a subsequent study, Schlipf et al. (2012) compared the performance of an FFCPC with the NMPC. Although the NMPC provided the opportunity for designing an "optimal" controller, the FF controller was less computationally complex whilst yielding similar performance improvements to the NMPC (see Schlipf et al., 2012).

Adaptive control - Filtered-x recursive least squares

In adaptive techniques, the control law can be updated at every time step according to the wind input conditions, meaning that they can reject disturbances across a wider range of wind turbine operating points. Recursive least squares is a technique that recurrently finds coefficients to minimise a weighted linear least squares cost function relating to the input signals (see D. Lee et al., 1981). Similarly to MPC, the computational demand of adaptive control is significant.

N. Wang et al. (2011) developed a filtered-x recursive least squares (FX-RLS) adaptive FFCPC for use in conjunction with LIDAR. The controller was able to improve the tower and blade bending moments, rotor speed regulation and pitch rate, with negligible reductions in power production, when compared to the baseline FB-only controller. The controller also performed promisingly when realistic (those used by real LIDAR equipment) 1 and 5 Hz update rates were used, as the standard deviation of the tower fore-aft bending moment was reduced by 20.6% and the blade flap-wise bending moment by 15.2%, compared to the baseline.

Spinner-mounted LIDAR

LAC can also be achieved through the use of a spinner-mounted LIDAR. Here, the LIDAR is placed within the spinner of the rotor, and so its beam(s) rotate with the rotor. It also has the advantage of not suffering from periodic blockage of the beams from the blades, as is the case with nacelle-mounted LIDAR. FFCPC enabled by a single-beam LIDAR mounted on the spinner of the IEA 15 MW turbine was modelled by Fu et al. (2023). Results of the spinner-based configurations of both CW and pulsed LIDARs were compared to those of a single-beam and four-beam nacelle-mounted LIDAR. The control benefits using the optimised spinner-based configurations were found to be much greater than the single-beam nacelle-mounted LIDAR, resulting in approximately 30% further reductions to the standard deviation of the rotor speed, compared to the baseline FB-only controller. The spinner-based configurations also yielded similar results to those of the four-beam nacelle-mounted LIDAR (see Fu et al., 2023).

Consideration of turbulence characteristics

In the studies discussed thus far, Taylor's frozen hypothesis was employed, which assumes that the turbulence measured by the LIDAR propagates unchanged from the point of measurement to the rotor. Recent work by Guo, Schlipf, and Cheng (2023) evaluated four-dimensional stochastic turbulence fields that included realistic longitudinal coherence (i.e. characterising the wind evolution over space and time), thus avoiding invocation of Taylor's frozen hypothesis. Through modelling of the NREL 5 MW reference wind turbine (see J. Jonkman et al., 2009) and a four-beam pulsed LIDAR system, it was found that the filter design of the FFCPC was not sensitive to the turbulence characteristics representative of the investigated atmospheric stability classes. Even with the removal of the Taylor's frozen hypothesis assumption, there were still significant reductions in the rotor speed variation (up to 40%), the tower fore-aft bending moment (up to 16.7%) and the power variation (up to 20%), compared to the baseline FB-only controller.

Field-testing of LIDAR-assisted pitch control

The studies discussed thus far have been computational studies of LAC of BFW turbines. However, there have been some published deployments of LIDAR on wind turbines in the field for the purpose of practical turbine control. Scholbrock et al. (2013) conducted field-testing of LIDAR-assisted FF control of NREL's Controls Advanced Research Turbine (CART) and found that this led to further rejection of the wind disturbance at low frequencies when compared to FB-only control. Schlipf et al. (2014) also performed field-testing of FFCPC using LIDAR and found positive impacts upon rotor speed regulation as well as on tower, blade and shaft loads.

Recent collaborative work between RES, ZX LIDARs, KK Wind Solutions and Sowento as part of the ReLACs project (see Pitter et al., 2023) focused on the retrofitting of a 1.3 MW turbine with the FFCPC developed by Schlipf et al. (2015). Within the study, the baseline and LIDAR-assisted load measurements were aligned with simulation predictions across a range of operational conditions. The FFCPC was found to deliver reductions to the DELs of the tower base-fore aft and blade root bending moments by 8.6% and 4.4% respectively compared to the baseline controller for wind speeds between 7 and 10 ms^{-1}, though this came at the expense of increased pitch activity, by 40%. The collaboration between Sowento and ZX LIDARs is currently continuing with a further initiative aiming to demonstrate the benefits of ZX CW LIDARs for LAC (see S. Henderson, 2023).

2.2.2 Floating offshore wind turbines

Although the control concepts are transferable between BFW and FOW, additional considerations have to be made for the negative damping caused by the coupling between the platform motions and the pitch control actuation, as described in Sect. 1.3.5. This imposes a bandwidth limitation on the pitch controller, which is often overcome by detuning the FB pitch controller. However, it is possible to bypass this limitation by using FF measurements provided by nacelle-mounted LIDAR to increase the bandwidth (see Navalkar et al., 2015; Schlipf et al., 2015).

Navalkar et al. (2015) designed a robust FFCPC for a 5 MW turbine on a TLP, assuming perfect LIDAR measurements. They noted that the high-bandwidth speed control may increase modal vibrations and thus aimed to design a controller to regulate generator speed without exciting the turbine modes. The FFCPC was designed under the assumption that the model was a perfect representation of the system and that the control effort could be arbitrarily large. It was noted by Navalkar et al. (2015) that both of these assumptions have limited validity. Under a wind speed step change, the robust FFCPC reduced extreme speed variation by 45% and the extreme tower displacement by 40%, when compared to FB-only control.

Navalkar et al. (2015) also investigated the performance under a turbulent wind field and with a wave height of 5 m. In this case, the standard deviation of the generator speed was reduced by 44% and the standard deviation of the loads was reduced by 24%, compared to FB-only control.

Schlipf, Schlipf, and Kühn (2013) and Schlipf et al. (2012) furthered their research into NMPC pitch control of BFW turbines by performing studies on FOWTs (see Raach et al., 2014; Schlipf, Sandner, et al., 2013). Schlipf, Sandner, et al. (2013) simulated their NMPC algorithm on a 5 MW spar under irregular waves and under extreme and turbulent wind conditions. Tower loads were reduced as well as the rotor speed standard deviation by up to 90%. It was concluded that this high-performing, computationally expensive controller can act as a benchmark for the development and comparison of less complex controllers.

Schlipf et al. (2015) also simulated the implementation of LIDAR on a 5 MW spar FOWT for FFCPC. They implemented the FFCPC that they had developed for onshore turbines (see Schlipf, Bossanyi, et al., 2011; Schlipf & Kühn, 2008) and subsequently applied in field-testing campaigns (see Schlipf et al., 2014). This FFCPC approach was also used within the work for this thesis and its theory is fully detailed in Sect. 4.2. The controller was adapted for FOWTs through the inclusion of data processing and adaptive filtering stages. Tests were first performed assuming perfect LIDAR wind preview, which resulted in significant improvements in turbine performance. For example, under a wind speed step change, the overshoot of the rotor speed was reduced by 98.9%, the maximum deviation from the static platform pitch angle by 93.7%, and the maximum tower base fore-aft bending moment by 37.8%, compared to the FB-only controller, as shown in Fig. 2.1.

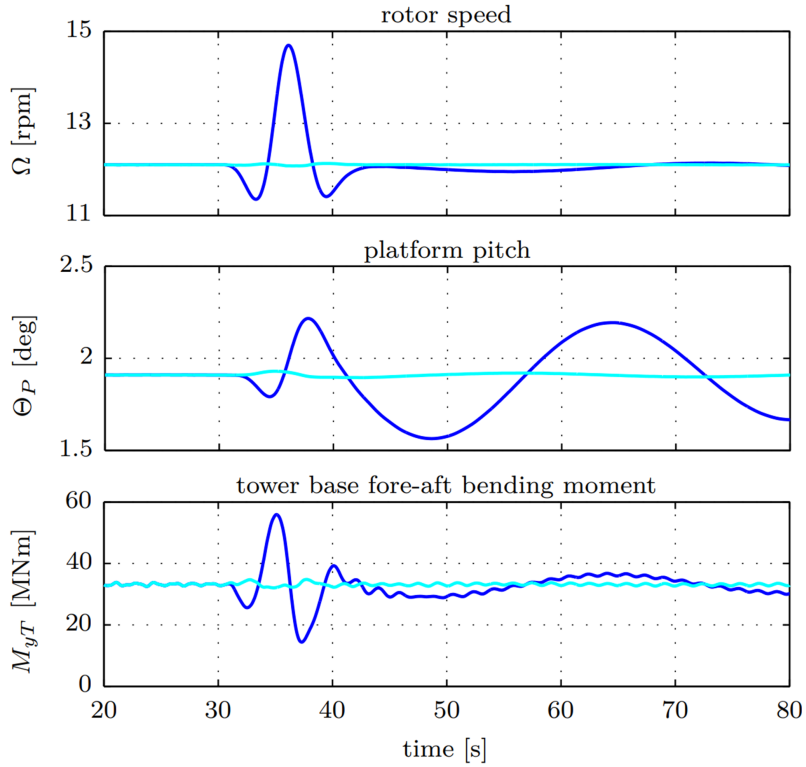


Figure 2.1: Findings from Schlipf et al. (2015), simulating FFCPC (light blue) and FB-only control (dark blue) assuming perfect LIDAR wind preview of a wind speed step change. Reprinted with permission from the International Society of Offshore and Polar Engineers (ISOPE).

Subsequently, simulations were performed under realistic conditions using a LIDAR simulator, developed by Schlipf et al. (2009), to scan the wind field used in the numerical simulations. For applicability to FOWTs, the simulator was also extended to include the platform motions in order to realistically reproduce LIDAR measurements from a FOWT through calculation of the LIDAR's position, velocity and inclination based on the six platform and four tower modes, and

their derivatives (see Schlipf et al., 2015). Following this, a motion compensation algorithm was developed, which sought to exclude the velocities of the LIDAR due to the motions from the wind velocity measurements. They reported that a good agreement was found between the REWS from the wind field and its filtered, time-shifted estimate from the LIDAR. Even with the platform motions included, the rotor speed variation was still significantly reduced, in addition to the platform motions and the tower base bending moment, compared to the baseline FB-only controller, as shown in Fig. 2.2. Reductions in the loads on the tower base of 20% were achieved as well as 7% and 9% reductions on the shaft and blade root loads, respectively, compared to the baseline FB-only controller.

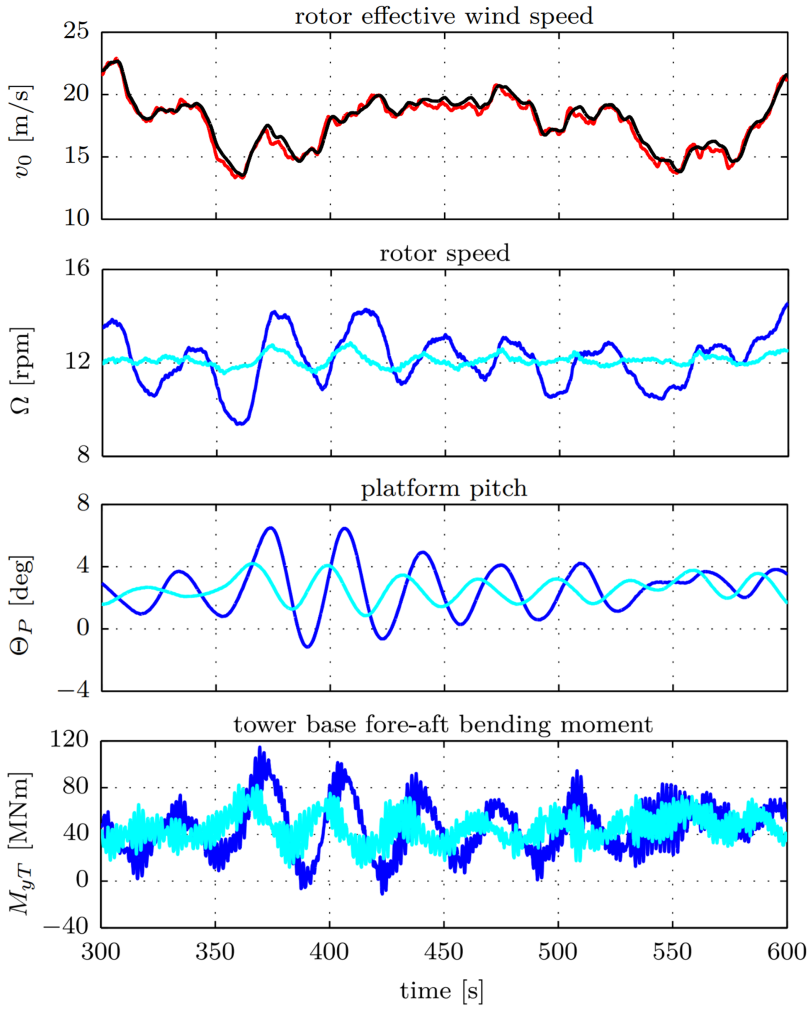


Figure 2.2: Findings from Schlipf et al. (2015), simulating FFCPC (light blue) and FB-only (dark blue) control with realistic LIDAR wind preview of a turbulent wind field. Top: rotor-effective wind speed (red) and its filtered LIDAR estimate (black). Reprinted with permission from ISOPE.

Guo, Schlipf, Lemmer, et al. (2023) assessed the performance of combined FFCPC and multi-variable FB (MVFB) pitch control, which used feedback of the platform pitching rate in addition to the generator speed feedback when determining the blade pitch command. In this study, a FOWT was modelled based on the FLOATGEN floating wind demonstration turbine, consisting of a 2 MW wind turbine mounted on the Damping Pool barge-type platform designed by BW Ideol (2018). The MVFB combined with the FFCPC showed potential for load reductions. Under the three investigated wave conditions by Guo, Schlipf, Lemmer, et al. (2023), the standard deviations of rotor speed, shaft torque, and blade root moment were decreased by more than 36%, 30%, and 15%, respectively, compared to the baseline FB-only controller.

Studies of 15 MW floating offshore wind turbines

Studies have recently progressed to large, commercial-scale FOWTs, primarily the IEA 15 MW reference turbine (see Gaertner et al., 2020) mounted on the UMaine VolturnUS-S semi-submersible (see Allen et al., 2020). Guo and Schlipf (2023) paired an FFCPC with an MVFB controller, as in their previous study on a 2 MW FOWT (see Guo, Schlipf, Lemmer, et al., 2023). In this study, it was reported that, for the large FOWT, the LIDAR had to be focused at further distances upstream of the turbine than the equivalent BFW turbine because the coherence between the LIDAR-measured and turbine experienced REWS diminished at lower frequencies due to the errors induced by the platform's motions. This meant that the cutoff frequencies of the low-pass filter were less, which incurred greater time delays, meaning that more lead time was required to allow for appropriate preview time of the FF commands. A notch filter, which acts by attenuating signals at a specified frequency, was also necessary for FOWTs to avoid the FFCPC conflicting with the MVFB controller. Results indicated significant reductions in the maximum values of blade tip deflection and platform pitch observed using combined LIDAR-assisted and MVFB pitch control. The tower lifetime was extended by 24.3 years, and the fairlead lifetime by 4.2 years. However, these benefits came at the expense of increases in blade pitch activity at very high wind speeds, which has the potential to lead to more damage to the gears and bearings of the pitch actuators.

2.3 Review of feedback and feedforward individual pitch control

IPC aims to mitigate the cyclic loads experienced by wind turbines due to variations in the wind profile across the rotor-swept area caused by wind shear, wind veer and turbulence. The concept of FBIPC was first explored by Bossanyi (2003), and applied to FOWTs by Namik and Stol (2010), who reported that the IPC reduced power fluctuations, the platform rolling rate and the platform pitching rate by 44%, 39% and 43%, respectively, relative to the baseline FB collective pitch controller. Within this strategy, the IPC requires the three out-of-plane blade

root bending moments and the rotor azimuth as FB inputs to allow for the derivation of a pitch command for each blade, which adds to the command provided by the FB collective pitch controller. The FBIPC was also used within this work, and its operation is fully detailed in Sect. 4.3.

Alternatively, FFIPC is able to deliver individual pitch commands in advance of the wind's impact and has been studied using various implementations (see Dunne et al., 2010, 2012; Raach et al., 2014; Schlipf, Schuler, et al., 2010).

Schlipf, Schuler, et al. (2010) used a H_∞ controller optimisation approach to design an FBIPC using a linear time-invariant (LTI) model. The LTI model was derived from an azimuth-dependent non-linear model considering the rotor motion, the first flap-wise bending modes of each blade and the first tower fore-aft bending mode. These were classed as the most relevant modes for the control problem. They also implemented a LIDAR-enabled cyclic FFIPC designed to accommodate for disturbances arising from horizontal and vertical wind shear. The H_∞ and cyclic FFIPC controllers were effective in decreasing the standard deviations of the blade root bending moments by 15% and 28%, respectively, in comparison to the baseline controller. The cyclic FFIPC was also able to decrease the standard deviations of the tower base fore-after bending moment (by 12%) and the rotor speed (by 52%), relative to the baseline FB-only controller. Both controllers delivered these benefits at the expense of increased pitch rate over the baseline FB-only controller (by 276% for the H_∞ FBIPC and 173% for the cyclic FFIPC).

Raach et al. (2014) used an NMPC to deliver individual blade FF commands to a 5 MW spar FOWT with the motivation of reducing blade loads by minimising the yawing and pitching moments of the rotor. The NMPC with FFIPC showed promising results in damping the platform pitch movement and reducing standard deviations of the rotor speed and electrical power, by up to 81.7%, compared to the baseline FB-only controller. Furthermore, it was found to significantly reduce the DELs of the flap-wise and out-of-plane blade bending moments, by up to 18.8%, compared to the baseline FB-only controller.

Dunne et al. (2010) developed five FFIPC designs and compared them to an FBIPC. The FFIPCs used separate iterations of the designed FFCPCs, one for each blade. The FFCPCs were designed in two ways, yielding the five FFIPC controllers. The first strategy used model-inverse FF control, designed to cancel the effect of wind disturbances on rotor speed, yielding two of the FFIPCs. Model-inverse control requires a linearised model of the turbine, which is then inverted such that the controller can calculate the required control input for a given desired output. These controllers used instantaneous wind speed measurements at the blades. The other strategy used a gain-scheduled shaped compensator FF controller, which utilised advanced wind measurements up to five seconds ahead of the blades. The three FFIPCs curated using this approach each considered different lengths of preview time (5, 10/3 and 5/3 s). The FFIPCs were evaluated on a 5 MW wind turbine model using realistic wind fields. Improvements were found when using the three gain-scheduled shaped compensator

FFIPCs (rather than the two model-inverse FFIPCs) due to their wind preview measurement functionality. The gain-scheduled shaped compensator FFIPCs allowed for improved rotor speed and power regulation at the expense of increased pitch rates compared to the FBIPC baseline. They also resulted in blade root and tower fore-aft load reductions between 5% and 15%, compared to the baseline controller, while slightly increasing the tower side-to-side motions (see Dunne, Simley, & Pao, 2011).

Dunne, Pao, et al. (2011) continued their work by investigating a non-causal series expansion (NCSE) approach, which is a model-inverse method that was not tested in their previous study (Dunne et al., 2010). The NCSE method was designed to directly regulate the blade root bending moment output. This was in contrast to their previously studied model-inverse controllers, which regulated the generator speed. Other differences between the NCSE and previous model-inverse controllers included that the previous controllers were based on a model linearised about an operating point of 18 ms^{-1} wind speed and were gain scheduled, while this NCSE control was linearised at 13 ms^{-1} wind speed to better match simulation conditions and was not gain scheduled. Finally, the wind speed measurements used by the FF control in their previous study (Dunne et al., 2010) were means of five points taken at the hub and outer edges of the rotor plane. Dunne, Pao, et al. (2011) instead used a mean of three points, one at 75% span of each blade. In the FFIPC mode, three copies of the FF controller were used, one controlling each blade. Three perfect-point wind speed measurements were taken, 2.5 seconds ahead of each blade, and these were fed into their respective FF controllers. This FFIPC was combined with the baseline FBIPC. The new FFIPC NCSE controller was found to not be an improvement upon the combined shaped compensator FFIPCs studied in Dunne et al. (2010) and resulted in increases to the tower-top DEL and the root mean squared (RMS) nacelle velocities in the x and y directions, compared to the FBIPC on its own.

Subsequently, Dunne, Pao, et al. (2011) designed an optimised finite impulse response (FIR) filter FF controller. The coefficients of the FIR filter were optimised by using a genetic algorithm, which converged on the set of coefficients that gave the best performance. The performance assessment was based on fatigue load reduction (blade and tower DELs), nacelle accelerations, RMS pitch rate, peak rotor speed and mean power achieved in a simulation of the non-linear turbine in above-rated wind conditions. The genetic algorithm optimisation approach was advantageous because it optimised the controller directly based on the non-linear turbine model in a turbulent wind field, with LIDAR errors modelled. This was in contrast to the FF controllers designed based on a linearised turbine model with uniform wind fields, which were often found to perform poorly in non-linear turbulent simulations. The peak of

the FIR response was designed to occur at a time close to the wind's impact upon the rotor. Therefore, for the shaped FFIPC with five seconds of preview time, the pulse response began and ended five seconds before and after time zero (when the wind impacted upon the turbine), as shown in Fig. 2.3.

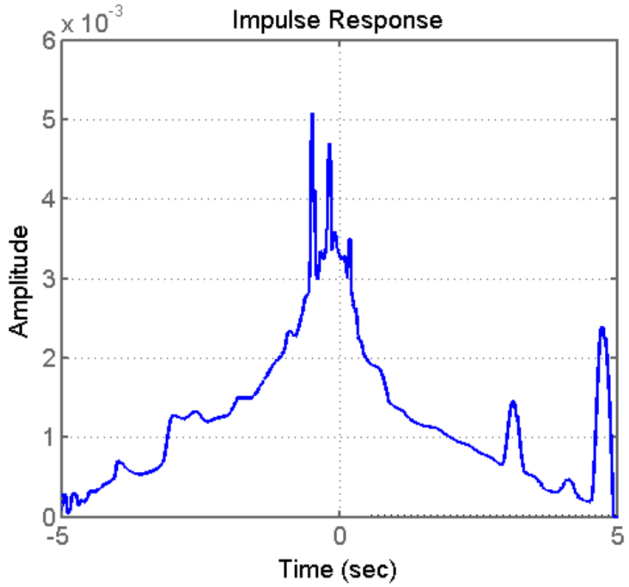


Figure 2.3: Response of the FIR FF controller designed by Dunne, Pao, et al. (2011). Reprinted with permission from the American Institute of Aeronautics and Astronautics, Inc.

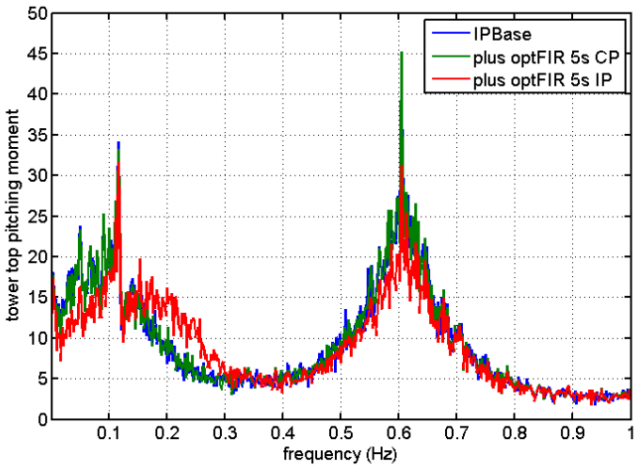
As in Dunne et al.'s other FFIPC controllers, three copies of the FIR FF controller were implemented, one controlling each blade. Three wind speed measurements were taken, five seconds ahead of each blade at 75% span, and fed into their respective FF controllers. This FFIPC was then combined with the FBIPC.

Dunne, Pao, et al. (2011) stated that taking measurements at each blade provides an estimation of the blade-effective wind speed. It was stated that the coherence between the single measurement ahead of each blade and each blade's actual effective wind speed should be greater than the coherence between each of the three-point measurements and the rotor-effective wind speed. This was later investigated by Simley and Pao (2013), who found that a high coherence could be maintained until higher frequencies when using a rotating LIDAR to estimate blade-effective wind speeds than when using stationary measurements and stationary blade-effective wind speeds. This can allow for higher cutoff frequencies of the low-pass filters applied to the LIDAR's wind speed measurements, which can allow for less restriction of the FF control action.

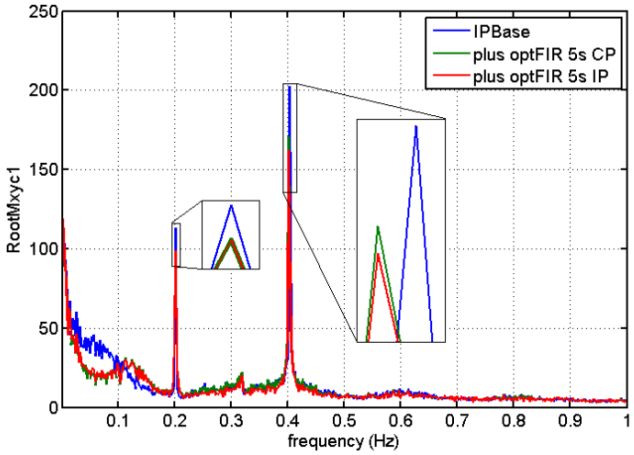
While the previous controllers used wind speed measurements that were perfect point measurements, the wind speed measurements used by Dunne, Simley, and Pao (2011)'s FIR FF controller incorporated a CW LIDAR model. The LIDAR beams were also assumed to rotate with the blades. The FIR FFIPC was found to reduce loads on different parts of the

turbine by varying amounts, with an average 5% load reduction overall, while maintaining rated power and rotor speed and without significantly increasing pitch rate, compared to the FBIPC baseline. Dunne et al. (2012) also found that the individual pitch variation of the FIR FF controller reduced blade root and tower-top DELs and RMS nacelle fore-aft acceleration more significantly (compared to the baseline FB-only controller) than the collective pitch variation.

Dunne et al. (2012) further investigated the FIR FFIPC in the frequency domain. They found that the tower-top pitching moment was reduced by the FFIPC at frequencies below 0.1 Hz, increased between 0.15-0.3 Hz, and reduced again between 0.4-0.7 Hz, when compared to the FFCPC, as shown in Fig. 2.4a.



(a) Tower-top pitching moment



(b) Blade root bending moment

Figure 2.4: Tower top fore-aft pitching moment and blade root bending moment as a function of frequency, from Dunne et al. (2012). Reprinted with permission from the American Institute of Aeronautics and Astronautics, Inc.

The FFIPC pitch action was also found to reduce blade root loads in frequencies between 0.23-0.45 Hz (Fig. 2.4b). Dunne et al. (2012) concluded that the benefit to the blade root loads did not occur at frequencies corresponding to 1P (0.2 Hz) because the FBIPC was already reducing the loads significantly at these frequencies, leaving little further room for improvement for the FFIPC. The frequency plots by Dunne et al. (2012) confirmed that the FFIPC load reductions occurred at the 1P and 2P frequencies and found that, in frequencies below 0.1 Hz, where wind excitation is apparent, the FFIPC reduced blade pitch actuation, generator speed error, and blade loads simultaneously, compared to the baseline FBIPC.

Studies are yet to investigate the benefits of LIDAR-assisted FFIPC of large (15 MW+) FOWTs, whose rotor diameters are approaching 300 m. These large diameters will give rise to greater variations in the wind speed across the rotor-swept area, leading to significant variations in the loadings on the turbine's structures, and so it is pertinent to assess the benefits that FFIPC can bring to these large turbines.

2.4 O&M

Maintenance of offshore wind turbines is performed to either prevent or repair the failure of turbine components. The most common causes of failure of wind turbines and components are relatively poorly understood within the public domain. However, work has been conducted by Carroll et al. (2016) to quantify the failure rates, technicians required, time-to-repair and cost of components for different turbine subsystems, based on data from approximately 350 installed offshore wind turbines. Key findings of their work indicated that the average failure rate of an offshore wind turbine is approximately ten failures per turbine per year by a wind farm's third operational year. Approximately 80% of the recorded repairs were minor repairs while 17.5% were major repairs and 2.5% were major replacements. The study also found that the pitch/hydraulic systems and the generators of offshore wind turbines fail most often. As with onshore wind, a trend was found by Carroll et al. (2016) that average failure rates increased with rising mean wind speeds. This will be increasingly pertinent for FOWTs located in areas further from shore, which generally experience greater mean wind speeds than closer-to-shore locations.

The failure rates and the most common causes of failure of FOWTs have not yet been fully characterised based on real data from turbines due to the relative scarcity of FOWT installed capacity. However, work conducted by Li and Soares (2022) aimed to implement a failure correction model to estimate the failure rates of FOWTs based on data from onshore and bottom-fixed offshore turbines, and estimated that the failure rates of FOWTs are 26% higher than those for onshore turbines. This, coupled with the heightened accessibility and repair challenges of FOWTs, means that maintaining them is likely to prove to be a substantial expense for FOW.

As FOW sites move into deeper waters, the average distance to shore is also expected to increase. The average distance to shore (of mainland Scotland) of the proposed ScotWind leasing round projects (see Crown Estate Scotland, 2022) exceeds 70 km, with this further increasing for the FOW sites, while the average distance to shore of the current operational offshore wind farms is 18.7 km, per Díaz and Soares (2020). The increase in distance to shore is expected to have a significant impact on the accessibility of the sites. Rowell et al. (2022) compared the accessibility of future ScotWind sites against that of currently operating wind farms (e.g. Moray East). This work used H_s as the limiting factor, which was kept consistent for both FOW and BFW sites. The results indicated that accessibility for FOW sites will be challenging, particularly for those using a crew transfer vessel (CTV) based approach.

In addition to the existing access limits, the wave-induced motion of the turbine could potentially further limit turbine access, as was introduced in Sect.1.3.7. Turbine motion is frequently referenced as a major obstacle in FOW O&M procedures (see Jenkins et al., 2021). Recent studies by C. Wang et al. (2010), Bashetty and Ozcelik (2021) and Otter et al. (2022) have analysed the potential impact of turbine motion within a number of sea states. Most of these studies have concentrated on ascertaining turbine motion under severe weather circumstances, aiming to establish the platform survivability and the impact of turbine motion on degradation and fatigue. However, the exploration of turbine motion for maintenance-related oceanic conditions (e.g. situations where significant wave heights are below 4 m, see Sperstad et al., 2017) remains relatively constrained.

When modelling O&M processes of FOWTs, it is crucial that these additional motions are considered in order to ensure the health and safety of technicians. As introduced in Sect.1.3.7, workability is a measure of how safe it is for technicians to perform maintenance on an asset in a number of sea states, which are a combination of H_s, $\bar{V}$ and T_p (see Jenkins et al., 2021). The workability index (WI), introduced by Scheu et al. (2018), is a term used to qualify the relative accessibility of a FOWT, depending on these met-ocean properties. This is based upon the ratio of the number of hours that are "workable" ($T_{wc,i}$), whereby conditions are favourable enough such that the turbine can be accessed safely, compared to the total number of hours (T_{t}).

$$WI = \frac{\sum T_{wc_i}}{T_t} \tag{2.1}$$

A WI of 1 indicates no impairment of working conditions due to FOWT motion and a value of 0 means that work is not possible under the respective conditions. Time-workable conditions are determined based on the specific response amplitude operator (RAO) of the selected offshore wind structure, under a specific set of H_s and T_p values. An acceptable motion threshold is

then imposed based on the existing oil and gas threshold from Nielsen (1987). This can then be normalised to allow for comparisons of WIs of different floating platforms. Currently, there is no definitive guidance to determine an acceptable WI for offshore wind operations. This introduces a trade-off between site accessibility and technician health and safety.

Scheu et al. (2018) provided WI results for a 10 MW wind turbine mounted on four different platform designs (A, B, C & D). Designs A-D included a spar, a TLP, a semi-submersible and a barge design. However, the specific design details of A-D were not disclosed. Workability was found to be a slight limiting factor for sea states with a WI of $\leq$ 90% and had a significant influence for WI < 60%. The concept of WI for specific platform designs was also explored within the CoreWind (2020) project, considering two specific floater designs, ActiveFloat (semi-submersible) and WindCrete (spar). Both platforms were connected to the 15 MW reference turbine, from Gaertner et al. (2020). While existing guidance addresses safe human exposure, specific limits for a safe WI in offshore operations are lacking, though Scheu et al. (2018) categorised WI impact > 90% as slight, 60-90% as significant and < 60% as major.

2.5 Floating offshore wind turbine tower design

Recent work by Papi et al. (2023) has sought to investigate the feasibility of lightweight towers of 15 MW FOWTs. They tested numerous lightweight designs, aiming to place the tower frequency within the interval between 1P and 3P. Results showed that the lightweight towers increased the nacelle acceleration and displacement with increasing degrees of magnitude as their natural frequencies reduced as the towers became more lightweight. The maximum stress on the lightweight towers also increased as the tower's natural frequency decreased (see Papi et al., 2023). The tower design case study in Chapter 7 will follow similar methodologies to those outlined by Papi et al. (2023), to investigate whether the viability of lightweight towers improves through the addition of LAC. Additionally, the case study extends these methodologies via an investigation into the impacts of wind-wave misalignment on the stress of the baseline and lightweight tower designs.

2.6 Array-level simulation

Within the established literature, the optimisation of wind turbine arrays is led by a desire to maximise annual energy production while limiting the impacts of wakes on downstream turbines. The impacts of wakes on a downstream turbine was studied by S. Lee et al. (2012), who found that, when spaced at 7 RD, the downstream turbine yielded higher DELs compared to the upstream turbine, indicating that the turbulent wakes from the upstream turbines can have a significant impact on downstream turbines.

Control strategies have also been employed in order to limit the effects of wakes on downstream turbines. Traditionally, these are focused on the use of yaw control for wake steering. Wake steering is the act of deliberately yawing a wind turbine such that its rotor is misaligned with respect to the wind direction. This ensures that its wake is deflected and avoids impacting the downstream turbine. Wake steering has shown benefits for mitigating the wake losses of downstream turbines, thereby improving the annual energy production of a wind farm. Fleming et al. (2015) simulated a two-turbine wind farm and found that, through wake steering, the net power of the two turbines was increased when the upstream turbine applied an intentional yaw misalignment. The simulated benefits were supported by the results of field-testing of wake steering applied to a commercial wind farm (see Fleming et al., 2017).

One application of nacelle-mounted, forward-facing LIDAR for increasing power capture is by enabling FF yaw control. This is used to ensure that the rotor is aligned with the wind direction and has been found to demonstrate an ability to offer a 1% expected increase in annual energy output when compared to measurements provided by a nacelle-based anemometer (see Schlipf, Kapp, et al., 2011). However, the prospect of LAC's ability to reduce wake impacts and power capture of downstream turbines is yet to be addressed and will be investigated within Chapter 8.

Chapter 3

OpenFAST, ROSCO and WEIS source code modifications

This Chapter provides an overview of the OpenFAST and ROSCO source code architecture and details the modifications made to their source codes to allow for LIDAR simulation and LAC, which enabled the pursuit of the aim and objectives. This Chapter will focus primarily on demonstrating how the enhancements and additions made to the source code were important for allowing the subsequent simulations and case studies to be performed. References to figures within Appendix B, consisting of screenshots of the source code files, will be provided throughout. These figures seek to support the descriptions of the source code modifications outlined within this Chapter. Work has been carried out by Schlipf et al. (2009) and Guo et al. (2022) on the development of a LIDAR simulator via an additional OpenFAST module, termed LidarSim, which is interfaced to ROSCO via a `.dll` chain. However, for the work of this research project, LIDAR simulation and LAC were required to be embedded within the original OpenFAST/ROSCO framework to allow for implementation within the source codes of other modelling tools such as FAST.Farm and WEIS/WISDEM. This was to enable LAC simulations of turbines at the array level or with modified tower geometry so that the aim and objectives of the research project could be addressed.

3.1 InflowWind source code architecture

OpenFAST is coded in Fortran programming language. Modules are configured in a manner similar to that for the InflowWind module, illustrated in Fig. 3.1.

`InflowWind.txt` specifies the variables to be used within the module. These are categorised depending on their function. Within the source code, these are defined as "`Types`", following the principles of type-oriented programming. The function of key `Types` used within the OpenFAST source code are outlined in Table 3.1.

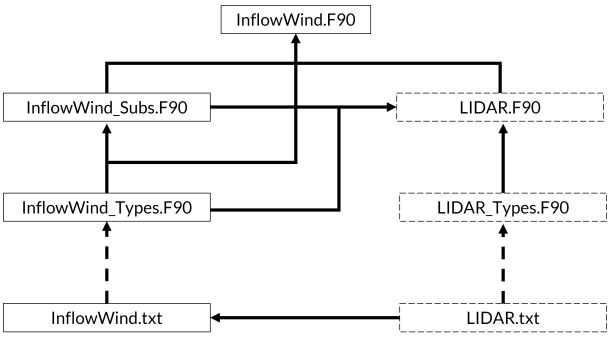


Figure 3.1: Illustrating the flow of the InflowWind module source code architecture. Dashed arrows indicate files that are written to the connecting script and solid arrows indicate where a script calls to the subroutines within the connecting script.

Table 3.1: Functions of key source code `Types` used within the OpenFAST source code.

Type name	Description
`InitInp`	Input values allocated when the module is initiated.
`InputFile`	Data specified in the module's input file.
`InitOutput`	Output values allocated when the module is initiated.
`Parameter`	Values to be stored within the module and used during calculations.
`Input`	Input data provided from other modules.
`Output`	Output data to be sent to other modules.

The `InflowWind.txt` (Figs. B.3 and B.4) also calls from the `LIDAR.txt` (Figs. B.1 and B.2) file to receive its inputs and outputs, which are assimilated to be used within the InflowWind module and LIDAR sub-module. Upon each new building of the software, the `.txt` files are automatically parsed to `_Types.f90` files, such that the variables specified within the `.txt` files are converted to a format compatible with Fortran language. The master script (`InflowWind.f90` for InflowWind) uses the variables within `InflowWind_Types.f90` and `LIDAR_Types.f90` to run subroutines within itself and its sub-scripts (`InflowWind_Subs.f90` and `LIDAR.f90`). `InflowWind_Subs.f90` contains scripts to parse the input file, validate inputs and calculate and assign outputs. `LIDAR.f90` contains scripts to initiate the LIDAR, perform calculations and interface outputs to `InflowWind_Subs.f90`.

3.2 Implementation of LIDAR simulation within InflowWind

To enable LIDAR simulation within OpenFAST, significant modifications were made to the program's source code within this research project. This involved a series of key steps, which will be outlined in this Section.

Within the original OpenFAST software prior to modifications, there was an existing `LIDAR.f90` script. However, this was not interfaced directly through the InflowWind input file and could only be accessed via the external Simulink control interface. The functionality was also quite limited, with only a fixed focal position of a single beam possible to be modelled. Therefore, this work identified the opportunity to fully integrate the `LIDAR.f90` script within the InflowWind module, while enhancing its capabilities and customisability. The following points will outline the contributions to the source code made within this project to enable LIDAR simulation within the InflowWind module of OpenFAST. These changes were approved by NREL and are present in releases of OpenFAST from version 3.5 onwards (see NREL, 2024b).

3.2.1 Specification of LIDAR inputs within the InflowWind input file

The first step entailed modification of the source code to allow for user input and customisation of the LIDAR setup via the InflowWind input file. This was achieved by additions to the `InflowWind.txt` (Figs. B.3 and B.4) and `LIDAR.txt` (Figs. B.1 and B.2) files, specifically within the `InputFile` type. Table 3.2 details the LIDAR input variables that were added to the InflowWind source code and included within the InflowWind input file.

Table 3.2: LIDAR input variables added to the InflowWind module source code and included within the InflowWind input file.

Variable	Description	Units
`SensorType`	Single-point, pulsed or CW.	-
`NumBeam`	Specifies the number of LIDAR beams (up to five).	-
`NumPulseGate`	Number of pulse gates when using pulsed LIDAR.	-
`PulseSpacing`	Distance between LIDAR pulses.	m
`RotorApexOffsetPos`	x,y,z distances of LIDAR device from rotor apex.	m
`FocalDistanceX`	x-direction focal distance(s) of the LIDAR beam(s).	m
`FocalDistanceY`	y-direction focal distance(s) of the LIDAR beam(s).	m
`FocalDistanceZ`	z-direction focal distance(s) of the LIDAR beam(s).	m
`MesurementInterval`	Time between consecutive measurements.	s
`URefLid`	Reference mean wind speed.	ms^{-1}
`ConsiderHubMotion`	Option to modify focal positions according to hub motions.	-

The addition of the capacity to specify the LIDAR modelling properties within the InflowWind input file greatly improved the customisability of OpenFAST's LIDAR functionality, whilst enabling the integration of the enhancements to the simulator, which will be outlined in Sect. 3.2.4.

3.2.2 Parsing of LIDAR inputs

Within the `InflowWind_Subs.f90` code lies a subroutine to parse the data from the InflowWind input file. This was modified to include the incorporated LIDAR inputs specified in the previous section (Fig. B.5). Following this addition, the OpenFAST program expected to read the LIDAR inputs from the InflowWind input file (Fig. B.7), returning an error if the required variables were not present or were incorrectly formatted.

3.2.3 Interface of parameters to LIDAR sub-module

Within Fortran language, at the beginning of each subroutine, the `Types` used within them must be specified along with the notation `INTENT(IN)`, meaning that the variables are input from a different subroutine and not modified during the current subroutine, `INTENT(OUT)`, meaning that the subroutine outputs these variables, which may change within the subroutine's computation, or `INTENT(INOUT)`, which specifies variables that are input to the current subroutine from a different subroutine to then be modified and output from the current subroutine.

The parsed input of each variable from the input file was then saved under the "`Parameter`" type (Fig. B.8). These were specified within the `LIDAR.txt` file (Figs. B.1 and B.2). `Parameter` types were used herein instead of the `InputFile` type because they allowed for the data to be transferred to other subroutines, namely `LIDAR.f90` (via `INTENT(IN)`). These parameters could then be used within the calculations by the LIDAR.

3.2.4 Enhancement of LIDAR calculations

The following additions were made to the `LIDAR.f90` (Figs. B.9 to B.17) script within this work to enhance the realism, capability and customisation aspects of the simulator.

Customisable LIDAR focal position

Within the original `LIDAR.f90` script, the focal position of the LIDAR beam was fixed at 150 m directly ahead of the turbine at hub height. This work therefore developed the `LIDAR.f90` script to allow for the customisation of the LIDAR focal position(s) via the `FocalDistanceX`, `FocalDistanceY` and `FocalDistanceZ` inputs added to the InflowWind input file, shown in Table 3.2. This addition was particularly important because the x-direction focal distance was used when determining the transport time from the point of measurement to the rotor. The x-direction focal distance was required to be defined in order to allow for adequate preview time for the turbine controller by considering the cutoff frequency of the low-pass filter, which incurs a time delay to the measurements. The cutoff frequency depends upon the quality of

the LIDAR preview, with lower quality preview requiring lower cutoff frequencies and longer time delays, thereby requiring the x-direction focal distance to be adjusted accordingly to ensure that enough preview time was provided to the controller before the wind's impact. These aspects and their importance to this work are fully detailed in Sect. 4.5.

Modification of LIDAR focal positions

Subsequently, code was added to modify the LIDAR focal coordinates as a result of the nacelle's motions. This is particularly relevant to FOWTs, whose motions are naturally significantly greater than those of BFW turbines. This was achieved by extraction of the x-, y- and z-direction hub displacements from the ElastoDyn module within the `FAST_Subs.f90` script (Fig. B.18), which were then specified as inputs to the LIDAR module in the `FAST_Solver.f90` script (Fig. B.19). Following this, the displacements were used to modify the specified LIDAR beam focal positions in real time by

$$x_i = x_{i,0} + d_x \tag{3.1a}$$

$$y_i = y_{i,0} + d_y \tag{3.1b}$$

$$z_i = z_{i,0} + d_z \tag{3.1c}$$

where x, y and z are the x, y and z focal distances of a LIDAR beam, i, relative to the nacelle, $x_{i,0}$, $y_{i,0}$ and $z_{i,0}$ are the initial focal positions (specified within the InflowWind input file) and d_x, d_y and d_z are the nacelle displacements from the origin.

The motions of the LIDAR device impact upon the LIDAR preview quality and, subsequently, the cutoff frequency of the low-pass filter. This cutoff frequency is generally lower for FOWTs due to the motions of the LIDAR device and the measurement errors that they induce (see Guo & Schlipf, 2023). Therefore, it was important to capture aspects of the motions and their impacts on the wind preview quality within this work.

Multiple beams

The modifications also allowed for the specification of multiple LIDAR beams and their respective focal positions. This was an important addition because it allowed for the configuration of a scan geometry (as used in real LIDAR systems), enabling calculation of the REWS used within the FFCPC, as well as for the use of the FFIPC strategy, as data from multiple positions within the rotor plane was required. Changes were made to allow users to specify the number of beams and their respective x, y and z focal coordinates via the InflowWind input file. The longitudinal wind speed could then be calculated at each position through the use of a `DO` loop, shown within Fig. B.13.

Measurement interval

Code was added to allow users to specify the measurement interval of the LIDAR beam(s). The code could determine whether a new output was required via

$$m_c = \frac{T}{\Delta T_m} \tag{3.2}$$

where m_c is the current simulation step, T is the current simulation time and ΔT_m is the specified LIDAR measurement interval. m_c was rounded down to the integer, which was critical because a new LIDAR measurement was only taken when

$$T = \Delta T_m \, m_c \tag{3.3}$$

This method ensured that the LIDAR measurement was only taken when T had incremented by an integer multiple of the specified ΔT_m. Real LIDAR devices record measurements at an interval that is an order of magnitude higher (between 0.2-1 seconds) than the OpenFAST simulation time used within this work (0.025 seconds) and so this was an important addition to consolidate the simulation of the wind measurements.

3.2.5 Wind speed determination

This Section details how the wind speed is determined by the OpenFAST LIDAR simulator. The wind speed is calculated slightly differently for the single-point, CW or pulsed configurations, and the source code for these are documented within Figs. B.13 to B.17. These configurations were already coded for within the existing `LIDAR.f90` script, though it remains important to detail their modes of operation and limitations. Single-point beams were used in the forthcoming control studies outlined in this work because they allowed for the specification of multiple beam positions (a feature that was added within this work), and therefore determination of the REWS, which was crucial for the FFCPC and FFIPC control strategies. Hence, as CW and pulsed beam configurations were not used in the subsequent case studies, their operation is not explicitly outlined in this work, though the OpenFAST source code relevant to their operation can be found in Figs. B.14 to B.17 while their theory is detailed in work by Sathe et al. (2015).

In the single-point configuration, the LIDAR measurement x, y and z coordinates (modified according to the hub displacements, as requested within the input file) were used to determine the u, v and w components of the wind velocity vector at the focal positions. The single-point beam configuration operated most similarly to the pulsed beam configuration, because measurements were taken in intervals rather than continuously - an approach that has been consistently taken in previous LAC studies (see Guo & Schlipf, 2023; Schlipf et al., 2020, 2015). In this implementation, as single point beams were used, a range weighting function

model was not used to characterise the spatial distribution of the beam's intensity, which further evidences the dissimilarity to the CW configuration, given that the probe volume of CW LIDAR beams is sensitive to distance, increasing by the square of the focus distance (see Fu et al., 2022; Simley et al., 2012), as was mentioned in Sect. 1.3.2. Instead, the u, v, w vector at each measurement position was extracted directly from the corresponding coordinates within the TurbSim wind field input at each measurement interval. Therefore, this assumed that measurement error and uncertainty otherwise resulting from the weighting function model was not incurred. Subsequently, the longitudinal wind speed associated with the beam's measurement point was determined by calculation of the magnitude of the u, v, w vector

$$u_L = \sqrt{V_{u,v,w} \cdot V_{u,v,w}} \tag{3.4}$$

where u_L is the LIDAR-measured longitudinal wind speed and $V_{u,v,w}$ is the wind velocity vector at the specified measurement coordinates. Despite the capability to configure multiple single-point beams, which was crucial to this work, it can be conceded that the LIDAR beams were not modelled like those of a true LIDAR system as it did not use a weighting function model, record a LOS wind speed, or perform a wind field reconstruction. However, the output required for LAC was the same as that obtained from a true LIDAR device - u, which dominates the aerodynamic loading of wind turbines over the v and w components. As the turbine and LIDAR beam(s) were assumed to be perfectly aligned with the wind direction, thereby neglecting the angling of the outer beams, it was possible to assume that the v and w components were equal to zero (as in Schlipf, 2016), side-stepping the need to perform a wind field reconstruction.

Furthermore, for FOWTs, the motions of the nacelle impact upon the apparent wind speed measured by the LIDAR, as these motions also cause the LIDAR device to move. This has led to the implementation of motion compensation algorithms, which seek to remove the contributions of the LIDAR velocity from the wind speed measurement (see Schlipf et al., 2020). Though the LIDAR simulator implemented to OpenFAST in this research accounts for the x, y and z displacements of the nacelle, a full motion compensation algorithm (accounting for the velocities, accelerations and rotations of the LIDAR device) was not implemented, which may have led to an overestimation of the LIDAR preview quality. An overestimation of the LIDAR preview quality can result in an overestimation of the cutoff frequency of the low-pass filter applied to the LIDAR-measured REWS, leading to less restriction upon the FF control action. However, at least with the single-point beam functionality, the LIDAR was not fully coupled to the motions of the nacelle. Rather, the simulator determined the u, v, and w components of the wind velocity vector at the instantaneous focal positions of the beams at each measurement point, and so the beams did not have a velocity associated with them. Therefore, it was deemed that this negated the need to apply a motion compensation

algorithm to the single-point beam measurements in the current setup. However, it is acknowledged that this assumption overlooked errors that may have arisen if an imperfect motion compensation algorithm was applied to a device that was fully coupled to the motions of the nacelle.

3.2.6 Export of LIDAR outputs

Following the calculation of the LIDAR-measured wind speed(s), these were transferred and assigned to InflowWind outputs within `InflowWind_Subs.f90` (Fig. B.6). OpenFAST has five output channels available for LIDAR wind measurement output, termed WindMeas. These were filled with the wind speed data either from multiple beams (when using single-point beams) or multiple range gates (when using pulsed LIDAR). The functionality to assign the outputs from the multiple beams was added within this work. These could be specified as outputs within the InflowWind input file for user analysis.

Code was added to allow for the outputs to be interfaced to ServoDyn, where they were registered as inputs within ServoDyn (Figs. B.20 and B.21). Once available within ServoDyn, these could be used to populate the avrSWAP array within the `BladedInterface_EX.f90` script, as shown in Fig. B.22. The avrSWAP array is a vector used by OpenFAST when transferring information to and from ROSCO. Outputs pertaining to the LIDAR simulator information required for LAC (the measurement positions, measured wind speeds, reference mean wind speed, sensor type, number of beams and number of pulse gates) were appended to the avrSWAP array within `BladedInterface_EX.f90` (Fig. B.22) to be used within ROSCO. This was a crucial addition because it established the link between the OpenFAST LIDAR information and measurements and the FF controllers that were added to ROSCO.

3.3 LIDAR simulation capabilities

Using the customisable LIDAR inputs, shown in Table 3.2, that were added to the InflowWind module, it was possible to configure the LIDAR simulator in multiple ways, as shown in Fig. 3.2.

Figure 3.2a, shows the ability to configure the LIDAR to measure at different sampling rates (in different simulation runs), while Fig. 3.2b shows the capability to focus the LIDAR beam at different x-direction focal distances in different simulation runs. Figure 3.2c, shows the outputs from a pulsed LIDAR, whereby measurements are taken at different range gates along the beam during one simulation. In each of Figs. 3.2a, b and c, the LIDAR-measured wind speeds are out of phase with the hub height wind speed because the measurements were taken at the specified focal distance ahead of the turbine. The exception to this is Fig. 3.2d, which illustrates the impact of including the x, y and z-direction displacements of the LIDAR beam

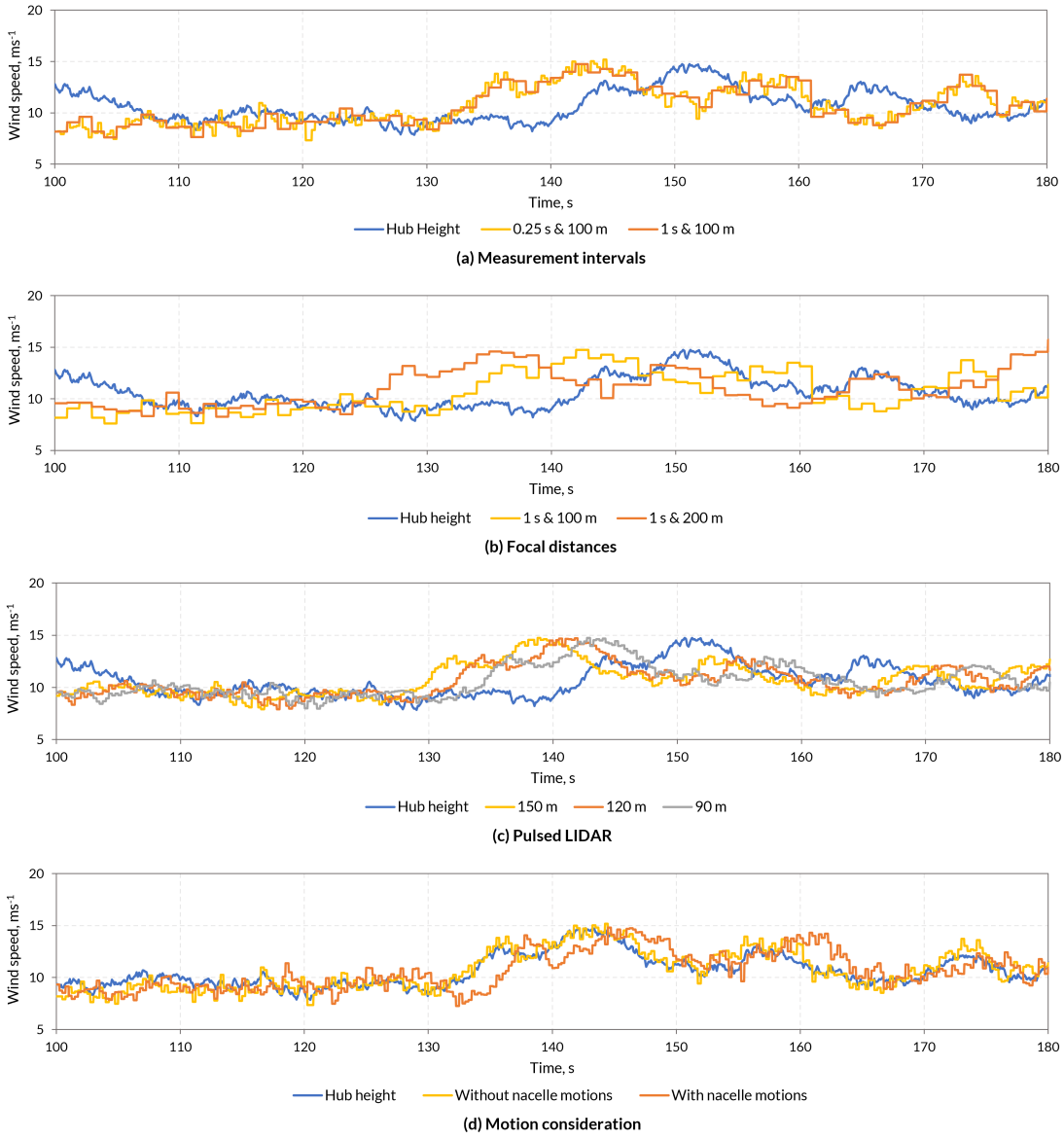


Figure 3.2: Time series of LIDAR-measured hub height wind speeds obtained from simulations of the modified OpenFAST code, with different customised LIDAR configurations. The time series plots in (d) have been time shifted to align with each other.

focal position on the measurement output, which is time-shifted to align with the output of the hub-height wind speed. From Fig. 3.2d, it is evident that the consideration of nacelle motions introduced errors in the LIDAR measured wind speed, compared to the hub height wind speed and the LIDAR output when the hub motion was not considered. This is expected to have an impact on the LIDAR preview quality, and therefore the cutoff frequency of the low-pass filter (determined within Sect. 4.5), as was reported by Guo and Schlipf (2023), which restricts the FF control action.

3.4 ROSCO source code architecture

Following the provision of the LIDAR outputs within the avrSWAP array, the ROSCO source code was modified to parse and use them within new subroutines created to enact FF control. To provide context to the ROSCO modifications, the architecture of the ROSCO source code is shown in Fig. 3.3, with scripts categorised according to their function.

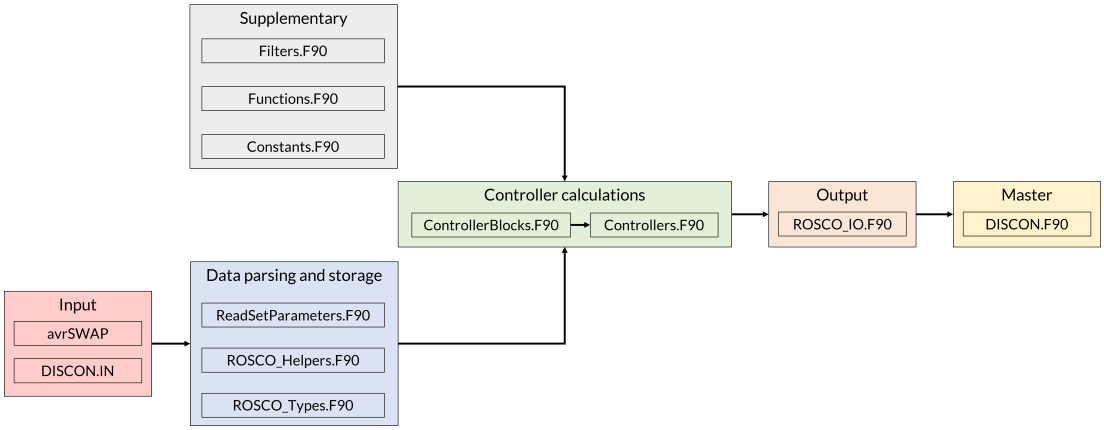


Figure 3.3: Outlining the flow of the ROSCO source code architecture. Scripts are categorised into boxes depending on their function.

The data from OpenFAST is transferred via the avSWAP array, with these parsed into ROSCO (Fig. B.23) along with those specified within the ROSCO input file (`DISCON.IN`) via `ReadSetParameters.f90` (Fig. B.24) according to variables defined in `ROSCO_Types.f90`. The different variable types and their functions within ROSCO are detailed in Table 3.3. The control action is then performed using the `ControllerBlocks.f90` and `Controllers.f90` scripts using the parsed variables and the filters, functions and constants outlined in their respective supplementary scripts. The outputs are collated within `ROSCO_IO.f90`. `DISCON.f90` is the master script that calls to the other aforementioned functions.

Table 3.3: Functions of key source code `Types` used within the ROSCO source code.

Type name	Description
`LocalVariables`	Variables within ROSCO that are subject to change during the simulations.
`ControlParameters`	Data specified in the ROSCO input file and are not subject to change during the simulations.
`FilterParameters`	Parameters used by the filters within ROSCO.
`DebugVariables`	Values output in the debug files of ROSCO.

3.5 Integration of LIDAR-assisted control within ROSCO

To enable LAC within ROSCO, significant modifications were made to its source code within this work to ensure that the required inputs and outputs were interfaced and the FF commands were calculated and issued. This involved a series of key steps, which will be outlined in this Section.

3.5.1 Parsing of LIDAR parameters from the avrSWAP array

To allow for the LIDAR outputs to be extracted from the avrSWAP array and used within ROSCO, these had to be called within the `ReadSetParameters.f90` script and assigned to the `LocalVariables` type (Fig. B.23). To enable this, the variables had to be specified within the `ROSCO_Types.f90` script (Fig. B.26). The `LocalVariables` type registers values that can change during the simulations. Once parsed, the LIDAR data could be used within ROSCO.

3.5.2 Specification of feedforward control inputs

In addition to the LIDAR outputs received from OpenFAST, ROSCO also had to be provided with values of parameters that were required for the FF control subroutines. ROSCO uses its own dedicated input file where values are defined by the user and certain features are switched on or off. Within ROSCO's source code, these are specified in `ROSCO_Types.f90` under the `ControlParameters` type (Fig. B.25). Unlike the `LocalVariables` type, the values of `ControlParameters` do not change during simulations. The required FF control inputs to be included within the ROSCO input file, shown in Table 3.4, were added within `ROSCO_Types.f90` along with a term to allow users to switch the FF controller off, to FFCPC or to FFCPC+FFIPC. Following this, code was added within the `ReadSetParameters.f90` script (Fig. B.24) to enable parsing of the data from ROSCO's input file, returning an error if the required fields were not present or were incorrectly formatted. The ROSCO input file with the additional LAC inputs is shown in Figs. B.27 and B.28.

3.5.3 Calculation of LIDAR beam azimuth angles

When using the FFIPC, the rotor (i.e. the y-z plane) azimuth angles of the beams were required to be known in order to assign the resulting pitch command associated with the beam's wind speed measurement to the appropriate blade, as will be fully outlined in Sect. 4.4. These were determined from the y- and z-axis focal distances. As these values change in real-time due to the motions of the nacelle following the addition of the interface of the hub displacement information from ElastoDyn, the azimuth angles of the beams were also subject to change. Therefore, the azimuth angles of the beams were calculated in real-time. This was achieved through the creation of an additional subroutine within the `ControllerBlocks.f90`

Table 3.4: LAC inputs added to the ROSCO input file.

Variable	Description	Units
`HubHeight`	The hub height of the turbine.	m
`CohWaveNum`	Maximum coherent wavenumber.	$radm^{-1}$.
`f_cutoff_IPC`	Cutoff frequency of the FFIPC low-pass filter.	Hz
`f_delay`	Critical frequency near where the rotor has high fluctuations.	$rads^{-1}$
`T_scan`	Time to complete a full LIDAR scan.	s
`K_IPC`	Gain applied to the FFIPC commands.	-
`NumPitchCurveFF`	Number of points on the wind speed-pitch curve.	-
`REWS_Curve`	Wind speed values used for the wind speed-pitch curve.	ms^{-1}
`Pitch_Curve`	Blade pitch values used for the wind speed-pitch curve.	rad

scripts termed `LidarConfiguration` (Fig. B.29). Within the `LidarConfiguration` sub-routine, the azimuth angle of each beam (i) was determined as follows. Firstly, the hub height was subtracted from the z-axis focal position(s) (z_i) to give the height relative to the hub ($z_{0,i}$).

$$z_{0,i} = z_i - z_{hub} \tag{3.5}$$

Using $z_{0,i}$ and the y-axis focal position (y_i), the azimuth angle of the beam (θ_i) was determined depending on which quadrant of the two-dimensional axis the focal point was positioned in, as shown in Table 3.5.

Table 3.5: Determination of beam rotor azimuth angles depending on y- and z-axis focal positions.

$z_{0,i}$	y_i	$\theta_i =$
> 0	= 0	0
= 0	> 0	$\frac{\pi}{2}$
< 0	= 0	π
= 0	< 0	$\frac{3\pi}{2}$
> 0	> 0	$\frac{\pi}{2} - arctan(\frac{z_{0,i}}{y_i})$
< 0	> 0	$arctan(\frac{\lvert z_{0,i}\rvert}{y_i}) + \frac{\pi}{2}$
< 0	< 0	$(\frac{\pi}{2} - arctan(\frac{z_{0,i}}{y_i})) + \pi$
> 0	< 0	$arctan(\frac{z_{0,i}}{\lvert y_i\rvert}) + \frac{3\pi}{2}$

This allowed for the determination of the rotor azimuth angle of each beam. These were used to compare with the azimuth angles of the blades, as will be outlined in Sect. 4.4.

3.5.4 Development of feedforward collective and individual pitch control subroutines

The final step was to enable the use of the FF collective and individual control elements. These had to be coded for as subroutines within the `Controller.f90` script (Figs. B.30 and B.32), which could then be called during the `PitchControl` subroutine (Figs. B.33 to B.35). For the FFCPC, a modified PI controller function was added within the `Functions.f90` script (see Fig. B.31). The theory behind the FF control strategies is explained in Chapter 4. Following the addition of a flag within the ROSCO input file, shown in Fig. B.27, users could choose to enable/disable the FF controllers.

3.6 WEIS source code modifications

The modifications to the OpenFAST and ROSCO source codes were carried forward and applied to their respective source codes used within WEIS/WISDEM. WEIS/WISDEM is primarily written in Python language but compiles OpenFAST and ROSCO from the Fortran source code that it is provided, as well as creating the OpenFAST module input and ROSCO input files. Modifications were made to the Python code of WEIS/WISDEM to ensure that the OpenFAST and ROSCO input files were created that included the LIDAR simulation and LAC inputs. This could allow for LAC to be simulated with FOWTs of modified geometry. An overview of the key WEIS/WISDEM source code files that were modified in order to achieve this objective is outlined in Table 3.6.

Table 3.6: Relevant WEIS/WISDEM Python scripts and `.yaml` files and their functions.

Script name	Purpose
`modelling_schema.yaml`	Specifies the OpenFAST input files' parameters and their values.
`toolbox_schema.yaml`	Specifies the ROSCO input file parameters and their values.
`FAST_reader.py`	Reads the input parameters and their respective values from `modelling_schema.yaml` (OpenFAST) and toolbox_schema.yaml (ROSCO).
`FAST_vars_out.py`	Specifies the OpenFAST output channels to be enabled/disabled within the input files.
`FAST_writer.py`	Writes the OpenFAST input files from the data read by `FAST_reader.py`.
`utilities.py`	Writes the ROSCO input file from the data read by `FAST_reader.py`.

The following changes were made to the source code scripts of WEIS to ensure that the LIDAR simulation and LAC inputs were incorporated within the computations:

- Updated `FAST_reader.py` (Fig. B.36) and `FAST_writer.py` (Fig. B.37) such that the new LIDAR configuration inputs were included within the InflowWind input file written by WEIS.
- Enabled WindMeas outputs within `FAST_vars_out.py` (Fig. B.38), ensuring that LIDAR outputs were recorded within the OpenFAST `.out` file.
- Updated `utilities.py` (Fig. B.39) to ensure that the new LAC inputs were included within the ROSCO input file written by WEIS.
- Updated the `modelling_schema.yaml` (for OpenFAST) and `toolbox_schema.yaml` (ROSCO) files so that the desired values of the LIDAR configuration and LAC inputs were written to the respective input files.

These changes ensured that LIDAR simulation and LAC could be configured within WEIS. The structural parameters, namely the tower properties, of the IEA 15 MW VolturnUS-S turbine could then be customised (within the FOWT's `.yaml` parameterised input file) and simulations performed with LAC enabled, with results shown in Chapter 7. The implementation and applications of these changes within WEIS represent the first time, to the best of the author's knowledge, in which LIDAR simulation and LAC have been used to perform analysis of a FOWT with modified geometry within a wind turbine structural design tool.

Chapter 4

Pitch control theory

This Chapter provides an overview of the pertinent blade pitch control theory of the controllers used within this work. This includes the FB collective and individual pitch controllers as well as Schlipf et al. (2015)'s FFCPC strategy and the novel FFIPC strategy curated during this research project. Both FF controllers were implemented within ROSCO in this work, as detailed in Sect. 3.4, receiving the outputs from the LIDAR simulator that was incorporated within OpenFAST. This Chapter will also provide details on the design of the low-pass filters used by the FF controllers, specifically the cutoff frequency, which depends upon the coherence between the LIDAR-measured and turbine-experienced REWSs.

4.1 Feedback collective pitch controller

The baseline FB pitch controllers of the simulated FOWTs used the respective default ROSCO inputs of the VolturnUS-S (see Allen et al., 2020) and WindCrete (see Mahfouz et al., 2020). They both utilise a gain-scheduled proportional-integral FB pitch controller, with gains detuned from the bottom-fixed configuration of the turbine to prevent undesirable coupling between the blade pitch control actuation frequency and the floating platform rigid-body pitch oscillation natural frequency. For the VolturnUS-S, Allen et al. (2020) modified the pitch controller's ζ_{des} and ω_{des} (used in Eq. 1.9) to 1.0 and 0.2 radians per second (rads^{-1}) from 0.7 and 0.6 rads^{-1}, respectively. The greater ζ_{des} and lower ω_{des} ensured that the bandwidth of the blade pitch controller was reduced such that coupling did not occur with the platform's pitching motion. For the WindCrete, Mahfouz et al. (2020) tuned the controller gains for the operating point corresponding to a wind speed of 13 ms^{-1} such that stability was achieved at this operating point. The tuning was based on the Ziegler and Nichols (1942) tuning method. After tuning the gains at 13 ms^{-1}, the ratio of the new PI gains to the original PI gains (of the BFW turbine) was determined and applied to the PI gains of the gain-schedule for all of the other above-rated operating points. The modified PI gains of the FB pitch controllers of the VolturnUS-S and WindCrete spar were not further adjusted in this work, and the default values available within their respective ROSCO input files were used.

The FB pitch controller requires the generator's rotational speed as the FB input. The overview of the fundamental FB collective pitch control theory was provided in Sect. 1.3.5. For floating turbines, an additional pitch command is provided through consideration of the nacelle rotational velocity to aid in overcoming negative aerodynamic damping. This is typically termed the floating FB damping loop. The tower-top velocity ($\dot{x}_{tt}$) is measured and multiplied by a proportional gain term ($K_{p,TFB}$) which is then combined with the command from the FB pitch controller, as was shown in Eq. 1.10. This additional control loop was only used within the VolturnUS-S model because it was not enabled within the WindCrete's default ROSCO input file.

4.2 Feedforward collective pitch controller

The FF control approach used a five-beam LIDAR, with four outer beams focused around the rotor disk and one focused at hub height directly ahead of the turbine, as illustrated in Fig. 4.1.

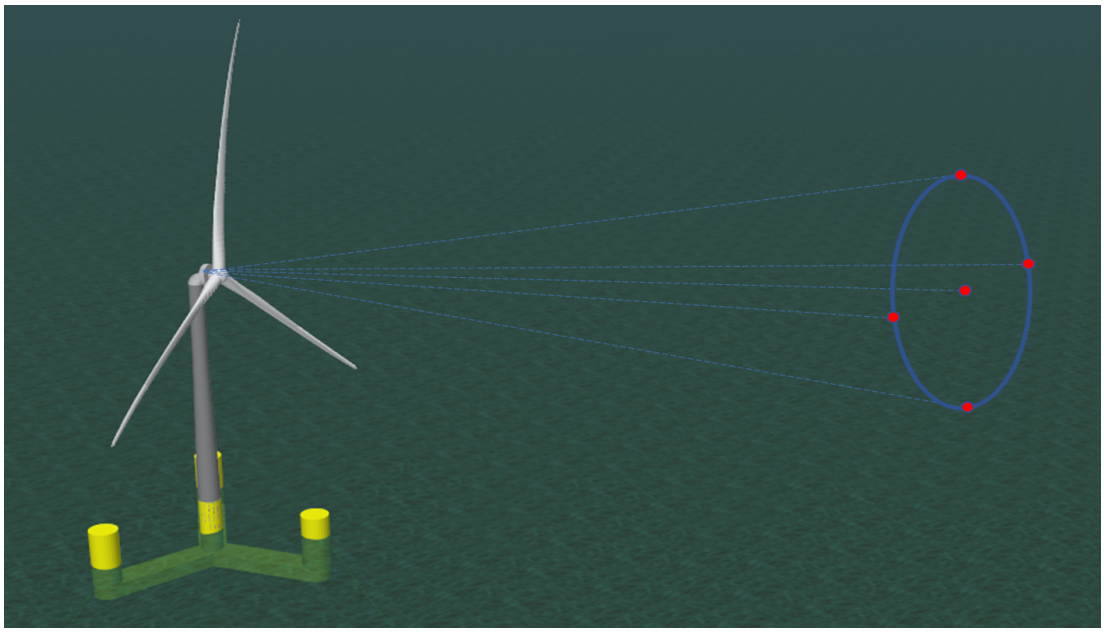


Figure 4.1: Illustration (not to scale) of a side view of the nacelle-mounted LIDAR beam trajectories, marked by blue dashed lines, and their focal positions, marked by red dots.

When using a multiple-beam LIDAR, as in this study, the REWS can be calculated. In this study, the LIDAR-measured REWS (v_L) was determined by averaging the longitudinal wind speeds recorded at the LIDAR measurement points, which were evenly distributed across the rotor-swept area

$$v_L = \frac{\sum_{i=1}^{n_{beams}} u_{L,i}}{n_{beams}} \tag{4.1}$$

where i is the beam number, $u_{L,i}$ is the longitudinal wind speed measured by a beam and n_{beams} is the total number of LIDAR beams. The appropriate LIDAR REWS record to be used for the FF collective pitch command had to be recalled at the required time. This required accounting for the transport delay of the wind from the point of measurement to the turbine as well as the time delays incurred by filtering and actuating. This was achieved through the use of a data storage vector. Upon each new time step of the simulation, the values stored in the vector were shifted across to vacate the first space of the array for the latest LIDAR REWS measurement. The current LIDAR REWS measurement was then passed through a first-order low-pass filter with the following transfer function

$$G_{LPF}(s) = \frac{2\pi f_{cutoff}}{s + 2\pi f_{cutoff}} \tag{4.2}$$

where f_{cutoff} is the cutoff frequency of the low-pass filter. Details on how this was obtained and the time delay that it incurred are provided in Sect. 4.5. The gain, G_{LPF}, was applied to filter the LIDAR REWS wind speed measurement, which was then assigned to the first position within the data storage vector. The LIDAR REWS wind speed measurement was assumed to not change between the point of measurement to the turbine, thereby assuming Taylor's frozen hypothesis, and so it was not changed after being saved to the data storage vector.

The approximate lead time (T_{lead}) of the current LIDAR measurement from the point of measurement to the turbine was

$$T_{lead,1} = \frac{\Delta x}{V_{ref}} \tag{4.3}$$

where Δx is the x-direction focal distance (equal for all beams) and V_{ref} is the reference wind speed (corresponding to the mean wind speed used within each simulation). The time within the simulation at which the latest LIDAR wind speed measurement was expected to reach the turbine ($T_{impact,1}$) was

$$T_{impact,1} = T_{meas} + T_{lead,1} \tag{4.4}$$

where T_{meas} is the simulation time at the point when the measurement is taken. The impact time was stored in a separate data storage vector housing estimated impact times for each LIDAR REWS record. Similarly to the REWS records, the latest impact time record assumed the first position after the previous records within the vector were shifted across. The two data storage vectors were aligned in their positions of the expected time of impact and the filtered LIDAR REWS. The time delays incurred by the first-order low-pass filter of the LIDAR measurements (T_{filter}), the second-order low-pass filter of the pitch actuators (T_{pitch}), the LIDAR full-scan duration, T_{scan} (1 s) and the current simulation time, T, were then subtracted from each of the values in the impact time storage vector, such that values within the vector decreased as T increased. The remaining storage time ($T_{storage,j}$) of data point j at a certain

T is defined as

$$T_{storage,j} = T_{impact,j} - T - T_{filter} - T_{pitch} - \frac{1}{2}T_{scan} \tag{4.5}$$

The location within the $T_{storage}$ vector that had a value closest to zero was then selected, and the value at the same location within the filtered LIDAR REWS vector was selected as the REWS record to be used when calculating the FF command for the current time step. This process ensured that the correct feedforward command was issued at the required time. The correct timing of LIDAR-assisted feedforward control commands is critical because it ensures that the wind turbine prepares for the expected wind profile. Incorrect timing may lead to control actions that do not suit the incoming wind conditions, which may lead to poor performance, instability and/or increased structural loads.

The FF blade pitch angle (β_{FF}) was then determined through interpolation of the wind-speed-to-blade-pitch-angle curve specified in the ROSCO input file, shown in Fig. 4.2.

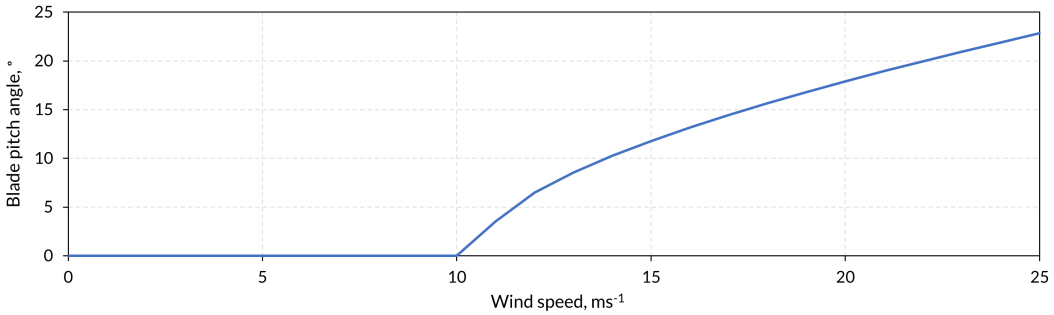


Figure 4.2: Wind-speed-to-blade-pitch-angle curve as used in the modified ROSCO input file.

The difference between β_{FF} and the FF command computed at the previous time step ($\beta_{FF,prev}$) was calculated and divided by the simulation time step duration, ΔT, to give the FF blade pitch rate ($\dot{\beta}_{FF}$)

$$\dot{\beta}_{FF} = \frac{\beta_{FF} - \beta_{FF,prev}}{\Delta T} \tag{4.6}$$

$\dot{\beta}_{FF}$ was then included within the calculation of I_{FB} (Eq. 1.7)

$$I_{FF\text{-}FB} = I_{FF\text{-}FB,prev} + \Delta T \left(K_i \, e_{FB} + \dot{\beta}_{FF} \right) \tag{4.7}$$

where $I_{FF\text{-}FB}$ is the FF modified FB controller integral component and $I_{FF\text{-}FB,prev}$ is the value of $I_{FF\text{-}FB}$ computed at the previous time-step. The FF rate addition methodology was developed and applied in previous studies (see Guo, Schlipf, & Cheng, 2023; Schlipf et al., 2020, 2015). Therefore, aside from its use of measurements from the aforementioned scan geometry and its application to different 15 MW FOWTs, the implementation of this FFCPC strategy within this work is acknowledged to not be an original contribution. $\dot{\beta}_{FF}$ was used by Schlipf et al. (2015) instead of β_{FF} to deliver the FFCPC command because it simplified the integration of

anti-windup algorithms and the switching of the FF controller on and off. P_{FB} (from Eq. 1.6) and $I_{FF\text{-}FB}$ were then combined to give the FF modified FB pitch command $(\beta_{FF\text{-}FB})$

$$\beta_{FF\text{-}FB} = (P_{FB} + I_{FF\text{-}FB}) + \beta_{TFB} \tag{4.8}$$

4.3 Feedback individual pitch controller

Simulations were performed combining FFCPC with FBIPC to enable comparison with those of FFCPC combined with the FFIPC strategy for the two FOWTs (VolturnUS-S and WindCrete). The FBIPC approach used in this study, as defined by Bossanyi (2003) and Magar et al. (2015) and configurable within ROSCO, uses the out-of-plane bending moment at the root of each blade and the rotor azimuth angle (θ) as its FB inputs. Magar et al. (2015), likened the sinusoidal variation in the blade root bending moments (driven by the vertical wind shear) of three-bladed horizontal axis wind turbines, whose blades are positioned 120° apart, to that of the three-phase system used in the electrical power systems. The blade pitch angles are given as a function of θ, per Bossanyi (2003)

$$\beta_1 = \bar{\beta}\ cos(\theta) \tag{4.9a}$$

$$\beta_2 = \bar{\beta}\ cos\left(\theta + \frac{2\pi}{3}\right) \tag{4.9b}$$

$$\beta_3 = \bar{\beta}\ cos\left(\theta + \frac{4\pi}{3}\right) \tag{4.9c}$$

where, β_1, β_2 and β_3 are the pitch angles of each blade and $\bar{\beta}$ is the mean blade pitch angle.

The blade pitch angles were then transformed to the direct-quadrature (d-q) axis using a multi-blade coordinate transformation, as was first demonstrated by Bossanyi (2003). This was applied to convert the three-phase blade root bending moments onto a 2-D perpendicular axis

$$\begin{bmatrix}\beta_d\\ \beta_q\end{bmatrix} = \frac{2}{3}\begin{bmatrix}cos(\theta) & cos\left(\theta+\frac{2\pi}{3}\right) & cos\left(\theta+\frac{4\pi}{3}\right)\\ sin(\theta) & sin\left(\theta+\frac{2\pi}{3}\right) & sin\left(\theta+\frac{4\pi}{3}\right)\end{bmatrix}\begin{bmatrix}\beta_1\\ \beta_2\\ \beta_3\end{bmatrix} \tag{4.10}$$

where β_d and β_q are the direct and quadrature axis outputs, respectively. The bending moments at the roots of the blades were assumed to be measured and then transformed to the d-q axis using Eq. 4.10. The controller was implemented during this transformation to keep the bending moments constant irrespective of changes in them due to variations in the wind speed. An inverse d-q transformation was then performed to convert the control outputs into

individual pitch commands

$$\begin{bmatrix}\beta_1\\ \beta_2\\ \beta_3\end{bmatrix} = \begin{bmatrix} cos(\theta) & sin(\theta)\\ cos\left(\theta+\frac{2\pi}{3}\right) & sin\left(\theta+\frac{2\pi}{3}\right)\\ cos\left(\theta+\frac{4\pi}{3}\right) & sin\left(\theta+\frac{4\pi}{3}\right)\end{bmatrix}\begin{bmatrix}\beta_d\\ \beta_q\end{bmatrix} \tag{4.11}$$

The individual pitch command for each blade was then combined with the collective pitch command from the FB controller. When combining FFCPC with FBIPC, the final blade pitch command was, therefore,

$$\beta_{FF\text{-}FB,b} = (P_{FB} + I_{FF\text{-}FB}) + \beta_{TFB} + \beta_{FB,b} \tag{4.12}$$

where $\beta_{FB,b}$ is the FBIPC command for an individual blade, b, and $\beta_{FF\text{-}FB,b}$ is the combined FFCPC+FBIPC pitch command for an individual blade.

4.4 Feedforward individual pitch controller

A novel individual pitch controller was designed and implemented within this work and aimed to tune each blade's pitch angle depending on its rotor azimuth position. It was used to provide an additional refinement to the FFCPC. The approach used a five-beam LIDAR, with beams configured as illustrated in Fig. 4.1. Only the four outer beams were used by the FFIPC. These beams were focused at 90° rotor plane azimuth increments, as illustrated in Fig. 4.3. Simulations were performed where the beams were positioned at various rotor radial span widths (30%, 50% and 70%) for the VolturnUS-S to determine the best-performing beam span positions for the FFCPC+FFIPC, with results presented in Sect. 5.4.3.

This approach was most similar to that used by Dunne, Simley, and Pao (2011) and Dunne et al. (2012), though still differed in numerous ways. Firstly, the LIDAR beams did not rotate with the blades. Instead, the blades were matched to one of the four outer beams depending on their azimuth positions. In Dunne et al. (2012)'s approach, separate instances of an FF PI controller were applied to each blade independently, and this could also be combined with the FBIPC. In this strategy, the FFCPC was still used to provide an identical command to all blades. The FFIPC was intended to provide additional refinement to the blade pitch angles according to the previewed wind information pertinent to the blade's azimuth position. The FFIPC was not combined with the FBIPC in this work, as was the case in Dunne, Simley, and Pao (2011)'s study. Therefore, the operation of the FFIPC was enabled solely from knowledge of the incoming wind measurements, the blade azimuths and the blade pitch angles. Hence,

the FFIPC did not require additional feedback from sensors at the blade roots as in the FBIPC approach. Finally, the FFIPC detailed in this work did not use a model to formulate commands (as in Dunne, Simley, & Pao, 2011) and so it was considered to be relatively computationally inexpensive and simple to execute.

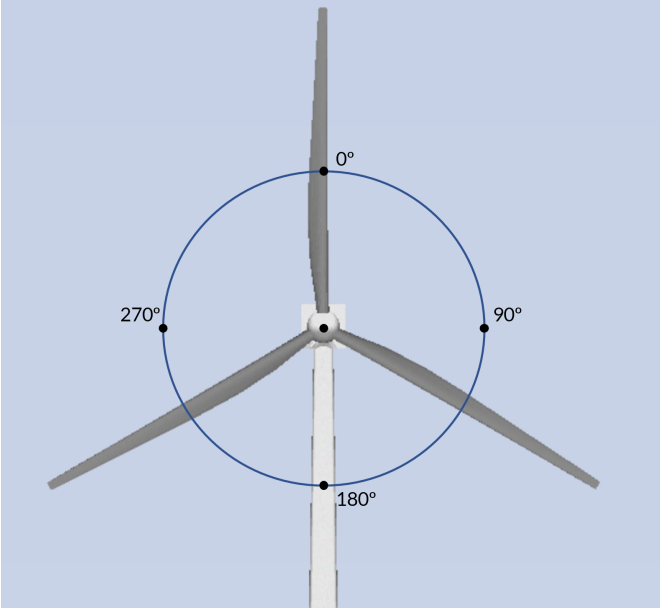


Figure 4.3: Azimuth positions of the LIDAR beam focal positions within the rotor plane at 50% rotor radial span.

Though the beams were focused at the specified points relative to the nacelle, the movement of the nacelle and its impact upon the measurement coordinates and beam azimuth angles were accounted for in real-time within the source code, as explained in Sect. 3.2.4. The longitudinal wind speed was recorded at each LIDAR beam's measurement point. These were then individually low-pass filtered using the same cutoff frequency as was used by the filter on the LIDAR-measured REWS for the FFCPC. The use of individual single-point beams for feedforward individual pitch control avoids the uncertainties arising from the aforementioned "Cyclops" dilemma (in Sect. 1.3.2). Although the Cyclops dilemma limits a single LIDAR beam's ability to distinguish between horizontal wind shear and wind direction changes, the single-point beams studied in this work did not employ a spatially-averaged range weighting function. This avoided the Cyclops dilemma issue because each single-point measurement was treated independently, without requiring spatial comparisons, and only recorded the wind velocity vector at the specific focal point, not across a probe volume, thereby eliminating any ambiguity between shear and wind direction changes. The beams were also assumed to be perfectly aligned with the wind direction, thereby excluding the angling of the outer beams and ensuring that the measured wind speed directly corresponded to the longitudinal component, u, of the wind velocity at the measurement point.

The pitch setpoint angle associated with the filtered wind speed recorded by each beam, $\hat{\beta}$, was then determined through interpolation of the wind-speed-to-blade-pitch-angle curve (Fig. 4.2). The azimuth positions of each blade (θ_b) at the current time were determined by feedback of the OpenFAST azimuth output (θ)

$$\theta_b = \theta + \frac{2\pi(b-1)}{3} \tag{4.13}$$

where b is the blade number from one to three. The current rotor rotational speed (Ω_r) was then used to predict the azimuth position of each blade ($\theta_{new,b}$) after the delay incurred by the low-pass filter that was later used on the individual commands (T_{IPC}), such that

$$\theta_{new,b} = \theta_b + \Omega_r \, T_{IPC} \tag{4.14}$$

An example scenario of this is illustrated in Fig. 4.4. This assumed that Ω_r did not change over the delay period incurred by the low-pass filter on the individual pitch commands. This assumption may have incurred errors in the estimation of the blade azimuths and subsequent matching to the beam azimuths. The challenge of assuming constant Ω_r for FFIPC was also briefly discussed by Simley and Pao (2013). The error resulting from this assumption was estimated, with results presented in Sect. 5.7.

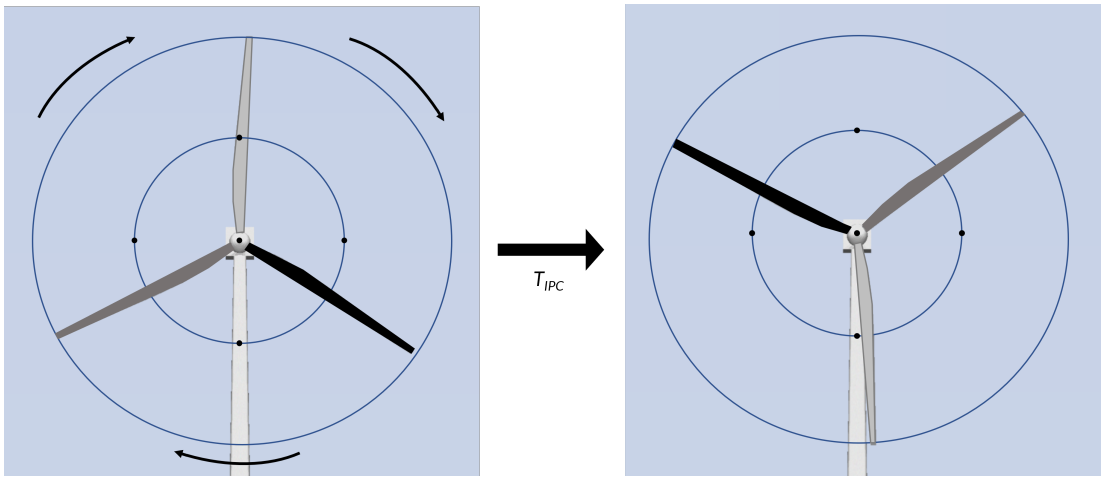


Figure 4.4: Illustrating an example scenario of the rotation of the turbine blades during the preview time provided for the FF individual pitch commands.

θ_{new} of each blade was then subtracted from the azimuth angle of each beam (θ_i)

$$\theta_{dif,b} = \theta_i - \theta_{new,b} \tag{4.15}$$

The beam associated with the smallest difference in azimuth angle to a blade ($\theta_{dif,b}$) was then selected to provide the individual pitch command to the blade. The error ($e_{FF,i}$) between the pitch angle setpoint associated with the beam, $\hat{\beta}_i$, and the current mean blade pitch angle ($\bar{\beta}$) was then determined.

$$e_{FF,i} = \hat{\beta}_i - \bar{\beta} \tag{4.16}$$

A low-pass filter with the following transfer function was used to prevent large variations in individual pitch commands

$$G_{LPF,IPC}(s) = \frac{2\pi f_{cutoff,IPC}}{s + 2\pi f_{cutoff,IPC}} \tag{4.17}$$

where $f_{cutoff,IPC}$ is the cutoff frequency of the low-pass filter applied to the FF individual pitch commands. Details on how $f_{cutoff,IPC}$ was determined and the time delay that it incurred are provided in Sect. 4.5. The gain, $G_{LPF,IPC}$, was applied to filter the FF individual pitch commands. This resulted in the FF pitch command for the individual blade, $\beta_{FF,b}$. The time series of an individual pitch command pre- and post-filtering are shown in Fig. 4.5.

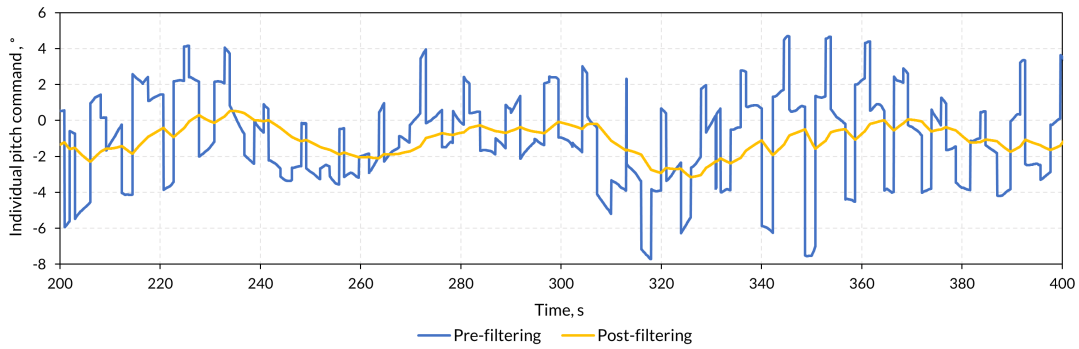


Figure 4.5: Excerpt of time series individual blade pitch commands pre- and post-low-pass filtering.

The time delay incurred by the filter required an additional term, T_{IPC}, to be considered within Eq. 4.5 when accounting for the storage time when undertaking FFIPC such that

$$T_{storage,j} = T_{impact,j} - T - T_{filter} - T_{pitch} - \frac{1}{2}T_{scan} - T_{IPC} \tag{4.18}$$

For the WindCrete, the magnitude of the FF individual pitch commands had to be further reduced by the introduction of a proportional gain term ($K_{p,IPC}$), in order to prevent excitation of the heave motion of the platform

$$\beta_{FF,b} = \beta_{FF,b}\, K_{p,IPC} \tag{4.19}$$

$K_{p,IPC}$ was tuned to 0.3 by trial and error. The individual component of the controller was designed to provide small adjustments to each blade in addition to the collective command that was provided to all of the blades

$$\beta_{FF\text{-}FB,b} = (P_{FB} + I_{FF\text{-}FB}) + \beta_{TFB} + \beta_{FF,b} \tag{4.20}$$

The application of the FFIPC resulted in an individual pitch variation as shown in Fig. 4.6.

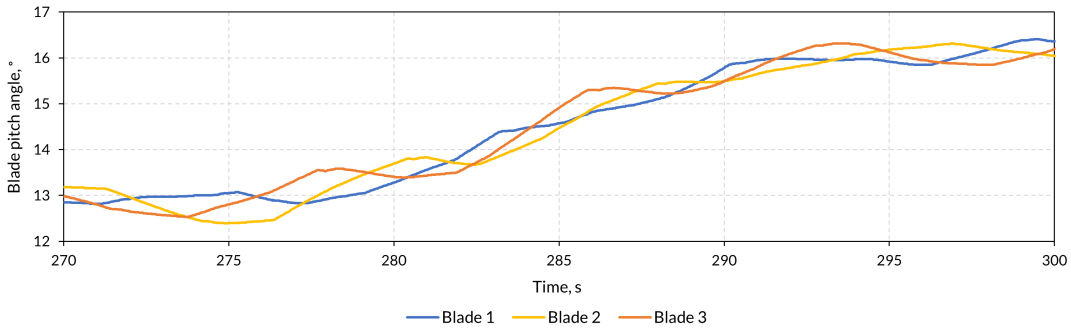


Figure 4.6: Excerpt of time series individual blade pitch variation when using FFIPC.

A flowchart illustrating the integration of the FFCPC and FFIPC loops within the turbine's pitch controller is presented in Fig. 4.7.

4.5 Filter design and time delays

This Section will outline how the cutoff frequencies and subsequent time delays of the low-pass filters used on the LIDAR REWS measurements, pitch actuators and FF individual pitch commands were determined. The time delays determined the minimum x-direction focal distance to ensure that the remaining preview time (T_{buffer}) in

$$T_{buffer} = T_{lead} - T_{filter} - T_{pitch} - \frac{1}{2}T_{scan} - T_{IPC} \tag{4.21}$$

was greater than zero at all reference wind speeds for both FOWTs, respectively. This was a critical requirement because it ensured that the feedforward commands were always delivered before the measured wind impacted upon the turbine so that the turbine could act in advance of the impact.

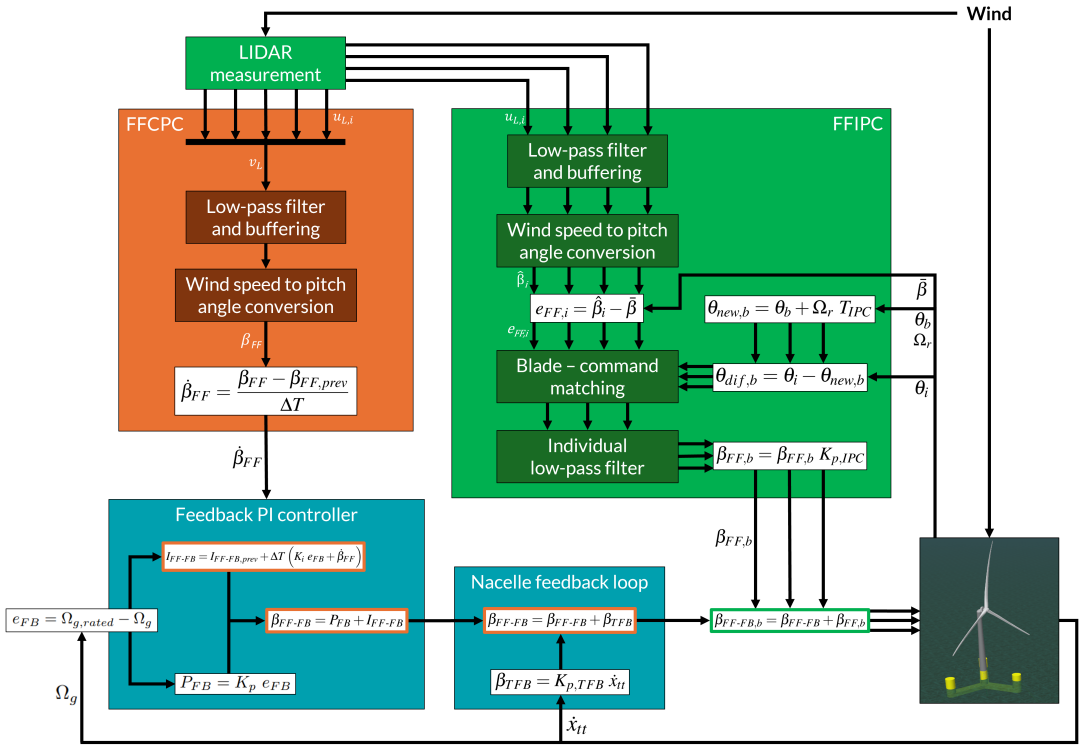


Figure 4.7: Flowchart illustrating the FB pitch and nacelle FB loops in addition to the FFCPC approach (from Schlipf et al., 2015) in orange and the novel FFIPC contribution in green (i = beams, b = blades).

4.5.1 LIDAR rotor-effective wind speed low-pass filter

A low-pass filter was applied to the LIDAR REWS time series measurements to ensure that frequencies above which there was a poor correlation between the LIDAR-measured REWS and the REWS experienced by the turbine were not considered. This point was defined by the cutoff frequency of the filter. Without filtering, or with the cutoff frequency set too high, uncorrelated frequencies would be used by the LIDAR-assisted controller, leading to incorrect control action, as was explained in Sects. 1.3.6 and 2.1. The cutoff frequency was determined through analysis of the transfer function between the time series of the LIDAR-measured and turbine REWSs ($|G_{RL}|$). $|G_{RL}|$ takes a value between 0 and 1, and indicates the quality of LIDAR preview (see Guo, Schlipf, & Cheng, 2023; Schlipf, 2016)

$$|G_{RL}(f)| = \frac{|S_{RL}(f)|}{S_{LL}(f)} \tag{4.22}$$

where S_{LL} is the auto-spectrum of the LIDAR estimated REWS and S_{RL} is the cross-spectrum of the LIDAR and turbine REWS estimates. From time series measurements, these spectra were calculated over a ten-minute period. The resulting frequency domain (f) was converted into the wavenumber domain (k) using Taylor's frozen turbulence hypothesis as in Held and Mann (2019)

$$k = \frac{2\pi f}{V_{ref}} \tag{4.23}$$

To compare to the LIDAR REWS measurements, the turbine-experienced REWS was determined by extraction of outputs from multiple nodes within OpenFAST, discretised to ten-metre increments across the rotor-swept area. These were then averaged to determine the REWS. S_{RL} and S_{LL} were determined using Matlab's cross power spectral density (see MathWorks, 2023a) and Welch's power spectral density estimate (see MathWorks, 2023b) functions, respectively. The spectra can also be determined analytically and details of how this can be performed are provided within the literature (see Held & Mann, 2019; Schlipf, 2016). The VolturnUS-S's LIDAR and turbine REWS time series used in the analysis are shown in Fig. 4.8. These resulted in the transfer function spectrum shown in Fig. 4.9.

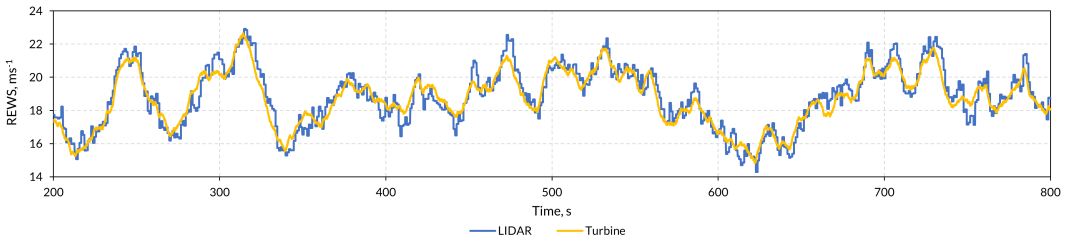


Figure 4.8: Ten-minute time series of VolturnUS-S's LIDAR-measured (radial span = 50%) and turbine-experienced REWSs used in the coherence analysis.

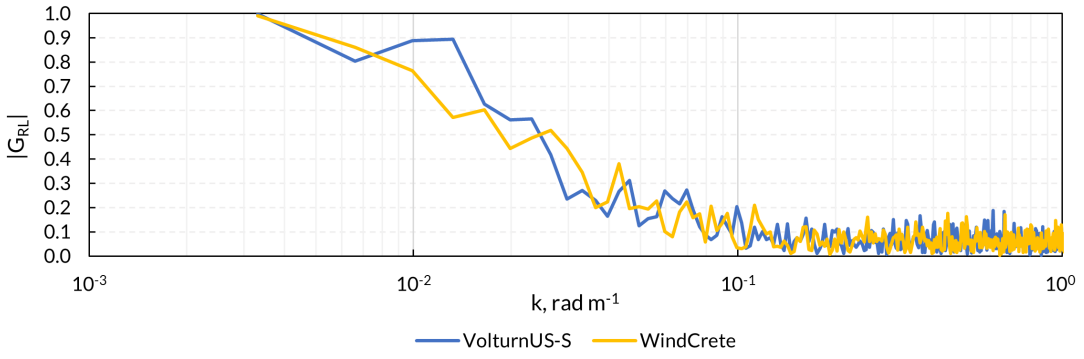


Figure 4.9: Transfer function ($|G_{RL}|$) between the LIDAR (radial span = 50%) and turbine REWS simulation data as a function of wavenumber (k) for the VolturnUS-S and the WindCrete.

The wavenumber at which $|G_{RL}|$ reaches -3 dB (at approximately 0.7) is termed the maximum coherent wavenumber, $\hat{k}$, above which the correlation between the LIDAR-measured and turbine-experienced REWSs can be considered to be poor. From Fig. 4.9, $\hat{k}$ occurred at a k of 0.01523 radm^{-1} for the VolturnUS-S and 0.01103 radm^{-1} for the WindCrete (both with outer LIDAR beams focused at 50% span). This process was repeated for beams positioned at 30% and 70% radial span for the VolturnUS-S, yielding a $\hat{k}$ of 0.0070 radm^{-1} and 0.0208 radm^{-1}, respectively. It was expected that $\hat{k}$ would increase as the radial span width increased because wind speeds covering a broader area of the rotor were captured, leading to a more representative preview of the REWS. The cutoff frequencies for different reference wind speeds for each LIDAR beam configuration were then determined, as per Schlipf (2016)

$$f_{cutoff} = \frac{\hat{k} V_{ref}}{2\pi} \tag{4.24}$$

Figure 4.10 shows the value of f_{cutoff} at different reference wind speeds for the two FOWTs with beams at 50% span.

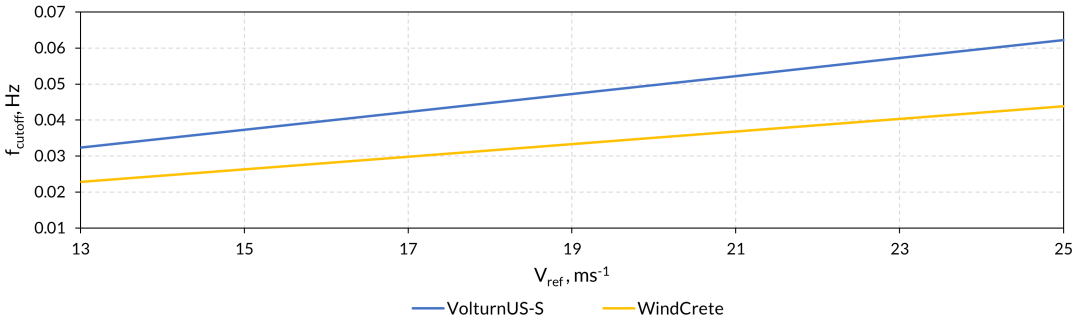


Figure 4.10: f_{cutoff} at different mean wind speeds at 50% radial span for the two FOWTs at their minimum x-direction focal distances.

The cutoff frequencies of the low-pass filter of the VolturnUS-S were slightly higher than those of the WindCrete as a result of its higher $\hat{k}$, resulting in less restriction on its FF control action.

Figure 4.11 shows the value of f_{cutoff} at different reference wind speeds for the different radial span configurations of the LIDAR on the VolturnUS-S.

The low-pass filter for the REWS is first-order and incurs a time delay (T_{filter}), which can be approximated at a certain frequency using equations from Schlipf (2016)

$$T_{filter} = \frac{arctan(\omega_n)}{\omega_{delay}} \; with \; \omega_n = \frac{\omega_{delay}}{\omega_{cutoff}} \tag{4.25}$$

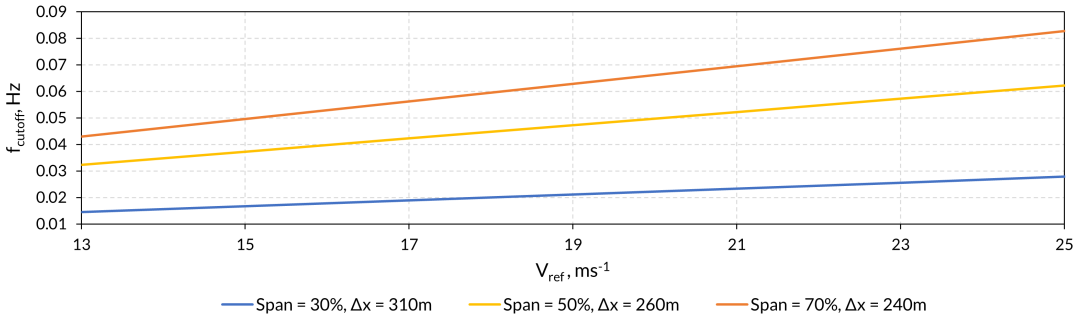


Figure 4.11: f_{cutoff} at different mean wind speeds for each of the radial span locations at their minimum x-direction focal distances for the VolturnUS-S.

where ω is an angular frequency ($\omega_{delay} = 2\pi f_{delay}$, $\omega_{cutoff} = 2\pi f_{cutoff}$). The pitch actuator uses a second-order low-pass filter, whose time delay can be approximated using (Schlipf, 2016)

$$T_{pitch} = \frac{arctan(\frac{\sqrt{2}\omega_n}{1-\omega_n^2})}{\omega_{delay}} \tag{4.26}$$

In both Eqs. 4.25 and 4.26, f_{delay} was set to 0.025 Hz, as this is the critical frequency near where the IEA 15 MW turbine's rotor has higher fluctuations (Guo & Schlipf, 2023).

4.5.2 Feedforward individual pitch controller low-pass filter

As discussed in Sect. 4.4, a low-pass filter was required to smoothen and reduce the magnitude of the FF individual pitch commands. To achieve this, the cutoff frequency had to be set to a low level, which also mitigated interference with key turbine and substructure frequencies. In this work, f_{cutoff} of the FFIPC filter was set to 0.0132 Hz, which is below the platform pitch rigid body natural frequency of both FOWTs (0.036 Hz for the VolturnUS-S and 0.024 Hz for the WindCrete) as well as the frequency where the turbine's rotor has higher fluctuations (0.025 Hz, as per Guo & Schlipf, 2023). There was not expected to be interference with the tower frequencies because, for these FOWTs, the tower's first natural frequencies were above the 3P range. Consequently, this filter resulted in an additional time delay (T_{IPC}) to be subtracted from T_{lead} in Eq. 4.21, calculated using Eq. 4.25. This resulted in the requirement for higher values of T_{lead} to ensure that T_{buffer} was greater than zero at all wind speeds, as shown in Fig. 4.12.

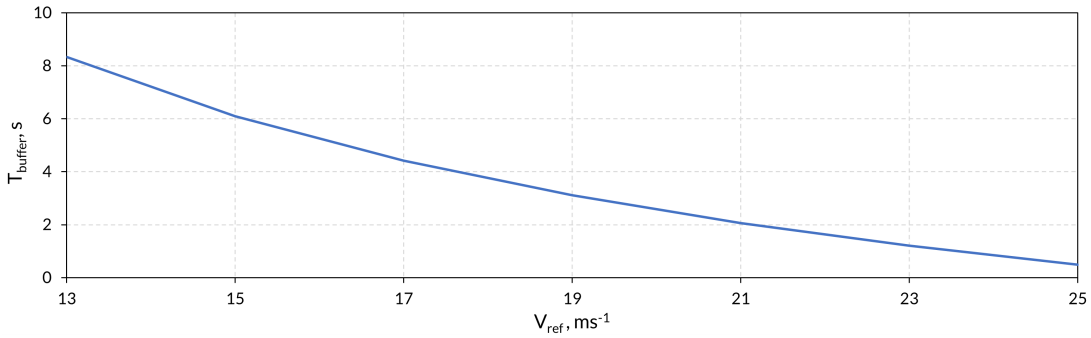


Figure 4.12: T_{buffer} at different reference wind speeds for beams positioned at 50% radial span of the VolturnUS-S at an x-direction focal distance of 260 m.

This was achieved by setting the x-direction focal distance of the beams to 310 m, 260 m and 240 m for beams focused at 30%, 50% and 70% radial span of the VolturnUS-S, respectively, and 280m for the WindCrete at 50% span. The minimum x-direction focal distance decreased as the radial span increased because values of $\hat{k}$ and, therefore, f_{cutoff} (Fig. 4.11) were greater as the radial span increased. Higher values of f_{cutoff} resulted in lesser time delays incurred by the low-pass filter on the LIDAR REWS. Therefore, less lead time was required and the LIDAR beams could be focused at positions closer to the turbine.

Chapter 5

OpenFAST floating offshore wind turbine simulations

This Chapter outlines the results of the OpenFAST simulations performed on the VolturnUS-S and WindCrete FOWTs. Firstly, results of an FB-only collective blade pitch controller (henceforth termed FB-only controller) comparison will be provided, which aided in ascertaining the key performance differences between the two FOWTs and identifying areas where the LAC strategies (FFCPC, FFCPC+FBIPC and FFCPC+FFIPC) could offer improvement. Initial comparisons also highlighted the need to apply limits to the FF commands of the WindCrete in low above-rated wind speed conditions. The results of the FF control simulations of the VolturnUS-S, followed by the WindCrete are presented in this Chapter, in addition to the comparisons made between them.

5.1 Simulation set-up

Turbulent wind fields were created using NREL's TurbSim stochastic full-field inflow simulator, developed by Kelley and Jonkman (2005), with mean wind speeds covering the above-rated wind spectrum, from 13-25 ms^{-1}, in intervals of 2 ms^{-1}. These were generated for four random seeds of one-hour length (3800 seconds were simulated but the first 200 seconds were discarded to remove start-up transients) to provide a variety of turbulent wind profiles. Though 11 ms^{-1} is above the rated wind speed of the IEA 15 MW turbine (10.69 ms^{-1}), results from simulations with a mean wind speed of 11 ms^{-1} are not presented in this Chapter because a significant proportion of those simulations were spent in below-rated wind speeds, where LAC was inactive. The turbulence was generated under the NTM, using the Kaimal spectrum (see Kaimal et al., 1972), with TI category B, corresponding to a TI of 14%. These conditions fell into design load DLC 1.2, as defined in the IEC 61400-3-1:2019 (2019) standards. DLC 1.2 was used because it embodies the loads resulting from turbulence that occurs during the normal operation of a wind turbine throughout its lifetime, and which contribute to the fatigue of turbine components. The FOWTs were both modelled in a water depth of 200 m and under irregular wave conditions. The wind and waves were aligned in the same direction,

propagating directly towards the FOWTs as these conditions impart the greatest loads upon the FOWTs while also being the wind-wave alignment conditions that occur most frequently. Moreover, wind-wave misalignment was not studied in this work because the FOWT hydrodynamic excitation files (used by HydroDyn within OpenFAST) pertaining to second-order wave loads for wave heading directions other than parallel to the turbines were not available for the FOWT models studied. These would have been required to be calculated within an external hydrodynamic solver, such as WAMIT (Wave Analysis at MIT, see C. H. Lee & Newman, 2006), which was outside of the scope of this study. H_s and T_p were adjusted to suit each mean wind speed integer, with values interpolated from those defined by Allen et al. (2020), and shown in Table 5.1.

Table 5.1: IEC DLC 1.2 NTM wind and wave conditions, from Allen et al. (2020).

$\bar{V}_{hub}$, ms^{-1}	13	15	17	19	21	23	25
H_s, m	2.0	2.4	2.8	3.3	3.8	4.3	4.8
T_p, s	7.5	7.6	7.8	8.3	8.8	9.2	9.7

The datasets obtained from the simulations of the FOWTs performed in this study can be found via links to data repositories referenced in Appendix C. Screenshots of example OpenFAST input files used when performing the simulations can be found in Appendix D.1.

5.2 Initial floating offshore wind turbine comparisons

This Section will provide an overview of the initial comparisons formed between the two FOWTs in terms of their FB-only controller behaviour and whether the findings indicated that restrictions were required to be applied to the FF controllers to avoid adverse impacts.

5.2.1 Feedback-only control

Figure 5.1 presents an excerpt of time series results comparing the behaviour of an FB-only controllers (which were the respective baseline controllers that the FF controllers were compared to) of the VolturnUS-S and the WindCrete under the same wind and wave conditions. The ranges recorded during the ten-minute excerpt shown in Fig. 5.1 for each of the parameters of each FOWT are shown in Table 5.2.

From Fig. 5.1 and Table 5.2, it is clear that, overall, the VolturnUS-S was more dynamic in its motions, with higher magnitude oscillations observed in the platform pitch, yaw, roll and sway, and higher frequency oscillations in heave. These greater motions resulted in a higher range in the VolturnUS-S's nacelle acceleration (by 87%), compared to the WindCrete's, despite utilising the floating FB damping loop.

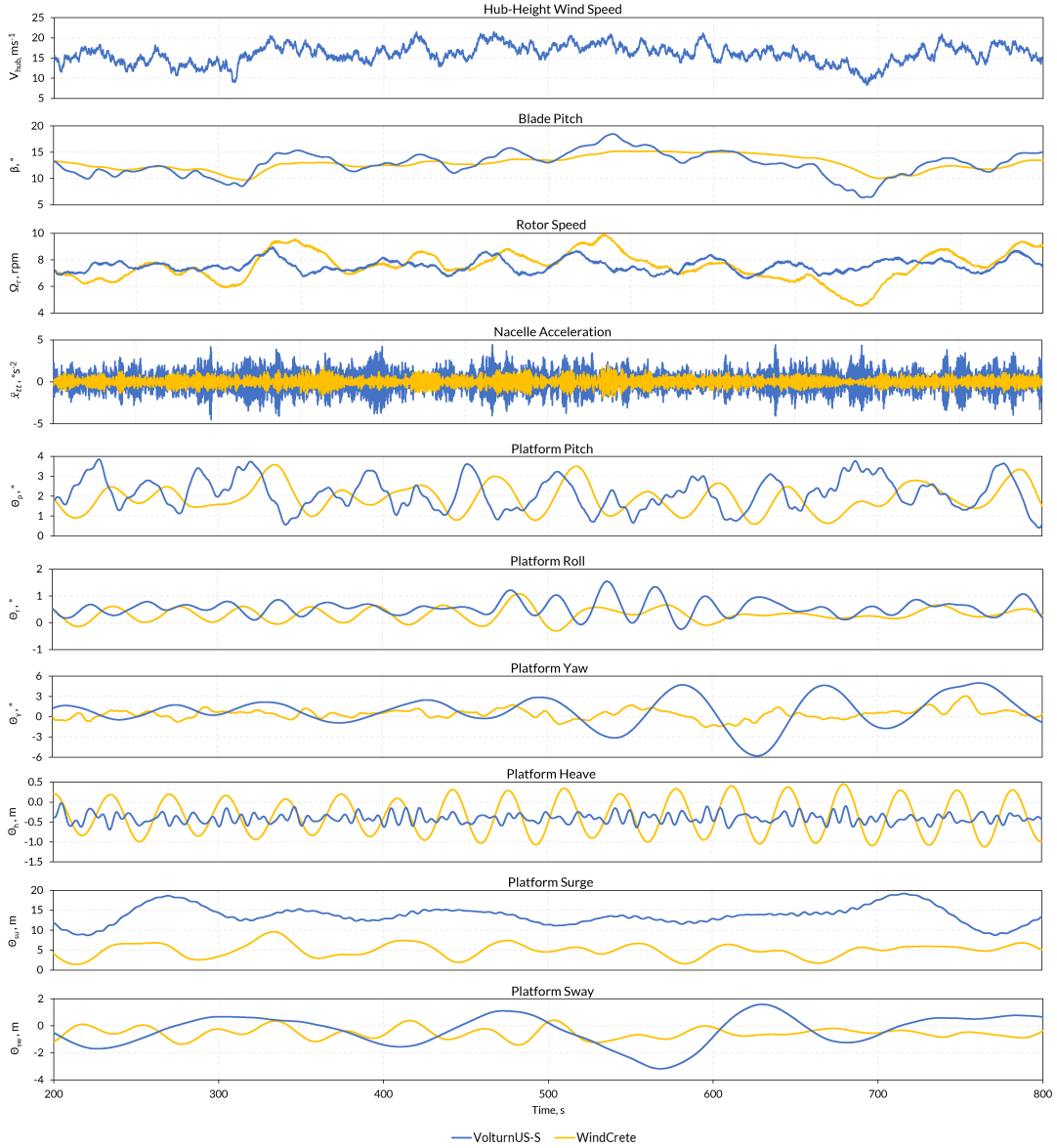


Figure 5.1: Ten-minute excerpt of time series results comparing the FB-only pitch controller performance of the two FOWTs. Simulations were performed under turbulent wind ($\bar{V}$ = 17 ms^{-1}, NTM, TI Category B) and irregular wave (H_s = 2.83 m, T_p = 7.85 s) conditions.

Another clear observation pertinent to the FF control implementation was that the WindCrete's FB-only control mean blade pitching rate was significantly slower (by 22%) than the VolturnUS-S's. This is because the responsiveness of an FB-only control system is dependent upon the proportional and integral gains of its PI controller. The average proportional and integral controller gains of the VolturnUS-S were found to be 4.17 times and 21.9 times greater than those of the WindCrete, respectively. This led to a significantly more responsive FB-only controller for the VolturnUS-S, as reflected in the time series plot of blade pitch variation, shown in Fig. 5.1. The WindCrete's pitch actuation frequency was required to be

less than that of the VolturnUS-S because it has a lower platform pitch natural frequency (0.024 Hz compared to 0.036 Hz for the VolturnUS-S, as documented in Table A.3), meaning that the controller was detuned to a greater degree by Mahfouz et al. (2020) to avoid coupling with the platform pitch motion. The consequence of this was that the FB-only controller of the WindCrete was less effective at maintaining its rotor speed at its setpoint, exhibiting a 71% greater range than the VolturnUS-S. The comparison of the two baseline turbines indicated that the VolturnUS-S had greater scope for motion reductions, while there was greater opportunity to improve the rotor speed regulation of the WindCrete through LAC.

Table 5.2: Ranges recorded during the ten-minute excerpts shown in Fig. 5.1 of each parameter of each FOWT.

Parameter	VolturnUS-S	WindCrete
Blade pitch (β), $^\circ$	12.8	6.80
Rotor speed (Ω_r), rpm	3.17	5.42
Nacelle acceleration ($\ddot{x}_{tt}$), $^\circ s^{-2}$	9.94	5.32
Platform pitch (Θ_p), $^\circ$	4.35	4.34
Platform roll (Θ_r), $^\circ$	3.10	1.79
Platform yaw (Θ_y), $^\circ$	10.8	5.86
Platform heave (Θ_h), m	1.21	1.58
Platform surge (Θ_{su}), m	10.5	9.39
Platform sway (Θ_{sw}), m	5.05	3.13

The differences in baseline performance were also examined through power spectral density (PSD) plots (Fig. 5.2). The PSD quantifies the energy present in the signal as a function of frequency, showing how the signal's power is distributed over different frequency components.

Figure 5.2 reinforces the time series trends. The activity of the blade pitch of the WindCrete (Fig. 5.2b) was significantly lower than that of the VolturnUS-S, leading to greater energy in the rotor speed (Fig. 5.2a) caused by wind loading at low frequencies (< 0.03 Hz, per Fig. 1.8). However, the increased blade pitch activity of the VolturnUS-S led to greater energy in the blade root bending moment than the WindCrete (Fig. 5.2d) at frequencies pertinent to turbulent wind and the 1P/wave frequency (0.126 Hz).

Parallels can be drawn between the tower base side-to-side and fore-aft bending moment (Figs. 5.2e and f) responses of the two FOWTs, which are linked to the natural frequency peaks seen in the platform pitch (Fig. 5.2g) and roll (Fig. 5.2h), respectively.

Differences in the natural frequencies of the other platform degrees of freedom (DOFs) can be observed in Fig. 5.2. Namely the platform sway (Fig. 5.2l), which occurred at a higher frequency for the WindCrete, but with lower energy compared to the VolturnUS-S, which also experienced much greater energy in the platform yaw motion (Fig. 5.2i). Differences were also

seen in the platform heave (Fig. 5.2j), which had a lower natural frequency for the WindCrete than the VolturnUS-S. The peak seen in the platform heave of the WindCrete was much greater than that of the VolturnUS-S due to its significantly smaller water-plane area rendering it more sensitive to heave motions.

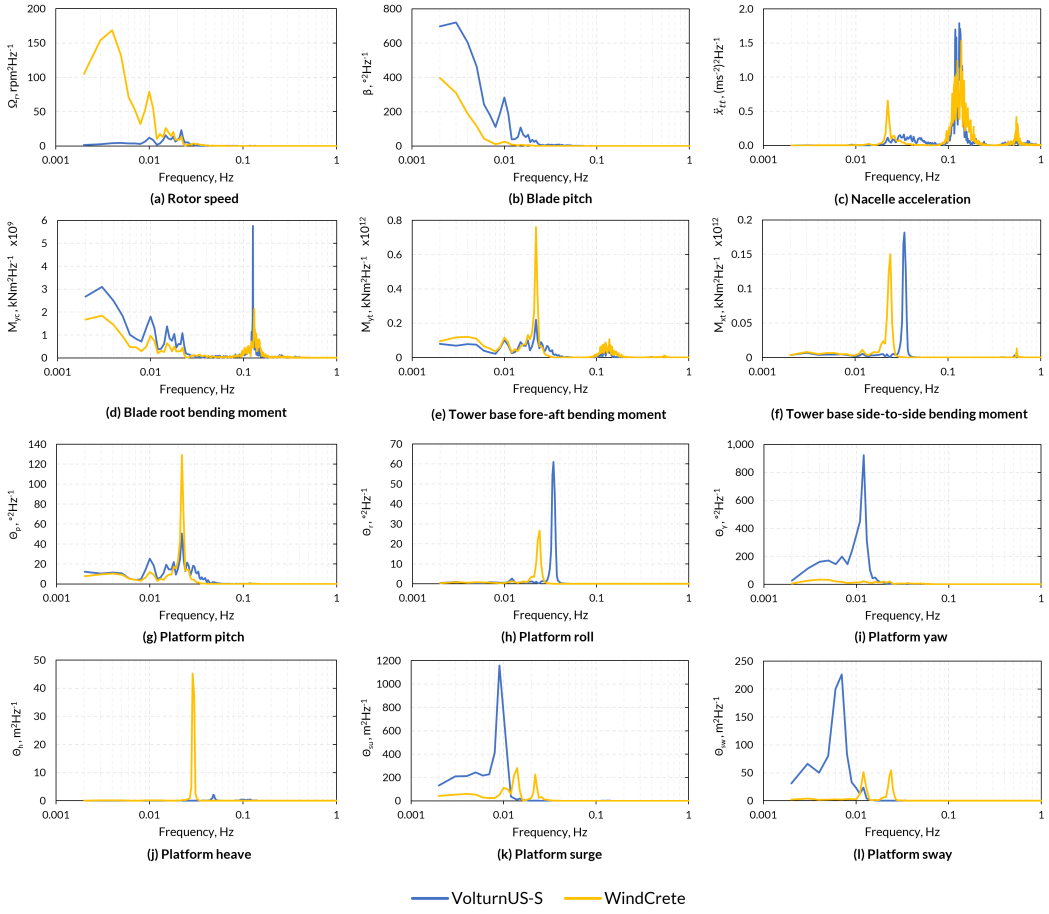


Figure 5.2: PSD plots generated from one-hour simulations of the two FOWTs using FB-only control under turbulent wind ($\bar{V}$ = 17 ms^{-1}, NTM, TI Category B) and irregular wave (H_s = 2.83 m, T_p = 7.85 s) conditions.

5.2.2 Feedforward control restriction requirements

While studying the FF control responses of the WindCrete it was found that, when the turbine was required to switch between control regions as wind speeds crossed the below-rated-to-above-rated transition point, there were detrimental impacts on numerous key turbine and substructure parameters as a consequence of its comparatively (to the FFCPC) slow FB-only controller. Time series results (at $\bar{V}$ = 13 ms^{-1}) of the two FOWTs using FB-only control or FFCPC are shown in Fig. 5.3.

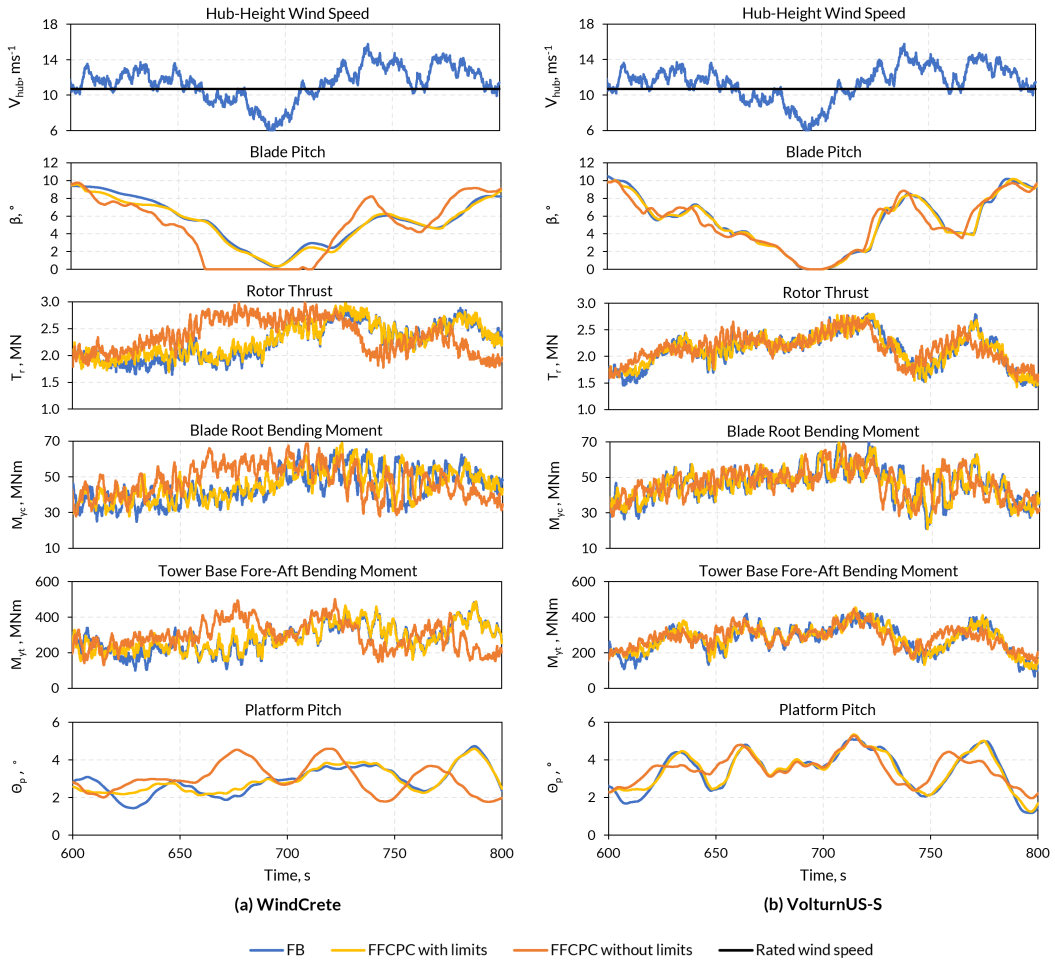


Figure 5.3: Time series plots showing the impact of applying or not applying limits to the FFCPC commands in measured wind speeds below 13 ms^{-1} for both FOWTs subjected to turbulent wind ($\bar{V}$ = 13 ms^{-1}, NTM, TI Category B) and irregular wave (H_s = 2.01 m, T_p = 7.45 s) conditions. The outer LIDAR beams were positioned at 50% rotor radial span for the FFCPC strategy for both FOWTs.

Fig. 5.3 shows that the hub height wind speed decreased to below the rated wind speed (10.69 ms^{-1}, indicated by the black horizontal line in Fig. 5.3) at approximately 660 seconds. The LIDAR detected this decrease in wind speed in advance and caused the FFCPC to act suddenly to bring the blade pitch angle to zero. The subsequent behaviour showed that the use of FFCPC by the WindCrete (Fig. 5.3a) in the transition between the above- and below-rated control regions induced responses akin to the negative damping problem explained in Sect. 1.3.5. The sudden reduction in the blade pitch angle led to increases in the thrust force, the blade root bending moment, the platform pitch and the tower base fore-aft bending moment. These increases did not occur when using FB-only control because the control actuation was significantly slower when using this strategy, meaning that the pitch angle

decreased more gradually when transitioning to the below-rated region, which helped to better maintain the normal levels seen in the parameters. The inverse of this was true when the wind speed increased to greater than the rated wind speed at approximately 715 seconds in Fig. 5.3, as the blade pitch was increased suddenly when using the FFCPC, subsequently decreasing the thrust force, the platform pitch and the tower base fore-aft bending moment. Ultimately, the unrestricted use of FFCPC by the WindCrete in this region led to increased instability of the FOWT.

To overcome these adverse impacts, in measured wind speeds below 13 ms^{-1}, the FF pitch rate from the FFCPC and the individual pitch commands from the FFIPC were set to zero for the WindCrete. This ensured that, in wind speeds close to the rated speed, the FF controllers did not provide any input to the control system, meaning that the control system could respond to these wind speeds by only using the FB controller, ensuring similar responses and avoiding the increases to the loadings and motions, as indicated in Fig. 5.3a.

Similar methodologies were also applied to the VolturnUS-S to investigate whether it also required the FF control action in low above-rated wind speeds to be restricted, with time series results shown in Fig. 5.3b. However, it was found that, due to the superior responsiveness (compared to the WindCrete) of the VolturnUS-S's FB-only controller, it was better able to follow the blade pitch changes driven by FFCPC and so the FFCPC without limits did not lead to increased loads or motions of the VolturnUS-S. When limits were applied to the VolturnUS-S model using FFCPC, the responses were almost identical to those when using FB-only control, but this reduced the benefits brought by the FFCPC overall. Therefore, it was concluded that the limits were not required to be applied to the FFCPC or FFIPC within the VolturnUS-S model.

5.3 Key statistical metrics

The benefits of LAC can be assessed by statistical metrics. The metric most commonly used is the standard deviation (SD), which provides knowledge of the degree of variation within the data set

$$SD = \sqrt{\frac{\sum(\alpha_i - \mu)^2}{N}} \tag{5.1}$$

where α_i is the value of a data point, i, μ is the mean of all values within the dataset and N is the number of data points. The standard deviation is the most common statistical metric used to assess the performance of LAC, particularly in terms of fatigue loadings, because fatigue is driven by the repeated cyclic loading of components. Therefore, reducing the magnitude of the cyclic loading variations around the mean, characterised by reductions in the standard deviation, is a more appropriate way of quantifying the LAC benefit than comparing the mean values, for example.

Similarly to the standard deviation, the root mean squared error (RMSE) metric can be applied to assess the ability of the controller to maintain the control variables (namely the rotor speed and power) at their desired setpoints by quantifying their mean error

$$RMSE = \sqrt{\frac{\sum(\alpha_i - \hat{\alpha})^2}{N}} \tag{5.2}$$

where $\hat{\alpha}$ is the setpoint of the parameter. The root mean square (RMS) can be used to quantify the mean magnitude of a varying parameter, and is useful for analysis of parameters that fluctuate between positive and negative values.

$$RMS = \sqrt{\frac{\sum \alpha_i^2}{N}} \tag{5.3}$$

For bending moments, the DELs can also be compared. DELs equate to the load experienced by the turbine structure/component over a period of time that is equivalent to the damage caused by a specific event. The other metrics used to compare the time series results of the controllers were the ranges, maximum values and rates of change. Table 5.3 highlights the key performance indicating metrics that were assessed for the various turbine and substructure parameters, with rationale explaining their significance.

Results will be presented as normalised values, whereby the values were divided by that of the respective FB-only controller (X_{FB}) of each FOWT.

$$X_{norm,i} = \frac{X_i}{X_{FB}} \tag{5.4}$$

where X_i is the value of a metric for a given parameter associated with a control strategy, i (FB-only, FFCPC, FFCPC+FBIPC or FFCPC+FFIPC), and $X_{norm,i}$ is the normalised value of a metric for a given control strategy. This approach allowed for direct comparison to the results of the FB-only controller, which were normalised to one. Therefore, a value that is less than one indicates a reduction (and therefore, generally, an improvement) compared to the FB-only controller and a value that is greater than one indicates an increase (and therefore, generally, a deterioration) compared to the FB-only controller.

Table 5.3: Key performance indicating metrics that were assessed for the specified parameter and their significance in demonstrating the efficacy of LAC.

Parameter	Metric	Significance
Rotor speed	RMSE	This metric can assess how closely the rotor speed is maintained at its rated value (7.56 rpm). A lower error therefore indicates that the pitch controller is better able to deliver its control objective.
	SD/Range	The SD and range of the rotor speed determine the 1P and 3P constraint bands, which are important for substructure and tower design (see Eq. 1.11). Allen et al. (2020) reported that, for the 15 MW VolturnUS-S, the constraining operational rotor speed range was 3.97 to 8.86 rpm, which impacted upon the tower design. Reductions to the 1P and 3P frequency constraint bands may allow for the use of a soft-stiff tower, which will be a lighter and cheaper option than a stiff-stiff tower.
	Maximum	FOWTs are prone to over-speeding by up to 30% greater than the rated rotor speed (see Fischer, 2013; T. Larsen & Hanson, 2007), which can lead to shutdowns and damage to turbine components.
Power	RMSE	This metric can assess how closely the generator power is maintained at its rated value (15 MW) in above-rated wind speeds. Significant variation around the setpoint can lead to excessive stress on the generator.
Blade pitch	SD	Decreased SDs indicate a reduced degree of pitch actuation and, therefore, less demand on the pitch actuation mechanisms.
	RMS of rate	This metric provides information on how quickly the blade pitch angle changes between measurements and can provide further indication of the demand on the pitch actuation mechanisms. The RMS will be calculated as this will determine the mean magnitude of the pitch rate, thereby considering both increases and decreases in the rate.
Thrust	SD	A reduced SD in the thrust indicates a reduction in the variation in the force applied to the wind turbine, which impacts upon the pitching of the platform and the tower base fore-aft bending moment.
Bending moments	SD	Reductions indicate that the loadings are more consistent, predictable and uniform across the structure, and can result in reduced fatigue, failure rates and, therefore, lifetime extension.
	DEL	Similarly to the SD, lower DELs indicate reduced stress on turbine components, which can result in reduced fatigue and, ultimately, lifetime extension.
Platform motions	SD	Reductions to the SD indicate reduced variation of platform motions, which can improve stability and lead to performance improvements.
	Maximum (Pitch only)	The maximum magnitude of platform motions, particularly the platform pitch angle, is a design constraint for the turbine-substructure design as it plays a role in dictating the aerodynamic efficiency and the loads at the tower base.

5.4 VolturnUS-S simulation results

This Section outlines the results of the OpenFAST simulations of the VolturnUS-S. Time series results of a single seed with a mean wind speed of 17 ms^{-1} will first be shown to highlight the improvements to the turbine and substructure parameters when using the FFCPC or FFCPC+FFIPC compared to the FB-only controller. Results of the FFCPC+FBIPC will also be presented to compare the effectiveness of the FB and FF IPC elements when combined with FFCPC. Subsequently, four-seed-average normalised results of simulations with a mean wind speed of 17 ms^{-1} will be provided for the key statistical metrics outlined in Table 5.3. 17 ms^{-1} was chosen as the mean wind speed when presenting these results because it lies close to the middle of the interval between the rated (10.69 ms^{-1}) and cut-out (25 ms^{-1}) wind speeds of the IEA 15 MW reference turbine, and so a range of wind speeds spanning the above-rated spectrum occurred within the simulations at this mean wind speed. Therefore, this offered a

good representation of the overall performance of the controllers. Four-seed-average results from across the above-rated spectrum will then be given to further demonstrate the capability of the FF controllers. These results will then be compared to those obtained from the LAC implementation within the WindCrete, presented in Sect. 5.5.

5.4.1 Single wind field

Figure 5.4 presents an excerpt of time series results of the selected turbine and substructure parameters of the VolturnUS-S. Fig. 5.5 shows spider-web plots of impacts upon the standard deviations of various parameters. Table 5.4 provides a summary of the average normalised values of the other metrics relevant to various parameters delivered by the FF control strategies relative to the FB-only controller across four random seed one-hour length simulations.

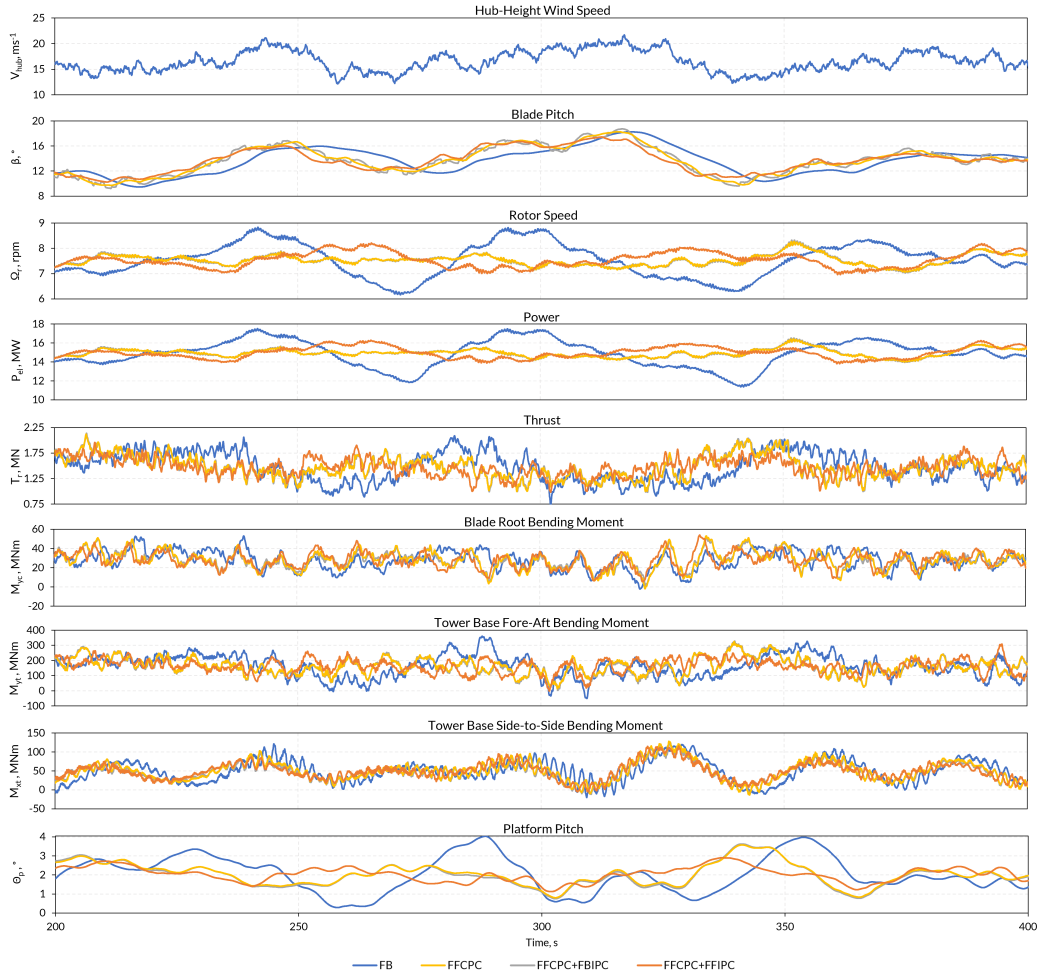


Figure 5.4: 200-second excerpt of time series results comparing the FB-only pitch controller to the FF controllers of the VolturnUS-S. Simulations were performed under turbulent wind ($\bar{V}$ = 17 ms^{-1}, NTM, TI Category B) and irregular wave (H_s = 2.83 m, T_p = 7.85 s) conditions. The outer LIDAR beams were positioned at 50% rotor radial span for all LAC strategies.

As can be seen from Fig. 5.4, Fig. 5.5 and Table 5.4, the collective and individual pitch control additions were able to deliver benefits to the key metrics of numerous turbine and substructure parameters. The rotor speed and generator power benefitted most significantly, with reductions of up to 54% in the RMSEs compared to the FB-only controller, demonstrating improved power tracking, and reduced ranges of 1P and 3P frequencies, by up to 46%, relative to the FB-only controller.

These benefits are attributed to the superior blade pitch preparation in advance of the wind's impact. This had a damping effect on the rotor's motion, thereby reducing its variation compared to the FB-only controller. However, to achieve this, the mean and RMS rate of change of the blade pitch were increased (compared to the FB-only controller) especially when using FFCPC+FBIPC (by 286%), which indicates an increased demand on the pitch actuation mechanisms. However, when using FFCPC or FFCPC+FFIPC, the mean pitch rate was only increased by approximately 0.06 $^{\circ}s^{-1}$ compared to the FB-only controller, and so the actual increase in the pitch rate was not as significant as the normalised value indicates. Despite the increases to the pitch rate, the standard deviation of the blade pitch angle was not affected significantly and was slightly reduced (by 5%) when FFCPC+FFIPC was employed (relative to when using FB-only control).

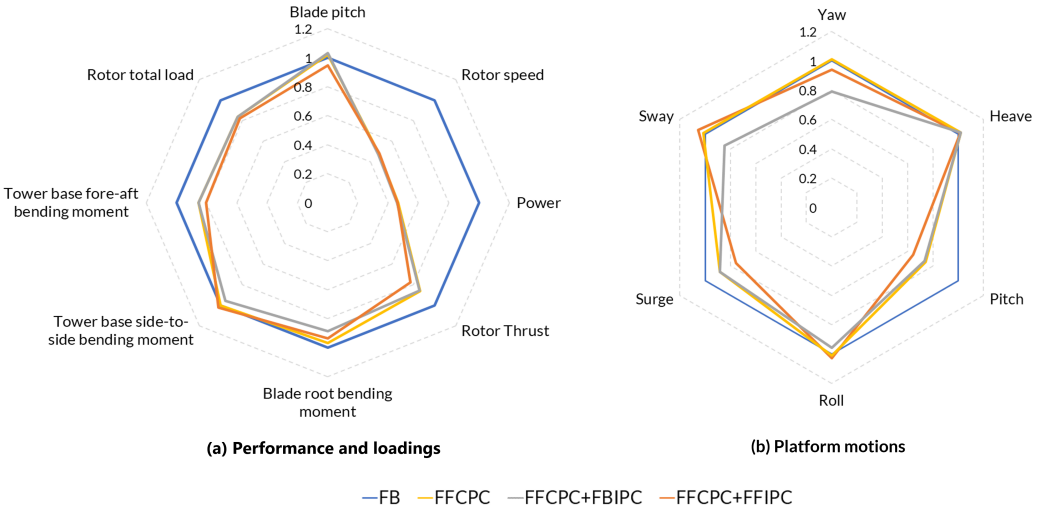


Figure 5.5: Spider-web plots showing the four-seed-average standard deviations of various parameters of the VolturnUS-S using the different FF control strategies under turbulent wind ($\bar{V}$ = 17 ms^{-1}, NTM, TI Category B) and irregular wave (H_s = 2.83 m, T_p = 7.85 s) conditions, normalised to those of the FB-only controller. The outer LIDAR beams were positioned at 50% rotor radial span for all LAC strategies.

Table 5.4: Four-seed-average metric values of various parameters of the VolturnUS-S using the different FF control strategies under turbulent wind ($\bar{V}$ = 17 ms^{-1}, NTM, TI Category B) and irregular wave (H_s = 2.83 m, T_p = 7.85 s) conditions, normalised to those of the FB-only controller. The outer LIDAR beams were positioned at 50% rotor radial span for all LAC strategies.

Parameter	Metric	Normalised FB	Normalised FFCPC	Normalised FFCPC+FBIPC	Normalised FFCPC+FFIPC
	RMSE	1.00	0.48	0.47	0.48
Rotor speed	Range	1.00	0.54	0.54	0.56
	Maximum	1.00	0.93	0.93	0.92
Power	RMSE	1.00	0.47	0.46	0.46
Blade pitch	RMS Rate	1.00	1.25	3.86	1.28
Blade root	DEL	1.00	1.00	0.93	0.96
Tower base F-A	DEL	1.00	0.91	0.92	0.89
Tower base S-S	DEL	1.00	0.98	0.96	1.00
Platform pitch	Maximum	1.00	0.90	0.89	0.82

The standard deviations of the bending moments acting on the rotor, blade roots and tower base were reduced, by up to 20%, which led to reductions in the DELs, by up to 11%, compared to the FB-only controller. The standard deviation of the blade root loads was reduced most (by 11%, relative to the FB-only controller) when combining the FFCPC with the traditional FBIPC (FFCPC+FBIPC) because maintaining the blade root bending moments as constant is a control objective of the FBIPC approach, as was explained in Sect. 4.3.

In terms of platform motions, the greatest beneficiary was the pitch DOF, where an up to 36% reduction in its standard deviation compared to the FB-only controller was recorded. This was expected because the standard deviation of the thrust force was reduced by up to 22%, meaning that a more consistent force was applied to the rotor when using FF control instead of FB-only control, resulting in reduced variation in the force pushing the turbine backwards. This led to the reduced variations seen in the pitching of the platform as well as the fore-aft bending moment acting on the tower base, relative to the FB-only controller.

Overall, the FFCPC+FFIPC strategy further enhanced the FFCPC by providing additional tuning to the pitch of the individual blades. Performance parameters benefitted from up to 8% further reductions in their standard deviations compared to the FB-only controller, although the benefits to the rotor speed and power were not enhanced further. The blade root and tower base fore-aft bending moments benefitted from further 3% and 5% reductions in their standard deviations compared to the FB-only controller, respectively. The most significant improvement delivered by FFCPC+FFIPC to the platform's motions was in surge, which achieved a further 12% reduction in standard deviation over the FFCPC compared to the FB-only controller, while the platform pitch observed a 10% further reduction in its standard deviation and an 8% further reduction to its maxima, compared to the FB-only controller, respectively. The FFIPC

contributions allowed for optimisation of the blades' pitch angles by dynamically adjusting them according to the LIDAR-measured azimuth-specific wind speeds. As the rotor rotated, the FFIPC continuously ensured that each blade was pitched appropriately for the varying wind speeds encountered at different azimuth positions. By accounting for the specific wind speed at each azimuth, the FFIPC effectively reduced the adverse effects of wind turbulence and shear across the rotor-swept area, leading to reductions in the variation in the thrust force acting upon the rotor and, by extension, reducing the variations in the tower base fore-aft load and the pitch and surge motions of the platform.

The FFCPC+FFIPC strategy also offered improvements upon the traditional FBIPC strategy, although both excelled in different areas. When combined with FFCPC, FFIPC gave greater improvements over the baseline to the rotor thrust (by 6%) than when FBIPC was used, while also greatly benefiting the standard deviations of the platform pitch (by 9%) and surge motions (by 12%) in addition to the tower-base fore-aft bending moment (by 5%). These benefits were also achieved without vast increases to the RMS of the blade pitching rate over the baseline (28% for FFCPC+FFIPC), as were observed with the FBIPC, which instead increased the RMS of the blade pitching rate by 286%, relative to the FB-only controller. However, FBIPC was able to offer benefits to the standard deviations over the FFIPC to the blade root and tower side-to-side bending moments (by 5% and 6%, respectively) as well as the yaw and sway motions of the platform (by 15% and 20%, respectively), relative to the FB-only controller.

The trends shown in the time series data can be further examined through the use of PSD plots. Figure 5.6 presents PSD plots of key turbine and substructure parameters obtained from one-hour length simulations at a mean wind speed of 17 ms^{-1}.

Overall, the FF control implementation was able to significantly reduce the energy seen in the responses of the parameters (relative to the FB-only controller) at numerous frequencies. Most notably, reductions occurred at frequencies most pertinent to wind loading (> 0.03 Hz) and at the pitch (0.022 Hz) and surge (0.01 Hz) natural frequencies of the platform. This was particularly evident for the rotor speed (Fig. 5.6a), where all FF control strategies were able to reduce the energy of the responses, relative to the FB-only controller, at these frequencies.

The FFCPC+FFIPC was able to slightly reduce the energy in the blade pitch (Fig. 5.6b) due to wind at low frequencies pertinent to wave loading, as well as at the surge natural frequency, relative to the FB-only controller. The addition of FBIPC resulted in a peak at the 1P/wave frequency (0.126 Hz), which was much less significant for the FFIPC addition.

The spectra of the rotor thrust (Fig. 5.6c), tower base fore-aft bending moment (Fig. 5.6e) and platform pitch (Fig. 5.6g) were similar, with the FFCPC+FFIPC offering the greatest reductions in the energy, compared to the FB-only controller, at frequencies pertinent to wind loading and the pitch and surge natural frequencies. The FFIPC addition also delivered reduced energy in the response of the platform surge (Fig. 5.6h) at its natural frequency and under wind loading, relative to all of the other control strategies.

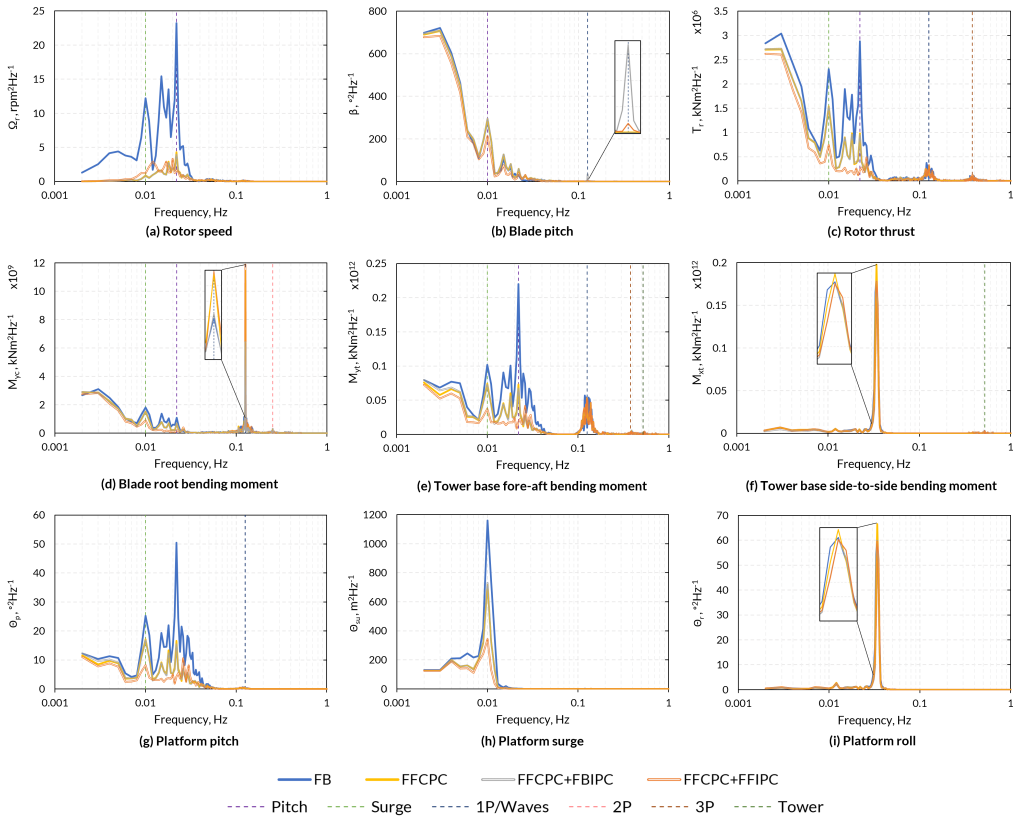


Figure 5.6: PSD plots of the turbine and substructure parameters generated from one-hour simulations of the VolturnUS-S subjected to turbulent wind ($\bar{V}$ = 17 ms^{-1}, NTM, TI Category B) and irregular wave (H_s = 2.83 m, T_p = 7.85 s) conditions. The outer LIDAR beams were positioned at 50% rotor radial span for all LAC strategies. Dashed vertical lines indicate pertinent frequency points.

The tower base side-to-side bending moment (Fig. 5.6f) and the platform roll (Fig. 5.6i) both exhibited a peak at the platform roll natural frequency, though there was little difference between the responses when using FB-only controller or the FF controllers, with the exception of a small increase with FFCPC.

Figure 5.6d reinforces that the main advantage of the FBIPC addition was its ability to offset the increase in the energy of the blade root bending moment at the 1P frequency. The FFCPC on its own and combined with FFIPC resulted in a significant increase in the energy at the 1P frequency, showing inferiority to the FBIPC for rejecting the cyclic loads on the blade roots, but was instead able to reduce the energy (compared to the FB-only controller) at the pitch and surge natural frequencies.

5.4.2 Above-rated wind spectrum

Simulations were performed across the above-rated wind spectrum from 13-25 ms^{-1} in intervals of 2 ms^{-1}. This aimed to capture the performance benefits attainable from FFCPC with FBIPC or FFIPC across the full operating range of the pitch controller to allow for the assessment of lifetime benefits. The four-seed-average above-rated wind speed results for various turbine and substructure parameters are shown in Fig. 5.7.

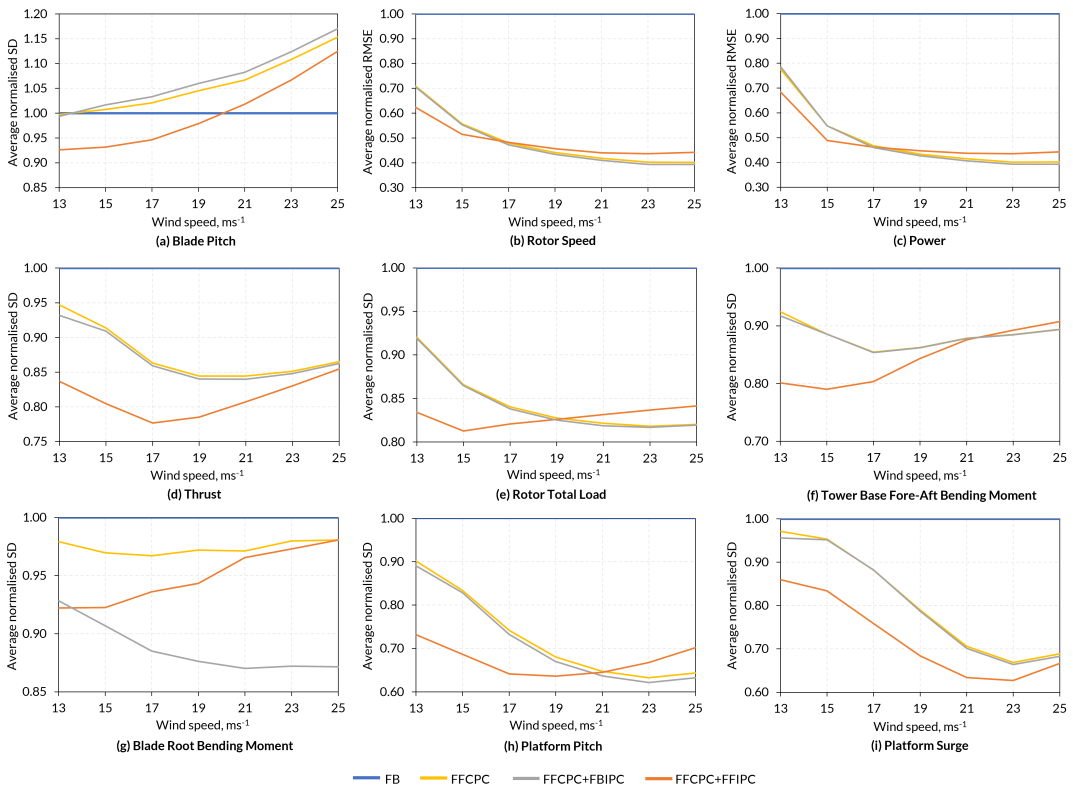


Figure 5.7: SD or RMSE values delivered by the FF controllers compared to the FB-only pitch controller of the VolturnUS-S across above-rated wind speeds. The LIDAR beams were positioned at 50% rotor radial span for all LAC strategies. The results shown are the four-seed-average normalised (to the FB-only controller) values from the one-hour simulations at each mean wind speed integer for each control configuration.

Figure 5.7 re-iterates the enhancements to FFCPC provided by the addition of the FFIPC and FBIPC controllers, as these were able to deliver greater standard deviation reductions over the FB-only controller for the majority of the above-rated wind speeds. The improvements over the FB-only controller delivered by the FF controllers to the turbine and substructure parameters (except for the rotor total load [Fig. 5.7e] and the blade root [Fig. 5.7g] and tower base fore-aft bending moments [Fig. 5.7f]) were least significant at lower above-rated mean wind speeds closer to the above-below-rated transition point (13 ms^{-1} and 15 ms^{-1}). This is because, as these were mean wind speeds, part of the simulations were spent with wind speeds below the rated wind speed, where the FF controllers did not operate, meaning that responses were almost identical to the FB-only controller under these conditions (as shown in Fig. 5.3b), thereby reducing the differences between the datasets.

As the wind speeds increased, some parameters experienced higher magnitude oscillations when using the FB-only controller (the standard deviations increased by up to 73% between mean wind speeds of 13 ms^{-1} and 25 ms^{-1}, as shown in Table E.1) and so this led to more room for improvement for the FF controllers. This was true for the rotor speed (Fig. 5.7b) and, by extension, the power (Fig. 5.7c), which benefitted most significantly, with reductions in their RMSEs by up to 60% at a mean wind speed of 25 ms^{-1}, compared to the FB-only controller. This came at the expense of increased variation in the blade pitch, relative to the FB-only controller, as the wind speed increased, by up to 17% greater than the baseline at a mean wind speed of 25 ms^{-1}, though FFCPC+FFIPC was able to reduce this increase, by up to 5%.

The rotor thrust (Fig. 5.7d) saw reductions in its standard deviation delivered by the FF controllers over the FB-only controller increase until approximately 17-19 ms^{-1}, before decreasing at higher wind speeds due to the diminishing standard deviation of the rotor thrust as the wind speed increased (by 28% between 13 ms^{-1} and 25 ms^{-1}, as shown in Table E.1) when using FB-only control. This is because the hydrodynamic forces dominate over the aerodynamic forces at higher wind speeds. This response was mirrored within the loadings and platform motions, as reductions to the standard deviations of the rotor total load (Fig. 5.7e), the tower base fore-aft bending moment (Fig. 5.7f), the platform pitch (Fig. 5.7h) and platform surge (Fig. 5.7i), relative to the FB-only controller, also diminished at higher wind speeds due to the decreased reduction to the standard deviation of the rotor thrust. The benefit to the blade root bending moment (Fig. 5.7g) over the FB-only controller remained relatively consistent across the full above-rated wind spectrum when using FFCPC. However, greater improvements were observed when pairing FFCPC with FBIPC, as maintaining the blade root bending moments constant is a control objective of this approach, as was explained in Sect. 4.3.

5.4.3 Comparison of beam span locations

Figure 5.8 shows the above-rated wind speed results of the simulations when using beams focused at 30%, 50% and 70% rotor radial span using FFCPC+FFIPC.

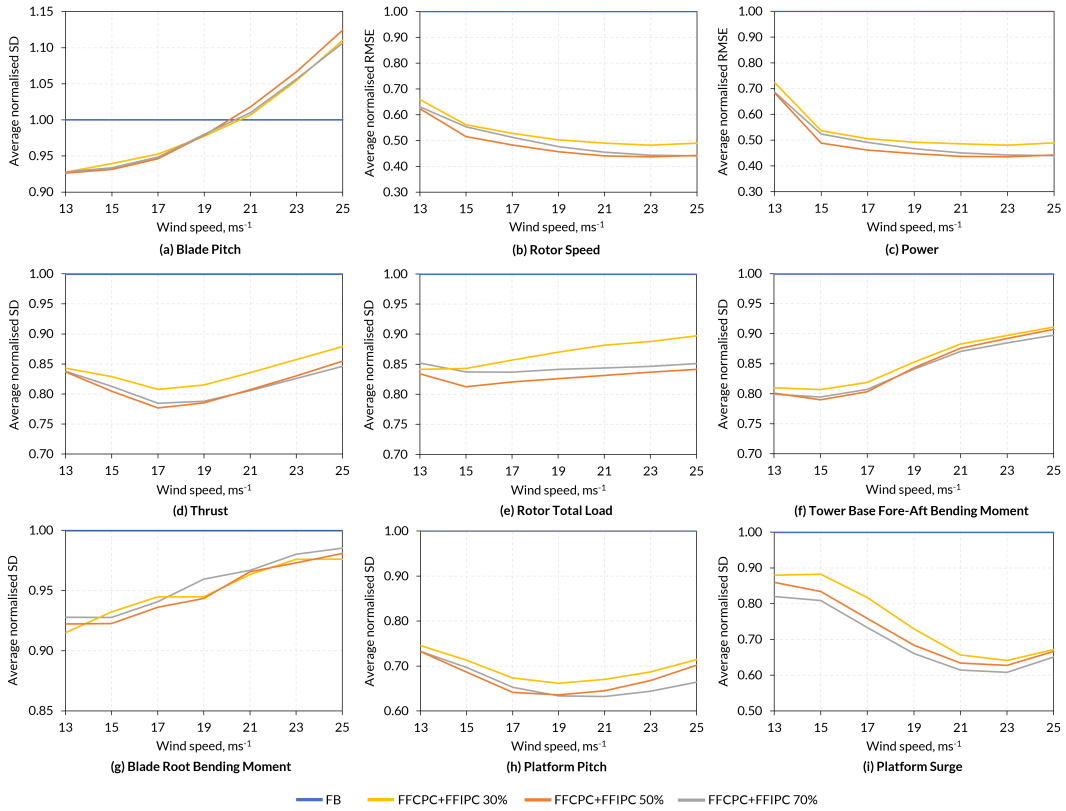


Figure 5.8: Normalised SD or RMSE values delivered by the FFCPC+FFIPC controller with beams positioned at 30%, 50% and 70% rotor radial span compared to the FB-only pitch controller of the VolturnUS-S across above-rated wind speeds. The results shown are the four-seed-average normalised values from the one-hour simulations at each mean wind speed integer for each control configuration.

The FFCPC+FFIPC performed marginally better across the above-rated wind spectrum when the beams were focused at 50% or 70% than at 30% span. At 30% span, the benefits to the standard deviations of rotor speed (Fig.5.8a), power (Fig. 5.8b), rotor thrust (Fig. 5.8d), rotor total load (Fig. 5.8d) and platform pitch (Fig. 5.8h), compared to the FB-only controller, were all slightly less significant, by up to 5%. The FFCPC+FFIPC performed similarly with beams positioned at 50% and 70% span. However, there were marginally greater benefits delivered to the standard deviations over the FB-only controller when the beams were focused at 50% span compared to at 70% span, namely in the rotor speed (Fig. 5.8b), power (Fig. 5.8c), rotor total load (Fig. 5.8e) and blade root bending moment (Fig. 5.8g), by up to 4%.

The differences seen in the performance of the FFIPC at various rotor radial spans can be explained through consideration of the variation in the wind speed across the rotor-swept area as the span positions increase. At 70% span, the distance between the opposite beams was approximately 170 m, compared to 120 m at 50% span and 75 m at 30% span. As a single mean wind speed was used by the FFCPC, the potentially larger differences in wind speed at the measurement locations impacted upon the REWS used and so this lead to sub-optimal commands issued by the FFCPC. Similarly, at the smallest span length simulated (30%), the variation in the measured wind velocities may not have been sufficient to represent the full rotor-swept area. This was evidenced by the lower coherence between the LIDAR-estimated and turbine-based REWS (detailed in Sect. 4.5), meaning that lower cutoff frequencies were required within the low-pass filter, limiting the FF control action. Furthermore, for the individual control element, it was pertinent to capture the variation across the rotor-swept area in order to provide the appropriate command to the individual blades. Therefore, with the beams positioned at 50% rotor radial span, the LIDAR was able to capture the wind velocities appropriate for the effective operation of both the collective and individual FF pitch control elements.

5.5 WindCrete simulation results

This Section outlines the results of the simulations of the WindCrete. In a similar manner to the VolturnUS-S, results obtained from a mean wind speed of 17 ms^{-1} will be presented, followed by four-seed-average results from across the above-rated spectrum. When undertaking simulations of the WindCrete, additional time had to be allowed for the settling of start-up transients at high wind speeds due to the slow nature of its FB-only pitch controller. Therefore, at mean wind speeds greater than or equal to 21 ms^{-1}, 4100 seconds of simulation time was performed, with the initial 500 seconds discarded.

5.5.1 Single wind field

Figure 5.9 presents an excerpt of time series results of the selected turbine and substructure parameters of the WindCrete. Figure 5.10 shows spider-web plots of impacts upon the standard deviations of various parameters. Table 5.5 provides a summary of the average normalised values of the other metrics relevant to various parameters delivered by the FF control strategies relative to the FB-only controller across four random seed one-hour length simulations.

As can be seen from Fig. 5.9, Fig. 5.10 and Table 5.5, the FF control additions were also able to deliver benefits to the key metrics of numerous turbine and substructure parameters of the WindCrete. The rotor speed and generator power benefitted most significantly, with reductions of up to 73% in the RMSEs, compared to the FB-only controller, demonstrating improved power tracking, and reduced ranges of 1P and 3P frequencies, by up to 68%.

These benefits are attributed to the superior blade pitch preparation in advance of the wind's impact. Similarly to the VolturnUS-S, this had a damping effect on the rotor's motion, thereby reducing its variation compared to the FB-only controller. However, this had the expense of significantly higher blade pitch variation and RMS rate of change (by up to 664%) than the FB-only controller.

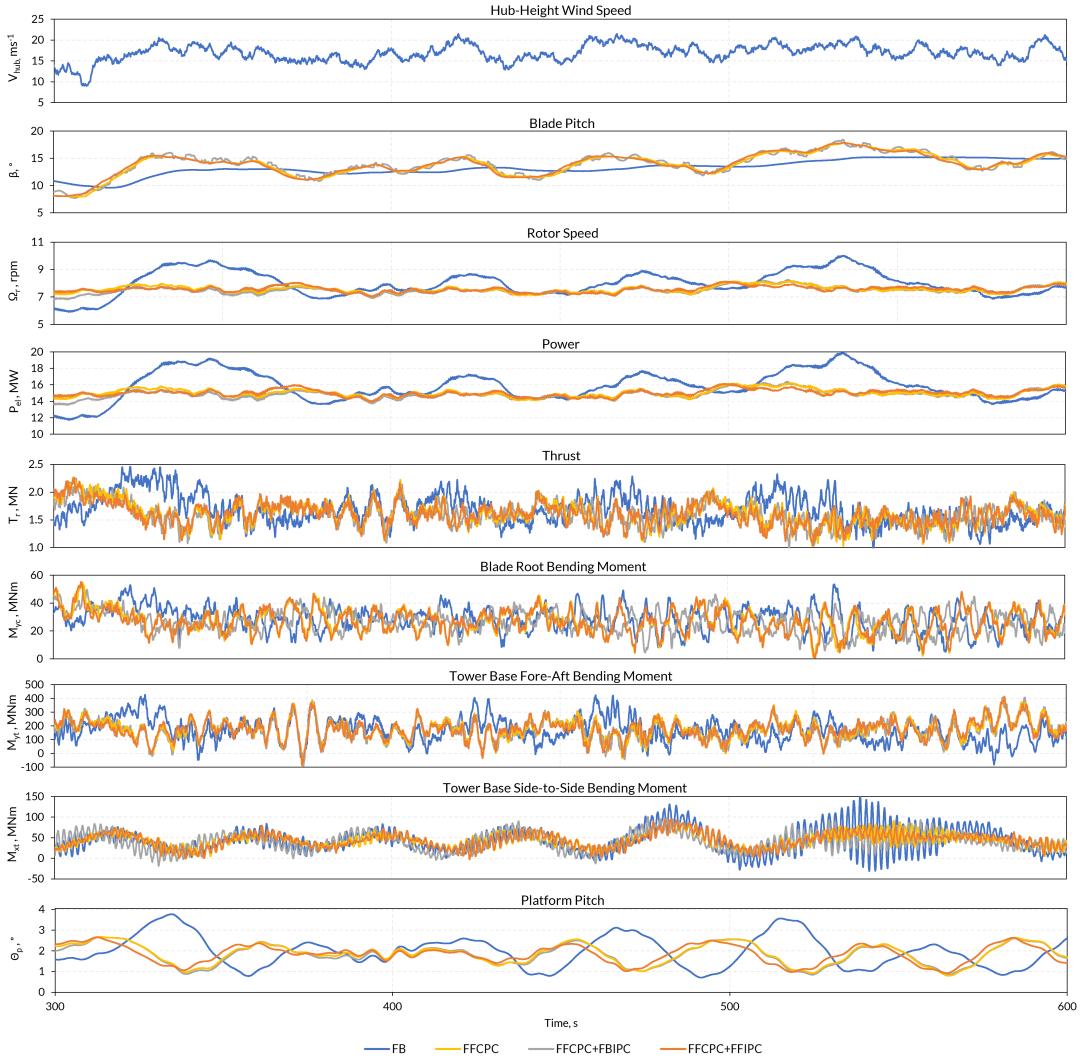


Figure 5.9: 300-second excerpt of time series results comparing the FB-only pitch controller to the FF controllers on the 15 MW WindCrete. Simulations were performed under turbulent wind ($\bar{V}$ = 17 ms^{-1}, NTM, TI Category B) and irregular wave (H_s = 2.83 m, T_p = 7.85 s) conditions. The outer LIDAR beams were positioned at 50% rotor radial span for all LAC strategies.

However, this did not lead to an introduction of negative damping, which had occurred at wind speeds close to the above-to-below-rated transition point before limits were applied to the FF control action (documented in Sect. 5.2). The standard deviations of the bending moments of the tower were also significantly reduced, by up to 15% compared to the FB-only controller, attributed to the reduction in the variation of the thrust force acting on the turbine, by up to 15%. The WindCrete saw significant reductions to the standard deviation (by up to 20%) and DEL (by up to 21%) of the tower base fore-aft bending moment but saw a slight increase in those of the blade root bending moment (by up to 4%) with the FFCPC and FFCPC+FFIPC strategies, relative to the FB-only controller. In terms of platform motions, the greatest beneficiaries were the pitch and surge DOFs, where up to 39% reductions to the standard deviations (compared to the FB-only controller) were recorded. Among the other DOFs, when using the FF control strategies, there were reductions in the standard deviation of the heave and yaw motions, by up to 35%, relative to the FB-only controller.

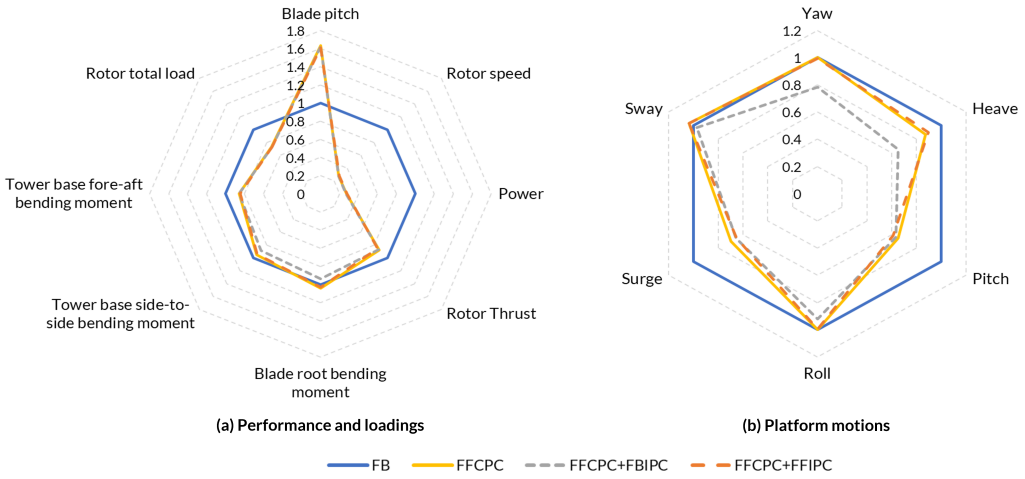


Figure 5.10: Spider-web plots showing the four-seed-average standard deviations of various parameters of the WindCrete using the different FF control strategies under turbulent wind ($\bar{V}$ = 17 ms^{-1}, NTM, TI Category B) and irregular wave (H_s = 2.83 m, T_p = 7.85 s) conditions, normalised to those of the FB-only controller. The outer LIDAR beams were positioned at 50% rotor radial span for all LAC strategies.

The addition of FFIPC to the FFCPC did not allow for significant further reductions to the standard deviations of the key parameters. This is because the FFCPC already delivered significant increases to the blade pitch variation (compared to the FB-only controller) in order to improve the rotor speed regulation and the reduce loadings, and so further tuning of the blade pitch via FFIPC offered little further impact over the FB-only controller. This is evidenced by PSD plots of the tower base fore-aft bending moment, shown in Fig. 5.11, where, in contrast to the VolturnUS-S (Fig. 5.11a), the implementation of FFIPC to the WindCrete

Table 5.5: Four-seed-average metric values of various parameters of the WindCrete using the FF control strategies under turbulent wind ($\bar{V}$ = 17 ms^{-1}, NTM, TI Category B) and irregular wave (H_s = 2.83 m, T_p = 7.85 s) conditions, normalised to those of the FB-only controller. The outer LIDAR beams were positioned at 50% rotor radial span for all LAC strategies.

Parameter	Metric	Normalised FB	Normalised FFCPC	Normalised FFCPC+FBIPC	Normalised FFCPC+FFIPC
	RMSE	1.00	0.28	0.27	0.28
Rotor speed	Range	1.00	0.36	0.32	0.39
	Maximum	1.00	0.83	0.83	0.83
Power	RMSE	1.00	0.28	0.27	0.28
Blade pitch	RMS Rate	1.00	2.18	7.64	2.22
Blade root	DEL	1.00	1.01	0.94	1.04
Tower base F-A	DEL	1.00	0.79	0.79	0.79
Tower base S-S	DEL	1.00	0.85	0.82	0.85
Platform pitch	Maximum	1.00	0.86	0.79	0.82

(Fig. 5.11b) provided little additional rejection of the wind disturbance (proportionally) over the FFCPC. This, paired with the fact that a proportional gain had to be introduced (Eq. 4.19) to further reduce the magnitude of the FFIPC commands to avoid heave excitation, limits the enhancement that could be brought to the WindCrete from the FFIPC in these conditions.

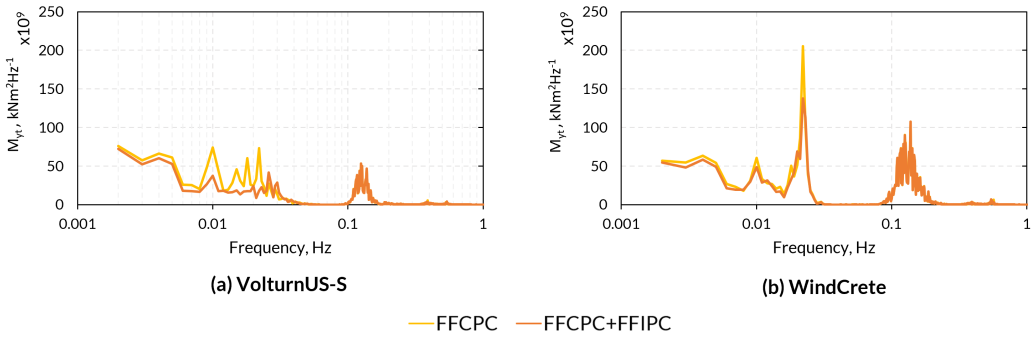


Figure 5.11: PSD plots of the tower base fore-aft bending moment of the VolturnUS-S and the WindCrete using FFCPC or FFCPC+FFIPC under turbulent wind ($\bar{V}$ = 17 ms^{-1}, NTM, TI Category B) and irregular wave (H_s = 2.83 m, T_p = 7.85 s) conditions.

Conversely to the VolturnUS-S, greater benefits to the WindCrete were found when using the traditional FBIPC strategy rather than the FFIPC strategy in terms of loadings and platform motions. It is proposed that this is because it used additional inputs (blade root bending moments) to formulate the individual commands as opposed to relying solely on the wind velocity used by the FFIPC, and so was able to mitigate the loads from other disturbances.

The FBIPC strategy was able to reduce the standard deviations and DELs of each of the blade root (by up to 6%), tower base fore-aft (by up to 21%) and tower base side-to-side bending moments (by up to 18%), relative to the FB-only controller. Additionally, further reductions to the standard deviations of the platform roll (by 8%), yaw (by 22%), heave (by 35%) and sway motions (by 3%) were recorded, relative to the FB-only controller. However, this came at the expense of a significantly greater RMS of the blade pitching rate (by 664%) compared to the FB-only controller, compounding the already significant increase (118%) that resulted from the addition of FFCPC.

The trends shown in the time series data can be further examined by the use of PSD plots. Figure 5.12 presents PSD plots of key turbine and substructure parameters obtained from one-hour length simulations at a mean wind speed of 17 ms^{-1}.

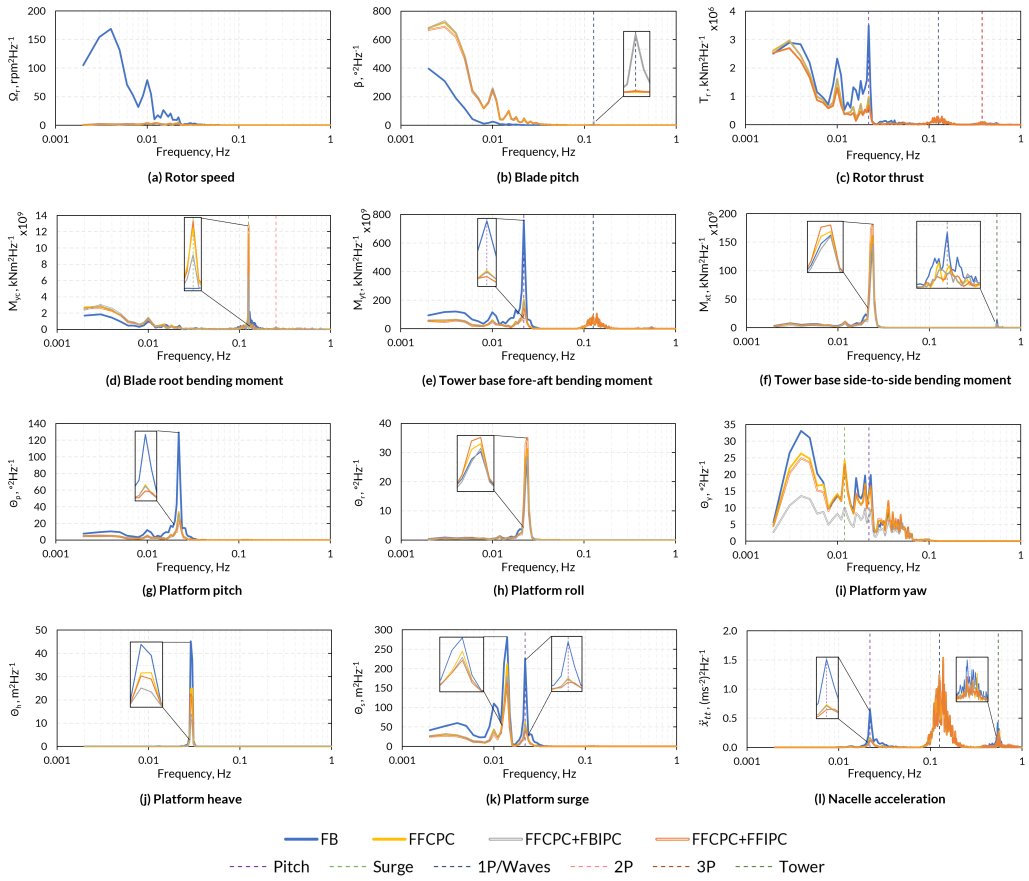


Figure 5.12: PSD plots of the turbine and substructure parameters of the WindCrete generated from one-hour simulations under turbulent wind ($\bar{V}$ = 17 ms^{-1}, NTM, TI Category B) and irregular wave (H_s = 2.83 m, T_p = 7.85 s) conditions. The outer LIDAR beams were positioned at 50% rotor radial span for all LAC strategies. Dashed vertical lines indicate pertinent frequency points.

Overall, the FF control implementation was able to significantly reduce the energy seen in the responses of the parameters at numerous frequencies, relative to the FB-only controller. Most notably, reductions occurred at frequencies pertinent to wind loading (> 0.03 Hz) and the pitch (0.022 Hz) natural frequency of the platform. As with the VolturnUS-S, this was particularly evident for the rotor speed (Fig. 5.12a), where all of the FF control strategies were able to reduce the energy of the responses at these frequencies, relative to the FB-only controller. However, for the WindCrete, this came at the expense of significantly greater energy in the blade pitch (Fig. 5.12b), relative to the FB-only controller.

The benefits reflected in the spectra of the rotor thrust (Fig. 5.12c), tower base fore-aft bending moment (Fig. 5.12e), platform pitch (Fig. 5.12g) and platform surge (Fig. 5.12k) were similar, though there were insignificant differences between the improvements offered by the different FF control strategies over the FB-only controller for these parameters.

Though the nacelle acceleration (Fig. 5.12l) caused by the 1P and wave frequencies at 0.126 Hz was the same for all control strategies, the FF control addition was able to reduce the energy (compared to the FB-only controller) at the pitch and tower natural frequencies (0.55 Hz).

Fig. 5.12 reinforces the advantage of the FBIPC addition observed from the time series results. Firstly, it did not result in as significant of a peak in the blade root bending moment (Fig. 5.12d) at the rotor 1P frequency. It also offered slight reductions to the energy of the tower base side-to-side (Fig. 5.12f) and platform roll (Fig. 5.12h) at this frequency, relative to the FB-only controller, which were instead increased when using FFCPC or FFCPC+FFIPC. The FFCPC+FBIPC also offered the most significant reductions over the FB-only controller to the energy of the yaw (Fig. 5.12i) and heave (Fig. 5.12j) motions of the platform.

5.5.2 Above-rated wind spectrum

Simulations were performed across the above-rated wind spectrum from 13-25 ms^{-1} in intervals of 2 ms^{-1}. This aimed to capture the performance benefits attainable from FFCPC, FFCPC+FBIPC and FFCPC+FFIPC across the full operating range of the pitch controller. The four-seed-average wind speed time series results for various turbine and substructure parameters are shown in Fig. 5.13.

As was indicated in Fig. 5.10, the standard deviation of the blade pitch of the WindCrete increased (compared to the FB-only controller) following the addition of FFCPC, FFCPC+FBIPC and FFCPC+FFIPC, by up to 63% at a mean wind speed of 17 ms^{-1}. The impact of this increased with increasing wind speed, rising to an up to 140% greater standard deviation of the blade pitch, relative to the FB-only controller, at a mean wind speed of 25 ms^{-1}. The tower base side-to-side bending moment and platform pitch of the WindCrete benefitted more from the FF control addition at higher above-rated wind speeds (by up to 58% reductions to

the standard deviations over the FB-only controller at a mean wind speed of 25 ms^{-1}). The inverse of this was true for the heave motion, whose standard deviation was reduced at low above-rated wind speeds (by up to 41%), relative to the FB-only controller, but increased to up to 5% greater than that of the FB-only controller at a mean wind speed of 25 ms^{-1}. The platform surge and, consequently, the fairlead tension, showed similar patterns in their responses, with standard deviation reductions over the FB-only controller increasing to 56% at 21 ms^{-1}, before plateauing as the wind speeds were increased further. In general, the improvements delivered by the FF controllers to the turbine and substructure parameters over the FB-only controller were least significant at lower above-rated mean wind speeds closer to the above-below-rated transition point (13 ms^{-1} and 15 ms^{-1}) due to the exclusions of the contributions of the FF controllers in measured wind speeds below 13 ms^{-1}.

As the wind speeds increased, some parameters experienced greater variation and higher magnitude oscillations when using the FB-only controller (the standard deviations increased by up to 31% between mean wind speeds of 13 ms^{-1} and 25 ms^{-1}, as shown in Table E.2) and so this led to more room for improvement for the FF controllers. This was particularly true for the rotor speed (Fig. 5.13b) and, by extension, the power (Fig. 5.13c), which benefitted most significantly, with reductions in their RMSEs, by up to 72%, compared to the FB-only controller. The rotor thrust (Fig. 5.13d) saw reductions in its standard deviation delivered by the FF controllers over the FB-only controller increase until 15% at 19 ms^{-1} before decreasing at higher wind speeds due to a diminishing standard deviation of the rotor thrust as the wind speed increased (by 31% between 13 ms^{-1} and 25 ms^{-1}, as shown in Table E.2) when using FB-only control, with hydrodynamic forces dominating. This was mirrored within the loadings and platform motions, as reductions to the standard deviations of the tower base fore-aft bending moment (Fig. 5.13f), platform pitch (Fig. 5.13i) and platform surge (Fig. 5.13k) (relative to the FB-only controller) also diminished at higher wind speeds due to the decreased reduction to the standard deviation of the rotor thrust.

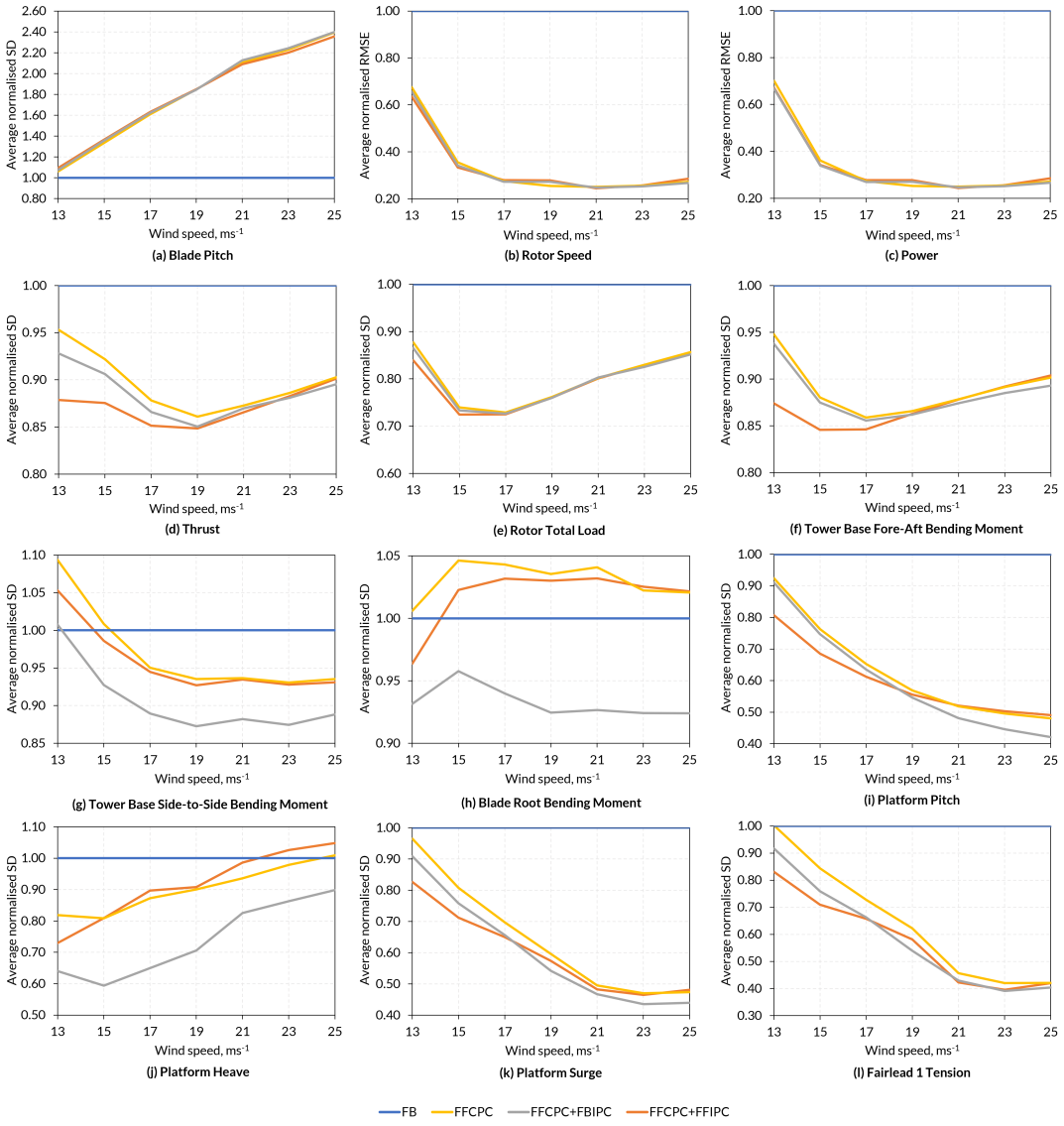


Figure 5.13: Normalised SD or RMSE values delivered by the FF controllers compared to the FB-only controller of the WindCrete across above-rated wind speeds. The LIDAR beams were positioned at 50% rotor radial span for all LAC strategies. The results shown are the four-seed-average normalised (to the FB-only controller) values from the one-hour simulations at each mean wind speed integer for each control configuration.

Though the benefit of the FFIPC addition for the WindCrete became insignificant at wind speeds above a mean of 17 ms^{-1}, at lower above-rated wind speeds, it was able to deliver further standard deviation reductions over the FB-only controller to the FFCPC, namely in the thrust (by up to 7%), the tower base bending moments (by up to 4% and 8% in the side-to-side and fore-aft directions, respectively) and the platform pitch and surge (by up to 11% and

13%, respectively). This was due to the fact that the aerodynamic forces dominate over the hydrodynamic forces at these wind speeds, where the FFIPC was better able to improve the disturbance rejection of the wind delivered by FFCPC. This is highlighted in Fig. 5.14, which presents PSD plots of the tower base fore-aft bending moment at different mean wind speeds.

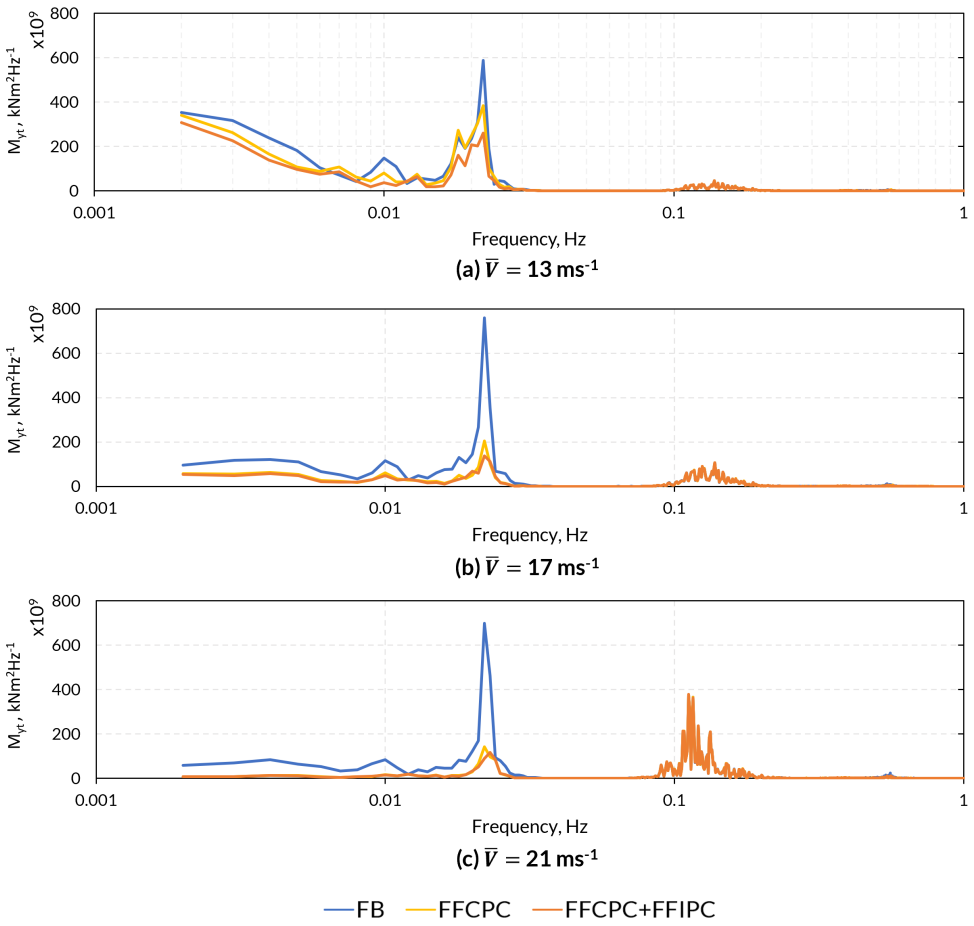


Figure 5.14: PSD plots of the tower base fore-aft bending moment of the WindCrete using FB-only control, FFCPC and FFCPC+FFIPC under turbulent wind fields with different mean wind speeds.

From Fig. 5.14, it is clear that the addition of FFIPC was able to offer greater disturbance rejection of the wind compared to FFCPC at lower above-rated wind speeds (13 ms^{-1}, Fig. 5.14a). This benefit diminished as the wind speed and hydrodynamic forces increased, signified by the similar PSD plots of the FFCPC and FFCPC+FFIPC at wind speeds of 17 ms^{-1} (Fig. 5.14b) and 21 ms^{-1} (Fig. 5.14c).

Overall, the FFCPC with the addition of FBIPC gave the greatest benefits for the WindCrete, with the only drawback being the significantly increased mean rate of blade pitching compared to the FB-only controller. FBIPC provided benefits to the FFCPC over the FB-only controller, including further reductions to the variation of the blade root (by up to 12%) and tower base side-to-side (by up to 7%) bending moments and the platform roll (by up to 10%) and heave motions (by up to 22%), and so is recommended over the FFIPC addition for the WindCrete if the additional wear and demand on the pitch actuation systems can be accounted for within the turbine's design.

5.6 Floating offshore wind turbine feedforward control results comparison

This Section compares the results of the best-performing FF controllers of the VolturnUS-S (FFCPC+FFIPC) and the WindCrete (FFCPC+FBIPC) to their respective FB-only controllers, highlighting the key differences between the impacts on the two FOWTs.

5.6.1 Single wind field

Figure 5.15 provides a spider-web plot of the average normalised standard deviation values for various parameters delivered by the FFCPC+FFIPC (VolturnUS-S) or FFCPC+FBIPC (WindCrete) strategy for the two substructures relative to their respective FB-only controller across four random seed one-hour length simulations under the aforementioned wind and wave conditions.

When comparing the results of the FFCPC+FFIPC for the VolturnUS-S and FFCPC+FBIPC for the WindCrete to their respective FB-only controllers, the WindCrete was able to benefit from greater rotor speed and power RMSE reductions than the VolturnUS-S, relative to its FB-only controller. However, this came at the expense of significant increases to the blade pitch variation and rate of change of the WindCrete, due to the slow nature its FB-only controller. The changes to the pitch behaviour greatly benefited the performance of the turbine, the loadings and the motions of the platform, while not introducing negative damping.



Figure 5.15: Spider-web plots showing the four-seed-average standard deviations of various parameters of the VolturnUS-S and WindCrete using the selected FF control strategies under turbulent wind ($\bar{V}$ = 17 ms^{-1}, NTM, TI Category B) and irregular wave (H_s = 2.83 m, T_p = 7.85 s) conditions, normalised to their respective FB-only controller. The outer LIDAR beams were positioned at 50% rotor radial span for all LAC strategies.

The standard deviations of the rotor total load and fore-aft bending moment of the tower were significantly reduced for both FOWTs, by up to 27%, relative to their respective FB-only controllers, attributed to the reduction in the variation of the thrust force acting on the turbines - by 22% for the VolturnUS-S and 15% for the WindCrete. The WindCrete saw a reduction in the standard deviation of its tower base side-to-side bending moment (by up to 11%) when using LAC, relative to when using its FB-only controller. Meanwhile, the standard deviation of the tower base side-to-side bending moment was found to slightly increase (by 2%) in the VolturnUS-S when using LAC, relative to when using its FB-only controller. Similarly, in terms of platform motions, the greatest beneficiary for both FOWTs was the pitch DOF, where 36% and 31% reductions in the standard deviations, compared to their respective FB-only controllers, were recorded for the VolturnUS-S and WindCrete, respectively. Among the other DOFs, the addition of LAC to the WindCrete saw greater reductions to the standard deviations of its respective FB-only surge (by 10%) and heave (by 37%) motions than the VolturnUS-S.

Ultimately, the addition of the FFIPC was more beneficial to the VolturnUS-S than the WindCrete as the controller behaviour was more responsive and so could be further enhanced by the FFIPC, while the WindCrete benefitted more from FBIPC due to the additional inputs provided to the controller.

5.6.2 Above-rated wind spectrum

As can be seen from Fig. 5.16, there were distinct differences and similarities in the benefits brought and the trends seen in the responses from the addition of the FFCPC and FFCPC+FFIPC (in the VolturnUS-S) or FFCPC+FBIPC (in the WindCrete) to the two FOWTs across the above-rated wind spectrum.

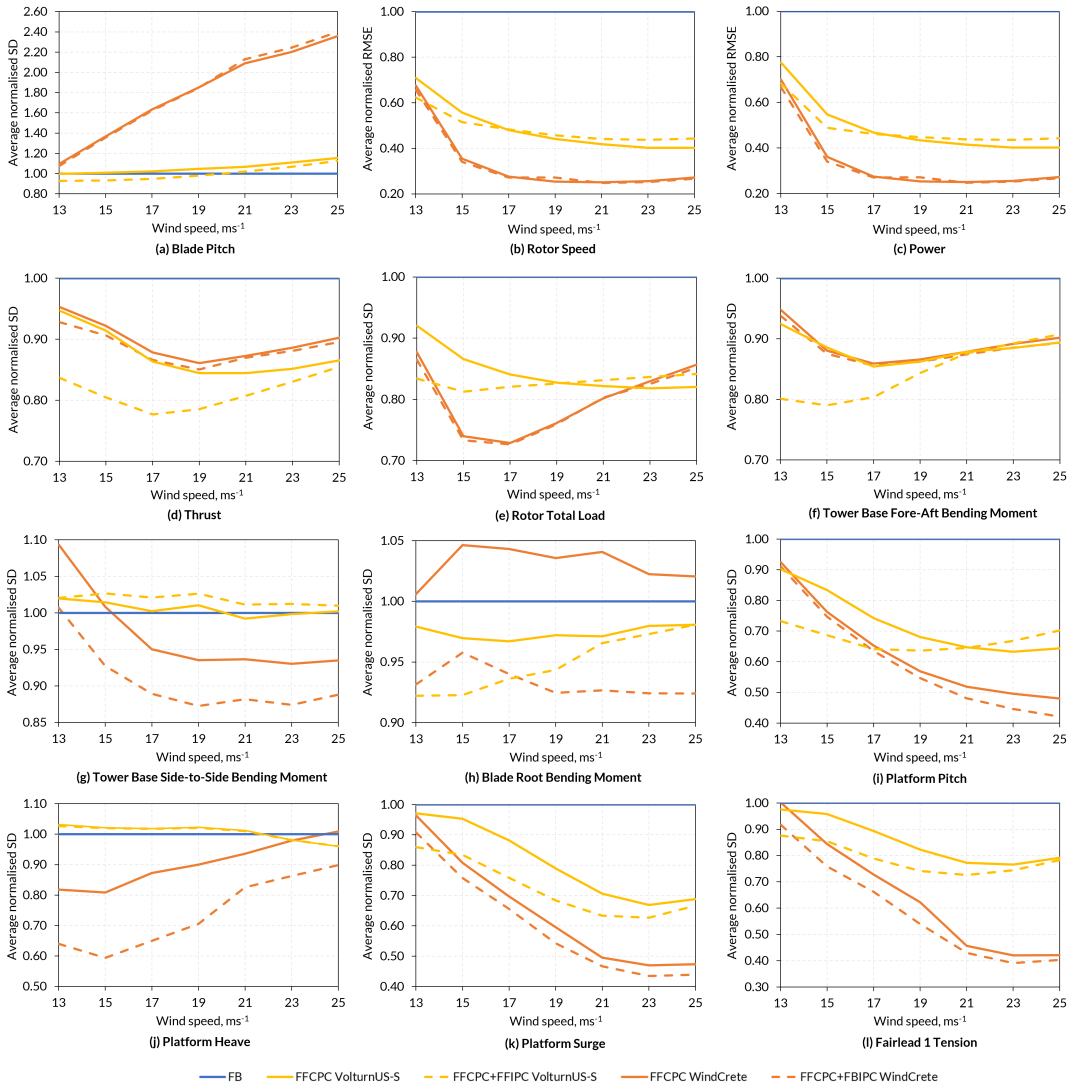


Figure 5.16: Normalised SD or RMSE values delivered by the FF controllers of the VolturnUS-S and WindCrete compared to their respective FB-only controllers across above-rated wind speeds. The LIDAR beams were positioned at 50% rotor radial span for all LAC strategies. The results shown are the four-seed-average normalised (to their respective FB-only controller) values from the one-hour simulations at each mean wind speed integer for each control configuration.

As was also shown in Fig. 5.15, the standard deviation of the blade pitch of the WindCrete (relative to its FB-only controller) increased from the addition of FFCPC and FBIPC and became greater with increasing wind speeds. Both turbines saw the reductions to their rotor speed and power RMSEs over their respective FB-only controllers remain consistent at mean wind speeds of 17 ms^{-1} and greater, with minimal difference between the FFCPC and FFCPC+FFIPC or FFCPC+FBIPC responses. Both FOWTs also followed similar trends within the rotor thrust and tower base fore-aft bending moment, with reductions to the standard deviations over their respective FB-only controllers increasing until approximately 17 ms^{-1} before diminishing at higher wind speeds due to the increased dominance from hydrodynamic loading. The standard deviation of the blade root bending moment of the VolturnUS-S was reduced for both FF control strategies (by up to 8%, relative to its FB-only controller), though this increased for the WindCrete FFCPC (by up to 5%, relative to its FB-only controller) due to its increased pitch activity. Differences in the trends were also observed in the platform heave, while the trends seen in the platform pitch and surge and the fairlead tension were similar for the two FOWTs.

Overall the VolturnUS-S was subject to greater reductions to the standard deviations (relative to its FB-only controller) brought by the FFIPC addition than the WindCrete (relative to its FB-only controller), despite similarities in where the greatest reductions were achieved. This is attributed to the greater responsiveness of the VolturnUS-S's FB-only controller, making it more suited to additional tuning from the FFIPC. The benefits to the VolturnUS-S delivered by the addition of FFIPC could be matched in the WindCrete through the addition of FBIPC instead. The FF control implementation to the WindCrete was also more problematic due to its slow rate of FB-only control actuation, as limits had to be applied to the FFCPC and FFIPC commands in close-to-rated wind speeds (documented in Sect. 5.2), in addition to a gain applied (in Eq. 4.19) on the individual pitch commands to reduce their magnitude in order to avoid platform heave excitation.

5.7 Estimation of feedforward individual pitch controller beam assignment errors

Within Sect. 4.4, the assumption of constant rotor rotational speed when predicting the azimuth angles of the blades (using Eq. 4.14) following the time delay incurred by the low-pass filter applied to the FFIPC commands (T_{IPC}) was declared. However, in the simulations, the rotor speed was subject to change during this interval and so the actual azimuth angle of the blades following the time delay is likely to have been different to that predicted. In some instances, these differences will have resulted in incorrect beam assignment to the blades and,

by extension, incorrect FF individual pitch commands delivered to the blades. The degree of error between the predicted and actual azimuth angles, and the subsequent percentage misassignment that may have resulted, is estimated in this Section. The following results were consistent across both FOWTs.

For each seed of each mean wind speed simulation with the FFCPC+FFIPC strategy enabled, the time series data was analysed. The OpenFAST azimuth and rotor speed outputs at each time step were applied within Eq. 4.14 to determine the azimuth angle predictions following the time delay period ($T + T_{IPC}$). This was compared to the actual azimuth angle output that was recorded at that time. The absolute value of the error between the predicted and actual blade azimuths at each time step was then determined. The four-seed-average mean and maximum error recorded across above-rated wind speeds are shown in Table 5.6.

From Table 5.6, it is clear that, as the wind speed increased, the mean and maximum error between the predicted and actual blade azimuth angles also increased, by 86% and 93% between mean wind speeds of 13 ms^{-1} and 25 ms^{-1}, respectively. This is because the standard deviation of the rotor speed increased with wind speed (by 73% between mean wind speeds of 13 ms^{-1} and 25 ms^{-1}), meaning that there were greater differences between the rotor speed used when predicting the azimuth angles (Ω_r) of the blades and the mean rotor speed over the delay period ($\bar{\Omega}_{r,T_{IPC}}$). These differences increased by 86% between mean wind speeds of 13 ms^{-1} and 25 ms^{-1}, as indicated in Table 5.6, leading to the greater errors in the blade azimuth predictions.

Table 5.6: Four-seed-average mean and maximum blade azimuth prediction errors and mean rotor speed errors when using the FFIPC across above-rated wind speeds.

Mean wind speed, ms^{-1}	**13**	**15**	**17**	**19**	**21**	**23**	**25**
Mean error, $^\circ$	3.71	3.90	4.43	5.11	5.78	6.40	6.90
Maximum error, $^\circ$	17.6	19.7	22.1	24.5	27.7	30.6	33.9
Mean $\lvert\Omega_r - \bar{\Omega}_{r,T_{IPC}}\rvert$, rpm	0.089	0.100	0.107	0.123	0.139	0.154	0.166

Although the prediction of the azimuth angles of the blades was imperfect, the incorrect predictions were only consequential if they led to incorrect assignment to the beams and, therefore, incorrect assignment of the individual pitch commands. This will have occurred only when the predicted and actual blade azimuth angles were situated in different azimuth quadrants, shown within Fig. 5.17. As the beam-blade allocation followed a nearest-neighbour approach, should the predicted and actual azimuth angles have been situated in different quadrants, then the blade will have been predicted to be closest to a different beam than the one it was closest to in reality, resulting in an incorrect beam assignment and subsequent incorrect individual command at that time step.

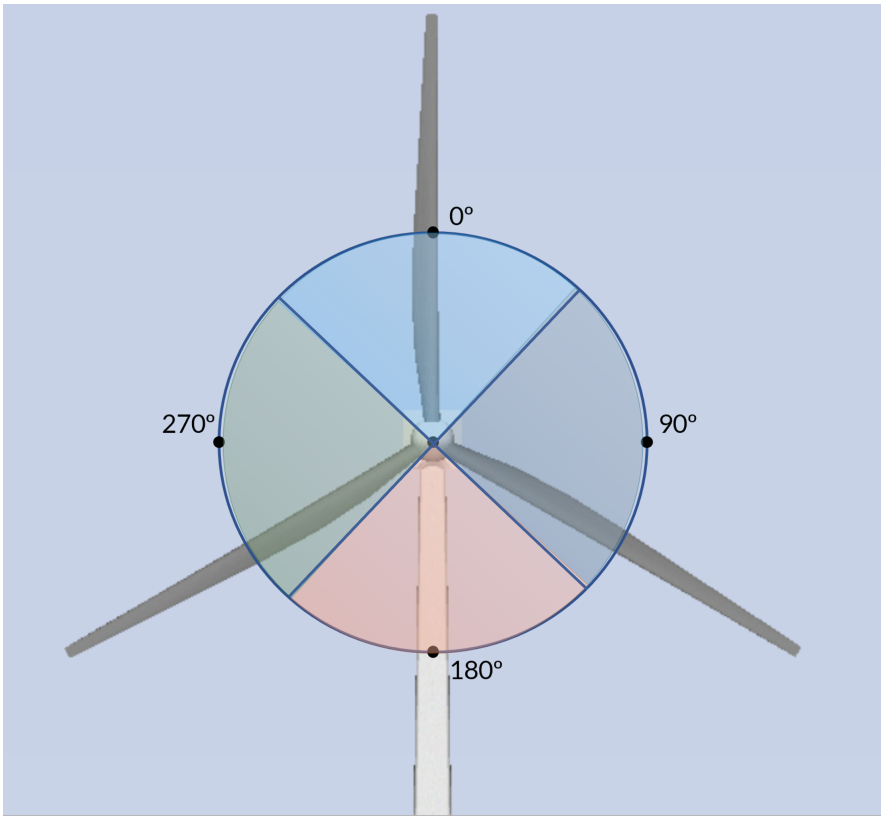


Figure 5.17: Rotor azimuth quadrants indicating the areas closest to each of the outer LIDAR beams. Predicted and actual blade azimuth angles situated in different quadrants will have resulted in the incorrect assignment of a blade a LIDAR beam for delivery of the FFIPC command.

From the four-seed time series results at each mean wind speed, the instances where the predicted and actual blade azimuth angles lay in different quadrants (of Fig. 5.17) were counted and used to provide an estimate of the percentage beam/command misassignment that may have occurred. These were only estimates because the default 90-degree azimuth increments of the outer beam positions were assumed (i.e. the influence of the motions on the beam azimuth positions was not considered here). Nonetheless, as the changes to the beam positions resulting from the nacelle motion were not expected to significantly alter their relative azimuth positions, the results are expected to provide a strong indication of the degree of beam assignment error that may have occurred during the simulations. The four-seed-average estimates of the percentage beam misassignment for each mean wind speed are shown in Fig. 5.18.

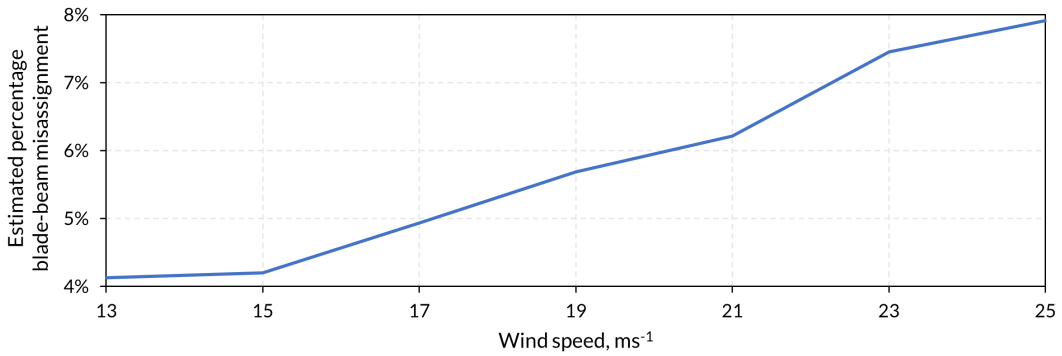


Figure 5.18: Estimated percentage misassignment of the LIDAR beams to the rotor blades when issuing the FFIPC commands across above rated wind speeds.

The results indicate that the beam-blade assignment error also increases with wind speed, by approximately 4% between mean wind speeds of 13 ms^{-1} and 25 ms^{-1}. This was expected given the results shown in Table 5.6, because larger errors in the blade azimuth estimates and larger differences between rotor speed used when predicting the azimuth angles of the blades and the mean rotor speed over the delay period occurred at higher wind speeds. This increased the likelihood of the actual and estimated beam azimuths being situated in different azimuth quadrants.

Ultimately, this led to the greatest individual pitch command assignment errors (up to 7.9%) occurring at higher wind speeds, which provides evidence as to why the improvements to the SDs and RMSEs of the turbine parameters of the VolturnUS-S delivered by the FFCPC+FFIPC (over the FB-only controller) deteriorated at higher above-rated wind speeds in Fig. 5.7. Nonetheless, the percentage beam assignment error was sufficiently low for the benefits of the FFIPC implementation to be realised, particularly at lower above-rated wind speeds, though it would be beneficial to investigate the degree to which results can be improved by reducing or eliminating the error in future work.

The data and results of the VolturnUS-S simulations were carried forward to the O&M case study in Chapter 6, where loading reductions were translated to reductions in component failure rates, which were then fed to the StrathO&M model. The VolturnUS-S results were used because it is the substructure topology most suited to the floating wind farm sites of the ScotWind leasing round (due to the water depth and the port infrastructure), which was examined within the case study in Chapter 6. This was also why the VolturnUS-S was modelled within the FAST.Farm study in Chapter 8. The VolturnUS-S was also modelled within WEIS/WISDEM within Chapter 7 to examine the loadings of modified FOWT turbine tower designs. The VolturnUS-S was used because it was available as a parameterised input example within WEIS.

Chapter 6

O&M case study

Building upon the OpenFAST simulation results of the VolturnUS-S, this case study focuses on translating the reductions in the variations of the structural loadings to component failure rate reductions and, subsequently, investigating how these can lead to reduced OpEx of FOW. The reductions in OpEx are expected to stem from the decreased lost revenue resulting from reduced downtime of turbines, as well as lower costs associated with the reduced frequency of maintenance operations, where vessel hire, personnel and material costs are incurred. Furthermore, the decreased failure rates achieved from the addition of LAC (which henceforth refers to the FFCPC+FFIPC strategy, specifically) may also alleviate the workability access constraints for FOWTs (introduced in Sect. 1.3.7) due to the reduced frequency of transfers to site. Therefore, this case study will assess how the cost benefits of the failure rate reductions provided by LAC vary when applied to the different FOWT platform designs studied by Scheu et al. (2018), which differ in the strictness of the accessibility constraints that they impose. Three separate wind farm locations will be examined to assess the impact of distances from shore, wind speed distributions and mean wave heights on the findings.

6.1 Statement of collaboration

The work presented in this Chapter was performed in collaboration with Dr. Jade McMorland a (now graduated, see McMorland, 2024) PhD student at the University of Strathclyde. The time series outputs of the OpenFAST simulations of the VolturnUS-S were converted to lifetime failure rates, damage and DELs of turbine components. These results were relayed to Dr. McMorland, who utilised them when setting the failure rates used within the O&M model. The wind farm sites analysed, the inputs to the simulations, the failure-rate-to-turbine-sub-system matching and the post-processing steps were decided upon collaboratively between the author and Dr. McMorland. The O&M modelling results reported within Sect. 6.4 were provided by Dr. McMorland (see McMorland, 2024), following completion of the simulations. The write-up of the O&M simulation results was performed collaboratively between the author and Dr. McMorland.

Dr. McMorland also contributed to parts within this thesis pertaining to key aspects of this case study. These included the provision of information of the costs associated with the maintenance of wind turbines (within Sect. 1.3.7), the details of the wind farm sites assessed within this case study (Tables 6.1 and 6.2), the set-up and operation of the O&M model used to perform the simulations (within Appendix A.3 and Sect. 6.3) as well as the information and explanation of the WI limits applied to the FOWTs within the model and their significance to the outcomes of this case study (within Sects. 1.3.7, 2.4 and 6.4.3).

6.2 ScotWind case study

In 2022, Crown Estate Scotland launched the ScotWind leasing round, aiming to secure 8-10 GW of additional offshore wind capacity, raising the Scottish target to 17-19 GW, mainly by 2030-32. The 2022 ScotWind leasing results (see Crown Estate Scotland, 2022) exceeded expectations by awarding eleven FOW projects with a combined capacity of approximately 18 GW, showcasing the appetite for rapid growth of the FOW industry in addition to Scotland's leading position within the sector. The number of projects is expected to lead to significant competition when securing a strike price through the UK government's contracts for difference (CfD) financing scheme (see Department for Energy Security and Net Zero, 2023a; Low Carbon Contracts Company, 2024). In addition, margins are tightened due to inflation and the rising costs of commodities. Therefore, technologies with the potential to offer OpEx reductions are of vital importance. Naturally, this formed a motivation for this case study, which sought to determine the potential for LAC to deliver decreased OpEx via FOWT component failure rate reductions.

This work investigated three of the recent ScotWind allocation sites as the basis of the case study: E1, NE1 and NE3. These sites were selected due to their spread of location, expected wind farm capacity and distance to shore. This case study aimed to assess how the cost-saving benefits brought by LAC differ for the three sites due to these variables. A map (from Clive et al., 2022) showing the location of these zones is presented in Fig. 6.1, with details of their distances to shore and their mean values of H_s ($\bar{H}_s$) provided in Table 6.1.

Table 6.1: ScotWind FOW regions examined in this case study, with detail of their O&M base, distance to shore and mean H_s.

ScotWind site	O&M base	Distance to shore, km	$\bar{H}_s$, m
E1	Aberdeen	120	1.7
NE1	Lerwick	55	1.5
NE3	Wick	55	1.6

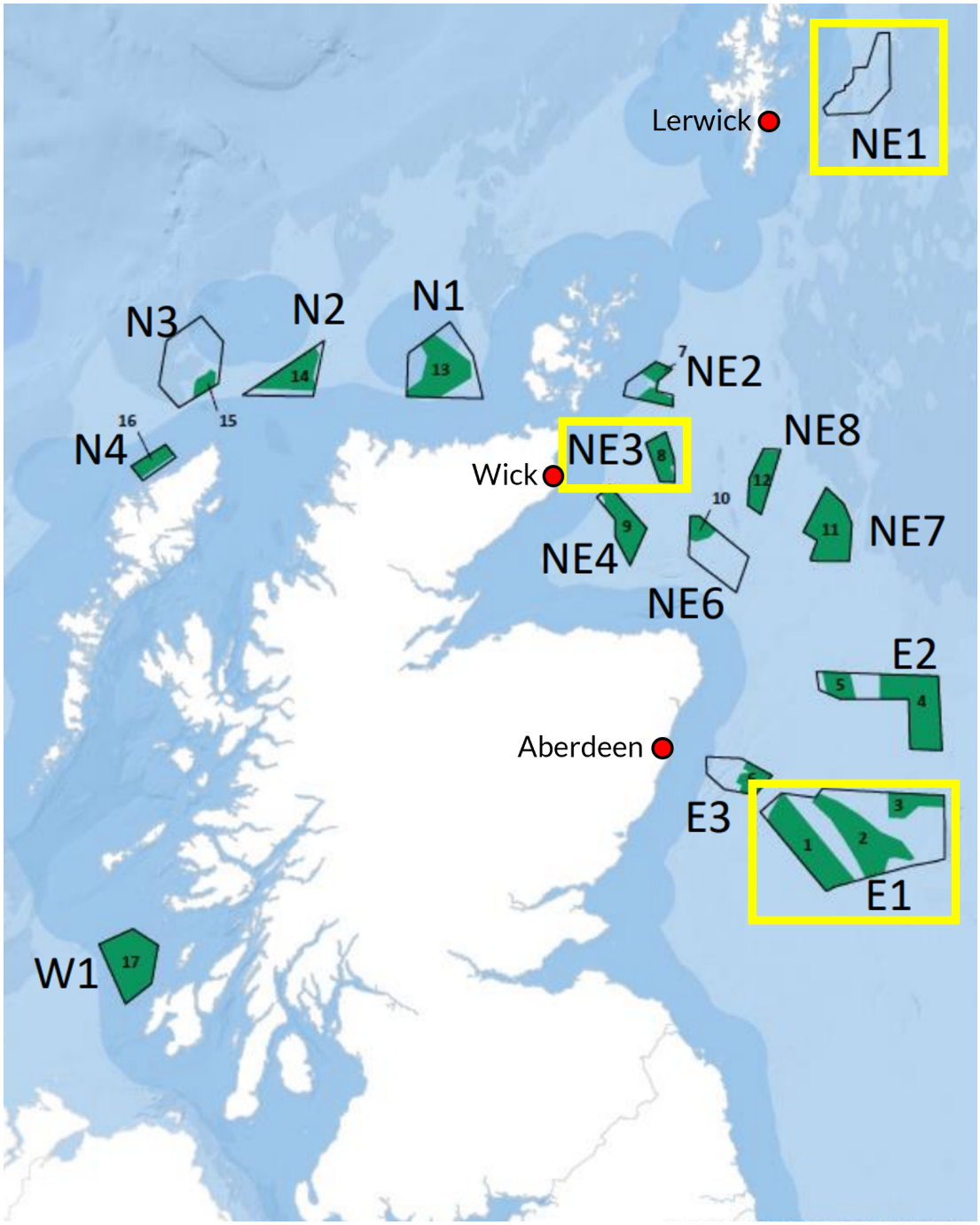


Figure 6.1: Map (adapted from Clive et al., 2022) showing the locations of the ScotWind leasing round sites studied within this case study, highlighted by the yellow boxes. Red dots are marked to show the O&M base of each site.

As was explained in Sects. 1.3.7 and 2.4, greater distances resulting in a longer travel time required to get to and from the site lead to greater costs for vessel hire and personnel. Furthermore, the weather window restrictions for these far offshore locations amplify the impacts of turbine downtime, as the waiting on weather periods will be longer and more

frequent. Therefore, E1 was expected to benefit most from the implementation of the LAC strategy because the reduced failure rates would not only decrease the downtime of the turbines, but also the frequency of maintenance visits and, therefore, the OpEx, compared to using the FB-only control strategy. This case study also applied accessibility constraints to four different prospective floating platform designs, based on the WI results provided by Scheu et al. (2018), to investigate the impact that these variable limits would have on the lifetime costs of maintaining FOW farms with different floating platform designs when the reduced failure rates from LAC were applied. The workability limits were determined from wave conditions (H_s and T_p) and so, within the O&M simulations, the differing mean values of H_s at each ScotWind site were anticipated to have impacts upon the OpEx for the different platform designs. These results were then compared to the BFW equivalent wind site using FB-only control to investigate how the OpEx compared to the traditionally lower-cost technology.

6.3 O&M methodology

The OpEx simulations within this case study were performed by Dr. McMorland using the Strath-OW O&M model created by Dinwoodie et al. (2012) and outlined within Appendix A.3. Within the model, access to site is limited by V and H_s, where the thresholds are vessel-dependent. The model has been adapted to FOW by accounting for workability limits imposed based on the floating platform design. The combination of T_p and H_s form the limiting criteria, dependent on the specific platform design. An example of limitations imposed upon Design A (from Scheu et al., 2018) is shown in Fig. 6.2.

Significant Wave Height, Hs [m] \ Peak Wave Period, Tp [s]	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0.5														
1.0			X											
1.5			X											
2.0			X	X	X									
2.5				X	X	X								
3.0					X	X	X	X						
3.5					X	X	X	X	X	X	X	X	X	X
4.0					X	X	X	X	X	X	X	X	X	X

X – WI limit
– vessel limit

Figure 6.2: Limitations imposed on the accessibility of Design A for specific H_s-T_p combinations, adapted from Scheu et al. (2018). The Xs indicate the T_p and H_s combinations that cause the FOWT to be inaccessible. Reprinted with permission from ISOPE.

The Xs marked within Fig. 6.2 indicate the T_p and H_s combinations that cause the FOWT to be inaccessible. This indicates that, when workability constraints are applied, access to the FOWTs can be prevented even in conditions where the vessel would otherwise be able to safely make a trip (signified by the non-green boxes in Fig. 6.2). This shows that the motions of FOWTs (while in their parked condition, where LAC is inactive) can become the limiting factor for accessing them in certain conditions.

6.3.1 Maintenance strategy

Within offshore wind, maintenance actions can be categorised as minor, major or replacement operations, as per Carroll et al. (2016) and ReliaWind (2011). While there are significant challenges surrounding the major component replacement of FOWTs, discussed in Sect. 1.3.7, this work focused on the impact of reduced failure rates on minor/major repairs, which make up 90% of total lifetime failure rates, according to Carroll et al. (2016). Therefore, the use of a heavy lift vessel (HLV) within the maintenance strategy was not included.

It was assumed that the cost of repair of components was identical for both BFW and FOW turbines to allow for direct comparison. Each turbine had a 15 MW rating, based on the IEA 15 MW reference turbine from Gaertner et al. (2020). All baseline failure rates, technicians required, time-to-repair and cost of components for different turbine sub-systems were taken from Carroll et al. (2016).

To allow for direct comparison between the ScotWind locations, each site had a capacity of 1 GW and followed the same maintenance strategy. Due to the scale of the sites, and their distances to shore, this case study utilised a service operation vessel (SOV)-based maintenance strategy. SOVs are used to allow technicians to remain at sea in close proximity to the turbines for extended periods of time (typically up to four weeks), as opposed to crew transfer vessels (CTVs), which transport technicians to and from a site within a given day. Each site had a dedicated SOV, which was supported by daughter craft. A set of six daughter craft was chosen to ensure that maintenance activities were not hindered due to resource limitations. The SOV and daughter craft had a H_s limit of 3.5 m and 2 m, respectively. Since the goal of this case study was to assess potential OpEx reductions via the implementation of LAC, the optimisation of the vessel fleet was not the focus of this case study. Planned maintenance was set to sixty hours per turbine per year and maintenance activities were not limited by daylight working hours.

6.3.2 Failure rate determination

The results of the FB-only controller and the best-performing FF controller of the VolturnUS-S (FFCPC+FFIPC) were post-processed using MLife. MLife is a Matlab-based tool created by NREL to post-process results from wind turbine tests and aero-elastic, dynamic simulations, including from OpenFAST. MLife computes statistical information and fatigue estimates for one or more data files and follows the techniques outlined in the IEC 61400-3-1:2019 (2019) standards. Essential details and equations pertaining to MLife's operation are provided within this Section. A complete theory manual by Hayman (2012) outlining all details of MLife's operation is also available for reference.

MLife works by accumulating fatigue damage due to fluctuating loads over the design life of a wind turbine, which are broken down into individual hysteresis cycles by rainflow counting (see Hayman, 2012). The damage (D) is assumed to accumulate linearly with each cycle according to Miner's Rule (see Miner, 1945). The total damage from all cycles is given by

$$D = \sum_i \frac{n_i}{N_i(L_i^{RF})} \tag{6.1}$$

where N_i is the number of cycles to failure, n_i is the cycle count and L_i^{RF} is the cycle's load range about a fixed load-mean value. The relationship between the load range and cycles to failure (the S-N curve) is modelled by

$$N_i = \left(\frac{L^{ult} - |L^{MF}|}{\frac{1}{2}(L_i^{RF})} \right)^m \tag{6.2}$$

where L^{ult} is the ultimate design load of a component, L^{MF} is the fixed load-mean and m is the Wöhler exponent, which is specific to the component under consideration. In this case study, an m of 4 was used for steel components and an m of 10 was used for composite components, as these are the values that are typically assumed (see Slot et al., 2019; Stanley et al., 2020). L^{ult} was determined for each component by taking the maximum value of its associated OpenFAST output from all of the time series simulation data and multiplying it by a safety factor.

As the simulations performed only cover a short period of the design lifetime, the time series damage-cycle counts were required to be extrapolated over the design lifetime. Equations 6.1 and 6.2 were rewritten to account for the accumulation of damage across the turbine's lifetime (see Hayman, 2012) as follows

$$D_j^{Life} = \sum_i \frac{n_{ji}^{Life}}{N_{ji}} \tag{6.3}$$

$$D^{Life} = \sum_j D_j^{Life} \tag{6.4}$$

$$N_{ji} = \left(\frac{L^{ult} - |L^{MF}|}{\frac{1}{2} L_{ji}^{RF}} \right)^m \tag{6.5}$$

where D_j^{Life} is the extrapolated damage over the design lifetime due to the j^{th} time series, n_{ji}^{Life} is the extrapolated cycle counts and N_{ji} is the number of cycles to failure.

To perform the extrapolations of the damage-cycle counts across the lifetime of the wind turbine at each of the sites studied, MLife required the wind speed distributions of the sites. This enabled MLife to model the wind with a Weibull distribution (see Weibull, 1951). The Weibull parameters and mean wind speeds that were input to MLife for each of the ScotWind sites analysed are given in Table 6.2.

Table 6.2: Mean wind speeds and Weibull parameters of the ScotWind leasing locations assessed in this case study (from LAUTEC ESOX, 2024).

ScotWind site	$\bar{V}$, ms^{-1}	C, ms^{-1}	K
E1	8.293	9.358	2.360
NE1	8.567	9.667	2.394
NE3	10.10	11.39	2.255

Subsequently, the wind speeds were placed into bins, which were then assigned a probability of occurrence based on the Weibull distribution of each site. The extrapolation was also affected by the turbine's availability factor, ψ, where a ψ of 1 indicated that the turbine was always online and producing when the wind speed was between the cut-in (V_{in}) and cut-out (V_{out}) wind speeds. A ψ of 0.95 was assumed for all sites within this case study.

When considering power production DLCs with mean wind speeds lying in between V_{in} and V_{out} of the turbine (as in this case study), the damage-cycle counts are scaled such that

$$\varepsilon_j^{Life} = \frac{T^{Life} \psi p_l^v}{T_l}, (V_{in} < \bar{V}_j \leq V_{out}) \tag{6.6}$$

where T^{Life} is the design lifetime period, T_l is the total elapsed time for all time series that have a mean wind speed falling in bin l, p is the probability factor of a mean speed falling within a bin and ε_j^{Life} is the extrapolation factor for time series j. The extrapolated cycle counts over the design lifetime were then obtained using

$$n_{ji}^{Life} = \varepsilon_j^{Life} n_{ji} \tag{6.7}$$

where n_j^{Life} is the i^{th} cycle count for time series j. The lifetime damage was then accumulated for all cycles and time series such that

$$D^{Life} = \sum_j \sum_i \frac{n_{ji}^{Life}}{N_{ji}} \tag{6.8}$$

As failure occurs when D^{Life} equals one, the time until failure, T^{Fail}, is the ratio between the design lifetime and the accumulated damage.

$$T^{Fail} = \frac{T^{Life}}{D^{Life}} \tag{6.9}$$

The failure rate, λ, is the inverse of the time until failure.

$$\lambda = \frac{1}{T^{Fail}} \tag{6.10}$$

Within the analysis performed in this case study, the time series outputs for numerous loadings associated with different turbine components were input to MLife and the time-to-failure associated with each loading on a component was determined. The loading with the lowest time-to-failure was assumed to be the time-to-failure of the component.

6.3.3 Assessed parameters

OpenFAST allows for the output of time series data of numerous loadings applied to the turbine's various sub-systems. These include the rotor blades, tower, shaft, bedplate and pitch bearings. The data acquired can be processed through MLife, which can return the damage, time-to-failure and DEL occurring as a result of a particular load applied on the subsystem. As the OpenFAST load outputs are defined at various cross-sections of the turbine, this process works well for structures composed of simple cylindrical shapes and isotropic materials such as the tower and shaft, as S-N curves are able to deliver a good indication of the time-to-failure at the cross sections of the components. Conversely, for non-cylindrical components, MLife is unable to accurately predict their times-to-failure as the S-N curves cannot be applied to these components. A detailed finite-element analysis (FEA), with the load outputs from OpenFAST as an input, is required to determine the resulting state of stress for these components. However, for the purposes of this case study, it was assumed that the fatigue loads can be directly translated into reductions in failure rate according to the S-N curves of composite and steel components that were used. Nevertheless, it is important to state that, while S-N curves are vital tools for the assessment of fatigue life of materials and have been used to do so for many years, there is evidence to suggest that S-N curves may not always accurately predict fatigue life under complex loading conditions and due to variations in

material properties (see Barbosa et al., 2019), and that the fitting capabilities vary depending on the S-N curve model that is used (see Barbosa et al., 2019). Studies suggest that the use of probabilistic S-N fields (see Barbosa et al., 2019) or modified state-space models (see Barbosa et al., 2019) may be better suited to estimating the fatigue life of composite and metal components of FOWTs.

The total aerodynamic load-induced bending moments on the rotor-nacelle assembly (RNA) were assumed to represent the forces acting upon the rotor blades, hub and pitch system, as defined by Dao et al. (2019). The approximately 15% reductions to the standard deviation of these loads delivered by LAC over the FB-only controller (shown in Fig. 5.7) were, therefore, used to estimate the reductions to the times-to-failure of the components within the RNA, which were used within the O&M model.

Table 6.3 gives the OpenFAST outputs that represent the loadings applied to the relevant sub-system. Time series data of these outputs, obtained from the OpenFAST simulations of the VolturnUS-S, were input to MLife for the turbine utilising the FB or FFCPC+FFIPC controllers. For the FFCPC+FFIPC, the same data as was used for the FB when mean wind speeds below 11 ms^{-1} were simulated as this was below the rated wind speed (10.69 ms^{-1}) and where the FF controller did not operate.

Table 6.3: Turbine sub-systems considered in this case study and the OpenFAST outputs associated with them.

Sub-system	OpenFAST output	Description	Units
RNA[a]	RtFldMxh	Total rotor aerodynamic load (moment in x direction).	Nm
Pitch bearings	RootMxc#	Blade in-plane moment moment at the blade root.[b]	kNm
	RootMyc#	Blade out-of-plane moment at the blade root.[b]	kNm
Blade roots	RootMxb#	Blade edgewise moment at the blade root.[b]	kNm
	RootMyb#	Blade flapwise moment at the blade root.[b]	kNm
Tower base	TwrBsMxt	Tower base side-to-side bending moment.	kNm
	TwrBsMyt	Tower base fore-aft bending moment.	kNm
Moorings	FAIRTEN1	Tension in mooring line 1 at the fairlead.	kN
	ANCHTEN1	Tension in mooring line 1 at the anchor.	kN

[a] RNA was assumed to represent the rotor blades, hub and pitch system.
[b] # is the blade number from 1 to 3.

Normally, a joint probability density function of wind and waves is considered within DLC 1.2 employed in this case study for lifetime fatigue estimates. However, this was not performed because the OpEx reduction results presented in Sect. 6.4 focus solely on the loading reductions experienced by the rotor, which was assumed to be aligned with the wind direction at all times. However, a joint wind and wave probability density function should be considered when further investigating the benefits to the tower (this will be assessed in the tower design case study in Chapter 7) and moorings. The H_s and T_p were adjusted to suit each mean wind speed integer within the OpenFAST simulations used to determine the failure rates, with

values interpolated from those defined by Allen et al. (2020), and shown in Table 6.4. Within the O&M modelling simulations, the access of the vessels to the sites and the FOWTs (when the turbines were in parked condition, where LAC was inactive) were determined based on the specific wind-wave conditions of each individual site.

Table 6.4: IEC DLC 1.2 NTM wind and wave conditions, from Allen et al. (2020).

$\bar{V}_{hub}$, ms^{-1}	3	5	7	9	11	13	15	17	19	21	23	25
H_s, m	1.10	1.14	1.25	1.43	1.69	2.01	2.39	2.83	3.34	3.82	4.27	4.80
T_p, s	8.52	8.41	8.16	7.83	7.55	7.45	7.55	7.85	8.28	8.75	9.22	9.70

When analysing the time-to-failure data for the sub-systems with multiple relevant loading outputs outlined in Table 6.3, namely the blade roots, a first-to-fail approach was taken such that the loading with the lowest time-to-failure was used for comparison between the FB-only and FFCPC+FFIPC control strategies.

6.4 Results

Within this Section, the reductions of the failure rates of turbine sub-systems enabled by the implementation of LAC are determined. Subsequently, the impact of these reductions is quantified through the simulation of OpEx over the lifetimes of the given ScotWind sites. Firstly, results will be presented documenting the OpEx reductions with only vessel met-ocean limits considered before investigating how the results differ when the additional workability limits of the four FOW platform designs (outlined by Scheu et al., 2018) are applied. The OpEx results of the FOWTs using LAC and with the vessel and floating platform accessibility limits imposed will be compared to the costs of their equivalent BFW site, with turbines using the FB-only controller (baseline failure rates) and only vessel met-ocean limitations imposed.

6.4.1 LIDAR-assisted control failure rate reductions

The results of the MLife post-processing using the data from the four random seed one-hour OpenFAST simulations are shown in Table 6.5. Screenshots of an example MLife input file used when performing the post-processing can be found in Appendix D.2. Results are presented as normalised values of failure rate compared to the FB-only controller.

Table 6.5 indicates that the FFCPC+FFIPC strategy was able to deliver significant reductions to the failure rates of turbine sub-systems over the FB-only controller (by up to 46%) at all of the investigated sites. The up to 30% reductions in the failure rate of the tower base may not be significant for O&M costs due to the relatively low failure rate of the tower, but reinforce the potential shown for structure optimisation, whereby it may be possible to use a thinner, lighter

Table 6.5: Normalised values of failure rates for turbine sub-systems resulting from the loads recorded by the OpenFAST simulation outputs when using FFCPC+FFIPC relative to the FB-only controller at each ScotWind site.

		Normalised failure rate			
Sub-system	**OpenFAST output**	**Baseline**	**E1**	**NE1**	**NE3**
RNA[a]	RtFldMxh	1.00	0.81	0.80	0.79
Pitch bearings	RootMyc#[b]	1.00	0.56	0.55	0.54
Blade roots	RootMyb#[b]	1.00	0.57	0.57	0.57
Tower base	TwrBsMyt	1.00	0.73	0.72	0.70
Moorings	FAIRTEN1	1.00	0.92	0.91	0.90
	ANCHTEN1	1.00	0.92	0.91	0.90

[a] RNA was assumed to represent the rotor blades, hub and pitch system.
[b] # is the blade number from 1 to 3. Values are means.

tower. This will be investigated in Chapter 7. The results shown in Table 6.5 also highlighted the promising potential to enable failure rate reductions resulting from the decreased loadings applied to the pitch bearings, the blade roots and the mooring system. However, concerning the failure rate reductions of the pitch bearings, it is important to note that the failure rate reductions reported here were solely based on the loadings imparted on the components and, therefore, they do not account for failures arising from mechanical wear of the pitch system from control actuation, which may be expected to increase due to the increased pitch rate reported in Chapter 5.

For the purpose of this case study, the most significant result was that of the rotor total load, assumed to lead to failures within the RNA. For each of the sites, when using LAC, an approximately 20% reduction in the failure rate (compared to when using the FB-only controller) resulting from the rotor total load was recorded. The results also indicated marginally greater reductions in the failure rate at site NE3 than at sites NE1 and E1, by up to 3%. This was due to its slightly higher mean wind speed at the site (as shown in Table 6.2), meaning that more of the turbines' lifetimes were expected to be spent with the FF controller operating in above-rated wind speeds, where the loading reductions could be achieved.

6.4.2 Baseline comparison

As discussed in Sect. 6.4.1, the LAC-enabled simulations saw a reduction in the fatigue and failure rates of the RNA, pitch bearings, blade roots, tower base and mooring system, by up to 46%, relative to the FB-only control simulations. Within the literature surrounding offshore wind turbine failure rates, there is a variety of nomenclature used, making it difficult to draw comparisons between the results of different studies. In the review by Dao et al. (2019), the definition of RNA is made up of the sub-assemblies of blades, hub, air brake and pitch system. These elements were matched to the failure rates for the sub-assemblies detailed in

the work of Carroll et al. (2016). For all FOW simulations performed within the O&M model, only the failure rates associated with the RNA were adjusted based on the normalised values presented in Table 6.5. The same failure rate reductions were assumed to apply to all wind turbines within the array. All other failure rates applied corresponded to those of the BFW equivalent site from Carroll et al. (2016).

The reduction in total OpEx and the sub-costs of the FOW farms with the reduced failure rates obtained from the implementation of LAC (relative to the OpEx and sub-costs of the FOW farms when using the baseline failure rates, associated with the use of the FB-only controller), are given in Table 6.6. This analysis was based on vessel limits only using the existing StrathO&M model from Dinwoodie et al. (2012) and so did not consider the additional workability constraints of the FOWTs.

Table 6.6: OpEx savings breakdown based on the reduction of failure rates.

	E1	NE1	NE3
Lost Revenue	5.38%	8.58%	1.70%
Repair Costs	7.10%	10.3%	11.0%
Total OpEx	**2.89%**	**4.29%**	**5.05%**

The case study's outcomes indicate a notable 3-5% reduction in operational expenditures when the LAC-enabled reduced failure rate of the RNA was applied (relative to the operational expenditures when using the baseline failure rate of the RNA, associated with the use of the FB-only controller), displaying variability among different sites. Particularly, NE3 emerged as the most advantageous because this exhibited the most substantial decrease in RNA failure rates compared to the other locations (up to 21%, as shown in Table 6.5) due to its higher mean wind speeds (shown in Table 6.2), meaning that LAC was active for a greater proportion of its lifetime. The overall decline in OpEx was attributed to lowered lost revenue, due to reduced downtime, and repair expenses. The key OpEx savings resulted from the potential for transport costs to be reduced by decreasing the number of vessels (and associated hiring costs) due to the reduced required transfers resulting from the lifetime extension of the components. The relationship between the reduction of failure rates, and therefore the number of transfers required, and the OpEx decrease is highlighted in Fig. 6.3. This illustrates that as the reduction to the failure rate achieved from LAC increased, the OpEx savings increased because fewer transfers to the wind farm site were required during the lifetime, thereby reducing costs of vessel hire and personnel. The reduced failure rates also meant that components required repair less frequently, resulting in the repair cost savings shown in Table 6.6.

These results are representative of the impact of the reduced failure rates on the WindCrete and ActiveFloat designs from CoreWind (2020), where workability would not impact operation using an SOV strategy with a H_s limit of 3 m.

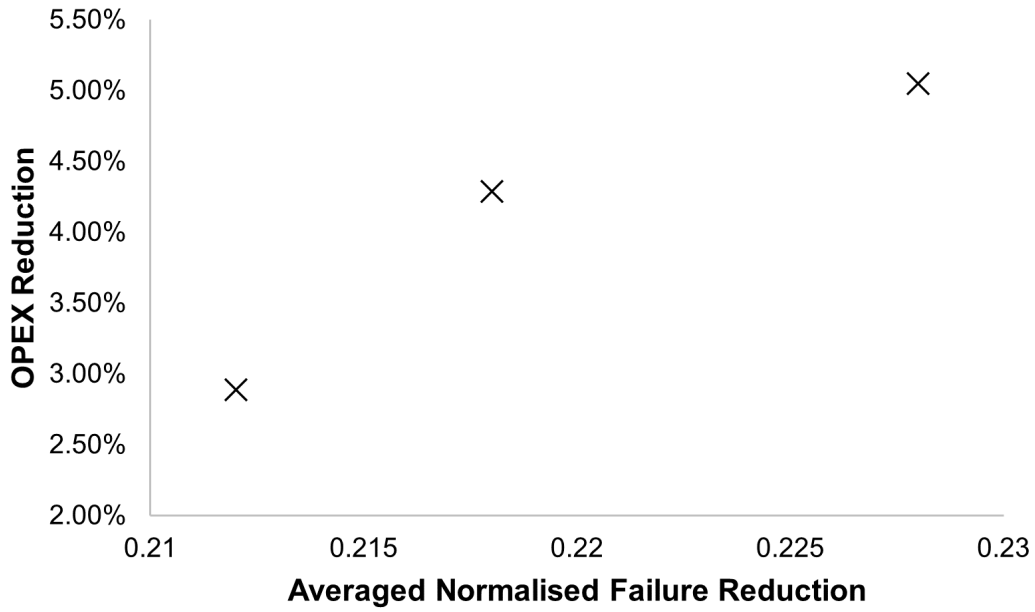


Figure 6.3: Relationship between average normalised failure rate reduction of the RNA and OpEx reductions.

6.4.3 Workability impact

As previously discussed in Sects. 1.3.7 and 2.4, FOWTs are also subject to additional weather constraints due to human exposure to the motion of the assets. As shown in Fig. 6.2, workability limitations are unique to each platform design and limit access to the turbine during specific combinations of T_p and H_s. Using designs A-D from Scheu et al. (2018), the workability limitations were imposed on access to the site, in addition to the existing vessel limits. As previously discussed, CoreWind (2020) also provides WI limitations for the ActiveFloat semi-submersible and WindCrete spar designs. However, these limitations only exist for WI combinations of 3.5 m and above, which were already captured within the vessel limits. Therefore, the OpEx results for the ActiveFloat and WindCrete would have been unchanged from those presented in Table 6.6.

Within this analysis, the WI limitations were placed on both the FOW baseline comparison and the FOW with the LAC-enabled reduced RNA failure rates. It was assumed that any H_s-T_p combinations resulting in WIs less than 1 were inaccessible. The estimated OpEx savings are shown in Fig. 6.4.

Unlike the results with no consideration of WI, the highest OpEx savings were not consistent with the site with the highest reduction in failure rates. Site E1 had the highest OpEx savings for all platform designs except for Design B. Site E1 experienced the highest OpEx primarily because it underwent the most significant reduction in accessibility resulting from the incorporation of workability limits, influenced by its greater distance from shore and its higher

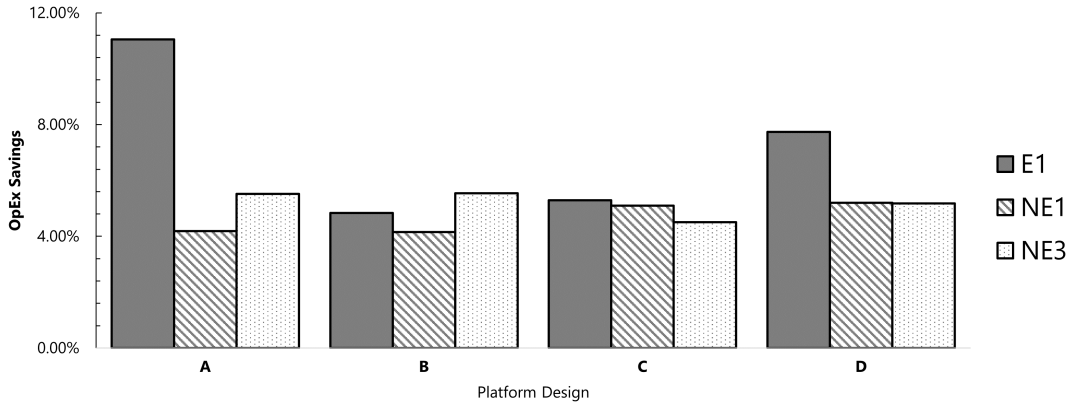


Figure 6.4: OpEx savings resulting from reduced failure rates for FOW sites with imposed WI limitations.

mean H_s, as shown in Table 6.1. As a result, it reaped the most substantial advantage from the decrease in failure rates brought by LAC. This advantage translated into a reduction in the necessary transfers and subsequently minimised the demand for optimal weather conditions. This highlights that sites with accessibility challenges, and therefore high predicted OpEx, can reap the greatest benefit from the addition of LAC.

The variability of the results of the four designs highlights the unique challenges of each specific design, and that there is no "one-fits-all" solution. As with the comparison in Table 6.6, for LAC with no WI considerations, the decrease in OpEx is attributed to the reduction of repair costs and lost revenue. However, for all sites and all designs, the savings in lost revenue was the main contributor to these savings, due to the challenges of accessibility and the link between access and OpEx, as per Van Bussel and Bierbooms (2003).

When comparing the FOW sites with both LAC and motion limitations against the BFW equivalent sites with FB-only control (i.e. without LAC) for specific designs, in some cases, the total OpEx for motion-limited FOW with LAC-delivered failure reductions was less than that of the BFW with original failure rates and no motion limits, by up to 4.97%. The percentage savings in OpEx for E1, NE1 and NE3 are detailed in Table 6.7.

Table 6.7: Percentage savings in OpEx from FOW with WI limits and LAC failure rate reductions compared to the equivalent BFW baseline, where "-" indicates that the BFW was cheaper than the FOW equivalent site. FOW was not more cost-effective at E1 for any of the FOWT designs.

ScotWind site	Design A	Design B	Design C	Design D
E1	-	-	-	-
NE1	-	-	1.89%	-
NE3	4.41%	4.97%	4.57%	0.39%

Site NE3 saw consistent savings across all FOWT designs when using the LAC-enabled failure rate reductions and WI limits relative to the BFW equivalent site using the FB-only control failure rates (by up to 4.97%), except for Design D. The NE1 FOW site with LAC-enabled failure rate reductions and WI limits was only more cost-effective than the BFW equivalent with FB-only control failure rates for Design C, by 1.89%. It was not found to be more cost-effective to apply the LAC-enabled failure rates and WI limits for any of the designs at the E1 FOW site due to the site's additional workability restrictions, which resulted in larger costs than the BFW equivalent site using the FB-only control failure rates.

The results show that while FOW sites face additional challenges due to location and motion, the implementation of specific technologies and control strategies, such as LAC, have the potential to bring certain aspects of the cost of some FOW sites below that of the equivalent BFW site using FB-only control.

Chapter 7

Tower design case study

As documented in Chapter 5, the LAC addition was capable of significantly reducing the variation in the rotor speed, platform pitch, platform surge and the tower base fore-aft bending moment of the VolturnUS-S, relative to its FB-only controller. This presents an opportunity to perform modifications to the structural properties of the FOWT's tower, with a view to reducing CAPEX through decreased material usage.

Within this case study, the baseline and modified towers are denoted as supporting a FOWT (the VolturnUS-S in this case study) that used either FB-only control (henceforth termed an FB-only-controlled FOWT) or LAC (henceforth termed a LIDAR-assisted-controlled FOWT). Thus, the following three distinct tower design cases will be examined:

1. The baseline tower supporting an FB-only-controlled FOWT.
2. The modified tower supporting an FB-only-controlled FOWT.
3. The baseline and modified tower supporting a LIDAR-assisted-controlled FOWT.

Case 2 was expected to have a negative impact with respect to Case 1, by way of increased stress on the tower and increased nacelle motions. It was hypothesised that, through the addition of LAC (Case 3), these negative impacts could be mitigated such that a lighter tower was feasible and similar levels of stress to Case 1 could be maintained.

Within Chapter 5, the LAC control addition (Case 3) was shown to reduce the standard deviation and range of the rotor speed of the VolturnUS-S by up to 60%, thereby reducing (relative to Case 2 and Case 1) the 1P and 3P frequency ranges and increasing the gap between them. This is highlighted in Fig. 7.1, which is adapted from Fig. 1.8, where reductions to the standard deviation of the rotor speed allow for reductions to the higher frequency regions (i.e. at the rated rotor speed where LAC is active in above-rated wind speed conditions) of the 1P and 3P constraint bands, such that

$$\Omega_{r,const,i,min} = min[\bar{\Omega}_r(V) - 2\Omega_{r,SD,i}(V)] \tag{7.1a}$$

$$\Omega_{r,const,i,max} = max[\bar{\Omega}_r(V) + 2\Omega_{r,SD,i}(V)] \tag{7.1b}$$

where $\Omega_{r,const,i}$ is the design constraint rotor rotational frequency of a control strategy, i (FB-only or LAC), $\bar{\Omega}_r(V)$ is the mean rotor rotational frequency at a wind speed, V, and $\Omega_{r,SD,i}(V)$ is the standard deviation (of six ten-minute simulations) of the rotor rotational frequency at a wind speed using a control strategy. LAC was unable to reduce the constraint bands at the lower frequency regions because the rotor rotated at these frequencies in below-rated wind speed conditions, when LAC was inactive.

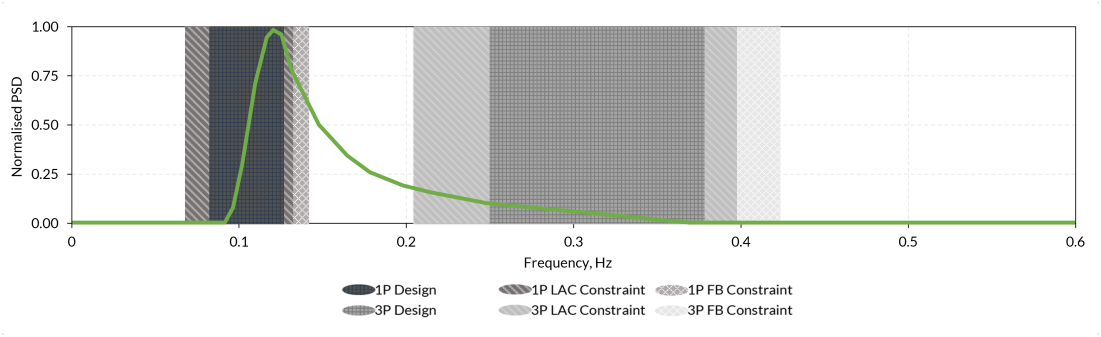


Figure 7.1: Rotor 1P and 3P design and constraint frequency ranges of the VolturnUS-S when using FB-only control (Case 1 or 2) or LAC (Case 3). The green line depicts a typical JONSWAP irregular wave frequency spectrum, as reported by Allen et al. (2020)

This presented the opportunity to modify the FOWT's tower design, such that its first natural frequencies are reduced to a lower value, which can either remain above the 3P range, or within the gap between the 1P and 3P ranges. This was achieved through reductions applied to the tower's outer diameter and/or shell thickness. Reductions to these parameters rendered the tower more elastic and flexible, thereby reducing its natural frequency. The loading reduction capability of LAC was also useful for reducing the stress on the tower, which was expected to offset any increases that occurred when reducing the geometrical dimensions of the tower.

This Chapter outlines the FOWT tower modification experiments performed within NREL's WEIS structural design tool (summarised in Appendix A.4). The goal was to achieve a simplified FOWT tower that experienced similar loadings and stress when supporting a LIDAR-assisted-controlled FOWT than when using the baseline tower supporting an FB-only-controlled FOWT. DLC simulations were performed to ensure that the modified designs remained within operational and structural strength limits, following similar methodologies used by Papi et al. (2023) while additionally assessing the impacts of wave-wind misalignment conditions on the stress of the towers. The datasets obtained from the OpenFAST (within WEIS) simulations of the VolturnUS-S with modified tower geometry performed in this case study can be found via links to data repositories referenced in Appendix C.

7.1 Methodologies

NREL's WEIS/WISDEM structural design tool, outlined in Appendix A.4, was used to perform this study. WEIS is primarily intended as a wind turbine design optimisation tool and uses OpenMDAO, an open-source framework for efficient multidisciplinary optimisation, developed by Gray et al. (2019), to achieve this. However, for the case study performed in this work, the optimisation approach was deemed to be overly complex, from both a computation time and design perspective. As an alternative, design variables were selected and simulations were performed independently at different set values for each design variable.

7.1.1 Design variables

This study focused on two primary design variables - the tower outer diameter and the tower shell thickness. Percentage reductions to the default values of the VolturnUS-S's tower design variables were applied independently, with increasing magnitudes.

The floating platform's geometry was also intended to be modified, but such changes would have significantly altered the hydrodynamic response of the FOWT. Determination of the new hydrodynamic coefficients (used within the HydroDyn module of OpenFAST) of the modified platforms would have required the use of pyHAMS (see Barter, 2024a) via RAFT (Response Amplitudes of Floating Wind Turbines, see Barter, 2024b) within WEIS, though this is only achievable when performing OpenMDAO optimisation. The OpenMDAO optimisation was not used within this work and so the use of pyHAMS within RAFT for calculating the new hydrodynamic coefficients of the modified floating platforms was not possible in this case study. Alternatively an external design tool, such as WAMIT (developed by C. H. Lee & Newman, 2006) can be used to determine the hydrodynamic coefficients of the modified floating platforms. WAMIT would have required the creation of a panel mesh of the platforms to calculate their hydrodynamic coefficients. This process was further complicated by the relatively complex geometry of the VolturnUS-S semi-submersible platform. The complexity and time-consuming nature of this task (within WAMIT) were deemed to be outside of the scope of this work, though it is recommended that future work seeks to investigate the possibility of performing modifications to the floating platform geometry. Therefore, within this case study, the structural modifications focused solely on the turbine's tower.

Changes to the tower do not change the geometry of the semi-submersible platform and WISDEM automatically adjusts the ballast mass of the platform to account for the change in mass of the tower. Therefore, it was deemed that the hydrodynamic coefficients did not need to be recalculated using pyHAMS or WAMIT based on the tower design modifications applied in this case study because the added mass/radiation damping coefficients and the first-order wave loads do not depend on the mass distribution. Second-order wave loads do depend on the mass distribution, but these were not considered in this case study.

Baseline design

The baseline VolturnUS-S tower diameter and thickness properties with height above the still-water level (SWL), from Barter (2023), are shown in Table 7.1.

Table 7.1: Outer diameter and shell thickness properties at different heights of the baseline tower (above the SWL) of the VolturnUS-S as within its default parameterised `.yaml` input file within WEIS (see Barter, 2023).

Height above SWL, m	Outer diameter, m	Shell thickness, m
15	10.0	0.090
28	10.0	0.087
41	10.0	0.070
54	10.0	0.067
67	10.0	0.049
80	10.0	0.046
93	10.0	0.028
106	10.0	0.026
119	10.0	0.010
136	10.0	0.010
144	6.50	0.006
Tower mass, ton	1,483	

Though the reference documentation by Allen et al. (2020) states that the tower of the VolturnUS-S is tapered, such that its outer diameter decreases with height, this design was later found to lead to resonance with the rotor 3P frequency. Consequently, the tower was modified to be modelled with a uniform 10 m outer diameter within OpenFAST, except for the point of the tower top, where an outer diameter of 6.5 m was required to integrate with the nacelle's bedplate (see Barter, 2023). The modifications made in this work applied reductions to all tower sections except for the highest section when adjusting the outer diameter, in order to maintain the compatibility of the tower with the nacelle. The shell thickness of the baseline tower decreases with height, and equal percentage reductions were applied to the shell thickness of each section to maintain the tapered design.

7.2 Simulation set-up

An experimental simulation set-up was curated to ascertain the impact of changes to each variable on the turbine's responses under different wind and wave conditions. Each design variable was reduced by increasing amounts in turn while keeping the other parameter at its default value for each section of the tower. This was to assess how changes to each parameter

affected a FOWT that used the FB-only controller and to allow for comparison between the parameters' impacts. The experimental set-up, documenting the percentage reductions to the baseline design for each parameter for each simulation experiment number, is shown in Table 7.2.

Table 7.2: Percentage reductions applied to the baseline tower outer diameter or shell thickness for each simulation experiment number.

No.	Outer diameter, %	Shell thickness, %
1	0	0
2	-10	0
3	-15	0
4	-20	0
5	-35	0
6	0	-20
7	0	-30
8	0	-40
9	0	-50
10	0	-70

Simulations were first performed of the FOWT that used the baseline (Case 1) and modified (Case 2) tower designs while using FB-only control to determine which attributes were negatively impacted by the changes to the tower. Primarily, these were expected to be the tower base fore-aft and side-to-side bending moments, platform pitch, tower-top displacement and nacelle acceleration, per previous literature findings by Papi et al. (2023). Simulations of the towers supporting a LIDAR-assisted-controlled FOWT enabled were then performed (Case 3), and results were compared to those of the baseline tower design supporting an FB-only-controlled FOWT (Case 1).

The most favourable lightweight tower design of a LIDAR-assisted-controlled FOWT was then identified following an evaluation of LAC's ability to offset the increases in stress resulting from the reductions to the baseline tower outer diameter or thickness. Subsequently, further analysis was performed. Firstly, this sought to estimate the stress on the most favourable lightweight tower design in conditions where LAC was inactive in extreme conditions, when the turbine was in its parked state, with results presented in Sect. 7.5. Secondly, the further analysis subjected the lightweight tower supporting a LIDAR-assisted-controlled FOWT and the baseline tower supporting an FB-only-controlled FOWT to wind-wave directionality misalignment, with results presented in Sect. 7.6.

Design load cases

The DLCs from the IEC standards (see IEC 61400-3-1:2019, 2019) that were simulated in this case study were as follows:

- DLC 1.2: Pertaining to normal operation in Normal Sea States (NSS) and with wind turbulence characterised by the NTM. Wind speeds between 5 ms^{-1} and 25 ms^{-1} in 2 ms^{-1} intervals were considered. Expected values of H_s and T_p were used for each mean wind speed, as shown in Table 6.4. Co-directional and misaligned wind and waves were simulated in this case study, although only co-directional waves were applied during the initial screening of the modified tower designs. This DLC was used to assess the ability of LAC to offset the fatigue loading of the tower.
- DLC 1.6: Pertaining to normal operation in Severe Sea States (SSS) and with wind turbulence characterised by the NTM. Wind speeds between 5 ms^{-1} and 25 ms^{-1} in 2 ms^{-1} intervals were considered. This DLC used a fifty-year return period H_s and an expected value of T_p conditioned on H_s. Only co-directional wind and waves were modelled for this DLC. This DLC subjected the tower to ultimate loading and was used to verify that the modified designs could withstand such conditions with and without LAC enabled.
- DLC 6.1: In this DLC, the FOWT is modelled in its parked condition under Extreme Wind Model (EWM) wind and Extreme Sea State (ESS) waves. Only co-directional wind and waves were modelled for this DLC. Similarly to DLC 1.6, DLC 6.1 was used to verify whether the modified designs could withstand extreme conditions when the turbine was in its parked condition, where LAC was inactive.

Six seeds of ten-minute simulations (700 seconds were simulated but the first 100 seconds were discarded to remove start-up transients) were performed for each DLC between the specified mean wind speed ranges and intervals. A 100-second allowance for start-up transients was found to be sufficient when undertaking this study. Ten-minute length simulations were performed in this case study instead of one-hour length simulations (as used in the simulations performed in Chapter 5) because the computation time when using OpenFAST within WEIS was vastly greater when performing one-hour length simulations.

7.2.1 Stress analysis

To allow for the comparison of the loadings applied to the towers of differing geometry to the baseline tower design, the equivalent design stresses were determined. Standard deviations (for DLC 1.2) and the maxima (for DLCs 1.6 and 6.1) of the loads on the tower were determined from the time series results at different tower heights. The standard deviation and maxima of the equivalent design stresses, $\sigma_{y,ED,j}(z)$, on the cross-sections were computed using methodologies outlined by Papi et al. (2023) while also considering the tower side-to-side bending moments

$$\sigma_{y,j}(z) = \frac{N_{ED,j}(z)}{A_x(z)} + \frac{M_{y,ED,j}(z)}{W(z)} + \frac{M_{x,ED,j}(z)}{W(z)} \tag{7.2}$$

where, for a given cross-section at height z of the tower, $N_{ED,j}(z)$ represents the axial force and $M_{yt,ED,j}(z)$ and $M_{xt,ED,j}(z)$ represent the tower fore-aft and side-to-side bending moments, with j representing the standard deviation or maxima, respectively. $A_x(z)$ and $W(z)$ represent the cross-sectional area of the tower shell and the resistance moment of each tower section, respectively.

$$A_x(z) = \pi \left(\left[\frac{D_o}{2} \right]^2 - \left[\frac{D_o}{2} - t \right]^2 \right) \tag{7.3}$$

$$W(z) = \pi \frac{D_o^4 - (D_o - 2t)^4}{32 D_o} \tag{7.4}$$

where D_o is the tower outer diameter and t is the tower shell thickness.

7.3 Feedback-only control results

This Section outlines the results of the experiments performed by applying reductions to one of the tower design variables (outer diameter or shell thickness) while maintaining the other at its baseline value, for each tower section. Results of the simulations of the FOWT using FB-only control while supported by the modified towers were first explored to assess how the changes affected the performance and loadings of the FOWT (Case 2). An assessment of how the addition of LAC (Case 3) can offset these will follow in Sect. 7.4. Simulations were performed under DLC 1.2 for this initial analysis, as these are the fatigue load accumulating conditions impacting the FOWT over its lifetime. This Section analyses results obtained from a mean above-rated wind speed of 13 ms^{-1}, as this was found to result in the greatest loadings on the tower within DLC 1.2.

7.3.1 Outer diameter

Figure 7.2 shows the results of the reduced tower outer diameter experiments. Results are normalised to those of the baseline tower design.

Figure 7.3 shows the first fore-aft natural frequency points of the different tower designs. These were obtained from PSD analysis of the tower base fore-aft bending moment.

As can be seen in Fig. 7.2, decreases to the tower outer diameter resulted in greater tower base loads than the baseline design, peaking at an 39% increase when a 20% reduction to the tower outer diameter was applied. By cross-referencing with Fig. 7.3, this corresponded to a tower first fore-aft natural frequency of approximately 0.38 Hz, which was at the 3P frequency

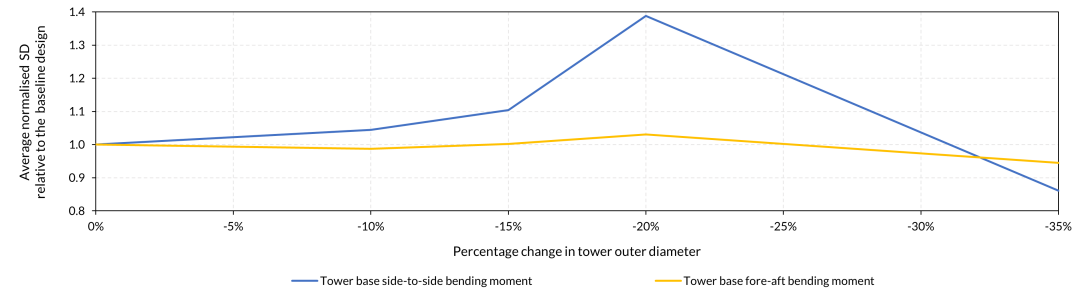


Figure 7.2: Impacts on the standard deviations of the tower base loads with increasing reductions in the tower outer diameter, normalised to the baseline design under DLC 1.2 ($\bar{V}$ = 13 ms^{-1}, NTM, TI Category B) with irregular waves (H_s = 2.01 m, T_p = 7.45 s).

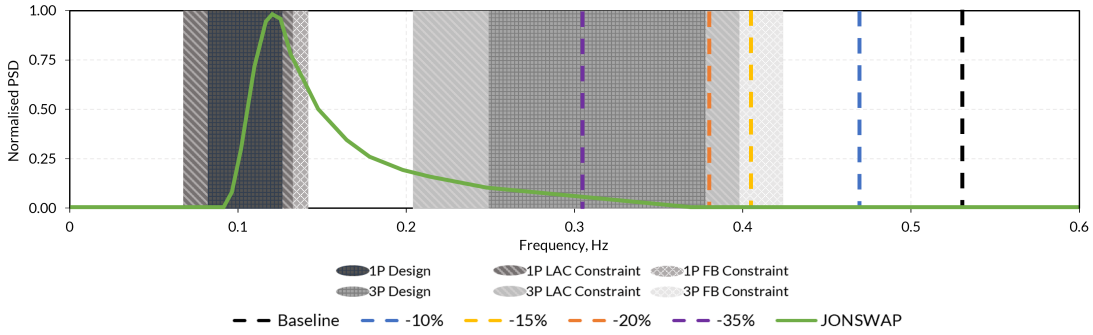


Figure 7.3: Vertical lines imposed on the rotor 1P and 3P frequency ranges of the VolturnUS-S, indicating the calculated tower first fore-aft natural frequencies of the tower designs with reduced outer diameter.

of the rotor at its rated speed (7.56 rpm), leading to the resonance and increased loads seen in Fig. 7.2. The tower side-to-side bending moment was most affected by the reductions to the tower's outer diameter. A time series plot showing the variation in the tower base side-to-side bending moment for different tower outer diameter reductions is shown in Fig. 7.4.

As can be seen in Fig. 7.4, resonance occurred when the tower outer diameter was reduced by 20%, relative to the baseline design, evidenced by the erratic variations in the tower base side-to-side bending moment. This was also evidenced by large peaks in the PSD plots of the tower base side-to-side bending moment at different mean wind speeds (Fig. 7.5).

Figures 7.2 and 7.4 also indicate that the tower loadings began to reduce again once the reduction to the baseline tower outer diameter exceeded 20%. This indicates that the tower first fore-aft natural frequency was below the 3P frequencies common for the rotor rotation in above-rated wind speeds, and so resonance did not occur at these wind speeds. However, when investigating the PSDs of the tower base side-to-side (M_{xt}) and fore-aft (M_{yt}) bending moments at various mean wind speeds under DLC 1.2, including below-rated wind speeds (Fig. 7.5), a peak occurred for the tower design with 35% reduced diameter (compared to

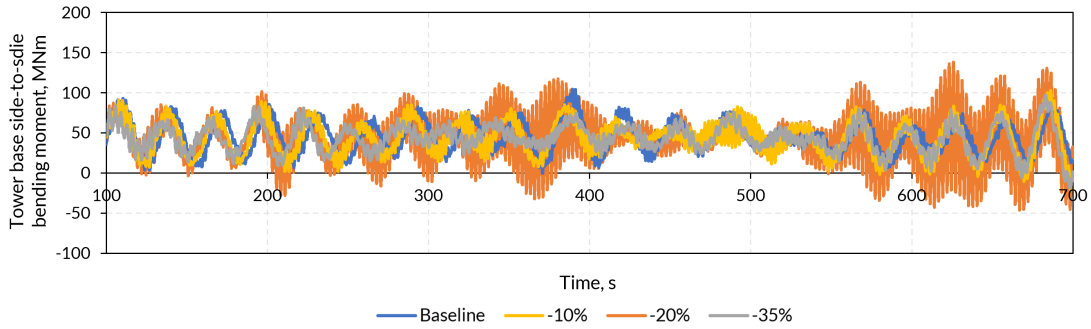


Figure 7.4: Time series of the tower base side-to-side bending moment with reduced tower outer diameter under DLC 1.2 ($\bar{V}$ = 13 ms^{-1}, NTM, TI Category B) with irregular waves (H_s = 2.01 m, T_p = 7.45 s).

the baseline design) at a mean wind speed of 9 ms^{-1}. This was because its first fore-aft natural frequency was situated close to the 3P frequency of the mean rotor speed at this wind speed (0.33 Hz). Therefore, towers with first fore-aft natural frequencies within the 3P operating range will, unsurprisingly, result in resonance and increased tower loads at below-rated wind speeds. Figure 7.5 also indicates the tower base bending moment responses of the other designs, where much less significant peaks occurred for those whose first fore-aft natural frequency lay above 3P. The natural frequency peaks in the fore-aft bending moment plots (Figs. 7.5b, d and f) are not as clearly defined as those for the side-to-side moment (Figs. 7.5a, c and e) as greater aerodynamic damping is present in this direction. In frequency ranges where inertial and gravitational loads induced by hydrodynamic loads are relevant, all of the modified tower designs responded similarly to the baseline tower.

7.3.2 Shell thickness

Figure 7.6 shows the results of the reduced tower shell thickness experiments. Results are normalised to those of the baseline tower design.

Figure 7.6 follows a similar trend to that for reductions to the tower outer diameter in Fig. 7.2, in that the standard deviations of the tower loads ultimately increased (relative to the baseline design) as the reduction to the tower shell thickness increased. However, the key difference observed is that reductions to the tower shell thickness had a lesser effect on the tower first fore-aft natural frequency, and so greater percentage reductions could be made before the frequency entered the 3P range at 0.38 Hz, as shown in Fig. 7.7. The tower loads peaked (at 40% greater than the baseline design) at a 50% reduction in shell thickness (relative to the baseline tower design), compared to a 20% reduction in outer diameter shown in Fig. 7.2, indicating that reductions to the shell thickness offered the superior route to tower mass reductions.

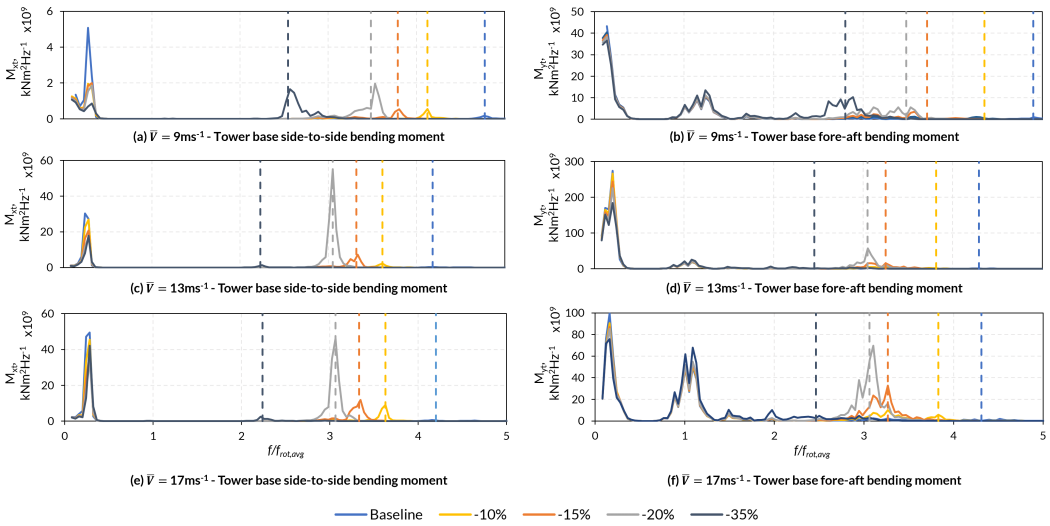


Figure 7.5: PSD plots of the tower base side-to-side and fore-aft bending moments of the modified tower outer diameter designs at different mean wind speeds within DLC 1.2. The x-axis is normalised to the mean rotational frequency of the rotor at each wind speed (1 = 1P and 3 = 3P). The dashed vertical lines indicate the first tower natural frequencies of each design.

Similarly to the tower outer diameter, as the tower shell thickness was reduced further, the tower loads began to decrease again as resonance with the rotor rotation no longer occurred. This reached a minimum at a 70% reduction to the baseline tower shell thickness, indicating that the tower frequency lay below the 3P frequencies common for the rotor speed in above-rated wind speeds. When investigating the PSDs of the tower base side-to-side and fore-aft bending moments of the different towers at various mean wind speeds under DLC 1.2 (Fig. 7.8), a large peak occurred for the tower design with 70% lesser shell thicknesses (relative to the baseline design) at a mean wind speed of 9 ms^{-1}.

Similarly to the 35% reduced diameter design shown in Fig. 7.5, this was due to its first fore-aft natural frequency being situated close to the 3P frequency of the mean rotor speed at this wind speed (0.33 Hz), which was below the 3P frequency of the rated rotor speed (0.38 Hz). Therefore, towers with reduced shell thickness whose first fore-aft natural frequencies lie within the 3P operating range will also, unsurprisingly, result in resonance and increased tower loads compared to the baseline design at below-rated wind speeds. Figure 7.8 also indicates the tower base bending moment responses of the other designs, where much less significant peaks occurred for those whose tower first natural frequencies lay above the 3P range.

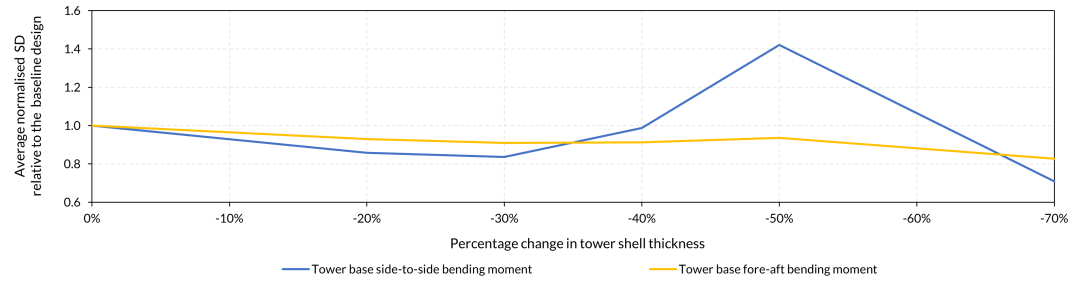


Figure 7.6: Impacts on tower loads and tower DEL with increasing reductions in the tower shell thickness, normalised to the baseline design under DLC 1.2 ($\bar{V}$ = 13 ms^{-1}, NTM, TI Category B) with irregular waves (H_s = 2.01 m, T_p = 7.45 s).

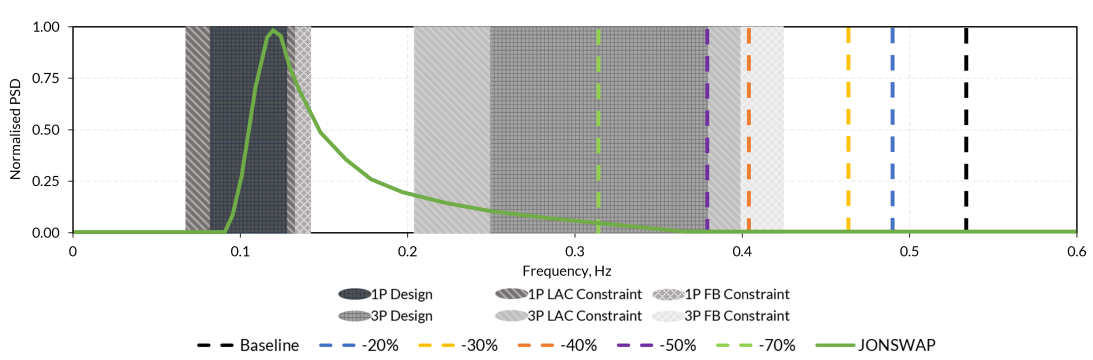


Figure 7.7: Vertical lines imposed on the rotor 1P and 3P frequency ranges of the VolturnUS-S when using FB-only control or LAC, indicating the calculated tower first fore-aft natural frequencies of the tower designs with reduced shell thickness.

7.3.3 Analysis of other parameters

Figure 7.9 shows the frequency domain PSD responses of other parameters relevant to tower design of the different tower outer diameter and shell thickness designs - Θ_p, the x-direction tower-top displacement ($d_{tt,x}$) and $\ddot{x}_{tt}$. Peaks within Fig. 7.9 occurred at the peak of the JONSWAP wave spectrum and the 1P frequency at approximately 0.126 Hz as well as at the frequencies pertinent to turbulent wind (< 0.03 Hz) and the rated rotor rotational frequency (0.38 Hz).

The detriment brought by reducing the tower outer diameter by 20% or reducing the tower shell thickness by 50% (compared to the baseline design) is clearly highlighted in Fig. 7.9. These designs resulted in greater nacelle accelerations at the turbine's rated rotor frequency due to the tower first natural frequencies being situated at the rated rotor speed 3P point.

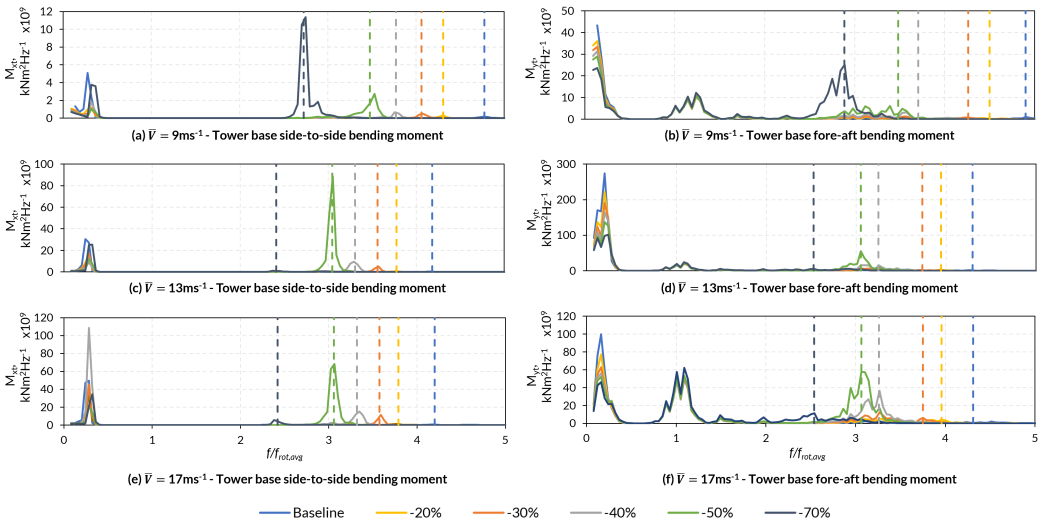


Figure 7.8: PSD plots of the tower base side-to-side and fore-aft bending moments of the modified tower shell thickness designs at different mean wind speeds within DLC 1.2. The x-axis is normalised to the mean rotational frequency of the rotor at each wind speed (1 = 1P and 3 = 3P). The dashed vertical lines indicate the first tower natural frequencies of each design.

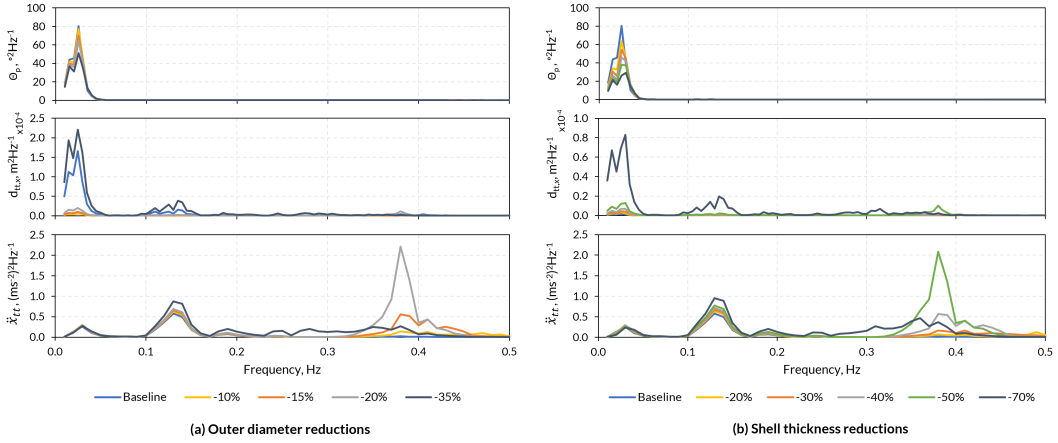


Figure 7.9: PSD plots of three parameters when using the modified tower designs with outer reduced diameter or shell thickness, relative to the baseline design, under DLC 1.2 ($\bar{V}$ = 13 ms^{-1}, NTM, TI Category B) with irregular waves (H_s = 2.01 m, T_p = 7.45 s).

Reductions from the other design points generally led to reductions in the response of the platform pitch at frequencies pertinent to turbulent wind. This was more apparent for reductions to the tower shell thickness than the tower outer diameter. The tower-top displacement was increased when the tower fore-aft natural frequency was positioned below the 3P rated rotor speed point, occurring with the 35% reduced outer diameter or 70% reduced shell thickness (compared to the baseline) designs, because these designs were more flexible, leading to greater motions of the nacelle.

7.3.4 Equivalent design stresses

Though the plots shown in Figs. 7.2 and 7.6 indicate how the standard deviations of the tower base loads were affected by the placement of the natural frequencies of the tower, these do not account for the change in equivalent design stress resulting from the changes in the tower geometry. The smaller diameters and thicknesses of the modified towers compared to the baseline mean that they experienced larger stress for a given load cycle amplitude, reducing the fatigue resistance of the proposed designs. Equation 7.2 was used to determine the standard deviations of the equivalent design stresses ($\sigma_{y,ED,SD}$) at the tower base for each of the modified designs for DLC 1.2 at a mean wind speed of 13 ms^{-1}. When normalised to the $\sigma_{y,ED,SD}$ on the tower base of the baseline design, the relationship between equivalent design stress and tower outer diameter or shell thickness reductions can be observed (Fig. 7.10).

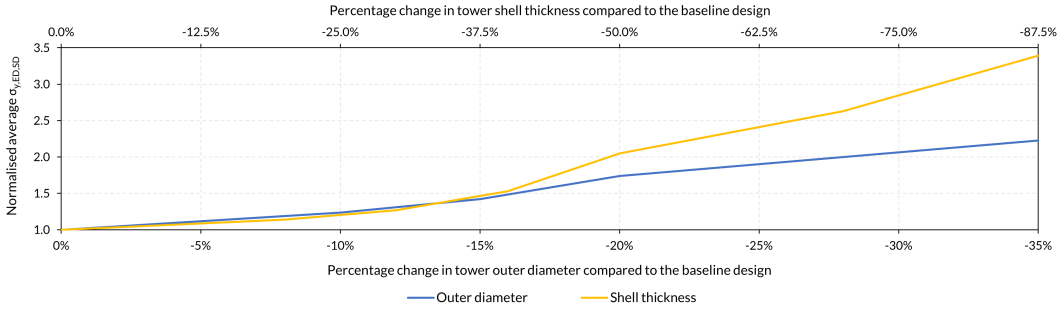


Figure 7.10: Normalised average standard deviations of equivalent design stress on the tower base of the modified tower designs compared to the baseline tower design under DLC 1.2 ($\bar{V}$ = 13 ms^{-1}, NTM, TI Category B) with irregular waves (H_s = 2.01 m, T_p = 7.45 s).

By consideration of $\sigma_{y,ED,SD}$, it is clear that the reductions beyond those outer diameters or shell thicknesses that resulted in the maximum loads in Figs. 7.2 and 7.6 (20% reduced outer diameter and 50% reduced shell thickness, relative to the baseline design) led to increasing stress on the structure, despite the reduced standard deviations of the tower loadings compared to the baseline design, shown in Figs. 7.2 and 7.6. This is because these towers were significantly narrower or thinner than the baseline tower and were, therefore, more prone to stress and fatigue. This finding, coupled with the significantly increased tower-top displacements, nacelle accelerations (Fig. 7.9) and tower base bending moments in below-rated wind speeds (Figs. 7.5 and 7.8), leads to the conclusion that tower designs with first natural frequencies below the rated rotor speed 3P point (0.38 Hz), where the rotor rotates at in below-rated wind speeds and where this LAC application cannot offer any benefit, are not viable for this FOWT, despite the significant mass reductions that they present. This also means that, according to these findings, it is not feasible to design the tower of a FOWT such that its first natural frequencies lay within the 1P and 3P interval.

7.4 LIDAR-assisted control results

Simulations were then performed by applying the single parameter reductions to the tower of a LIDAR-assisted-controlled FOWT (Case 3). The results reported in Chapter 5 showed that LAC was able to deliver reductions to the tower loads (by up to 21%) and, therefore, the stress of the baseline tower design supporting an FB-only-controlled VolturnUS-S. Therefore, this Section investigates whether the implementation of LAC within the FOWT is able to offset the increased loads and stress resulting from reduced structural properties of the tower. There was also an opportunity for the tower first natural frequencies to be placed closer to the rated rotor 3P frequency because the reduced variation of the rotor speed meant that the size of the 3P constraint band was reduced, as shown in Fig. 7.1.

Figure 7.11 compares the impacts of the LAC addition on the $\sigma_{y,ED,SD}$ applied to the tower base, and the impacts on tower frequency and tower mass as a result of reductions to the tower outer diameter or shell thickness. Crossing points are marked to show the points at which the normalised (to the baseline tower supporting an FB-only-controlled FOWT) $\sigma_{y,ED,SD}$ passes one, signifying that the loadings experienced by the lightweight design supporting a LIDAR-assisted-controlled FOWT closely matched those of the baseline tower supporting an FB-only-controlled FOWT.

From Fig. 7.11, the addition of LAC was unable to prevent the resonance, increased loadings and increased stress that occurred when reductions to the tower outer diameter or shell thickness brought the tower's first fore-aft natural frequency close to and within the 3P range (occurring at 0.38 Hz). However, the FF controller was able to deliver reductions to the $\sigma_{y,ED,SD}$, by up to 18% (relative to the baseline tower supporting an FB-only-controlled FOWT), when the tower frequency remained above the 3P range. The tower shell thickness showed greater potential for reductions, with the stress matching point occurring at a 20% reduction, compared to an approximately 6.5% reduction for the tower outer diameter, relative to the baseline tower design supporting an FB-only-controlled FOWT. This corresponded to more significant mass reductions (20%) over the baseline tower design for the tower shell thickness stress matching point, as shown in the bottom plot of Fig. 7.11. Given the relatively minor reduction to the tower mass that was achieved at the tower outer diameter matching point, this design was omitted from the subsequent analysis. The 20% reduced tower shell thickness design showed promise for enabling mass and cost savings and so was carried forward for further analysis. Thus, this design is henceforth termed the lightweight tower design. Figure 7.12 shows the PSD plots of the lightweight tower design supporting a LIDAR-assisted-controlled FOWT, compared to the baseline design supporting an FB-only-controlled FOWT.

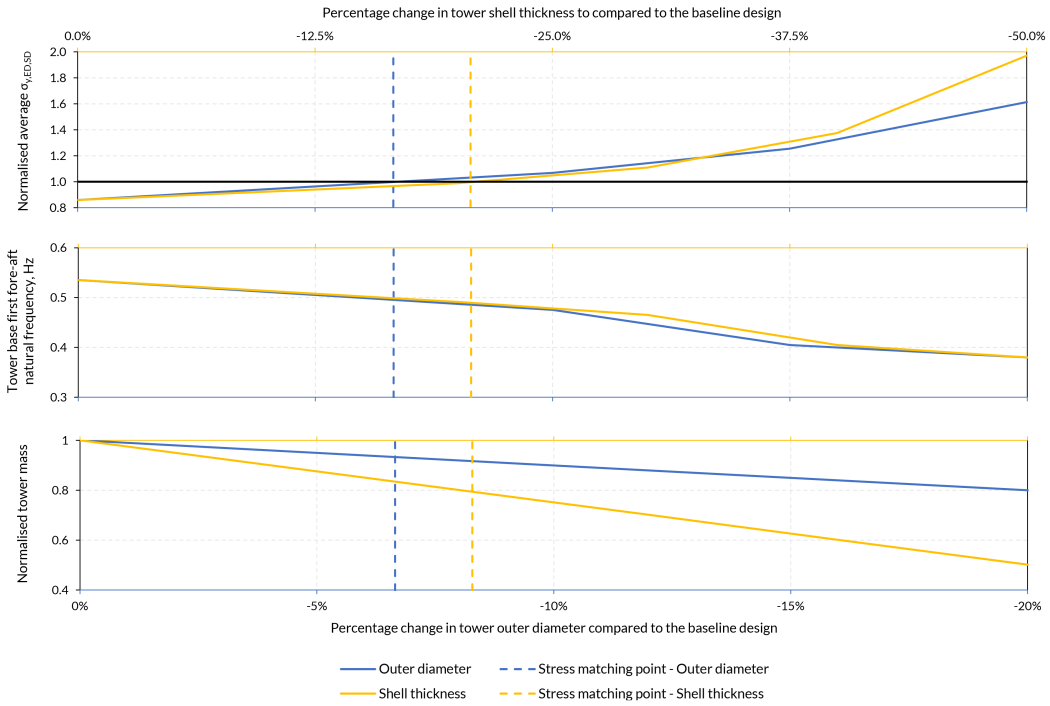


Figure 7.11: Comparing the tower base $\sigma_{y,ED,SD}$ applied to towers with different percentage reductions in the outer diameter (bottom x-axis) and the tower shell thickness (top x-axis) supporting a LIDAR-assisted-controlled FOWT under DLC 1.2 ($\bar{V}$ = 13 ms^{-1}, NTM, TI Category B) with irregular waves (H_s = 2.01 m, T_p = 7.45 s), normalised to the baseline tower supporting an FB-only-controlled FOWT. The dashed vertical lines indicate the $\sigma_{y,ED,SD}$ matching points of the modified towers supporting a LIDAR-assisted-controlled FOWT and the baseline tower supporting an FB-only-controlled FOWT. The middle and bottom sub-figures show the first tower base fore-aft natural frequency and normalised tower mass, respectively.

In comparison to the baseline tower supporting an FB-only-controlled FOWT, the lightweight tower supporting a LIDAR-assisted-controlled FOWT was able to reduce the peaks seen in the PSD plots of the tower base fore-aft and platform pitch resulting from wind loading. The lightweight tower also did not result in significant increases to the PSDs of the tower base side-to-side bending moment, the tower-top displacement or the nacelle acceleration, relative to the baseline tower supporting an FB-only controlled FOWT.

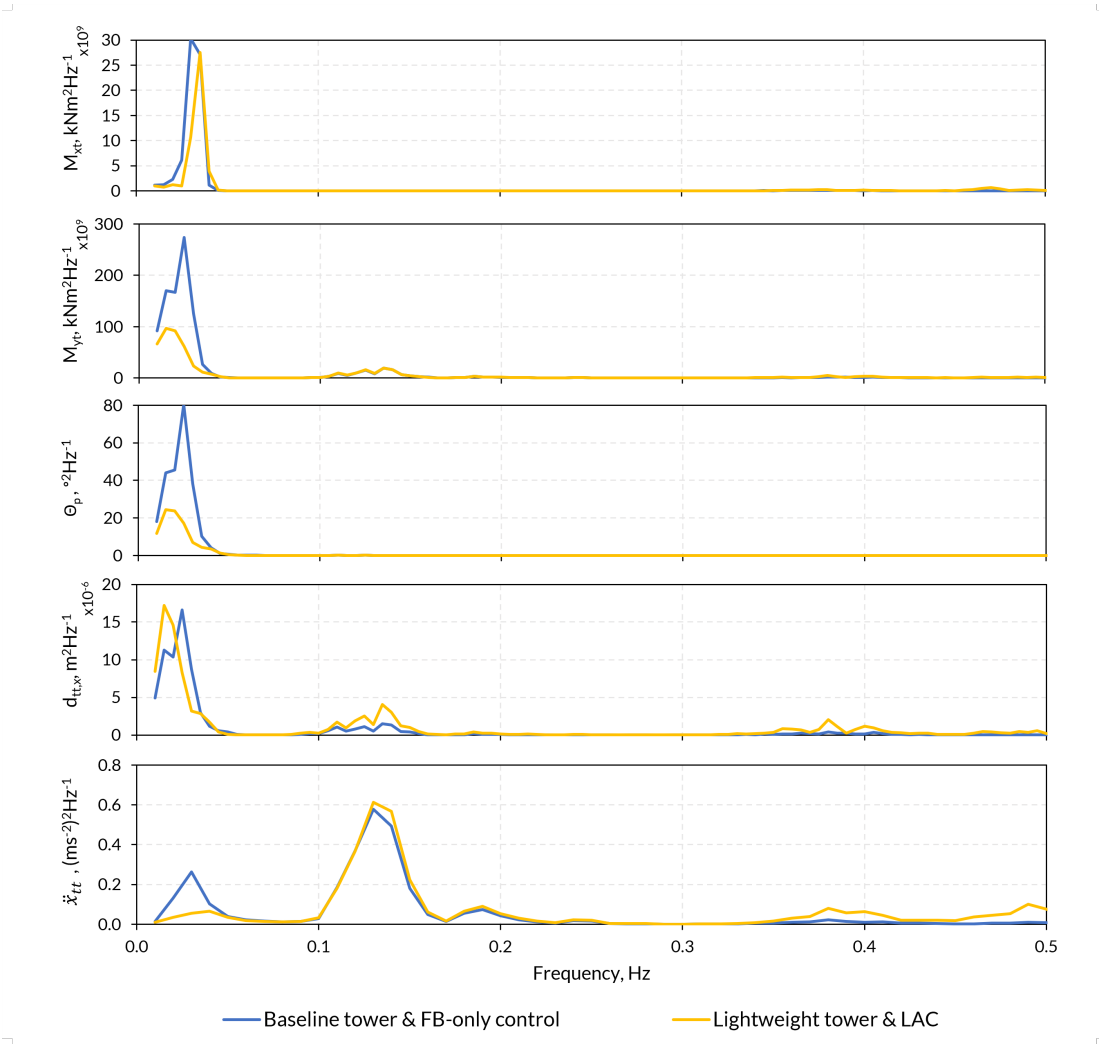


Figure 7.12: PSD plots of five parameters when using the lightweight tower supporting a LIDAR-assisted-controlled FOWT, compared to the baseline tower supporting an FB-only-controlled FOWT. under DLC 1.2 ($\bar{V}$ = 13 ms^{-1}, NTM, TI Category B) with irregular waves (H_s = 2.01 m, T_p = 7.45 s).

7.5 Further analysis - Extreme conditions

Following the initial screening process, the most favourable lightweight tower design (20% reduced tower shell thickness, relative to the baseline tower), was analysed further. Firstly, the further analysis aimed to subject the lightweight and baseline towers to extreme loading conditions via DLCs 1.6 and 6.1. This was because Papi et al. (2023) reported that these DLCs impart the highest fore-aft loads on the tower. It is noted that other DLCs can also lead to extreme loads on the tower, namely in conditions of emergency shutdown, such as DLC 5.1, and extreme operating gusts (DLCs 2.3, 3.2 and 4.2), though these were not considered in this work because DLCs 1.6 and 6.1 were expected to capture the most severe extreme loads. The objective was to determine whether the modified FOWT structure could withstand

the loads and remain within platform motion and nacelle acceleration limits when LAC was active as well as when it was inactive when the turbine was in its parked state in extreme conditions. Secondly, the lightweight tower supporting a LIDAR-assisted-controlled FOWT and the baseline tower supporting an FB-only-controlled FOWT were subjected to wind-wave misalignment scenarios, with results presented in Sect. 7.6.

7.5.1 Responses to extreme conditions

Simulations of DLC 1.6 and DLC 1.6 were undertaken to verify the performance and survivability of the lightweight tower supporting a LIDAR-assisted-controlled FOWT and the baseline tower supporting an FB-only-controlled FOWT. Key parameters to be investigated following these simulations were the tower loads in addition to the FOWT's platform pitch angle, tower-top displacement and nacelle acceleration, to ensure that these remained within acceptable limits. An analysis of the maximum equivalent design stress ($\sigma_{y,ED,max}$) and strength factor was also performed in order to investigate the capability (or lack thereof) of the lightweight tower supporting a LIDAR-assisted-controlled FOWT and the baseline tower supporting an FB-only-controlled FOWT to withstand extreme loadings.

Table 7.3 shows an FB-only-controlled FOWT's platform pitch and nacelle acceleration responses under extreme conditions when supported by the baseline or lightweight tower. For DLC 1.6, the maximum values were found to occur at a mean wind speed of 11 ms^{-1}. The operational limits for FOWTs, outlined by Ramachandran et al. (2013) and within the DNVGL-RP-0286 rules and regulations (see DNV GL, 2021), state that the platform pitch must not exceed a mean of 5° and a maximum of 10° during operation, or a maximum of 15° when parked during storms. Nacelle acceleration must remain below 2.943 ms^{-2} during operation and 5.886 ms^{-2} when parked (as per Ramachandran et al., 2013).

Table 7.3: Mean and maximum platform pitch angle (Θ_p) and maximum tower-top acceleration ($\ddot{x}_{tt,max}$) of an FB-only-controlled FOWT supported by the baseline or lightweight tower, recorded during six ten-minute simulations of DLCs 1.6 and 6.1.

Parameter	Baseline tower	Lightweight tower
$\Theta_{p,mean}$ DLC 1.6, °	4.32	3.13
$\Theta_{p,max}$ DLC 1.6, °	7.01	6.29
$\Theta_{p,max}$ DLC 6.1, °	3.61	3.55
$\ddot{x}_{tt,max}$ DLC 1.6, ms^{-2}	1.37	1.42
$\ddot{x}_{tt,max}$ DLC 6.1, ms^{-2}	1.86	1.93

Both the baseline- and lightweight-tower FOWT designs were below the operational limits. The lightweight design resulted in a reduction in both the mean (by up to 28%) and maximum platform pitch angle (by up to 10%) recorded during operation, relative to the baseline design. The maximum nacelle acceleration was also only marginally increased when the lightweight tower was used.

7.5.2 Strength analysis

To verify that the new FOWT tower design was able to withstand the maximal loads applied to it, a strength analysis in terms of equivalent design stress was performed. Maximum loads on the tower were extracted from the time series results at different tower heights and multiplied by a partial safety factor (1.35 for DLC 1.6, per IEC 61400-3-1:2019, 2019). The maximum equivalent design stresses, $\sigma_{y,ED,max}(z)$, on the cross-sections were computed using Eq. 7.2. Steel class S355, with yielding strength, σ_{yk}, of 355 MPa and tensile strength of 510 MPa was considered. At each section of the tower, the strength requirement was

$$\frac{|\sigma_{y,ED,max}(z)|}{\sigma_{RD}} \leq 1 \tag{7.5}$$

where $\sigma_{RD} = \sigma_{yk}/\gamma_{m0}$ and γ_{m0} is the material safety factor (1, per BS EN 1993-1-1:2005, 2005; IEC 61400-3-1:2019, 2019).

While the fatigue of the welded and bolted joints of the tower is a key design driving aspect, these were not considered as this would have required further analysis of the tower designs within an FEA model, which was outside of the scope of this preliminary case study. Instead, the S355 steel strength was considered to give an indication of the impacts of reducing the geometrical dimensions on the strength of the tower. This was achievable in this case study because the maximum stresses acting on the cross-sections of the tower could be determined from the data output from the WEIS/OpenFAST simulations.

Figure 7.13 presents the results of the strength analysis of the baseline- and lightweight-tower FB-only-controlled FOWT designs for DLC 1.6 with a mean wind speed of 11 ms^{-1}, as the maximum loads in the fore-aft direction in this DLC occurred when the wind speed was close to the rated wind speed.

Figure 7.13 shows that the lightweight tower remained below the strength requirement, indicating that it would not be expected to fail under extreme loads. Figure 7.14 shows the results of the strength analysis of the baseline and lightweight tower designs supporting FB-only controlled and LIDAR-assisted-controlled FOWTs, respectively.

From Fig. 7.14, the addition of LAC was able to reduce the strength fraction of the two towers, relative to when the respective FOWTs used FB-only control, by 7.3% on average. However, the strength factor of the lightweight tower supporting the LIDAR-assisted-controlled FOWT was greater, by 16% on average, than the stress fraction of the baseline tower supporting an FB-only-controlled FOWT. This is because LAC was inactive during the parts of the simulations where the wind speed was below the rated speed and so was unable to impart load reductions in those periods. Additionally, as DLC 1.6 models extreme wave conditions, the hydrodynamic loads dominate over the aerodynamic loads and so LAC was less effective at reducing the loads on the tower under these conditions. Most importantly, although the

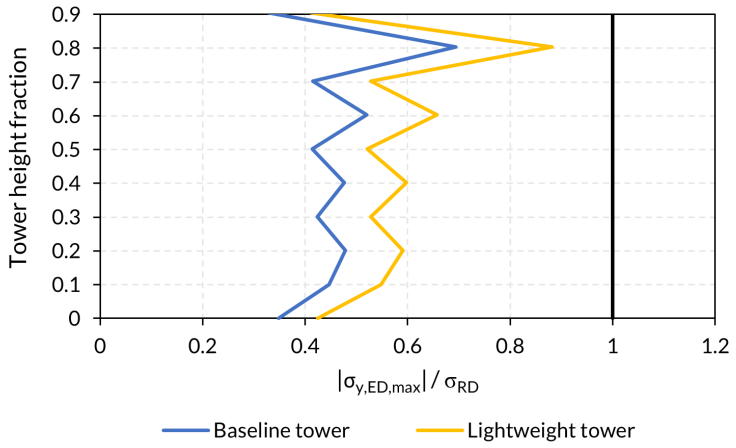


Figure 7.13: Strength analysis of an FB-only-controlled FOWT supported by the baseline or lightweight tower under DLC 1.6 ($\bar{V}$ = 11 ms^{-1}, NTM, TI Category B) with extreme waves (H_s = 8.3 m, T_p = 13.0 s). The vertical black line indicates the strength limit.

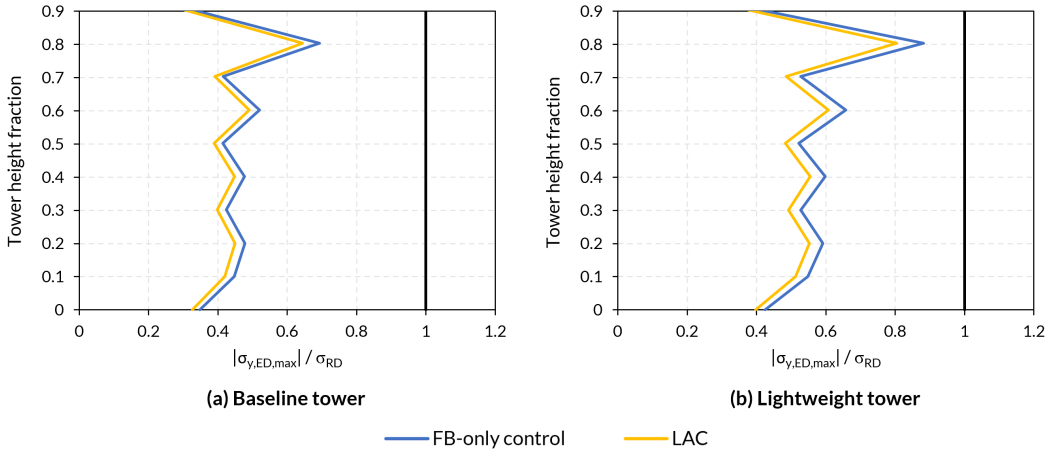


Figure 7.14: Strength analysis of the baseline and lightweight tower designs supporting a LIDAR-assisted-controlled an FB-only controlled FOWT under DLC 1.6 ($\bar{V}$ = 11 ms^{-1}, NTM, TI Category B) with extreme waves (H_s = 8.3 m, T_p = 13.0 s). The black vertical line indicates the tower strength threshold.

strength factor of the lightweight tower supporting the LIDAR-assisted-controlled FOWT was greater than that of the baseline tower supporting an FB-only-controlled FOWT, it remained below the threshold. This indicates that the lightweight tower would not be expected to fail under extreme loads.

7.5.3 Stress envelope

Simulations were performed under DLC 1.6 across operational wind speeds. Turbulent wind fields with mean wind speeds ranging from 5 ms^{-1} to 25 ms^{-1}, in intervals of 2 ms^{-1}, were simulated. For the baseline and lightweight tower, the stress envelope, consisting of the maximum and minimum of all of the maximal equivalent design stresses at each tower height from all of the simulations, was determined. The stress envelopes of the baseline and lightweight tower supporting a LIDAR-assisted-controlled FOWT were compared to those supporting an FB-only controlled FOWT. Figure 7.15 presents the stress envelopes of the two tower designs.

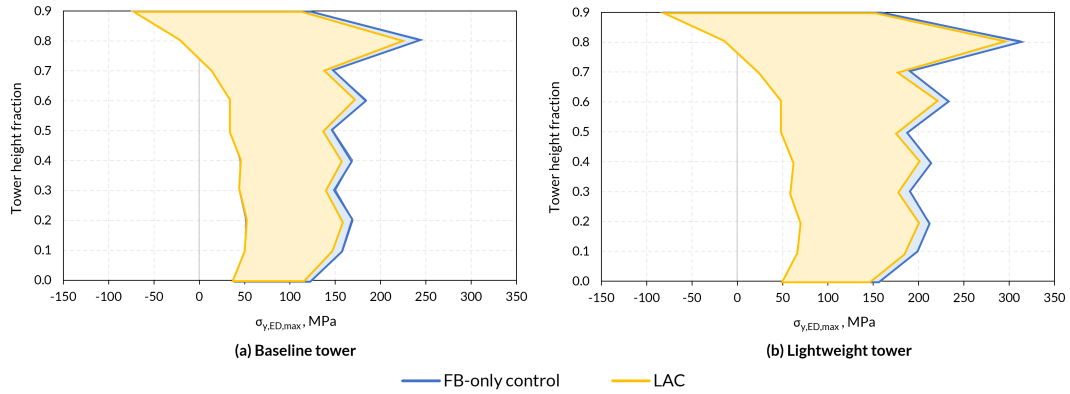


Figure 7.15: Stress envelope of the baseline and lightweight tower designs supporting an FB-only controlled or LIDAR-assisted controlled FOWT under DLC 1.6.

The maximum of the maximal equivalent design stresses occurred at a mean wind speed of 11 ms^{-1}, near the rated wind speed, while the minimum occurred at a mean wind speed of 5 ms^{-1}. The results from the stress envelope support those of the strength analysis presented in Fig. 7.14. Figure 7.15 also indicates that the stress envelope of the lightweight tower supporting the LIDAR-assisted-controlled FOWT was greater than that of the baseline tower supporting an FB-only-controlled FOWT for the reasons discussed in Sect. 7.5.2. However, it did not result in an exceedance of the yield strength of 355 MPa, indicating that the tower would not be expected to fail under the maximum stress that it would be subjected to, even with the inclusion of a conservative partial safety factor (1.35, as per IEC 61400-3-1:2019, 2019). Furthermore, given the ductility and strain-hardening behaviour of S355 steel, failure of the lightweight tower due to yielding is not expected under these loading conditions, justifying the viability of the lightweight tower supporting a LIDAR-assisted controlled turbine.

7.6 Further analysis - Wind-wave misalignment

Although the highest loads applied to offshore wind turbines occur when the wind and wave directions are aligned, the misalignment of the wind and wave directions induces larger side-to-side vibrations of the tower because there is a lack of aerodynamic damping in the tower side-to-side direction. Therefore, when further analysing the lightweight tower design, it was important to capture scenarios of wind-wave misalignment (under DLC 1.2 within this case study). This was because the LAC strategy primarily offered loading reductions in the fore-aft direction of the VolturnUS-S's tower (by up to 21% over the FB-only controller, under co-directional wind and wave conditions), as shown in Fig. 5.7, as opposed to negligible impact upon the side-to-side direction (as shown in Fig. 5.5). Therefore, it was important to capture whether the larger tower side-to-side vibrations resulting from wind-wave misalignment impacted the stress and fatigue of the lightweight tower, despite the use of LAC by the FOWT.

Only first-order wave loads were considered in this case study. This meant that different wave heading directions could be customised within HydroDyn without needing to re-calculate the hydrodynamic coefficients using pyHAMS or WAMIT (as is required when considering second-order wave loads). Wave heading directions between 0° and 180° in 30° increments were simulated. The rotor was assumed to be perfectly aligned (i.e. no yaw misalignment was considered) with the incoming wind direction (fixed at 0°) and so the wind-wave misalignment was equal to the wave heading direction. Equal positive and negative wind-wave misalignments (i.e. +90° and -90°, for example) were assumed to result in identical responses within this case study.

7.6.1 Loading and stress changes with increasing wind speeds

Figure 7.16 presents the impacts of wind-wave misalignment on the standard deviations of the tower base side-to-side and fore-aft bending moments as well as on the $\sigma_{y,ED,SD}$ acting on the base (calculated using Eq. 7.2 for each wind-wave misalignment angle increment) of the baseline tower supporting an FB-only-controlled FOWT at different mean wind speeds. Results are normalised to those at 0° wind-wave misalignment for each mean wind speed.

As expected, greater perpendicularity of the wind and wave direction resulted in increased standard deviations of the tower base side-to-side bending moment (shown in Fig. 7.16) - increasing by up to 74% at a wind-wave misalignment of 120° at a mean wind speed of 21 ms^{-1} (in Fig. 7.16c), relative to those obtained from co-directional (0°) wind and wave simulations. The standard deviations of the tower base fore-aft bending moment of the baseline tower remained relatively unaffected by wind-wave misalignment at low wind speeds, as shown in Fig. 7.16a. However, the standard deviations of the tower base fore-aft bending moment decreased (by up to 27% at 90° misalignment compared to at 0° misalignment at a mean wind speed of 21 ms^{-1}) with increasing misalignment at higher wind speeds (Fig. 7.16c).

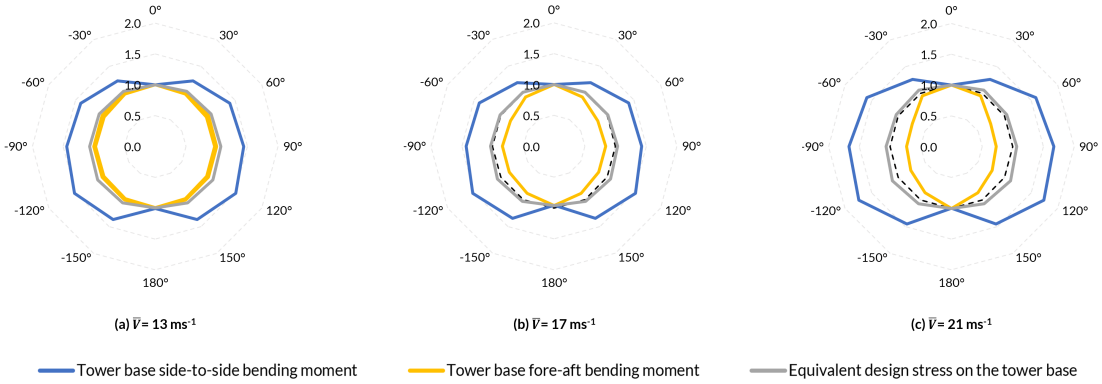


Figure 7.16: Spider-web plots of the six-seed-average standard deviations of the tower base fore-aft and side-to-side bending moments and the $\sigma_{y,ED,SD}$ acting on the base of the baseline tower supporting an FB-only-controlled FOWT. Results are presented at different mean wind speeds modelled with expected wave conditions ($\bar{V}$ = 13 ms^{-1} - H_s = 2.01 m, T_p = 7.45 s. $\bar{V}$ = 17 ms^{-1} - H_s = 2.83 m, T_p = 7.85 s. $\bar{V}$ = 21 ms^{-1} - H_s = 3.82 m, T_p = 8.75 s, as per Allen et al., 2020) and wind-wave misalignment angles. Results are normalised to those at 0° wind-wave misalignment angle for each mean wind speed, indicated by the dashed black line.

This was because the hydrodynamic forces dominated over the aerodynamic forces (which strongly dictate the fore-aft bending moments when the rotor is aligned with the wind, as was always the case in this study) even more so in wind-wave misalignment conditions at high wind speeds. Conversely, the standard deviation of the side-to-side bending moment doubled between mean wind speeds of 13 ms^{-1} and 21 ms^{-1}, as shown in Figs. 7.17a and c. Therefore, at high wind speeds, the tower-side-to-side loads dominated the stress on the tower, as shown in Fig. 7.16c. This occurred because of:

- The greater tower loads in the side-to-side direction in aligned wind and wave conditions and, to an even greater extent, under wind-wave misalignment conditions, at high wind speeds.
- The greater reduction of the tower base fore-aft loads under wind-wave misalignment conditions (as shown in Figs. 7.16c and 7.17c)

Ultimately, because of these factors, when considering the average normalised $\sigma_{y,ED,SD}$ in Fig. 7.16, it is evident that the stress on the base of the baseline tower supporting an FB-only-controlled FOWT increased (by up to 11%, relative to the stress at 0° misalignment) with increasing wind-wave misalignment angles.

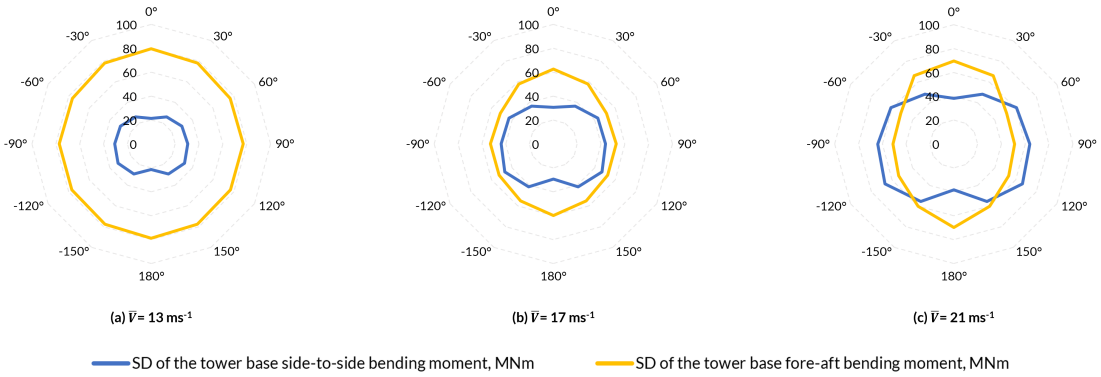


Figure 7.17: Spider-web plots of the six-seed-average standard deviations of the tower base fore-aft and side-to-side bending moments of the baseline tower supporting an FB-only controlled FOWT. Results are presented at different mean wind speeds modelled with expected wave conditions ($\bar{V}$ = 13 ms^{-1} - H_s = 2.01 m, T_p = 7.45 s. $\bar{V}$ = 17 ms^{-1} - H_s = 2.83 m, T_p = 7.85 s. $\bar{V}$ = 21 ms^{-1} - H_s = 3.82 m, T_p = 8.75 s, as per Allen et al., 2020) and wind-wave misalignment angles.

7.6.2 Loads and stress on the lightweight tower

The lightweight tower was selected because its $\sigma_{y,ED,SD}$ approximately matched that of the baseline tower supporting an FB-only-controlled FOWT (as shown in Fig. 7.11) under the maximum wind and wave loads within DLC 1.2 (occurring at $\bar{V}$ = 13 ms^{-1} with co-directional irregular waves of H_s = 2.01 m and T_p = 7.45 s). However, under wind-wave misalignment and at higher wind speeds, it was anticipated that the $\sigma_{y,ED,SD}$ of the lightweight tower supporting a LIDAR-assisted-controlled FOWT would increase to greater than those of the baseline tower supporting an FB-only-controlled FOWT under wind-wave misalignment conditions due to:

- The lesser shell thickness of the lightweight tower, meaning that it was expected to experience larger stress for a given load cycle amplitude.
- The greater variations of the tower side-to-side loads and lesser variations of the tower base fore-aft loads at higher wind speeds and under wind-wave misalignment. This conflicted with the outcomes of the LAC strategy which, for the VolturnUS-S, was ineffective at reducing the tower base side-to-side loads (shown in Fig. 5.5) and less effective (compared to at low above-rated wind speeds) at reducing the variation of the tower base fore-aft bending moment at high wind speeds (shown in Fig. 5.7).

Figure 7.18 compares the impacts of wind-wave misalignment on the standard deviations of the tower base side-to-side and tower base fore-aft and the equivalent design stress on the tower base of the baseline tower supporting an FB-only-controlled FOWT to those of the lightweight tower supporting a LIDAR-assisted-controlled FOWT at different mean wind speeds. Results are normalised to those of the baseline tower supporting an FB-only-controlled FOWT with 0° wind-wave misalignment for each mean wind speed.



Figure 7.18: Spider-web plots of the six-seed-average standard deviations of the tower base fore-aft and side-to-side bending moments and the equivalent design stress acting on the base of the lightweight tower supporting a LIDAR-assisted-controlled FOWT and the baseline tower supporting an FB-only-controlled VolturnUS-S. Results are presented at different mean wind speeds modelled with expected wave conditions ($\bar{V}$ = 13 ms^{-1} - H_s = 2.01 m and T_p = 7.45 s, $\bar{V}$ = 17 ms^{-1} - H_s = 2.83 m and T_p = 7.85 s and $\bar{V}$ = 21 ms^{-1} - H_s = 3.82 m and T_p = 8.75 s, as per Allen et al., 2020) and wind-wave misalignment angles. Results are normalised to the baseline tower supporting an FB-only-controlled FOWT at a 0° wind-wave misalignment angle, indicated by the dashed black line.

The results shown in Fig. 7.18c indicate that the lightweight tower supporting a LIDAR-assisted-controlled FOWT experienced lesser average normalised $\sigma_{y,ED,SD}$ at all wind-wave misalignments, relative to the baseline tower supporting an FB-only-controlled FOWT, under the same wind-wave misalignment conditions, by up to 9%, at a mean wind speed of 13 ms^{-1}. This formed a greater disparity between the average normalised $\sigma_{y,ED,SD}$ on the base of the two

tower designs than at 0° misalignment, where the difference was 1%. This was because the standard deviation of the tower base fore-aft bending moment at this wind speed (13 ms^{-1}, Fig. 7.18b) of the lightweight tower supporting a LIDAR-assisted-controlled FOWT decreased, by up to 13% (compared to at 0° misalignment), with increasing wind-wave misalignment. As the standard deviations of the tower base fore-aft bending moment were much greater than those of the side-to-side bending moment (by up to 274%), at a mean wind speed of 13 ms^{-1}, as shown in Fig. 7.17, these dominated in the calculation of $\sigma_{y,ED,SD}$ (Eq. 7.2) at low wind speeds. Therefore, reductions to the standard deviation of the tower base fore-aft bending moment by LAC under wind-wave misalignment at low wind speeds (by up to 34%, relative to the baseline tower supporting an FB-only-controlled FOWT at 0° wind-wave misalignment, as shown in Fig. 7.18b), were able to offset the increases seen in the standard deviation of the tower base side-to-side bending moment (by up to 40% at 90° misalignment in Fig. 7.18a) as the wind-wave misalignment angle increased.

However, as the wind speed increased, the standard deviation of the tower base side-to-side bending moment increased to greater than that of the tower base fore-aft bending moment (which instead decreased with increasing wind speed) at high wind wave-misalignment angles, as shown in Fig. 7.17. Meanwhile, the LAC strategy was less able to reduce the standard deviations of the tower base fore-after bending moment of the lightweight tower supporting a LIDAR-assisted controlled turbine, relative to the baseline tower supporting an FB-only controlled FOWT, as the wind speeds increased. The reduction to the standard deviation of the tower base fore-after bending moment diminished by 14% at 0° misalignment between wind speeds of 13 ms^{-1} (Fig. 7.18b) and 21 ms^{-1} (Fig. 7.18i). Therefore, despite the LAC strategy reducing the variations of the loads on the lightweight tower design at all wind speeds and wind-wave misalignments, relative to the baseline tower supporting an FB-only-controlled FOWT (as shown in Figs. 7.18a, b, d, e, h and i), when accounting for the lesser thickness of the lightweight tower using Eq.7.2, the addition of LAC to the FOWT supported by the lightweight tower was unable to offset the increased stress that this brought as wind speeds increased (as shown in Figs. 7.18c, f and j). At a mean wind speed of 21 ms^{-1} (Fig. 7.18c), the $\sigma_{y,ED,SD}$ on the tower base of the lightweight tower supporting a LIDAR-assisted controlled FOWT was up to 28% greater than that of the baseline tower supporting an FB-only-controlled FOWT.

7.7 Further analysis - Joint probability density function

The results presented in Sect. 7.6.2 indicate that the tower base of the lightweight tower supporting a LIDAR-assisted-controlled FOWT experienced reduced $\sigma_{y,ED,SD}$, relative to those of the baseline tower supporting an FB-only-controlled FOWT, at low above-rated wind speeds (Fig. 7.18a). However, the $\sigma_{y,ED,SD}$ on the tower base of the lightweight tower supporting a LIDAR-assisted-controlled FOWT was increased, relative to that of the baseline tower supporting an FB-only-controlled FOWT, at high above-rated wind speeds (Fig. 7.18i). The LAC strategy was also even less effective at mitigating the increased stress on the tower base at high wind speeds and under wind-wave misalignment due to the increased tower side-to-side loads occurring under those conditions.

Despite these detrimental impacts, occurrences of high wind speeds and high wind-wave alignment are infrequent compared to low above-rated wind speeds and co-directional wind and waves, where the LAC strategy was most effective at reducing the stress on the base of the tower. Therefore, it was pertinent to consider a wind speed/wind-wave misalignment angle joint probability density (JPD) function to provide a more realistic estimation of the resulting stress on the base of the lightweight tower supporting a LIDAR-assisted-controlled FOWT compared to the baseline tower supporting an FB-only-controlled FOWT.

7.7.1 Probability density functions

Within this analysis, a two-dimensional JPD function was determined based on the combination of a wind speed probability density function (PDF) and a wind-wave misalignment PDF, as was demonstrated by Stewart and Lackner (2014). Subsequently, the $\sigma_{y,ED,SD}$ on the tower base at different wind speeds and wind-wave misalignment angles were weighted according to the respective probabilities of mean wind speed/wind-wave misalignment angle bin combinations.

The wind speed probability density was determined from the Weibull parameters of the NE3 ScotWind wind farm site (shown in Table 6.2), obtained from met-ocean data sourced from LAUTEC ESOX (2024). NE3 was selected as the site within this study because it showed the highest potential (over the two other sites that were investigated) for OpEX reductions in Chapter 6 (when WI limits were not included), and so it was deemed relevant to investigate the suitability for CAPEX reductions through LAC-enabled simplification of FOWT towers. The Weibull wind speed PDF of NE3 is shown in Fig. 7.19a. Subsequently, the probabilities of each 2 ms^{-1}-width wind speed bin (twelve in total) were determined from the cumulative distribution function (Φ) of the Weibull distribution, with the simulated mean wind speeds at the centre of each bin

$$P_{\bar{V}}(\bar{V}_L < \bar{V} < \bar{V}_H) = \Phi_{\bar{V}}(\bar{V}_H; C, K) - \Phi_{\bar{V}}(\bar{V}_L; C, K) \tag{7.6}$$

where $P_{\bar{V}}$ is the probability of a given mean wind speed bin, $\bar{V}_L$ and $\bar{V}_H$ are the lower and upper limits of the wind speed bin, respectively, and $\Phi_{\bar{V}}$ is the cumulative distribution of the wind speed.

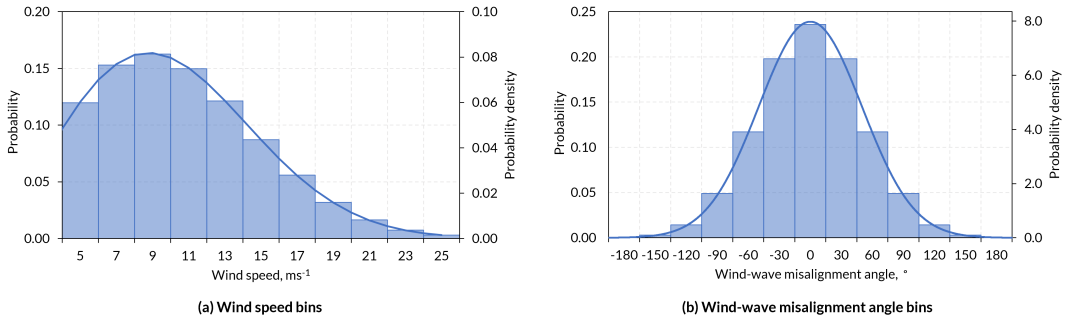


Figure 7.19: Weibull wind speed probability density function of the ScotWind NE3 wind farm site (from LAUTEC ESOX, 2024), and a normal distribution curve representing a probability density function of wind-wave misalignment. The bars (corresponding to the left-hand side y-axis) indicate the probabilities of the simulated mean wind speed and wind-wave misalignment bins.

The wind-wave misalignment probability density curve used in this case study is shown in Fig. 7.19b. Typically, wind-wave misalignment probabilities are determined from gathered met-ocean data. Stewart and Lackner (2014) and Barj et al. (2014) used a Von Mises distribution to fit wind-wave misalignment values to mean wind speed bins. However, due to time constraints, met-ocean wind-wave misalignment data of the NE3 site could not be sourced to determine the actual wind-wave misalignment probability density in this case study.

As an alternative, a normal distribution function (NDF) was assumed to represent the probability density of the wind-wave misalignment. Twelve wind-wave misalignment angle bins of 30° width, spanning from -180° to 150° were defined. The Φ of the normal distribution curve was used to determine the probability of occurrence of each wind-wave misalignment angle (η) bin.

$$P_\eta(\eta_L < \eta < \eta_H) = \Phi_\eta(\eta_H; \mu, SD) - \Phi_\eta(\eta_L; \mu, SD) \tag{7.7}$$

where P_η is the probability of a given wind-wave misalignment angle (η) bin, η_L and η_H are the lower and upper limits of the wind-wave misalignment angle bin, respectively, and Φ_η is the cumulative distribution of the wind-wave misalignment according to the NDF's standard deviation and mean (μ). The probabilities of the wind-wave misalignment bins were assumed to be equal for all values of $\bar{V}$, H_s and T_p.

The probability of a wind speed/wind-wave misalignment bin combination was determined by multiplication of the probability of a wind speed bin with that of a wind-wave misalignment bin

$$P(\eta_L < \eta < \eta_H)|(\bar{V}_L < \bar{V} < \{\bar{V}_H) = P_\eta(\eta_L < \eta < \eta_H) \times P_{\bar{V}}(\bar{V}_L < \bar{V} < \bar{V}_H) \tag{7.8}$$

This resulted in a JPD surface plot, shown in Fig. 7.20.

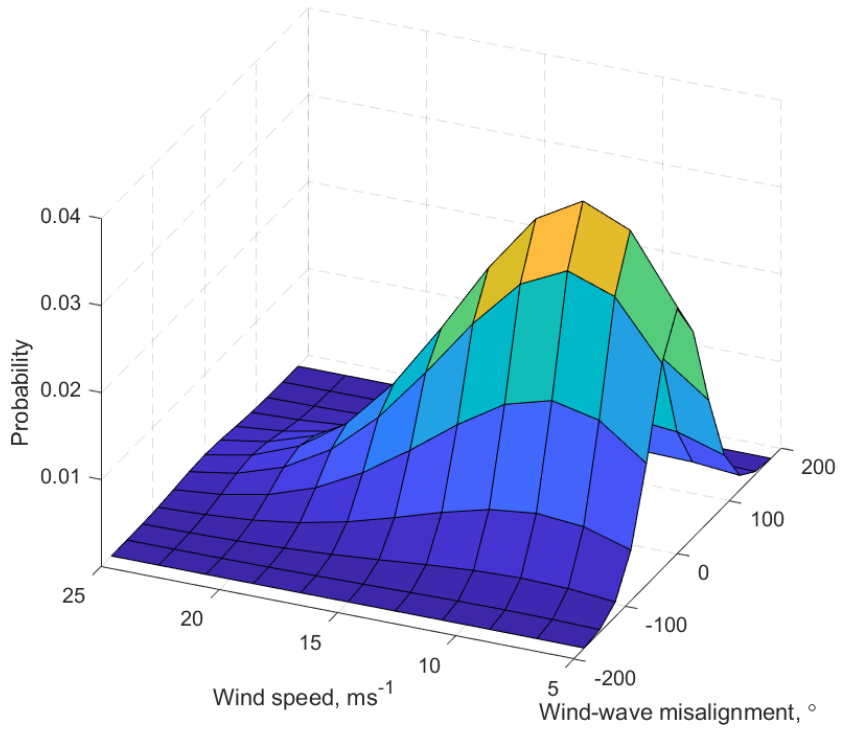


Figure 7.20: JPD function of mean wind speed and wind-wave misalignment angle.

The standard deviation of the normal distribution (NDF_{SD}) used to give a wind-wave misalignment distribution was assumed to be 50, such that the surface plot in Fig. 7.20 was similar (in shape in the x-z plane) to that presented by Stewart and Lackner (2014). It is appreciated that this method is likely to have resulted in an unrealistic probability density function of the wind-wave misalignment and, subsequently, a JPD function that is not wholly representative of the NE3 site. Therefore, it is recommended that, in future work, an accurately determined (based on gathered met-ocean data) JPD is applied to the preceding results. This should use a Von Mises distribution of the data, dependent on wind speed (as in Stewart & Lackner, 2014), to determine site-specific wind-wave misalignment bin probabilities.

Nonetheless, and as this was only intended as a preliminary case study, this exercise aided in estimating the weighted (according to the wind speed/wind-wave misalignment probabilities) $\sigma_{y,ED,SD}$ on the tower base of the lightweight tower supporting a LIDAR-assisted-controlled FOWT and how it compared to the weighted stress applied to the baseline tower supporting an FB-only-controlled FOWT.

In order to account for inaccuracies that may have resulted from the method used to determine the wind-wave misalignment probability density function, sensitivity analysis was performed on the NDF_{SD} within Sect. 7.7.2. This allowed for the comparison of the results with those using greater values of NDF_{SD}, which provided more conservative (in this context, as higher probabilities of greater wind-wave misalignment detracted from the stress reduction benefit brought by LAC of the lightweight tower, particularly at high above-rated wind speeds) estimates of the wind-wave misalignment probability density function.

7.7.2 Results

The six-seed-average $\sigma_{y,ED,SD}$ of each wind speed/wind-wave wave misalignment bin combination for the two FOWT tower designs were weighted according to their respective probabilities of occurrence within the JPD. The total $\sigma_{y,ED,SD}$ across all of the above-rated mean wind speed and wind-wave misalignment angle bin combinations of the lightweight tower supporting a LIDAR-assisted-controlled FOWT (8.36 MPa) was found to be 9.1% less than that of the baseline tower supporting an FB-only-controlled FOWT (9.20 MPa). This indicates that the reductions in the stress delivered by LAC at low wind speeds and low wind-wave misalignment angles outweighed the increased stress at high wind speeds and high wind-wind wave misalignments due to the significantly greater occurrence of the more favourable (for LAC) conditions.

The effect of the JPD weighting on the stress at different wind speeds and wind-wave misalignment angles is shown in Fig. 7.21. The $\sigma_{y,ED,SD}$ at each mean wind speed bin in Fig. 7.21a is the total weighted $\sigma_{y,ED,SD}$ of all wind wave-misalignments of that mean wind speed bin. Conversely, the $\sigma_{y,ED,SD}$ at each wind-wave misalignment bin in Fig. 7.21b is the total weighted $\sigma_{y,ED,SD}$ of all wind speeds at that wind-wave misalignment angle bin.

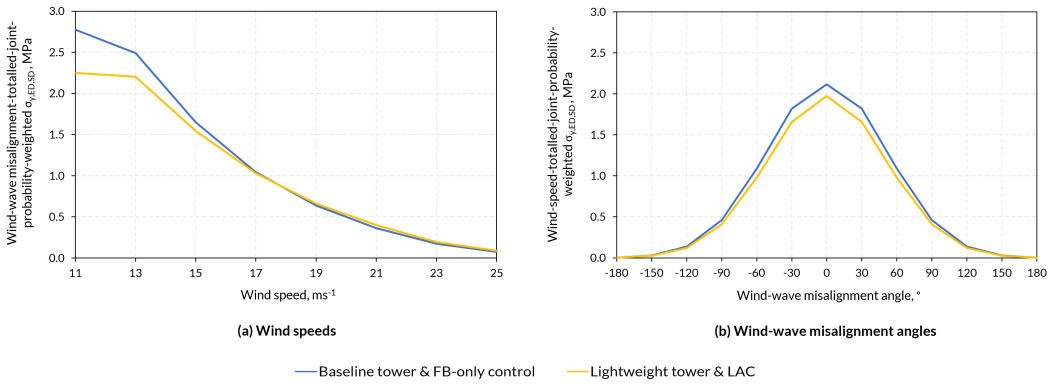


Figure 7.21: Total JPD-weighted six-seed-average $\sigma_{y,ED,SD}$ of each wind speed (across all wind-wave misalignment angles) and wind-wave misalignment angle (across all wind speeds) bin.

From Fig. 7.21a, it is clear that the JPD-weighted $\sigma_{y,ED,SD}$ of the lightweight tower supporting a LIDAR-assisted-controlled FOWT was significantly less than (by up to 19%) that of the baseline tower supporting an FB-only-controlled FOWT, at low above-rated wind speeds. This is because the stress reductions brought by LAC (primarily in the fore-aft direction) were greatest at low above-rated wind speeds, even at high wind-wave misalignment angles, as was shown in Fig. 7.18b and c. The lesser stress on the lightweight tower supporting a LIDAR-assisted-controlled FOWT relative to the baseline tower supporting an FB-only-controlled FOWT diminished as the wind speed increased, with the JPD-weighted stress reaching parity at circa 18 ms^{-1}. The JPD-weighted stress on the lightweight tower supporting a LIDAR-assisted-controlled FOWT was up to 9.8% greater than that on the tower of the baseline tower supporting an FB-only-controlled FOWT in wind speeds greater than 18 ms^{-1}. However, the low probability of these high wind speeds, combined with the low probabilities of the larger-stress-inducing high wind-wave misalignment angles, meant that the increase in the total weighted $\sigma_{y,ED,SD}$ at wind speeds greater than 18 ms^{-1} was only 0.091 MPa, which was far outweighed by the up to 0.921 MPa reductions to the total weighted $\sigma_{y,ED,SD}$ at low above-rated winds speeds.

Figure 7.21b indicates that total weighted $\sigma_{y,ED,SD}$ of the lightweight tower supporting a LIDAR-assisted-controlled FOWT was reduced (relative to the baseline tower supporting an FB-only-controlled FOWT) most significantly (by up to 7.3%) in co-directional wind and wave conditions. This was unsurprising given that the LAC benefit was greater and the tower stress was lesser in co-directional wind-wave conditions, as shown in Figs.7.17. As the wind-wave misalignment angles increased, differences in the total weighted $\sigma_{y,ED,SD}$ between the two designs diminished. The total weighted $\sigma_{y,ED,SD}$ of the lightweight tower supporting a LIDAR-assisted-controlled FOWT did not increase to greater than that of the baseline tower supporting an FB-only-controlled FOWT at any wind-wave misalignment angle. This was because the greater stress reductions that were delivered by LAC at the high-occurrence low above-rated wind speeds were not significantly affected by high wind-wave misalignment angles, as shown in Figs.7.18c. Therefore, these stress reductions were imparted on all of the wind-wave misalignment bins within Fig.7.21b.

Sensitivity analysis

The standard deviation of the NDF used to determine the probabilities of the wind-wave misalignment angles bin was assumed to be 50, based solely on the shape (in the x-z) plane of the JPD plot presented by Stewart and Lackner (2014). Given that this arbitrarily defined NDF was not determined based on met-ocean data, specifically (for this case study), the NE3 ScotWind site, it was deemed pertinent to capture the impact that increasing NDF_{SD} had upon the results. Increasing the standard deviation of a NDF results in a broader (but lower in peak probability density) bell-curve of a NDF. In the context of this case study,

increasing the NDF_{SD} raises the probability of occurrence of high wind-wave misalignment angles, while decreasing the probability of occurrence of low wind-wave misalignment angles and co-directional wind and waves. Therefore, high values of NDF_{SD} may reduce the stress-alleviating benefits of LAC, which is ineffective at reducing tower side-to-side loads, ultimately exposing the lightweight tower to greater stress and fatigue.

This sensitivity analysis will compare the aforementioned results (at NDF_{SD} = 50) to those obtained using higher values of NDF_{SD} (70 and 90). Lower values of NDF_{SD} were not simulated because this would have resulted in higher probabilities of low wind-wave misalignment angles and co-directional wind and waves. This would have resulted in greater LAC-enabled stress-reductions than those presented in Sect. 7.7.2. However, this sensitivity analysis intended to focus on conditions that would render the LAC less able to reduce the stress on the lightweight FOWT tower relative to that of the baseline-tower/FB-only-control FOWT. Figure 7.21 (which used NDF_{SD} = 50 to determine the wind-wave misalignment angle bin probabilities) with additional plots at NDF_{SD} = 70 and NDF_{SD} = 90 is shown in Fig. 7.22.

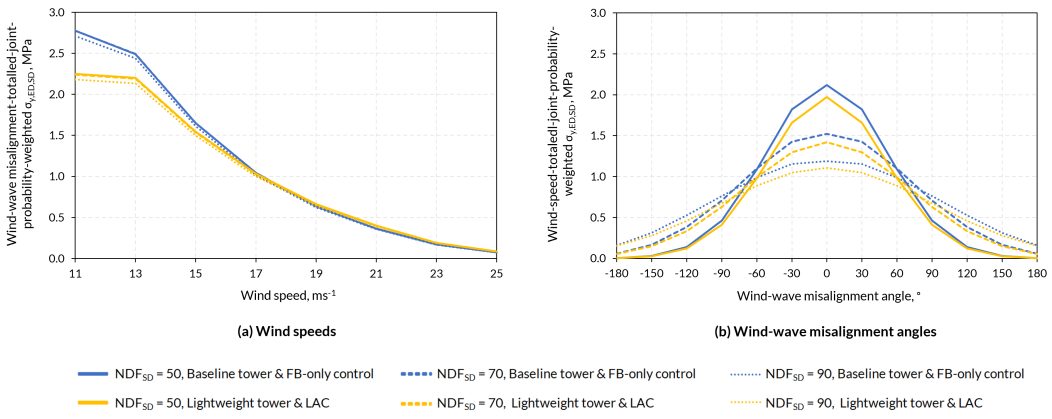


Figure 7.22: Total JPD-weighted six-seed-average $\sigma_{y,ED,SD}$ of the baseline tower supporting an FB-only controlled FOWT and the lightweight tower supporting an FB-only controlled FOWT at each wind speed (across all wind-wave misalignment angles) and wind-wave misalignment angle (across all wind speeds) bin achieved when using different values of NDF_{SD}.

As expected, Fig. 7.22b indicates that the bell-curve of the wind-wave misalignment angle probability broadened with increasing values of NDF_{SD}. This caused a decrease in $\sigma_{y,ED,SD}$ for both FOWT designs, by up to 44% (0.929 MPa), in co-directional wind and wave conditions, and an increase, by to up 181% (0.389 MPa) at high wind-wave misalignment angles, compared to when an NDF_{SD} was used. However, this did not significantly affect the changes in the total weighted $\sigma_{y,ED,SD}$ with wind speed (Fig. 7.22a). Furthermore, the reduction of the total weighted $\sigma_{y,ED,SD}$ of the lightweight tower supporting a LIDAR-assisted-controlled FOWT over the baseline tower supporting an FB-only-controlled FOWT remained consistent

across all of the NDF_{SD} values that were studied. The total $\sigma_{y,ED,SD}$ of the lightweight tower supporting a LIDAR-assisted-controlled FOWT remained consistently (approximately 10%) below that of the baseline tower supporting an FB-only-controlled FOWT across all of the above-rated mean wind speed and wind-wave misalignment angle bin combinations. This leads to the conclusion that the tower stress reduction delivered by LAC is not significantly impacted by increasing variations in wind-wave misalignment angles due to the dominance of the LAC benefits in the higher probability low above-rated wind speed and low/no wind-wave misalignment angle conditions.

Chapter 8

Array-level benefits of LIDAR-assisted pitch control

This Chapter investigates the impacts of LAC on downstream turbines within an array by simulating multiple iterations of the VolturnUS-S. This work assesses whether the loading reductions and superior rotor speed regulation of FOWTs brought by LAC can allow for additional benefits for downstream turbines via improved wake consistency and recovery of the upstream turbines.

8.1 Simulation set-up

Simulations were performed using NREL's FAST.Farm wind farm modelling tool. FAST.Farm allows users to perform simulations of turbine arrays, using separate instances of OpenFAST to solve the aero-elastic-servo-hydro dynamics of each individual turbine. It also considers additional physics for wind farm-wide ambient wind in the atmospheric boundary layer, a wind-farm supercontroller and wake behaviour (deficits, advection, deflection, meandering and merging). FAST.Farm is based on some of the principles of the dynamic wake meandering (DWM) model from G. C. Larsen et al. (2007). The DWM model accounts for temporal and spatial variations in the wake, capturing changes in the wake over time and distance. FAST.Farm considers a high-resolution grid in the immediate vicinity of the wind turbines and a low-resolution grid in the surrounding areas in order to manage the computational cost. Further details on FAST.Farm's operation can be found in Appendix A.5 and within its user guide and theory manual, from J. Jonkman and Shaler (2021).

This study considered three VolturnUS-S FOWTs (T1, T2 and T3), which were aligned with each other as well as the wind and wave directions. The turbines were spaced equidistant from each other, with separations in multiples of the turbine's rotor diameter (RD - 240 m), as shown in Fig. 8.1. It is typically recommended for turbines to be placed between 8 and 12 RD apart (see Patel & Beik, 2021), to minimise wake impacts.



Figure 8.1: Layout of the three FOWTs, separated by multiples (X) of the turbine's rotor diameter (RD), as simulated within FAST.Farm.

This study performed simulations under DLC 1.2, with above-rated mean wind speeds ranging from 11 ms^{-1} to 25 ms^{-1}, in intervals of 2 ms^{-1}, and expected values of H_s and T_p applied for each mean wind speed, according to the values stated in Table 6.4. Six random seed simulations were performed at each mean wind speed, with ten minutes of data extracted for each turbine. Ten-minute data excerpts were extracted for each turbine instead of one-hour excerpts due to the significantly greater computation time that one-hour-length FAST.Farm simulations incurred. The first 100 seconds of simulation data were removed for the three FOWTs in order to account for start-up transients. 100 seconds allowance for start-up transients was found to be sufficient when undertaking this study. Data for the downstream turbines was time-shifted to account for the transport time of the wind between the turbines, ensuring comparable wind profiles between the three FOWTs. This is further explained in Appendix F, where an accompanying example of the excerpt start times of each turbine and the total simulation times for each mean wind speed for the 8 RD separation arrangement is provided in Table F.1.

This case study first investigated the differences in the performance of the three FOWTs when using FB-only control, identifying areas in which LAC was able to provide benefit. Simulations were performed with the FOWTs spaced at various RDs to ascertain the ideal spacing to minimise power capture reductions of the downstream turbines. Following this assessment, the FOWTs were simulated with LAC enabled. This sought to determine how the addition of LAC could aid in mitigating the detrimental impacts on the downstream turbines via their use of LAC and also via the benefits to the wake profile of the upstream turbine brought by its use of LAC.

8.2 Methodologies

The methodologies followed in performing the simulations and analysing the data within this study are outlined in Fig. 8.2. This required appropriately configuring features of the simulations within the FAST.Farm master input file (shown in Figs. D.14 and D.15) such as the high- and low-resolution grids and the turbine origin positions according to the turbine

spacings for each RD separation and specifying the outputs required for analysis. FAST.Farm was configured to output separate OpenFAST output files for each FOWT in addition to its own output file, which contained information about the wake behaviour, reflecting the output parameters specified within the FAST.Farm master input file.

The time series dataset outputs (of individual turbines and of the wind farm) obtained from the FAST.Farm simulations of the FOWTs at 8 RD separation can be found via a link to a data repository provided in Appendix C.

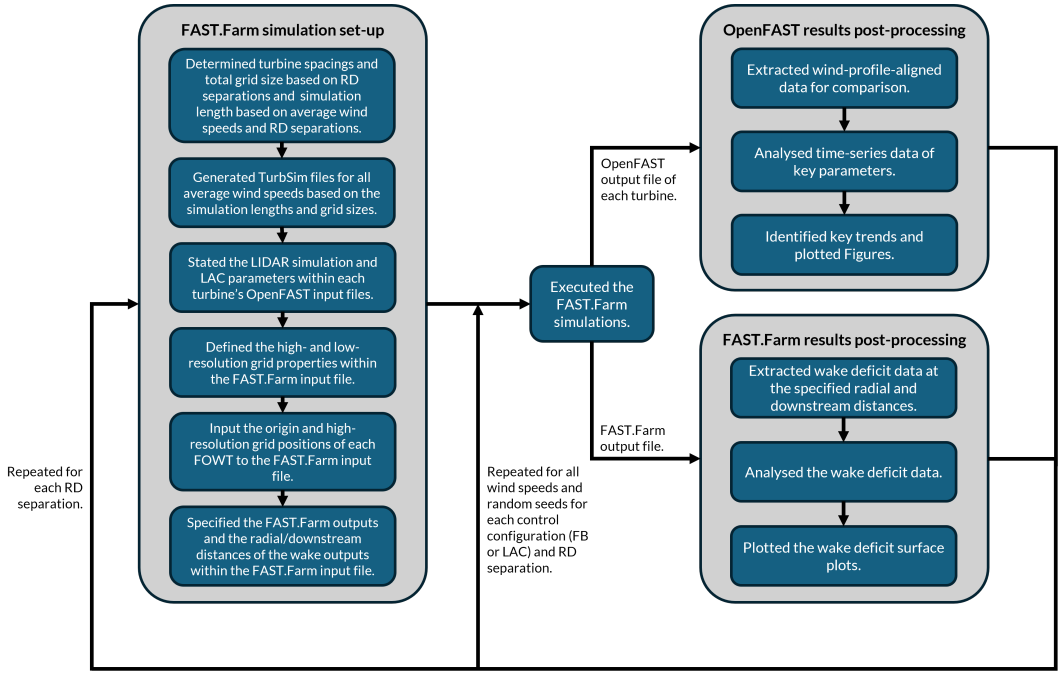


Figure 8.2: Flowchart illustrating the steps undertaken to perform the array-level LAC study.

8.3 Results

This Section will outline the results of the FAST.Farm simulations. Firstly, the results of the turbines using FB-only control will be examined, to determine the impacts of varying spacing on downstream turbines and the differences between the responses of the FOWTs. This will be followed by a comparison with the turbines implementing LAC, to ascertain any apparent benefits that emerge.

8.3.1 Feedback-only control

Table 8.1 documents the mean power ($\bar{P}_{el}$) of the three turbines (all using FB-only control), normalised to the mean power of turbine 1 (T1), at different turbine spacings. The results are from data recorded during simulations at a mean ambient wind speed of 11 ms^{-1}. a mean wind speed of 11 ms^{-1} was used because it is close to the rated speed of the turbine (10.69 ms^{-1}) and so there was a strong possibility that the downstream turbines experienced a below-rated wind speed should insufficient spacing have been afforded between the turbines, thereby reducing their power capture.

Table 8.1: Six-seed-average mean power of each turbine, normalised to that of T1, for different turbine spacings under turbulent wind ($\bar{V}$ = 11 ms^{-1}, NTM, TI Category B) and irregular wave (H_s = 2.39 m, T_p = 7.85 s) conditions.

Spacing, RD	Norm. (to T1) $\bar{P}_{el}$ of T1	Norm. $\bar{P}_{el}$ (to T1) of T2	Norm. (to T1) $\bar{P}_{el}$ of T3
8	1.000	0.974	0.978
6	1.000	0.841	0.824
4	1.000	0.682	0.596

From Table 8.1, reducing the spacing between the turbines resulted in significantly reduced mean power capture of the downstream turbines at a mean ambient wind speed of 11 ms^{-1}, relative to T1. When turbines were placed at either 6 or 4 RD separation, insufficient space was afforded to allow for the wind to return to free stream. This resulted in an over 40% reduction in the mean power capture of T3 compared to T1 when the turbines were spaced at 4 RD. Similar findings were reported by Choi et al. (2014). Therefore, the remainder of this Chapter will focus on the 8 RD spacing, as this led to the lowest reduction in the mean power of the downstream turbines. It is also noteworthy that the spacing of turbines is a balance between power capture and wind farm power density, as more turbines can be placed within the same area if there is less distance between them. Therefore, 8 RD was seen as the ideal separation distance to balance these factors and so layouts of 10 and 12 RD separations were not investigated in this study.

Although the mean power of the downstream turbines was reduced (relative to T1), the mean values of some turbine and substructure parameters were found to increase compared to T1. Figure 8.3 shows the mean values of the turbine and substructure parameters most affected, with data normalised to the mean value of T1, across the above-rated wind spectrum.

The trends seen in the mean values of the thrust (Fig. 8.3a), the platform pitch (Fig. 8.3c), the blade root bending moment (Fig. 8.3e) and the tower base fore-aft bending moment (Fig. 8.3f) were similar for both T2 and T3, relative to T1. The results showed increased mean values of these parameters, by up to 11%, at a mean ambient wind speed of 15 ms^{-1}, compared to T1, with increases slightly greater (by up to 1.4%) for T2 than for T3. These increased average loads can result in greater fatigue and damage to the downstream turbines, making them more

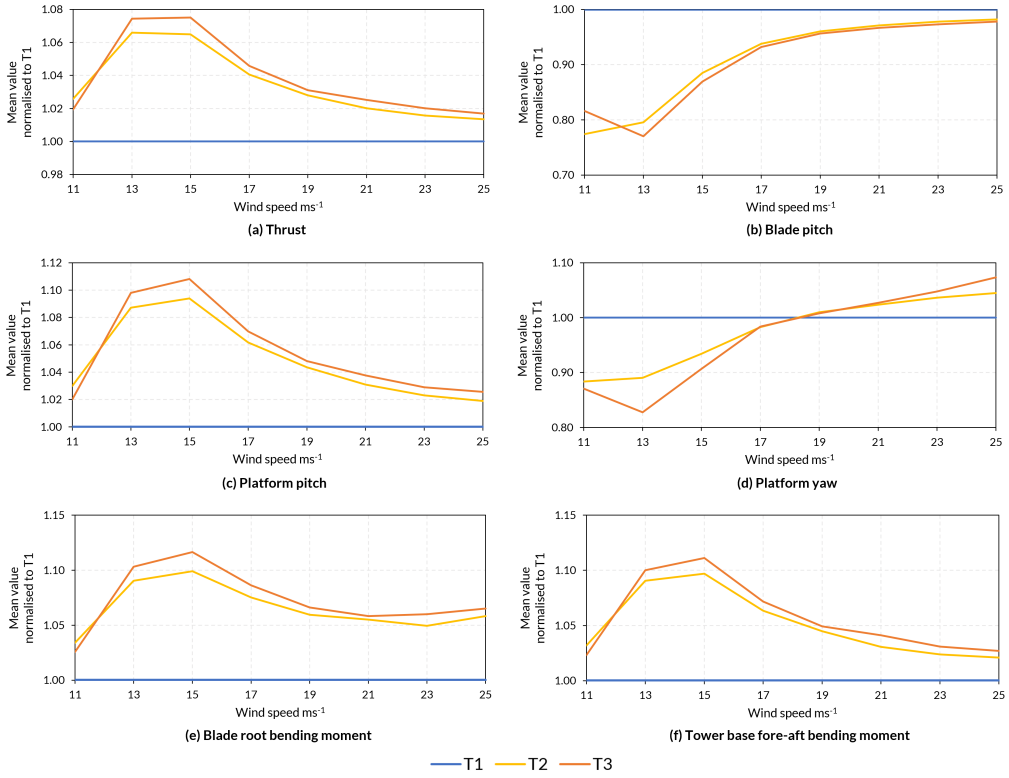


Figure 8.3: Six-seed-average mean values of turbine parameters, normalised to that of T1, when using FB-only control across above-rated wind speeds and with a turbine spacing of 8 RD.

prone to failures requiring maintenance. The means of the platform yaw of the downstream turbines (Fig. 8.3d) were reduced (relative to T1) at lower above-rated wind speeds (by up to 17%) but increased to greater than T1 (by up to 7%) at mean wind speeds greater than 17 ms^{-1}. The blade pitch angle (Fig. 8.3b) of the downstream turbines was also reduced relative to T1 at all wind speeds, by up to 23%, which was expected due to the reduced wind speeds they were subjected to requiring lower pitch angles. The reduction in the blade pitch angle compared to T1 was much less significant (reducing to 2%) at higher wind speeds because T1 extracted a lesser proportion of the kinetic energy in the wind, thereby resulting in a relatively lower wake deficit. This is shown in Fig. 8.4, where the mean wake deficit 1000 m downstream of T1 decreased by approximately 35% as the mean ambient wind speed increased from 11 ms^{-1} to 25 ms^{-1}.

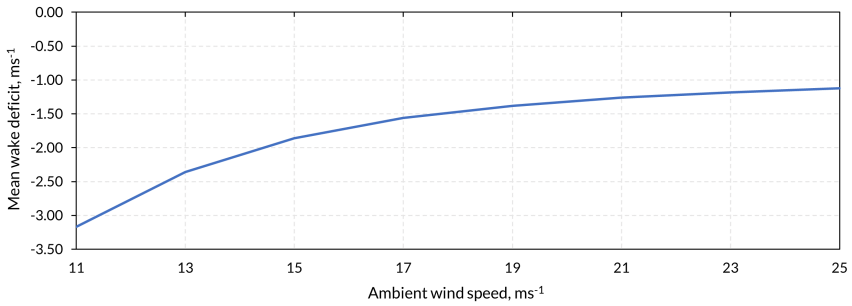


Figure 8.4: Six-seed-average mean wake deficit at 0 m radial distance and 1000 m downstream distance of T1 across above-rated mean ambient wind speeds.

8.3.2 LIDAR-assisted control

This Section will investigate the potential for LAC to mitigate the impacts on downstream turbines within an array. Figure 8.5 shows the mean values of the same parameters shown in Fig. 8.3, with data normalised to the results of T1 with FB-only control, and compares the results of a downstream turbine (T2) when both turbines used FB-only control compared to when both used LAC.

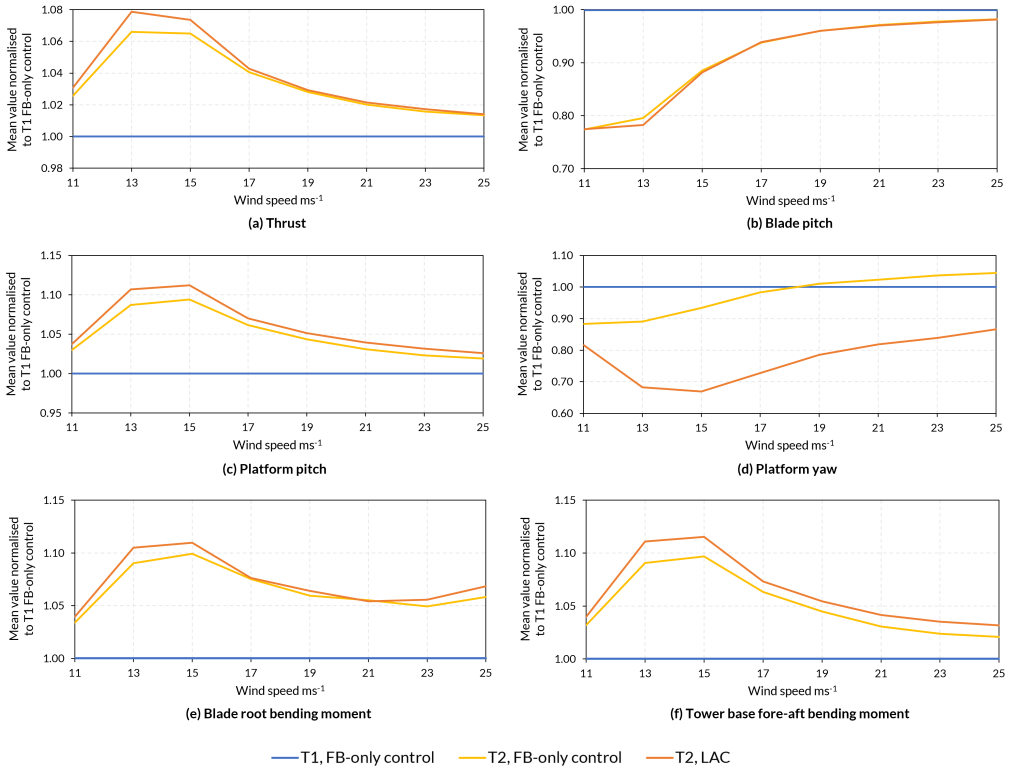
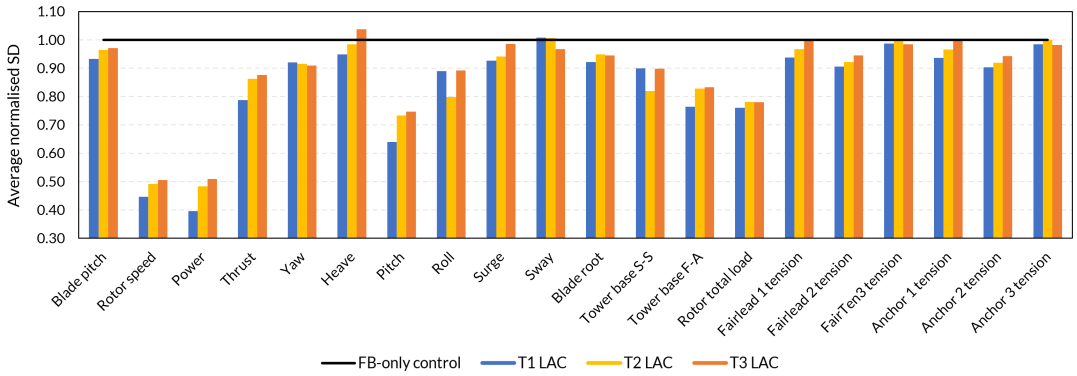
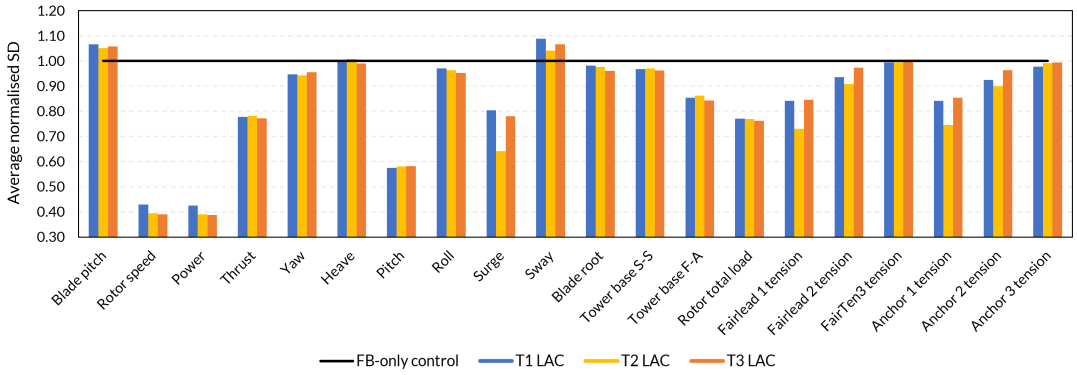


Figure 8.5: Six-seed-average mean values of T2 parameters, normalised to that of T1, when using FB-only control or LAC across above-rated wind speeds and with a turbine spacing of 8 RD.

It is clear that when both turbines used LAC, the increases to the means of the parameters could not be overcome and so LAC was unable to mitigate the detriment to the fatigue of turbine components that this brought. However, the addition of LAC was able to offer reductions to the standard deviations of the parameters of all three turbines, by up to 62%, showing similar reductions to those reported in Chapter 5, relative to their own turbine using FB-only control. The reductions to the standard deviations varied depending on the mean wind speed, as well as between the three FOWTs as a result of their positions, as shown in Fig. 8.6.



(a) $\bar{V}$ = 15 ms^{-1}



(b) $\bar{V}$ = 23 ms^{-1}

Figure 8.6: Six-seed-average standard deviations of properties associated with the three turbines when using LAC, normalised to that when using FB-only control for each turbine under $\bar{V}$ = 15 ms^{-1} or $\bar{V}$ = 23 ms^{-1} and with a turbine spacing of 8 RD.

Figure 8.6a shows that the reductions to the standard deviations of the parameters delivered by LAC over the FB-only control were generally greater for T1 than for the downstream turbines (T2 and T3), compared to their respective FB-only control responses, at a mean wind speed of 15 ms^{-1}. However at 23 ms^{-1} (Fig. 8.6b), there was significantly less disparity between the results of the three turbines, as average differences between the reductions to the standard deviations of T1, T2 and T3 decreased by 44%, from 4.3% to 2.4% between mean wind speeds of 15 ms^{-1} and 23 ms^{-1}. This was again due to the reduced wake deficit

at higher wind speeds (shown in Fig. 8.4) as a result of the wind turbines extracting a smaller proportion of the kinetic energy from the wind. This meant that the three turbines experienced comparable wind speeds at the higher mean wind speeds, meaning that their responses, and the LAC improvements over the FB-only control responses, were more similar than those at lower wind speeds.

The addition of LAC was also able to significantly reduce the standard deviations seen in the parameters of the downstream turbines relative to T1 using FB-only control, as shown in Fig. 8.7.

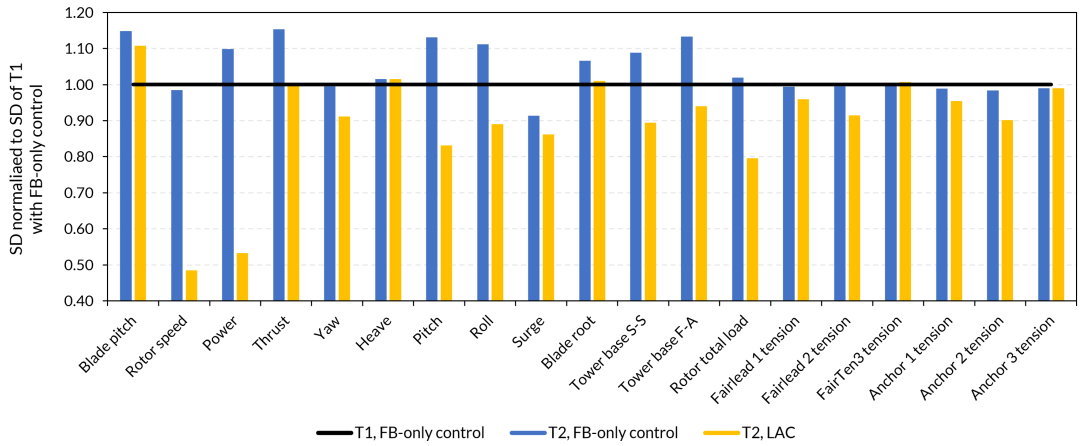


Figure 8.7: Six-seed-average standard deviations of various parameters of T2 when using FB-only control or LAC, normalised to those of T1 FB under turbulent wind ($\bar{V}$ = 15 $^{-1}$, NTM, TI Category B) and irregular wave (H_s = 2.39 m, T_p = 7.85 s) conditions and with a turbine spacing of 8 RD.

The standard deviations of T2's parameters when using FB-only control generally increased compared to T1 using FB-only control at a mean ambient wind speed of 15 ms^{-1}. This was particularly evident for the power, rotor thrust, platform pitch, platform roll and tower base and blade root bending moments, with increases in standard deviations of up to 15%, relative to T1 using FB-only control. However, the addition of LAC was able to prevent many of these increases, with standard deviations of T2's parameters reduced to similar or less than that of T1 (using FB-only control) by up to 52% when using LAC. This resulted in reduced loadings and motions of the downstream turbines, which may allow for lifetime extension and reduced failures of turbine components compared to when FB-only control is employed, which may yield similar OpEx savings as were reported in Chapter 6.

Wake assessment

Many of the benefits brought by the addition of LAC to the downstream turbines can be explained through the examination of the wake behaviour of the upstream turbine. This analysis included assessing the impacts upon the turbulence intensity, wake diameter, wake centre line and disturbed wind velocity profile. For these parameters, no or negligible difference was recorded between the simulations with and without LIDAR-assisted FFCPC+FFIPC enabled.

However, of the recorded FAST.Farm output parameters, significant differences were observed in the wake velocity deficit behaviour. The wake velocity deficit (V_d) is defined as the difference between the disturbed wake velocity (V_w) and the ambient wind velocity (V_a).

$$V_d = V_w - V_a \tag{8.1}$$

In FAST.Farm, wake deficits are recorded through the specification of various radial distances in the rotor plane at different distances behind the turbine rotors. A time-series example of the benefit brought by the LAC addition compared to the use of FB-only control of T1 at a rotor radial position of 0 m and 800 m downstream of T1 at a mean ambient wind speed of 13 ms^{-1} is shown in Fig. 8.8.

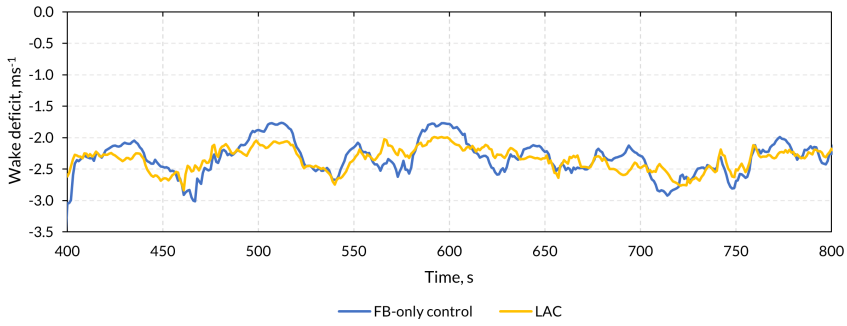


Figure 8.8: 400-second excerpt of the wake deficit recorded at 0 m radial position and 800 m downstream of T1 under turbulent wind ($\bar{V}$ = 13 ms^{-1}, NTM, TI Category B) and irregular wave (H_s = 2.01 m and T_p = 7.45 s) conditions.

From Fig. 8.8, it can be observed that the addition of LAC was able to reduce the standard deviation of T1's wake deficit (at 0 m radial distance and 800 m downstream distance) by 13% compared to when it used FB-only control, attributed to the reductions of the standard deviations of the rotor speed (by 47%) and power by (39%) of T1, compared to when using FB-only control.

When examining different properties of the wake deficit at different radial and downstream distances of T1 at a mean ambient wind field of 13 ms^{-1}, the benefits of the LAC addition became more evident. Figure 8.9 shows the average range of the wake deficit obtained from time series simulations of the wake at different radial nodes and distances downstream of T1.

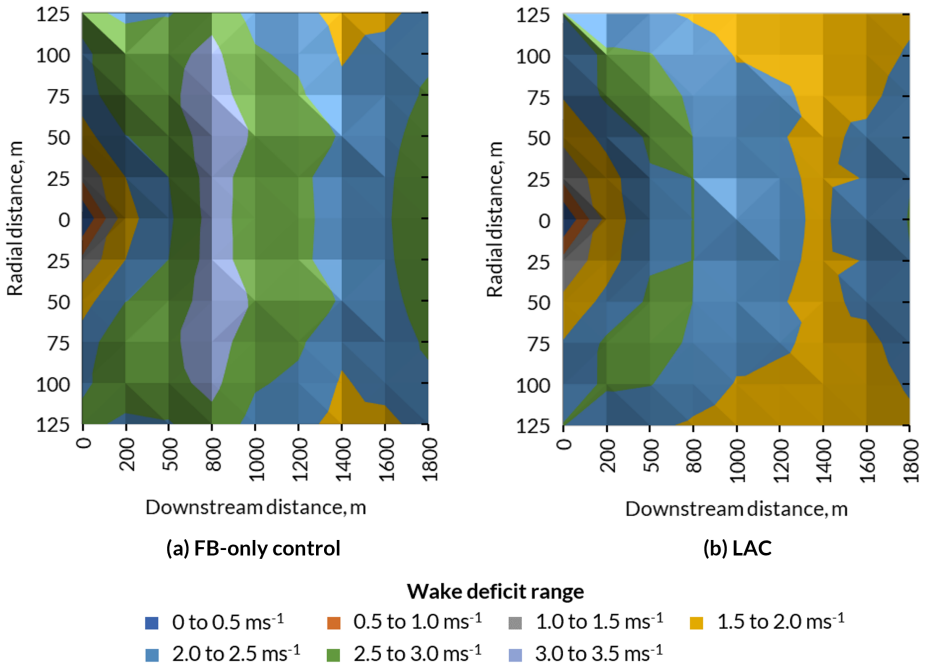


Figure 8.9: Six-seed-average range of the wake deficit recorded at various radial and downstream distances of T1 under turbulent wind ($\bar{V}$ = 13 ms^{-1}, NTM, TI Category B) and irregular wave (H_s = 2.01 m, T_p = 7.45 s) conditions when using FB-only control or LAC.

With the addition of LAC, the range of the wake deficit was reduced across, effectively, the full length of the propagation of the wake between two turbines. The most noticeable improvement in the range of the wake deficit shown in Fig. 8.9 occurred between radial distances of +/- 100 m at an 800 m downstream distance, where the range of the wake deficit was reduced by approximately 25%, from 3.26 ms^{-1} when using FB-only control to 2.50 ms^{-1} when using LAC. Overall, this signifies a more consistent wake when using LAC, attributed to the reduction in the variation of the rotor speed and power capture. This was further supported when examining the maximal wake deficits of T1's wake at different radial nodes and downstream distances in Fig. 8.10.

The addition of LAC allowed for the maximum wake deficit to be decreased at lesser distances downstream of T1, with up to 8% reductions in the maximal wake deficit compared to the wake with FB-only control at a mean wind speed of 13 ms^{-1}. This is evident when focusing on the right-hand side of each sub-figure, where the maximum deficit when using FB-only control (Fig. 8.10a) remained between -3.5 and -3.0 ms^{-1} at a downstream distance of 1600 m and between radial distances of +/- 25 m, while this was reduced to between -3.0 and -2.5 ms^{-1} with LAC enabled (Fig. 8.10b) for all radial distances at 1600 m downstream distance. Furthermore, at a downstream distance of 1800 m, the maximum wake deficit when using FB-only control was between -3.0 and -2.5 ms^{-1} between radial distances of +/- 80 m, while this was only between +/- 25 m when using LAC.

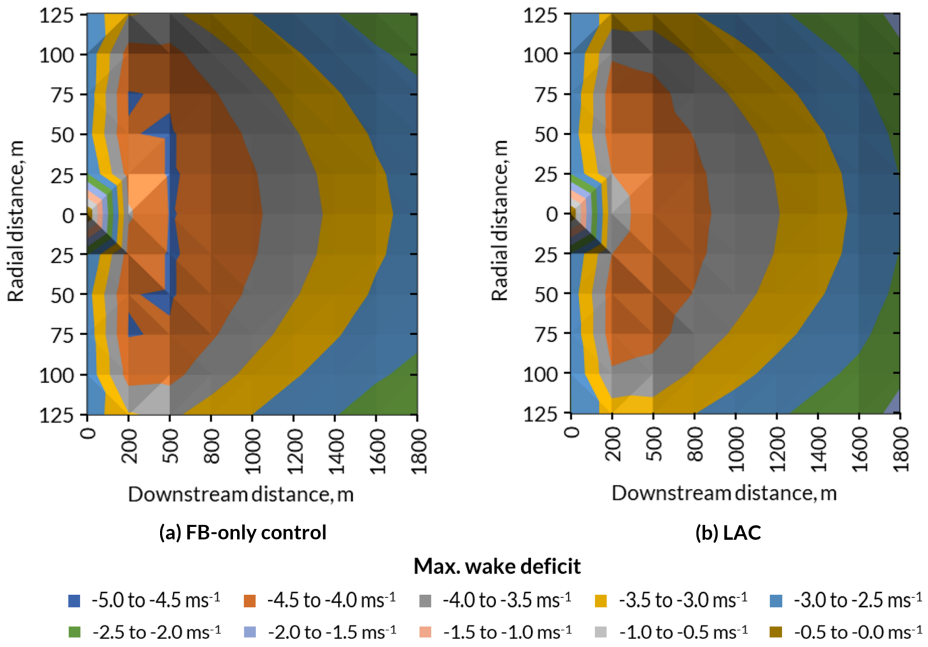


Figure 8.10: Six-seed-average maximum wake deficit recorded at various radial and downstream distances of T1 under turbulent wind ($\bar{V}$ = 13 ms^{-1}, NTM, TI Category B) and irregular wave (H_s = 2.01 m, T_p = 7.45 s) conditions when using FB-only control or LAC.

Improvements to the wake recovery and consistency were also highlighted by examining the standard deviation of the wake deficit at different radial and downstream distances of T1, shown in Fig. 8.11.

Figure 8.11 indicates a clear benefit of LAC to reduce the standard deviation of the wake deficit, by up to 16%, relative to when using FB-only control, at a mean ambient wind speed of 13 ms^{-1}. Again, this is evident when examining the right-hand side of the sub-figures, where, when using LAC (Fig. 8.11b), the standard deviation of the wake deficit reduced to between 0.3 and 0.4 ms^{-1} at all radial distances at 1500 m downstream distance, whereas this remained between 0.4 and 0.5 ms^{-1} at radial distances up to +/- 100 m when using FB-only control (Fig. 8.11a). There was also less variation in the wake at downstream distances between 700 and 1200 m, where the standard deviation of the wake remains below 0.5 ms^{-1} when using LAC compared to up to 0.6 ms^{-1} when using FB-only control. When using LAC, the standard deviation increased to between 0.4 and 0.5 ms^{-1} between radial distances up to +/- 60 m and a downstream distance of 1800m. This is proposed to be due to the overlap with the high-resolution domain of the downstream turbine, which began at a 1620 m downstream distance.

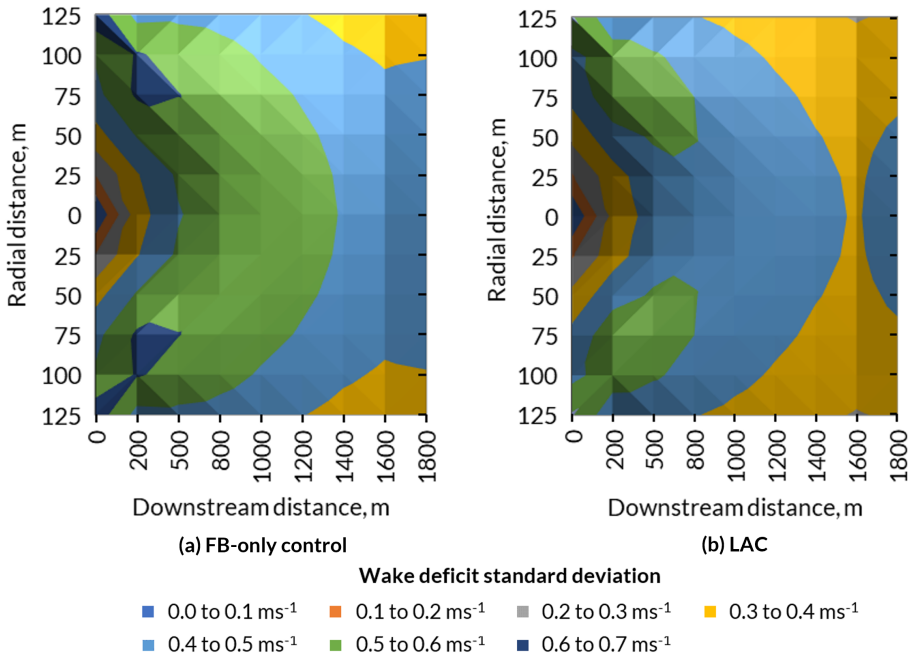


Figure 8.11: Six-seed-average standard deviation of the wake deficit, recorded at various radial and downstream distances of T1 under turbulent wind ($\bar{V}$ = 13 ms^{-1}, NTM, TI Category B) and irregular wave (H_s = 2.01 m, T_p = 7.45 s) conditions when using FB-only control or LAC.

Ultimately, the impacts of the improved wake consistency and recovery brought by LAC can be shown by examining the mean power of the downstream turbines at different mean wind speeds relative to T1, shown in Fig. 8.12.

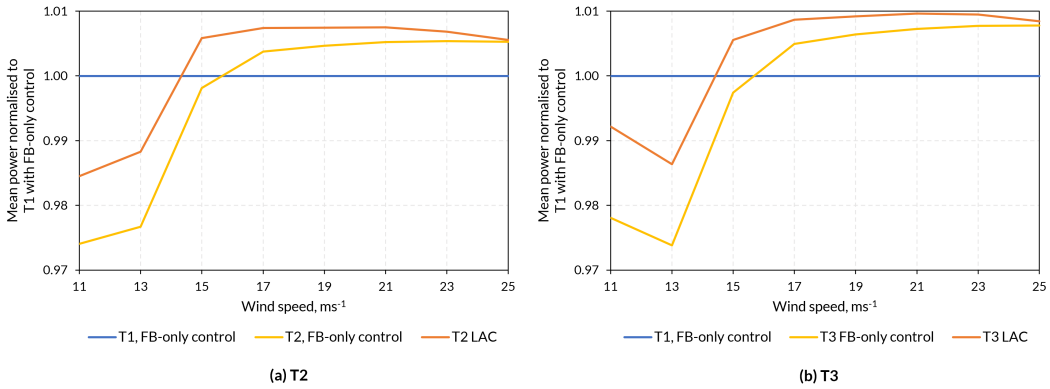


Figure 8.12: Six-seed-average mean power of T2 and T3 when using FB-only control or LAC, normalised to the mean power of T1 using FB-only under above-rated wind speeds and with a turbine spacing of 8 RD.

Here, it can be seen that LAC was able to increase the mean power of the downstream turbines by between 1.0-1.4% for turbines spaced at 8 RD for mean wind speeds between 11-15 ms^{-1}, relative to T1 using FB-only control. This was due to decreased reductions in the ambient wind speed and improved wake recovery. This meant that, when the ambient wind speed was slightly above the rated wind speed, the addition of LAC allowed for the wake to recover faster, meaning that the downstream turbines experienced a less significantly reduced wind speed and, therefore, more power could be captured at rated capacity.

Chapter 9

Discussion, conclusions and opportunities for future work

Within this Chapter, the research findings will first be discussed, outlining the limitations of the methodologies and providing critique on them, whilst noting the key results and their significance in the wider context of the FOW industry. The discussion will be followed by the conclusions drawn from the research findings. Subsequently, future work recommendations will be presented, which will offer guidance on ways to address the limitations of the methodologies used in this research and to further explore research areas pertinent to LAC within FOW.

9.1 Discussion

9.1.1 LIDAR simulator, control strategies and floating offshore wind turbine platform configuration results

Following the point raised in Sect. 1.3.3 regarding the applicability of LAC to different markets around the world, the results have shown that LAC can provide benefits to the platform designs that would be suitable for different locations and markets. However, the implementation of LAC may be required to be different (e.g. FFCPC+FFIPC for the VolturnUS-S and FFCPC+FBIPC for the WindCrete) and may also be more challenging for some FOWTs than others due to the differing characteristics of their baseline controllers and their platform motions, which impact upon the economic benefits brought by LAC. For example, the WindCrete's significantly less responsive baseline pitch controller was detrimental to the FF control implementation, which meant that the FF commands had to be set to zero in measured wind speeds below 13 ms^{-1} to avoid instability of the FOWT. This resulted in minimal FF control benefits to the WindCrete in low above-rated wind speeds. Although this was not studied explicitly within this work, the imposed limits to the WindCrete are expected to detract from the benefits brought by LAC to the tower base loads (which drove the VolturnUS-S tower design simplifications detailed in Chapter 7) and the power capture of downstream turbines at the array level, which both

occurred most significantly at low above-rated wind speeds where LAC was inactive within the WindCrete. This ultimately reduces the economic benefits of LAC for the WindCrete and, therefore, it may be possible to conclude that LAC will be best suited to deliver economic benefits to FOWT designs where the FF control is not restricted in low above-rated wind speeds. In the unlikely scenario that the control behaviour of the VolturnUS-S and WindCrete are consistent across all semi-submersible and spar designs, it can be concluded that LAC will be best suited to shallow-water FOW farm locations, where semi-submersibles are more suitable than spars.

However, despite these findings, there is also scope for reconfiguration of the FB controller when combined with the FF control elements, due to the ability of the FF controller to bypass the bandwidth limitation of the pitch controller of the FOWTs. This may allow for the FB controller gains of the FOWTs to be increased from the default values used within the work, which would render the FB pitch controller more responsive and lead to improved benefits of the FF control addition. This may also allow for the restrictions applied to the FF control of the WindCrete in close to rated wind speeds to be alleviated, which may allow for it to reap similar economic benefits as those reported for the implementation within the VolturnUS-S.

Despite the results presented throughout this work, the limitations of the LIDAR simulator implemented within OpenFAST and the FF controllers implemented within ROSCO, and their impacts on the findings, must be discussed. Firstly, the wind velocity was assumed to not evolve from the point of measurement to the turbine (thereby assuming Taylor's frozen hypothesis). This assumption is applicable when using InflowWind within OpenFAST, because InflowWind itself, and by extension the LIDAR simulator implemented within it in this work, utilises Taylor's frozen turbulence hypothesis to propagate turbulent wind data generated by TurbSim over time. The InflowWind module applies Taylor's frozen hypothesis to propagate the 2D grid (in the Y and Z directions) of turbulent inflow data from TurbSim along the X-axis, creating a time-marching 3D grid of turbulent inflow data. However, in a real-life scenario, the Taylor's frozen hypothesis assumption may lead to errors in the feedforward control action, and subsequent detrimental impacts upon the turbine's performance, loadings and stability, because commands will be formulated based on a measured wind velocity profile that will be different from the one impacting upon the turbine. A Kalman filter can be applied, as in Guo, Schlipf, and Cheng (2023), to predict the wind speed at the rotor based on the upstream LIDAR measurements and a wind evolution model. This can allow for corrections to be made to the feedforward control commands, thereby reducing errors resulting from wind evolution and turbulence. Secondly, 100% LIDAR measurement availability was assumed. Consequently, any potential LIDAR availability and reliability issues that may result from the contamination, aerosol concentration and hardware reliability issues introduced in Sect. 1.3.2, as well as the impact of periodic blockage of the LIDAR beams by the blades, were not considered.

Furthermore, the impacts of the velocities, accelerations and rotations of the LIDAR device and their effects on the LIDAR measurements were not accounted for, though these impacts were assumed to not have been captured because the single-point beams were not coupled to these movements, as was explained in Sect. 3.2.5. The assumptions made may have resulted in an overestimation of the LIDAR preview quality, meaning that the control action was less restricted by the low-pass filtering, which may have resulted in an overestimation of the FF control benefits and the subsequent cost reductions reported.

The nature of the single-point beams used in the LIDAR setup of this work also overlooks the potential uncertainties and errors resulting from the invocation of a wind field reconstruction method as well as the requirement for LOS wind velocity correction of the outer beams as a result of the angling of the beams. If multiple pulsed of CW LIDAR beams had instead been used, uncertainties resulting from the spatial averaging of the wind velocity measurements over the probe volume would have emerged. Furthermore, when considering the individual beam measurements used for the FFIPC, the use of pulsed or CW LIDAR could introduce concerns over the Cyclops dilemma, as single-beam measurements may struggle to distinguish between horizontal wind shear and wind direction changes. In this case, an approach of pairing LIDAR beams, such as by pairing the outer beams with the centre beam, may be necessary to ensure that the Cyclops dilemma is avoided. Additionally, given that the outer LIDAR beams would be projected at angles not perpendicular to the wind flow, when modelled as continuous or pulsed beams, corrections would have been required to accurately estimate the longitudinal wind component.

A potential underestimation is that, when using the FFIPC, each beam's measurement was low-pass filtered according to the coherence of the turbine-experienced and LIDAR-measured REWSs, determined in Sect. 4.5.1. Dunne, Simley, and Pao (2011) and Simley and Pao (2013) discussed the use of a blade-effective wind speed, which characterises the wind speed experienced by an individual blade as opposed to the full rotor when using the rotor-effective wind speed. The previewed wind at different azimuth positions captured by the multiple LIDAR beams may better match the wind speeds experienced by the individual blades than the REWS, which may lead to higher coherences between the measured and actual blade-effective wind speeds. It follows that an assessment of the coherence between the experienced and LIDAR-measured blade-effective wind speeds may allow for a higher cutoff frequency of the low-pass filter used on the individual beam wind speed measurements. This may allow for more freedom in the FFIPC control action, potentially improving the results or, at least, cancelling out the reduced benefits that may emerge from a more accurate estimation of the LIDAR preview quality. However, this is only a hypothesis and the blade-effective wind speed will also be subject to the aforementioned, as yet, unquantified overestimation of the

LIDAR preview quality. Therefore, it is recommended that the OpenFAST simulation results and, by extension, the outcomes of the O&M, tower design, and array-level case studies, be considered the best-case scenario achievable with the detailed LIDAR-assisted controllers used in this work, until proven otherwise.

The FFIPC configuration also assumed that the rotor speed did not change in the duration of the time delay incurred by the low-pass filter applied to the individual commands. This was estimated to result in a percentage misassignment between the blades and the beams when issuing the FFIPC commands of approximately 4-8%, depending on the wind speed. This resulted from errors in the predictions of the azimuth angles of the blades following the time delay incurred by the low-pass filter applied to the FFIPC commands. The resulting impact of these misassignments on the FFCPC+FFIPC results was not quantified in this work, though this is recommended to be estimated in future work.

Furthermore, only aligned wind and wave conditions propagating perpendicular to the turbines' rotor planes under DLC 1.2 were simulated within Chapter 5. Therefore, the impacts of wind-wave misalignment, wind direction changes or extreme conditions were not considered.

9.1.2 O&M case study

The O&M case study presented in Chapter 6 focused on the OpEx savings resulting from the reduction of failures associated with the RNA for minor and major repairs using an SOV supported by a fleet of daughter craft. Yet, as highlighted within the study, the moorings, blade root, pitch bearing and tower also experienced reductions in fatigue due to the implementation of LAC, and therefore, the reported potential OpEx savings could be increased. However, a full joint probability density function of wind and wave directions should be performed, in a similar manner to that performed within the tower design case study (Chapter 7) in order to better determine the lifetime failure rate reductions of the moorings and towers.

An important aspect when considering the implementation of FF collective and individual pitch control is that of its impact on the blade pitching rate, which was found to increase for both FOWTs compared to their respective baseline FB-only controllers within Chapter 5. The O&M case study in Chapter 6 reported upon the benefits of reduced failure rates resulting from reduced loadings at the pitch bearing. However, it is important to note that the failure rate reductions reported were solely based on the loadings imparted on the components and so they did not account for failures arising from mechanical wear of the pitch system from control actuation, which may be expected to increase due to the increased pitch rate. Yet, despite the pitch rate increases that were recorded when using the FFIPC, these were significantly less than those resulting from use of the FBIPC and, therefore, the FFIPC offers a less detrimental alternative to the FBIPC in regards to the pitch rate increase. Nevertheless, for the pitch

actuation systems and other turbine components of complex geometry that cannot be fitted accurately to an S-N curve, it is recommended that future work further investigates the benefits or detriment of LAC to these components/sub-systems through the use of an FEA tool, which can provide a higher-fidelity estimation of the fatigue of these components.

Another key discussion point of the O&M case study is that the baseline failure rates used within the O&M simulations, and to which reductions were applied according to the LAC loading reductions, were based on those reported by Carroll et al. (2016). The failure rates reported by Carroll et al. (2016) were obtained from data from BFW turbines with ratings between 2 and 4 MW. Therefore, it is probable that the failure rates of the sub-assemblies of 15 MW FOWTs will be different to those presented by Carroll et al. (2016). However, as 15 MW BFW and FOWTs are yet to be installed at scale, comprehensive failure rate data for these turbines has not yet been established. Hence, the failure rate data from Carroll et al. (2016) was deemed to be the best available reference for the baseline failure rates used within the O&M case study.

When considering the WI impacts on the OpEx of the four different FOWT designs used by Scheu et al. (2018), the same failure rate reductions (obtained from the VolturnUS-S OpenFAST simulations) were applied to each FOWT within the O&M case study because specifics of each platform were not disclosed by Scheu et al. (2018). As the results in Chapter 5 demonstrated, different FOWT platform designs can lead to variations in the benefits achievable from the implementation of LAC, leading to potential dissimilarities in the failure rate reductions across different FOWTs. Consequently, this is likely to impact upon the OpEx reductions achievable for different platform designs, in addition to the considered varying WI limits. Therefore, further work is required to categorise the LAC-delivered failure rate reductions for different FOWT designs and how this can affect the OpEx reductions.

Furthermore, while minor/major repairs can account for up to 90% of the total lifetime failure rates, the costs associated with major component replacements (which were not considered within the O&M case study) make up a significant portion of the total OpEx for the full site life cycle for the reasons discussed in Sect. 1.3.7. Major component replacement operations of FOW are highly costly due to the time involved in tow-to-shore operations and the resulting downtime of the asset(s), or the cost of specialist vessels for in situ maintenance. Therefore, the reduction in failures resulting from LAC for these maintenance events could have significant operational savings.

9.1.3 Tower design case study

Chapter 7 documented a case study that aimed to utilise the superior rotor speed regulation and reduced tower loads delivered by LAC to enable reductions in the VolturnUS-S's tower outer diameter or shell thickness.

The finalised proposed lightweight tower consisted of sections which each had a 20% lesser shell thickness than the equivalent section of the baseline tower. This design was selected because it maintained similar levels of equivalent design stress on the tower when supporting a LIDAR-assisted-controlled FOWT, to that of the baseline tower of a FB-only-controlled FOWT under aligned and misaligned wind and wave directionality. It therefore demonstrates the potential to reduce the mass and, by extension, the cost of towers supporting LIDAR-assisted-controlled FOWTs, by up to 20%, without sacrificing structural integrity.

The lightweight design remained conservative by maintaining the tower's first natural frequencies above 3P (see Figs. 7.3 and 7.7), which ensured no interference with the rotor's 3P rotation frequency, while limiting increases to the nacelle acceleration and tower-top displacement of the FOWT. Though it was desirable to place the tower's natural frequencies within the 1P and 3P interval of the rotor to maximise the mass and cost reductions, these were found to not be viable. As was also reported by Papi et al. (2023), their unsuitability was not only due to the increased stresses on the towers, but it was also due to the increased nacelle accelerations and displacements as a result of the increased elasticity of these towers.

Furthermore, prior to conducting the tower design case study simulations, it was anticipated that the enhanced rotor speed regulation provided by LAC would allow for reduced tower stress. This reduction was expected because the improved rotor speed regulation delivered by LAC allowed for the constraining rotor speed range to be reduced, as shown in Fig. 7.1. This was expected to minimise the risk of excitation of the tower by the rotor as the first natural frequencies of the tower designs approached the rated 3P frequency point. When incrementally reducing the thickness or outer diameter, and, by extension, the first natural frequencies of the tower, the stress on the lightweight towers supporting LIDAR-assisted-controlled FOWTs surpassed that of the baseline tower design supporting an FB-only controlled FOWT. This occurred at tower first natural frequencies greater than the reduced 3P constraining frequency range of the rotor. As only lightweight towers that had a similar level of stress (when supporting a LIDAR-assisted-controlled FOWT) to that of the baseline tower supporting a FB-only controlled FOWT were considered within the tower design case study, the improved rotor speed regulation provided by LAC, and the subsequent reduced risk of excitation of the tower by the rotor, ultimately proved to be irrelevant to the viability of the modified tower designs.

Nonetheless, the lesser tower mass of the finalised lightweight tower has the potential to deliver economic benefits through reduced material usage. Given the price volatility of steel, it is difficult to provide an accurate estimate of the cost savings resulting from the use of the lightweight tower. However, as the mass of the lightweight tower was 20% less than that of the baseline tower, it is logical to assume that this can lead to an approximately 20% reduction in the capital cost per tower.

The reduced steel requirement also presents a route to reduce the water usage and carbon emissions of the tower manufacturing process. The IEA estimates that direct CO_2 emissions due to crude steel production are approximately 1.4 tonnes CO_2 per tonne of steel produced (see Sustainable Ships, 2023), while water usage can range from 4,180 to 26,700 gallons per tonne of steel (see Walling & Otts, 1967). Therefore, a 20% reduction of the mass of the baseline tower, approximately 300 tonnes of steel per tower, may enable a reduction of up to 420 tonnes of CO_2 emissions and between 1-8 million gallons of water per tower. Thus, the findings of this research highlight the value, not only from an economic perspective but from an environmental one as well, of wind turbine structural design involving LAC.

The total standard deviation of the equivalent design stress across all of the above-rated mean wind speed and wind-wave misalignment angle bin combinations of the lightweight tower supporting a LIDAR-assisted-controlled FOWT remained 9.1% below that of the baseline-tower/FB-only-controlled FOWT. Therefore, there is scope for further structural optimisation to be performed. As the dynamic behaviour of a FOWT tower is influenced by other components such as the floating platform and mooring line design, control strategies and nacelle-drivetrain weight, the tower design case study could be expanded with a co-design or optimisation approach, in order to fully reap the benefits of LAC for structural design.

Despite the extreme condition and joint-probability density distribution weighted wind speed/wind-wave misalignment analysis performed within the tower design case study, the simulations performed did not cover an exhaustive suite of DLC simulations covering all potential wind, wave and current scenarios that the FOWT will be subject to over its lifetime. Therefore, further analysis is required to verify that the proposed lightweight tower supporting a LIDAR-assisted-controlled FOWT could, beyond any reasonable doubt, withstand the lifetime extreme and fatigue loads that it would be subjected to. This may include performing simulations characterising a wider variety of normal and extreme sea states, with a greater number of combinations of wind speeds, significant wave heights, peak spectral periods and wind-wave misalignment, as was performed by Barj et al. (2014). Additionally, data-driven joint-probability density distribution characterisation of wind-wave conditions should be performed in order to fully capture the site-specific lifetime extreme and fatigue loads of the lightweight-tower/LIDAR-assisted-controlled FOWT. This is because, within the tower design case study, only the wind probability density distribution was obtained from data of the investigated site. Only the expected values of H_s and T_p equal to those used by Allen et al. (2020) were applied

within each mean wind speed bin. Meanwhile, a normal distribution of wind-wave direction misalignment angles that was not based on site-specific data was used to represent the probability density distribution of the wind-wave misalignment. Ultimately, the results of the tower design case study performed in Chapter 7 should be considered to be preliminary until these more exhaustive design simulations are performed alongside realistic joint wind-wave probability density distributions.

The tower design case study also applied assumptions, primarily when investigating scenarios of wind-wave direction misalignment, to simplify the analysis. Firstly, the wind direction was fixed at 0° with no yaw misalignment and, therefore, the wind-wave misalignment was equal to the wave direction. As a fixed wind direction of 0° focuses the fore-aft tower loads on the 0° side of the tower, it subsequently does not capture the full extent of the loads applied to the remaining circumference of the tower. Future work should consider separate wind and wave direction distributions to more accurately characterise the lifetime fatigue damage accumulation across the full circumference of the tower. Secondly, only positive wind-wave misalignment angles between 0° and 180° were considered in the case study. Equal positive and negative wind-wave misalignments (i.e. +90° and -90°, for example) were assumed to result in identical responses. This can lead to asymmetrical fatigue loads upon the tower. Therefore, it may be necessary for future work to determine the unique responses resulting from positive and negative misalignment angles and their impacts upon the tower extreme and fatigue loads.

Furthermore, in the tower design case study, constraints on the diameter-to-thickness ratio of the tower were not considered, which may have otherwise limited the degree of reduction that could have been applied to the diameter or thickness, due to the risk of buckling of the structure. Moreover, as the tower design case study only involved independent adjustments of the design variables (tower outer diameter and tower shell thickness), it may have overlooked the coupling effects of adjusting the design variables simultaneously. Additionally, the tower design case study did not investigate the significant stress expected to be applied to the towers during the towing process of the FOWT installation, and whether the baseline and modified designs could withstand them.

The tower design case study also assumed that the coherence between the LIDAR-measured and turbine-experienced REWS did not change with changing tower geometry. Due to the slightly increased nacelle accelerations and tower-top displacement, it is plausible that the LIDAR preview quality will deteriorate as the towers become more flexible. This will result in lower cutoff frequencies of the low-pass filter of the LIDAR measurements, thus restricting the control action. The change in LIDAR preview quality due to tower structural changes was not examined in the tower design case study, and future work may seek to investigate this and its impact upon the results.

9.1.4 Array-level case study

The key result of the VolturnUS-S array-level case study (Chapter 8) was that LAC was found to improve the mean power of downstream turbines by up to 1.4% at ambient wind speeds close to the rated wind speed. Through consideration of a Weibull distribution, this power capture enhancement is estimated to equate to an up to 0.8% increase in the annual energy production (AEP) of the downstream turbines and, therefore, provides a route to increased revenues. Given the indications that LAC can improve the wake consistency and recovery of FOWTs, it may also help to reduce the global blockage effect of wind farms by allowing for expedited wind recovery between neighbouring wind farms, and thereby increasing the energy yield of downstream wind farms.

Despite the results presented, the array-level study was not without its limitations and scope for further work to consolidate the findings. Firstly, the coherence between the measured and actual wind was assumed to be the same for all turbines. In reality, the coherence of downstream turbines (T2 and T3) will likely be less than that of the upstream turbine (T1) due to the disturbance brought by the wake, and its subsequent evolution from the point of measurement to the rotor, which the LIDAR simulator and LAC setup do not account for in their current states. This, therefore, may have resulted in an overestimation of the FF control benefits of the downstream turbines.

Furthermore, this array-level case study has only investigated arrangements where turbines are directly in line with each other, and so has not examined situations where turbines are arranged in a staggered fashion, and where the wake of the upstream turbine does not propagate directly towards the rotor plane of the downstream turbine. Given that the turbines were aligned in the study performed in Chapter 8, the downstream turbines will have experienced stronger wake effects, higher turbulence and lower wind speeds than if a staggered arrangement had been simulated. A staggered arrangement can aid in mitigating the intensity of the wake effect by alleviating the wind speed deficit and increased turbulence experienced by the downstream turbines (see Zhao et al., 2024), which may negate the power capture improvements shown by LAC for the aligned downstream turbines. However, this may also positively influence the effectiveness of LAC of downstream turbines by providing more accurate control inputs based on the lesser turbulent wind, instead leading to increased performance improvements and loading and platform motion reductions of downstream turbines that better match those of the first turbine.

The effects of yaw misalignment, wake steering and wind farm control strategies were also outside of the scope of this work, though should also be considered alongside an individual turbine LAC approach.

In performing the simulations, all three turbines used the same control strategy within a given simulation run (LAC or FB) and, therefore, the array-level case study has not examined the effects of varying control strategies such as LAC on T1 and FB-only control on T2 and T3, for example. If the benefits to downstream turbines persist with only the first turbine using LAC, this could help to reduce the CAPEX associated with purchasing and implementing multiple LIDAR devices. Additionally, the three turbines' FB pitch controller gains were the same, with no tuning performed depending on the turbine positions, which may have otherwise improved the baseline control performance of the downstream turbines.

Finally, as FAST.Farm is a mid-fidelity tool employing the dynamic wake meandering model to simulate the wake behaviour, there may be some inaccuracies in the representation of the complex wake behaviour, particularly for FOWTs. For example, FAST.Farm assumes steady conditions for modelling wakes (see J. Jonkman & Shaler, 2021), but for FOWTs, the wake deflection is strongly influenced by the platform motions and rotations, which may have lead to inaccuracies in the predictions of the wake's impact on the downstream turbines. Furthermore, the motions of FOWTs can lead to increased mixing of air in the wake region, which can cause the wakes to recover faster than predicted by FAST.Farm's steady-state wake model (see Carmo et al., 2024). Consequently, this means that the downstream FOWTs may experience greater wind speeds than those predicted, which may reduce the power capture benefits brought by LAC in the study presented in this work. In this instance, it may be possible to reduce the spacing between the turbines to allow for the LAC to realise similar downstream turbine power capture benefits as were documented in Chapter 8. , FAST.Farm considers steady-flow assumptions of the wake (see J. Jonkman & Shaler, 2021), which may not fully account for the increased turbulence in the wake due to the platform motions of FOWTs. This may have led to an underestimation of the loadings applied to the downstream turbines and subsequent inaccuracies in the LAC loading reduction predictions of the downstream turbines. Moreover, recent studies, such as that by Bernard et al. (2024), have commented on the shortcomings of the DWM model used by FAST.Farm and how its use according to the IEC standards bears a risk of non-conservative load design. Given these findings and the limitations of the FAST.Farm wake model, it is acknowledged that studies using higher fidelity (such as computational fluid dynamics [CFD]) tools are required in order to validate and verify the presented results.

9.2 Conclusions

This thesis has presented a novel LIDAR-assisted feedforward (FF) individual pitch control (IPC) (FFIPC) approach applied to two 15 MW floating offshore wind turbine (FOWT) models within an aero-elastic-servo-hydro simulation tool. The results of the FOWT simulations were carried forward and applied to an operations and maintenance (O&M) model, a structural design tool and a wind farm simulation tool. These applications aimed to ascertain the cost benefits achievable via:

- Assessment of LIDAR-assisted feedforward pitch control (LAC)'s applicability to different FOWT platform designs.
- Decreased failure rates as a result of reduced loads applied to the FOWT's components and, therefore, reduced maintenance activities and operational expenditure (OpEx).
- Reduced steel requirement and, hence, capital costs of FOWT towers due to the reduced tower loads and improved rotor speed regulation of FOWTs enabled by the use of LAC.
- Improved power capture of downstream turbines resulting from improved wake recovery of upstream turbines, leading to higher revenues.

LIDAR simulation was implemented within the original framework of NREL's OpenFAST aero-elastic-servo-hydro wind modelling tool by modification to the source code of the InflowWind module. Modifications enabled the curation of a new LIDAR parameters input section within the InflowWind input file. Although some LIDAR functionality existed within OpenFAST, the work done within this research project greatly improved its usability, and further enhanced its capabilities through coding for customisable focal positions, measurement position displacements, measurement intervals and multiple beams. The LIDAR outputs were interfaced with the ServoDyn control module, where measurements and focal positions were relayed to the ROSCO controller. The modifications made to the source code of OpenFAST were approved by NREL and are present in releases from v3.5 onwards. OpenFAST is widely used within both academia and industry and so these modifications represent a valuable contribution that can be used by the community to further build upon the knowledge of LIDAR simulation/LAC capabilities.

The source code of ROSCO was also modified to enable simulations of LAC. For this, changes were made to the source code of ROSCO to allow it to receive the LIDAR measurement and position data sent from ServoDyn but also for the inclusion of additional inputs within the ROSCO input file relevant to FF control. FF control subroutines were then implemented within ROSCO, conforming to its original pitch control framework. This included the curation of a novel FFIPC strategy, which provided tuning to the blade pitch angles, in addition to the command provided by the FF-feedback (FB) collective pitch controller (FFCPC), based upon each blade's azimuth position.

9.2.1 OpenFAST simulation results

Within the OpenFAST simulations performed in this work, with results documented in Chapter 5, the LIDAR simulator was configured with five beams. One beam was positioned directly ahead of the turbine and the others were positioned in 90° azimuth increments at 50% rotor radial span. This placement was found to give superior results compared to when beams were positioned at 30% and 70% rotor radial span. This was due to the 50% rotor radial span configuration offering the best balance between the rotor-effective wind speed used by the collective control element whilst capturing the variation across the rotor-swept area used for the individual control component. The LIDAR simulator and the FF controllers (FFCPC, FFCPC+FBIPC and FFCPC+FFIPC) were applied to two 15 MW FOWTs, the UMaine VolturnUS-S semi-submersible and the WindCrete spar, and compared to their respective baseline FB pitch controllers. The FOWTs exhibited significant differences in their baseline FB pitch control behaviour, which influenced the results obtained from the addition of LAC. Namely, the FB blade pitch actuation of the WindCrete was significantly slower than that of the VolturnUS-S due to its lesser proportional and integral pitch controller gains, meaning that its rotor speed was more prone to fluctuations, while the VolturnUS-S platform was found to be more dynamic in its motions. In summary of the simulation results outlined in Chapter 5, the capability of FFCPC to deliver reductions in the variations of the rotor speed, the rotor thrust, tower base and blade root bending moments and platform motions of the two FOWT concepts, by up to 73% relative to their respective baseline FB-only controllers, was demonstrated.

The addition of FFIPC to the FFCPC was found to have differing impacts upon the two FOWTs, and yielded different results to those of the addition of FBIPC to the FFCPC. For the VolturnUS-S, the FFIPC was able to enhance the benefit delivered by the FFCPC by further reducing the variation of the thrust force, the tower base bending moment and the platform pitch and surge motions, by up to 12%, compared to its baseline FB-only controller. These were more significant improvements than those delivered by the FBIPC addition, which instead reduced the variation of the blade root and side-to-side bending moments at the expense of significantly greater mean rates of pitch actuation, relative to its baseline FB-only controller.

For the WindCrete, the benefits of the addition of FFIPC to the FFCPC were less pronounced, particularly at higher above-rated wind speeds. This is because the FFCPC was able to reject much of the wind disturbance by increasing the blade pitch preparedness for the incoming wind, offering significant improvements over the FB-only pitch controller at the expense of increased blade pitch variation. Therefore, further tuning of the blade pitch by FFIPC was not found to add additional benefit to this. However, the addition of FBIPC to the FFCPC offered significant improvements for the WindCrete, as this used the additional feedback inputs (blade root bending moments) to formulate the individual commands as opposed to the wind velocity

used by the FFIPC. This allowed the FBIPC addition to deliver further improvements over the FFCPC addition, relative to the baseline FB-only controller, by reductions to the variation in the blade root and tower base side-to-side bending moments as well as the platform yaw and heave.

However, the main drawback of the FBIPC remains the vastly increased blade pitch actuation rate that it applies, which will lead to greater demand and wear of the pitch actuation mechanisms. Additionally, the FF control implementation into the WindCrete was found to be more problematic than the implementation into the VolturnUS-S. This was due to the WindCrete's slow rate of FB pitch control actuation, with limits required to be applied to the FFCPC and FFIPC commands in close-to-rated wind speeds to avoid the introduction of aerodynamic negative damping and instability of the FOWT. Furthermore, the WindCrete necessitated the addition of a gain applied on the individual pitch commands to reduce their magnitude, which was required in order to avoid platform heave excitation. Ultimately, it was concluded that the implementation of LAC is possible for different FOWTs, but is required to differ depending on the specific FB pitch control and platform motion behaviour of each FOWT.

The results of the OpenFAST simulations are considered to be the best-case scenario of what may be achieved from the implementation of the outlined controllers. This is because the motions of the nacelle were not fully coupled to the LIDAR device within the simulator (which otherwise reduces the wind preview quality and restricts the FF control action), a wind field reconstruction was not performed, 100% LIDAR availability was assumed and blade blockage was not considered.

9.2.2 O&M case study

The loading reductions of the VolturnUS-S using FFCPC+FFIPC control were translated to reductions in component failure rates using NREL's loads post-processor, MLife. Failure rate reductions based on OpenFAST outputs were translated to failure rate reductions of turbine components. The decreased failure rates were used within the StrathO&M cost model, where a case study was performed (in Chapter 6) on different sites awarded within the recent ScotWind leasing round. The implementation of LAC indicated the ability to deliver reductions in the failure rate of components in the rotor nacelle assembly by 20%, the tower base by up to 30% and the pitch bearings and blade roots by 46%, relative to the baseline FB-only controller.

When considering only the failure rate reductions applied to the rotor-nacelle assembly (RNA), these translated to OpEx savings of up to 5% (compared to when the baseline failure rates, pertaining to FB-only control, were applied within the O&M model) due to reduced maintenance activities as a result of component lifetime extension. Results were also found to vary depending on the location of the wind farm, with sites with higher mean wind speeds showing higher potential for cost savings due to a greater proportion of the turbine's lifetimes being

spent with LAC active in above-rated wind speeds. FOW sites most impacted by the addition of workability limits, which impose access constraints based on the met-ocean conditions' impacts on the floating platforms' motions, saw the largest OpEx savings from LAC integration (by up to 11%) due to the reduction in downtime and lost revenue. This was particularly evident for the site situated furthest from shore, as the restricted accessibility amplified the lengths of downtime and the associated lost revenue. For specific designs and sites, the overall OpEx for FOW sites with motion limits and the failure rate reductions delivered by LAC were found to be more cost-effective than the equivalent bottom-fixed offshore wind site using FB-only control, by up to 5%.

Overall, the O&M case study highlighted the potential OpEx benefits from the introduction of a LIDAR control strategy for future FOW developments. However, the study was limited by the availability of representative failure rate data of large-scale (15 MW+) FOWTs and MLife's inability to capture the failure of the pitch systems resulting from wear of the mechanical systems due to actuating, which was slightly increased when using FFCPC+FFIPC.

9.2.3 Tower design case study

Simulations of the VolturnUS-S using LAC and with modified tower geometry were performed using the WEIS and WISDEM structural design tools (Chapter 7). This was enabled by implementing the modifications performed to the source codes of OpenFAST and ROSCO within the source code of WEIS/WISDEM. The study intended to investigate the benefit of tower load reduction and rotor speed regulation improvements for FOWT tower design. It performed a preliminary analysis of how the reductions to the loadings, motions, and rotor speed variations brought by the implementation of LAC to the VolturnUS-S can lead to reduced tower mass of the FOWT. Various levels of percentage reductions were applied to the baseline tower design's outer diameter or shell thickness. These reductions caused the natural frequencies of the tower to reduce towards the 3P blade passing frequency band. It was found that greater reductions to the tower shell thickness could be undertaken before the tower frequencies entered the 3P band (50% compared to a 20% reduction relative to the baseline tower outer diameter). Super-lightweight tower designs (70% reduction to the shell thickness or 35% reduction to the outer diameter relative to the baseline tower), whose natural frequencies lay within the 3P band, caused resonance with the rotor rotation and increased loadings at below-rated wind speeds, whilst also greatly increasing the nacelle acceleration and tower-top displacement due to the increased flexibility of the tower. It was therefore concluded that towers with first natural frequencies either within the 3P band or within the 1P and 3P interval were not viable for the FOWT that was studied.

It was found that, following the initial screening of proposed modified tower designs under DLC 1.2 with a mean wind speed of 13 ms^{-1} and co-directional wind and waves, the increased stress on the towers due to their reduced diameter or thickness could be offset somewhat by the addition of LAC. This allowed for the tower outer diameter to be reduced by 6.5% or the tower shell thickness by 20% while achieving similar levels of stress to the baseline tower supporting an FB-only controlled FOWT. The 20% reduced shell thickness design was termed the lightweight tower and analysed further.

Firstly, ultimate loads were applied to the lightweight tower to verify whether it could withstand them even in conditions when LAC was not available, such as in below-rated or above-cut-out wind speeds. The maximal stresses experienced by the lightweight design were found to be within the prescribed limits.

Secondly, the lightweight tower supporting a LIDAR-assisted-controlled FOWT and the baseline tower supporting an FB-only-controlled FOWT were subjected to wind-wave misalignment. This aimed to ascertain the impact of the greater tower side-to-side loads occurring due to wind-wave misalignment (which the LAC strategy was ineffective at reducing) on the lightweight design. The results of the wind-wave misalignment study found that the standard deviation of the design equivalent stress on the tower base was less (by up to 6%) for the lightweight tower supporting a LIDAR-assisted-controlled FOWT than the baseline tower supporting a FB-only-controlled FOWT at all wind-wave misalignment angles in low above-rated wind speeds. However, at high above-rated wind speeds, the standard deviation of the design equivalent stress on the tower base was greater (by up to 30%) in the lightweight tower supporting a LIDAR-assisted-controlled FOWT than the baseline tower supporting a FB-only-controlled FOWT at all wind-wave misalignment angles, with the highest levels of stress occurring when the wind and wave directions were perpendicular. The benefit of LAC to the lightweight design deteriorated with increasing wind speed wind-wave misalignment angles for two reasons. Firstly, this was because the side-to-side loads increased with wind speed (its standard deviation doubled between mean wind speeds of 13 ms^{-1} and 21 ms^{-1}) and wind-wave misalignment (its standard deviation increased by up to 74% at 120° misalignment relative to 0° misalignment). Secondly, the LAC strategy wasn't able to reduce the stress on the lightweight tower at high wind speeds because its reduction to the standard deviation of the tower base fore-aft bending moment diminished with increased wind speeds, by up to 14% between mean wind speeds of 13 ms^{-1} and 21 ms^{-1}. As the fatigue and stress benefit to the tower delivered by LAC was heavily driven by reductions to the tower base fore-aft loads, it was unsurprising that it was unable to reduce the stress on the tower significantly at greater wind speeds.

Given that, generally, high wind speeds close to the cut-out wind speed and/or high degrees of wave-misalignment occur significantly less frequently than low above-rated wind speeds and/or co-directional wind and wave conditions, it was pertinent to apply a joint probability density function to weight the tower stresses according to the wind speed and wind-wave misalignment probabilities. A joint probability density function was determined by the combination of a Weibull distribution of wind speeds of a prospective FOW farm site and a normal distribution used to model the probabilities of wind-wave misalignment angles.

By assessment of the total weighted stress across all wind-wave misalignment angles bins at each wind speed bin, the results showed that the stress on the lightweight tower supporting a LIDAR-assisted-controlled FOWT was up to 19% less than that of the baseline tower supporting a FB-only-controlled FOWT at low above-rated wind speeds (11 ms^{-1} to 15 ms^{-1}). Meanwhile, the stress on the lightweight tower supporting a LIDAR-assisted-controlled FOWT was up to 9.8% greater than that of the baseline tower supporting a FB-only-controlled FOWT at high above-rated wind speeds (19 ms^{-1} to 25 ms^{-1}). However, the total joint-probability weighted stress variation reductions brought by LAC at low above-rated wind speeds (0.921 MPa) vastly outweighed the detrimental increases in the stress at high above-rated wind speeds (0.091 MPa). This was due to the significantly greater probabilities of occurrence of low above-rated wind speed and low/no wind-wave misalignment combinations (where LAC was able to deliver the greatest tower loading reductions) than high above-rated wind speed and high wind-wave misalignment combinations. Similarly, by assessment of the total weighted stress across all wind speed bins at each wind-wave misalignment angle bin, the results showed that the stress reduction on the lightweight tower supporting a LIDAR-assisted-controlled FOWT over that of the baseline tower supporting a FB-only-controlled FOWT was greatest (by up to 7.3%) at low/no wind-wave misalignment angles (-30° to +30°). As the wind-wave misalignments increased, the stress reduction on the lightweight tower supporting a LIDAR-assisted-controlled FOWT over that of the baseline tower supporting an FB-only-controlled FOWT diminished. Yet, the total weighted stress on the lightweight tower supporting a LIDAR-assisted-controlled FOWT did not become greater than that of the baseline tower supporting an FB-only-controlled FOWT at any wind-wave misalignment angle (when considering all wind speeds at each wind-wave misalignment angle bin). This was because the load reductions at the high-probability low above-rated wind speeds were consistent across all wind-wave misalignment angles, meaning the stress reductions that this brought vastly outweighed the increases in the stress at low-probability high above-rated wind speeds and high wind-wave misalignment combinations. The total stress across all of the above-rated mean wind speed and wind-wave misalignment angle bin combinations on the lightweight tower supporting a LIDAR-assisted-controlled FOWT was 9.1% less than that of the baseline tower supporting an FB-only-controlled FOWT, indicating scope for further optimisation of towers supporting LIDAR-assisted-controlled FOWTs

The results of the tower design case study provide compelling evidence for the viability of lightweight towers supporting LIDAR-assisted controlled FOWTs. The loads assessment, covering normal and extreme operating conditions and wind-wave misalignment scenarios strongly indicates that the lightweight tower supporting the LIDAR-assisted controlled FOWT can experience similar stress to that of the baseline tower supporting the FB-only controlled FOWT. Consequently, the use of lightweight towers supporting LIDAR-assisted controlled FOWTs can reduce their steel material usage, by 20% per tower, and, therefore, their cost. Furthermore, lighter towers will also incur lesser water usage and CO_2 emissions during their manufacture, further justifying their use.

Nonetheless, despite the results presented, the tower design case study was also not without limitations. These include that changes in the coherence between the LIDAR-measured and turbine-experienced REWSs with changing tower geometry were assumed to be negligible. The changes to the tower may have deteriorated the LIDAR preview quality due to the increased flexibility of the towers and subsequent increased nacelle motions. Had this effect been considered, lower cutoff frequencies of the low-pass filter of the LIDAR measurements would have been applied, which would have restricted the control action. Furthermore, as only a preliminary design case study was performed, further work is required, considering greater wind and wave condition combinations in addition to a more accurate joint wind-wave probability density distribution analysis, to ascertain the lifetime fatigue of the lightweight tower design using LAC.

9.2.4 Array-level case study

Simulations of the VolturnUS-S were extended to the array level (within Chapter 8) in order to investigate the benefits that LAC may bring to the performance of downstream FOWTs within a wind farm. The increases seen in the standard deviations of properties of downstream turbines relative to the first turbine when using FB-only control were able to be offset by the addition of LAC. This can ensure that downstream turbines are less prone to fluctuating loads and fatigue. Furthermore, it was found that the addition of LAC presents notable benefits to the wake deficit behaviour of FOWTs, by reductions to the maximum deficit by up to 8% and the standard deviation by up to 16% at a mean ambient wind speed of 13 ms^{-1}. This ultimately led to increases in the mean power of FOWTs using LAC that were, importantly, downstream of a FOWT using LAC, by up to 1.4% (relative to when they used FB-only control) at ambient wind speeds close to the rated wind speed when turbines were spaced eight rotor diameters apart. This offers the potential to increase the annual energy production of downstream turbines (relative to turbines using FB-only control) by approximately 0.8% and therefore increase revenues, which may lead to a lower LCOE of FOW.

Although the results show promise, some overestimation may have occurred because the preview quality of the LIDAR measurement was assumed to be the same for each turbine, thereby neglecting the impact of waked inflow on the preview quality, the cutoff frequency of the low pass filter and the FF control action. Furthermore, the array-level case study only considered aligned turbines and did not consider wind farm control strategies such as wake steering.

The benefits reported from the array-level case study were only applicable to the VolturnUS-S as the LIDAR-assisted controllers were inactive within the WindCrete in wind speeds close to the rated wind speed. This highlights a further discrepancy when considering the implementation of LAC into different FOWT platform designs.

Overall, the results have been focused on providing quantification of the cost-saving benefits of LAC for the FOW sector, while also noting the environmental benefits that can result. This thesis has provided strong evidence of how the implementation of LAC can address the prominent challenges faced by the FOW industry with regard to financing, design and maintenance. The simulation results of 15 MW turbines are especially relevant to the current industry landscape and may remain so for an extended period of time due to the concerns over the practicality of further increasing the sizes of offshore wind turbines. It is believed that the large-scale implementation of LAC on FOWTs can deliver valuable reductions to the LCOE of FOW power for the reasons outlined within the case studies presented in this work, which can aid in making projects more commercially viable.

9.3 Opportunities for future work

This thesis has highlighted the potential cost-saving benefits of large-scale LAC deployment. However, the work is not without its limitations, which may be addressed in future work. Further work is required in order to address the limitations of the InflowWind LIDAR simulator and the FFIPC control strategy, as well as to expand the field of research into new areas, with the goal of promoting large-scale deployment of LAC.

9.3.1 LIDAR simulator and feedforward individual pitch controller enhancements

The first aspect of further work should address the potential degree of overestimation of the LIDAR preview quality from the OpenFAST InflowWind LIDAR simulator resulting from the omission of the LIDAR velocities, accelerations and rotations. This can be performed by comparing results to an alternative simulator that includes these features, such as that developed by Schlipf et al. (2009) and Guo et al. (2022). The motion compensation algorithm used within the Schlipf et al. (2009) and Guo et al. (2022) simulator may also be implemented

into the OpenFAST InflowWind LIDAR simulator in order to remove the contribution of the LIDAR device's velocity from the LOS wind speed measurements. The preview quality can be affected by the linear velocity of the LIDAR, which will be included within the Doppler effect, highlighting the need for a motion compensation algorithm to remove the contribution of the LIDAR velocity from the LOS wind speed measurement. However, within the work presented, the velocities of the nacelle, and by extension the LIDAR device and beams, were not coupled, and so it is not anticipated that the wind speed measurements were affected. However, this neglects errors that may be induced by an imperfect motion compensation algorithm. Thus, in an effort to improve the realism of the simulator, the LIDAR device and beams should be fully coupled with the motions of the nacelle, with a motion compensation algorithm subsequently implemented. The simulator can be greatly enhanced by enabling the use of multiple pulsed or continuous-wave beams, instead of just multiple single-point beams. This would result in a more realistic LIDAR simulation because the pulsed and CW beams would entail the use of a weighting function to spatially average the LOS wind velocity measurement across the probe volume. The LOS velocities measured by the angled beams would also have to be corrected to remove the contributions from the lateral and vertical wind components, ensuring isolation of the longitudinal component. Additionally, a wind field reconstruction method would be necessary to estimate the rotor-effective wind speed. The FFIPC may also need to be reconfigured somewhat to ensure that 2-beam pairs, instead of single beams, are used for determining the azmiuth-specific wind velocities used by the FFIPC, helping to avoid errors emerging from the Cyclops dilemma, whereby single pulsed or CW beams cannot distinguish between horizontal wind shear and wind direction changes.

Further work may also consider the use of 4D turbulence files, which do not assume temporal and spatial coherence, within LAC simulations, as was performed by Guo, Schlipf, and Cheng (2023) in order to capture the impacts of removing the Taylor's frozen hypothesis assumption, and give better estimations of the feedforward control performance in real-life applications.

Further enhancements to the OpenFAST LIDAR simulator can include the consideration of blade blockage impacts on the LIDAR beams. This could be relatively simple to implement by setting the LIDAR outputs to zero when the azimuth of a blade is at an angle that will cause the beams to be obstructed. Finally, the LIDAR simulator is currently only configured to receive data from binary TurbSim (`.bts`) files and so LIDAR measurements are only recorded (and therefore LAC is only possible) when simulations under turbulent wind conditions are performed. Therefore, further work may seek to ensure compatibility of the LIDAR simulator with the other InflowWind module inflow settings to enable assessment of the LAC performance under wind events such as gusts, low-level jets and wind direction changes.

Further development of the FFIPC strategy may seek to quantify the impact of the estimated 4-8% misassignment of the LIDAR beams to the blades, and subsequent FFIPC commands, resulting from the errors in the blade azimuth predictions when assuming constant rotor rotational speed over the delay period incurred by the low-pass filter applied to the FFIPC commands. The FFIPC strategy could be enhanced or corrected to ensure that the beam-blade misassignments could be reduced or eliminated entirely. The resulting improvements to the results could then be investigated.

The cutoff frequency of the low-pass filter applied to each beam's wind speed measurement may also be re-evaluated by determining if the blade-effective wind speed provides a superior estimate of the wind speeds impacting upon the individual blades than the rotor-effective wind speed. This can be undertaken by evaluation of the transfer function (i.e. Eq. 4.22) between the LIDAR-measured wind speeds at the different rotor azimuth points and the actual wind speeds impacting upon the rotor at those points. Should this indicate a superior LIDAR preview quality, the cutoff frequency of the low-pass filter applied to each beam's wind speed measurement may be increased such that the FFIPC control action is less restricted, which may allow for improvements to the results.

The results presented assume 100% LIDAR availability and, in addition to the other limitations discussed, therefore should be considered to be the best-case scenario for the benefits delivered by this LAC approach. Further work should seek to perform sensitivity analysis to investigate the impact of reduced LIDAR availability and LAC benefit on the failure rates and cost savings. When considering reduced LIDAR availability due to factors such as contamination, aerosol concentration and hardware malfunction, fallback strategies may need to be implemented to ensure that turbine stability is maintained. Such strategies may involve phasing out or exclusion of the feedforward commands, thereby reverting the controller to the FB-only mode. Another strategy may employ a wind estimation model to approximate the incoming wind speed and generate contingency feedforward commands in cases where the LIDAR is unavailable.. Further work may also seek to address how nacelle-mounted LIDAR wind measurement can help in understanding the loadings on FOWTs in their inactive (feathered blades) state.

9.3.2 Fatigue estimations

Concerning the O&M case study, a finite element analysis (FEA) with LAC should be performed to give a more thorough indication of the loading reductions and extended lifetimes of components of complex geometry, whose fatigue cannot be estimated effectively with an S-N curve. Further work may also consider a joint wind-wave probability density distribution to determine the lifetime benefits on components such as the tower and mooring system, whose

fatigue are subject to the directionality of wind and wave loading. The failure rate reductions delivered by LAC to different FOWT platform designs should also be categorised (alongside consideration of their WI limitations) and applied within an O&M simulation model in order to better determine how the OpEx can be reduced for different FOWT designs.

9.3.3 Structural design optimisation

Future work is required to address the assumptions and shortcomings of the methodologies used in undertaking the tower design case study. Firstly, simulations capturing a greater number of combinations of mean wind speeds, significant wave heights and peak spectral wave periods and extreme tower load cases, such as DLCs 2.3, 3.2, 4.2 and 5.1, should be performed. Secondly, in furthering the investigation into wind-wave misalignment scenarios, work should consider cases of separate wind and wave direction distributions to more accurately characterise the fatigue damage accumulation of the tower. This is because only a fixed wind direction of 0° was assumed in the tower design case study. Thirdly, a more accurate joint probability density function of wind speed and wave misalignment conditions should be considered. In particular, the wave misalignment probability density function should be determined based on a Von Mises distribution of measured site-specific met-ocean data.

Finally, the tower design case study should be expanded to better capture the benefits of LAC (or lack thereof) for key constraints of FOWT tower design. Primarily, further work should focus on performing a more detailed structural analysis of the proposed lightweight towers using an FEA model. This can enable an assessment of global and local buckling, which forms a design constraint when the tower shell thickness is significantly reduced. Further work should seek to verify whether or not the proposed lightweight tower designs (supporting turbines using either LAC or FB-only control) breach the buckling constraints. If breaches to the buckling constraints of the higher-risk higher diameter-to-thickness ratio sections of the proposed lightweight tower are found to occur, then the shell thickness reductions can be applied in a different manner. For example, instead of the equal percentage shell thickness reductions applied to each section of the tower within the tower design case study, further work could apply different percentage reductions to separate sections (or groups of sections) of the tower. Larger reductions could be applied to thicker lower-risk-of-buckling sections of the baseline tower while applying lesser or no reductions to the thinner high-risk-of-buckling sections.

In order to reap more cost-saving benefits, the structural optimisation of FOWTs implementing LAC may be extended to the floating substructure. This may result in significant cost savings due to reduced material usage. To perform this task (when making incremental adjustments to design variables, as in Chapter 7), a panel mesh of the baseline and modified VolturnUS-S platform should be created and input to an external hydrodynamic wave-body interaction program such as WAMIT, developed by C. H. Lee and Newman (2006). This will then use the

three-dimensional numerical-panel method in the frequency domain to solve the linearized hydrodynamic radiation and diffraction problems for the interaction of surface waves with the platforms. Alternatively, this can be achieved using pyHAMS within WEIS, but only as part of the OpenMDAO optimisation framework, which is applicable when undertaking a co-design optimisation approach. Importantly, these methods generate the hydrodynamic coefficient files (.1, .3, .hst) required by HydroDyn to model the wave interactions of the platform. These hydrodynamic coefficient files are unique to each substructure geometry and must be generated for each new design. Simulations should be performed using different platform designs, with and without LAC enabled, to ascertain whether the benefits of LAC are able to offset any potential deteriorations that may emerge in the performance, motions and/or loadings of the FOWT. Structural optimisation of the floating platform may also be coupled with that of the tower in order to converge on an optimal design of FOWTs implemented with LAC.

A co-design optimisation study may also be performed, considering the wider impacts of LAC for the integrated design of the tower, floating platform and moorings. Further work may also perform sensitivity analysis accounting for potential downtime of the LIDAR device as well as assess the impacts of the loads on the towers experienced during the towing process of the FOWT installation.

9.3.4 Array-level simulation

The work performed on the implementation of LIDAR simulation and LAC at the wind farm level may also be greatly expanded. Although the results in Chapter 8 highlighted the potential for LAC to reduce the power deficit of downstream turbines due to improved wake recovery, numerous opportunities were identified to expand the research and consolidate the findings. Firstly, a broader suite of DLCs should be simulated, to investigate the effects of extreme conditions, gusts, wind direction changes and wind and wave misalignments. As all of the turbines used the same control strategy within a simulation, either LAC or FB-only control, it would be beneficial to investigate the differences in the results that emerge when only the first turbine employs LAC, for example. As the array-level study only considered aligned turbines, further work may also seek to assess the differences in results obtained from staggered (in the y direction) turbines, and investigate how the reduced wake brought by such an arrangment impacts upon the LAC performance of the downstream turbines, as well as turbines with yaw misalignment. Additional work is also required to examine the efficacy of LIDAR measurement under waked inflow, and how this impacts upon the preview quality, the coherence and, ultimately, the controller performance. The simulation results should also be validated by comparison to results obtained within a computational fluid dynamics (CFD) tool, which is able to model the wake behaviour to a much higher degree of fidelity than FAST.Farm (though at the expense of considerably greater computational demand).

9.3.5 Field testing of LIDAR-assisted control

Finally, opportunities for the deployment of LIDAR technology on FOWTs for field testing of LAC should continue to be pursued. Given the relative scarcity of FOWT deployments active during the timescales of this research project, it was challenging to fulfil potential LIDAR deployments on FOWTs. This was exacerbated by challenges of financing, commercial sensitivity and logistics for the installation of the technology. Given the projected growth of the FOW industry, it is expected that more opportunities to deploy and test LAC on commercial and demonstration FOW projects will emerge. The data acquired from these will be invaluable for validating numerical models and providing greater evidence to reinforce the argument for large-scale deployment of LIDAR technology on FOWTs.

Bibliography

Abbas, N. J., Wright, A., & Pao, L. (2020). An update to the National Renewable Energy Laboratory baseline wind turbine controller. In *Journal of Physics: Conference Series* (Vol. 1452, p. 012002). doi: 10.1088/1742-6596/1452/1/012002

Abbas, N. J., Zalkind, D. S., Pao, L., & Wright, A. (2022). A reference open-source controller for fixed and floating offshore wind turbines. *Wind Energy Science*, *7*(1), 53–73. doi: 10.5194/wes-7-53-2022

ACTEON. (2020). Floating wind: What are the mooring options? - A Q&A with Kent Longridge. (URL: `https://acteon.com/blog/floating-wind-mooring-options/`)

Ahn, D., Shin, S. C., Kim, S. Y., Kharoufi, H., & Kim, H. (2017). Comparative evaluation of different offshore wind turbine installation vessels for Korean west–south wind farm. *International Journal of Naval Architecture and Ocean Engineering*, *9*(1), 45–54. doi: 10.1016/j.ijnaoe.2016.07.004

Allen, C., Viscelli, A., Dagher, H., Goupee, A., Gaertner, E., Abbas, N., ... Barter, G. (2020). Definition of the UMaine VolturnUS-S reference platform developed for the IEA-Wind 15 megawatt offshore reference wind turbine. Tech. Rep. NREL/TP-5000-76773, NREL, Golden, CO (US). doi: 10.2172/1660012

Bao, J., Yue, H., Leithead, W., & Wang, J. (2016). LIDAR-assisted wind turbine gain scheduling control for load reduction. In *2016 22nd International Conference on Automation and Computing (ICAC)* (pp. 15–20). doi: 10.1109/IConAC.2016.7604887

Barbosa, J. F., Correia, J. A., R., F. J., Zhu, S.-P., & De Jesus, A. M. (2019). Probabilistic sn fields based on statistical distributions applied to metallic and composite materials: State of the art. *Advances in Mechanical Engineering*, *11*(8), 1687814019870395. doi: 10.1177/1687814019870395

Barj, L., Jonkman, J. M., Robertson, A., Stewart, G., Lackner, M., Haid, L., ... Stewart, S. (2014). Wind/wave misalignment in the loads analysis of a floating offshore wind turbine. In *32nd ASME Wind Energy Symposium* (p. 0363). doi: 10.2514/6.2014-0363

Barter, G. (2023). WISDEM/WEIS: IEA-15-240-RWT_VolturnUS-S.yaml. (URL: `https://github.com/WISDEM/WEIS/blob/main/examples/06_IEA-15-240-RWT/IEA-15-240-RWT_VolturnUS-S.yaml`)

Barter, G. (2024a). WISDEM/pyHAMS. (URL: `https://github.com/WISDEM/pyHAMS?tab=readme-ov-file`)

Barter, G. (2024b). WISDEM/RAFT. (URL: `https://github.com/WISDEM/RAFT`)

Bashetty, S., & Ozcelik, S. (2021). Review on dynamics of offshore floating wind turbine platforms. *Energies*, *14*(19), 6026. doi: doi.org/10.3390/en14196026

Bernard, V., Andersen, S., Leon, J. M., Beaudet, L., Verelst, D., & Iliopoulos, A. (2024). Observation and modelling of asymmetric loading on large offshore wind turbines in wake conditions. In *Journal of Physics: Conference Series* (Vol. 2767, p. 092092). doi: 10.1088/1742-6596/2767/9/092092

Blackburne, A. (2023). Race for ever-larger wind turbines might be unsustainable. (URL: `https://www.spglobal.com/marketintelligence/en/news-insights/latest-news-headlines/race-for-ever-larger-wind-turbines-might-be-unsustainable-75519950`)

Bossanyi, E. A. (2003). Individual blade pitch control for load reduction. *Wind Energy*, *6*(2), 119–128. doi: 10.1002/we.76

BS EN 1993-1-1:2005 Eurocode 3: Design of steel structures. General rules and rules for buildings, BSI. (2005). Tech. Rep. British Standards Institution, London, UK.

Burton, T., Jenkins, N., Sharpe, D., & Bossanyi, E. (2011). *Wind energy handbook*. John Wiley & Sons.

BVG. (2023). Guide to a floating offshore wind farm. (URL: `https://guidetofloatingoffshorewind.com/wind-farm-costs/`)

BW Ideol. (2018). FLOATGEN. (URL: `https://www.bw-ideol.com/en/floatgen-demonstrator`)

Carmo, L., Jonkman, J., & Thedin, R. (2024). Investigating the interactions between wakes and floating wind turbines using fast. farm. *Wind Energy Science*, *9*(9), 1827–1847. doi: 10.5194/wes-9-1827-2024

Carroll, J., McDonald, A., & McMillan, D. (2016). Failure rate, repair time and unscheduled O&M cost analysis of offshore wind turbines. *Wind Energy*, *19*(6), 1107–1119. doi: 10.1002/we.1887

Choi, N., Nam, S., Jeong, J., & Kim, K. (2014). CFD study on aerodynamic power output changes with inter-turbine spacing variation for a 6 MW offshore wind farm. *Energies*, *7*(11), 7483–7498. doi: 10.3390/en7117483

Clifton, A., Boquet, M., Burin Des Roziers, E., Westerhellweg, A., Hofsass, M., Klaas, T., ... others (2015). *Remote sensing of complex flows by doppler wind lidar: Issues and preliminary recommendations* (Tech. Rep.).

Clive, S., O'Brien, P., & O'Connor, V. (2022). ScotWind 1 leasing round: Scottish clusters membership briefing. Offshore Wind Scotland. (URL: https://www.offshorewindscotland.org.uk/media/12675/deepwind-scotwind-round-one-briefing.pdf)

CoreWind. (2020). Floating wind O&M strategies assessment. (URL: https://corewind.eu/wp-content/uploads/files/publications/COREWIND-D4.2-Floating-Wind-O-and-M-Strategies-Assessment.pdf)

Crown Estate Scotland. (2022). Awards: Lead applicants, project partners, area, capacity and foundations. (URL: https://www.crownestatescotland.com/resources/documents/scotwind-list-of-successful-project-partners)

CRU Group. (2021). Offshore wind and steel: a huge mutual dependency. (URL: https://www.crugroup.com/knowledge-and-insights/insights/2021/offshore-wind-and-steel-a-huge-mutual-dependency/)

Dao, C., Kazemtabrizi, B., & Crabtree, C. (2019). Wind turbine reliability data review and impacts on levelised cost of energy. *Wind Energy*, *22*(12), 1848–1871. doi: 10.1002/we.2404

Department for Energy Security and Net Zero. (2023a). Contracts for Difference. *H.M Government*. (URL: https://www.gov.uk/government/collections/contracts-for-difference)

Department for Energy Security and Net Zero. (2023b). Net zero government initiative. *H.M Government*. (URL: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/1167856/offshore-wind-investment-roadmap)

Díaz, H., & Soares, C. G. (2020). Review of the current status, technology and future trends of offshore wind farms. *Ocean Engineering*, *209*, 107381. doi: 10.1016/j.oceaneng.2020.107381

Dinwoodie, I., Endrerud, O.-E. V., Hofmann, M., Martin, R., & Sperstad, I. B. (2015). Reference cases for verification of operation and maintenance simulation models for offshore wind farms. *Wind Engineering*, *39*(1), 1–14. doi: 10.1260/0309-524X.39.1.1

Dinwoodie, I., Quail, F., & McMillan, D. (2012). Analysis of offshore wind turbine operation and maintenance using a novel time domain meteo-ocean modeling approach. In *Turbo Expo: Power for Land, Sea, and Air* (Vol. 44724, pp. 847–857). doi: 10.1115/GT2012-68985

DNV GL. (2019). Goldwind receives world's first certification for LiDAR Assisted Control technology by DNV GL. (URL: https://www.dnv.com/news/goldwind-receives-world-s-first-certification-for-lidar-assisted-control-technology-by-dnv-gl-161205/)

DNV GL. (2021). DNV-RP-0286: Coupled analysis of floating wind turbines. (URL: https://www.dnv.com/energy/standards-guidelines/dnv-rp-0286-coupled-analysis-of-floating-wind-turbines/)

Dunne, F., & Pao, L. (2013). Benefit of wind turbine preview control as a function of measurement coherence and preview time. In *2013 American Control Conference* (pp. 647–652). doi: 10.1109/ACC.2013.6579910

Dunne, F., Pao, L., Wright, A., Jonkman, B., & Kelley, N. (2010). Combining standard feedback controllers with feedforward blade pitch control for load mitigation in wind turbines. In *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition* (p. 250). doi: 10.2514/6.2010-250

Dunne, F., Pao, L., Wright, A., Jonkman, B., Kelley, N., & Simley, E. (2011). Adding feedforward blade pitch control for load mitigation in wind turbines: Non-causal series expansion, preview control, and optimized FIR filter methods. In *49th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition* (p. 819). doi: 10.2514/6.2011-819

Dunne, F., Schlipf, D., Pao, L., Wright, A., Jonkman, B., Kelley, N., & Simley, E. (2012). Comparison of two independent LIDAR-based pitch control designs. In *50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition* (p. 1151). doi: 10.2514/6.2012-1151

Dunne, F., Simley, E., & Pao, L. Y. (2011). LIDAR wind speed measurement analysis and feedforward blade pitch control for load mitigation in wind turbines: January 2010–January 2011. Tech. Rep. NREL/SR-5000-52098, NREL, Golden, CO.

Durakovic, A. (2023). 22 MW offshore wind turbine in the works for 2024/25. (URL: https://www.offshorewind.biz/2023/10/23/22-mw-offshore-wind-turbine-in-the-works-for-2024-25/)

Energy Numbers. (2022). UK offshore wind capacity factors. (URL: https://energynumbers.info/uk-offshore-wind-capacity-factors)

Feng, Y., Tavner, P. J., & Long, H. (2010). Early experiences with UK round 1 offshore wind farms. *Proceedings of the Institution of Civil Engineers-Energy*, *163*(4), 167–181. doi: doi.org/10.1680/ener.2010.163.4.167

Fischer, B. (2013). Reducing rotor speed variations of floating wind turbines by compensation of non-minimum phase zeros. *IET Renewable Power Generation*, *7*(4), 413–419. doi: 10.1049/iet-rpg.2012.0263

Fleming, P., Annoni, J., Shah, J. J., Wang, L., Ananthan, S., Zhang, Z., ... Chen, L. (2017). Field test of wake steering at an offshore wind farm. *Wind Energy Science*, *2*(1), 229–239. doi: 10.5194/wes-2-229-2017

Fleming, P., Gebraad, P. M., Lee, S., van Wingerden, J.-W., Johnson, K., Churchfield, M., ... Moriarty, P. (2015). Simulation comparison of wake mitigation control strategies for a two-turbine case. *Wind Energy*, *18*(12), 2135–2143. doi: 10.1002/we.1810

Fleming, P., Pineda, I., Rossetti, M., Wright, A., & Arora, D. (2014). Evaluating methods for control of an offshore floating turbine. In *International Conference on Offshore Mechanics and Arctic Engineering* (Vol. 45547, p. V09BT09A019).

Flotation Energy and Vårgrønn. (2024). Green Volt floating offshore wind farm. (URL: `https://greenvoltoffshorewind.com/`)

Fu, W., Guo, F., Schlipf, D., & Peña, A. (2023). Feedforward pitch control for a 15 MW wind turbine using a spinner-mounted single-beam lidar. *Wind Energy Science Discussions*, *2023*, 1–23. doi: 10.5194/wes-8-1893-2023

Fu, W., Sebastiani, A., Peña, A., & Mann, J. (2022). Dependence of turbulence estimations on nacelle-lidar scanning strategies. *Wind Energy Science Discussions*, *2022*, 1–21. doi: 10.5194/wes-8-677-2023

Gaertner, E., Rinker, J., Sethuraman, L., Zahle, F., Anderson, B., Barter, G. E., ... others (2020). Definition of the IEA-Wind 15 megawatt offshore reference wind turbine. Tech. Rep. NREL/TP-5000-75698, NREL, Golden, CO. doi: 10.2172/1660012

Gray, J. S., Hwang, J. T., Martins, J. R., Moore, K. T., & Naylor, B. A. (2019). OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization. *Structural and Multidisciplinary Optimization*, *59*, 1075–1104. doi: 10.1007/s00158-019-02211-z

Guo, F., & Schlipf, D. (2023). Assessing LIDAR-assisted feedforward and multivariable feedback controls for large floating wind turbines. *Wind Energy Science Discussions*, *2023*, 1–25. doi: 10.5194/wes-2023-9

Guo, F., Schlipf, D., & Cheng, P. (2023). Evaluation of LIDAR-assisted wind turbine control under various turbulence characteristics. *Wind Energy Science*, *8*(2), 149–171. doi: 10.5194/wes-8-149-2023

Guo, F., Schlipf, D., Lemmer, F., Raach, S., Özinan, U., Adam, R., & Choisnet, T. (2023). The performance of two control strategies for floating wind turbines: lidar-assisted feedforward and multi-variable feedback. In *Journal of Physics: Conference Series* (Vol. 2626, p. 012005). doi: 10.1088/1742-6596/2626/1/012005

Guo, F., Schlipf, D., Zhu, H., Platt, A., Cheng, P. W., & Thomas, F. (2022). Updates on the OpenFAST LIDAR simulator. In *Journal of Physics: Conference Series* (Vol. 2265, p. 042030). doi: 10.1088/1742-6596/2265/4/042030

Hansen, M. (2015). *Aerodynamics of wind turbines*. Routledge.

Hayman, G. (2012). MLife theory manual for version 1.00. Tech. Rep. NREL, Golden, CO (US).

Held, D. P., & Mann, J. (2019). LIDAR estimation of rotor-effective wind speed-an experimental comparison. *Wind Energy Science*, *4*(3), 421–438. doi: 10.5194/wes-4-421-2019

Henderson, A., Collu, M., & Masciola, M. (2016). Floating offshore wind energy: The next generation of wind energy. *Springer*, *15*(86).

Henderson, S. (2023). ZX Lidars and sowento embark on a new Lidar-Assisted Control initiative to support industry innovation. (URL: `https://www.zxlidars.com/zx-lidars-and-sowento-embark-on-a-new-lidar-assisted-control-initiative-to-support-industry-innovation/`)

HM Government. (2023). Offshore wind net zero investment roadmap. (URL: `https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/1167856/offshore-wind-investment-roadmap`)

IEC 61400-3-1:2019 - Wind energy generation systems - Part 3-1: Design Requirements for fixed offshore wind turbines. (2019). Tech. Rep. British Standards Institution, London, UK.

IEC 61400-1:2019 - Wind energy generation systems - Part 1: Design Requirements. (2019). Tech. Rep. British Standards Institution, London, UK, *2019*.

International Energy Agency (IEA)-Wind. (2022). IEA Wind TCP Task 52: Large-scale deployment of wind LIDAR. (URL: `https://iea-wind.org/task52/`)

Jenkins, B., Prothero, A., Collu, M., Carroll, J., McMillan, D., & McDonald, A. (2021). Limiting wave conditions for the safe maintenance of floating wind turbines. , *2018*(1), 012023. doi: 10.1088/1742-6596/2018/1/012023

Jonkman, B. (2014). TurbSim user's guide v2.00.00. *NREL*.

Jonkman, J. (2022). OpenFAST: An open-source tool for wind turbine physics-based engineering modeling. In *NAWAE/WindTech 2022*. (URL: https://drive.google.com/file/d/1bD5a6rRg6cCKht9Ar8AFJQ8YrI4-wsFe/view)

Jonkman, J., Butterfield, S., Musial, W., & Scott, G. (2009). Definition of a 5 MW reference wind turbine for offshore system development. Tech. Rep. NREL/TP-500-38060, NREL, Golden, CO (US). doi: 10.2172/947422

Jonkman, J., & Matha, D. (2011). Dynamics of offshore floating wind turbines—analysis of three concepts. *Wind Energy*, *14*(4), 557–569. doi: doi.org/10.1002/we.442

Jonkman, J., & Shaler, K. (2021). FAST.Farm user's guide and theory manual. National Renewable Energy Laboratory Golden, CO, USA.

Jonkman, J. M., Robertson, A., & Hayman, G. J. (2014). HydroDyn user's guide and theory manual. *NREL*.

Kaimal, J. C., Wyngaard, J., Izumi, Y., & Coté, O. (1972). Spectral characteristics of surface-layer turbulence. *Quarterly Journal of the Royal Meteorological Society*, *98*(417), 563–589. doi: 10.1002/qj.49709841707

Kelley, N. D., & Jonkman, B. J. (2005). Overview of the TurbSim stochastic inflow turbulence simulator. Tech. Rep. NREL/TP-500-36971, NREL, Golden, CO (US). doi: 10.2172/15020329

Laks, J., Pao, L., Simley, E., Wright, A., Kelley, N., & Jonkman, B. (2011). Model predictive control using preview measurements from LIDAR. In *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition* (p. 813). doi: 10.2514/6.2011-813

Larsen, G. C., Aagaard Madsen, H., & Bingöl, F. (2007). Dynamic wake meandering modeling. *Tech Rep., DTU Wind Energy*.

Larsen, T., & Hanson, T. (2007). A method to avoid negative damped low frequent tower vibrations for a floating, pitch controlled wind turbine. In *Journal of Physics: Conference Series* (Vol. 75, p. 012073). doi: 10.1088/1742-6596/75/1/012073

LAUTEC ESOX. (2024). ERA5 Data Point Map. (URL: https://esox.lautec.com/map/)

Lee, C. H., & Newman, J. N. (2006). WAMIT, inc - The state of the art in wave interaction analysis. (URL: https://www.wamit.com/)

Lee, D., Morf, M., & Friedlander, B. (1981). Recursive least squares ladder estimation algorithms. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *29*(3), 627–641. doi: 10.1109/TASSP.1981.1163587

Lee, S., Churchfield, M., Moriarty, P., Jonkman, J., & Michalakes, J. (2012). Atmospheric and wake turbulence impacts on wind turbine fatigue loadings. In *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition* (p. 540). doi: 10.2514/6.2012-540

Leithead, W., Leith, D., Hardan, F., & Markou, H. (2014). Global gain-scheduling control for variable speed wind turbines. In *1999 European Wind Energy Conference* (pp. 853–856).

Lescher, F., Zhao, J. Y., & Borne, P. (2005). Robust gain scheduling controller for pitch regulated variable speed wind turbine. *Studies in Informatics and Control*, *14*(4), 299.

Li, H., & Soares, C. G. (2022). Assessment of failure rates and reliability of floating offshore wind turbines. *Reliability Engineering & System Safety*, *228*, 108777. doi: doi.10.1016/j.ress.2022.108777

Low Carbon Contracts Company. (2024). Contracts for Difference (CfD). (URL: `https://www.lowcarboncontracts.uk/our-schemes/contracts-for-difference/`)

Magar, K. T., Balas, M. J., & Frost, S. (2015). Direct adaptive control for individual blade pitch control of wind turbines for load reduction. *Journal of Intelligent Material Systems and Structures*, *26*(12), 1564–1572. doi: 10.1177/1045389X14566527

Mahfouz, M., Salari, M., Hernandez, S., Vigara, F., Molins, C., Trubat, P., ... Pegalar-Jurado, A. (2020). Public design and FAST models of the two 15 MW floater-turbine concepts. COREWIND, Barcelona, Spain. doi: 10.5281/zenodo.4385727

Mann, J. (1994). The spatial structure of neutral atmospheric surface-layer turbulence. *Journal of fluid mechanics*, *273*, 141–168. doi: 10.1017/S0022112094001886

MathWorks. (2018). Understanding model predictive control, part 1: Why use MPC? (URL: `https://www.youtube.com/watch?v=8U0xiOkDcmw`)

MathWorks. (2023a). *cpsd.* (URL: `https://uk.mathworks.com/help/signal/ref/cpsd.html`)

MathWorks. (2023b). *pwelch.* (URL: `https://uk.mathworks.com/help/signal/ref/pwelch.html`)

McDonald, A., & Schrattenholzer, L. (2001). Learning rates for energy technologies. *Energy policy*, *29*(4), 255–261. doi: 10.1016/S0301-4215(00)00122-1

McMorland, J. (2024). *Operations and maintenance of floating wind: A holistic approach review, modelling, and opportunistic maintenance strategies*. The University of Strathclyde.

Mikkelsen, T. (2014). LIDAR-based research and innovation at DTU wind energy–a review. In *Journal of Physics: Conference Series* (Vol. 524, p. 012007). doi: 10.1088/1742-6596/524/1/012007

Miner, M. A. (1945). Cumulative damage in fatigue. *Journal of Applied Mechanics*, *12*(3), 159–164. doi: 10.1115/1.4009458

Namik, H., & Stol, K. (2010). Individual blade pitch control of floating offshore wind turbines. *Wind Energy*, *13*(1), 74–85. doi: 10.1002/we.332

Navalkar, S., van Wingerden, J., Fleming, P. A., & Van Kuik, G. (2015). Integrating robust LIDAR-based feedforward with feedback control to enhance speed regulation of floating wind turbines. In *2015 American Control Conference* (pp. 3070–3075). doi: 10.1109/ACC.2015.7171804

Nielsen, I. R. (1987). Assessment of ship performance in a seaway. NORDFORSK, Sortedam Dossering 19, DK-200 Copenhagen, Denmark, ISBN: 87-982637-1-4.

NREL. (2024a). OpenFAST user documentation. (URL: `https://openfast.readthedocs.io/en/main/source/user/index.html`)

NREL. (2024b). OpenFAST v3.5.0. (URL: `https://github.com/OpenFAST/openfast/releases/tag/v3.5.0`)

NREL. (2024c). WEIS. (URL: `https://github.com/WISDEM/WEIS`)

NREL. (2024d). WISDEM. (URL: `https://github.com/WISDEM/WISDEM`)

Otter, A., Murphy, J., Pakrashi, V., Robertson, A., & Desmond, C. (2022). A review of modelling techniques for floating offshore wind turbines. *Wind Energy*, *25*(5), 831–857. doi: https://doi.org/10.1002/we.2701

Papi, F., Bianchini, A., Ferri, G., Bruschi, N., & Marino, E. (2023). Influence of tower design on floating offshore wind turbine dynamics. In *International Conference on Offshore Mechanics and Arctic Engineering* (Vol. 86908, p. V008T09A053). doi: 10.1115/OMAE2023-104864

Patel, M. R., & Beik, O. (2021). Wind and solar power systems: design, analysis, and operation. CRC press.

Paya, E., & Zigeng, A. (2020). The frontier between fixed and floating foundations in offshore wind. (URL: `https://www.empireengineering.co.uk/the-frontier-between-fixed-and-floating-foundations-in-offshore-wind/`)

Pitter, M., Slinger, C., Guo, F., Schlipf, D., Raach, S., & White, S. (2023). A data-driven approach to the design and implementation of retrofit lidar assisted control system. In *Wind Energy Science Conference 2023*. (URL: `https://zenodo.org/records/8037351`)

Quest FWE. (2024). 2024 – 2025 Q FWE floating wind projects map. (URL: `https://questfwe.com/projects-map/`)

Raach, S., Schlipf, D., Sandner, F., Matha, D., & Cheng, P. W. (2014). Nonlinear model predictive control of floating wind turbines with individual pitch control. In *2014 American Control Conference* (pp. 4434–4439). doi: 10.1109/ACC.2014.6858718

Ramachandran, G., Robertson, A., Jonkman, J., & Masciola, M. (2013). Investigation of response amplitude operators for floating offshore wind turbines. In *ISOPE International Ocean and Polar Engineering Conference* (pp. ISOPE–I).

ReliaWind. (2011). Reliability focused research on optimizing Wind Energy systems design, operation and maintenance: Tools, proof of concepts, guidelines methodologies for a new generation. (URL: `https://cordis.europa.eu/project/id/212966/reporting`)

Rott, A., Schneemann, J., Theuer, F., Trujillo Quintero, J. J., & Kühn, M. (2022). Alignment of scanning lidars in offshore wind farms. *Wind Energy Science*, *7*(1), 283–297. doi: 10.5194/wes-7-283-2022

Rowell, D., Jenkins, B., Carroll, J., & McMillan, D. (2022). How does the accessibility of floating wind farm sites compare to existing fixed bottom sites? *Energies*, *15*(23), 8946. doi: 10.3390/en15238946

Rubin, E. S., Azevedo, I. M., Jaramillo, P., & Yeh, S. (2015). A review of learning rates for electricity supply technologies. *Energy Policy*, *86*, 198–218. doi: 10.1016/j.enpol.2015.06.011

Russell, A., Collu, M., McDonald, A., Thies, P., Keane, A., & Quayle, A. (2024). LIDAR-assisted feedforward individual pitch control of a 15 MW floating offshore wind turbine. *Wind Energy*, *27*(4), 341–362. doi: doi.org/10.1002/we.2891

Russell, A., Collu, M., McDonald, A., Thies, P., Mortimer, A., & Quayle, A. (2022). Review of LIDAR-assisted control for offshore wind turbine applications. In *Journal of Physics: Conference Series* (Vol. 2362, p. 012035). doi: 10.1088/1742-6596/2362/1/012035

Russell, A., Collu, M., McDonald, A., Thies, P., Mortimer, A., & Quayle, A. (2023). LIDAR-assisted feedforward pitch control of 15 MW floating offshore wind turbines. In *International Conference on Offshore Mechanics and Arctic Engineering* (Vol. 86908, p. V008T09A022). doi: 10.1115/OMAE2023-100822

Russell, A., McMorland, J., Collu, M., McDonald, A., Thies, P., Keane, A., ... Coraddu, A. (2024). The impact of LIDAR-assisted pitch control on floating offshore wind operational expenditure. *[under review]*.

Russell, P. B., Swissler, T. J., & McCormick, M. P. (1979). Methodology for error analysis and simulation of lidar aerosol measurements. *Applied Optics*, *18*(22), 3783–3797. doi: 10.1364/AO.18.003783

Sathe, A., Banta, R., Pauscher, L., Vogstad, K., Schlipf, D., & Wylie, S. (2015). Estimating turbulence statistics and parameters from ground-and nacelle-based LIDAR measurements: IEA Wind expert report. DTU Wind Energy.

Scheu, M. N., Kaufer, D., Schwarzkopf, M.-A., & Kolios, A. (2018). Workability on offshore wind turbines - A comparative study of fixed-bottom and floating applications. In *ISOPE International Ocean and Polar Engineering Conference* (pp. ISOPE–I).

Schlager, B., Goelles, T., Muckenhuber, S., & Watzenig, D. (2022). Contaminations on lidar sensor covers: Performance degradation including fault detection and modeling as potential applications. *IEEE Open Journal of Intelligent Transportation Systems*, *3*, 738–747. doi: 10.1109/OJITS.2022.3214094

Schlipf, D. (2016). LIDAR-assisted control concepts for wind turbines. *Wind Energy Science*. doi: 10.18419/opus-8796

Schlipf, D., Bossanyi, E., Carcangiu, C., Fischer, T., Maul, T., & Rossetti, M. (2011). LIDAR assisted collective pitch control. University of Stuttgart.

Schlipf, D., Fischer, T., Carcangiu, T., Rossetti, C., & Bossanyi, E. (2010). Load analysis of look-ahead collective pitch control using LIDAR. University of Stuttgart. doi: 10.18419/OPUS-3902

Schlipf, D., Fleming, P., Haizmann, F., Scholbrock, A., Hofsäss, M., Wright, A., & Cheng, P. (2014). Field testing of feedforward collective pitch control on the CART2 using a nacelle-based LIDAR scanner. In *Journal of Physics: Conference Series* (Vol. 555, p. 012090). doi: 10.1088/1742-6596/555/1/012090

Schlipf, D., Guo, F., Raach, S., & Lemmer, F. (2023). A tutorial on LIDAR-assisted control for floating offshore wind turbines. In *2023 American Control Conference* (pp. 2536–2541). doi: 10.23919/ACC55779.2023.10156419

Schlipf, D., Kapp, S., Anger, J., Bischoff, O., Hofsäss, M., Rettenmeier, A., & Kühn, M. (2011). Prospects of optimization of energy production by LIDAR-assisted control of wind turbines. University of Stuttgart. doi: 10.18419/OPUS-3916

Schlipf, D., & Kühn, M. (2008). Prospects of a collective pitch control by means of predictive disturbance compensation assisted by wind speed measurements. University of Stuttgart. doi: 10.18419/OPUS-3900

Schlipf, D., Lemmer, F., & Raach, S. (2020). Multi-variable feedforward control for floating wind turbines using LIDAR. In *30th ISOPE International Ocean and Polar Engineering Conference* (p. ISOPE-I-20-1174). doi: 10.18419/opus-11067

Schlipf, D., Pao, L., & Cheng, P. (2012). Comparison of feedforward and model predictive control of wind turbines using LIDAR. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)* (pp. 3050–3055). doi: 10.1109/CDC.2012.6426063

Schlipf, D., Sandner, F., Raach, S., Matha, D., & Cheng, P. (2013). Nonlinear model predictive control of floating wind turbines. In *The Twenty-third International Offshore and Polar Engineering Conference.*

Schlipf, D., Schlipf, D., & Kühn, M. (2013). Nonlinear model predictive control of wind turbines using LIDAR. *Wind Energy*, *16*(7), 1107–1129. doi: 10.1002/we.1533

Schlipf, D., Schuler, S., Grau, P., Allgöwer, F., & Kühn, M. (2010). Look-ahead cyclic pitch control using LIDAR. University of Stuttgart. doi: 10.18419/opus-4538

Schlipf, D., Simley, E., Lemmer, F., Pao, L., & Cheng, P. W. (2015). Collective pitch feedforward control of floating wind turbines using LIDAR. In *25th ISOPE International Ocean and Polar Engineering Conference* (p. ISOPE-I-15-755).

Schlipf, D., Trabucchi, D., Bischoff, O., Hofsäss, M., Mann, J., Mikkelsen, T., ... Kühn, M. (2011). Testing of frozen turbulence hypothesis for wind turbine applications with a scanning LIDAR system. University of Stuttgart. doi: 10.18419/OPUS-3915

Schlipf, D., Trujillo, J. J., Basterra, V., & Kühn, M. (2009). Development of a wind turbine LIDAR simulator. University of Stuttgart. doi: 10.18419/opus-3914

Scholbrock, A., Fleming, P., Fingersh, L., Wright, A., Schlipf, D., Haizmann, F., & Belen, F. (2013). Field testing LIDAR-based feedforward controls on the NREL controls advanced research turbine. In *51st AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition* (p. 818). doi: doi.org/10.2514/6.2013-818

Scholbrock, A., Fleming, P., Schlipf, D., Wright, A., Johnson, K., & Wang, N. (2016). LIDAR-enhanced wind turbine control: Past, present, and future. In *2016 American Control Conference* (pp. 1399–1406). doi: 10.1109/ACC.2016.7525113

Simley, E., & Pao, L. (2013). Correlation between rotating lidar measurements and blade effective wind speed. In *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition* (p. 749). doi: 10.2514/6.2013-749

Simley, E., Pao, L., Gebraad, P., & Churchfield, M. (2014). Investigation of the impact of the upstream induction zone on LIDAR measurement accuracy for wind turbine control applications using large-eddy simulation. In *Journal of Physics: Conference Series* (Vol. 524, p. 012003). doi: 10.1088/1742-6596/524/1/012003

Simley, E., Pao, L., Kelley, N., Jonkman, B., & Frehlich. (2012). Lidar wind speed measurements of evolving wind fields. In *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition* (p. 540). doi: 10.2514/6.2012-656

Simley, E., & Scholbrock, A. (2020). Wind workforce development series: Nacelle mounted LIDAR for wind energy applications. (URL: `https://www.youtube.com/watch?v=yQIMDoYmFno`)

Slot, R. M. M., Sørensen, J. D., Svenningsen, L., Moser, W., & Thøgersen, M. L. (2019). Effective turbulence and its implications in wind turbine fatigue assessment. *Wind Energy*, *22*(12), 1699–1715. doi: 10.1002/we.2397

sowento. (2021). Important milestone: 1000 wind turbines with lidar-assisted control. (URL: `https://www.sowento.com/important-milestone-reached-for-lidar-assisted-control-of-wind-turbines-1000-wind-turbines/`)

Sperstad, I. B., Stålhane, M., Dinwoodie, I., Endrerud, O.-E. V., Martin, R., & Warner, E. (2017). Testing the robustness of optimal access vessel fleet selection for operation and maintenance of offshore wind farms. *Ocean Engineering*, *145*, 334–343. doi: 10.1016/j.oceaneng.2017.09.009

Stanley, A. P. J., King, J., Bay, C., & Ning, A. (2020). A model to calculate fatigue damage caused by partial waking during wind farm optimization. *Wind Energy Science Discussions*, *2020*, 1–34. doi: 10.5194/wes-7-433-2022

statista. (2024). Offshore wind energy capacity worldwide from 2009 to 2023. (URL: `https://www.statista.com/statistics/476327/global-capacity-of-offshore-wind-energy/#statisticContainer`)

Stewart, G., & Lackner, M. A. (2014). The impact of passive tuned mass dampers and wind-wave misalignment on offshore wind turbine loads. *Engineering Structures*, *73*, 54–61. doi: 10.1016/j.engstruct.2014.04.045

Sustainable Ships. (2023). What is the carbon footprint of steel? (URL: `https://www.sustainable-ships.org/stories/2022/carbon-footprint-steel`)

Taylor, G. (1938). The spectrum of turbulence. *Proceedings of the Royal Society of London. Series A-Mathematical and Physical Sciences*, *164*(919), 476–490. doi: 10.1098/rspa.1938.0032

Towers, P., & Jones, B. L. (2016). Real-time wind field reconstruction from lidar measurements using a dynamic wind model and state estimation. *Wind Energy*, *19*(1), 133–150. doi: 10.1002/we.1824

UK Energy Research Centre. (2023). Wind turbines are already skyscraper-sized – is there any limit to how big they will get? (URL: `https://ukerc.ac.uk/news/wind-turbines-are-already-skyscraper-sized-is-there-any-limit-to-how-big-they-will-get/`)

Van Bussel, G. J., & Bierbooms, W. (2003). The DOWEC offshore reference windfarm: analysis of transportation for operation and maintenance. *Wind Engineering*, *27*(5), 381–391. doi: 10.1260/030952403322770986

Van der Hoven, I. (1957). Power spectrum of horizontal wind speed in the frequency range from 0.0007 to 900 cycles per hour. *Journal of Atmospheric Sciences*, *14*(2), 160–164. doi: 10.1175/1520-0469(1957)014%3C0160:PSOHWS%3E2.0.CO;2

van der Veen, G. J., Couchman, I. J., & Bowyer, R. (2012). Control of floating wind turbines. In *2012 American Control Conference (ACC)* (pp. 3148–3153). doi: 10.1109/ACC.2012.6315120

Vermeer, L., Sørensen, J. N., & Crespo, A. (2003). Wind turbine wake aerodynamics. *Progress in Aerospace Sciences*, *39*(6-7), 467–510. doi: 10.1016/S0376-0421(03)00078-2

Walling, F. B., & Otts, L. E. (1967). Water requirements of the iron and steel industry. US Government Printing Office.

Wang, C., Utsunomiya, T., Wee, S., & Choo, Y. (2010). Research on floating wind turbines: a literature survey. *The IES Journal Part A: Civil & Structural Engineering*, *3*(4), 267–277. doi: 10.1080/19373260.2010.517395

Wang, N., Johnson, K., & Wright, A. (2011). FX-RLS-based feedforward control for LIDAR-enabled wind turbine load mitigation. In *IEEE Transactions on Control Systems Technology* (Vol. 20, pp. 1212–1222). IEEE. doi: 10.1109/TCST.2011.2163515

Weibull, W. (1951). A statistical distribution function of wide applicability. *Journal of Applied Mechanics*.

WHOI. (2023). Know your ocean. (URL: `https://www.whoi.edu/know-your-ocean/`)

Wind Europe. (2017). Floating wind vision statement. (URL: `https://windeurope.org/wp-content/uploads/files/about-wind/reports/Floating-offshore-statement.pdf`)

Yu, W., Lemmer, F., Schlipf, D., Cheng, P. W., Visser, B., Links, H., ... Serna, J. (2018). Evaluation of control methods for floating offshore wind turbines. In *Journal of Physics: Conference Series* (Vol. 1104, p. 012033). doi: 10.1088/1742-6596/1104/1/012033

Zhao, D., Xin, J., Pan, X., Ma, Y., Ren, X., Jiang, Y., ... Gong, C. (2024). Large-eddy simulation of the effects of horizontal and vertical adjustments in a wind farm. *Frontiers in Energy Research*, *11*, 1333578. doi: 10.3389/fenrg.2023.1333578

Ziegler, J. G., & Nichols, N. B. (1942). Optimum settings for automatic controllers. *Transactions of the American Society of Mechanical Engineers*, *64*(8), 759–765. doi: 10.1115/1.4019264

Appendix A

Simulation tools

This Appendix will outline the aero-elastic-servo-hydro simulation tools used in conducting the research (OpenFAST and FAST.Farm) as well as the reference turbine and substructure models used. Detail will also be provided on the O&M modelling tool (StrathO&M) employed by Dr. Jade McMorland in completing the O&M case study and on NREL's Wind Energy with Integrated Servo-control (WEIS) tool used to perform the tower design modifications.

A.1 OpenFAST

The National Renewable Energy Laboratory (NREL)'s open source Fatigue, Aerodynamics, Structures, and Turbulence (FAST) code was used to perform the wind turbine simulations throughout this work. This software is classed as an aero-elastic-servo-hydro tool for the modelling of full nonlinear turbine models. It allows for the intricate specification of the wind turbine's design, the external environment (wind, wave and current conditions) and, critically for this research project, the controller design, as shown in Fig. A.1. It also allows for results to be recorded in multiple degrees of freedom. OpenFAST (see NREL, 2024b) is the latest iteration of the FAST code, and houses numerous modules which each specify a different part of the simulation. These are all called from a master .fst file, an example of which is provided in Fig. D.1.

A.1.1 Key modules

OpenFAST is comprised of several modules that handle different aspects of the simulation. This section will outline the key modules relevant to this research. More details can be found within its user documentation (see NREL, 2024a). A flowchart summary, from J. Jonkman (2022) of the OpenFAST modules and the interactions between them is shown in Fig. A.1.

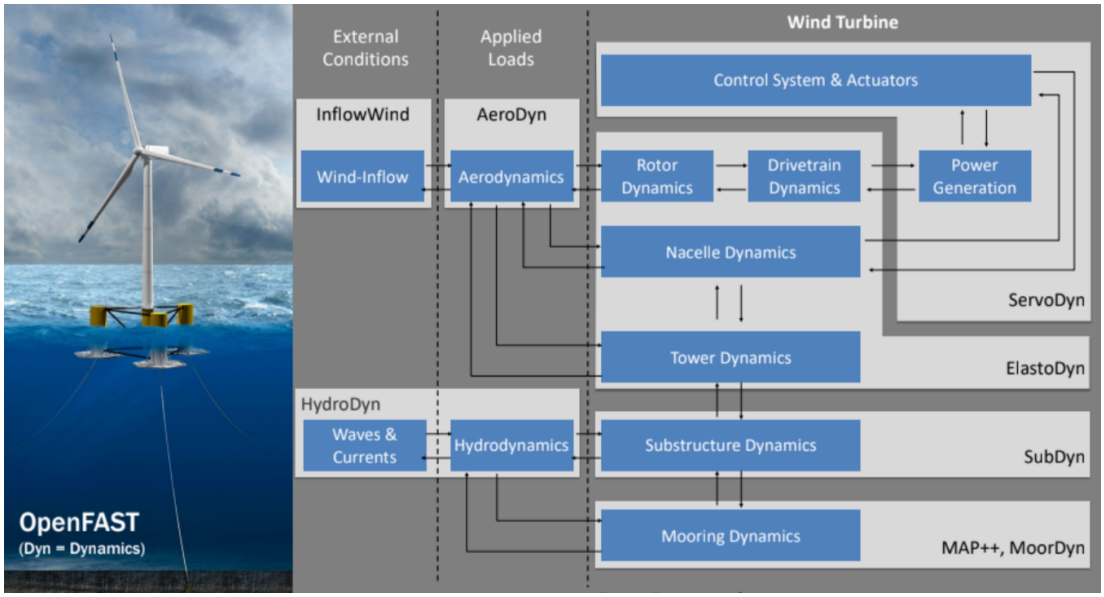


Figure A.1: Flowchart showing the OpenFAST modules and the interactions between them, from J. Jonkman (2022). Reprinted with permission from NREL.

ElastoDyn specifies the general features of the turbine and the simulation, including the degrees of freedom to be enabled, the initial conditions, the turbine configuration and the component masses. It captures the dynamics of the rotor, drivetrain, nacelle and tower. The degrees of freedom influence the complexity of the simulation and there are seventeen in total, covering the blade and tower modes, the generator and the platform motions of a FOWT.

InflowWind allows for the specification of the incoming wind field to be applied to the turbine within the simulations. This can either take the form of a steady (single wind speed that varies with height above sea level), uniform (single wind speed that does not vary with height above sea level) or turbulent wind field. Turbulent wind fields can be generated using NREL's TurbSim stochastic inflow simulator, developed by Kelley and Jonkman (2005). The input file of TurbSim can be configured as per the user guide outlined by B. Jonkman (2014) to generate different wind fields with characteristic durations, mean wind speeds, turbulence intensities and power law coefficients (for wind shear). The turbulence can be generated using the Mann (1994) or Kaimal et al. (1972) turbulence models. The wind fields generated by TurbSim act in a marching manner such that the wind speeds do not evolve as the wind moves through space. The InflowWind module receives the TurbSim-generated wind field, which is applied to the turbine within the simulation.

The **AeroDyn** module characterises the airfoils of the turbine blades used within the blade-element/momentum (BEM) theory model in response to the inflow wind conditions delivered by the InflowWind module. Within BEM, a blade is split into several elements, which are analysed individually. BEM is based on the principle of conservation of momentum and accounts for both the aerodynamic forces acting on the blades and the velocities induced by neighbouring blade elements. BEM is used to determine performance parameters such as the rotor thrust and power of the turbine as well as the loadings applied to the blades.

The **ServoDyn** module characterises the features of the control system. Within ServoDyn, the four primary control systems (pitch, torque, yaw and high-speed shaft brake) of a turbine can be configured. ServoDyn enables two primary control interfaces, either to a Simulink model or to a dynamic link library (`.dll`), typically the reference open-source controller (ROSCO), detailed by Abbas et al. (2022), which are then used to compute the controller commands. The results documented in this work were obtained when using the `.dll` interface to ROSCO.

HydroDyn is used to specify the details of the wave kinematics, the wave forces and the floating platform hydrodynamic properties, which determine the platform's response to wave excitation. HydroDyn links to the **SubDyn** and **MoorDyn** modules, which capture the dynamics of the floating substructure and mooring system, respectively. Features of the wave profile, such as whether regular (periodic) or irregular waves (characterised by the JONSWAP or Pierson-Moskowitz spectra) are modelled, in addition to the significant wave height (H_s) and peak spectral period (T_p). More details on the operation of the HydroDyn module can be found within its user guide and theory manual (see J. M. Jonkman et al., 2014). The hydrodynamic response of the floating platform is determined via a pre-processing step and requires modelling of the platform within a numerical-panel code such as WAMIT (Wave Analysis at MIT), developed by C. H. Lee and Newman (2006). WAMIT uses a three-dimensional numerical-panel method in the frequency domain to solve the linearized hydrodynamic radiation and diffraction problems for the interaction of surface waves with offshore platforms and generates output files to be input to HydroDyn which characterise the hydrodynamic response of the floating platform.

The key modules of interest for this research project were InflowWind and ServoDyn, as these were required to be modified for the implementation of LIDAR simulation and LAC. The InflowWind module was identified as the point of integration for the LIDAR simulator, while modifications to ServoDyn were required to receive the outputs from the LIDAR and interface them to ROSCO. The changes made to the OpenFAST and ROSCO source codes, written in Fortran, are detailed in Chapter 3.

A.1.2 ROSCO

ROSCO is the baseline controller used by OpenFAST for the control of wind turbines, developed by Abbas et al. (2022). Outputs from OpenFAST are interfaced to ROSCO, which is compiled as a `.dll`, which subsequently computes the controller outputs to be relayed to the turbine. ROSCO primarily offers feedback torque, collective and individual pitch and yaw control capabilities. Additionally, it contains setpoint smoothing and wind speed estimator features. Abbas et al. (2020) implemented setpoint smoothing to ensure smooth transitioning between control regions 2 and 3 (shown in Fig. 1.4) by separating the torque and pitch controller setpoints, preventing unnecessary switching between the controllers. An extended Kalman-filter-based wind speed estimator is employed to estimate the optimal rotor speeds in below-rated operation and to define minimum blade pitch angles in above-rated operation. Both of these features were enabled within the simulations performed in the work outlined in this thesis.

ROSCO uses a gain-scheduled feedback proportional-integral blade pitch controller in above-rated wind speed conditions, as detailed by Abbas et al. (2020) and described in Sect. 1.3.5. The source code of ROSCO was modified in this work to enable LIDAR-assisted feedforward collective and individual pitch control. These changes are outlined in Sect. 3.4.

A.2 Turbine and substructure models

This section provides information regarding the turbine and supporting structure used when performing the simulations.

A.2.1 Turbine

The turbine studied in this work was the IEA 15 MW reference turbine, developed between NREL and the Technical University of Denmark (DTU), via the IEA, as defined by Gaertner et al. (2020). The specifications of the reference turbine are given in Table A.1.

The turbine rotor operates with a minimum rotational speed of 5 rpm to avoid 3P interference with the tower/monopile natural frequencies (see Gaertner et al., 2020), and reaches a rated rotational speed of 7.56 rpm at 10.59 ms^{-1}. The blades begin pitching at the rated wind speed to maintain the rotor rotation at its rated speed. As a result of the design optimisation process, the IEA 15 MW turbine has a negligible control region 2.5, since the rated rotor speed is reached at rated torque (see Gaertner et al., 2020).

Table A.1: Parameters of the IEA 15 MW reference turbine, from Gaertner et al. (2020).

Parameter	Value	Units
Power rating	15	MW
Turbine class	IEC Class B	-
Cut-in wind speed	3.0	ms^{-1}
Rated wind speed	10.59	ms^{-1}
Cut-out wind speed	25.0	ms^{-1}
Design tip-speed ratio	9.0	-
Minimum rotor speed	5.0	rpm
Maximum rotor speed	7.56	rpm
Rotor diameter	240	m
Hub height	150	m
Drive train	Direct drive	-

A.2.2 Floating platforms

The floating substructure models utilised for this work were the University of Maine (UMaine)'s VolturnUS-S semi-submersible from Allen et al. (2020) and the WindCrete spar from Mahfouz et al. (2020), designed to support the IEA 15 MW reference wind turbine. Key specifications of the substructure are shown in Table A.2.

Table A.2: Parameters of the VolturnUS-S (see Allen et al., 2020) and WindCrete (see Mahfouz et al., 2020) FOWT models.

Parameter	VolturnUS-S	WindCrete	Units
Topology	Semi-submersible	Spar	-
Material	Steel	Concrete	-
Hub height	150	135	m
Draft	20	155	m
Mass (including ballast)	17,839	39,805	kg
Mooring system	Three-line catenary	Three-line catenary	-

Rigid body natural frequencies

In order to give context to the responses of the two FOWTs to the LAC addition, it is important to consider their rigid body natural frequencies, obtained by performing free-decay tests. The results of these tests for the two designs were published within their respective reference documents, and are shown in Table A.3.

Table A.3: Rigid body natural frequencies of the VolturnUS-S (see Allen et al., 2020) and WindCrete (see Mahfouz et al., 2020) FOWT models obtained from free decay tests.

Degree of freedom	VolturnUS-S, Hz	WindCrete, Hz
Pitch	0.036	0.024
Roll	0.036	0.024
Yaw	0.011	0.091
Heave	0.049	0.031
Surge	0.007	0.012
Sway	0.007	0.012
Tower 1st fore-aft bending mode	0.496	0.494
Tower 1st side-to-side bending mode	0.483	0.513

From Table A.3, it can be seen that the VolturnUS-S has higher natural frequencies of motion, in pitch, roll and heave than the WindCrete, whereas the WindCrete has higher natural frequencies in Yaw, Surge and Sway. Table A.3 also highlights that the natural frequencies of the towers' 1st fore-aft and side-to-side bending modes were set above the 1P and 3P ranges for both substructure designs.

A.3 StrathO&M

The O&M and OpEx calculation simulations reported within this research project were performed by Dr. McMorland using the StrathO&M model developed by Dinwoodie et al. (2012). This model has previously been validated and used to assist in the development of maintenance campaigns for currently operational offshore wind farms (see Dinwoodie et al., 2015). This model was adapted to account for workability limits imposed based on the floating platform design.

The model uses a Monte Carlo time domain simulation approach and consists of four input modules: climate; vessel specifications and fleet configuration; wind farm/turbine; and cost and failures. For the analysis reported in this work within Chapter 6, the model was adapted for FOW day-to-day maintenance procedures, excluding major replacement procedures. This model comprises of three main parts: climate modelling, turbine failure modelling and resource and cost modelling. Users input sample data, allowing the model to simulate wind speed and significant wave height time series over the defined wind farm lifetime using a multivariate auto-regressive model. These values play a crucial role in calculating energy production and losses, as well as determining turbine accessibility for maintenance tasks.

Within the current model by Dinwoodie et al. (2012), access to site is limited by V and H_s, where the thresholds are vessel-dependent. Within this adapted model, where workability limits are imposed, the combination of T_p and H_s form the limiting criteria, dependent on the specific platform design.

A.4 WEIS

The structural design tool employed for this study was NREL (2024c)'s Wind Energy with Integrated Servo-control (WEIS). WEIS is a framework that combines multiple NREL-developed tools to enable design optimisation of floating offshore wind turbines. These tools are the Wind-Plant Integrated System Design and Engineering Model (WISDEM) (see NREL, 2024d), OpenFAST, TurbSim and ROSCO. WISDEM is a set of models used for assessing the overall cost of energy (COE) of the wind turbine. The models use wind turbine cost and energy production revenue as well as financial models to estimate the COE and other wind plant system attributes.

Within WEIS, the turbine's components (blade, tower, platform, mooring, and more) are parameterised in a geometry input file. WISDEM determines component masses, costs, and engineering parameters that can be used to determine modelling inputs. These were crucial steps for performing the work reported in Chapter 7, as the tower was customised and new masses for the modified designs were calculated. Subsequently, the OpenFAST and ROSCO input files are created, reflecting the component parametric inputs specified by the user. This includes the calculation of the tower frequency mode shapes. WEIS then uses TurbSim to create turbulent wind files used during the OpenFAST simulations.

WEIS is primarily intended as an optimisation tool and uses OpenMDAO, an open-source framework for efficient multi-disciplinary optimisation developed by Gray et al. (2019), to achieve this. In this approach, users can specify design variables, which they intend to be modified during the optimisation process, and constraints, which ensure that design variables are not set to values that lead to the constraints being violated.

A.5 FAST.Farm

FAST.Farm is an extension of OpenFAST, also published by NREL, that enables the simulation of multiple wind turbines within an array. FAST.Farm uses separate instances of OpenFAST and ROSCO to solve the aero-elastic-servo-hydro dynamics of each individual turbine, but considers additional physics for wind-farm-wide ambient wind in the atmospheric boundary layer as well as wake properties such as wake deficits, advection, deflection, meandering, and merging (see J. Jonkman & Shaler, 2021). It achieves this through the simulation of a high-resolution grid in the immediate vicinity of the wind turbines, and a low-resolution grid in the remaining space. To model the wakes, FAST.Farm employs the dynamic wake meandering (DWM) model, from G. C. Larsen et al. (2007). The DWM model accounts for temporal and spatial variations in the wake, capturing changes in the wake over time and

distance. Within FAST.Farm, the wake deficit evolution is modelled via the thin shear-layer approximation of the Reynolds-averaged Navier-Stokes equations under quasi-steady-state conditions in axisymmetric coordinates, with turbulence closure captured by using an eddy-viscosity formulation (see J. Jonkman & Shaler, 2021).

Appendix B

Relevant OpenFAST, ROSCO and WEIS source code

This Appendix outlines the modifications made to the source codes of OpenFAST, ROSCO and WEIS, as referenced in Chapter 3.

B.1 OpenFAST

```
########################################################################################################################
# Registry for Lidar module in the FAST Modularization Framework
# Entries are of the form
# <keyword> <ModuleName/ModName>    <TypeName>  <FieldType> <FieldName> <Dims>  <IO>    <DNAME> <DESCRIP>   <UNITS>
#
# Use ^ as a shortcut for the value in the same column from the previous line.
########################################################################################################################
# ..... Include files (definitions from NWTC Library) ..................................................................
#include    Registry_NWTC_Library.txt

# LIDAR Constants
param   Lidar   -   IntKi   SensorType_None              -   0   -       -
param   Lidar   -   IntKi   SensorType_SinglePoint      -   1   -       -
param   ^   -   IntKi   SensorType_ContinuousLidar  -   2   -       -
param   ^   -   IntKi   SensorType_PulsedLidar      -   3   -       -

# ..... LIDAR_InitInputType data .......................................................................................
typedef ^   Lidar_InitInputType IntKi   SensorType       -  SensorType_None - "SensorType_* parameter"  -
typedef ^   Lidar_InitInputType DbKi    Tmax              -       -       -    "the length of the simulation" "s"
typedef ^   Lidar_InitInputType ReKi    RotorApexOffsetPos  {3}     -       -    "position of the lidar unit relative to the rotor apex of rotation" "m"
typedef ^   Lidar_InitInputType ReKi    HubPosition     {3}     -       -    "initial position of the hub (lidar mounted on hub) [0,0,HubHeight]" "m"
typedef ^   Lidar_InitInputType IntKi   NumPulseGate        -       -       -    "the number of range gates to return wind speeds at"    -
typedef ^   Lidar_InitInputType LOGICAL LidRadialVel        -       -       -    "TRUE => return radial component, FALSE => return 'x' direction estimate"   -

# ..... LIDAR_InitInputType data .......................................................................................
typedef ^   Lidar_InitOutputType    ReKi    DummyInitOut         -

# ..... LIDAR_ParameterType data .......................................................................................
typedef ^   Lidar_ParameterType IntKi   NumPulseGate        -       -       -    "the number of range gates to return wind speeds at; pulsed lidar only" -
typedef ^   Lidar_ParameterType ReKi    RotorApexOffsetPos  {3}     -       -    "position of the lidar unit relative to the rotor apex of rotation" "m"
typedef ^   Lidar_ParameterType ReKi    RayRangeSq          -       -       -    "Rayleigh Range Squared"
typedef ^   Lidar_ParameterType ReKi    SpatialRes          -       -       -    "spatial sampling distance of weighting function (1/2)*(avg ws)*dt"
typedef ^   Lidar_ParameterType IntKi   SensorType          -       -       -    "SensorType_* parameter"    -
typedef ^   Lidar_ParameterType ReKi    WtFnTrunc           -       -       -    "Percentage of the peak value at which to truncate weighting function"
typedef ^   Lidar_ParameterType ReKi    PulseRangeOne       -       -       -    "the range to the closest range gate" "m"
typedef ^   Lidar_ParameterType ReKi    DeltaP              -       -       -    "the distance between range gates" "m"
typedef ^   Lidar_ParameterType ReKi    DeltaR              -       -       -    "the FWHM width of the pulse"
typedef ^   Lidar_ParameterType ReKi    r_p                 -       -       -
typedef ^   Lidar_ParameterType LOGICAL LidRadialVel        -       -       -    "TRUE => return radial component, FALSE => return 'x' direction estimate"   -
typedef ^   Lidar_ParameterType ReKi    DisplacementLidarX  -       -       -    "Displacement of the lidar system from the focal measurement point" "m"
typedef ^   Lidar_ParameterType ReKi    DisplacementLidarY  -       -       -    "Displacement of the lidar system from the focal measurement point" "m"
typedef ^   Lidar_ParameterType ReKi    DisplacementLidarZ  -       -       -    "Displacement of the lidar system from the focal measurement point" "m"
typedef ^   Lidar_ParameterType IntKi   NumBeam             -       -       -   "Number of lidar beams"  -
typedef ^   Lidar_ParameterType ReKi    FocalDistanceX      :       -       -   "LIDAR LOS focal distance co-ordinates in the x direction" "m"
typedef ^   Lidar_ParameterType ReKi    FocalDistanceY      :       -       -   "LIDAR LOS focal distance co-ordinates in the y direction" "m"
typedef ^   Lidar_ParameterType ReKi    FocalDistanceZ      :       -       -   "LIDAR LOS focal distance co-ordinates in the z direction" "m"
typedef ^   Lidar_ParameterType ReKi    MsrPosition     {:}{:}      -       -    "Position of the desired wind measurement (was XMsrPt, YMsrPt, ZMsrPt)" "m"
typedef ^   Lidar_ParameterType ReKi    PulseSpacing        -       -       -   "Distance between range gates" "m"
typedef ^   Lidar_ParameterType ReKi    URefLid             -       -       -   "Reference average wind speed for the lidar" "m/s"
typedef ^   Lidar_ParameterType IntKi   ConsiderHubMotion   -       -       -    "Flag whether to consider the hub motion's impact on the Lidar measurement" -
typedef ^   Lidar_ParameterType ReKi    MeasurementInterval  -      -       -    "Time steps between lidar measurements" "s"
typedef ^   Lidar_ParameterType ReKi    LidPosition     {3}     -       -    "Position of the Lidar unit (was XLidPt, YLidPt, ZLidPt)" "m"

# ..... States .........................................................................................................
# Define continuous (differentiable) states here:
typedef ^   ContinuousStateType ReKi    DummyContState     - -     - "Remove this variable if you have continuous states"    -
# Define discrete (nondifferentiable) states here:
typedef ^   DiscreteStateType   ReKi    DummyDiscState     - -     - "Remove this variable if you have discrete states"  -
# Define constraint states here:
typedef ^   ConstraintStateType ReKi    DummyConstrState   - -     - "Remove this variable if you have constraint states"    -
# Define "other" states (any data that are not considered actual states) here:
typedef ^   OtherStateType          ReKi    DummyOtherState    - -     -
```

Figure B.1: `LIDAR.txt` file (part 1) following modifications made to enable LIDAR simulation within the InflowWind module of OpenFAST. The green boxes indicate the parts of the code that were added or modified within this research project.

```
# ..... Misc/Optimization variables.....................................................................................
# Define any data that are used only for efficiency purposes (these variables are not associated with time):
#   e.g. indices for searching in an array, large arrays that are local variables in any routine called multiple times, etc.
typedef ^       MiscVarType             ReKi     DummyMiscVar     -     -     -  "Remove this variable if you have misc variables" -


# ..... LIDAR_InputType data ............................................................................................
typedef ^   Lidar_InputType         ReKi    PulseLidEl    - -     -  "the angle off of the x axis that the lidar is aimed (0 would be sta
typedef ^   Lidar_InputType         ReKi    PulseLidAz    - -     -  "the angle in the YZ plane that the lidar is staring (if PulseLidEl
typedef ^   Lidar_InputType         ReKi    HubDisplacementX  - -     -  "X direction hub displacement of the lidar (from ElastoDyn)" "m"
typedef ^   Lidar_InputType         ReKi    HubDisplacementY  - -     -  "Y direction hub displacement of the lidar (from ElastoDyn)" "m"
typedef ^   Lidar_InputType         ReKi    HubDisplacementZ  - -     -  "Z direction hub displacement of the lidar (from ElastoDyn)" "m"

# ..... LIDAR_OutputType data ...........................................................................................
typedef ^   Lidar_OutputType    ReKi    LidSpeed     {:}    -     -  "Speed detected by Lidar at measurement point (range gates for pulse
typedef ^   Lidar_OutputType    ReKi    WtTrunc          {:}    -     -  "Contains the fraction of the peak that the weighting function w
typedef ^   Lidar_OutputType    ReKi    MsrPositionsX    {:}    -     -  "Lidar X direction measurement points" "m"
typedef ^   Lidar_OutputType    ReKi    MsrPositionsY    {:}    -     -  "Lidar Y direction measurement points" "m"
typedef ^   Lidar_OutputType    ReKi    MsrPositionsZ    {:}    -     -  "Lidar Z direction measurement points" "m"
#typedef    ^   Lidar_OutputType    IntKi   LidErr           {:}    -     -  "Error code; THIS NEEDS TO GET FIXED (no integer outputs)"
```

Figure B.2: `LIDAR.txt` file (part 2) following modifications made to enable LIDAR simulation within the InflowWind module of OpenFAST. The green boxes indicate the parts of the code that were added or modified within this research project.

```
################################################################################################################################
# Registry for InflowWind, creates MODULE InflowWind_Types
# Module InflowWind_Types contains all of the user-defined types needed in InflowWind. It also contains copy, destroy, pack, and
# unpack routines associated with each defined data types.
################################################################################################################################
# Entries are of the form
# keyword <InflowWind[/ModuleAbvr]> <TypeName>   <FieldType>    <FieldName>  <Dims>  <Ctrl> <DNAME> <DESCRIP> <UNITS>
################################################################################################################################

usefrom IfW_FlowField.txt
usefrom InflowWind_IO.txt
usefrom Lidar.txt

include Registry_NWTC_Library.txt


param    InflowWind/InflowWind   -               IntKi           Undef_WindNumber     -    0    -    "This is the code for an undefined WindFileType"
param    ^                       -               IntKi           Steady_WindNumber    -    1    -    "Steady wind.  Calculated internally."
param    ^                       -               IntKi           Uniform_WindNumber   -    2    -    "Uniform wind.  Formally known as a Hub-Height wind file."
param    ^                       -               IntKi           TSFF_WindNumber      -    3    -    "TurbSim full-field binary file."
param    ^                       -               IntKi           BladedFF_WindNumber  -    4    -    "Bladed style binary full-field file.  Includes native bladed format"
param    ^                       -               IntKi           HAWC_WindNumber      -    5    -    "HAWC wind file."
param    ^                       -               IntKi           User_WindNumber      -    6    -    "User defined wind."
param    ^                       -               IntKi           BladedFF_Shr_WindNumber  - 7    -    "Native Bladed binary full-field file."      -
param    ^                       -               IntKi           FDext_WindNumber     -    8    -    "4D wind from external souce (i.e., FAST.Farm)."
param    ^                       -               IntKi           Point_WindNumber     -    9    -    "1D wind components from ExtInflow"
param    ^                       -               IntKi           Highest_WindNumber   -    9    -    "Highest wind number supported."

param    ^                       -               IntKi           IfW_NumPtsAvg        -    144    -    "Number of points averaged for rotor-average wind speed"

######################
# ..... Input file data ........................................................................................
# This is data defined in the Input File for this module (or could otherwise be passed in)
# ..... Primary Input file data ................................................................................
typedef  InflowWind/InflowWind   InflowWind_InputFile   LOGICAL      EchoFlag          -     -     -    "Echo the input file"                                                  -
typedef  ^                       ^                       IntKi         WindType          -     0     -    "Type of windfile"                                                     -
typedef  ^                       ^                       ReKi          PropagationDir    -     -     -    "Direction of wind propagation (meteorological direction)"  (degrees)
typedef  ^                       ^                       ReKi          VFlowAngle        -     -     -    "Vertical (upflow) angle"                                              degrees
typedef  ^                       ^                       LOGICAL       VelInterpCubic    -   .FALSE. -    "Use cubic interpolation for velocity in time (false=linear, true=cubic) |
typedef  ^                       ^                       IntKi         NWindVel          -     -     -    "Number of points to output the wind velocity (0 to 9)"
typedef  ^                       ^                       ReKi          WindVxiList       :     -     -    "List of X coordinates for wind velocity measurements"   meters
typedef  ^                       ^                       ReKi          WindVyiList       :     -     -    "List of Y coordinates for wind velocity measurements"   meters
typedef  ^                       ^                       ReKi          WindVziList       :     -     -    "List of Z coordinates for wind velocity measurements"   meters
typedef  ^                       ^                       ReKi          Steady_HWindSpeed -     -     -    "Steady wind -- horizontal windspeed"                    meters/s
typedef  ^                       ^                       ReKi          Steady_RefHt      -     -     -    "Steady wind -- reference height"                        meters
typedef  ^                       ^                       ReKi          Steady_PLexp      -     -     -    "Steady wind -- power law exponent"                      -
typedef  ^                       ^                       CHARACTER(1024)  Uniform_FileName  -     -     -    "Uniform wind -- filename"                               -
typedef  ^                       ^                       ReKi          Uniform_RefHt     -     -     -    "Uniform wind -- reference height"                       meters
typedef  ^                       ^                       ReKi          Uniform_RefLength -     -     -    "Uniform wind -- reference length"                       meters
typedef  ^                       ^                       CHARACTER(1024)  TSFF_FileName     -     -     -    "TurbSim Full-Field -- filename"                         -
typedef  ^                       ^                       CHARACTER(1024)  BladedFF_FileName -     -     -    "Bladed-style Full-Field -- filename"                    -
typedef  ^                       ^                       LOGICAL       BladedFF_TowerFile -     -     -    "Bladed-style Full-Field -- tower file exists"           -
typedef  ^                       ^                       LOGICAL       CTTS_CoherentTurb -     .FALSE.     -     "Coherent turbulence data exists"                                  -
typedef  ^                       ^                       CHARACTER(1024)  CTTS_FileName     -     -     -    "Name of coherent turbulence file"                       -
typedef  ^                       ^                       CHARACTER(1024)  CTTS_Path         -     -     -    "Path to coherent turbulence binary data files"          -
typedef  ^                       ^                       CHARACTER(1024)  HAWC_FileName_u   -     -     -    "HAWC -- u component binary data file name"              -
typedef  ^                       ^                       CHARACTER(1024)  HAWC_FileName_v   -     -     -    "HAWC -- v component binary data file name"              -
typedef  ^                       ^                       CHARACTER(1024)  HAWC_FileName_w   -     -     -    "HAWC -- w component binary data file name"              -
typedef  ^                       ^                       IntKi         HAWC_nx           -     -     -    "HAWC -- number of grids in x direction"                 -
typedef  ^                       ^                       IntKi         HAWC_ny           -     -     -    "HAWC -- number of grids in y direction"                 -
typedef  ^                       ^                       IntKi         HAWC_nz           -     -     -    "HAWC -- number of grids in z direction"                 -
typedef  ^                       ^                       ReKi          HAWC_dx           -     -     -    "HAWC -- distance between points in x direction"         meters
typedef  ^                       ^                       ReKi          HAWC_dy           -     -     -    "HAWC -- distance between points in y direction"         meters
typedef  ^                       ^                       ReKi          HAWC_dz           -     -     -    "HAWC -- distance between points in z direction"         meters
typedef  ^                       ^                       LOGICAL       SumPrint          -     -     -    "Write summary info to a file <ROOTNAME>.IfW.Sum"        -
typedef  ^                       ^                       IntKi         NumOuts           -     -     -    "Number of parameters in the output list (number of outputs requested)" -
typedef  ^                       ^                       CHARACTER(ChanLen) OutList       :     -     -    "List of user-requested output channels"                 -
typedef  ^                       ^                       IntKi         SensorType        -     -     -    "Sensor type (for lidar/sensor module)"                  -
typedef  ^                       ^                       IntKi         NumBeam           -     -     -    "Number of lidar beams"                                  -
typedef  ^                       ^                       IntKi         NumPulseGate      -     -     -    "The number of range gates to return wind speeds at"     -
typedef  ^                       ^                       ReKi          RotorApexOffsetPos {3}  -     -    "Position of the lidar unit relative to the rotor apex of rotation" m
typedef  ^                       ^                       ReKi          FocalDistanceX    :     -     -    "LIDAR LOS focal distance co-ordinates in the x direction" m
typedef  ^                       ^                       ReKi          FocalDistanceY    :     -     -    "LIDAR LOS focal distance co-ordinates in the y direction" m
typedef  ^                       ^                       ReKi          FocalDistanceZ    :     -     -    "LIDAR LOS focal distance co-ordinates in the z direction" m
typedef  ^                       ^                       ReKi          PulseSpacing      -     -     -    "Distance between range gates" m
typedef  ^                       ^                       ReKi          MeasurementInterval -     -     -    "Time between each measurement" s
typedef  ^                       ^                       ReKi          URefLid           -     -     -    "Reference average wind speed for the lidar" m/s
typedef  ^                       ^                       LOGICAL       LidRadialVel      -     -     -    "TRUE => return radial component, FALSE => return 'x' direction estimate
typedef  ^                       ^                       IntKi         ConsiderHubMotion -     -     -    "Flag whether or not the hub motion's impact on the Lidar measurement wi
typedef  ^                       ^                       Grid3D_InitInputType     FF     -     -     -    "scaling data" -
```

Figure B.3: `InflowWind.txt` file (part 1) following the modifications made. The green box indicates the part of the code that was added/modified within this research project.

```
typedef  ^                     InitInputType     CHARACTER(1024)   InputFileName      -     -     -     "Name of the InflowWind input file to use"
typedef  ^                     ^                 LOGICAL           Linearize          -     .FALSE. -   "Flag that tells this module if the glue code wants to linearize." -
typedef  ^                     ^                 LOGICAL           Use4Dext           -     .FALSE. -   "Flag that tells this module if an external module will pass it 4-D velocity gr
typedef  ^                     ^                 IntKi             NumWindPoints      -     -     -     "Number of wind velocity points expected"
typedef  ^                     ^                 IntKi             TurbineID          -     0     -     "Wind turbine ID number in the fixed (DEFAULT) file name when FixedWindFileRoot
typedef  ^                     ^                 LOGICAL           FixedWindFileRootName - .FALSE. -   "Do the wind data files have a fixed (DEFAULT) file name? (used by FAST.Farm)"
typedef  ^                     ^                 LOGICAL           UseInputFile       -     .TRUE.  -   "Should we read everthing from an input file, or do we get it some other way"
typedef  ^                     ^                 CHARACTER(1024)   RootName           -     -     -     "RootName for writing output files"
typedef  ^                     ^                 FileInfoType      PassedFileData     -     -     -     "If we don't use the input file, pass everything through this"
typedef  ^                     ^                 LOGICAL           WindType2UseInputFile -    .TRUE.      -       "Flag for toggling file based IO in wind type 2."
typedef  ^                     ^                 FileInfoType      WindType2Data      -     -     -     "Optional slot for wind type 2 data if file IO is not used."
typedef  ^                     ^                 LOGICAL           OutputAccel        -     .FALSE.     -       "Flag to output wind acceleration"
typedef  ^                     ^                 Lidar_InitInputType  lidar           -     -     -     "InitInput for lidar data"
typedef  ^                     ^                 Grid4D_InitInputType FDext           -     -     -     "InitInput for 4D external wind data"
typedef  ^                     ^                 ReKi              RadAvg             -     -     -     "Radius (from hub) used for averaging wind speed" -
typedef  ^                     ^                 IntKi             MHK                -     -     -     "MHK turbine type switch"   -
typedef  ^                     ^                 ReKi              WtrDpth            -     -     -     "Water depth" m
typedef  ^                     ^                 ReKi              MSL2SWL            -     -     -     "Mean sea level to still water level" m
typedef  ^                     ^                 IntKi             BoxExceedAllowIdx  -  -1      -     "Extrapolate winds outside box starting at this index (for OLAF wakes and L
typedef  ^                     ^                 LOGICAL           BoxExceedAllowF    -  .FALSE. -     "Flag to allow Extrapolation winds outside box starting at this index (for

# Init Output
typedef  ^                     InitOutputType    CHARACTER(ChanLen)   WriteOutputHdr    :   -   -   "Names of output-to-file channels"
typedef  ^                     ^                 CHARACTER(ChanLen)   WriteOutputUnt    :   -   -   "Units of output-to-file channels"
typedef  ^                     ^                 ProgDesc             Ver               -   -   -   "Version information of InflowWind module"
typedef  ^                     ^                 WindFileDat          WindFileInfo      -   -   -   "Meta data from the wind file"
typedef  ^                     InitOutputType    CHARACTER(LinChanLen) LinNames_y       {:} -   -   "Names of the outputs used in linearization" -
typedef  ^                     InitOutputType    CHARACTER(LinChanLen) LinNames_u       {:} -   -   "Names of the inputs used in linearization" -
typedef  ^                     InitOutputType    LOGICAL              RotFrame_y        {:} -   -   "Flag that tells FAST/MBC3 if the outputs used in linearization are in the rotat
typedef  ^                     InitOutputType    LOGICAL              RotFrame_u        {:} -   -   "Flag that tells FAST/MBC3 if the inputs used in linearization are in the rotat
typedef  ^                     InitOutputType    LOGICAL              IsLoad_u          {:} -   -   "Flag that tells FAST if the inputs used in linearization are loads (for precon


# ..... Parameters ............................................................................................
# Define parameters here:
# Time step for integration of continuous states (if a fixed-step integrator is used) and update of discrete states:
typedef  ^                     ParameterType     CHARACTER(1024)   RootFileName      -     -     -     "Root of the InflowWind input   filename"
typedef  ^                     ^                 IntKi             WindType           -     0     -     "Type of wind -- set to Undef_Wind initially"
typedef  ^                     ^                 DbKi              DT                 -     -     -     "Time step for cont. state integration & disc. state update"   seconds
typedef  ^                     ^                 ReKi              WindViXYZprime     ::    -     -     "List of XYZ coordinates for velocity measurements, translated to the wind coor
typedef  ^                     ^                 ReKi              WindViXYZ          ::    -     -     "List of XYZ coordinates for wind velocity measurements, 3xNWindVel" meters
typedef  ^                     ^                 FlowFieldType     FlowField          -     -     -     "Parameters from Full-Field"  -
typedef  ^                     ^                 ReKi              PositionAvg        ::    -     -     "(non-rotated) positions of points used for averaging wind speed"  meters
typedef  ^                     ^                 IntKi             NWindVel           -     -     -     "Number of points in the wind velocity list"
typedef  ^                     ^                 IntKi             NumOuts            -     0     -     "Number of parameters in the output list (number of outputs requested)" -
typedef  ^                     ^                 OutParmType       OutParam           {:}   -     -     "Names and units (and other characteristics) of all requested output parameters"
typedef  ^                     ^                 IntKi             OutParamLinIndx    {:}{:} -    -     "Index into WriteOutput for WindViXYZ in linearization analysis" -
typedef  ^                     ^                 lidar_ParameterType lidar            -     -     -     "Lidar parameter data"
typedef  ^                     ^                 LOGICAL           OutputAccel        -     .FALSE.     -       "Flag to output wind acceleration"


# ..... Inputs ................................................................................................
# Define inputs that are not on this mesh here:
typedef  ^                     InputType         ReKi              PositionXYZ        ::    -     -     "Array holding the input positions at a given timestep"  meters
typedef  ^                     ^                 lidar_InputType   lidar              -     -     -     "Lidar data"
typedef  ^                     ^                 ReKi              HubPosition        {3}   -     -     "position of the hub (inertial frame)" m
typedef  ^                     ^                 ReKi              HubOrientation     {3}{3} -    -     "orientation of the hub (direction cosine matrix)" -


# ..... Outputs ...............................................................................................
# Define outputs that are contained on the mesh here:
typedef  ^                     OutputType        ReKi              VelocityUVW        ::    -     -     "Array holding the U,V,W velocity for a given timestep"  meters/sec
typedef  ^                     OutputType        ReKi              AccelUVW           ::    -     -     "Array holding the U,V,W acceleration for a given timestep"  meters/sec
typedef  ^                     OutputType        ReKi              WriteOutput        :     -     -     "Array with values to output to file"
typedef  ^                     ^                 ReKi              DiskVel            {3}   -     -     "Vector holding the U,V,W average velocity of the disk"  meters/sec
typedef  ^                     ^                 ReKi              HubVel             {3}   -     -     "Vector holding the U,V,W velocity at the hub"  meters/sec
typedef  ^                     ^                 lidar_OutputType  lidar              -     -     -     "Lidar data"

# ..... States not used by this module ........................................................................
typedef  ^                     ContinuousStateType  ReKi           DummyContState     -     -     -     "Remove this variable if you have continuous states"
typedef  ^                     DiscreteStateType    ReKi           DummyDiscState     -     -     -     "Remove this variable if you have discrete states"
typedef  ^                     ConstraintStateType  ReKi           DummyConstrState   -     -     -     "Remove this variable if you have constraint states"
typedef  ^                     OtherStateType       ReKi           DummyOtherState    -     -     -     "Remove this variable if you have other states"


# ..... Misc/Optimization variables............................................................................
# Define any data that are used only for efficiency purposes (these variables are not associated with time):
#   e.g. indices for searching in an array, large arrays that are local variables in any routine called multiple times, etc.
typedef  ^                     MiscVarType       ReKi              AllOuts            :     -     -     "An array holding the value of all of the calculated (not only selected) output
typedef  ^                     ^                 ReKi              WindViUVW          ::    -     -     "List of UVW velocities for wind velocity measurements, 3xNWindVel. corresponds
typedef  ^                     ^                 ReKi              WindAiUVW          ::    -     -     "List of UVW accelerations for wind acceleration measurements, 3xNWindVel. corre
typedef  ^                     ^                 InflowWind_InputType  u_Avg          -     -     -     "inputs for computing rotor-averaged values"  -
typedef  ^                     ^                 InflowWind_OutputType y_Avg          -     -     -     "outputs for computing rotor-averaged values"  -
typedef  ^                     ^                 InflowWind_InputType  u_Hub          -     -     -     "inputs for computing hub values"  -
typedef  ^                     ^                 InflowWind_OutputType y_Hub          -     -     -     "outputs for computing hub values"  -
```

Figure B.4: `InflowWind.txt` file (part 2) following the modifications made. The orange boxes indicate where the `InflowWind.txt` file receives information from the `LIDAR.txt file`.

```
485      !------------------------------------------------------------------------------------------------
486      !> Read the _Parameters for LIDAR section
487      !------------------------------------------------------------------------------------------------
488
489     CurLine = CurLine + 1  ! Skip section break
490
491        ! LIDAR Sensor Type
492      CALL ParseVar( InFileInfo, CurLine, "SensorType", InputFileData%SensorType, TmpErrStat, TmpErrMsg, UnEc )
493      CALL SetErrStat( TmpErrStat, TmpErrMsg, ErrStat, ErrMsg, RoutineName )
494      IF (Failed()) RETURN
495
496        ! Number of Range Gates
497      CALL ParseVar( InFileInfo, CurLine, "NumPulseGate", InputFileData%NumPulseGate, TmpErrStat, TmpErrMsg, UnEc )
498      CALL SetErrStat( TmpErrStat, TmpErrMsg, ErrStat, ErrMsg, RoutineName )
499      IF (Failed()) RETURN
500
501        ! Pulse Gate Spacing
502      CALL ParseVar( InFileInfo, CurLine, "PulseSpacing", InputFileData%PulseSpacing, TmpErrStat, TmpErrMsg, UnEc )
503      CALL SetErrStat( TmpErrStat, TmpErrMsg, ErrStat, ErrMsg, RoutineName )
504      IF (Failed()) RETURN
505
506        ! NumBeam: Number of points to output the lidar measured wind velocity (1 to 5)
507      CALL ParseVar( InFileInfo, CurLine, "NumBeam", InputFileData%NumBeam, TmpErrStat, TmpErrMsg, UnEc )
508      CALL SetErrStat( TmpErrStat, TmpErrMsg, ErrStat, ErrMsg, RoutineName )
509      if (Failed()) return
510
511        ! Before proceeding, make sure that NumBeam makes sense
512      IF ((InputFileData%SensorType == 1) .and. (InputFileData%NumBeam < 1 .OR. InputFileData%NumBeam > 5)) THEN
513         CALL SetErrStat( ErrID_Fatal, 'NumBeam must be greater than or equal to one and less than 6.', &
514                          ErrStat, ErrMsg, RoutineName )
515         RETURN
516      ELSE
517              ! Allocate space for the output location arrays:
518         CALL AllocAry( InputFileData%FocalDistanceX, InputFileData%NumBeam, 'NumBeam', TmpErrStat, TmpErrMsg )
519         CALL SetErrStat( TmpErrStat, TmpErrMsg, ErrStat, ErrMsg, RoutineName )
520         CALL AllocAry( InputFileData%FocalDistanceY, InputFileData%NumBeam, 'NumBeam', TmpErrStat, TmpErrMsg )
521         CALL SetErrStat( TmpErrStat, TmpErrMsg, ErrStat, ErrMsg, RoutineName )
522         CALL AllocAry( InputFileData%FocalDistanceZ, InputFileData%NumBeam, 'NumBeam', TmpErrStat, TmpErrMsg )
523         CALL SetErrStat( TmpErrStat, TmpErrMsg, ErrStat, ErrMsg, RoutineName )
524         if (Failed()) return
525      ENDIF
526
527       ! Focal Distance X
528      CALL ParseAry( InFileInfo, CurLine, 'FocalDistanceX', InputFileData%FocalDistanceX, InputFileData%NumBeam, Tmp
529      CALL SetErrStat( TmpErrStat, TmpErrMsg, ErrStat, ErrMsg, RoutineName )
530      if (Failed()) return
531
532       ! Focal Distance Y
533      CALL ParseAry( InFileInfo, CurLine, 'FocalDistanceY', InputFileData%FocalDistanceY, InputFileData%NumBeam, Tmp
534      CALL SetErrStat( TmpErrStat, TmpErrMsg, ErrStat, ErrMsg, RoutineName )
535      if (Failed()) return
536
537       ! Focal Distance Z
538      CALL ParseAry( InFileInfo, CurLine, 'FocalDistanceZ', InputFileData%FocalDistanceZ, InputFileData%NumBeam, Tmp
539      CALL SetErrStat( TmpErrStat, TmpErrMsg, ErrStat, ErrMsg, RoutineName )
540      if (Failed()) return
541
542       ! Rotor Apex Offset Position
543      CALL ParseAry( InFileInfo, CurLine, "RotorApexOffsetPos", InputFileData%RotorApexOffsetPos, 1, TmpErrStat, Tmp
544      CALL SetErrStat( TmpErrStat, TmpErrMsg, ErrStat, ErrMsg, RoutineName )
545      IF (Failed()) RETURN
546
547       ! URefIni
548      CALL ParseVar( InFileInfo, CurLine, "URefLid", InputFileData%URefLid, TmpErrStat, TmpErrMsg, UnEc )
549      CALL SetErrStat( TmpErrStat, TmpErrMsg, ErrStat, ErrMsg, RoutineName )
550      IF (Failed()) RETURN
551
552       ! Measurement Interval
553      CALL ParseVar( InFileInfo, CurLine, "MeasurementInterval", InputFileData%MeasurementInterval, TmpErrStat, TmpE
554      CALL SetErrStat( TmpErrStat, TmpErrMsg, ErrStat, ErrMsg, RoutineName )
555      IF (Failed()) RETURN
556
557       ! Lidar Radial Vel
558      CALL ParseLoVar( InFileInfo, CurLine, "LidRadialVel", InputFileData%LidRadialVel, TmpErrStat, TmpErrMsg, UnEc
559      CALL SetErrStat( TmpErrStat, TmpErrMsg, ErrStat, ErrMsg, RoutineName )
560      IF (Failed()) RETURN
561
562       ! Consider Hub Motion
563      CALL ParseVar( InFileInfo, CurLine, "ConsiderHubMotion", InputFileData%ConsiderHubMotion, TmpErrStat, TmpErrMs
564      CALL SetErrStat( TmpErrStat, TmpErrMsg, ErrStat, ErrMsg, RoutineName )
565      IF (Failed()) RETURN
```

Figure B.5: Code added within the `InflowWind_Subs.f90` script to allow for the parsing of LIDAR configuration inputs from the InflowWind input file.

```
1302      IF (p%lidar%SensorType == SensorType_SinglePoint) THEN
1303         DO I=p%lidar%NumBeam+1,5
1304            InvalidOutput( WindMeas(I) ) = .TRUE.
1305         END DO
1306      ELSE
1307         DO I=p%lidar%NumPulseGate+1,5
1308            InvalidOutput( WindMeas(I) ) = .TRUE.
1309         END DO
1310      END IF

1534   IF ( p%lidar%SensorType == SensorType_SinglePoint) THEN
1535        DO I = 1,MIN(5, p%lidar%NumBeam )
1536           m%AllOuts( WindMeas(I) ) = y%lidar%LidSpeed(I)
1537        END DO
1538     ELSE
1539        DO I = 1,MIN(5, p%lidar%NumPulseGate )
1540          m%AllOuts( WindMeas(I) ) = y%lidar%LidSpeed(I)
1541       END DO
1542     END IF
```

Figure B.6: Code added within the `InflowWind_Subs.f90` script to allow for the allocation of LIDAR measurement outputs.

```
------- InflowWind v3.01.* INPUT FILE ------------------------------------------------------------------------
IEA 15 MW Offshore Reference Turbine
--------------------------------------------------------------------------------------------------------------
False                   Echo        - Echo input data to <RootName>.ech (flag)
3                       WindType       - switch for wind file type (1=steady; 2=uniform; 3=binary TurbSim FF; 4=binary Bladed-style FF; 5=HAWC for
0.0                     PropagationDir - Direction of wind propagation (meteoroligical rotation from aligned with X (positive rotates towards -Y)
0                       VFlowAng       - Upflow angle (degrees) (not used for native Bladed format WindType=7)
True                    VelInterpCubic - Use cubic interpolation for velocity in time (false=linear, true=cubic) [Used with WindType=2,3,4,5,7]
1                       NWindVel    - Number of points to output the wind velocity    (0 to 9)
0.0                     WindVxiList - List of coordinates in the inertial X direction (m)
0.0                     WindVyiList - List of coordinates in the inertial Y direction (m)
150.0                    WindVziList - List of coordinates in the inertial Z direction (m)
================== Parameters for Steady Wind Conditions [used only for WindType = 1] =========================
10.0                    HWindSpeed  - Horizontal windspeed                            (m/s)
150.0                   RefHt       - Reference height for horizontal wind speed      (m)
0.12                    PLexp       - Power law exponent                              (-)
================== Parameters for Uniform wind file   [used only for WindType = 2] ============================
"Wind/NoShr_9-14_Inc1_50s.wnd" Filename_Uni    - Filename of time series data for uniform wind field.      (-)
150.0                   RefHt_Uni       - Reference height for horizontal wind speed               (m)
125.88                  RefLength   - Reference length for linear horizontal and vertical sheer (-)
================== Parameters for Binary TurbSim Full-Field files   [used only for WindType = 3] ==============
"DLC1.1_21ms_Seed4.bts" FileName_BTS    - Name of the Full field wind file to use (.bts)
================== Parameters for Binary Bladed-style Full-Field files   [used only for WindType = 4] =========
"Wind/90m_12mps_twr"    FilenameRoot - Rootname of the full-field wind file to use (.wnd, .sum)
False                   TowerFile   - Have tower file (.twr) (flag)
================== Parameters for HAWC-format binary files  [Only used with WindType = 5] =====================
"wasp/Output/basic_5u.bin" FileName_u  - name of the file containing the u-component fluctuating wind (.bin)
"wasp/Output/basic_5v.bin" FileName_v  - name of the file containing the v-component fluctuating wind (.bin)
"wasp/Output/basic_5w.bin" FileName_w  - name of the file containing the w-component fluctuating wind (.bin)
64                      nx          - number of grids in the x direction (in the 3 files above) (-)
32                      ny          - number of grids in the y direction (in the 3 files above) (-)
32                      nz          - number of grids in the z direction (in the 3 files above) (-)
16.0                    dx          - distance (in meters) between points in the x direction    (m)
3.0                     dy          - distance (in meters) between points in the y direction    (m)
3.0                     dz          - distance (in meters) between points in the z direction    (m)
150.0                   RefHt_Hawc  - reference height; the height (in meters) of the vertical center of the grid (m)
-------------   Scaling parameters for turbulence   ----------------------------------------------------------
2                       ScaleMethod - Turbulence scaling method   [0 = none, 1 = direct scaling, 2 = calculate scaling factor based on a desired s
1.0                     SFx         - Turbulence scaling factor for the x direction (-)   [ScaleMethod=1]
1.0                     SFy         - Turbulence scaling factor for the y direction (-)   [ScaleMethod=1]
1.0                     SFz         - Turbulence scaling factor for the z direction (-)   [ScaleMethod=1]
1.2                     SigmaFx     - Turbulence standard deviation to calculate scaling from in x direction (m/s)    [ScaleMethod=2]
0.8                     SigmaFy     - Turbulence standard deviation to calculate scaling from in y direction (m/s)    [ScaleMethod=2]
0.2                     SigmaFz     - Turbulence standard deviation to calculate scaling from in z direction (m/s)    [ScaleMethod=2]
-------------   Mean wind profile parameters (added to HAWC-format files)   ----------------------------------
12.0                    URef        - Mean u-component wind speed at the reference height (m/s)
2                       WindProfile - Wind profile type (0=constant;1=logarithmic,2=power law)
0.2                     PLExp_Hawc  - Power law exponent (-) (used for PL wind profile type only)
0.03                    Z0          - Surface roughness length (m) (used for LG wind profile type only)
0                       XOffset     - Initial offset in +x direction (shift of wind box)
================== LIDAR Parameters ==========================================================================
1                       SensorType              - Switch for lidar configuration (0 = None, 1 = Single Point Beam(s), 2 = Continuous, 3 = Pulsed)
3                       NumPulseGate            - Number of lidar measurement gates (0-5) (used when SensorType = 3)
30                      PulseSpacing            - Distance between range gates (m) (used when SensorType = 3)
5                       NumBeam                 - Number of lidar measurement beams (0-5)(used when SensorType = 1)
-260,-260,-260,-260,-260  FocalDistanceX      - Focal distance co-ordinates of the lidar beam in the x direction (relative to hub height) (only f
0,60,0,-60,0              FocalDistanceY      - Focal distance co-ordinates of the lidar beam in the y direction (relative to hub height) (only f
0,0,-60,0,60               FocalDistanceZ      - Focal distance co-ordinates of the lidar beam in the z direction (relative to hub height) (only
0.0, 0.0, 0.0           RotorApexOffsetPos      - Offset of the lidar from hub height (m)
17                      URefLid                 - Reference average wind speed for the lidar[m/s]
1                       MeasurementInterval     - Time between each measurement [s]
False                   LidRadialVel            - TRUE => return radial component, FALSE => return 'x' direction estimate
1                       ConsiderHubMotion       - Flag whether to consider the hub motion's impact on Lidar measurements
====================== OUTPUT ================================================
False                   SumPrint    - Print summary data to <RootName>.IfW.sum (flag)
OutList      - The next line(s) contains a list of output parameters.  See OutListParameters.xlsx for a listing of available output channels, (-)
Wind1VelX
WindMeas1
WindMeas2
WindMeas3
WindMeas4
WindMeas5
END of input file (the word "END" must appear in the first 3 columns of this last OutList line)
---------------------------------------------------------------------------------------
```

Figure B.7: Modified InflowWind input file containing a new section for the specification of LIDAR simulator parameters, indicated by the green box.

```
216     ! initialize sensor data:
217     p%lidar%NumBeam = InputFileData%NumBeam
218     p%lidar%RotorApexOffsetPos = InputFileData%RotorApexOffsetPos
219     p%lidar%SensorType = InputFileData%SensorType
220     p%lidar%LidRadialVel   = InputFileData%LidRadialVel
221     p%lidar%NumPulseGate = InputFileData%NumPulseGate
222     p%lidar%FocalDistanceX =  InputFileData%FocalDistanceX
223     p%lidar%FocalDistanceY =  InputFileData%FocalDistanceY
224     p%lidar%FocalDistanceZ =  InputFileData%FocalDistanceZ
225     p%lidar%MeasurementInterval = InputFileData%MeasurementInterval
226     p%lidar%PulseSpacing = InputFileData%PulseSpacing
227     p%lidar%URefLid = InputFileData%URefLid
228     p%lidar%ConsiderHubMotion = InputFileData%ConsiderHubMotion
229
230
231     CALL Lidar_Init( InitInp, InputGuess, p, ContStates, DiscStates, ConstrStateGuess, OtherStates,   &
232                      y, m, TimeInterval, InitOutData, TmpErrStat, TmpErrMsg )
233        CALL SetErrStat( TmpErrStat, TmpErrMsg, ErrStat, ErrMsg, RoutineName )


753     !------------------------------
754     ! Outputs: OutputData%lidar%LidSpeed and OutputData%lidar%WtTrunc
755     !------------------------------
756
757        ! return sensor values
758     IF (p%lidar%SensorType /= SensorType_None) THEN
759
760        CALL Lidar_CalcOutput(Time, InputData, p, &
761                              ContStates, DiscStates, ConstrStates, OtherStates, &
762                              OutputData, m, TmpErrStat, TmpErrMsg )
763        CALL SetErrStat( TmpErrStat, TmpErrMsg, ErrStat, ErrMsg, RoutineName )
764
765     END IF
```

Figure B.8: Code added within the `InflowWind.f90` script to allow for the saving of LIDAR inputs as parameters as well as to initiate and calculate LIDAR outputs.

```
!**********************************************************************************************************************************
! LICENSING
! Copyright (C) 2015  CU Boulder
! Copyright (C) 2015-2016  National Renewable Energy Laboratory
!
!    Lidar module, a submodule of InflowWind
!
! Licensed under the Apache License, Version 2.0 (the "License");
! you may not use this file except in compliance with the License.
! You may obtain a copy of the License at
!
!     http://www.apache.org/licenses/LICENSE-2.0
!
! Unless required by applicable law or agreed to in writing, software
! distributed under the License is distributed on an "AS IS" BASIS,
! WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
! See the License for the specific language governing permissions and
! limitations under the License.
!
!**********************************************************************************************************************************
MODULE Lidar

   USE InflowWind_Subs
   USE InflowWind_Types
   USE Lidar_Types

   IMPLICIT NONE

   PRIVATE

   TYPE(ProgDesc), PARAMETER            :: Lidar_Ver = ProgDesc( 'Lidar', '', '' )
   CHARACTER(*),   PARAMETER            :: Lidar_Nickname = 'Lidar'


   REAL(ReKi),     PARAMETER            :: BeamRad      =  0.028
   REAL(ReKi),     PARAMETER            :: LsrWavLen    =  0.000001565                    ! Laser wavelength


! ==========================================================================================="


      ! ..... Public Subroutines ...................................................................................................

   PUBLIC :: Lidar_Init                           ! Initialization routine
   PUBLIC :: Lidar_End                            ! Ending routine (includes clean up)
   PUBLIC :: Lidar_CalcOutput                     ! Routine for computing outputs


!bjj:
!! @todo: add mesh to map nacelle rotor apex position in ElastoDyn to lidar location (possibly an array of lidars) in InflowWind
!! @todo: add input file (part of InflowWind input file)
!!    - number of lidars, type, location, number of pulse range gates, etc
!!    - initial measurement position(s)
!!    - scan pattern & associated values [remove this functionality from Matlab]
!! @todo: add subroutine for scanning patterns
!! future work:
!! @todo: do we want to know if the blade is in front of the lidar so we can return garbage to simulate that scenario, too?

CONTAINS
!----------------------------------------------------------------------------------------------------------------------------------
!> This routine is called at the start of the simulation to perform initialization steps.
!! The parameters are set here and not changed during the simulation.
!! The initial states and initial guess for the input are defined.
!! note that we're calling this with the InflowWind data types, so that data is INOUT instead of OUT
SUBROUTINE Lidar_Init( InitInp, u, p, x, xd, z, OtherState, y, m, Interval, InitOut, ErrStat, ErrMsg )

   TYPE(InflowWind_InitInputType),        INTENT(IN   )  :: InitInp     !< Input data for initialization routine
   TYPE(InflowWind_InputType),            INTENT(INOUT)  :: u           !< An initial guess for the input; input mesh must be defined
   TYPE(InflowWind_ParameterType),        INTENT(INOUT)  :: p           !< Parameters
   TYPE(InflowWind_ContinuousStateType),  INTENT(INOUT)  :: x           !< Initial continuous states
   TYPE(InflowWind_DiscreteStateType),    INTENT(INOUT)  :: xd          !< Initial discrete states
   TYPE(InflowWind_ConstraintStateType),  INTENT(INOUT)  :: z           !< Initial guess of the constraint states
   TYPE(InflowWind_OtherStateType),       INTENT(INOUT)  :: OtherState  !< Initial other states
   TYPE(InflowWind_MiscVarType),          INTENT(INOUT)  :: m           !< Misc variables for optimization (not copied in glue code)
   TYPE(InflowWind_OutputType),           INTENT(INOUT)  :: y           !< Initial system outputs (outputs are not calculated;
                                                                        !!   only the output mesh is initialized)
   REAL(DbKi),                            INTENT(IN )    :: Interval    !< Coupling interval in seconds: the rate that
                                                                        !!   (1) Lidar_UpdateStates() is called in loose coupling &
                                                                        !!   (2) Lidar_UpdateDiscState() is called in tight coupling.
                                                                        !!   Input is the suggested time from the glue code;
                                                                        !!   Output is the actual coupling interval that will be used
                                                                        !!   by the glue code.
```

Figure B.9: Complete modified `LIDAR.f90` script (part 1).

```
83      TYPE(InflowWind_InitOutputType),       INTENT(INOUT)  :: InitOut     !< Output for initialization routine
84      INTEGER(IntKi),                        INTENT(  OUT)  :: ErrStat     !< Error status of the operation
85      CHARACTER(*),                          INTENT(  OUT)  :: ErrMsg      !< Error message if ErrStat /= ErrID_None
86
87         ! Temporary variables for error handling
88       INTEGER(IntKi)                                        ::  TmpErrStat          !< temporary error message
89       CHARACTER(ErrMsgLen)                                  ::  TmpErrMsg
90
91         ! local variables
92      INTEGER(IntKi)                                       :: IBeam
93      INTEGER(IntKi)                                       :: ErrStat2    ! temporary Error status of the operation
94      CHARACTER(ErrMsgLen)                                 :: ErrMsg2     ! temporary Error message if ErrStat /= ErrID_None
95
96      CHARACTER(*),   PARAMETER                            :: RoutineName = 'Lidar_Init'
97
98      REAL(ReKi)                                           :: TempWindSpeed(3)
99
100        ! Initialize variables
101
102     ErrStat = ErrID_None
103     ErrMsg  = ""
104
105        ! Check for errors in the InflowWind Input File
106
107        ! Make sure that NumPulseGate makes sense
108      IF ( (p%lidar%SensorType == 3) .and. (p%lidar%NumPulseGate < 0 .OR. p%lidar%NumPulseGate > 5) ) THEN
109        CALL SetErrStat( ErrID_Fatal, 'NumPulseGate must be greater than or equal to zero and less than 5.', &
110                         ErrStat, ErrMsg, RoutineName )
111        RETURN
112      ENDIF
113
114         ! Make sure that multiple beams are only used when using single point beams
115      IF ( p%lidar%NumBeam > 1 .AND. p%lidar%SensorType > 1) THEN
116        CALL SetErrStat( ErrID_Fatal, 'Multiple beams can only be used with single point lidar', &
117                         ErrStat, ErrMsg, RoutineName )
118        RETURN
119      ENDIF
120
121     CALL AllocAry(p%lidar%MsrPosition , 3, p%lidar%NumBeam, 'Array for measurement coordinates',
122                                                             & TmpErrStat, TmpErrMsg )
123     CALL SetErrStat( TmpErrStat, TmpErrMsg, ErrStat, ErrMsg, RoutineName)
124     IF ( ErrStat>= AbortErrLev ) RETURN
125
126
127     IF (p%lidar%SensorType == SensorType_None) THEN
128        p%lidar%NumPulseGate = 0
129     ELSEIF (p%lidar%SensorType == SensorType_SinglePoint) THEN
130        p%lidar%NumPulseGate = 1
131     ELSE
132
133           ! variables for both pulsed and continuous-wave lidars
134
135
136
137
138        p%lidar%SpatialRes     =  0.5_ReKi*p%lidar%URefLid*Interval
139        p%lidar%RayRangeSq     =  (Pi*(BeamRad**2)/LsrWavLen)**2
140
141
142
143
144        IF (p%lidar%SensorType == SensorType_ContinuousLidar) THEN
145
146           p%lidar%WtFnTrunc    = 0.02_ReKi
147           p%lidar%NumPulseGate = 1
148
149        ELSEIF (p%lidar%SensorType == SensorType_PulsedLidar) THEN
150
151           p%lidar%WtFnTrunc    = 0.01_ReKi
152
153
154              ! values for the WindCube
155           ! p%lidar%DeltaP        = 30.0_ReKi   !Replaced by pulse spacing
156           p%lidar%DeltaR        = 30.0_ReKi
157           ! p%lidar%PulseRangeOne = 50.0 ReKi   ! Replaced by the focal distance
158
159           p%lidar%r_p           = p%lidar%DeltaR/(2.0_ReKi*SQRT(LOG(2.0_ReKi)))
160
161        ELSE
162
163           CALL SetErrStat(ErrID_Fatal, "Invalid sensor type.", ErrStat, ErrMsg, RoutineName)
```

Figure B.10: Complete modified `LIDAR.f90` script (part 2). The green boxes indicate the parts of the code that were added or modified within this research project.

```
164            RETURN
165
166         END IF
167
168      END IF
169
170
171         !.....................................................................................
172         ! Define initial system states here:
173         !.....................................................................................
174
175      !x%lidar%DummyContState           = 0.0_ReKi
176      !xdlidar%%DummyDiscState          = 0.0_ReKi
177      !z%lidar%DummyConstrState         = 0.0_ReKi
178      !OtherState%lidar%DummyOtherState = 0.0_ReKi
179      !
180
181         !.....................................................................................
182         ! Define initial guess for the system inputs here:
183         !.....................................................................................
184
185      p%lidar%LidPosition = InitInp%lidar%HubPosition
186      DO IBeam = 1,p%lidar%NumBeam
187         p%lidar%MsrPosition(:,IBeam) = (/ p%lidar%FocalDistanceX(IBeam), p%lidar%FocalDistanceY(IBeam),
188                                                                         & p%lidar%FocalDistanceZ(IBeam) /)
189      END DO
190      u%lidar%PulseLidEl  = 0.0_ReKi
191      u%lidar%PulseLidAz  = 0.0_ReKi
192
193
194         !.....................................................................................
195         ! Define system output initializations (set up mesh) here:
196         !.....................................................................................
197      !CALL AllocAry( y%WriteOutput, p%NumOuts, 'WriteOutput', ErrStat2, ErrMsg2 )
198      !   CALL SetErrStat(ErrStat2,ErrMsg2,ErrStat,ErrMsg,RoutineName)
199      !   IF (ErrStat >= AbortErrLev) RETURN
200      !y%WriteOutput = 0
201
202      IF (p%lidar%SensorType == SensorType_SinglePoint) THEN
203         CALL AllocAry( y%lidar%LidSpeed, p%lidar%NumBeam, 'y%lidar%LidSpeed', ErrStat2, ErrMsg2 )
204         CALL SetErrStat(ErrStat2, ErrMsg2, ErrStat, ErrMsg, RoutineName );
205
206         !CALL AllocAry( y%LidErr, p%NumPulseGate, 'y%LidErr', ErrStat2, ErrMsg2 )
207         !   CALL SetErrStat(ErrStat2, ErrMsg2, ErrStat, ErrMsg, RoutineName );
208
209         CALL AllocAry( y%lidar%WtTrunc, p%lidar%NumBeam, 'y%lidar%WtTrunc', ErrStat2, ErrMsg2 )
210         CALL SetErrStat(ErrStat2, ErrMsg2, ErrStat, ErrMsg, RoutineName );
211
212         CALL AllocAry( y%lidar%MsrPositionsX, p%lidar%NumBeam, 'y%lidar%MsrPositionsX', ErrStat2, ErrMsg2 )
213         CALL SetErrStat(ErrStat2, ErrMsg2, ErrStat, ErrMsg, RoutineName );
214
215         CALL AllocAry( y%lidar%MsrPositionsY, p%lidar%NumBeam, 'y%lidar%MsrPositionsY', ErrStat2, ErrMsg2 )
216         CALL SetErrStat(ErrStat2, ErrMsg2, ErrStat, ErrMsg, RoutineName );
217
218         CALL AllocAry( y%lidar%MsrPositionsZ, p%lidar%NumBeam, 'y%lidar%MsrPositionsZ', ErrStat2, ErrMsg2 )
219         CALL SetErrStat(ErrStat2, ErrMsg2, ErrStat, ErrMsg, RoutineName );
220      ELSEIF (p%lidar%NumPulseGate > 0) then    ! the Cray Fortran compiler doesn't like allocating size zero
221         CALL AllocAry( y%lidar%LidSpeed, p%lidar%NumPulseGate, 'y%lidar%LidSpeed', ErrStat2, ErrMsg2 )
222         CALL SetErrStat(ErrStat2, ErrMsg2, ErrStat, ErrMsg, RoutineName );
223
224         !CALL AllocAry( y%LidErr, p%NumPulseGate, 'y%LidErr', ErrStat2, ErrMsg2 )
225         !CALL SetErrStat(ErrStat2, ErrMsg2, ErrStat, ErrMsg, RoutineName );
226
227         CALL AllocAry( y%lidar%WtTrunc, p%lidar%NumPulseGate, 'y%lidar%WtTrunc', ErrStat2, ErrMsg2 )
228         CALL SetErrStat(ErrStat2, ErrMsg2, ErrStat, ErrMsg, RoutineName );
229
230         CALL AllocAry( y%lidar%MsrPositionsX, p%lidar%NumPulseGate, 'y%lidar%MsrPositionsX', ErrStat2, ErrMsg2 )
231         CALL SetErrStat(ErrStat2, ErrMsg2, ErrStat, ErrMsg, RoutineName );
232
233         CALL AllocAry( y%lidar%MsrPositionsY, p%lidar%NumPulseGate, 'y%lidar%MsrPositionsY', ErrStat2, ErrMsg2 )
234         CALL SetErrStat(ErrStat2, ErrMsg2, ErrStat, ErrMsg, RoutineName );
235
236         CALL AllocAry( y%lidar%MsrPositionsZ, p%lidar%NumPulseGate, 'y%lidar%MsrPositionsZ', ErrStat2, ErrMsg2 )
237         CALL SetErrStat(ErrStat2, ErrMsg2, ErrStat, ErrMsg, RoutineName );
238      ENDIF
239
240
241      IF (ErrStat >= AbortErrLev) RETURN
242      if (allocated(y%lidar%LidSpeed))  y%lidar%LidSpeed = 0.0
243      if (allocated(y%lidar%WtTrunc ))  y%lidar%WtTrunc  = 0.0
244
```

Figure B.11: Complete modified LIDAR.f90 script (part 3). The green boxes indicate the parts of the code that were added or modified within this research project.

```
245         !.....................................................................................
246         ! Define initialization-routine output here:
247         !.....................................................................................
248
249
250      RETURN
251
252   END SUBROUTINE Lidar_Init
253   !----------------------------------------------------------------------------------------------------------------------------------
254   !> This routine is called at the end of the simulation.
255   SUBROUTINE Lidar_End( u, p, x, xd, z, OtherState, y, m, ErrStat, ErrMsg )
256   !.....................................................................................................................................
257
258         TYPE(InflowWind_InputType),           INTENT(INOUT)  :: u           !< System inputs
259         TYPE(InflowWind_ParameterType),       INTENT(INOUT)  :: p           !< Parameters
260         TYPE(InflowWind_ContinuousStateType), INTENT(INOUT)  :: x           !< Continuous states
261         TYPE(InflowWind_DiscreteStateType),   INTENT(INOUT)  :: xd          !< Discrete states
262         TYPE(InflowWind_ConstraintStateType), INTENT(INOUT)  :: z           !< Constraint states
263         TYPE(InflowWind_OtherStateType),      INTENT(INOUT)  :: OtherState  !< Other/optimization states
264         TYPE(InflowWind_OutputType),          INTENT(INOUT)  :: y           !< System outputs
265         TYPE(InflowWind_MiscVarType),         INTENT(INOUT)  :: m           !< Misc variables for optimization (not copied in glue c
266         INTEGER(IntKi),                       INTENT(  OUT)  :: ErrStat     !< Error status of the operation
267         CHARACTER(*),                         INTENT(  OUT)  :: ErrMsg      !< Error message if ErrStat /= ErrID_None
268
269
270
271            ! Initialize ErrStat
272
273         ErrStat = ErrID_None
274         ErrMsg  = ""
275
276            ! Close files here:
277
278
279            ! Destroy the input data:
280
281         CALL Lidar_DestroyInput( u%lidar, ErrStat, ErrMsg )
282
283
284            ! Destroy the parameter data:
285
286         CALL Lidar_DestroyParam( p%lidar, ErrStat, ErrMsg )
287
288
289            ! Destroy the state data:
290
291         !CALL Lidar_DestroyContState(   x,           ErrStat, ErrMsg )
292         !CALL Lidar_DestroyDiscState(   xd,          ErrStat, ErrMsg )
293         !CALL Lidar_DestroyConstrState( z,           ErrStat, ErrMsg )
294         !CALL Lidar_DestroyOtherState(  OtherState,  ErrStat, ErrMsg )
295
296
297            ! Destroy the output data:
298
299         CALL Lidar_DestroyOutput( y%lidar, ErrStat, ErrMsg )
300
301
302
303
304   END SUBROUTINE Lidar_End
305   !----------------------------------------------------------------------------------------------------------------------------------
306   !> Routine for computing outputs, used in both loose and tight coupling.
307   !! @note this breaks the framework because we're passing the IfW types instead of the Lidar types... this is necessary to get
308   !!    appropriate wind speeds for the lidar measurements.
309   SUBROUTINE Lidar_CalcOutput( t, u, p, x, xd, z, OtherState, y, m, ErrStat, ErrMsg )
310   !.....................................................................................................................................
311
312      REAL(DbKi),                           INTENT(IN   )  :: t           !< Current simulation time in seconds
313      TYPE(InflowWind_InputType),           INTENT(IN   )  :: u           !< Inputs at t
314      TYPE(InflowWind_ParameterType),       INTENT(IN   )  :: p           !< Parameters
315      TYPE(InflowWind_ContinuousStateType), INTENT(IN   )  :: x           !< Continuous states at t
316      TYPE(InflowWind_DiscreteStateType),   INTENT(IN   )  :: xd          !< Discrete states at t
317      TYPE(InflowWind_ConstraintStateType), INTENT(IN   )  :: z           !< Constraint states at t
318      TYPE(InflowWind_OtherStateType),      INTENT(IN   )  :: OtherState  !< Other/optimization states
319      TYPE(InflowWind_OutputType),          INTENT(INOUT)  :: y           !< Outputs computed at t (Input only so that mesh
320                                                                          !!   nectivity information does not have to be rec
321      TYPE(InflowWind_MiscVarType),         INTENT(INOUT)  :: m           !< Misc variables for optimization (not copied in
322      INTEGER(IntKi),                       INTENT(  OUT)  :: ErrStat     !< Error status of the operation
323      CHARACTER(*),                         INTENT(  OUT)  :: ErrMsg      !< Error message if ErrStat /= ErrID_None
324
```

Figure B.12: Complete modified `LIDAR.f90` script (part 4).

```
325         ! Local variables
326
327      REAL(ReKi)                                    :: FocDist             ! Focus Distance of the laser
328      REAL(ReKi)                                    :: FocDistMod          ! Focus Distance of the laser?
329      REAL(ReKi)                                    :: LidPhi              ! angle used with LidTheta to describe the directi
330      REAL(ReKi)                                    :: LidTheta            ! angle used with LidPhi to describe the direction
331      REAL(ReKi)                                    :: LidRange            ! lidar range
332
333      REAL(ReKi)                                    :: WtFuncSum           ! sum of weight function, used to normalize summat
334      REAL(ReKi)                                    :: LidWt               ! The weighting function value
335      REAL(ReKi)                                    :: LidWtMax            ! maximum weighting function value?
336      REAL(ReKi)                                    :: LidWtRatio          ! LidWt/LidWtMax
337      REAL(ReKi)                                    :: LidDirUnVec(3)      ! lidar look direction unit vector
338
339      REAL(ReKi)                                    :: Distance(3)         ! distance vector between input measurement and li
340
341      TYPE(InflowWind_InputType)                    :: Input               ! position where wind speed should be returned
342      TYPE(InflowWind_OutputType)                   :: Output              ! velocity at Input%Position
343      REAL(ReKi)                                    ::  LidPosition(3)     ! Lidar Position
344      REAL(ReKi)                                    ::  LidPosition_N(3)   !Transformed Lidar Position
345      REAL(ReKi)                                    ::  LidarMsrPosition(3)    !Transformed Lidar Position
346      REAL(ReKi)                                    ::  MeasurementCurrentStep
347
348
349      REAL(ReKi)                                    :: OutputVelocity(3)
350
351      INTEGER(IntKi)                                :: IBeam
352      INTEGER(IntKi)                                :: IRangeGt
353      INTEGER(IntKi)                                :: ErrStat2
354      CHARACTER(ErrMsgLen)                          :: ErrMsg2
355
356      CHARACTER(*), PARAMETER                       :: RoutineName = 'Lidar_CalcOutput'
357
358
359         ! Initialize ErrStat
360
361      ErrStat = ErrID_None
362      ErrMsg  = ""
363
364         MeasurementCurrentStep =  INT(t / p%lidar%MeasurementInterval)
365
366      IF ( (p%lidar%MeasurementInterval * MeasurementCurrentStep) /= t ) THEN
367          Output%VelocityUVW(:,1) = 0
368      RETURN
369      ENDIF
370
371         IF (p%lidar%ConsiderHubMotion == 1) THEN
372              LidPosition_N =  (/ u%lidar%HubDisplacementX, u%lidar%HubDisplacementY, u%lidar%HubDisplacementZ /) & ! rotor apex posit
373                                                                                    + p%lidar%RotorApexOffsetPos    ! lidar offset-fro
374              LidPosition   =  p%lidar%LidPosition + LidPosition_N
375         ELSE
376              LidPosition_N =  p%lidar%RotorApexOffsetPos
377              LidPosition   =  p%lidar%LidPosition + LidPosition_N
378         END IF
379
380      IF (p%lidar%SensorType == SensorType_None) RETURN
381
382       ! allocate arrays to compute outputs
383      CALL AllocAry(Input%PositionXYZ, 3,1, 'Input%PositionXYZ',ErrStat2, ErrMsg2)
384         CALL SetErrStat(ErrStat2, ErrMsg2, ErrStat, ErrMsg, RoutineName )
385
386      CALL AllocAry(Output%VelocityUVW, 3,1, 'Output%VelocityUVW',ErrStat2, ErrMsg2)
387         CALL SetErrStat(ErrStat2, ErrMsg2, ErrStat, ErrMsg, RoutineName )
388
389      IF (ErrStat >= AbortErrLev) THEN
390         CALL Cleanup()
391         RETURN
392      END IF
393
394      !.....................................................................................................................
395      ! Compute the outputs
396      !.....................................................................................................................
397
398
399      IF (p%lidar%SensorType == SensorType_SinglePoint) THEN
400         DO IBeam = 1,p%lidar%NumBeam
401           !get lidar speed at the focal point to see if it is out of bounds
402         Input%PositionXYZ(:,1) =  LidPosition + p%lidar%MsrPosition(:,IBeam)
403         CALL CalculateOutput( t, Input, p, x, xd, z, OtherState, Output, m, .FALSE., ErrStat2, ErrMsg2 )
404            CALL SetErrStat( ErrStat2, ErrMsg2, ErrStat, ErrMsg, RoutineName )
405
```

Figure B.13: Complete modified LIDAR.f90 script (part 5). The green boxes indicate the parts of the code that were added or modified within this research project.

```
406         y%lidar%LidSpeed(IBeam) = SQRT( DOT_PRODUCT(Output%VelocityUVW(:,1), Output%VelocityUVW(:,1)) )
407         y%lidar%WtTrunc  = 1.0_ReKi
408
409         y%lidar%MsrPositionsX(IBeam) = Input%PositionXYZ(1,1)
410         y%lidar%MsrPositionsY(IBeam) = Input%PositionXYZ(2,1)
411         y%lidar%MsrPositionsZ(IBeam) = Input%PositionXYZ(3,1)
412
413       END DO
414
415     ELSEIF (p%lidar%SensorType == SensorType_ContinuousLidar) THEN
416        !calculate the focal distance of the lidar as well as the modified focal distance so that the peak of
417        !the weighting func is at the intended focal distance
418
419        Distance   = p%lidar%MsrPosition(:,1)  - LidPosition
420        FocDist    = SQRT( DOT_PRODUCT( Distance, Distance ) ) !TwoNorm
421
422        IF(EqualRealNos(FocDist,0.0_ReKi)) THEN ! Avoid division-by-zero
423           y%lidar%LidSpeed = -99.0
424           y%lidar%WtTrunc  = 0.0
425           CALL SetErrStat(ErrID_Fatal,"Measurement position cannot be the same as the lidar position.",
426                                                                    & ErrStat, ErrMsg, RoutineName)
427           CALL Cleanup()
428           RETURN
429        END IF
430
431        FocDistMod = (p%lidar%RayRangeSq - SQRT(p%lidar%RayRangeSq**2 - 4*p%lidar%RayRangeSq*(FocDist**2)))/(2*FocDist);
432
433
434        !Find angles that the lidar is pointed at
435        LidPhi = ATAN(Distance(3)/SQRT(Distance(1)**2 + Distance(2)**2))
436        !LidTheta = ATAN(Distance(2)/ABS(Distance(1)))
437        LidTheta = ATAN2(Distance(2),-Distance(1))
438
439
440        !calculate the unit vector of the lidar look direction
441        LidDirUnVec = Distance / FocDist
442        LidWt = 1.0/(FocDist**2 + ((1 - FocDist/FocDistMod)**2)*p%lidar%RayRangeSq)
443        LidWtMax = LidWt
444        LidWtRatio = 1.0_ReKi !LidWt/LidWtMax
445
446        !get lidar speed at the focal point to see if it is out of bounds
447        Input%PositionXYZ(:,1) = LidPosition + p%lidar%MsrPosition(:,1)
448
449        y%lidar%MsrPositionsX(1) = LidPosition(1) + p%lidar%MsrPosition(1,1)
450        y%lidar%MsrPositionsY(1) = LidPosition(2) + p%lidar%MsrPosition(2,1)
451        y%lidar%MsrPositionsZ(1) = LidPosition(3) + p%lidar%MsrPosition(3,1)
452
453        CALL CalculateOutput( t, Input, p, x, xd, z, OtherState, Output, m, .FALSE., ErrStat2, ErrMsg2 )
454           CALL SetErrStat( ErrStat2, ErrMsg2, ErrStat, ErrMsg, RoutineName )
455
456           !if out of bounds
457        IF (ErrStat >= AbortErrLev) THEN
458           !y%lidar%LidErr = 1
459           y%lidar%LidSpeed = -99.0
460           CALL Cleanup()
461           RETURN !escape function
462        ENDIF
463
464        y%lidar%LidSpeed = LidWt*DOT_PRODUCT(-1*LidDirUnVec,Output%VelocityUVW(:,1))
465
466        WtFuncSum = LidWt
467        y%lidar%WtTrunc = p%lidar%WtFnTrunc
468
469        !initialize lidar range
470        LidRange = 0.
471
472
473
474        !calculate the weighted lidar returns
475        DO
476
477           !escape loop if desired truncation point has been reached
478           IF (LidWtRatio < p%lidar%WtFnTrunc) THEN
479              EXIT
480           ENDIF
481
482
483           !calculate range of current beam point
484           LidRange = LidRange + p%lidar%SpatialRes
485
```

Figure B.14: Complete modified LIDAR.f90 script (part 6). The green boxes indicate the parts of the code that were added or modified within this research project.

```
            LidWt = 1.0/((FocDist + LidRange)**2 + ((1 - (FocDist + LidRange)/FocDistMod)**2)*p%lidar%RayRangeSq)
            LidWtRatio = LidWt/LidWtMax


            !trunc point is behind lidar
            IF (LidRange > FocDist) THEN
               IF (NWTC_VerboseLevel == NWTC_Verbose) &
                  CALL SetErrStat( ErrID_Info, "Lidar truncation point is behind the lidar. Truncation ratio is "
                       //trim(num2lstr(LidWtRatio))//'.', ErrStat, ErrMsg, RoutineName)  ! set informational message about point be
               !y%LidErr = 3
               y%lidar%WtTrunc = LidWtRatio
               EXIT
            ENDIF


            !calculate points to scan for current beam point
            Input%PositionXYZ(3,1) = LidPosition(3) + SIN(LidPhi)*(LidRange + FocDist)
            Input%PositionXYZ(1,1) = LidPosition(1) - COS(LidTheta)*COS(LidPhi)*(LidRange + FocDist)
            Input%PositionXYZ(2,1) = LidPosition(2) + SIN(LidTheta)*COS(LidPhi)*(LidRange + FocDist)

            CALL CalculateOutput( t, Input, p, x, xd, z, OtherState, Output, m, .FALSE., ErrStat2, ErrMsg2 )
               IF (ErrStat2 >= AbortErrLev ) THEN !out of bounds
                  IF (NWTC_VerboseLevel == NWTC_Verbose) &
                     CALL SetErrStat( ErrID_Warn, "Lidar speed truncated. Truncation ratio is "//trim(num2lstr(LidWtRatio))//".",
                                                                                                  & ErrStat, ErrMsg, RoutineName )
               !y%LidErr = 2
                  y%lidar%WtTrunc = LidWtRatio
                  EXIT
               ENDIF

            OutputVelocity = Output%VelocityUVW(:,1)


            !calculate points to scan for current beam point
            Input%PositionXYZ(3,1) = LidPosition(3) + SIN(LidPhi)*(FocDist - LidRange)
            Input%PositionXYZ(1,1) = LidPosition(1) - COS(LidTheta)*COS(LidPhi)*(FocDist - LidRange)
            Input%PositionXYZ(2,1) = LidPosition(2) + SIN(LidTheta)*COS(LidPhi)*(FocDist - LidRange)

            CALL CalculateOutput( t, Input, p, x, xd, z, OtherState, Output, m, .FALSE., ErrStat2, ErrMsg2 )
               IF (ErrStat2 >= AbortErrLev) THEN !out of bounds
                  IF (NWTC_VerboseLevel == NWTC_Verbose) &
                     CALL SetErrStat( ErrID_Warn, "Lidar speed truncated. Truncation ratio is "//trim(num2lstr(LidWtRatio))//".",
                                                                                                  & ErrStat, ErrMsg, RoutineName )
               !y%lidar%LidErr = 2
                  y%lidar%WtTrunc = LidWtRatio
                  EXIT
               ENDIF


            y%lidar%LidSpeed = y%lidar%LidSpeed + LidWt*DOT_PRODUCT(-1*LidDirUnVec, OutputVelocity + Output%VelocityUVW(:,1))
            WtFuncSum = WtFuncSum + 2*LidWt


         END DO

         !Normalize the weighting function summation to 1

         IF ( p%lidar%LidRadialVel )  THEN
               !This detects the radial component
               y%lidar%LidSpeed = y%lidar%LidSpeed/WtFuncSum
         ELSE
               !This returns the 'x' component estimate
               y%lidar%LidSpeed = -1*y%lidar%LidSpeed/(WtFuncSum*LidDirUnVec(1))
         ENDIF

      ELSE !p%SensorType == SensorType_PulsedLidar



         LidDirUnVec(1) = -1*COS(u%lidar%PulseLidEl)
         LidDirUnVec(2) = SIN(u%lidar%PulseLidEl)*SIN(u%lidar%PulseLidAz)
         LidDirUnVec(3) = SIN(u%lidar%PulseLidEl)*COS(u%lidar%PulseLidAz)


         DO IRangeGt = 1,p%lidar%NumPulseGate

            !y%lidar%LidErr(IRangeGt) = 0

             LidPosition(2) = LidPosition(2) + p%lidar%MsrPosition(2,1)
             LidPosition(3) = LidPosition(3) + p%lidar%MsrPosition(3,1)
```

Figure B.15: Complete modified LIDAR.f90 script (part 7). The green boxes indicate the parts of the code that were added or modified within this research project.

```
566
567             !get lidar speed at the focal point to see if it is out of bounds
568             Input%PositionXYZ(:,1) = LidPosition + LidDirUnVec*(-p%lidar%MsrPosition(1,1) - (IRangeGt-1)*p%lidar%PulseSpacing)
569
570             y%lidar%MsrPositionsX(IRangeGt) = LidPosition(1) + LidDirUnVec(1)*(-p%lidar%MsrPosition(1,1)
571                                                                 & - (IRangeGt-1)*p%lidar%PulseSpacing)
572             y%lidar%MsrPositionsY(IRangeGt) = LidPosition(2) + p%lidar%MsrPosition(2,1)
573             y%lidar%MsrPositionsZ(IRangeGt) = LidPosition(3) + p%lidar%MsrPosition(3,1)
574
575             CALL CalculateOutput( t, Input, p, x, xd, z, OtherState, Output, m, .FALSE., ErrStat2, ErrMsg2 )
576                CALL SetErrStat( ErrStat2, ErrMsg2, ErrStat, ErrMsg, RoutineName )
577
578             LidWt = NWTC_ERF((p%lidar%PulseSpacing/2)/p%lidar%r_p)/p%lidar%PulseSpacing
579             LidWtMax = LidWt
580             LidWtRatio = 1.0_ReKi !LidWt/LidWtMax
581
582
583             !if out of bounds
584             IF (ErrStat2 >= AbortErrLev) THEN
585                !y%LidErr(IRangeGt) = 1
586                y%lidar%LidSpeed(IRangeGt) = -99
587                CALL Cleanup()
588                RETURN !escape function
589             ENDIF
590
591             y%lidar%LidSpeed(IRangeGt) = LidWt*DOT_PRODUCT(-1*LidDirUnVec,Output%VelocityUVW(:,1))
592
593             WtFuncSum = LidWt
594             y%lidar%WtTrunc(IRangeGt) = p%lidar%WtFnTrunc
595
596             !initialize lidar range
597             LidRange = 0.
598
599             DO
600                !escape loop if desired truncation point has been reached
601                IF (LidWtRatio < p%lidar%WtFnTrunc) THEN
602                      EXIT
603                ENDIF
604
605                !calculate range of current beam point
606                LidRange = LidRange + p%lidar%SpatialRes
607
608                LidWt = (NWTC_ERF((LidRange + p%lidar%PulseSpacing/2.)/p%lidar%r_p) - NWTC_ERF((LidRange -
609                                   & p%lidar%PulseSpacing/2.)/p%lidar%r_p))/(2.*p%lidar%PulseSpacing)
610                LidWtRatio = LidWt/LidWtMax
611
612
613                   !trunc point is behind lidar
614                IF (LidRange > (p%lidar%PulseRangeOne + (IRangeGt-1)*p%lidar%PulseSpacing)) THEN
615                   IF (NWTC_VerboseLevel == NWTC_Verbose) &
616                      CALL SetErrStat( ErrID_Info, "Lidar truncation point at gate "//trim(num2lstr(IRangeGt))//&
617                      " is behind the lidar. Truncation ratio is "& //trim(num2lstr(LidWtRatio))//'.', ErrStat, ErrMsg, RoutineName)
618                   !y%LidErr(IRangeGt) = 3
619                   y%lidar%WtTrunc(IRangeGt) = LidWtRatio
620                   EXIT
621                ENDIF
622
623
624                !calculate points to scan for current beam point
625                Input%PositionXYZ(:,1) = LidPosition + LidDirUnVec*(-p%lidar%MsrPosition(1,1) - (IRangeGt-1)*p%lidar%PulseSpacing
626                                                                                                  &  + LidRange)
627                CALL CalculateOutput( t, Input, p, x, xd, z, OtherState, Output, m, .FALSE., ErrStat2, ErrMsg2 )
628                   IF (ErrStat2 >= AbortErrLev) THEN !out of bounds
629                   IF (NWTC_VerboseLevel == NWTC_Verbose) &
630                      CALL SetErrStat( ErrID_Warn, "Lidar speed at gate "//trim(num2lstr(IRangeGt))//
631                      & " truncated. Truncation ratio is "//trim(num2lstr(LidWtRatio))//".", ErrStat, ErrMsg, RoutineName )
632                      !y%LidErr(IRangeGt) = 2
633                      y%lidar%WtTrunc(IRangeGt) = LidWtRatio
634                      EXIT
635                   ENDIF
636                OutputVelocity = Output%VelocityUVW(:,1)
637
638                !calculate points to scan for current beam point
639                Input%PositionXYZ(:,1) = LidPosition + LidDirUnVec*(-p%lidar%MsrPosition(1,1) - (IRangeGt-1)*p%lidar%PulseSpacing
640                                                                                                  & - LidRange)
641                CALL CalculateOutput( t, Input, p, x, xd, z, OtherState, Output, m, .FALSE., ErrStat2, ErrMsg2 )
642                   IF (ErrStat2 >= AbortErrLev) THEN !out of bounds
643                   IF (NWTC_VerboseLevel == NWTC_Verbose) &
644                      CALL SetErrStat( ErrID_Warn, "Lidar speed at gate "//trim(num2lstr(IRangeGt))//
645                      & " truncated. Truncation ratio is "//trim(num2lstr(LidWtRatio))//".", ErrStat, ErrMsg, RoutineName )
646                      !y%lidar%LidErr(IRangeGt) = 2
647                      y%lidar%WtTrunc(IRangeGt) = LidWtRatio
```

Figure B.16: Complete modified LIDAR.f90 script (part 8). The green boxes indicate the parts of the code that were added or modified within this research project.

```
                  EXIT
               ENDIF


            y%lidar%LidSpeed(IRangeGt) = y%lidar%LidSpeed(IRangeGt) + LidWt*DOT_PRODUCT(-1*LidDirUnVec,Output%VelocityUVW(:,1)
                                                                                                    & + OutputVelocity)
            WtFuncSum = WtFuncSum + 2*LidWt

         END DO


         IF ( p%lidar%LidRadialVel )  THEN
            !This detects the radial component
            y%lidar%LidSpeed(IRangeGt) = y%lidar%LidSpeed(IRangeGt)/WtFuncSum
         ELSE
            !This returns the 'x' component estimate
            y%lidar%LidSpeed(IRangeGt) = -1*y%lidar%LidSpeed(IRangeGt)/(LidDirUnVec(1)*WtFuncSum)
         ENDIF

      END DO

   END IF

   CALL Cleanup()

   RETURN
CONTAINS
   SUBROUTINE Cleanup()

      IF (ALLOCATED(Input%PositionXYZ)) DEALLOCATE(Input%PositionXYZ)
      IF (ALLOCATED(Output%VelocityUVW)) DEALLOCATE(Output%VelocityUVW)

   END SUBROUTINE Cleanup

END SUBROUTINE Lidar_CalcOutput
!----------------------------------------------------------------------------------------------------------------------------------
END MODULE Lidar
!**********************************************************************************************************************************
```

Figure B.17: Complete modified `LIDAR.f90` script (part 9).

```
   ! lidar
   Init%InData_IfW%lidar%Tmax                  = p_FAST%TMax
   Init%InData_IfW%lidar%HubPosition           = ED%y%HubPtMotion%Position(:,1)

   Init%InData_IfW%lidar%HubPosition = ED%y%HubPtMotion%Position(:,1)
```

Figure B.18: Code added within the `FAST_Subs.f90` script to initiate the transfer of the hub position from ElastoDyn to InflowWind.

```
IF (p_FAST%CompInflow == Module_IfW) THEN
   ! get Lidar position directly from hub mesh (may map meshes later)
   IfW%Input%lidar%HubDisplacementX = ED%y%HubPtMotion%TranslationDisp(1,1)
   IfW%Input%lidar%HubDisplacementY = ED%y%HubPtMotion%TranslationDisp(2,1)
   IfW%Input%lidar%HubDisplacementZ = ED%y%HubPtMotion%TranslationDisp(3,1)
```

Figure B.19: Code added within the `FAST_Solver.f90` script to set hub position outputs from ElastoDyn to InflowWind LIDAR inputs.

```
635         IF ( p_FAST%CompServo == Module_SrvD ) THEN !assign the number of gates to ServD
636            if (allocated(IfW%y%lidar%LidSpeed)) then    ! make sure we have the array allocated before setting it
637               CALL AllocAry(Init%InData_SrvD%LidSpeed, size(IfW%y%lidar%LidSpeed), 'Init%InData_SrvD%LidSpeed',     errStat2,
638               CALL SetErrStat(ErrStat2,ErrMsg2,ErrStat,ErrMsg,RoutineName)
639               Init%InData_SrvD%LidSpeed = IfW%y%lidar%LidSpeed
640            endif
641            if (allocated(IfW%y%lidar%MsrPositionsX)) then    ! make sure we have the array allocated before setting it
642               CALL AllocAry(Init%InData_SrvD%MsrPositionsX, size(IfW%y%lidar%MsrPositionsX), 'Init%InData_SrvD%MsrPositionsX',
643               CALL SetErrStat(ErrStat2,ErrMsg2,ErrStat,ErrMsg,RoutineName)
644               Init%InData_SrvD%MsrPositionsX = IfW%y%lidar%MsrPositionsX
645            endif
646            if (allocated(IfW%y%lidar%MsrPositionsY)) then    ! make sure we have the array allocated before setting it
647               CALL AllocAry(Init%InData_SrvD%MsrPositionsY, size(IfW%y%lidar%MsrPositionsY), 'Init%InData_SrvD%MsrPositionsY',
648               CALL SetErrStat(ErrStat2,ErrMsg2,ErrStat,ErrMsg,RoutineName)
649               Init%InData_SrvD%MsrPositionsY = IfW%y%lidar%MsrPositionsY
650            endif
651            if (allocated(IfW%y%lidar%MsrPositionsZ)) then    ! make sure we have the array allocated before setting it
652               CALL AllocAry(Init%InData_SrvD%MsrPositionsZ, size(IfW%y%lidar%MsrPositionsZ), 'Init%InData_SrvD%MsrPositionsZ',
653               CALL SetErrStat(ErrStat2,ErrMsg2,ErrStat,ErrMsg,RoutineName)
654               Init%InData_SrvD%MsrPositionsZ = IfW%y%lidar%MsrPositionsZ
655            endif
656            Init%InData_SrvD%SensorType    = IfW%p%lidar%SensorType
657            Init%InData_SrvD%NumBeam       = IfW%p%lidar%NumBeam
658            Init%InData_SrvD%NumPulseGate  = IfW%p%lidar%NumPulseGate
659            Init%InData_SrvD%PulseSpacing  = IfW%p%lidar%PulseSpacing
660         END IF
```

Figure B.20: Code added within the `FAST_Subs.f90` script to initiate the transfer of LIDAR outputs as ServoDyn inputs.

```
903      IF ( p_FAST%CompInflow == Module_IfW )  THEN
904
905         u_SrvD%WindDir  = ATAN2( y_IfW%VelocityUVW(2,1), y_IfW%VelocityUVW(1,1) )
906         u_SrvD%HorWindV = SQRT( y_IfW%VelocityUVW(1,1)**2 + y_IfW%VelocityUVW(2,1)**2 )
907         if (allocated(y_IfW%lidar%LidSpeed     ))  u_SrvD%LidSpeed      = y_IfW%lidar%LidSpeed
908         if (allocated(y_IfW%lidar%MsrPositionsX))  u_SrvD%MsrPositionsX = y_IfW%lidar%MsrPositionsX
909         if (allocated(y_IfW%lidar%MsrPositionsY))  u_SrvD%MsrPositionsY = y_IfW%lidar%MsrPositionsY
910         if (allocated(y_IfW%lidar%MsrPositionsZ))  u_SrvD%MsrPositionsZ = y_IfW%lidar%MsrPositionsZ
```

Figure B.21: Code added within the `FAST_Solver.f90` script to set LIDAR outputs as ServoDyn inputs.

```
375     subroutine InitLidarMeas()
376        integer  :: I,J
377        if (p%NumBeam == 0) return ! Nothing to set
378        ! Allocate arrays for inputs
379        if (allocated(InitInp%LidSpeed)) then    ! make sure we have the array allocated before setting it
380           CALL AllocAry(u%LidSpeed, size(InitInp%LidSpeed), 'u%LidSpeed', errStat2, ErrMsg2)
381           if (Failed())  return
382           u%LidSpeed = InitInp%LidSpeed
383        endif
384        if (allocated(InitInp%MsrPositionsX)) then    ! make sure we have the array allocated before setting it
385           CALL AllocAry(u%MsrPositionsX, size(InitInp%MsrPositionsX), 'u%MsrPositionsX', errStat2, ErrMsg2)
386           if (Failed())  return
387           u%MsrPositionsX = InitInp%MsrPositionsX
388        endif
389        if (allocated(InitInp%MsrPositionsY)) then    ! make sure we have the array allocated before setting it
390           CALL AllocAry(u%MsrPositionsY, size(InitInp%MsrPositionsY), 'u%MsrPositionsY', errStat2, ErrMsg2)
391           if (Failed())  return
392           u%MsrPositionsY = InitInp%MsrPositionsY
393        endif
394        if (allocated(InitInp%MsrPositionsZ)) then    ! make sure we have the array allocated before setting it
395           CALL AllocAry(u%MsrPositionsZ, size(InitInp%MsrPositionsZ), 'u%MsrPositionsZ', errStat2, ErrMsg2)
396           if (Failed())  return
397           u%MsrPositionsZ = InitInp%MsrPositionsZ
398        endif
399        ! Write summary info to summary file
400        if (p%SensorType > 0) then    ! Set these here rather than overwrite every loop step in SensorType 1 or 3
401           J=LidarMsr_StartIdx
402           call WrSumInfoRcvd( J+0, '','Lidar input: Sensor Type')
403           call WrSumInfoRcvd( J+1, '','Lidar input: Number of Beams')
404           call WrSumInfoRcvd( J+2, '','Lidar input: Number of Pulse Gates')
405           call WrSumInfoRcvd( J+3, '','Lidar input: Reference average wind speed for the lidar')
406        endif
407        if (p%SensorType == 1) THEN
408           do I=1,min(p%NumBeam,(LidarMsr_MaxChan-4)/4)    ! Don't overstep the end for the lidar measure group
409              J=LidarMsr_StartIdx + 4 + (I-1)
410              call WrSumInfoRcvd( J+0,                '','Lidar input: Measured Wind Speeds ('//trim(Num2LStr(I))//')')
411              call WrSumInfoRcvd( J+p%NumBeam*1,      '','Lidar input: Measurement Points X ('//trim(Num2LStr(I))//')')
412              call WrSumInfoRcvd( J+p%NumBeam*2,      '','Lidar input: Measurement Points Y ('//trim(Num2LStr(I))//')')
413              call WrSumInfoRcvd( J+p%NumBeam*3,      '','Lidar input: Measurement Points Z ('//trim(Num2LStr(I))//')')
414           enddo
415        elseif (p%SensorType == 2) THEN
416           J=LidarMsr_StartIdx
417           call WrSumInfoRcvd( J+4,                '','Lidar input: Measured Wind Speeds')
418           call WrSumInfoRcvd( J+5,                '','Lidar input: Measurement Points X')
419           call WrSumInfoRcvd( J+6,                '','Lidar input: Measurement Points Y')
420           call WrSumInfoRcvd( J+7,                '','Lidar input: Measurement Points Z')
421        elseif (p%SensorType == 3) THEN
422           do I=1,min(p%NumPulseGate,(LidarMsr_MaxChan-4)/4)    ! Don't overstep the end for the lidar measure group
423              J=LidarMsr_StartIdx + 4 + (I-1)
424              call WrSumInfoRcvd( J+0,                '','Lidar input: Measured Wind Speeds ('//trim(Num2LStr(I))//')')
425              call WrSumInfoRcvd( J+p%NumPulseGate*1, '','Lidar input: Measurement Points X ('//trim(Num2LStr(I))//')')
426              call WrSumInfoRcvd( J+p%NumPulseGate*2, '','Lidar input: Measurement Points Y ('//trim(Num2LStr(I))//')')
427              call WrSumInfoRcvd( J+p%NumPulseGate*3, '','Lidar input: Measurement Points Z ('//trim(Num2LStr(I))//')')
428           enddo
429        endif
430     end subroutine InitLidarMeas

542     !> Set the Lidar related sensor inputs
543     !!    avrSWAP(2001:2500)
544     subroutine SetEXavrSWAP_LidarSensors()
545          ! in case something got set wrong, don't try to write beyond array
546        if (size(dll_data%avrswap) < (LidarMsr_StartIdx + LidarMsr_MaxChan - 1) ) return
547        if (p%NumBeam == 0) return ! Nothing to set
548
549        if (p%SensorType > 0) then    ! Set these here rather than overwrite every loop step in SensorType 1 or 3
550           dll_data%avrswap(LidarMsr_StartIdx)       = real(p%SensorType,SiKi)    ! Sensor Type
551           dll_data%avrswap(LidarMsr_StartIdx+1)     = real(p%NumBeam,SiKi)       ! Number of Beams
552           dll_data%avrswap(LidarMsr_StartIdx+2)     = real(p%NumPulseGate,SiKi)  ! Number of Pulse Gates
553           dll_data%avrswap(LidarMsr_StartIdx+3)     = p%URefLid                  ! Reference average wind speed for the lidar
554        endif
555        if (p%SensorType == 1) THEN
556           do I=1,min(p%NumBeam,(LidarMsr_MaxChan-4)/4)    ! Don't overstep the end for the lidar measure group
557              J=LidarMsr_StartIdx + 4 + (I-1)
558              dll_data%avrswap(J)                    = u%LidSpeed(I)             ! Lidar Measured Wind Speeds
559              dll_data%avrswap(J+p%NumBeam)          = u%MsrPositionsX(I)        ! Lidar Measurement Points X
560              dll_data%avrswap(J+(p%NumBeam*2))      = u%MsrPositionsY(I)        ! Lidar Measurement Points Y
561              dll_data%avrswap(J+(p%NumBeam*3))      = u%MsrPositionsZ(I)        ! Lidar Measurement Points Z
562           enddo
563        elseif (p%SensorType == 2) THEN
564           dll_data%avrswap(LidarMsr_StartIdx+4)     = u%LidSpeed(1)             ! Lidar Measured Wind Speeds
565           dll_data%avrswap(LidarMsr_StartIdx+5)     = u%MsrPositionsX(1)        ! Lidar Measurement Points X
566           dll_data%avrswap(LidarMsr_StartIdx+6)     = u%MsrPositionsY(1)        ! Lidar Measurement Points Y
567           dll_data%avrswap(LidarMsr_StartIdx+7)     = u%MsrPositionsZ(1)        ! Lidar Measurement Points Z
568        elseif (p%SensorType == 3) THEN
569           do I=1,min(p%NumPulseGate,(LidarMsr_MaxChan-4)/4)    ! Don't overstep the end for the lidar measure group
570              J=LidarMsr_StartIdx + 4 + (I-1)
571              dll_data%avrswap(J)                    = u%LidSpeed(I)             ! Lidar Measured Wind Speeds
572              dll_data%avrswap(J+p%NumPulseGate)     = u%MsrPositionsX(I)        ! Lidar Measurement Points X
573              dll_data%avrswap(J+(p%NumPulseGate*2)) = u%MsrPositionsY(I)        ! Lidar Measurement Points Y
574              dll_data%avrswap(J+(p%NumPulseGate*3)) = u%MsrPositionsZ(I)        ! Lidar Measurement Points Z
575           enddo
576        endif
577     end subroutine SetEXavrSWAP_LidarSensors
```

Figure B.22: Code added within the `BladedInterface_EX.f90` script to enable the allocation of LIDAR outputs to the avrSWAP array.

B.2 ROSCO

```
57          ! Load Lidar Inputs
58
59           IF (LocalVar%iStatus == 0) THEN
60               LocalVar%LidSpeed(1) = 0
61               LocalVar%LidSpeed(2) = 0
62               LocalVar%LidSpeed(3) = 0
63               LocalVar%LidSpeed(4) = 0
64               LocalVar%LidSpeed(5) = 0
65           ELSE
66               LocalVar%LidSpeed(1) = avrSWAP(2001)
67               LocalVar%LidSpeed(2) = avrSWAP(2002)
68               LocalVar%LidSpeed(3) = avrSWAP(2003)
69               LocalVar%LidSpeed(4) = avrSWAP(2004)
70               LocalVar%LidSpeed(5) = avrSWAP(2005)
71
72          END IF
73               LocalVar%MsrPositionsX(1) = avrSWAP(2006)
74               LocalVar%MsrPositionsX(2) = avrSWAP(2007)
75               LocalVar%MsrPositionsX(3) = avrSWAP(2008)
76               LocalVar%MsrPositionsX(4) = avrSWAP(2009)
77               LocalVar%MsrPositionsX(5) = avrSWAP(2010)
78
79
80               LocalVar%MsrPositionsY(1) = avrSWAP(2011)
81               LocalVar%MsrPositionsY(2) = avrSWAP(2012)
82               LocalVar%MsrPositionsY(3) = avrSWAP(2013)
83               LocalVar%MsrPositionsY(4) = avrSWAP(2014)
84               LocalVar%MsrPositionsY(5) = avrSWAP(2015)
85
86
87               LocalVar%MsrPositionsZ(1) = avrSWAP(2016)
88               LocalVar%MsrPositionsZ(2) = avrSWAP(2017)
89               LocalVar%MsrPositionsZ(3) = avrSWAP(2018)
90               LocalVar%MsrPositionsZ(4) = avrSWAP(2019)
91               LocalVar%MsrPositionsZ(5) = avrSWAP(2020)
92
93
94               LocalVar%SensorType    = avrSWAP(2021)
95               LocalVar%NumBeam       = avrSWAP(2022)
96               LocalVar%NumPulseGate  = avrSWAP(2023)
97               LocalVar%URefLid       = avrSWAP(2024)
98
```

Figure B.23: Code added within the `ReadSetParameters.f90` script to enable the reading of LIDAR data from the avrSWAP array.

```
453         !------------ Lidar Assisted Control ------------
454     CALL ReadEmptyLine(UnControllerParameters,CurLine)
455     CALL ParseInput(UnControllerParameters,CurLine,'Hub_Height',accINFILE(1),CntrPar%Hub_Height,ErrVar)
456     CALL ParseInput(UnControllerParameters,CurLine,'CohWaveNum',accINFILE(1),CntrPar%CohWaveNum,ErrVar)
457     CALL ParseInput(UnControllerParameters,CurLine,'f_cutoff_IPC',accINFILE(1),CntrPar%f_cutoff_IPC,ErrVar)
458     CALL ParseInput(UnControllerParameters,CurLine,'f_delay',accINFILE(1),CntrPar%f_delay,ErrVar)
459     CALL ParseInput(UnControllerParameters,CurLine,'Kp_IPC',accINFILE(1),CntrPar%Kp_IPC,ErrVar)
460     CALL ParseInput(UnControllerParameters,CurLine,'T_scan',accINFILE(1),CntrPar%T_scan,ErrVar)
461     CALL ParseInput(UnControllerParameters,CurLine,'NumPitchCurveFF',accINFILE(1),CntrPar%NumPitchCurveFF,ErrVar)
462     CALL ParseAry(UnControllerParameters,CurLine,'REWS_curve',CntrPar%REWS_curve,CntrPar%NumPitchCurveFF,accINFILE(1),ErrVar)
463     CALL ParseAry(UnControllerParameters,CurLine,'Pitch_curve',CntrPar%Pitch_curve,CntrPar%NumPitchCurveFF,accINFILE(1),ErrVar)
464     CALL ReadEmptyLine(UnControllerParameters,CurLine)
```

Figure B.24: Code added within the `ReadSetParameters.f90` script to enable the reading of LIDAR-assisted control inputs from the ROSCO input file.

```
    INTEGER(4)                          :: LIDAR_ControlMode     ! Flag of lidar assisted Pitch forward control
    REAL(8)                             :: Hub_Height            ! Turbine hub height, [m]
    REAL(8)                             :: CohWaveNum            ! Coherent wave number, [rad/m]
    REAL(8)                             :: f_cutoff_IPC          ! Corner frequency (-3dB point) in the low-pass filters, [rad/s]
    REAL(8)                             :: Kp_IPC                ! Proportional gain of feedforward IPC commands
    REAL(8)                             :: f_delay               ! Frequency where the turbine experiences oscillations
    REAL(8)                             :: T_scan                ! Scan duration of the LIDAR
    INTEGER(4)                          :: NumPitchCurveFF       ! Number of points in the steady state pitch curve
    REAL(8), DIMENSION(:), ALLOCATABLE  :: REWS_curve            ! REWS Curve
    REAL(8), DIMENSION(:), ALLOCATABLE  :: Pitch_curve           ! Pitch Curve
END TYPE ControlParameters
```

Figure B.25: Variables added to the `ControlParameters` type within the `ROSCO_Types.f90` script to enable the specification of the LAC parameters added to the ROSCO input file.

```
273           ! ---------- LIDAR variables ----------
274     INTEGER(4)                          :: MaxBufferStep = 20          ! the maximum buffer steps for store the u_est
275     INTEGER(4)                          :: MaxBufferStep_REWS = 2000   ! the maximum buffer steps for store the REWS, 25 second for a 80Hz
276     REAL(8)                              :: NumPulseGate               ! number of gates in each lidar beam (pulsed lidar) received from the Discon.in  just need to cross
277     INTEGER(4)                           :: NewLidarMeasurement        ! Flag whether it is a successful new measurement
278     INTEGER(4)                           :: URefLid                    ! Reference average wind speed for the lidar [m/s]
279     REAL(8)                              :: LidSpeed(5)                ! Line-of-sight wind speeds measurement [m/s]
280     REAL(8)                              :: LidSpeed_F(5)              ! Line-of-sight wind speeds measurement [m/s]
281     REAL(8)                              :: MsrPositionsX(5)           ! X direction lidar measurement coordinates [m]
282     REAL(8)                              :: MsrPositionsY(5)           ! Y direction lidar measurement coordinates [m]
283     REAL(8)                              :: MsrPositionsZ(5)           ! Z direction lidar measurement coordinates [m]
284     REAL(8), DIMENSION(:,:), ALLOCATABLE  :: MsrPosition               ! XYZ Coordinates of measurement position [m]
285     REAL(8)                              :: SensorType                 ! Lidar sensor type
286     REAL(8)                              :: NumBeam                    ! Number of beams
287     REAL(8), DIMENSION(:), ALLOCATABLE   :: BeamAzimuth                ! Rotor azimuth of the Lidar beams [rad]
288     REAL(8)                              :: PulseSpacing               ! Distance between range gates
289     REAL(8), DIMENSION(:), ALLOCATABLE   :: REWS_f                     ! Low pass filtered rotor effective wind speed,  estimated by lidar for feed-forward control, with a
290     REAL(8)                              :: REWS_b                     ! filtered and buffered Rotor effective wind speed [m/s]
291     REAL(8), DIMENSION(:), ALLOCATABLE   :: REWS_f_Time                ! Rotor effective wind speed,  buffer time[s]
292     REAL(8)                              :: LeadTime_1                 ! The leading time of the first measurement plane [s]
293     REAL(8)                              :: FF_Pitch                   ! The pitch of feed forwad control [rad]
294     REAL(8)                              :: FF_Pitch_old               ! The previous pitch of feed forwad control [rad]
295     REAL(8)                              :: FF_PitchRate               ! The pitch rate of feed forward control [rad/s]
296     REAL(8)                              :: w_delay                    ! Angular delay frequency [Hz]
297     REAL(8)                              :: f_cutoff                   ! Cut-off frequency [Hz]
298     REAL(8)                              :: w_cutoff                   ! Angular cut-off frequency [Hz]
299     REAL(8)                              :: w_cutoff_PA                ! Angular cut-off frequency for pitch actuator [Hz]
300     REAL(8)                              :: w_cutoff_IPC               ! Angular cut-off frequency for IPC [Hz]
301     REAL(8)                              :: w_n                        ! w_delay/w_cutoff
302     REAL(8)                              :: T_filter                   ! Time delay incurred by LPF on REWS [s]
303     REAL(8)                              :: w_n_PA                     ! w_delay/w_cutoff for pitch actuator
304     REAL(8)                              :: T_PA                       ! Time delay incurred by LPF on pitch actuator [s]
305     REAL(8)                              :: T_buffer                   ! Buffer time required for FF commands [s]
306     REAL(8)                              :: T_filter_IPC               ! Time delay incurred by LPF on FFIPC [s]
307     REAL(8)                              :: w_n_IPC                    ! w_delay/w_cutoff for FFIPC
308     ! ---------- LIDAR IPC Variables
309     REAL(8), DIMENSION(:,:), ALLOCATABLE  :: REWS_f_IPC                ! Low pass filtered rotor effective wind speed,  estimated by lidar for feed-forward control, with a
310     REAL(8), DIMENSION(:), ALLOCATABLE    :: REWS_Filtered_IPC
311     REAL(8), DIMENSION(:), ALLOCATABLE    :: REWS_b_IPC                ! filtered and buffered Rotor effective wind speed [m/s]
312     REAL(8), DIMENSION(:,:), ALLOCATABLE  :: REWS_f_Time_IPC           ! Rotor effective wind speed,  buffer time, [s]
313     REAL(8), DIMENSION(:), ALLOCATABLE    :: LeadTime_1_IPC            ! The leading time of the first measurement plane, [s]
314     REAL(8), DIMENSION(:), ALLOCATABLE    :: FF_Pitch_IPC              ! The feedforward IPC command, [rad]
315     REAL(8), DIMENSION(:), ALLOCATABLE    :: FF_Pitch_old_IPC          ! The previous feedforward IPC command, [rad]
316     REAL(8), DIMENSION(:), ALLOCATABLE    :: IPC_PitComFF              ! Commanded pitch of each blade as calculated by the feedforward individual pitch controller [rad].
317     REAL(8), DIMENSION(:), ALLOCATABLE    :: IPC_Com_FF                ! Commanded pitch of each blade as calculated by the feedforward individual pitch controller [rad].
318
319 END TYPE LocalVariables
```

Figure B.26: Variables added to the `LocalVariables` type within the `ROSCO_Types.f90` script for use within the feedforward control subroutines.

```
! Controller parameter input file for the IEA-15-240-RWT-UMaineSemi wind turbine
!    - File written using ROSCO version 2.6.0 controller tuning logic on 09/23/22
|
!------- DEBUG ------------------------------------------------------------
0                   ! LoggingLevel              - {0: write no debug files, 1: write standard output .dbg-file, 2: LoggingLeve

!------- CONTROLLER FLAGS -------------------------------------------------
2                   ! F_LPFType                 - {1: first-order low-pass filter, 2: second-order low-pass filter}, [rad/s] (
2                   ! F_NotchType               - Notch on the measured generator speed and/or tower fore-aft motion (for flo
0                   ! IPC_ControlMode    - Turn Individual Pitch Control (IPC) for fatigue load reductions (pitch contribution)
2                   ! VS_ControlMode     - Generator torque control mode in above rated conditions {0: constant torque, 1: cons
1                   ! PC_ControlMode     - Blade pitch control mode {0: No pitch, fix to fine pitch, 1: active PI blade pitch c
0                   ! Y_ControlMode              - Yaw control mode {0: no yaw control, 1: yaw rate control, 2: yaw-by-IPC}
2                   ! LIDAR_ControlMode - Lidar-assisted control mode {0: No LAC, 1: FFCPC, 2: FFCPC+FFIPC}
1                   ! SS_Mode           - Setpoint Smoother mode {0: no setpoint smoothing, 1: introduce setpoint smoothing}
2                   ! WE_Mode           - Wind speed estimator mode {0: One-second low pass filtered hub height wind speed, 1:
1                   ! PS_Mode           - Pitch saturation mode {0: no pitch saturation, 1: implement pitch saturation}
0                   ! SD_Mode           - Shutdown mode {0: no shutdown procedure, 1: pitch to max pitch at shutdown}
2                   ! Fl_Mode           - Floating specific feedback mode {0: no nacelle velocity feedback, 1: feed back trans
0                   ! TD_Mode           - Tower damper mode {0: no tower damper, 1: feed back translational nacelle accelerat
0                   ! Flp_Mode          - Flap control mode {0: no flap control, 1: steady state flap angle, 2: Proportional f
0                   ! OL_Mode           - Open loop control mode {0: no open loop control, 1: open loop control vs. time}
2                   ! PA_Mode           - Pitch actuator mode {0 - not used, 1 - first order filter, 2 - second order filter}
0                   ! Ext_Mode          - External control mode {0 - not used, 1 - call external dynamic library}
0                   ! ZMQ_Mode          - Fuse ZeroMQ interface {0: unused, 1: Yaw Control}

!------- FILTERS ----------------------------------------------------------
1.00810             ! F_LPFCornerFreq    - Corner frequency (-3dB point) in the low-pass filters, [rad/s]
0.70000             ! F_LPFDamping              - Damping coefficient {used only when F_FilterType = 2} [-]
3.35500             ! F_NotchCornerFreq - Natural frequency of the notch filter, [rad/s]
0.000000  0.250000  ! F_NotchBetaNumDen - Two notch damping values (numerator and denominator, resp) - determines the width ar
0.62830             ! F_SSCornerFreq    - Corner frequency (-3dB point) in the first order low pass filter for the setpoint sr
0.20944             ! F_WECornerFreq    - Corner frequency (-3dB point) in the first order low pass filter for the wind speed
0.17952             ! F_YawErr          - Low pass filter corner frequency for yaw controller [rad/s].
0.213000  1.000000  ! F_FlCornerFreq    - Natural frequency and damping in the second order low pass filter of the tower-top f
0.01042             ! F_FlHighPassFreq    - Natural frequency of first-order high-pass filter for nacelle fore-aft motion [rad
10.461600  1.000000  ! F_FlpCornerFreq   - Corner frequency and damping in the second order low pass filter of the blade root

!------- BLADE PITCH CONTROL ----------------------------------------------
30                  ! PC_GS_n                   - Amount of gain-scheduling table entries
0.062094  0.088628  0.109270  0.127148  0.143223  0.158025  0.171764  0.184722  0.197105  0.209011  0.220338  0.231364  0.242(
-1.143968  -0.988219  -0.859873  -0.752282  -0.660788  -0.582031  -0.513524  -0.453388  -0.400177  -0.352760  -0.310240  -0.27
-0.119726  -0.108178  -0.098661  -0.090684  -0.083900  -0.078060  -0.072981  -0.068522  -0.064576  -0.061061  -0.057908  -0.05
0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000(
0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000(
1.570000000000      ! PC_MaxPit                 - Maximum physical pitch limit, [rad].
0.000000000000      ! PC_MinPit                 - Minimum physical pitch limit, [rad].
0.034900000000      ! PC_MaxRat                 - Maximum pitch rate (in absolute value) in pitch controller, [rad/s].
-0.03490000000      ! PC_MinRat                 - Minimum pitch rate (in absolute value) in pitch controller, [rad/s].
0.791680000000      ! PC_RefSpd                 - Desired (reference) HSS speed for pitch controller, [rad/s].
0.000000000000      ! PC_FinePit                - Record 5: Below-rated pitch angle set-point, [rad]
0.017450000000      ! PC_Switch                 - Angle above lowest minimum pitch angle for switch, [rad]

!------- INDIVIDUAL PITCH CONTROL -----------------------------------------
8.592000  10.740000  ! IPC_Vramp               - Start and end wind speeds for cut-in ramp function. First entry: IPC inactiv
0.007                ! IPC_IntSat              - Integrator saturation (maximum signal amplitude contribution to pitch from I
0.001e+00 0.000e+00 ! IPC_KP                  - Proportional gain for the individual pitch controller: first parameter for 1
0.001e+00 0.000e+00  ! IPC_KI                 - Integral gain for the individual pitch controller: first parameter for 1P re
0.000000  0.000000  ! IPC_aziOffset           - Phase offset added to the azimuth angle for the individual pitch controller,
0.0                 ! IPC_CornerFreqAct - Corner frequency of the first-order actuators model, to induce a phase lag in the IP

!------- VS TORQUE CONTROL ------------------------------------------------
95.75600000000      ! VS_GenEff                 - Generator efficiency mechanical power -> electrical power, [should match the
19786767.46773      ! VS_ArSatTq                - Above rated generator torque PI control saturation, [Nm]
4500000.000000      ! VS_MaxRat                 - Maximum torque rate (in absolute value) in torque controller, [Nm/s].
21765444.21450      ! VS_MaxTq                  - Maximum generator torque in Region 3 (HSS side), [Nm].
0.000000000000      ! VS_MinTq                  - Minimum generator torque (HSS side), [Nm].
0.523600000000      ! VS_MinOMSpd               - Minimum generator speed [rad/s]
39394899.67345      ! VS_Rgn2K                  - Generator torque constant in Region 2 (HSS side), [Nm/(rad/s)^2]
15000000.00000      ! VS_RtPwr                  - Wind turbine rated power [W]
19786767.46773      ! VS_RtTq                   - Rated torque, [Nm].
0.791680000000      ! VS_RefSpd                 - Rated generator speed [rad/s]
1                   ! VS_n                              - Number of generator PI torque controller gains
-35730593.18196      ! VS_KP                            - Proportional gain for generator PI torque controller [-]. (Only usec
-4499370.31680      ! VS_KI                             - Integral gain for generator PI torque controller [s]. (Only used in
9.00                ! VS_TSRopt                 - Power-maximizing region 2 tip-speed-ratio [rad].

!------- SETPOINT SMOOTHER ------------------------------------------------
1.00000             ! SS_VSGain         - Variable speed torque controller setpoint smoother gain, [-].
0.00100             ! SS_PCGain         - Collective pitch controller setpoint smoother gain, [-].
```

Figure B.27: Updated ROSCO input file (part 1) with the new LIDAR-assisted control flag, indicated by the green box.

```
!------- WIND SPEED ESTIMATOR ----------------------------------------------
120.970             ! WE_BladeRadius     - Blade length (distance from hub center to blade tip), [m]
1                   ! WE_CP_n                    - Amount of parameters in the Cp array
0.0                 ! WE_CP - Parameters that define the parameterized CP(lambda) function
0.0                    ! WE_Gamma                      - Adaption gain of the wind speed estimator algorithm [m/rad]
1.0                 ! WE_GearboxRatio    - Gearbox ratio [>=1],  [-]
312456272.00000       ! WE_Jtot                 - Total drivetrain inertia, including blades, hub and casted generator ine
1.225               ! WE_RhoAir                  - Air density, [kg m^-3]
"../IEA-15-240-RWT/Cp_Ct_Cq.IEA15MW.txt"      ! PerfFileName      - File containing rotor performance tables (Cp,Ct,Cq) (a
36      26          ! PerfTableSize      - Size of rotor performance tables, first number refers to number of blade pitch a
60                  ! WE_FOPoles_N       - Number of first-order system poles used in EKF
3.0000 3.2669 3.5338 3.8007 4.0676 4.3345 4.6014 4.8683 5.1352 5.4021 5.6690 5.9359 6.2028 6.4697 6.7366 7.0034 7.2703 7.5
-0.02567895 -0.02796349 -0.03024803 -0.03253257 -0.03481711 -0.03710166 -0.03938620 -0.04167074 -0.04395528 -0.04623982 -0

!------- YAW CONTROL ------------------------------------------------------
0.00000             ! Y_uSwitch          - Wind speed to switch between Y_ErrThresh. If zero, only the first value of Y_Err
4.000000  8.000000  ! Y_ErrThresh     - Yaw error threshold/deadband. Turbine begins to yaw when it passes this. If Y_uSwit
0.00870             ! Y_Rate                     - Yaw rate [rad/s]
0.00000             ! Y_MErrSet          - Integrator saturation (maximum signal amplitude contribution to pitch from yaw-b
0.00000             ! Y_IPC_IntSat               - Integrator saturation (maximum signal amplitude contribution to pitch fr
0.00000             ! Y_IPC_KP                   - Yaw-by-IPC proportional controller gain Kp
0.00000             ! Y_IPC_KI                   - Yaw-by-IPC integral controller gain Ki

!------- LIDAR-ASSISTED CONTROL -------------------------------------------
150                 ! Hub_Height - Turbine hub height, [m]
0.01563             ! CohWaveNum  - Coherent wave number [rad/m]
0.01321             ! f_cutoff_IPC  - Corner frequency (-3dB point) in the low-pass filters, [rad/s]
0.025               ! f_delay  -  Critical frequency near where the rotor has higher fluctuations [rad/s]
1                   ! T_scan - Duration to complete one full scane
19                  ! NumPitchCurveFF - Number of points in the wind-pitch-Kp curve
2 9.3 10 11     12 13 14 15     16      17      18      19      20      21      22      23      24      25      30     ! F
0.0000  0.0000  0.0000  0.0614  0.1130  0.1491  0.1790  0.2055  0.2296  0.2521  0.2732  0.2932  0.3124  0.3308  0.3485  0.

!------- TOWER FORE-AFT DAMPING -------------------------------------------
-1.00000            ! FA_KI                          - Integral gain for the fore-aft tower damper controller [rad s/m]
0.0                 ! FA_HPFCornerFreq  - Corner frequency (-3dB point) in the high-pass filter on the fore-aft acceleratı
0.0                 ! FA_IntSat                  - Integrator saturation (maximum signal amplitude contribution to pitch fr

!------- MINIMUM PITCH SATURATION -----------------------------------------
60                  ! PS_BldPitchMin_N  - Number of values in minimum blade pitch lookup table (should equal number of val
3.0000 3.2669 3.5338 3.8007 4.0676 4.3345 4.6014 4.8683 5.1352 5.4021 5.6690 5.9359 6.2028 6.4697 6.7366 7.0034 7.2703 7.5
0.05997374 0.05997374 0.05997374 0.05997374 0.05997374 0.05997374 0.05624970 0.05205911 0.04715388 0.04147557 0.03553359 0

!------- SHUTDOWN ---------------------------------------------------------
0.698100000000      ! SD_MaxPit          - Maximum blade pitch angle to initiate shutdown, [rad]
0.418880000000      ! SD_CornerFreq      - Cutoff Frequency for first order low-pass filter for blade pitch angle, [rad/s]

!------- Floating --------------------------------------------------------
-9.25611000000      ! Fl_Kp              - Nacelle pitching proportional feedback gain [s]

!------- FLAP ACTUATION ---------------------------------------------------
0.000000000000      ! Flp_Angle          - Initial or steady state flap angle [rad]
0.00000000e+00      ! Flp_Kp             - Blade root bending moment proportional gain for flap control [s]
0.00000000e+00      ! Flp_Ki             - Flap displacement integral gain for flap control [-]
0.174500000000      ! Flp_MaxPit         - Maximum (and minimum) flap pitch angle [rad]
```

Figure B.28: Updated ROSCO input file (part 2) with a new section added for specification of LIDAR-assisted control parameters, indicated by the green box.

```
!----------------------------------------------------------------------------------------------------------------------
    SUBROUTINE LidarConfiguration(LocalVar, CntrPar, objInst)
    ! Sets the lidar configuration and measurement coordinates

        USE ROSCO_Types, ONLY : LocalVariables, ControlParameters, ObjectInstances
        IMPLICIT NONE

        ! Inputs
        TYPE(ControlParameters),    INTENT(IN    )      :: CntrPar
        TYPE(LocalVariables),       INTENT(INOUT)       :: LocalVar
        TYPE(ObjectInstances),      INTENT(INOUT)       :: objInst

        REAL(IntKi)                                     :: I        !Loop counter
        REAL(ReKi)                                      :: HubHeightCorrectedZ(5)

   IF (LocalVar%SensorType == 1) THEN

        DO I = 1,LocalVar%NumBeam
         IF (.not. allocated(LocalVar%MsrPosition)) THEN
             Allocate(LocalVar%MsrPosition(3,5))
           LocalVar%MsrPosition(1,I) = LocalVar%MsrPositionsX(I)
           LocalVar%MsrPosition(2,I) = LocalVar%MsrPositionsY(I)
           LocalVar%MsrPosition(3,I) = LocalVar%MsrPositionsZ(I)
         END IF
           LocalVar%MsrPosition(1,I) = LocalVar%MsrPositionsX(I)
           LocalVar%MsrPosition(2,I) = LocalVar%MsrPositionsY(I)
           LocalVar%MsrPosition(3,I) = LocalVar%MsrPositionsZ(I)


         !Calculate beam azimuth angles

         IF (.not. allocated(LocalVar%BeamAzimuth)) THEN
             Allocate(LocalVar%BeamAzimuth(5))
             LocalVar%BeamAzimuth(:) = 0
         END IF
           HubHeightCorrectedZ(I) =   LocalVar%MsrPosition(3,I) -  CntrPar%Hub_Height  ! NEED TO CREATE AN AVRSWAP VARIABLE FOR HUB HEIGHT

          IF (HubHeightCorrectedZ(I) > 0  .AND. LocalVar%MsrPosition(2,I) > 0) THEN
               LocalVar%BeamAzimuth(I)  = Pi/2 -  ATAN(HubHeightCorrectedZ(I)/LocalVar%MsrPosition(2,I))

           ELSEIF (HubHeightCorrectedZ(I) == 0  .AND. LocalVar%MsrPosition(2,I) > 0) THEN
               LocalVar%BeamAzimuth(I)  = Pi/2

           ELSEIF (HubHeightCorrectedZ(I) == 0  .AND. LocalVar%MsrPosition(2,I) < 0) THEN
               LocalVar%BeamAzimuth(I)  = 3*Pi/2

           ELSEIF (HubHeightCorrectedZ(I) < 0  .AND. LocalVar%MsrPosition(2,I) > 0) THEN
               LocalVar%BeamAzimuth(I)  = ATAN(ABS(HubHeightCorrectedZ(I))/LocalVar%MsrPosition(2,I)) + Pi/2

           ELSEIF (HubHeightCorrectedZ(I) < 0  .AND. LocalVar%MsrPosition(2,I) < 0) THEN
               LocalVar%BeamAzimuth(I)  = (Pi/2 - ATAN(HubHeightCorrectedZ(I)/LocalVar%MsrPosition(2,I))) + Pi

           ELSEIF (HubHeightCorrectedZ(I) > 0  .AND. LocalVar%MsrPosition(2,I) < 0) THEN
               LocalVar%BeamAzimuth(I)  = ATAN(HubHeightCorrectedZ(I)/ABS(LocalVar%MsrPosition(2,I))) + 3*Pi/2

           ELSEIF (HubHeightCorrectedZ(I) > 0  .AND. LocalVar%MsrPosition(2,I) == 0) THEN
               LocalVar%BeamAzimuth(I)  = 0

           ELSEIF (HubHeightCorrectedZ(I) < 0  .AND. LocalVar%MsrPosition(2,I) == 0) THEN
               LocalVar%BeamAzimuth(I)  = Pi
           END IF
        END DO

        !REWS
        LocalVar%LidSpeed(1) = (LocalVar%LidSpeed(1) + LocalVar%LidSpeed(2) + LocalVar%LidSpeed(3) + LocalVar%LidSpeed(4) + LocalVar%LidSpeed(5))/LocalVar%NumBeam

        ELSE IF (LocalVar%SensorType == 2) THEN

         IF (.not. allocated(LocalVar%MsrPosition)) THEN
             Allocate(LocalVar%MsrPosition(3,1))
             LocalVar%MsrPosition(1,1) = LocalVar%MsrPositionsX(1)
             LocalVar%MsrPosition(2,1) = LocalVar%MsrPositionsY(1)
             LocalVar%MsrPosition(3,1) = LocalVar%MsrPositionsZ(1)
         END IF
             LocalVar%MsrPosition(1,1) = LocalVar%MsrPositionsX(1)
             LocalVar%MsrPosition(2,1) = LocalVar%MsrPositionsY(1)
             LocalVar%MsrPosition(3,1) = LocalVar%MsrPositionsZ(1)

        ELSE IF (LocalVar%SensorType == 3) THEN

        DO I = 1,LocalVar%NumPulseGate

           IF (.not. allocated(LocalVar%MsrPosition)) THEN
             Allocate(LocalVar%MsrPosition(3,5))

            LocalVar%MsrPosition(1,I) = LocalVar%MsrPositionsX(1) - ((I - 1)*LocalVar%PulseSpacing)
    LocalVar%MsrPosition(2,I) = LocalVar%MsrPositionsY(1)
    LocalVar%MsrPosition(3,I) = LocalVar%MsrPositionsZ(1)

    END IF

    LocalVar%MsrPosition(1,I) = LocalVar%MsrPositionsX(1) - ((I - 1)*LocalVar%PulseSpacing)
    LocalVar%MsrPosition(2,I) = LocalVar%MsrPositionsY(1)
    LocalVar%MsrPosition(3,I) = LocalVar%MsrPositionsZ(1)

 END DO

    LocalVar%LidSpeed(1) = (LocalVar%LidSpeed(1) + LocalVar%LidSpeed(2) + LocalVar%LidSpeed(3) + LocalVar%LidSpeed(4) + LocalVar%LidSpeed(5))/LocalVar%NumPulseGate

END IF


END SUBROUTINE LidarConfiguration
```

Figure B.29: LIDAR configuration subroutines created to determine the rotor azimuth angles of the LIDAR beams and to determine the LIDAR-measured rotor-effective wind speed.

```
SUBROUTINE FeedForwardCPC(LocalVar, CntrPar, DebugVar, objInst, ErrVar)

   USE ROSCO_Types, ONLY : ControlParameters, LocalVariables, DebugVariables, ObjectInstances, ErrorVariables

   ! REAL(C_FLOAT), INTENT(INOUT) :: avrSWAP(*)   ! The swap array, used to pass data to, and receive data from, the DLL controller.
    TYPE(LocalVariables), INTENT(INOUT)        :: LocalVar
    TYPE(ControlParameters), INTENT(INOUT)        :: CntrPar
    TYPE(DebugVariables),    INTENT(INOUT)     :: DebugVar
    TYPE(ObjectInstances), INTENT(INOUT)       :: objInst
    TYPE(ErrorVariables), INTENT(INOUT)       :: ErrVar
    INTEGER(4)                                  :: iBuffer
    REAL(8)                                     :: IND_Time(LocalVar%MaxBufferStep_REWS)
    INTEGER(4)                                  :: It


    !--------- Shift the filter and Time in the buffer and assign new values ------------------------

   LocalVar%REWS_f(2:LocalVar%MaxBufferStep_REWS) = LocalVar%REWS_f(1:LocalVar%MaxBufferStep_REWS-1)
   LocalVar%REWS_f_Time(2:LocalVar%MaxBufferStep_REWS) = LocalVar%REWS_f_Time(1:LocalVar%MaxBufferStep_REWS-1)
   LocalVar%REWS_f(1)    = LPFilter(LocalVar%LidSpeed(1), LocalVar%DT, LocalVar%w_cutoff, LocalVar%FP, LocalVar%iStatus, .FALSE., objInst%instLPF)
   LocalVar%LeadTime_1   =  (-LocalVar%MsrPosition(1,1))/LocalVar%URefLid
   LocalVar%REWS_f_time(1) = LocalVar%LeadTime_1 + LocalVar%Time
   LocalVar%w_delay = 2*Pi*CntrPar%f_delay
   LocalVar%w_n = LocalVar%w_delay/LocalVar%w_cutoff
   LocalVar%T_filter = ATAN(LocalVar%w_n)/LocalVar%w_delay
   LocalVar%w_cutoff_PA = CntrPar%PA_CornerFreq*2*Pi
   LocalVar%w_n_PA = LocalVar%w_delay/LocalVar%w_cutoff_PA
   LocalVar%T_PA = ATAN((2**(1/2)*LocalVar%w_n_PA)/(1-LocalVar%w_n_PA**2))/LocalVar%w_delay
   LocalVar%T_buffer = LocalVar%LeadTime_1 - LocalVar%T_filter - LocalVar%T_PA - CntrPar%T_scan/2

   !---------timing  Get the buffer ID that has the time closest to the pitch action time
   IND_Time   =  ABS(LocalVar%REWS_f_Time - LocalVar%Time - LocalVar%T_filter - LocalVar%T_PA - CntrPar%T_scan/2)    ! This determines the time tha
   It         = INT(MINLOC(IND_Time, 1))                                                                            ! This determines the position  in the RES array that needs t
   LocalVar%REWS_b  = LocalVar%REWS_f(It)                                                                           !REWS_b is the wind speed AT the desired time
   LocalVar%FF_Pitch = interp1d(CntrPar%REWS_curve,CntrPar%Pitch_curve, LocalVar%REWS_b, ErrVar)
   LocalVar%FF_PitchRate = (LocalVar%FF_Pitch-LocalVar%FF_Pitch_old)/LocalVar%DT
   LocalVar%FF_Pitch_old = LocalVar%FF_Pitch

   IF (LocalVar%REWS_b < 13) THEN !Only applied to the WindCrete
       LocalVar%FF_PitchRate = 0
   ENDIF

   DebugVar%REWS = LocalVar%LidSpeed(1)
   DebugVar%T_buffer = LocalVar%T_buffer
   DebugVar%f_cutoff = LocalVar%f_cutoff
   DebugVar%REWS_f = LocalVar%REWS_f(1)
   DebugVar%T_filter  = LocalVar%T_filter
   DebugVar%T_PA  = LocalVar%T_PA
   DebugVar%REWS_b = LocalVar%REWS_b
   DebugVar%FF_PitchRate = LocalVar%FF_PitchRate

END SUBROUTINE FeedForwardCPC
```

Figure B.30: FFCPC subroutine added within the `Controllers.f90` script.

```
 REAL(DbKi) FUNCTION PIControllerFF(error, kp, ki, PitchRate, minValue, maxValue, DT, I0, piP, reset, inst)
! PI controller, capable of accepting forward pitch rate with output saturation

   USE ROSCO_Types, ONLY : piParams

   IMPLICIT NONE
   ! Allocate Inputs
   REAL(8), INTENT(IN)         :: error
   REAL(8), INTENT(IN)         :: kp
   REAL(8), INTENT(IN)         :: ki
   REAL(8), INTENT(IN)         :: PitchRate
   REAL(8), INTENT(IN)         :: minValue
   REAL(8), INTENT(IN)         :: maxValue
   REAL(8), INTENT(IN)         :: DT
   INTEGER(4), INTENT(INOUT)   :: inst
   REAL(8), INTENT(IN)         :: I0
   TYPE(piParams), INTENT(INOUT)  :: piP
   LOGICAL, INTENT(IN)         :: reset
   ! Allocate local variables
   INTEGER(4)                      :: i                                        ! Counter for making arrays
   REAL(8)                         :: PTerm                                    ! Proportional term


   ! Initialize persistent variables/arrays, and set inital condition for integrator term
   IF (reset) THEN
       piP%ITerm(inst) = I0
       piP%ITermLast(inst) = I0

       PIControllerFF = I0
   ELSE


       PTerm = kp*error
       piP%ITerm(inst) = piP%ITermLast(inst) + DT*ki*error + DT*PitchRate
       piP%ITerm(inst) = saturate(piP%ITerm(inst), minValue, maxValue)
       PIControllerFF = saturate(PTerm + piP%ITerm(inst), minValue, maxValue)

       piP%ITermLast(inst) = piP%ITerm(inst)
   END IF
   inst = inst + 1

 END FUNCTION PIControllerFF
```

Figure B.31: FFCPC PI controller function added within the `Functions.f90` script.

```
SUBROUTINE FeedForwardIPC(LocalVar, CntrPar, DebugVar, objInst, ErrVar)

   USE ROSCO_Types, ONLY : ControlParameters, LocalVariables, DebugVariables, ObjectInstances, ErrorVariables

  ! REAL(C_FLOAT), INTENT(INOUT) :: avrSWAP(*)   ! The swap array, used to pass data to, and receive data from, the DLL controller.
   TYPE(LocalVariables), INTENT(INOUT)         :: LocalVar
   TYPE(ControlParameters), INTENT(INOUT)         :: CntrPar
   TYPE(ObjectInstances), INTENT(INOUT)        :: objInst
   TYPE(DebugVariables), INTENT(INOUT)        :: DebugVar
   TYPE(ErrorVariables), INTENT(INOUT)        :: ErrVar
   INTEGER(4)                                   :: iBuffer
   REAL(8), DIMENSION(:,:), ALLOCATABLE         :: IND_Time_IPC
   INTEGER(4)                                   :: It
   INTEGER(4)                                   :: IBeam
   INTEGER(4)                                   :: I
   INTEGER(4)                                   :: ISample
   INTEGER(4)                                   :: k
   INTEGER(4)                                   :: LoopStart

   REAL(8)                                      :: BlAzimuth
   REAL(8)                                      :: BlAzimuth_New
   REAL(8), DIMENSION(:), ALLOCATABLE           :: AzimuthDiff
   REAL(8), DIMENSION(:), ALLOCATABLE           :: BeamAzimuth
   INTEGER(4)                                   :: BeamID
   INTEGER(4)                                   :: RollAvgSamp = 160


   IF  (ABS(LocalVar%MsrPosition(2,1) < 20) .AND. (ABS(LocalVar%MsrPosition(3,1)-CntrPar%Hub_Height) < 20)) THEN
       LoopStart = 2
   ELSE
       LoopStart = 1
   END IF

   DO  IBeam = LoopStart, LocalVar%NumBeam
   !--------- Shift the filter and Time in the buffer and assign new values ------------------------
   LocalVar%REWS_f_IPC(2:LocalVar%MaxBufferStep_REWS, IBeam) = LocalVar%REWS_f_IPC(1:LocalVar%MaxBufferStep_REWS-1, IBeam)
   LocalVar%REWS_f_Time_IPC(2:LocalVar%MaxBufferStep_REWS, IBeam) = LocalVar%REWS_f_Time_IPC(1:LocalVar%MaxBufferStep_REWS-1, IBeam)
   LocalVar%REWS_f_IPC(1, IBeam)       = LPFilter(LocalVar%LidSpeed(IBeam), LocalVar%DT, LocalVar%w_cutoff, LocalVar%FP, LocalVar%iStatus, .FALSE., objInst%instLPF)
   LocalVar%LeadTime_1_IPC(IBeam)   =  (-LocalVar%MsrPosition(1,IBeam))/LocalVar%URefLid
   LocalVar%REWS_f_time_IPC(1,IBeam) = LocalVar%LeadTime_1_IPC(IBeam) + LocalVar%Time
   LocalVar%w_delay = 2*Pi*CntrPar%f_delay
   LocalVar%w_cutoff_IPC = 2*Pi*CntrPar%f_cutoff_IPC
   LocalVar%w_n_IPC = LocalVar%w_delay/LocalVar%w_cutoff_IPC
   LocalVar%T_filter_IPC = ATAN(LocalVar%w_n_IPC)/LocalVar%w_delay

    !---------timing  Get the buffer ID that has the time closest to the pitch action time
   IND_Time_IPC(:,IBeam)    = ABS(LocalVar%REWS_f_Time_IPC(:,IBeam) - LocalVar%Time - LocalVar%T_filter - LocalVar%T_filter_IPC -  LocalVar%T_PA - CntrPar%T_scan/2)    ! This d
   It                   =  INT(MINLOC(IND_Time_IPC(:,IBeam), 1))                                                                    ! This determines the position in the REWS array that needs to be cal

   LocalVar%REWS_b_IPC(IBeam)  = LocalVar%REWS_f_IPC(It, IBeam)                                                                     !REWS_b is the wind speed AT the desired time
   LocalVar%FF_Pitch_IPC(IBeam) = interp1d(CntrPar%REWS_curve,CntrPar%Pitch_curve, LocalVar%REWS_b_IPC(IBeam), ErrVar)

   !--------Determine the current azimuth position of the blades
   DO I=1,LocalVar%NumBl
        BlAzimuth = MODULO(LocalVar%Azimuth + (2*Pi*(I-1))/LocalVar%NumBl, 2*Pi)
   !--------Anticipate the blade azimuth positions after the preview time based on the rotor speed
        BlAzimuth_New = MODULO(BlAzimuth + (LocalVar%Rotspeed * LocalVar%T_filter_IPC), 2*Pi)
        AzimuthDiff = LocalVar%BeamAzimuth(LoopStart:5) - BlAzimuth
        BeamID = INT(MINLOC(ABS(AzimuthDiff), 1))

     IF(LoopStart == 2) THEN
       LocalVar%IPC_PitComFF(I) =  LocalVar%FF_Pitch_IPC(BeamID+1) - (LocalVar%BlPitch(1)+ LocalVar%BlPitch(2)+ LocalVar%BlPitch(3))/3
     ELSE
       LocalVar%IPC_PitComFF(I) =  LocalVar%FF_Pitch_IPC(BeamID) - (LocalVar%BlPitch(1)+ LocalVar%BlPitch(2)+ LocalVar%BlPitch(3))/3
     ENDIF

       LocalVar%IPC_Com_FF(I) = LPFilter(LocalVar%IPC_PitComFF(I), LocalVar%DT, LocalVar%w_cutoff_IPC, LocalVar%FP, LocalVar%iStatus, .FALSE., objInst%instLPF) * CntrPar%Kp_IPC

      IF (LocalVar%REWS_b_IPC(5) < 13) THEN !Only applied to the WindCrete
            LocalVar%IPC_Com_FF(I) = 0
      ENDIF


       DebugVar%T_filter_IPC = LocalVar%T_filter_IPC
       DebugVar%IPC_Com_1 = LocalVar%IPC_PitComFF(1)
       DebugVar%IPC_Com_2 = LocalVar%IPC_PitComFF(2)
       DebugVar%IPC_Com_3 = LocalVar%IPC_PitComFF(3)
       DebugVar%IPC_Com_1_f = LocalVar%IPC_Com_FF(1)
       DebugVar%IPC_Com_2_f = LocalVar%IPC_Com_FF(2)
       DebugVar%IPC_Com_3_f = LocalVar%IPC_Com_FF(3)

   END DO
   END DO

END SUBROUTINE FeedForwardIPC
```

Figure B.32: FFIPC subroutine added within the `Controllers.f90` script.

```
26      SUBROUTINE PitchControl(avrSWAP, CntrPar, LocalVar, objInst, DebugVar, ErrVar)
27      ! Blade pitch controller, generally maximizes rotor speed below rated (region 2) and regulates rotor speed above rated (region 3)
28      !       PC_State = 0, fix blade pitch to fine pitch angle (PC_FinePit)
29      !       PC_State = 1, is gain scheduled PI controller
30          USE ROSCO_Types, ONLY : ControlParameters, LocalVariables, ObjectInstances, DebugVariables, ErrorVariables
31
32          ! Inputs
33          REAL(ReKi),             INTENT(INOUT)       :: avrSWAP(*)   ! The swap array, used to pass data to, and receive data from the
34          TYPE(ControlParameters),    INTENT(INOUT)       :: CntrPar
35          TYPE(LocalVariables),       INTENT(INOUT)       :: LocalVar
36          TYPE(ObjectInstances),      INTENT(INOUT)       :: objInst
37          TYPE(DebugVariables),       INTENT(INOUT)       :: DebugVar
38          TYPE(ErrorVariables),       INTENT(INOUT)       :: ErrVar
39
40          ! Allocate Variables:
41          INTEGER(IntKi)                                  :: K            ! Index used for looping through blades.
42
43          CHARACTER(*),               PARAMETER           :: RoutineName = 'PitchControl'
44
45          ! ------- Blade Pitch Controller --------
46          ! Load PC State
47          IF (LocalVar%PC_State == 1) THEN ! PI BldPitch control
48              LocalVar%PC_MaxPit = CntrPar%PC_MaxPit
49          ELSE ! debug mode, fix at fine pitch
50              LocalVar%PC_MaxPit = CntrPar%PC_FinePit
51          END IF
52
53
54          ! Compute (interpolate) the gains based on previously commanded blade pitch angles and lookup table:
55          LocalVar%PC_KP = interp1d(CntrPar%PC_GS_angles, CntrPar%PC_GS_KP, LocalVar%PC_PitComTF, ErrVar) ! Proportional gain
56          LocalVar%PC_KI = interp1d(CntrPar%PC_GS_angles, CntrPar%PC_GS_KI, LocalVar%PC_PitComTF, ErrVar) ! Integral gain
57          LocalVar%PC_KD = interp1d(CntrPar%PC_GS_angles, CntrPar%PC_GS_KD, LocalVar%PC_PitComTF, ErrVar) ! Derivative gain
58          LocalVar%PC_TF = interp1d(CntrPar%PC_GS_angles, CntrPar%PC_GS_TF, LocalVar%PC_PitComTF, ErrVar) ! TF gains (derivative filter)
59
60          ! Compute the cut off frequency for FF control based on the coherent wave number and the reference average wind speed:
61          LocalVar%f_cutoff = (CntrPar%CohWaveNum*LocalVar%URefLid)/(2*Pi)
62          LocalVar%w_cutoff = LocalVar%URefLid*CntrPar%CohWaveNum
63
64
65         ! Compute the collective pitch command associated with the proportional and integral gains:
66
67          IF (LocalVar%iStatus == 0) THEN
68               LocalVar%PC_PitComT = PIController(LocalVar%PC_SpdErr, LocalVar%PC_KP, LocalVar%PC_KI, LocalVar%PC_MinPit, LocalVar%PC_M
69               LocalVar%REWS_f(1)    = LPFilter(LocalVar%LidSpeed(1), LocalVar%DT, LocalVar%w_cutoff, LocalVar%FP, LocalVar%iStatus, .T
70               LocalVar%FF_PitchRate = 0
71          ELSE
72              IF (CntrPar%LIDAR_ControlMode >= 1)  THEN
73
74                  CALL LidarConfiguration(LocalVar, CntrPar, objInst)
75
76                  IF ((CntrPar%LIDAR_ControlMode == 2) .AND. (ABS(LocalVar%MsrPosition(2,1) > 20))) THEN
77                                  ErrVar%aviFAIL = -1
78                                  ErrVar%ErrMsg  = 'When using combined LIDAR CPC and IPC, the first measurement point must be focussed
79                  END IF
80
81                  CALL FeedforwardCPC(LocalVar, CntrPar, DebugVar, objInst, ErrVar)
82                       LocalVar%PC_PitComT = PIControllerFF(LocalVar%PC_SpdErr, LocalVar%PC_KP, LocalVar%PC_KI, LocalVar%FF_PitchRate,
83              ELSE
84                       LocalVar%PC_PitComT = PIController(LocalVar%PC_SpdErr, LocalVar%PC_KP, LocalVar%PC_KI,LocalVar%PC_MinPit, LocalV
85              END IF
86
87           END IF
88           DebugVar%PC_PICommand = LocalVar%PC_PitComT
89
90          ! Find individual pitch control contribution
91          IF ((CntrPar%IPC_ControlMode >= 1) .OR. (CntrPar%Y_ControlMode == 2)) THEN
92
93              CALL IPC(CntrPar, LocalVar, objInst, DebugVar, ErrVar)
94
95              IF (CntrPar%LIDAR_ControlMode == 2) THEN
96
97                  CALL LidarConfiguration(LocalVar, CntrPar, objInst)
98                  CALL FeedforwardIPC(LocalVar, CntrPar, DebugVar, objInst, ErrVar)
99
00                  LocalVar%IPC_PitComF = LocalVar%IPC_PitComF + LocalVar%IPC_Com_FF
01
02              END IF
03
04          ELSEIF ((CntrPar%LIDAR_ControlMode == 2) .AND. ((CntrPar%IPC_ControlMode == 0) .AND. (CntrPar%Y_ControlMode == 0)))  THEN
05
```

Figure B.33: Complete pitch control subroutine (part 1), which calls to the LIDAR configuration, FFCPC and FFIPC subroutines. The green box indicates the part of the code that was added/modified within this research project.

```
106             CALL LidarConfiguration(LocalVar, CntrPar, objInst)
107             CALL FeedforwardIPC(LocalVar, CntrPar, DebugVar, objInst, ErrVar)
108
109              LocalVar%IPC_PitComF =  LocalVar%IPC_Com_FF
110
111         ELSE
112             LocalVar%IPC_PitComF = 0.0 ! THIS IS AN ARRAY!!
113         END IF
114
115         ! Include tower fore-aft tower vibration damping control
116         IF ((CntrPar%TD_Mode > 0) .OR. (CntrPar%Y_ControlMode == 2)) THEN
117             CALL ForeAftDamping(CntrPar, LocalVar, objInst)
118         ELSE
119             LocalVar%FA_PitCom = 0.0 ! THIS IS AN ARRAY!!
120         ENDIF
121
122         ! Pitch Saturation
123         IF (CntrPar%PS_Mode > 0) THEN
124             LocalVar%PC_MinPit = PitchSaturation(LocalVar,CntrPar,objInst,DebugVar, ErrVar)
125             LocalVar%PC_MinPit = max(LocalVar%PC_MinPit, CntrPar%PC_FinePit)
126         ELSE
127             LocalVar%PC_MinPit = CntrPar%PC_FinePit
128         ENDIF
129         DebugVar%PC_MinPit = LocalVar%PC_MinPit
130
131         ! FloatingFeedback
132         IF (CntrPar%Fl_Mode > 0) THEN
133             LocalVar%Fl_PitCom = FloatingFeedback(LocalVar, CntrPar, objInst)
134             DebugVar%Fl_PitCom = LocalVar%Fl_PitCom
135             LocalVar%PC_PitComT = LocalVar%PC_PitComT + LocalVar%Fl_PitCom
136         ENDIF
137
138         ! Shutdown
139         IF (CntrPar%SD_Mode == 1) THEN
140             LocalVar%PC_PitComT = Shutdown(LocalVar, CntrPar, objInst)
141         ENDIF
142
143         ! Saturate collective pitch commands:
144         LocalVar%PC_PitComT = saturate(LocalVar%PC_PitComT, LocalVar%PC_MinPit, CntrPar%PC_MaxPit)                    ! Satu
145         LocalVar%PC_PitComT = ratelimit(LocalVar%PC_PitComT, LocalVar%PC_PitComT_Last, CntrPar%PC_MinRat, CntrPar%PC_MaxRat,
146         LocalVar%PC_PitComT_Last = LocalVar%PC_PitComT
147
148         ! Combine and saturate all individual pitch commands in software
149         DO K = 1,LocalVar%NumBl ! Loop through all blades, add IPC contribution and limit pitch rate
150             LocalVar%PitCom(K) = LocalVar%PC_PitComT + LocalVar%FA_PitCom(K)
151             LocalVar%PitCom(K) = saturate(LocalVar%PitCom(K), LocalVar%PC_MinPit, CntrPar%PC_MaxPit)                  ! Sa
152             LocalVar%PitCom(K) = LocalVar%PitCom(K) + LocalVar%IPC_PitComF(K)                                        ! Add
153             LocalVar%PitCom(K) = saturate(LocalVar%PitCom(K), LocalVar%PC_MinPit, CntrPar%PC_MaxPit)                  ! Sa
154             LocalVar%PitCom(K) = ratelimit(LocalVar%PitCom(K), LocalVar%BlPitch(K), CntrPar%PC_MinRat, CntrPar%PC_MaxRat, Lo
155         END DO
156
157         ! Open Loop control, use if
158         !   Open loop mode active         Using OL blade pitch control
159         IF ((CntrPar%OL_Mode == 1) .AND. (CntrPar%Ind_BldPitch > 0)) THEN
160             IF (LocalVar%Time >= CntrPar%OL_Breakpoints(1)) THEN    ! Time > first open loop breakpoint
161                 DO K = 1,LocalVar%NumBl ! Loop through all blades
162                     LocalVar%PitCom(K) = interp1d(CntrPar%OL_Breakpoints,CntrPar%OL_BldPitch,LocalVar%Time, ErrVar)
163                 END DO
164             ENDIF
165         ENDIF
166
167         ! Place pitch actuator here, so it can be used with or without open-loop
168         DO K = 1,LocalVar%NumBl ! Loop through all blades, add IPC contribution and limit pitch rate
169             IF (CntrPar%PA_Mode > 0) THEN
170                 IF (CntrPar%PA_Mode == 1) THEN
171                     LocalVar%PitComAct(K) = LPFilter(LocalVar%PitCom(K), LocalVar%DT, CntrPar%PA_CornerFreq, LocalVar%FP, Lo
172                 ELSE IF (CntrPar%PA_Mode == 2) THEN
173                     LocalVar%PitComAct(K) = SecLPFilter(LocalVar%PitCom(K),LocalVar%DT,CntrPar%PA_CornerFreq,CntrPar%PA_Damp
174                 END IF
175             ELSE
176                 LocalVar%PitComAct(K) = LocalVar%PitCom(K)
177             ENDIF
178         END DO
179
180         ! Hardware saturation: using CntrPar%PC_MinPit
181         DO K = 1,LocalVar%NumBl ! Loop through all blades, add IPC contribution and limit pitch rate
182             ! Saturate the pitch command using the overall (hardware) limit
183             LocalVar%PitComAct(K) = saturate(LocalVar%PitComAct(K), LocalVar%PC_MinPit, CntrPar%PC_MaxPit)
184             ! Saturate the overall command of blade K using the pitch rate limit
185             LocalVar%PitComAct(K) = ratelimit(LocalVar%PitComAct(K), LocalVar%BlPitch(K), CntrPar%PC_MinRat, CntrPar%PC_MaxR
186         END DO
```

Figure B.34: Complete pitch control subroutine (part 2), which calls to the LIDAR configuration, FFCPC and FFIPC subroutines. The green box indicates the part of the code that was added/modified within this research project.

```
187
188         ! Command the pitch demanded from the last
189         ! call to the controller (See Appendix A of Bladed User's Guide):
190         avrSWAP(42) = LocalVar%PitComAct(1)    ! Use the command angles of all blades if using individual pitch
191         avrSWAP(43) = LocalVar%PitComAct(2)    ! "
192         avrSWAP(44) = LocalVar%PitComAct(3)    ! "
193         avrSWAP(45) = LocalVar%PitComAct(1)    ! Use the command angle of blade 1 if using collective pitch
194
195         ! Add RoutineName to error message
196         IF (ErrVar%aviFAIL < 0) THEN
197             ErrVar%ErrMsg = RoutineName//':'//TRIM(ErrVar%ErrMsg)
198         ENDIF
199
200
201     END SUBROUTINE PitchControl
```

Figure B.35: Complete pitch control subroutine (part 3), which calls to the LIDAR configuration, FFCPC and FFIPC subroutines.

B.3 WEIS

```
669         # LIDAR parameters
670         f.readline()
671         self.fst_vt['InflowWind']['SensorType'] = int(f.readline().split()[0])
672         self.fst_vt['InflowWind']['NumPulseGate'] = int(f.readline().split()[0])
673         self.fst_vt['InflowWind']['PulseSpacing'] = float_read(f.readline().split()[0])
674         self.fst_vt['InflowWind']['NumBeam'] = int(f.readline().split()[0])
675         self.fst_vt['InflowWind']['FocalDistanceX'] = [idx.strip() for idx in
676                                                        f.readline().split('FocalDistanceX')[0].split(',')]
677         self.fst_vt['InflowWind']['FocalDistanceY'] = [idx.strip() for idx in
678                                                        f.readline().split('FocalDistanceY')[0].split(',')]
679         self.fst_vt['InflowWind']['FocalDistanceZ'] = [idx.strip() for idx in
680                                                        f.readline().split('FocalDistanceZ')[0].split(',')]
681         self.fst_vt['InflowWind']['RotorApexOffsetPos'] = [idx.strip() for idx in
682                                                        f.readline().split('RotorApexOffsetPos')[0].split(',')]
683         self.fst_vt['InflowWind']['URefLid'] = float_read(f.readline().split()[0])
684         self.fst_vt['InflowWind']['MeasurementInterval'] = float_read(f.readline().split()[0])
685         self.fst_vt['InflowWind']['LidRadialVel'] = bool_read(f.readline().split()[0])
686         self.fst_vt['InflowWind']['ConsiderHubMotion'] = int(f.readline().split()[0])
```

Figure B.36: Code added within the `FAST_Reader.py` script to enable parsing of LIDAR simulator inputs the `modelling_schema.yaml` input file.

```
651     '-------------   LIDAR Parameters   ----------------------------------------------------------------------\n')
652 .write('{:<22} {:<11} {:}'.format(self.fst_vt['InflowWind']['SensorType'], 'SensorType', '- Switch for lidar configuration (0 = None, 1 = Single Point Beam(s), 2 = Conti
653 .write('{:<22} {:<11} {:}'.format(self.fst_vt['InflowWind']['NumPulseGate'], 'NumPulseGate', '- Number of lidar measurement gates (used when SensorType = 3)\n'))
654 .write('{:<22} {:<11} {:}'.format(self.fst_vt['InflowWind']['PulseSpacing'], 'PulseSpacing', '- Distance between range gates (m) (used when SensorType = 3)\n'))
655 .write('{:<22} {:<11} {:}'.format(self.fst_vt['InflowWind']['NumBeam'], 'NumBeam', '- NUmber of lidar measurement beams (0-5)(used when SensorType = 1)\n'))
656 .write('{:<22} {:<11} {:}'.format(', '.join(np.array(self.fst_vt['InflowWind']['FocalDistanceX'], dtype=str)), 'FocalDistanceX', '- Focal distance co-ordinates of the li
657 .write('{:<22} {:<11} {:}'.format(', '.join(np.array(self.fst_vt['InflowWind']['FocalDistanceY'], dtype=str)), 'FocalDistanceY', '- Focal distance co-ordinates of the li
658 .write('{:<22} {:<11} {:}'.format(', '.join(np.array(self.fst_vt['InflowWind']['FocalDistanceZ'], dtype=str)), 'FocalDistanceZ', '- Focal distance co-ordinates of the li
659 .write('{:<22} {:<11} {:}'.format(', '.join(np.array(self.fst_vt['InflowWind']['RotorApexOffsetPos'], dtype=str)), 'RotorApexOffsetPos', '- Offset of the lidar from hub
660 .write('{:<22} {:<11} {:}'.format(self.fst_vt['InflowWind']['URefLid'], 'URefLid', '- Reference average wind speed for the lidar[m/s]\n'))
661 .write('{:<22} {:<11} {:}'.format(self.fst_vt['InflowWind']['MeasurementInterval'], 'MeasurementInterval', '- Time between each measurement [s]\n'))
662 .write('{!s:<22} {:<11} {:}'.format(self.fst_vt['InflowWind']['LidRadialVel'], 'LidRadialVel', '- TRUE => return radial component, FALSE => return x direction estimate\r
663 .write('{:<22} {:<11} {:}'.format(self.fst_vt['InflowWind']['ConsiderHubMotion'], 'ConsiderHubMotion', '- Flag whether to consider the hub motions impact on Lidar measur
```

Figure B.37: Code added within the `FAST_Writing.py` script to enable writing of LIDAR simulator inputs to the InflowWind input file.

```
# Wind Sensor Measurements
InflowWind['WindMeas1']        = True     # (m/s); Wind measurement at sensor 1; Defined by sensor
InflowWind['WindMeas2']        = True     # (m/s); Wind measurement at sensor 2; Defined by sensor
InflowWind['WindMeas3']        = True     # (m/s); Wind measurement at sensor 3; Defined by sensor
InflowWind['WindMeas4']        = True     # (m/s); Wind measurement at sensor 4; Defined by sensor
InflowWind['WindMeas5']        = True     # (m/s); Wind measurement at sensor 5; Defined by sensor
```

Figure B.38: Enabling of LIDAR simulation wind measurement outputs within the `FAST_Vars_Out.py` script to be output within the `.out` file of OpenFAST.

```
    file.write('!------- LIDAR-ASSISTED CONTROL ----------------------------------------------------------\n')
    file.write('{:<13.5f}       ! Hub_Height     - Turbine hub height, [m]\n'.format(rosco_vt['Hub_Height']))
    file.write('{:<13.5f}       ! CohWaveNum     - Coherent wave number, [rad/m]\n'.format(rosco_vt['CohWaveNum']))
    file.write('{:<13.5f}       ! f_cutoff_IPC   - Corner frequency (-3dB point) in the low-pass filters, [rad/s]\n'.format(rosco_vt['f_cutoff_IPC']))
    file.write('{:<13.5f}       ! f_delay        -  Critical frequency near where the rotor has higher fluctuations, [rad/s]\n'.format(rosco_vt['f_delay']))
    file.write('{:<13.5f}       ! T_scan         - Duration to complete one full scan, [s]\n'.format(rosco_vt['T_scan']))
    file.write('{:<11d}         ! NumPitchCurveFF - Number of points in the wind-pitch-Kp curve\n'.format(rosco_vt['NumPitchCurveFF']))
    file.write('{}              ! REWS_curve      - Points of wind spped in the wind-pitch curve, [m/s]\n'.format(
        ''.join('{:<4.6f}  '.format(rosco_vt['REWS_curve'][i]) for i in range(len(rosco_vt['REWS_curve'])))))
    file.write('{}              ! Pitch_curve     - Points of pitch in the wind-pitch curve, [rad]\n'.format(
        ''.join('{:<4.6f}  '.format(rosco_vt['Pitch_curve'][i]) for i in range(len(rosco_vt['Pitch_curve'])))))
    file.write('\n')
```

Figure B.39: Code added within the `Utilities.py` script to enable writing of LIDAR-assisted control inputs to the ROSCO input file.

```
        # Controller Flags
        self.LoggingLevel       = controller_params['LoggingLevel']
        self.F_LPFType          = controller_params['F_LPFType']
        self.F_NotchType        = controller_params['F_NotchType']
        self.IPC_ControlMode    = controller_params['IPC_ControlMode']
        self.VS_ControlMode     = controller_params['VS_ControlMode']
        self.PC_ControlMode     = controller_params['PC_ControlMode']
        self.Y_ControlMode      = controller_params['Y_ControlMode']
        self.LIDAR_ControlMode  = controller_params['LIDAR_ControlMode']
        self.SS_Mode            = controller_params['SS_Mode']
        self.WE_Mode            = controller_params['WE_Mode']
        self.PS_Mode            = controller_params['PS_Mode']
        self.SD_Mode            = controller_params['SD_Mode']
        self.Fl_Mode            = controller_params['Fl_Mode']
        self.TD_Mode            = controller_params['TD_Mode']
        self.Flp_Mode           = controller_params['Flp_Mode']
        self.PA_Mode           = controller_params['PA_Mode']
        self.Ext_Mode           = controller_params['Ext_Mode']
        self.ZMQ_Mode           = controller_params['ZMQ_Mode']
```

Figure B.40: Code added within the `Controllers.py` script to enable LIDAR control flag within the ROSCO input file.

Appendix C

Dataset repository access

This Appendix provides links to repositories containing the relevant source code and gathered datasets pertaining to the simulation studies documented in this thesis:

- The original modified version of OpenFAST v.3.4, which includes the LIDAR simulator used in this work can be accessed via `https://github.com/Russell9798/OpenFAST-v3.4-Lidar-IfW-Original/releases/tag/2.0`.
- The official OpenFAST release including the LIDAR simulator changes can be accessed via `https://github.com/OpenFAST/openfast/releases/tag/v3.5.0`.
- The modified version of ROSCO v2.6 with the feedforward control additions can be accessed via `https://github.com/Russell9798/ROSCO-v2.6-LAC-IPC/releases/tag/2.0`.
- The VolturnUS-S OpenFAST LAC simulation datasets that were obtained and analysed within Chapter 5 of this thesis can be assessed via `https://doi.org/10.17605/OSF.IO/FE6YP`.
- The WindCrete OpenFAST LAC simulation datasets that were obtained and analysed within Chapter 5 of this thesis can be assessed via `https://doi.org/10.17605/OSF.IO/5RTX8`.
- The OpenFAST simulation datasets acquired from the experiments performed on the 15 MW VolturnUS-S with modified tower geometry and used within the tower design case study (Chapter 7) can be accessed via:
  - `https://zenodo.org/doi/10.5281/zenodo.10961265` (DLC 1.2 with aligned wind and waves),
  - `https://doi.org/10.17605/OSF.IO/2W8T5` (DLC 1.2 with misaligned wind and waves) and
  - `https://zenodo.org/doi/10.5281/zenodo.10927038` (DLCs 1.6 and 6.1).
- The OpenFAST and FAST.Farm simulation datasets obtained and analysed within the array-level case study (Chapter 8) can be accessed via `https://doi.org/10.17605/OSF.IO/S4EBF`.

Appendix D

Simulation input files

This Appendix provides screenshots of example input files used within the OpenFAST and FAST.Farm simulation studies, as well as an example MLife input file used for the post-processing of the OpenFAST simulation outputs.

D.1 OpenFAST

```
------- OpenFAST INPUT FILE -------------------------------------------
IEA 15 MW offshore reference model on UMaine VolturnUS-S semi-submersible floating platform
---------------------- SIMULATION CONTROL --------------------------------------
False                  Echo        - Echo input data to <RootName>.ech (flag)
"FATAL"                AbortLevel  - Error level when simulation should abort (string) {"WARNING", "SEVERE", "FATAL"}
3800                   TMax        - Total run time (s)
0.025                  DT          - Integration time step (s)
2                      InterpOrder - Interpolation order for input/output time history (-) {1=linear, 2=quadratic}
0                      NumCrctn    - Number of correction iterations (-) {0=explicit calculation, i.e., no corrections}
99999.0                DT_UJac     - Time between calls to get Jacobians (s)
1000000.0              UJacSclFact - Scaling factor used in Jacobians (-)
---------------------- FEATURE SWITCHES AND FLAGS ------------------------------
      1   CompElast      - Compute structural dynamics (switch) {1=ElastoDyn; 2=ElastoDyn + BeamDyn for blades}
      1   CompInflow     - Compute inflow wind velocities (switch) {0=still air; 1=InflowWind; 2=external from OpenFOAM}
      2   CompAero       - Compute aerodynamic loads (switch) {0=None; 1=AeroDyn v14; 2=AeroDyn v15}
      1   CompServo      - Compute control and electrical-drive dynamics (switch) {0=None; 1=ServoDyn}
      1   CompHydro      - Compute hydrodynamic loads (switch) {0=None; 1=HydroDyn}
      0   CompSub        - Compute sub-structural dynamics (switch) {0=None; 1=SubDyn; 2=External Platform MCKF}
      3   CompMooring    - Compute mooring system (switch) {0=None; 1=MAP++; 2=FEAMooring; 3=MoorDyn; 4=OrcaFlex}
      0   CompIce        - Compute ice loads (switch) {0=None; 1=IceFloe; 2=IceDyn}
      0   MHK            - MHK turbine type (switch) {0=Not an MHK turbine; 1=Fixed MHK turbine; 2=Floating MHK turbine}
---------------------- ENVIRONMENTAL CONDITIONS --------------------------------
9.81                   Gravity     - Gravitational acceleration (m/s^2)
1.225                  AirDens     - Air density (kg/m^3)
1025                   WtrDens     - Water density (kg/m^3)
1.464e-05              KinVisc     - Kinematic viscosity of working fluid (m^2/s)
335                    SpdSound    - Speed of sound in working fluid (m/s)
103500                 Patm        - Atmospheric pressure (Pa) [used only for an MHK turbine cavitation check]
1700                   Pvap        - Vapour pressure of working fluid (Pa) [used only for an MHK turbine cavitation check]
200                     WtrDpth    - Water depth (m)
0                      MSL2SWL     - Offset between still-water level and mean sea level (m) [positive upward]
---------------------- INPUT FILES ---------------------------------------------
"IEA-15-240-RWT-UMaineSemi_ElastoDyn.dat"    EDFile          - Name of file containing ElastoDyn input parameters (quoted string)
""                     BDBldFile(1) - Name of file containing BeamDyn input parameters for blade 1 (quoted string)
""                     BDBldFile(2) - Name of file containing BeamDyn input parameters for blade 2 (quoted string)
""                     BDBldFile(3) - Name of file containing BeamDyn input parameters for blade 3 (quoted string)
"IEA-15-240-RWT_InflowFile.dat"    InflowFile      - Name of file containing inflow wind input parameters (quoted string)
"../IEA-15-240-RWT/IEA-15-240-RWT_AeroDyn15.dat"   AeroFile     - Name of file containing aerodynamic input parameters (quoted string)
"IEA-15-240-RWT-UMaineSemi_ServoDyn.dat"    ServoFile       - Name of file containing control and electrical-drive input parameters (quoted string)
"IEA-15-240-RWT-UMaineSemi_HydroDyn.dat"      HydroFile       - Name of file containing hydrodynamic input parameters (quoted string)
"none"                 SubFile          - Name of file containing sub-structural input parameters (quoted string)
"IEA-15-240-RWT-UMaineSemi_MoorDyn.dat"      MooringFile     - Name of file containing mooring system input parameters (quoted string)
"none"                 IceFile         - Name of file containing ice input parameters (quoted string)
---------------------- OUTPUT --------------------------------------------------
False                  SumPrint    - Print summary data to "<RootName>.sum" (flag)
100.0                  SttsTime    - Amount of time between screen status messages (s)
99999.0                ChkptTime   - Amount of time between creating checkpoint files for potential restart (s)
default                DT_Out      - Time step for tabular output (s) (or "default")
200.000000              TStart     - Time to begin tabular output (s)
1                      OutFileFmt  - Format for tabular (time-marching) output file (switch) {1: text file [<RootName>.out], 2: binary file [<RootName>.outb], 3: both}
True                   TabDelim    - Use tab delimiters in text tabular output file? (flag) {uses spaces if false}
"ES10.3E2"             OutFmt      - Format used for text tabular output, excluding the time channel.  Resulting field should be 10 characters. (quoted string)
---------------------- LINEARIZATION -------------------------------------------
False                  Linearize   - Linearization analysis (flag)
False                  CalcSteady  - Calculate a steady-state periodic operating point before linearization? [unused if Linearize=False] (flag)
3                      TrimCase    - Controller parameter to be trimmed {1:yaw; 2:torque; 3:pitch} [used only if CalcSteady=True] (-)
0.001                  TrimTol     - Tolerance for the rotational speed convergence [used only if CalcSteady=True] (-)
0.01                   TrimGain    - Proportional gain for the rotational speed error (>0) [used only if CalcSteady=True] (rad/(rad/s) for yaw or pitch; Nm/(rad/s) for torque)
0                      Twr_Kdmp    - Damping factor for the tower [used only if CalcSteady=True] (N/(m/s))
0                      Bld_Kdmp    - Damping factor for the blades [used only if CalcSteady=True] (N/(m/s))
2                      NLinTimes   - Number of times to linearize (-) [>=1] [unused if Linearize=False]
30.000000, 60.000000   LinTimes    - List of times at which to linearize (s) [1 to NLinTimes] [used only when Linearize=True and CalcSteady=False]
1                      LinInputs   - Inputs included in linearization (switch) {0=none; 1=standard; 2=all module inputs (debug)} [unused if Linearize=False]
1                      LinOutputs  - Outputs included in linearization (switch) {0=none; 1=from OutList(s); 2=all module outputs (debug)} [unused if Linearize=False]
False                  LinOutJac   - Include full Jacobians in linearization output (for debug) (flag) [unused if Linearize=False; used only if LinInputs=LinOutputs=2]
False                  LinOutMod   - Write module-level linearization output files in addition to output for full system? (flag) [unused if Linearize=False]
---------------------- VISUALIZATION -------------------------------------------
0                      WrVTK       - VTK visualization data output: (switch) {0=none; 1=initialization data only; 2=animation}
2                      VTK_type    - Type of VTK visualization data: (switch) {1=surfaces; 2=basic meshes (lines/points); 3=all meshes (debug)} [unused if WrVTK=0]
False                  VTK_fields  - Write mesh fields to VTK data files? (flag) {true/false} [unused if WrVTK=0]
15.0                   VTK_fps     - Frame rate for VTK output (frames per second){will use closest integer multiple of DT} [used only if WrVTK=2]
```

Figure D.1: Example of a master .fst OpenFAST input file used when performing the OpenFAST simulations.

```
------- InflowWind v3.01.* INPUT FILE ------------------------------------------------------------------------
IEA 15 MW Offshore Reference Turbine
---------------------------------------------------------------------------------------------------------------
False                   Echo        - Echo input data to <RootName>.ech (flag)
3                       WindType    - switch for wind file type (1=steady; 2=uniform; 3=binary TurbSim FF; 4=binary Bladed-style FF; 5=HAWC format; 6=User defined; 7=native Bladed FF)
0.0                     PropagationDir - Direction of wind propagation (meteorolgical rotation from aligned with X (positive rotates towards -Y) -- degrees)
0.0                     VFlowAng    - Upflow angle (degrees) (not used for native Bladed format WindType=7)
1                       NWindVel    - Number of points to output the wind velocity    (0 to 9)
0.0                     WindVxiList - List of coordinates in the inertial X direction (m)
0.0                     WindVyiList - List of coordinates in the inertial Y direction (m)
150                     WindVziList - List of coordinates in the inertial Z direction (m)
================== LIDAR Parameters ===========================================================================
1                       SensorType - Switch for lidar configuration (0 = None, 1 = Single Point Beam(s), 2 = Continuous, 3 = Pulsed)
3                       NumPulseGate - Number of lidar measurement gates (used when SensorType = 3)
30                      PulseSpacing - Distance between range gates (m) (used when SensorType = 3)
5                       NumBeam - Number of lidar measurement beams (0-5)(used when SensorType = 1)
-260,-260,-260,-260,-260     FocalDistanceX - Focal distance co-ordinates of the lidar in the x direction (relative to hub height) (only first coordinate used for SensorType 2 and 3) (m)
0,60,0,-60,0                 FocalDistanceY - Focal distance co-ordinates of the lidar in the y direction (relative to hub height) (only first coordinate used for SensorType 2 and 3) (m)
0,0,-60,0,60                 FocalDistanceZ - Focal distance co-ordinates of the lidar in the z direction (relative to hub height) (only first coordinate used for SensorType 2 and 3) (m)
0.0, 0.0, 0.0           RotorApexOffsetPos - Offset of the lidar from hub height (m)
13                      URefLid - Reference average wind speed for the lidar[m/s]
1                       MeasurementInterval - Time between each measurement [s]
False                   LidRadialVel - TRUE => return radial component, FALSE => return 'x' direction estimate
1                       ConsiderHubMotion - Flag whether to consider the hub motion's impact on Lidar measurements
================== Parameters for Steady Wind Conditions [used only for WindType = 1] =========================
10.0                        HWindSpeed  - Horizontal windspeed                            (m/s)
150.0                       RefHt       - Reference height for horizontal wind speed      (m)
0.12                        PLexp       - Power law exponent                              (-)
================== Parameters for Uniform wind file   [used only for WindType = 2] ============================
"1.1_NTM/EOGR+2.0.wnd"          Filename_Uni - Filename of time series data for uniform wind field.      (-)
150.0                       RefHt_Uni   - Reference height for horizontal wind speed              (m)
240.0                       RefLength   - Reference length for linear horizontal and vertical sheer (-)
================== Parameters for Binary TurbSim Full-Field files   [used only for WindType = 3] ==============
"1.1_NTM/TI_Category_B/Seed_1/DLC1.1_13ms_Seed1_1h.bts"    FileName_BTS - Name of the Full field wind file to use (.bts)
================== Parameters for Binary Bladed-style Full-Field files   [used only for WindType = 4] =========
"Kaimal_NTM_URef18_A_Ly280_Lz280_Ny33_Nz33_H150_T800_dt0d1_Seed41"    FilenameRoot - Rootname of the full-field wind file to use (.wnd, .sum)
False                       TowerFile   - Have tower file (.twr) (flag)
================== Parameters for HAWC-format binary files  [Only used with WindType = 5] =====================
"none"                      FileName_u  - name of the file containing the u-component fluctuating wind (.bin)
"none"                      FileName_v  - name of the file containing the v-component fluctuating wind (.bin)
"none"                      FileName_w  - name of the file containing the w-component fluctuating wind (.bin)
64                          nx          - number of grids in the x direction (in the 3 files above) (-)
32                          ny          - number of grids in the y direction (in the 3 files above) (-)
32                          nz          - number of grids in the z direction (in the 3 files above) (-)
16.0                        dx          - distance (in meters) between points in the x direction    (m)
3.0                         dy          - distance (in meters) between points in the y direction    (m)
3.0                         dz          - distance (in meters) between points in the z direction    (m)
150.0                       RefHt_Hawc  - reference height; the height (in meters) of the vertical center of the grid (m)
-------------   Scaling parameters for turbulence   ------------------------------------------------------------
2                           ScaleMethod - Turbulence scaling method   [0 = none, 1 = direct scaling, 2 = calculate scaling factor based on a desired standard deviation]
1.0                         SFx         - Turbulence scaling factor for the x direction (-)   [ScaleMethod=1]
1.0                         SFy         - Turbulence scaling factor for the y direction (-)   [ScaleMethod=1]
1.0                         SFz         - Turbulence scaling factor for the z direction (-)   [ScaleMethod=1]
1.2                         SigmaFx     - Turbulence standard deviation to calculate scaling from in x direction (m/s)    [ScaleMethod=2]
0.8                         SigmaFy     - Turbulence standard deviation to calculate scaling from in y direction (m/s)    [ScaleMethod=2]
0.2                         SigmaFz     - Turbulence standard deviation to calculate scaling from in z direction (m/s)    [ScaleMethod=2]
-------------   Mean wind profile parameters (added to HAWC-format files)   ----------------------------------
12.0                        URef        - Mean u-component wind speed at the reference height (m/s)
2                           WindProfile - Wind profile type (0=constant;1=logarithmic,2=power law)
0.2                         PLExp_Hawc  - Power law exponent (-) (used for PL wind profile type only)
0.03                        Z0          - Surface roughness length (m) (used for LG wind profile type only)
0                           XOffset     - Initial offset in +x direction (shift of wind box) (-)
====================== OUTPUT ==================================================
False                       SumPrint    - Print summary data to <RootName>.IfW.sum (flag)
OutList      - The next line(s) contains a list of output parameters.  See OutListParameters.xlsx for a listing of available output channels, (-)
Wind1VelX
Wind1VelY
Wind1VelZ
WindMeas1
WindMeas2
WindMeas3
WindMeas4
WindMeas5
END of input file (the word "END" must appear in the first 3 columns of this last OutList line)
---------------------------------------------------------------------------------------
```

Figure D.2: Example of an InflowWind input file used when performing the OpenFAST simulations.

```
------- ELASTODYN v1.03.* INPUT FILE -------------------------------------------
IEA 15 MW offshore reference model on UMaine VolturnUS-S semi-submersible floating platform
---------------------- SIMULATION CONTROL --------------------------------------
False         Echo        - Echo input data to "<RootName>.ech" (flag)
          3   Method      - Integration method: {1: RK4, 2: AB4, or 3: ABM4} (-)
"default"     DT          Integration time step (s)
---------------------- DEGREES OF FREEDOM --------------------------------------
True          FlapDOF1    - First flapwise blade mode DOF (flag)
True          FlapDOF2    - Second flapwise blade mode DOF (flag)
True          EdgeDOF     - First edgewise blade mode DOF (flag)
False         TeetDOF     - Rotor-teeter DOF (flag) [unused for 3 blades]
False         DrTrDOF     - Drivetrain rotational-flexibility DOF (flag)
True          GenDOF      - Generator DOF (flag)
False         YawDOF      - Yaw DOF (flag)
True          TwFADOF1    - First fore-aft tower bending-mode DOF (flag)
True          TwFADOF2    - Second fore-aft tower bending-mode DOF (flag)
True          TwSSDOF1    - First side-to-side tower bending-mode DOF (flag)
True          TwSSDOF2    - Second side-to-side tower bending-mode DOF (flag)
True          PtfmSgDOF   - Platform horizontal surge translation DOF (flag)
True          PtfmSwDOF   - Platform horizontal sway translation DOF (flag)
True          PtfmHvDOF   - Platform vertical heave translation DOF (flag)
True          PtfmRDOF    - Platform roll tilt rotation DOF (flag)
True          PtfmPDOF    - Platform pitch tilt rotation DOF (flag)
True          PtfmYDOF    - Platform yaw rotation DOF (flag)
---------------------- INITIAL CONDITIONS --------------------------------------
          0   OoPDefl     - Initial out-of-plane blade-tip displacement (meters)
          0   IPDefl      - Initial in-plane blade-tip deflection (meters)
         1.   BlPitch(1)  - Blade 1 initial pitch (degrees)
         1.   BlPitch(2)  - Blade 2 initial pitch (degrees)
         1.   BlPitch(3)  - Blade 3 initial pitch (degrees) [unused for 2 blades]
          0   TeetDefl    - Initial or fixed teeter angle (degrees) [unused for 3 blades]
          0   Azimuth     - Initial azimuth angle for blade 1 (degrees)
       7.55   RotSpeed    - Initial or fixed rotor speed (rpm)
          0   NacYaw      - Initial or fixed nacelle-yaw angle (degrees)
          0   TTDspFA     - Initial fore-aft tower-top displacement (meters)
          0   TTDspSS     - Initial side-to-side tower-top displacement (meters)
          0   PtfmSurge   - Initial or fixed horizontal surge translational displacement of platform (meters)
          0   PtfmSway    - Initial or fixed horizontal sway translational displacement of platform (meters)
          0   PtfmHeave   - Initial or fixed vertical heave translational displacement of platform (meters)
          0   PtfmRoll    - Initial or fixed roll tilt rotational displacement of platform (degrees)
          0   PtfmPitch   - Initial or fixed pitch tilt rotational displacement of platform (degrees)
          0   PtfmYaw     - Initial or fixed yaw rotational displacement of platform (degrees)
---------------------- TURBINE CONFIGURATION -----------------------------------
          3   NumBl       - Number of blades (-)
     120.97   TipRad      - The distance from the rotor apex to the blade tip (meters)
       3.97   HubRad      - The distance from the rotor apex to the blade root (meters)
         -4   PreCone(1)  - Blade 1 cone angle (degrees)
         -4   PreCone(2)  - Blade 2 cone angle (degrees)
         -4   PreCone(3)  - Blade 3 cone angle (degrees) [unused for 2 blades]
          0   HubCM       - Distance from rotor apex to hub mass [positive downwind] (meters)
          0   UndSling    - Undersling length [distance from teeter pin to the rotor apex] (meters) [unused for 3 blades]
          0   Delta3      - Delta-3 angle for teetering rotors (degrees) [unused for 3 blades]
          0   AzimB1Up    - Azimuth value to use for I/O when blade 1 points up (degrees)
    -12.098   OverHang    - Distance from yaw axis to rotor apex [3 blades] or teeter pin [2 blades] (meters)
        0.0   ShftGagL    - Distance from rotor apex [3 blades] or teeter pin [2 blades] to shaft strain gages [positive for upwind rotors] (meters)
       -6.0   ShftTilt    - Rotor shaft tilt angle (degrees)
    -4.7201   NacCMxn     - Downwind distance from the tower-top to the nacelle CM (meters)
        0.0   NacCMyn     - Lateral  distance from the tower-top to the nacelle CM (meters)
     4.2751   NacCMzn     - Vertical distance from the tower-top to the nacelle CM (meters)
        0.0   NcIMUxn     - Downwind distance from the tower-top to the nacelle IMU (meters)
        0.0   NcIMUyn     - Lateral  distance from the tower-top to the nacelle IMU (meters)
        0.0   NcIMUzn     - Vertical distance from the tower-top to the nacelle IMU (meters)
     4.3495   Twr2Shft    - Vertical distance from the tower-top to the rotor shaft (meters)
    144.386   TowerHt     - Height of tower above ground level [onshore] or MSL [offshore] (meters)
         15   TowerBsHt   - Height of tower base above ground level [onshore] or MSL [offshore] (meters)
          0   PtfmCMxt    - Downwind distance from the ground level [onshore] or MSL [offshore] to the platform CM (meters)
          0   PtfmCMyt    - Lateral distance from the ground level [onshore] or MSL [offshore] to the platform CM (meters)
    -14.400   PtfmCMzt    - Vertical distance from the ground level [onshore] or MSL [offshore] to the platform CM (meters)
          0   PtfmRefzt   - Vertical distance from the ground level [onshore] or MSL [offshore] to the platform reference point (meters)
---------------------- MASS AND INERTIA ----------------------------------------
          0   TipMass(1)  - Tip-brake mass, blade 1 (kg)
          0   TipMass(2)  - Tip-brake mass, blade 2 (kg)
          0   TipMass(3)  - Tip-brake mass, blade 3 (kg) [unused for 2 blades]
      69360   HubMass     - Hub mass (kg)
     973520   HubIner     - Hub inertia about rotor axis [3 blades] or teeter axis [2 blades] (kg m^2)
    1836784   GenIner     - Generator inertia about HSS (kg m^2)
     646895   NacMass     - Nacelle mass (kg)
   24240914   NacYIner    - Nacelle inertia about yaw axis (kg m^2)
      28280   YawBrMass   - Yaw bearing mass (kg)
1.7838E+07    PtfmMass    - Platform mass (kg)
1.2507E+10    PtfmRIner   - Platform inertia for roll tilt rotation about the platform CM (kg m^2)
1.2507E+10    PtfmPIner   - Platform inertia for pitch tilt rotation about the platform CM (kg m^2)
2.3667E+10    PtfmYIner   - Platform inertia for yaw rotation about the platform CM (kg m^2)
---------------------- BLADE ---------------------------------------------------
         50   BldNodes    - Number of blade nodes (per blade) used for analysis (-)
"../IEA-15-240-RWT/IEA-15-240-RWT_ElastoDyn_blade.dat"    BldFile1    - Name of file containing properties for blade 1 (quoted string)
"../IEA-15-240-RWT/IEA-15-240-RWT_ElastoDyn_blade.dat"    BldFile2    - Name of file containing properties for blade 2 (quoted string)
"../IEA-15-240-RWT/IEA-15-240-RWT_ElastoDyn_blade.dat"    BldFile3    - Name of file containing properties for blade 3 (quoted string) [unused for 2 blades]
---------------------- ROTOR-TEETER --------------------------------------------
          0   TeetMod     - Rotor-teeter spring/damper model {0: none, 1: standard, 2: user-defined from routine UserTeet} (switch) [unused for 3 blades]
          0   TeetDmpP    - Rotor-teeter damper position (degrees) [used only for 2 blades and when TeetMod=1]
          0   TeetDmp     - Rotor-teeter damping constant (N-m/(rad/s)) [used only for 2 blades and when TeetMod=1]
          0   TeetCDmp    - Rotor-teeter rate-independent Coulomb-damping moment (N-m) [used only for 2 blades and when TeetMod=1]
```

Figure D.3: Example of an ElastoDyn input file used when performing the OpenFAST simulations (part 1).

```
          0   TeetSStP    - Rotor-teeter soft-stop position (degrees) [used only for 2 blades and when TeetMod=1]
          0   TeetHStP    - Rotor-teeter hard-stop position (degrees) [used only for 2 blades and when TeetMod=1]
          0   TeetSSSp    - Rotor-teeter soft-stop linear-spring constant (N-m/rad) [used only for 2 blades and when TeetMod=1]
          0   TeetHSSp    - Rotor-teeter hard-stop linear-spring constant (N-m/rad) [used only for 2 blades and when TeetMod=1]
---------- --------- DRIVETRAIN --------------------------------------------------
        100   GBoxEff     - Gearbox efficiency (%)
          1   GBRatio     - Gearbox ratio (-)
51140939610   DTTorSpr    - Drivetrain torsional spring (N-m/rad)
   42676833   DTTorDmp    - Drivetrain torsional damper (N-m/(rad/s))
---------------------- FURLING -----------------------------------------------
False         Furling     - Read in additional model properties for furling turbine (flag) [must currently be FALSE)
"unused"      FurlFile    - Name of file containing furling properties (quoted string) [unused when Furling=False]
---------------------- TOWER -------------------------------------------------
         20   TwrNodes    - Number of tower nodes used for analysis (-)
"IEA-15-240-RWT-UMaineSemi_ElastoDyn_tower.dat"    TwrFile     - Name of file containing tower properties (quoted string)
---------------------- OUTPUT ------------------------------------------------
True         SumPrint    - Print summary data to "<RootName>.sum" (flag)
             1   OutFile     - Switch to determine where output will be placed: {1: in module output file only; 2: in glue code output file only; 3: both} (currently unused)
True             TabDelim    - Use tab delimiters in text tabular output file? (flag) (currently unused)
"ES10.3E2"       OutFmt      - Format used for text tabular output (except time).  Resulting field should be 10 characters. (quoted string) (currently unused)
             0   TStart      - Time to begin tabular output (s) (currently unused)
             1   DecFact     - Decimation factor for tabular output {1: output every time step} (-) (currently unused)
             1   NTwGages    - Number of tower nodes that have strain gages for output [0 to 9] (-)
            20   TwrGagNd    - List of tower nodes that have strain gages [1 to TwrNodes] (-) [unused if NTwGages=0] TwrGagNd    - List of tower nodes that have strain gages
             3   NBlGages    - Number of blade nodes that have strain gages for output [0 to 9] (-)
             5,          9,         13    BldGagNd    - List of blade nodes that have strain gages [1 to BldNodes] (-) [unused if NBlGages=0] BldGagNd    - List of blade node
                   OutList             - The next line(s) contains a list of output parameters.  See OutListParameters.xlsx for a listing of available output channels, (-)
"Azimuth"
"B1dPitch1"
"B1dPitch2"
"B1dPitch3"
"GenSpeed"
"IPDef11"
"LSSGagMya"
"LSSGagMza"
"LSSTipMys"
"LSSTipMzs"
"LSShftFys"
"LSShftFzs"
"NacYaw"
"NcIMUTAxs"
"NcIMUTAys"
"NcIMUTAzs"
"NcIMURAys"
"OoPDef11"
"PtfmHeave"
"PtfmPitch"
"PtfmRoll"
"PtfmSurge"
"PtfmSway"
"PtfmYaw"
"RootFxb1"
"RootFxb2"
"RootFxb3"
"RootFxc1"
"RootFxc2"
"RootFxc3"
"RootFyb1"
"RootFyb2"
"RootFyb3"
"RootFyc1"
"RootFyc2"
"RootFyc3"
"RootFzb1"
"RootFzb2"
"RootFzb3"
"RootFzc1"
"RootFzc2"
"RootFzc3"
"RootMxb1"
"RootMxb2"
"RootMxb3"
"RootMxc1"
"RootMxc2"
"RootMxc3"
"RootMyb1"
"RootMyb2"
"RootMyb3"
"RootMyc1"
"RootMyc2"
"RootMyc3"
"RootMzb1"
"RootMzb2"
"RootMzb3"
"RootMzc1"
"RootMzc2"
"RootMzc3"
"RotSpeed"
"RotThrust"
"RotTorq"
```

Figure D.4: Example of an ElastoDyn input file used when performing the OpenFAST simulations (part 2).

```
"Spn1MLxb1"
"Spn1MLyb1"
"Spn2MLxb1"
"Spn2MLyb1"
"Spn3MLxb1"
"Spn3MLyb1"
"TTDspTwst"
"TwHt1TPxi"
"TwHt1TPyi"
"TwrBsFxt"
"TwrBsFyt"
"TwrBsFzt"
"TwrBsMxt"
"TwrBsMyt"
"TwrBsMzt"
"TwstDefl1"
"YawBrMxp"
"YawBrMyp"
"YawBrMzp"
"YawBrTDxt"
"YawBrTDyt"
"LSShftMxa"
"LSSTipMya"
"LSSTipMza"
"LSSGagMxs"
"LSSGagMys"
"LSSGagMzs"
"YawBrMxn"
"YawBrMyn"
"YawBrMzn"
"TwHt1MLxt"
"TwHt1MLyt"
"TwHt1MLzt"
"Spn3MLzb1"
"Spn1MLxb2"
"Spn1MLyb2"
"Spn1MLzb2"
"Spn2MLxb2"
"Spn2MLyb2"
"Spn2MLzb2"
"Spn3MLxb2"
"Spn3MLyb2"
"Spn3MLzb2"
"Spn1MLxb3"
"Spn1MLyb3"
"Spn1MLzb3"
"Spn2MLxb3"
"Spn2MLyb3"
"Spn2MLzb3"
"Spn3MLxb3"
"Spn3MLyb3"
"Spn3MLzb3"
END of input file (the word "END" must appear in the first 3 columns of this last OutList line)
---------------------- NODE OUTPUTS --------------------------------------------
          1   BldNd_BladesOut  - Blades to output
         99   BldNd_BlOutNd    - Blade nodes on each blade (currently unused)
              OutList      - The next line(s) contains a list of output parameters.  See OutListParameters.xlsx, ElastoDyn_Nodes tab for a listing of available output channels, (-)
END of input file (the word "END" must appear in the first 3 columns of this last OutList line)
--------------------------------------------------------------------------------
```

Figure D.5: Example of an ElastoDyn input file used when performing the OpenFAST simulations (part 3).

```
------- ELASTODYN V1.00.* TOWER INPUT FILE ------------------------------------
IEA 15 MW offshore reference model on UMaine VolturnUS-S semi-submersible floating platform
---------------------- TOWER PARAMETERS ----------------------------------------
10                   NTwInpSt    - Number of input stations to specify tower geometry
1.0                  TwrFADmp(1) - Tower 1st fore-aft mode structural damping ratio (%)
1.0                  TwrFADmp(2) - Tower 2nd fore-aft mode structural damping ratio (%)
1.0                  TwrSSDmp(1) - Tower 1st side-to-side mode structural damping ratio (%)
1.0                  TwrSSDmp(2) - Tower 2nd side-to-side mode structural damping ratio (%)
---------------------- TOWER ADJUSTMUNT FACTORS --------------------------------
1.0                  FAStTunr(1) - Tower fore-aft modal stiffness tuner, 1st mode (-)
1.0                  FAStTunr(2) - Tower fore-aft modal stiffness tuner, 2nd mode (-)
1.0                  SSStTunr(1) - Tower side-to-side stiffness tuner, 1st mode (-)
1.0                  SSStTunr(2) - Tower side-to-side stiffness tuner, 2nd mode (-)
1.0                  AdjTwMa     - Factor to adjust tower mass density (-)
1.0                  AdjFASt     - Factor to adjust tower fore-aft stiffness (-)
1.0                  AdjSSSt     - Factor to adjust tower side-to-side stiffness (-)
---------------------- DISTRIBUTED TOWER PROPERTIES ----------------------------
  HtFract       TMassDen          TwFAStif       TwSSStif
    (-)          (kg/m)           (Nm^2)         (Nm^2)
 0.0000000000000000e+00  2.300642131074250e+04  6.770513741893661e+12  6.770513741893661e+12
 1.114034260838268e-01  2.038926791017103e+04  6.012509438660999e+12  6.012509438660999e+12
 2.228068521676536e-01  1.774601273898164e+04  5.243768263364071e+12  5.243768263364071e+12
 3.342102782514804e-01  1.506280308193614e+04  4.460141520157670e+12  4.460141520157670e+12
 4.456137043353072e-01  1.234240777801470e+04  3.662296012994249e+12  3.662296012994249e+12
 5.570171304191340e-01  9.633099266285813e+03  2.864340844739592e+12  2.864340844739592e+12
 6.684205565029608e-01  7.026968999232271e+03  2.093608918221600e+12  2.093608918221600e+12
 7.798239825867876e-01  4.645483574127345e+03  1.386599057692824e+12  1.386599057692824e+12
 8.912274086706145e-01  2.596093483450980e+03  7.761060822822759e+11  7.761060822822759e+11
 1.000000000000000e+00  1.715031343116859e+03  3.489844253733395e+11  3.489844253733395e+11
---------------------- TOWER FORE-AFT MODE SHAPES ------------------------------
0.9413693374950306     TwFAM1Sh(2) - Mode 1, coefficient of x^2 term
0.3468546636521823     TwFAM1Sh(3) -       , coefficient of x^3 term
-1.07324250422707761   TwFAM1Sh(4) -       , coefficient of x^4 term
1.3138921365298792     TwFAM1Sh(5) -       , coefficient of x^5 term
-0.5288736334500164    TwFAM1Sh(6) -       , coefficient of x^6 term
139.39115610617628     TwFAM2Sh(2) - Mode 2, coefficient of x^2 term
111.93048260721132     TwFAM2Sh(3) -       , coefficient of x^3 term
-656.9153343478758     TwFAM2Sh(4) -       , coefficient of x^4 term
845.7553274809915      TwFAM2Sh(5) -       , coefficient of x^5 term
-439.16163184650327    TwFAM2Sh(6) -       , coefficient of x^6 term
---------------------- TOWER SIDE-TO-SIDE MODE SHAPES --------------------------
0.928651504815363      TwSSM1Sh(2) - Mode 1, coefficient of x^2 term
0.3183110416167346     TwSSM1Sh(3) -       , coefficient of x^3 term
-0.9597593899992671    TwSSM1Sh(4) -       , coefficient of x^4 term
1.164027965180537      TwSSM1Sh(5) -       , coefficient of x^5 term
-0.4512311216133674    TwSSM1Sh(6) -       , coefficient of x^6 term
-1970.3543341226898    TwSSM2Sh(2) - Mode 2, coefficient of x^2 term
-2284.293209902991     TwSSM2Sh(3) -       , coefficient of x^3 term
11346.453188064506     TwSSM2Sh(4) -       , coefficient of x^4 term
-14750.85609454158     TwSSM2Sh(5) -       , coefficient of x^5 term
7660.050450502754      TwSSM2Sh(6) -       , coefficient of x^6 term
```

Figure D.6: Example of an ElastoDyn tower input file used when performing the OpenFAST simulations.

```
------- SERVODYN v1.05.* INPUT FILE -------------------------------------------
IEA 15 MW offshore reference model on UMaine VolturnUS-S semi-submersible floating platform
---------------------- SIMULATION CONTROL --------------------------------------
False                  Echo        - Echo input data to <RootName>.ech (flag)
"default"              DT          - Communication interval for controllers (s) (or "default")
---------------------- PITCH CONTROL -------------------------------------------
5                      PCMode      - Pitch control mode {0: none, 3: user-defined from routine PitchCntrl, 4: user-defined from Simulink/Labview, 5: user-defined
0.0                    TPCOn       - Time to enable active pitch control (s) [unused when PCMode=0]
9999.9                 TPitManS(1) - Time to start override pitch maneuver for blade 1 and end standard pitch control (s)
9999.9                 TPitManS(2) - Time to start override pitch maneuver for blade 2 and end standard pitch control (s)
9999.9                 TPitManS(3) - Time to start override pitch maneuver for blade 3 and end standard pitch control (s) [unused for 2 blades]
2.0                    PitManRat(1) - Pitch rate at which override pitch maneuver heads toward final pitch angle for blade 1 (deg/s)
2.0                    PitManRat(2) - Pitch rate at which override pitch maneuver heads toward final pitch angle for blade 2 (deg/s)
2.0                    PitManRat(3) - Pitch rate at which override pitch maneuver heads toward final pitch angle for blade 3 (deg/s) [unused for 2 blades]
0.0                    BlPitchF(1) - Blade 1 final pitch for pitch maneuvers (degrees)
0.0                    BlPitchF(2) - Blade 2 final pitch for pitch maneuvers (degrees)
0.0                    BlPitchF(3) - Blade 3 final pitch for pitch maneuvers (degrees) [unused for 2 blades]
---------------------- GENERATOR AND TORQUE CONTROL ----------------------------
5                      VSContrl    - Variable-speed control mode {0: none, 1: simple VS, 3: user-defined from routine UserVSCont, 4: user-defined from Simulink/L
1                      GenModel    - Generator model {1: simple, 2: Thevenin, 3: user-defined from routine UserGen} (switch) [used only when VSContrl=0]
95.756                 GenEff      - Generator efficiency [ignored by the Thevenin and user-defined generator models] (%)
True                   GenTiStr    - Method to start the generator {T: timed using TimGenOn, F: generator speed using SpdGenOn} (flag)
True                   GenTiStp    - Method to stop the generator {T: timed using TimGenOf, F: when generator power = 0} (flag)
9999.9                 SpdGenOn    - Generator speed to turn on the generator for a startup (HSS speed) (rpm) [used only when GenTiStr=False]
0.0                    TimGenOn    - Time to turn on the generator for a startup (s) [used only when GenTiStr=True]
9999.9                 TimGenOf    - Time to turn off the generator (s) [used only when GenTiStp=True]
---------------------- SIMPLE VARIABLE-SPEED TORQUE CONTROL --------------------
9999.9                 VS_RtGnSp   - Rated generator speed for simple variable-speed generator control (HSS side) (rpm) [used only when VSContrl=1]
9999.9                 VS_RtTq     - Rated generator torque/constant generator torque in Region 3 for simple variable-speed generator control (HSS side) (N-m) [u
9999.9                 VS_Rgn2K    - Generator torque constant in Region 2 for simple variable-speed generator control (HSS side) (N-m/rpm^2) [used only when VSC
9999.9                 VS_SlPc     - Rated generator slip percentage in Region 2 1/2 for simple variable-speed generator control (%) [used only when VSContrl=1]
---------------------- SIMPLE INDUCTION GENERATOR ------------------------------
9999.9                 SIG_SlPc    - Rated generator slip percentage (%) [used only when VSContrl=0 and GenModel=1]
9999.9                 SIG_SySp    - Synchronous (zero-torque) generator speed (rpm) [used only when VSContrl=0 and GenModel=1]
9999.9                 SIG_RtTq    - Rated torque (N-m) [used only when VSContrl=0 and GenModel=1]
9999.9                 SIG_PORt    - Pull-out ratio (Tpullout/Trated) (-) [used only when VSContrl=0 and GenModel=1]
---------------------- THEVENIN-EQUIVALENT INDUCTION GENERATOR -----------------
9999.9                 TEC_Freq    - Line frequency [50 or 60] (Hz) [used only when VSContrl=0 and GenModel=2]
100                    TEC_NPol    - Number of poles [even integer > 0] (-) [used only when VSContrl=0 and GenModel=2]
9999.9                 TEC_SRes    - Stator resistance (ohms) [used only when VSContrl=0 and GenModel=2]
9999.9                 TEC_RRes    - Rotor resistance (ohms) [used only when VSContrl=0 and GenModel=2]
9999.9                 TEC_VLL     - Line-to-line RMS voltage (volts) [used only when VSContrl=0 and GenModel=2]
9999.9                 TEC_SLR     - Stator leakage reactance (ohms) [used only when VSContrl=0 and GenModel=2]
9999.9                 TEC_RLR     - Rotor leakage reactance (ohms) [used only when VSContrl=0 and GenModel=2]
9999.9                 TEC_MR      - Magnetizing reactance (ohms) [used only when VSContrl=0 and GenModel=2]
---------------------- HIGH-SPEED SHAFT BRAKE ----------------------------------
0                      HSSBrMode   - HSS brake model {0: none, 1: simple, 3: user-defined from routine UserHSSBr, 4: user-defined from Simulink/Labview, 5: user-
9999.9                 THSSBrDp    - Time to initiate deployment of the HSS brake (s)
9999.9                 HSSBrDT     - Time for HSS-brake to reach full deployment once initiated (sec) [used only when HSSBrMode=1]
9999.9                 HSSBrTqF    - Fully deployed HSS-brake torque (N-m)
---------------------- NACELLE-YAW CONTROL --------------------------------------
0                      YCMode      - Yaw control mode {0: none, 3: user-defined from routine UserYawCont, 4: user-defined from Simulink/Labview, 5: user-defined
9999.9                 TYCOn       - Time to enable active yaw control (s) [unused when YCMode=0]
0.0                    YawNeut     - Neutral yaw position--yaw spring force is zero at this yaw (degrees)
8458453511.0           YawSpr      - Nacelle-yaw spring constant (N-m/rad)
28561762.0              YawDamp     - Nacelle-yaw damping constant (N-m/(rad/s))
9999.9                 TYawManS    - Time to start override yaw maneuver and end standard yaw control (s)
0.25                   YawManRat   - Yaw maneuver rate (in absolute value) (deg/s)
0.0                    NacYawF     - Final yaw angle for override yaw maneuvers (degrees)
---------------------- Aerodynamic Flow Control --------------------------------
0                      AfCmode     - Airfoil control mode {0- none, 1- cosine wave cycle, 4- user-defined from Simulink/Labview, 5- user-defined from Bladed-styl
0.0                    AfC_Mean    - Mean level for sinusoidal cycling or steady value (-) [used only with AfCmode==1]
0.0                    AfC_Amp     - Amplitude for for cosine cycling of flap signal (AfC = AfC_Amp*cos(Azimuth+phase)+AfC_mean) (-) [used only with AfCmode==1]
0.0                    AfC_phase   - Phase relative to the blade azimuth (0 is vertical) for for cosine cycling of flap signal (deg) [used only with AfCmode==1]
---------------------- STRUCTURAL CONTROL --------------------------------------
0                      NumBStC     - Number of blade structural controllers (integer)
"unused"               BStCfiles   - Name of the file for blade tuned mass damper (quoted string) [unused when CompNTMD is false]
0                      NumNStC     - Number of nacelle structural controllers (integer)
"unused"               NStCfiles   - Name of the file for nacelle tuned mass damper (quoted string) [unused when CompNTMD is false]
0                      NumTStC     - Number of tower structural controllers (integer)
"unused"               TStCfiles   - Name of the file for tower tuned mass damper (quoted string) [unused when CompNTMD is false]
0                      NumSStC     - Number of sbustructure structural controllers (integer)
"unused"               SStCfiles   - Name of the file for sbustructure tuned mass damper (quoted string) [unused when CompNTMD is false]
---------------------- CABLE CONTROL -------------------------------------------
0                      CCmode      - Cable control mode {0- none, 4- user-defined from Simulink/Labview, 5- user-defineAfC_phased from Bladed-style DLL}
---------------------- BLADED INTERFACE ----------------------------------------- [used only with Bladed Interface]
"discon_FFIPC_Final.dll" DLL_FileName - Name/location of the dynamic library {.dll [Windows] or .so [Linux]} in the Bladed-DLL format (-) [used only with Bladed Interface]
"IEA-15-240-RWT-UMaineSemi_DISCON_LAC_Final.IN"   DLL_InFile  - Name of input file sent to the DLL (-) [used only with Bladed Interface]
"DISCON"               DLL_ProcName - Name of procedure in DLL to be called (-) [case sensitive; used only with DLL Interface]
"default"              DLL_DT      - Communication interval for dynamic library (s) (or "default") [used only with Bladed Interface]
False                  DLL_Ramp    - Whether a linear ramp should be used between DLL_DT time steps [introduces time shift when true] (flag) [used only with Bladed Interface]
9999.9                 BPCutoff    - Cuttoff frequency for low-pass filter on blade pitch from DLL (Hz) [used only with Bladed Interface]
0.0                    NacYaw_North - Reference yaw angle of the nacelle when the upwind end points due North (deg) [used only with Bladed Interface]
1.0                    Ptch_Cntrl  - Record 28: Use individual pitch control {0: collective pitch; 1: individual pitch control} (switch) [used only with Bladed Interface]
0.0                    Ptch_SetPnt - Record  5: Below-rated pitch angle set-point (deg) [used only with Bladed Interface]
0.0                    Ptch_Min    - Record  6: Minimum pitch angle (deg) [used only with Bladed Interface]
0.0                    Ptch_Max    - Record  7: Maximum pitch angle (deg) [used only with Bladed Interface]
0.0                    PtchRate_Min - Record  8: Minimum pitch rate (most negative value allowed) (deg/s) [used only with Bladed Interface]
0.0                    PtchRate_Max - Record  9: Maximum pitch rate  (deg/s) [used only with Bladed Interface]
0.0                    Gain_OM     - Record 16: Optimal mode gain (Nm/(rad/s)^2) [used only with Bladed Interface]
0.0                    GenSpd_MinOM - Record 17: Minimum generator speed (rpm) [used only with Bladed Interface]
0.0                    GenSpd_MaxOM - Record 18: Optimal mode maximum speed (rpm) [used only with Bladed Interface]
0.0                    GenSpd_Dem  - Record 19: Demanded generator speed above rated (rpm) [used only with Bladed Interface]
0.0                    GenTrq_Dem  - Record 22: Demanded generator torque above rated (Nm) [used only with Bladed Interface]
0.0                    GenPwr_Dem  - Record 13: Demanded power (W) [used only with Bladed Interface]
---------------------- BLADED INTERFACE TORQUE-SPEED LOOK-UP TABLE -------------
0                      DLL_NumTrq  - Record 26: No. of points in torque-speed look-up table {0 = none and use the optimal mode parameters; nonzero = ignore the optimal mode P
GenSpd_TLU              GenTrq_TLU
(rpm)                  (Nm)
---------------------- OUTPUT --------------------------------------------------
False                   SumPrint    - Print summary data to <RootName>.sum (flag) (currently unused)
1                      OutFile     - Switch to determine where output will be placed: {1: in module output file only; 2: in glue code output file only; 3: both} (currently un
True                   TabDelim    - Use tab delimiters in text tabular output file? (flag) (currently unused)
"ES10.3E2"             OutFmt      - Format used for text tabular output (except time).  Resulting field should be 10 characters. (quoted string) (currently unused)
30.0                   TStart      - Time to begin tabular output (s) (currently unused)
OutList      - The next line(s) contains a list of output parameters.
"GenPwr"
"GenTq"
END of input file (the word "END" must appear in the first 3 columns of this last OutList line)
---------------------------------------------------------------------------------------
```

Figure D.7: Example of a ServoDyn input file used when performing the OpenFAST simulations.

```
------- AERODYN v15.03.* INPUT FILE ------------------------------------------------
IEA 15 MW Offshore Reference Turbine
======  General Options  ============================================================================
False                   Echo        - Echo the input to "<rootname>.AD.ech"?  (flag)
default                 DTAero      - Time interval for aerodynamic calculations {or "default"} (s)
1                       WakeMod     - Type of wake/induction model (switch) {0=none, 1=BEMT, 2=DBEMT, 3=OLAF} [WakeMod cannot be 2 or 3 when linearizing]
2                       AFAeroMod   - Type of blade airfoil aerodynamics model (switch) {1=steady model, 2=Beddoes-Leishman unsteady model} [AFAeroMod must
1                       TwrPotent   - Type tower influence on wind based on potential flow around the tower (switch) {0=none, 1=baseline potential flow, 2=p
1                       TwrShadow   - Calculate tower influence on wind based on downstream tower shadow (switch) {0=none, 1=Powles model, 2=Eames model}
True                    TwrAero     - Calculate tower aerodynamic loads? (flag)
False                   FrozenWake  - Assume frozen wake during linearization? (flag) [used only when WakeMod=1 and when linearizing]
False                   CavitCheck  - Perform cavitation check? (flag) [AFAeroMod must be 1 when CavitCheck=true]
False                   Buoyancy    - Include buoyancy effects? (flag)
False                   CompAA      - Flag to compute AeroAcoustics calculation [only used when WakeMod=1 or 2]
AeroAcousticsInput.dat  AA_InputFile - AeroAcoustics input file [used only when CompAA=true]
======  Environmental Conditions  ===================================================================
"default"               AirDens     - Air density (kg/m^3)
"default"               KinVisc     - Kinematic air viscosity (m^2/s)
"default"               SpdSound    - Speed of sound (m/s)
"default"               Patm        - Atmospheric pressure (Pa) [used only when CavitCheck=True]
"default"               Pvap        - Vapour pressure of fluid (Pa) [used only when CavitCheck=True]
======  Blade-Element/Momentum Theory Options  ========================================== [used only when WakeMod=1]
2                       SkewMod     - Type of skewed-wake correction model (switch) {1=uncoupled, 2=Pitt/Peters, 3=coupled} [unused when WakeMod=0 or 3]
default                 SkewModFactor - Constant used in Pitt/Peters skewed wake model {or "default" is 15/32*pi} (-) [used only when SkewMod=2; unused when
True                    TipLoss     - Use the Prandtl tip-loss model? (flag) [unused when WakeMod=0 or 3]
True                    HubLoss     - Use the Prandtl hub-loss model? (flag) [unused when WakeMod=0 or 3]
True                    TanInd      - Include tangential induction in BEMT calculations? (flag) [unused when WakeMod=0 or 3]
True                    AIDrag      - Include the drag term in the axial-induction calculation? (flag) [unused when WakeMod=0 or 3]
True                    TIDrag      - Include the drag term in the tangential-induction calculation? (flag) [unused when WakeMod=0,3 or TanInd=FALSE]
default                 IndToler    - Convergence tolerance for BEMT nonlinear solve residual equation {or "default"} (-) [unused when WakeMod=0 or 3]
500                     MaxIter     - Maximum number of iteration steps (-) [unused when WakeMod=0]
======  Dynamic Blade-Element/Momentum Theory Options  ================================================ [used only when WakeMod=1]
2                       DBEMT_Mod   - Type of dynamic BEMT (DBEMT) model {1=constant tau1, 2=time-dependent tau1} (-) [used only when WakeMod=2]
29.03                   tau1_const  - Time constant for DBEMT (s) [used only when WakeMod=2 and DBEMT_Mod=1]
======  OLAF -- cOnvecting LAgrangian Filaments (Free Vortex Wake) Theory Options  ================== [used only when WakeMod=3]
IEA-15-240-RWT_OLAF.dat OLAFInputFileName - Input file for OLAF [used only when WakeMod=3]
======  Beddoes-Leishman Unsteady Airfoil Aerodynamics Options  ==================================== [used only when AFAeroMod=2]
3                       UAMod       - Unsteady Aero Model Switch (switch) {1=Baseline model (Original), 2=Gonzalez's variant (changes in Cn,Cc,Cm), 3=Minnem
True                    FLookup     - Flag to indicate whether a lookup for f' will be calculated (TRUE) or whether best-fit exponential equations will be u
0                       UAStartRad  - Starting radius for dynamic stall (fraction of rotor radius) [used only when AFAeroMod=2]
1                       UAEndRad    - Ending radius for dynamic stall (fraction of rotor radius) [used only when AFAeroMod=2]
======  Airfoil Information ==========================================================================
1                       AFTabMod    - Interpolation method for multiple airfoil tables {1=1D interpolation on AoA (first table only); 2=2D interpolation on
1                       InCol_Alfa  - The column in the airfoil tables that contains the angle of attack (-)
2                       InCol_Cl    - The column in the airfoil tables that contains the lift coefficient (-)
3                       InCol_Cd    - The column in the airfoil tables that contains the drag coefficient (-)
4                       InCol_Cm    - The column in the airfoil tables that contains the pitching-moment coefficient; use zero if there is no Cm column (-)
0                       InCol_Cpmin - The column in the airfoil tables that contains the Cpmin coefficient; use zero if there is no Cpmin column (-)
50                      NumAFfiles  - Number of airfoil files used (-)
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_00.dat"    AFNames             - Airfoil file names (NumAFfiles lines) (quoted strings)
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_01.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_02.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_03.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_04.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_05.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_06.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_07.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_08.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_09.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_10.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_11.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_12.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_13.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_14.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_15.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_16.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_17.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_18.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_19.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_20.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_21.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_22.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_23.dat"
```

Figure D.8: Example of an AeroDyn input file used when performing the OpenFAST simulations (part 1).

```
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_24.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_25.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_26.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_27.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_28.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_29.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_30.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_31.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_32.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_33.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_34.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_35.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_36.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_37.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_38.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_39.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_40.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_41.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_42.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_43.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_44.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_45.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_46.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_47.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_48.dat"
"Airfoils/IEA-15-240-RWT_AeroDyn15_Polar_49.dat"
======  Rotor/Blade Properties  ====================================================================
True                    UseBlCm     - Include aerodynamic pitching moment in calculations?  (flag)
"IEA-15-240-RWT_AeroDyn15_blade.dat" ADBlFile(1) - Name of file containing distributed aerodynamic properties for Blade #1 (-)
"IEA-15-240-RWT_AeroDyn15_blade.dat" ADBlFile(2) - Name of file containing distributed aerodynamic properties for Blade #2 (-) [unused if NumBl < 2]
"IEA-15-240-RWT_AeroDyn15_blade.dat" ADBlFile(3) - Name of file containing distributed aerodynamic properties for Blade #3 (-) [unused if NumBl < 3]
====== Hub Properties ======================================================================================== [used only when Buoyancy=True]
7.0                 VolHub - Hub volume (m^3)
0.5                 HubCenBx - Hub center of buoyancy x direction offset (m)
====== Nacelle Properties ==================================================================================== [used only when Buoyancy=True]
32.0                VolNac - Nacelle volume (m^3)
0.4,0,0             NacCenB - Position of nacelle center of buoyancy from yaw bearing in nacelle coordinates (m)
====== Tail fin Aerodynamics ====================================================================
False               TFinAero - Calculate tail fin aerodynamics model (flag)
"AD_Fin.dat"        TFinFile - Input file for tail fin aerodynamics [used only when TFinAero=True]
======  Tower Influence and Aerodynamics ========================================================== [used only when TwrPotent/=0, TwrShadow/=0, or
11                      NumTwrNds   - Number of tower nodes used in the analysis  (-) [used only when TwrPotent/=0, TwrShadow/=0, or TwrAero=True]
TwrElev         TwrDiam         TwrCd           TwrTI               TwrCb         ! (used only with TwrShadow=2)
(m)               (m)             (-)               (-)             (-)
  15.000            10.00           0.5               0.1         0.0
  28.000            10.000          0.5               0.1         0.0
  41.000            9.926           0.5               0.1         0.0
  54.000            9.443           0.5               0.1         0.0
  67.000            8.833           0.5               0.1         0.0
  80.000            8.151           0.5               0.1         0.0
  93.000            7.39            0.5               0.1         0.0
 106.000            6.909           0.5               0.1         0.0
 119.000            6.748           0.5               0.1         0.0
 132.000            6.572           0.5               0.1         0.0
 144.386            6.5             0.5               0.1         0.0
======  Outputs  ===========================================================================================
False                   SumPrint    - Generate a summary file listing input options and interpolated properties to "<rootname>.AD.sum"?  (flag)
9                       NBlOuts     - Number of blade node outputs [0 - 9] (-)
6, 11, 16, 21, 25, 30, 35, 40, 45 BlOutNd     - Blade nodes whose values will be output  (-)
0                       NTwOuts     - Number of tower node outputs [0 - 9]  (-)
0                       TwOutNd     - Tower nodes whose values will be output  (-)
                    OutList             - The next line(s) contains a list of output parameters.  See OutListParameters.xlsx for a listing of availabl
"RtFldFxh"
"RtFldFyh"
"RtFldFzh"
"RtFldMxh"
"RtFldMyh"
"RtFldMzh"
"RtArea"
"RtSpeed"
"RtTSR"
END of input file (the word "END" must appear in the first 3 columns of this last OutList line)
====== Outputs for all blade stations (same ending as above for B1N1.... ========================== [optional section]
   1                BldNd_BladesOut     - Number of blades to output all node information at.  Up to number of blades on turbine. (-)
   "All"            BldNd_BlOutNd       - Future feature will allow selecting a portion of the nodes to output.  Not implemented yet. (-)
                    OutList             - The next line(s) contains a list of output parameters.  See OutListParameters.xlsx for a listing of available
END (the word "END" must appear in the first 3 columns of this last OutList line in the optional nodal output section)
```

Figure D.9: Example of an AeroDyn input file used when performing the OpenFAST simulations (part 2).

```
------- HydroDyn v2.03.* Input File ------------------------------------------
IEA 15 MW offshore reference model on UMaine VolturnUS-S semi-submersible floating platform
False          Echo            - Echo the input file data (flag)
---------------------- ENVIRONMENTAL CONDITIONS -------------------------------
        "default"     WtrDens        - Water density (kg/m^3)
        "default"     WtrDpth        - Water depth (meters)
        "default"     MSL2SWL        - Offset between still-water level and mean sea level (meters) [positive upward; unused when WaveMod = 6; must be zero if PotMod=1 or 2]
---------------------- WAVES --------------------------------------------------
           2     WaveMod     - Incident wave kinematics model {0: none=still water, 1: regular (periodic), 1P#: regular with user-specified phase, 2: JONSWAP/Pierson-Moskowit
           0     WaveStMod   - Model for stretching incident wave kinematics to instantaneous free surface {0: none=no stretching, 1: vertical stretching, 2: extrapolatio
        3800.00  WaveTMax    - Analysis time for incident wave calculations (sec) [unused when WaveMod=0; determines WaveDOmega=2Pi/WaveTMax in the IFFT]
           0.25  WaveDT       - Time step for incident wave calculations     (sec) [unused when WaveMod=0; 0.1<=WaveDT<=1.0 recommended; determines WaveOmegaMax=Pi/WaveDT
         2.0120  WaveHs       - Significant wave height of incident waves (meters) [used only when WaveMod=1 or 2]
         7.4510  WaveTp       - Peak spectral period of incident waves (sec) [used only when WaveMod=1 or 2]
        DEFAULT  WavePkShp    - Peak-shape parameter of incident wave spectrum (-) or DEFAULT (string) [used only when WaveMod=2; use 1.0 for Pierson-Moskowitz]
       0.111527  WvLowCOff    - Low  cut-off frequency or lower frequency limit of the wave spectrum beyond which the wave spectrum is zeroed (rad/s) [unused when WaveMod=
           3.2   WvHiCOff     - High cut-off frequency or upper frequency limit of the wave spectrum beyond which the wave spectrum is zeroed (rad/s) [unused when WaveMod=
           0.0   WaveDir      - Incident wave propagation heading direction                 (degrees) [unused when WaveMod=0 or 6]
           0     WaveDirMod   - Directional spreading function {0: none, 1: COS2S}         (-)       [only used when WaveMod=2,3, or 4]
           1     WaveDirSpread  - Wave direction spreading coefficient ( > 0 )             (-)       [only used when WaveMod=2,3, or 4 and WaveDirMod=1]
           1     WaveNDir     - Number of wave directions                                  (-)       [only used when WaveMod=2,3, or 4 and WaveDirMod=1; odd numbe
          90     WaveDirRange   - Range of wave directions (full range: WaveDir +/- 1/2*WaveDirRange) (degrees) [only used when WaveMod=2,3,or 4 and WaveDirMod=1]
  -561580799     WaveSeed(1)  - First  random seed of incident waves [-2147483648 to 2147483647]   (-)       [unused when WaveMod=0, 5, or 6]
      RANLUX     WaveSeed(2)  - Second random seed of incident waves [-2147483648 to 2147483647]   (-)       [unused when WaveMod=0, 5, or 6]
        TRUE     WaveMultiDir - Flag for normally distributed amplitudes                   (flag)    [only used when WaveMod=2, 3, or 4]
       "none"    WvKinFile    - Root name of externally generad wave data file(s)       (quoted string)   [used only when WaveMod=5 or 6]
           1     NWaveElev    - Number of points where the incident wave elevations can be computed (-)    [maximum of 9 output locations]
           0     WaveElevxi   - List of xi-coordinates for points where the incident wave elevations can be output (meters) [NWaveElev points, separated by commas or white
           0     WaveElevyi   - List of yi-coordinates for points where the incident wave elevations can be output (meters) [NWaveElev points, separated by commas or white
---------------------- 2ND-ORDER WAVES ----------------------------------------- [unused with WaveMod=0 or 6]
        True     WvDiffQTF    - Full difference-frequency 2nd-order wave kinematics (flag)
        True     WvSumQTF     - Full summation-frequency  2nd-order wave kinematics (flag)
           0     WvLowCOffD   - Low  frequency cutoff used in the difference-frequencies (rad/s) [Only used with a difference-frequency method]
       0.737863  WvHiCOffD    - High frequency cutoff used in the difference-frequencies (rad/s) [Only used with a difference-frequency method]
       0.314159  WvLowCOffS   - Low  frequency cutoff used in the summation-frequencies  (rad/s) [Only used with a summation-frequency  method]
       3.2       WvHiCOffS    - High frequency cutoff used in the summation-frequencies  (rad/s) [Only used with a summation-frequency  method]
---------------------- CURRENT ------------------------------------------------ [unused with WaveMod=6]
           0     CurrMod      - Current profile model {0: none=no current, 1: standard, 2: user-defined from routine UserCurrent} (switch)
           0     CurrSSV0     - Sub-surface current velocity at still water level  (m/s) [used only when CurrMod=1]
     "DEFAULT"   CurrSSDir    - Sub-surface current heading direction (degrees) or DEFAULT (string) [used only when CurrMod=1]
          20     CurrNSRef    - Near-surface current reference depth          (meters) [used only when CurrMod=1]
           0     CurrNSV0     - Near-surface current velocity at still water level (m/s) [used only when CurrMod=1]
           0     CurrNSDir    - Near-surface current heading direction        (degrees) [used only when CurrMod=1]
           0     CurrDIV      - Depth-independent current velocity               (m/s) [used only when CurrMod=1]
           0     CurrDIDir    - Depth-independent current heading direction   (degrees) [used only when CurrMod=1]
---------------------- FLOATING PLATFORM --------------------------------------- [unused with WaveMod=6]
           1     PotMod       - Potential-flow model {0: none=no potential flow, 1: frequency-to-time-domain transforms based on WAMIT output, 2: fluid-impulse theory (FIT)} (
           1     ExctnMod     - Wave Excitation model {0: None, 1: DFT, 2: state-space} (switch) [only used when PotMod=1; STATE-SPACE REQUIRES *.ssexctn INPUT FILE]
           1     RdtnMod      - Radiation memory-effect model {0: no memory-effect calculation, 1: convolution, 2: state-space} (switch) [only used when PotMod=1; STATE-SPACE
        60.0     RdtnTMax     - Analysis time for wave radiation kernel calculations (sec) [only used when PotMod=1; determines RdtnDOmega=Pi/RdtnTMax in the cosine transform;
     "DEFAULT"   RdtnDT       - Time step for wave radiation kernel calculations (sec) [only used when PotMod=1; DT<=RdtnDT<=0.1 recommended; determines RdtnOmegaMax=Pi/RdtnDT
           1     NBody        - Number of WAMIT bodies to be used (-) [>=1; only used when PotMod=1. If NBodyMod=1, the WAMIT data contains a vector of size 6*NBody x 1 and m
           1     NBodyMod     - Body coupling model {1: include coupling terms between each body and NBody in HydroDyn equals NBODY in WAMIT, 2: neglect coupling terms between
"HydroData/IEA-15-240-RWT-UMaineSemi"  PotFile      - Root name of potential-flow model data; WAMIT output files containing the linear, nondimensionalized, hydrostat
           1     WAMITULEN    - Characteristic body length scale used to redimensionalize WAMIT output (meters) [only used when PotMod=1]
           0.0   PtfmRefxt     - The xt offset of the body reference point(s) from (0,0,0) (meters) [1 to NBody] [only used when PotMod=1]
           0.0   PtfmRefyt     - The yt offset of the body reference point(s) from (0,0,0) (meters) [1 to NBody] [only used when PotMod=1]
           0.0   PtfmRefzt     - The zt offset of the body reference point(s) from (0,0,0) (meters) [1 to NBody] [only used when PotMod=1. If NBodyMod=2,PtfmRefzt=0.0]
           0.0   PtfmRefztRot   - The rotation about zt of the body reference frame(s) from xt/yt (degrees) [1 to NBody] [only used when PotMod=1]
    20206.34889  PtfmVol0      - Displaced volume of water when the platform is in its undisplaced position (m^3) [only used when PotMod=1; USE THE SAME VALUE COMPUTED BY WA
           0     PtfmCOBxt    - The xt offset of the center of buoyancy (COB) from the platform reference point (meters)  [only used when PotMod=1]
           0     PtfmCOByt    - The yt offset of the center of buoyancy (COB) from the platform reference point (meters)  [only used when PotMod=1]
---------------------- 2ND-ORDER FLOATING PLATFORM FORCES ---------------------- [unused with WaveMod=0 or 6, or PotMod=0 or 2]
           0     MnDrift      - Mean-drift 2nd-order forces computed                                                {0: None; [7, 8, 9, 10, 11, or 12]: WAMIT file to use} [Only one
           0     NewmanApp    - Mean- and slow-drift 2nd-order forces computed with Newman's approximation {0: None; [7, 8, 9, 10, 11, or 12]: WAMIT file to use} [Only one
                    12     DiffQTF       - Full difference-frequency 2nd-order forces computed with full QTF         {0: None; [10, 11, or 12]: WAMIT file to use}
           0     SumQTF       - Full summation -frequency 2nd-order forces computed with full QTF         {0: None; [10, 11, or 12]: WAMIT file to use}
---------------------- PLATFORM ADDITIONAL STIFFNESS AND DAMPING  --------------
           0.0   AddF0    - Additional preload (N, N-m) [If NBodyMod=1, one size 6*NBody x 1 vector; if NBodyMod>1, NBody size 6 x 1 vectors]
           0.0
           0.0
           0.0
           0.0
           0.0
           0.0
           0.0
           0.0
             0            0            0            0            0            0   AddCLin  - Additional linear stiffness (N/m, N/rad, N-m/m, N-m/rad)
             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   AddBLin  - Additional linear damping(N/(m/s), N/(rad/s), N-m/(m/s), N-m/(rad/s))
             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
9.23E+05     0.00E+00     0.00E+00     0.00E+00    -8.92E+06     0.00E+00 AddBQuad - Additional quadratic drag(N/(m/s)^2, N/(rad/s)^2, N-m(m/s)^2, N-m/(r
0.00E+00     9.23E+05     0.00E+00     8.92E+06     0.00E+00     0.00E+00
0.00E+00     0.00E+00     2.30E+06     0.00E+00     0.00E+00     0.00E+00
0.00E+00     8.92E+06     0.00E+00     1.68E+10     0.00E+00     0.00E+00
-8.92E+06    0.00E+00     0.00E+00     0.00E+00     1.68E+10     0.00E+00
0.00E+00     0.00E+00     0.00E+00     0.00E+00     0.00E+00     4.80E+10
---------------------- AXIAL COEFFICIENTS -------------------------------------
1                   NAxCoef     - Number of axial coefficients (-)
 AxCoefID      AxCd        AxCa        AxCp
    (-)         (-)         (-)         (-)
     1          0.0         0.0         1.0
---------------------- MEMBER JOINTS ------------------------------------------
0                   NJoints     - Number of joints (-)   [must be exactly 0 or at least 2]
  JointID     Jointxi     Jointyi     Jointzi    JointAxID  JointOvrlp    [JointOvrlp= 0: do nothing at joint, 1: eliminate overlaps by calculating super member]
    (-)         (m)         (m)         (m)         (-)      (switch)
---------------------- MEMBER CROSS-SECTION PROPERTIES ------------------------
0           NPropSets   - Number of member property sets (-)
 PropSetID     PropD     PropThck
    (-)         (m)         (m)
---------------------- SIMPLE HYDRODYNAMIC COEFFICIENTS (model 1) --------------
  SimplCd    SimplCdMG    SimplCa    SimplCaMG    SimplCp    SimplCpMG   SimplAxCd  SimplAxCdMG  SimplAxCa  SimplAxCaMG  SimplAxCp  SimplAxCpMG
    (-)         (-)         (-)         (-)         (-)         (-)         (-)         (-)         (-)         (-)         (-)         (-)
    0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0
---------------------- DEPTH-BASED HYDRODYNAMIC COEFFICIENTS (model 2) ---------
0           NCoefDpth   - Number of depth-dependent coefficients (-)
   Dpth       DpthCd     DpthCdMG     DpthCa     DpthCaMG     DpthCp     DpthCpMG    DpthAxCd   DpthAxCdMG   DpthAxCa   DpthAxCaMG   DpthAxCp   DpthAxCpMG
    (m)         (-)         (-)         (-)         (-)         (-)         (-)         (-)         (-)         (-)         (-)         (-)         (-)
---------------------- MEMBER-BASED HYDRODYNAMIC COEFFICIENTS (model 3) --------
0           NCoefMembers - Number of member-based coefficients (-)
MemberID_HydC  MemberCd1    MemberCd2  MemberCdMG1 MemberCdMG2  MemberCa1    MemberCa2  MemberCaMG1 MemberCaMG2  MemberCp1    MemberCp2  MemberCpMG1 MemberCpMG2 MemberAxCd
    (-)         (-)         (-)         (-)         (-)         (-)         (-)         (-)         (-)         (-)         (-)         (-)         (-)         (-)
---------------------- MEMBERS ------------------------------------------------
0           NMembers    - Number of members (-)
 MemberID     MJointID1   MJointID2  MPropSetID1 MPropSetID2  MDivSize    MCoefMod     PropPot     [MCoefMod=1: use simple coeff table, 2: use depth-based coeff table, 3
    (-)         (-)         (-)         (-)         (-)         (m)      (switch)     (flag)
---------------------- FILLED MEMBERS -----------------------------------------
             0   NFillGroups     - Number of filled member groups (-) [If FillDens = DEFAULT, then FillDens = WtrDens; FillFSLoc is related to MSL2SWL]
FillNumM FillMList            FillFSLoc     FillDens
(-)      (-)                  (m)           (kg/m^3)
---------------------- MARINE GROWTH ------------------------------------------
             0   NMGDepths       - Number of marine-growth depths specified (-)
MGDpth      MGThck       MGDens
(m)         (m)          (kg/m^3)
---------------------- MEMBER OUTPUT LIST -------------------------------------
             0   NMOutputs       - Number of member outputs (-) [must be < 10]
MemberID   NOutLoc    NodeLocs [NOutLoc < 10; node locations are normalized distance from the start of the member, and must be >=0 and <= 1] [unused if NMOutputs=0]
  (-)        (-)         (-)
---------------------- JOINT OUTPUT LIST --------------------------------------
             0   NJOutputs       - Number of joint outputs [Must be < 10]
             0   JOutLst         - List of JointIDs which are to be output (-)[unused if NJOutputs=0]
---------------------- OUTPUT -------------------------------------------------
TRUE            HDSum       - Output a summary file [flag]
False           OutAll      - Output all user-specified member and joint loads (only at each member end, not interior locations) [flag]
             2  OutSwtch     - Output requested channels to: [1=Hydrodyn.out, 2=GlueCode.out, 3=both files]
"ES11.4e2"      OutFmt      - Output format for numerical results (quoted string) [not checked for validity!]
"A11"           OutSFmt     - Output format for header strings (quoted string) [not checked for validity!]
---------------------- OUTPUT CHANNELS ----------------------------------------
"Wave1Elev"             - Wave elevation at the platform reference point (  0,  0)
END of output channels and end of file. (the word "END" must appear in the first 3 columns of this line)
```

Figure D.10: Example of a HydroDyn input file used when performing the OpenFAST simulations.

D.2 MLife

```
-----  MLife version 1.0 Input File  ------------------------------------------
FB
-----  Job Options  -----------------------------------------------------------
true              EchoInp           Echo input to <rootname>.echo as this file is being read.
true              StrNames          Use channel names following a "$" instead of numbers when specifying channels in this input file.
false             OutData           Output modified data array after scaling and calculated channels. (currently unavailable)
"%11.3g"          RealFmt           Format for outputting floating-point values.
"FB"              RootName          Root name for output files.
-----  Input-Data Layout  -----------------------------------------------------
0                 TitleLine         The row with the file title on it (zero if no title is available).
7                 NamesLine         The row with the channel names on it (zero if no names are available or are specified below).
0                 UnitsLine         The row with the channel units on it (zero if no units are available or are specified below).
9                 FirstDataLine     The first row of data.
0                NumChans         The number of channels in each input file.
ChanTitle    ChanUnits    Scale  Offset   PSF_Type  NumCols rows of data follow.  Title and units strings must be 10 characters or less.
-----  Calculated Channels  ---------------------------------------------------
66                NumCChan          The number calculated channels to generate.
1234567890        Seed              The integer seed for the random number generator (-2,147,483,648 to 2,147,483,647).
Col_Title   Units    Equation       Put each field in quotes.  Titles and units are limited to 10 characters.  NumCChan rows of data follow.
"RootMxb1"  "(kN-m)" "timeSeriesData(:,52)"
"RootMxb2"  "(kN-m)" "timeSeriesData(:,53)"
"RootMxb3"  "(kN-m)" "timeSeriesData(:,54)"
"RootMyb1"  "(kN-m)" "timeSeriesData(:,58)"
"RootMyb2"  "(kN-m)" "timeSeriesData(:,59)"
"RootMyb3"  "(kN-m)" "timeSeriesData(:,60)"
"RootMzb1"  "(kN-m)" "timeSeriesData(:,64)"
"RootMzb2"  "(kN-m)" "timeSeriesData(:,65)"
"RootMzb3"  "(kN-m)" "timeSeriesData(:,66)"
"RootMyc1"  "(kN-m)" "timeSeriesData(:,61)"
"RootMyc2"  "(kN-m)" "timeSeriesData(:,62)"
"RootMyc3"  "(kN-m)" "timeSeriesData(:,63)"
"RootMxc1"  "(kN-m)" "timeSeriesData(:,55)"
"RootMxc2"  "(kN-m)" "timeSeriesData(:,56)"
"RootMxc3"  "(kN-m)" "timeSeriesData(:,57)"
"LSShftMxa" "(kN-m)" "timeSeriesData(:,94)"
"LSSTipMya" "(kN-m)" "timeSeriesData(:,95)"
"LSSTipMza" "(kN-m)" "timeSeriesData(:,96)"
"LSSGagMxs" "(kN-m)" "timeSeriesData(:,97)"
"LSSGagMys" "(kN-m)" "timeSeriesData(:,98)"
"LSSGagMzs" "(kN-m)" "timeSeriesData(:,99)"
"YawBrMxn"  "(kN-m)" "timeSeriesData(:,100)"
"YawBrMyn"  "(kN-m)" "timeSeriesData(:,101)"
"YawBrMzn"  "(kN-m)" "timeSeriesData(:,102)"
"YawBrMxp"  "(kN-m)" "timeSeriesData(:,89)"
"YawBrMyp"  "(kN-m)" "timeSeriesData(:,90)"
"YawBrMzp"  "(kN-m)" "timeSeriesData(:,91)"
"TwrBsMxt"  "(kN-m)" "timeSeriesData(:,85)"
"TwrBsMyt"  "(kN-m)" "timeSeriesData(:,86)"
"TwrBsMzt"  "(kN-m)" "timeSeriesData(:,87)"
"TwHt1MLxt" "(kN-m)" "timeSeriesData(:,103)"
"TwHt1MLyt" "(kN-m)" "timeSeriesData(:,104)"
"TwHt1MLzt" "(kN-m)" "timeSeriesData(:,105)"
"Spn1MLxb1" "(kN-m)" "timeSeriesData(:,73)"
"Spn1MLyb1" "(kN-m)" "timeSeriesData(:,74)"
"Spn2MLxb1" "(kN-m)" "timeSeriesData(:,75)"
"Spn2MLyb1" "(kN-m)" "timeSeriesData(:,76)"
"Spn3MLxb1" "(kN-m)" "timeSeriesData(:,77)"
"Spn3MLyb1" "(kN-m)" "timeSeriesData(:,78)"
"Spn3MLzb1" "(kN-m)" "timeSeriesData(:,106)"
"Spn1MLxb2" "(kN-m)" "timeSeriesData(:,107)"
"Spn1MLyb2" "(kN-m)" "timeSeriesData(:,108)"
"Spn1MLzb2" "(kN-m)" "timeSeriesData(:,109)"
"Spn2MLxb2" "(kN-m)" "timeSeriesData(:,110)"
"Spn2MLyb2" "(kN-m)" "timeSeriesData(:,111)"
"Spn2MLzb2" "(kN-m)" "timeSeriesData(:,112)"
"Spn3MLxb2" "(kN-m)" "timeSeriesData(:,113)"
"Spn3MLyb2" "(kN-m)" "timeSeriesData(:,114)"
"Spn3MLzb2" "(kN-m)" "timeSeriesData(:,115)"
"Spn1MLxb3" "(kN-m)" "timeSeriesData(:,116)"
"Spn1MLyb3" "(kN-m)" "timeSeriesData(:,117)"
"Spn1MLzb3" "(kN-m)" "timeSeriesData(:,118)"
"Spn2MLxb3" "(kN-m)" "timeSeriesData(:,119)"
"Spn2MLyb3" "(kN-m)" "timeSeriesData(:,120)"
"Spn2MLzb3" "(kN-m)" "timeSeriesData(:,121)"
"Spn3MLxb3" "(kN-m)" "timeSeriesData(:,122)"
"Spn3MLyb3" "(kN-m)" "timeSeriesData(:,123)"
"Spn3MLzb3" "(kN-m)" "timeSeriesData(:,124)"
"RtFldFxh"  "N"      "timeSeriesData(:,125)"
"RtFldFyh"  "N"      "timeSeriesData(:,126)"
"RtFldFzh"  "N"      "timeSeriesData(:,127)"
"RtFldMxh"  "N"      "timeSeriesData(:,128)"
"RtFldMyh"  "N"      "timeSeriesData(:,129)"
"RtFldMzh"  "N"      "timeSeriesData(:,130)"
"FAIRTEN1"  "N"      "timeSeriesData(:,137)"
"ANCHTEN1"  "N"      "timeSeriesData(:,140)"
-----  Load Roses  ------------------------------------------------------------
0                 nLoadRoses        The number of load roses to generate.
Rose Name     Units    Channel1       Channel2     nSectors
-----  Time and Wind Speed  ---------------------------------------------------
1                 TimeChan          The channel containing time.
2                 WSChan            The primary wind-speed channel (used for mean wind speed and turbulence intensity, 0 for none).
-----  Statistics and Extreme Events  -----------------------------------------
true             DoStats           Generate statistics of all the channels.
```

Figure D.11: Example of an MLife input file used when undertaking the OpenFAST simulation post-processing (part 1).

```
False            WrStatsTxt        Write the stats to a text file?
true             WrStatsXLS        Write the stats to an Excel file?
0                NumSFChans        Number of channels that will have summary statistics generated for them.
52      53       54       55       56       57       58       59       60       61       62       63       64       65       66       73       74       75       76       77       78
-----  Distributions ---------------------------------------------------------------------
false            UserDistrib       User defined distribution?  true = load user-specified distribution, false = only use Weibull wind distribution
2.360338149         WeibullShape      Weibull shape factor. If WeibullShape=2, enter the mean wind speed for WeibullScale.
9.357494714         WeibullScale      Weibull scale factor. If WeibullShape<>2.  Otherwise, enter the mean wind speed.
3                WSin              Cut-in wind speed for the turbine.
25               WSout             Cut-out wind speed for the turbine.
31               WSmax             Maximum  wind speed value for the wind-speed bins.
2                WSMaxBinSize      Maximum width of a wind-speed bin.
0                nDistribVars      Number of independent variables in the user-specified distribution, ignored if UserDistrib = false
""               DistribName       Filename of the user-supplied distribution table, ignored if UserDistrib = false
-----  Fatigue  --------------------------------------------------------------------
66               nFatigueChannels  The number of rainflow channels.  Next six lines ignored if zero.
0.0              FiltRatio         The fraction of the maximum range of each channel used as a cutoff range for the racetrack filter.  Use zero for no filter.
7.884e+8         DesignLife        Number of seconds in the design lifetime (20 years = 630720000 seconds).
0.95             Availability      Fraction of the design life the turbine is operating when winds are between Vin and Vout
True             BinCycles         Bin the rainflow cycles?
0.5              UCMult            Multiplier for binning unclosed cycles.  (0 discards, 1 counts as a full cycle)
True             DoShortTerm       Compute simple (unweighted) damage-equivalent loads and damage rates.
True             DoLife            Do lifetime-related calculations?
True             DoAggregate       Compute a DELs and a damage result based on an aggregate of all the input files (does not use the wind spped distribution)
false            WrShortTermTxt    Write short-term results to plain-text files?
True             WrShortTermXLS    Write short-term resultsto an Excel workbook?
false            WrLifeTxt         Write lifetime results to plain-text files?
True             WrLifeXLS         Write lifetime results to an Excel workbook?
1                EquivalentFrequency The frequency of the damage equivalent load (Hz)
true             DEL_AsRange       true = report DELs as a range value,  false = report as a one-sided amplitude
3                DEL_Type          1 = fixed mean, 2 = zero mean, 3 = both
2                GoodmanFlag       0 = no Goodman correction, 1 = use Goodman correction, 2 = compute results with and without Goodman correction
Channel#     NSlopes  SNslopeLst     BinFlag  MaxBinWidth/Number  TypeLMF     LULt    BinWidth not used when BinCycles is false. nFatigueChannels rows of data follow.
$RootMxb1$       6       3 4 5 8 10 12          BW              1000    WM              153000
$RootMxb2$       6       3 4 5 8 10 12          BW              1000    WM              153000
$RootMxb3$       6       3 4 5 8 10 12          BW              1000    WM              153000
$RootMyb1$       6       3 4 5 8 10 12          BW              1000    WM              153000
$RootMyb2$       6       3 4 5 8 10 12          BW              1000    WM              153000
$RootMyb3$       6       3 4 5 8 10 12          BW              1000    WM              153000
$RootMzb1$       6       3 4 5 8 10 12          BW              1000    WM              153000
$RootMzb2$       6       3 4 5 8 10 12          BW              1000    WM              153000
$RootMzb3$       6       3 4 5 8 10 12          BW              1000    WM              153000
$RootMyc1$       6       3 4 5 8 10 12          BW              1000    WM              153000
$RootMyc2$       6       3 4 5 8 10 12          BW              1000    WM              153000
$RootMyc3$       6       3 4 5 8 10 12          BW              1000    WM              153000
$RootMxc1$       6       3 4 5 8 10 12          BW              1000    WM              153000
$RootMxc2$       6       3 4 5 8 10 12          BW              1000    WM              153000
$RootMxc3$       6       3 4 5 8 10 12          BW              1000    WM              153000
$LSShftMxa$      6       3 4 5 8 10 12          BW              1000    WM              108000
$LSSTipMya$      6       3 4 5 8 10 12          BW              10000   WM              1120000
$LSSTipMza$      6       3 4 5 8 10 12          BW              10000   WM              1120000
$LSSGagMxs$      6       3 4 5 8 10 12          BW              1000    WM              90000
$LSSGagMys$      6       3 4 5 8 10 12          BW              10000   WM              2400000
$LSSGagMzs$      6       3 4 5 8 10 12          BW              10000   WM              2400000
$YawBrMxn$       6       3 4 5 8 10 12          BW              1000    WM              120000
$YawBrMyn$       6       3 4 5 8 10 12          BW              10000   WM              1030000
$YawBrMzn$       6       3 4 5 8 10 12          BW              10000   WM              1030000
$YawBrMxp$       6       3 4 5 8 10 12          BW              1000    WM              120000
$YawBrMyp$       6       3 4 5 8 10 12          BW              10000   WM              1030000
$YawBrMzp$       6       3 4 5 8 10 12          BW              10000   WM              1030000
$TwrBsMxt$       6       3 4 5 8 10 12          BW              10000   WM              3000000
$TwrBsMyt$       6       3 4 5 8 10 12          BW              10000   WM              3000000
$TwrBsMzt$       6       3 4 5 8 10 12          BW              10000   WM              3000000
$TwHt1MLxt$      6       3 4 5 8 10 12          BW              10000   WM              300000
$TwHt1MLyt$      6       3 4 5 8 10 12          BW              10000   WM              300000
$TwHt1MLzt$      6       3 4 5 8 10 12          BW              10000   WM              300000
$Spn1MLxb1$      6       3 4 5 8 10 12          BW              1000    WM              130000
$Spn1MLyb1$      6       3 4 5 8 10 12          BW              1000    WM              130000
$Spn2MLxb1$      6       3 4 5 8 10 12          BW              1000    WM              130000
$Spn2MLyb1$      6       3 4 5 8 10 12          BW              1000    WM              130000
$Spn3MLxb1$      6       3 4 5 8 10 12          BW              1000    WM              130000
$Spn3MLyb1$      6       3 4 5 8 10 12          BW              1000    WM              130000
$Spn3MLzb1$      6       3 4 5 8 10 12          BW              1000    WM              130000
$Spn1MLxb2$      6       3 4 5 8 10 12          BW              1000    WM              130000
$Spn1MLyb2$      6       3 4 5 8 10 12          BW              1000    WM              130000
$Spn1MLzb2$      6       3 4 5 8 10 12          BW              1000    WM              130000
$Spn2MLxb2$      6       3 4 5 8 10 12          BW              1000    WM              130000
$Spn2MLyb2$      6       3 4 5 8 10 12          BW              1000    WM              130000
$Spn2MLzb2$      6       3 4 5 8 10 12          BW              1000    WM              130000
$Spn3MLxb2$      6       3 4 5 8 10 12          BW              1000    WM              130000
$Spn3MLyb2$      6       3 4 5 8 10 12          BW              1000    WM              130000
$Spn3MLzb2$      6       3 4 5 8 10 12          BW              1000    WM              130000
$Spn1MLxb3$      6       3 4 5 8 10 12          BW              1000    WM              130000
$Spn1MLyb3$      6       3 4 5 8 10 12          BW              1000    WM              130000
$Spn1MLzb3$      6       3 4 5 8 10 12          BW              1000    WM              130000
$Spn2MLxb3$      6       3 4 5 8 10 12          BW              1000    WM              130000
$Spn2MLyb3$      6       3 4 5 8 10 12          BW              1000    WM              130000
$Spn2MLzb3$      6       3 4 5 8 10 12          BW              1000    WM              130000
$Spn3MLxb3$      6       3 4 5 8 10 12          BW              1000    WM              130000
$Spn3MLyb3$      6       3 4 5 8 10 12          BW              1000    WM              129000
$Spn3MLzb3$      6       3 4 5 8 10 12          BW              1000    WM              129000
$RtFldFxh$       6       3 4 5 8 10 12          BW              100000  WM              10000000
$RtFldFyh$       6       3 4 5 8 10 12          BW              100000  WM              10000000
```

Figure D.12: Example of an MLife input file used when undertaking the OpenFAST simulation post-processing (part 2).

```
$RtFldFzh$      6       3 4 5 8 10 12           BW              100000  WM              10000000
$RtFldMxh$      6       3 4 5 8 10 12           BW              100000  WM              1500000000
$RtFldMyh$      6       3 4 5 8 10 12           BW              100000  WM              1500000000
$RtFldMzh$      6       3 4 5 8 10 12           BW              100000  WM              1500000000
$FAIRTEN1$      6       3 4 5 8 10 12           BW              100000  WM              10000000
$ANCHTEN1$  6   3 4 5 8 10 12       BW      100000  WM      10000000
0               nGroups             Number of fatigue groups
Group Name  NChannels      ChannelList
-----  Input Files  ------------------------------------------------------------
1               FileFormat          Format of input files.  1 = FAST ascii, 2 = FAST binary
48  1.1   1.3   1.5   1.7    (Weibull-Weighted Normal Operation: NumNormFiles, PSF1, PSF2, PSF3, PSF4)
"IEA-15-240-RWT-UMaineSemi_FB_3ms_S1_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_3ms_S2_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_3ms_S3_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_3ms_S4_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_5ms_S1_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_5ms_S2_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_5ms_S3_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_5ms_S4_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_7ms_S1_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_7ms_S2_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_7ms_S3_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_7ms_S4_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_9ms_S1_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_9ms_S2_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_9ms_S3_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_9ms_S4_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_11ms_S1_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_11ms_S2_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_11ms_S3_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_11ms_S4_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_13ms_S1_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_13ms_S2_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_13ms_S3_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_13ms_S4_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_15ms_S1_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_15ms_S2_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_15ms_S3_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_15ms_S4_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_17ms_S1_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_17ms_S2_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_17ms_S3_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_17ms_S4_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_19ms_S1_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_19ms_S2_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_19ms_S3_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_19ms_S4_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_21ms_S1_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_21ms_S2_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_21ms_S3_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_21ms_S4_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_23ms_S1_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_23ms_S2_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_23ms_S3_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_23ms_S4_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_25ms_S1_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_25ms_S2_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_25ms_S3_1hr.out"
"IEA-15-240-RWT-UMaineSemi_FB_25ms_S4_1hr.out"
0  1.1   1.3   1.5   1.7    (Weibull-Weighted Idling: NumIdleFiles, PSF1, PSF2, PSF3, PSF4)
0  1.2   1.3   1.4   1.6    (Discrete Events: NumDiscFiles, PSF1, PSF2, PSF3, PSF4)
==EOF==                         DO NOT REMOVE OR CHANGE.  MUST COME JUST AFTER LAST LINE OF VALID INPUT.
```

Figure D.13: Example of an MLife input file used when undertaking the OpenFAST simulation post-processing (part 3).

D.3 FAST.Farm

```
FAST.Farm v1.00.* INPUT FILE
Sample FAST.Farm input file
--- SIMULATION CONTROL ---
False           Echo            Echo input data to <RootName>.ech? (flag)
FATAL           AbortLevel      Error level when simulation should abort (string) {"WARNING", "SEVERE", "FATAL"}
995.0           TMax            Total run time (s) [>=0.0]
False           UseSC           Use a super controller? (flag)
2               Mod_AmbWind           Ambient wind model (-) (switch) {1: high-fidelity precursor in VTK format, 2: one InflowWind module, 3: multiple ins
2          Mod_WaveField         Wave field handling (-) (switch) {1: use individual HydroDyn inputs without adjustment, 2: adjust wave phases based on t
0          Mod_SharedMooring     Shared mooring system model (switch) {0: None, 3=MoorDyn}}
--- SUPER CONTROLLER --- [used only for UseSC=True]
"SC_DLL.dll"    SC_FileName     Name/location of the dynamic library {.dll [Windows] or .so [Linux]}, containing the Super Controller algorithms (quoted st
--- SHARED MOORING SYSTEM --- [used only for Mod_SharedMoor>0]
""           SharedMoorFile         Name of file containing shared mooring system input parameters (quoted string) [used only when Mod_SharedMooring > 0]
0.01            DT_Mooring          Time step for farm-level mooring coupling with each turbine (s) [used only when Mod_SharedMooring > 0]
--- AMBIENT WIND: PRECURSOR IN VTK FORMAT --- [used only for Mod_AmbWind=1]
2.0             DT_Low-VTK          Time step for low-resolution wind data input files; will be used as the, global FAST.Farm time step (s) [>0.0]
0.5             DT_High-VTK     Time step for high-resolution wind data input files (s) [>0.0]
"/AmbWind/steady"  WindFilePath Path name to wind data files from precursor (string)
False           ChkWndFiles     Check all the ambient wind files for data consistency (flag)
--- AMBIENT WIND: INFLOWWIND MODULE --- [used only for Mod_AmbWind=2 or 3]
1               DT_Low          Time step for low-resolution wind data interpolation; will be used as the global FAST.Farm time step (s) [>0.0]
0.1             DT_High         Time step for high-resolution wind data interpolation (s) [>0.0]
1200            NX_Low          Number of low-resolution spatial nodes in X direction for wind data interpolation (-) [>=2]
57              NY_Low          Number of low-resolution spatial nodes in Y direction for wind data interpolation (-) [>=2]
57              NZ_Low          Number of low-resolution spatial nodes in Z direction for wind data interpolation (-) [>=2]
-300.0          X0_Low          Origin of low-resolution spatial nodes in X direction for wind data interpolation (m)
-140.0          Y0_Low          Origin of low-resolution spatial nodes in Y direction for wind data interpolation (m)
10.0            Z0_Low          Origin of low-resolution spatial nodes in Z direction for wind data interpolation (m)
5.0             dX_Low          Spacing of low-resolution spatial nodes in X direction for wind data interpolation (m) [>0.0]
5.0             dY_Low          Spacing of low-resolution spatial nodes in Y direction for wind data interpolation (m) [>0.0]
5.0        dZ_Low               Spacing of low-resolution spatial nodes in Z direction for wind data interpolation (m) [>0.0]
50         NX_High              Number of high-resolution spatial nodes in X direction for wind data interpolation (-) [>=2]
35         NY_High              Number of high-resolution spatial nodes in Y direction for wind data interpolation (-) [>=2]
30         NZ_High     Number of high-resolution spatial nodes in Z direction for wind data interpolation (-) [>=2]
"IEA-15-240-RWT_InflowFile_FastFarm.dat"   InflowFile    Name of file containing InflowWind module input parameters (quoted string)
--- WIND TURBINES ---
3 NumTurbines Number of wind turbines (-) [>=1] [last 6 columns below used only for Mod_AmbWind=2 or 3]
WT_X      WT_Y    WT_Z            WT_FASTInFile                                  X0_High         Y0_High     Z0_High      dX_High      dY_High      dZ_High
(m)       (m)     (m)              (string)                                        (m)             (m)         (m)          (m)          (m)          (m)
 0.0      0.0     0.0       "Turbine1/IEA-15-240-RWT-UMaineSemi_T1.fst"       -300.0        -150.0       10.0         10.0         10.0         10.0
1920.0    0.0     0.0       "Turbine2/IEA-15-240-RWT-UMaineSemi_T2.fst"       1620.0        -150.0       10.0         10.0         10.0         10.0
3840.0    0.0     0.0       "Turbine3/IEA-15-240-RWT-UMaineSemi_T3.fst"       3540.0        -150.0       10.0         10.0         10.0         10.0
--- WAKE DYNAMICS ---
1          Mod_Wake       Switch between wake formulations {1:Polar, 2:Curl, 3:Cartesian} (-) (switch)
5.0        dr             Radial increment of radial finite-difference grid (m) [>0.0]
60         NumRadii       Number of radii in the radial finite-difference grid (-) [>=2]
200        NumPlanes      Number of wake planes (-) [>=2]
DEFAULT    f_c            Cutoff (corner) frequency of the low-pass time-filter for the wake advection, deflection, and meandering model (Hz) [>0.0] or DE
DEFAULT    C_HWkDfl_O     Calibrated parameter in the correction for wake deflection defining the horizontal offset at the rotor (m) or DEFAULT [=0.0]
DEFAULT    C_HWkDfl_OY    Calibrated parameter in the correction for wake deflection defining the horizontal offset at the rotor scaled with yaw error (m,
DEFAULT    C_HWkDfl_x     Calibrated parameter in the correction for wake deflection defining the horizontal offset scaled with downstream distance (-) or
DEFAULT    C_HWkDfl_xY    Calibrated parameter in the correction for wake deflection defining the horizontal offset scaled with downstream distance and ya
DEFAULT    C_NearWake     Calibrated parameter for the near-wake correction (-) [>1. and <2.5] or DEFAULT [=1.8]
DEFAULT    k_vAmb         Calibrated parameter for the influence of ambient turbulence in the eddy viscosity (-) [>=0.0] or DEFAULT [=0.05]
DEFAULT    k_vShr         Calibrated parameter for the influence of the shear layer in the eddy viscosity (-) [>=0.0] or DEFAULT [=0.016]
DEFAULT    C_vAmb_DMin    Calibrated parameter in the eddy viscosity filter function for ambient turbulence defining the transitional diameter fraction be
DEFAULT    C_vAmb_DMax    Calibrated parameter in the eddy viscosity filter function for ambient turbulence defining the transitional diameter fraction be
DEFAULT    C_vAmb_FMin    Calibrated parameter in the eddy viscosity filter function for ambient turbulence defining the value in the minimum region (-) [
DEFAULT    C_vAmb_Exp     Calibrated parameter in the eddy viscosity filter function for ambient turbulence defining the exponent in the exponential regio
DEFAULT    C_vShr_DMin    Calibrated parameter in the eddy viscosity filter function for the shear layer defining the transitional diameter fraction betwe
DEFAULT    C_vShr_DMax    Calibrated parameter in the eddy viscosity filter function for the shear layer defining the transitional diameter fraction betwe
DEFAULT    C_vShr_FMin    Calibrated parameter in the eddy viscosity filter function for the shear layer defining the value in the minimum region (-) [>=0
DEFAULT    C_vShr_Exp     Calibrated parameter in the eddy viscosity filter function for the shear layer defining the exponent in the exponential region (
DEFAULT    Mod_WakeDiam   Wake diameter calculation model (-) (switch) {1: rotor diameter, 2: velocity based, 3: mass-flux based, 4: momentum-flux based}
DEFAULT    C_WakeDiam     Calibrated parameter for wake diameter calculation (-) [>0.0 and <0.99] or DEFAULT [=0.95] [unused for Mod_WakeDiam=1]
DEFAULT    Mod_Meander    Spatial filter model for wake meandering (-) (switch) {1: uniform, 2: truncated jinc, 3: windowed jinc} or DEFAULT [=3]
DEFAULT    C_Meander      Calibrated parameter for wake meandering (-) [>=1.0] or DEFAULT [=1.9]
--- CURLED-WAKE PARAMETERS [only used if Mod_Wake=2 or 3] ---
True       Swirl               Switch to include swirl velocities in wake [only used if Mod_Wake=2 or Mod_Wake=3] (-) (switch) [DEFAULT=TRUE]
0          k_VortexDecay       Vortex decay constant for curl (-) [DEFAULT=0]
DEFAULT    NumVortices         The number of vortices in the curled wake model (-) [DEFAULT=100]
```

Figure D.14: Example of a FAST.Farm .fstf master input file used when performing the FAST.Farm simulations (part 1).

```
DEFAULT       sigma_D               The width of the vortices in the curled wake model non-dimesionalized by rotor diameter (-) [DEFAULT=0.2]
DEFAULT       FilterInit            Switch to filter the initial wake plane deficit and select the number of grid points for the filter {0: no filter, 1
DEFAULT       k_vCurl               Calibrated parameter for scaling the eddy viscosity in the curled-wake model (-) [only used if Mod_Wake=2 or Mod_Wak
DEFAULT       Mod_Projection        Switch to select how the wake plane velocity is projected in AWAE {1: keep all components, 2: project against plane
--- VISUALIZATION ---
False         WrDisWind      Write low- and high-resolution disturbed wind data to <RootName>.Low.Dis.t<n>.vtk etc. (flag)
0           NOutDisWindXY  Number of XY planes for output of disturbed wind data across the low-resolution domain to <RootName>.Low.DisXY<n_out>.t<n>.vtk
150.0       OutDisWindZ    Z coordinates of XY planes for output of disturbed wind data across the low-resolution domain (m) [1 to NOutDisWindXY] [unused
0           NOutDisWindYZ  Number of YZ planes for output of disturbed wind data across the low-resolution domain to <RootName>/Low.DisYZ<n_out>.t<n>.vtk
0.0,250,750,1000,1250,1750,2000,2250,2750 OutDisWindX    X coordinates of YZ planes for output of disturbed wind data across the low-resolution domain (m
0           NOutDisWindXZ  Number of XZ planes for output of disturbed wind data across the low-resolution domain to <RootName>/Low.DisXZ<n_out>.t<n>.vtk
1500.0      OutDisWindY    Y coordinates of XZ planes for output of disturbed wind data across the low-resolution domain (m) [1 to NOutDisWindXZ] [unused
5.0         WrDisDT        Time step for disturbed wind visualization output (s) [>0.0] or DEFAULT [=DT_Low or DT_Low-VTK] [unused for WrDisWind=False an
--- OUTPUT ---
False         SumPrint       Print summary data to <RootName>.sum? (flag)
99999.9     ChkptTime      Amount of time between creating checkpoint files for potential restart (s) [>0.0]
100.0       TStart         Time to begin tabular output (s) [>=0.0]
1           OutFileFmt     Format for tabular (time-marching) output file (-) (switch) {1: text file [<RootName>.out], 2: binary file [<RootName>.outb],
True        TabDelim       Use tab delimiters in text tabular output file? (flag) {uses spaces if False}
"ES20.3E2"  OutFmt         Format used for text tabular output, excluding the time channel. Resulting field should be 10 characters. (quoted string)
DEFAULT     OutAllPlanes   Output all wake planes at all time steps. [DEFAULT=False]
6           NOutRadii      Number of radial nodes for wake output for an individual rotor (-) [0 to 20]
0,5,10,15,20,25 OutRadii       List of radial nodes for wake output for an individual rotor (-)
9           NOutDist       Number of downstream distances for wake output for an individual rotor (-) [1 to NOutRadii] [unused for NOutRadii=0] rotor (-)
0.0, 200, 400, 600, 800, 1000, 1200, 1400, 1600 OutDist        List of downstream distances for wake output for an individual rotor (m) [1 to NOutDist] [
9           NWindVel       Number of points for wind output (-) [0 to 9]
10, 960, 1910, 1930, 2830, 3830, 3850, 4800, 5760 WindVelX       List of coordinates in the X direction for wind output (m) [1 to NWindVel] [unused for NW
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0    WindVelY       List of coordinates in the Y direction for wind output (m) [1 to NWindVel] [unused for NWin
150.0, 150, 150.0, 150.0, 150.0, 150.0, 150, 150, 150        WindVelZ       List of coordinates in the Z direction for wind output (m) [1 to NWindVel] [u
OutList The next line(s) contains a list of output parameters. (quoted string)
"RtAxsXT1, RtAxsYT1, RtAxsZT1"
"RtPosXT1, RtPosYT1, RtPosZT1"
"RtVRelT1, RtVRelT2, RtVRelT3"
"RtVAmbT1, RtVAmbT2, RtVAmbT3"
"TIAmbT1, TIAmbT2, TIAmbT3"
"CtT1N01, CtT1N02, CtT1N03, CtT1N04, CtT1N05"
"WkDfVxT1N01D1, WkDfVxT1N02D1, WkDfVxT1N03D1, WkDfVxT1N04D1, WkDfVxT1N05D1, WkDfVxT1N06D1"
"WkDfVxT1N01D2, WkDfVxT1N02D2, WkDfVxT1N03D2, WkDfVxT1N04D2, WkDfVxT1N05D2, WkDfVxT1N06D2"
"WkDfVxT1N01D3, WkDfVxT1N02D3, WkDfVxT1N03D3, WkDfVxT1N04D3, WkDfVxT1N05D3, WkDfVxT1N06D3"
"WkDfVxT1N01D4, WkDfVxT1N02D4, WkDfVxT1N03D4, WkDfVxT1N04D4, WkDfVxT1N05D4, WkDfVxT1N06D4"
"WkDfVxT1N01D5, WkDfVxT1N02D5, WkDfVxT1N03D5, WkDfVxT1N04D5, WkDfVxT1N05D5, WkDfVxT1N06D5"
"WkDfVxT1N01D6, WkDfVxT1N02D6, WkDfVxT1N03D6, WkDfVxT1N04D6, WkDfVxT1N05D6, WkDfVxT1N06D6"
"WkDfVxT1N01D7, WkDfVxT1N02D7, WkDfVxT1N03D7, WkDfVxT1N04D7, WkDfVxT1N05D7, WkDfVxT1N06D7"
"WkDfVxT1N01D8, WkDfVxT1N02D8, WkDfVxT1N03D8, WkDfVxT1N04D8, WkDfVxT1N05D8, WkDfVxT1N06D8"
"WkDfVxT1N01D9, WkDfVxT1N02D9, WkDfVxT1N03D9, WkDfVxT1N04D9, WkDfVxT1N05D9, WkDfVxT1N06D9"
"WkDfVxT2N01D1, WkDfVxT2N02D1, WkDfVxT2N03D1, WkDfVxT2N04D1, WkDfVxT2N05D1, WkDfVxT2N06D1"
"WkDfVxT2N01D2, WkDfVxT2N02D2, WkDfVxT2N03D2, WkDfVxT2N04D2, WkDfVxT2N05D2, WkDfVxT2N06D2"
"WkDfVxT2N01D3, WkDfVxT2N02D3, WkDfVxT2N03D3, WkDfVxT2N04D3, WkDfVxT2N05D3, WkDfVxT2N06D3"
"WkDfVxT2N01D4, WkDfVxT2N02D4, WkDfVxT2N03D4, WkDfVxT2N04D4, WkDfVxT2N05D4, WkDfVxT2N06D4"
"WkDfVxT2N01D5, WkDfVxT2N02D5, WkDfVxT2N03D5, WkDfVxT2N04D5, WkDfVxT2N05D5, WkDfVxT2N06D5"
"WkDfVxT2N01D6, WkDfVxT2N02D6, WkDfVxT2N03D6, WkDfVxT2N04D6, WkDfVxT2N05D6, WkDfVxT2N06D6"
"WkDfVxT2N01D7, WkDfVxT2N02D7, WkDfVxT2N03D7, WkDfVxT2N04D7, WkDfVxT2N05D7, WkDfVxT2N06D7"
"WkDfVxT2N01D8, WkDfVxT2N02D8, WkDfVxT2N03D8, WkDfVxT2N04D8, WkDfVxT2N05D8, WkDfVxT2N06D8"
"WkDfVxT2N01D9, WkDfVxT2N02D9, WkDfVxT2N03D9, WkDfVxT2N04D9, WkDfVxT2N05D9, WkDfVxT2N06D9"
"WkDfVxT3N01D1, WkDfVxT3N02D1, WkDfVxT3N03D1, WkDfVxT3N04D1, WkDfVxT3N05D1, WkDfVxT3N06D1"
"WkDfVxT3N01D2, WkDfVxT3N02D2, WkDfVxT3N03D2, WkDfVxT3N04D2, WkDfVxT3N05D2, WkDfVxT3N06D2"
"WkDfVxT3N01D3, WkDfVxT3N02D3, WkDfVxT3N03D3, WkDfVxT3N04D3, WkDfVxT3N05D3, WkDfVxT3N06D3"
"WkDfVxT3N01D4, WkDfVxT3N02D4, WkDfVxT3N03D4, WkDfVxT3N04D4, WkDfVxT3N05D4, WkDfVxT3N06D4"
"WkDfVxT3N01D5, WkDfVxT3N02D5, WkDfVxT3N03D5, WkDfVxT3N04D5, WkDfVxT3N05D5, WkDfVxT3N06D5"
"WkDfVxT3N01D6, WkDfVxT3N02D6, WkDfVxT3N03D6, WkDfVxT3N04D6, WkDfVxT3N05D6, WkDfVxT3N06D6"
"WkDfVxT3N01D7, WkDfVxT3N02D7, WkDfVxT3N03D7, WkDfVxT3N04D7, WkDfVxT3N05D7, WkDfVxT3N06D7"
"WkDfVxT3N01D8, WkDfVxT3N02D8, WkDfVxT3N03D8, WkDfVxT3N04D8, WkDfVxT3N05D8, WkDfVxT3N06D8"
"WkDfVxT3N01D9, WkDfVxT3N02D9, WkDfVxT3N03D9, WkDfVxT3N04D9, WkDfVxT3N05D9, WkDfVxT3N06D9"
"W1VDisX, W2VDisX, W3VDisX, W4VDisX, W5VDisX, W6VDisX, W7VDisX, W8VDisX, W9VDisX "
END of input file (the word "END" must appear in the first 3 columns of
this last OutList line)
```

Figure D.15: Example of a FAST.Farm .fstf master input file used when performing the FAST.Farm simulations (part 2).

Appendix E

Feedback-only standard deviation changes with increasing wind speeds

This Appendix documents how the standard deviations of key turbine and substructure parameters of the two FOWTs (Table E.1 for the VolturnUS-S and Table E.2 for the WindCrete) changed with increasing wind speeds when using FB-only control. These provide context to the LAC above-rated wind speed results of the VolturnUS-S and WindCrete FOWTs, shown in Sects. 5.4.2 and 5.5.2. Results are normalised to the standard deviations of the parameters recorded at a mean wind speed of 13 ms^{-1}.

Table E.1: Four-seed average normalised (to those at $\bar{V}$ = 13 ms^{-1}) standard deviations of key turbine and substructure parameters of the VolturnUS-S using FB-only control under increasing above-rated wind speeds.

$\bar{V}$, ms^{-1}	Blade pitch	Rotor speed	Power	Thrust	Platform pitch	Blade root bending moment	Tower base F-A bending moment
13	1.00	1.00	1.00	1.00	1.00	1.00	1.00
15	0.87	1.08	0.77	0.85	0.86	0.96	0.90
17	0.78	1.22	0.79	0.75	0.75	0.94	0.86
19	0.73	1.39	0.87	0.72	0.71	0.98	0.89
21	0.71	1.54	0.96	0.72	0.70	1.04	0.94
23	0.69	1.67	1.02	0.72	0.70	1.12	0.97
25	0.67	1.73	1.06	0.72	0.70	1.20	1.01

Table E.2: Four-seed-average normalised (to those at $\bar{V}$ = 13 ms^{-1}) standard deviations of key turbine and substructure parameters of the WindCrete using FB-only control under increasing above-rated wind speeds.

$\bar{V}$, ms^{-1}	Blade pitch	Rotor speed	Power	Thrust	Platform pitch	Blade root bending moment	Tower base F-A bending moment
13	1.00	1.00	1.00	1.00	1.00	1.00	1.00
15	0.72	1.07	0.98	0.81	0.90	0.93	0.88
17	0.57	1.13	1.02	0.74	0.89	0.97	0.86
19	0.49	1.16	1.04	0.71	0.90	1.02	0.88
21	0.44	1.14	1.02	0.68	0.90	1.11	0.89
23	0.42	1.13	1.01	0.68	0.92	1.20	0.92
25	0.39	1.10	0.99	0.69	0.95	1.31	0.96

Appendix F

FAST.Farm time series data alignment

This Appendix documents how the simulation data excerpts of the FOWTs simulated within FAST.Farm were extracted. When analysing the time series output of the FAST.Farm simulations of multiple FOWTs, the ten-minute data excerpts for each FOWT were staggered to account for the transport time of the wind between the turbines

$$\Delta T_{xRD} = \Delta X_{xRD} \bar{V} \tag{F.1}$$

Where ΔT_{xRD} is the transport time of the wind between the FOWTs within a simulation for a given RD separation, ΔX_{xRD} is the x-direction distance between the FOWTs for a given RD separation.

This ensured that, when analysing the data, comparable wind fields applied to the FOWTs were considered. This ultimately led to the requirement to begin and end the excerpts of the FOWTs at different points within the simulations. This required varying the total simulation times, depending on the RD separations and the mean wind speeds, to ensure that at least ten minutes of data was excerpted for each FOWT. The times within the simulation data to begin the ten-minute data excerpt of each FOWT for the 8 RD (1920 m) separation case, and the total simulation time allowed at each mean wind speed is shown in Table F.1.

Table F.1: Showing the times of the beginnings of the data excerpts for each FOWT at each mean wind speed at a separation distance of 8 RD, as well as the total simulation time allowed (rounded up to the nearest 5-second multiple) to ensure ten-minute data excerpts for each FOWT.

$\bar{V}$, ms^{-1}	Beginning of excerpt of T1, s	Beginning of excerpt of T2, s	Beginning of excerpt of T3, s	Total simulation time, s
11	100	275	450	1050
13	100	248	396	1000
15	100	228	356	960
17	100	213	326	930
19	100	202	303	905
21	100	192	283	885
23	100	184	267	870
25	100	177	254	855

Appendix G

Journal publications

Over the course of the research project, journal paper publications were pursued in order to affirm the novelty and original contributions made in the project. The following papers were published in the Wind Energy journal in January 2024 and August 2024, respectively. The first publication documents the OpenFAST and ROSCO source code modifications detailed in Chapter 3 and the FFCPC and FFIPC implementation and the filter design, documented in Chapter 4, in addition to the results of the 15 MW VolturnUS-S OpenFAST simulations presented in Chapter 5. The second publication documents the methodologies and results of the O&M case study presented in Chapter 6.

Check for updates

Received: 31 July 2023 | Revised: 21 November 2023 | Accepted: 12 December 2023

DOI: 10.1002/we.2891

RESEARCH ARTICLE

WILEY

LIDAR-assisted feedforward individual pitch control of a 15 MW floating offshore wind turbine

Andrew J. Russell[1] | Maurizio Collu[2] | Alasdair S. McDonald[3] | Philipp R. Thies[4] | Aidan Keane[5] | Alexander R. Quayle[6]

[1]Industrial Doctoral Centre for Offshore Renewable Energy, School of Engineering, King's Buildings, University of Edinburgh, Edinburgh, UK

[2]Department of Naval Architecture, Ocean and Marine Engineering, University of Strathclyde, Glasgow, UK

[3]School of Engineering, King's Buildings, University of Edinburgh, Edinburgh, UK

[4]Faculty of Environment, Science and Economy Engineering, University of Exeter, Penryn Campus, Penryn, UK

[5]Wood Renewables, Glasgow, UK

[6]Flotation Energy Ltd, Edinburgh, UK

Correspondence
Andrew J. Russell, Industrial Doctoral Centre for Offshore Renewable Energy, School of Engineering, King's Buildings, University of Edinburgh, Edinburgh EH9 3JL, UK.
Email: A.J.Russell-1@sms.ed.ac.uk

Funding information
UK Research and Innovation, Grant/Award Number: EP/S023933/1

Abstract

Nacelle-mounted, forward-facing light detection and ranging (LIDAR) technology can deliver benefits to rotor speed regulation and loading reductions of floating offshore wind turbines (FOWTs) when assisting with blade pitch control in above-rated wind speed conditions. Large-scale wind turbines may be subject to significant variations in structural loads due to differences in the wind profile across the rotor-swept area. These loading fluctuations can be mitigated by individual pitch control (IPC). This paper presents a novel LIDAR-assisted feedforward IPC approach that uses each blade's rotor azimuth position to allocate an individual pitch command from a multi-beam LIDAR. In this study, the source code of OpenFAST wind turbine modelling software was modified to enable LIDAR simulation and LIDAR-assisted control. The LIDAR simulation modifications were accepted by the National Renewable Energy Laboratory (NREL) and are now present within OpenFAST releases from v3.5 onwards. Simulations of a 15 MW FOWT were performed across the above-rated wind spectrum. Under a turbulent wind field with an average wind speed of 17 ms^{-1}, the LIDAR-assisted feedforward IPC delivered up to 54% reductions in the root mean squared errors and standard deviations of key FOWT parameters. Feedforward IPC delivered enhancements of up to 12% over feedforward collective pitch control, relative to the baseline feedback controller. The reductions to the standard deviation and range of the rotor speed may enable structural optimization of the tower, while the reductions in the variations of the loadings present an opportunity for reduced fatigue damage on turbine components and, consequently, a reduction in maintenance expenditure.

KEYWORDS
feedforward control, individual pitch control, LIDAR-assisted control, nacelle-mounted LIDAR

1 | INTRODUCTION

The application of light detection and ranging (LIDAR) technology for assisting wind turbine control is gathering momentum in its path towards commercialization. Nacelle-mounted, forward-facing LIDAR can provide measurements of the incoming wind field, the characteristics of which

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.
© 2024 The Authors. *Wind Energy* published by John Wiley & Sons Ltd.

can be used for feedforward turbine control. Traditionally, wind turbine control strategies have utilized feedback control, which updates the controller settings following the impact of the incoming disturbance. However, the drawback of this type of control strategy is that the wind turbine controller provides a delayed reaction to the incoming wind. Feedforward control overcomes this drawback by enabling turbines to actuate their torque, pitch or yaw systems in advance of the wind's impact upon the rotor.

According to the existing literature, the greatest benefits delivered by LIDAR-assisted control have been achieved when assisting with blade pitch control in above-rated wind speed conditions.[1–3] LIDAR-assisted feedforward pitch control can enhance turbine performance through improved rotor speed regulation and therefore more stable power capture. It is also capable of reducing the variation in the structural loadings, which has the potential to reduce the levelized cost of offshore wind energy by extending maintenance intervals and the lifetime of wind turbines.

Floating offshore wind turbines (FOWTs) are able to capture wind power from further distances offshore, where waters are too deep for the use of bottom-fixed turbines. However, as FOWTs are subject to softer foundation properties, their control systems have to be designed differently from their bottom-fixed counterparts, primarily to prevent negative aerodynamic damping as a consequence of unfavourable coupling between the platform's motion and the blade pitch controller in above-rated wind speeds.[4,5] These modifications give rise to greater rotor speed variation of FOWTs compared to bottom-fixed turbines,[4,6] resulting in larger rotor rotational (1P) and blade passing (3P) frequency ranges, and leading to the requirement for larger, stiffer turbine towers.

Within the literature, LIDAR-assisted pitch control has also demonstrated its ability to deliver reductions to the variations of the rotor speed, structural loads and platform motions of FOWTs. Studies have focused on FOWTs using spar-buoy[1] and tension-leg platform[2] foundations supporting NREL's 5 MW reference turbine, as defined by Jonkman et al,[7] and semi-submersible foundations for a 10 MW turbine model.[3] By 2024, turbines of up to 16 MW capacity will be commercially available.[8] Studies have now begun to predict the benefits attainable for these large floating turbines through simulations of LIDAR-assisted pitch control applied to the International Energy Agency (IEA)-Wind 15 MW reference turbine[9] mounted on the University of Maine's VolturnUS-S[10] semi-submersible platform.[11,12]

Individual pitch control (IPC) aims to mitigate the cyclic loads experienced by wind turbines due to variations in the wind profile across the rotor-swept area caused by wind shear, wind veer and turbulence. The concept of feedback IPC (FBIPC) was first explored by Bossanyi[13] and applied to floating turbines by Namik and Stol.[14] Within this strategy, the individual controller requires the three out-of-plane blade root bending moments and the rotor azimuth as feedback inputs to allow for the derivation of a pitch command for each blade, which adds to the command provided by the collective pitch controller.

Alternatively, feedforward individual pitch control (FFIPC) is able to deliver individual pitch commands in advance of the wind's impact and has been studied using various implementations.[15–18] Schlipf et al[15] used H_∞ optimization to design an individual pitch controller using a linear time-invariant (LTI) model. Raach et al[16] used a non-linear model predictive controller (NMPC) to deliver individual blade feedforward commands to a 5 MW floating wind turbine. Dunne et al[17,18] studied a LIDAR whose beams rotated with the blades and was used to assign a command to each blade using the same proportional–integral (PI) controller as was used for the collective pitch controller. The LIDAR used three wind speed inputs, one at each blade, at approximately 75% span to deliver the individual commands and resulted in improvements over the baseline feedback individual pitch controller.[18] Studies are yet to investigate the benefits of LIDAR-assisted FFIPC of large (15 MW+) FOWTs, whose rotor diameters are approaching 250 m. These large diameters will give rise to greater variations in the wind speed across the rotor-swept area, leading to significant variations in the loadings on the turbine's structures.

This paper aims to quantify the performance improvements and motion and loading reductions achievable through simulations of LIDAR-assisted pitch control of a 15 MW semi-submersible FOWT in above-rated wind speed conditions. The source code of OpenFAST wind turbine modelling software was modified to allow for the simulation of nacelle-mounted LIDAR wind velocity measurement. These modifications were accepted by NREL and are now present within OpenFAST releases from v3.5[19] onwards. Furthermore, the LIDAR measurements were interfaced with OpenFAST's control framework to enable LIDAR-assisted control within the Reference Open-Source Controller (ROSCO),[20] which was also modified within this work. A multi-beam LIDAR was simulated to enable calculation of the rotor-effective wind speed (REWS), representing the average wind speed across the rotor-swept area. The REWS was used by the implemented feedforward controller, which included collective pitch control (CPC) and IPC capabilities. The FFIPC approach used the rotor azimuth position of each blade to allocate an appropriate individual pitch command, which was combined with the command provided by the feedforward-feedback collective pitch controller (FFCPC). The time series performance of the FFCPC, the FFCPC with traditional FBIPC (FFCPC + FBIPC) and the FFCPC with feedforward IPC (FFCPC + FFIPC) was compared to that of the traditional feedback-only collective pitch controller (FB), which was used as the baseline. The impact of the rotor radius percentage span locations of the LIDAR beams on the performance of the FFCPC + FFIPC was also assessed. Performance benefits were quantified through comparison of normalized values of key performance metrics of various turbine and substructure parameters.

The paper is outlined as follows: Section 2 describes the modelling software, turbine and substructure models analysed. Section 3 outlines the modifications made to the OpenFAST and ROSCO source code to enable LIDAR measurement and feedforward control. Section 4 details the

10991824, 2024, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.2891 by Test, Wiley Online Library on [10/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

feedback and feedforward collective and individual pitch control theory. Section 5 describes the simulation setup and the key statistical metrics. Section 6 outlines the results, provides discussion on them and details the limitations of the study, and Section 7 offers conclusions from the study and the next steps of the research.

2 | TURBINE–SUBSTRUCTURE MODELLING

This section provides information regarding the software, turbine and supporting structure used when performing the simulations.

The numerical modelling tool employed for this study was NREL's open-source Fatigue, Aerodynamics, Structures, and Turbulence (FAST) code. This software is classed as an aero-hydro-servo-elastic tool for the modelling of full non-linear wind turbine models. It allows for the intricate specification of the turbine's design, external environment (wind, wave and current conditions) and controller. OpenFAST is the latest iteration of the FAST code and v3.4 was the original (before modifications) version of the program used in this study.[21]

2.1 | Turbine and support structure models

2.1.1 | Turbine

The turbine studied in this work was the IEA-Wind 15 MW reference turbine, developed between NREL and the Technical University of Denmark (DTU), via the IEA, as defined by Gaertner et al.[9] The specifications of the reference turbine are given in Table 1.

The turbine rotor operates with a minimum rotational speed of 5 rpm to avoid 3P interference with the tower/monopile natural frequencies[9] and reaches a rated rotational speed of 7.56 rpm at 10.59 ms^{-1}. The blades begin pitching at the rated wind speed in order to maintain the rotor rotation at its rated speed.

2.1.2 | Substructure

The floating substructure model utilized for this work was the University of Maine (UMaine) VolturnUS-S semi-submersible,[10] designed to support the IEA 15 MW reference turbine. Key specifications of the substructure are shown in Table 2.

TABLE 1 Parameters of the IEA-Wind 15 MW reference turbine.[9]

Parameter	Value	Units
Power rating	15	MW
Turbine class	IEC Class B	-
Cut-in wind speed	3	ms^{-1}
Rated wind speed	10.59	ms^{-1}
Cut-out wind speed	25	ms^{-1}
Minimum rotor speed	5.00	rpm
Maximum rotor speed	7.56	rpm
Rotor diameter	240	m
Hub height	150	m
Drive train	Direct drive	-

TABLE 2 Parameters of the VolturnUS-S semi-submersible model.[10]

Parameter	Value	Units
Material	Steel	-
Draft	20	m
Mass (including ballast)	17,839	kg
Mooring system	Three line catenary	-

10991824, 2024, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.2891 by Test, Wiley Online Library on [10/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

2.1.3 | Tower

The towers of FOWTs are required to have higher stiffness characteristics than bottom-fixed configurations due to increased inertial and gravity loads resulting from platform motions.[10] Therefore, the tower for the VolturnUS-S was designed separately from the configuration of the monopile outlined by Gaertner et al.[9] The tower was designed to have its first fore–aft and side-to-side natural frequencies outside of the 1P and 3P ranges. The 1P and 3P ranges were modified for the 15 MW turbine on the VolturnUS-S to account for the increased rotor speed variation caused by the platform motions. Therefore, a stiff-stiff tower, with first natural frequencies above 3P was required because of a very narrow soft-stiff range between 1P and 3P when additional safety factors were applied.[10] Further detail on the tower design for the 15 MW VolturnUS-S can be found within its definition document.[10]

3 | INTEGRATION OF LIDAR SIMULATION AND FEEDFORWARD CONTROL WITHIN OPENFAST

In the original version of OpenFAST v3.4 used in this study, LIDAR simulation was not possible within the program itself. Work has been conducted by Schlipf et al[22] and Guo et al[23] to develop an additional LIDAR simulator module to be called by the OpenFAST input file, with an additional .dll chain created for interface with the ROSCO controller. However, within this work, a LIDAR simulator has been developed that is integrated within the original OpenFAST framework. Here, LIDAR simulation is incorporated within the InflowWind module. The modifications made to OpenFAST (as will be outlined in Section 3.1) were proposed to NREL via a GitHub pull request. They were subsequently accepted and are present in the latest release of the program, v3.5.[19] The LIDAR measurements are linked with ServoDyn to enable interface with the modified ROSCO controller, developed to include LIDAR-assisted control. This section outlines the modifications made to the OpenFAST and ROSCO source code to allow for the integration of LIDAR simulation and LIDAR-assisted control.

3.1 | Integration of LIDAR simulation within InflowWind

As the InflowWind module of OpenFAST specifies the input wind conditions, it was identified as the point of integration of the LIDAR model. This was achieved through modification and addition to the source code, written in Fortran. Firstly, the source code was modified to allow for the LIDAR parameters and configuration to be specified within the InflowWind input file, allowing for customization by the user. The input files are then parsed and saved as parameters within the source code to be used within the program.

The LIDAR beam(s) can be configured in multiple ways. Single-point beam(s) can be configured where focal distances can be specified independently for each beam (as is the case in this study). Alternatively, a single beam can be configured in a continuous or pulsed manner. In the continuous mode, a single focal distance is configured, whereas, in the pulsed mode, measurements can be taken at multiple range gates. In both continuous and pulsed configurations, the beam is modelled as a Gaussian beam, whereby a weighting function defines the probe length at the focal point. The measurement frequency can also be specified. Additionally, the simulator can account for the impacts of platform motions on the movement of the nacelle, which are extracted from the ElastoDyn module. These can be used to modify the measurement coordinates of the LIDAR beam(s). The simulator considers the displacements in the x, y and z directions, though does not account for velocities, accelerations or rotations. The authors are aware that these omissions may lead to an overestimation of the LIDAR preview quality, which will be investigated in future work. For example, the preview quality can be affected by the linear velocity of the LIDAR, which will be included within the Doppler effect, and so, in this case, a motion compensation algorithm will be required to remove the contribution of the LIDAR velocity from the LOS speed measurement.

The LIDAR device's wind speed measurements are taken from the wind field provided to InflowWind such as those generated by TurbSim. The LIDAR's wind speed measurements can be outputted from the InflowWind module for user analysis and interface with other modules, such as ServoDyn, for LIDAR-assisted control.

In a real LIDAR system, each measurement consists of a measurement of the projection of the wind velocity vector onto a LIDAR beam direction, referred to as the line-of-sight (LOS) velocity, at selected points along the beam direction. The wind velocity vector is then determined through the wind field reconstruction (WFR) method, which is based upon trigonometric considerations.[22–24] In this work, the wind field is modelled as a horizontally homogeneous inflow. The single-point LIDAR beams are assumed to be perfectly aligned with the wind direction and are also assumed to not rotate during the simulations, meaning that the LIDAR beam directions are maintained in their orientations with respect to the horizontal plane. It follows that the transverse components of the wind field are assumed to be zero in the LIDAR frame of reference. This means that the wind field at a point can be determined from a single longitudinal velocity measurement, and sidesteps the need to conduct a full WFR under the current setup. In addition, it has been assumed that the wind velocity does not change from the point of measurement to the rotor, therefore assuming Taylor's frozen hypothesis.[25] In reality, the wind velocity would evolve in this interval, though previous studies have

10991824, 2024, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.2891 by Test, Wiley Online Library on [10/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

considered the wind evolution and found it to not be significantly impactful on the performance of LIDAR-assisted control, and that a simple linear filter design for LIDAR-assisted control is sufficient for various atmospheric stability conditions.[24]

3.2 | Interface of LIDAR measurement to ROSCO

ServoDyn is the module within OpenFAST responsible for defining control inputs and outputs and calling to the requested controller interface, which is typically a .dll file, namely, ROSCO. As ROSCO is compiled to a .dll using a separate Fortran solution to OpenFAST, an interface between the two is required so that inputs and outputs can be transferred. Within OpenFAST, this is achieved using the avrSWAP array, which is constructed within ServoDyn.

To achieve interface to the ServoDyn module from InflowWind, the LIDAR-measured wind speed outputs of the InflowWind module had to be specified as inputs within the ServoDyn module's source code, and these could then be added to the avrSWAP array to be sent to ROSCO.

The LIDAR-assisted collective pitch controller was created by adding an additional feedforward control subroutine within the ROSCO source code. The pitch controller calls the feedforward control script to compute the outputs to be sent to the feedforward-feedback controller for the determination of the pitch command. If individual feedforward pitch control is enabled, a further separate subroutine is called to compute the individual commands to be assigned to each blade.

4 | PITCH CONTROLLER THEORY

4.1 | Feedback pitch controller

The baseline feedback pitch controller used the default ROSCO inputs for the 15 MW VolturnUS-S,[10] accessible from GitHub.[26] It utilizes a gain-scheduled PI pitch controller, with gains detuned from the bottom-fixed version to prevent undesirable coupling between the blade pitch control actuation frequency and the floating platform rigid-body pitch oscillation natural frequency.[10] The feedback pitch controller requires the generator's rotational speed as the feedback input. The error (e_{FB}) between the low-pass filtered generator rotational speed (Ω_g) and the rated generator rotational speed ($\Omega_{g,rated}$) is first calculated as

$$e_{FB} = \Omega_{g,rated} - \Omega_g. \tag{1}$$

The proportional component of the feedback controller (P_{FB}) is then calculated through multiplication of e_{FB} with a scheduled (according to the blade pitch angle) proportional gain term (K_p)

$$P_{FB} = K_p e_{FB}. \tag{2}$$

The integral component of the feedback controller (I_{FB}) is calculated through multiplication of e_{FB} with the time step duration (Δt) and the gain-scheduled integral gain (K_I), which is added to the previous feedback integral component ($I_{FB,prev}$)

$$I_{FB} = I_{FB,prev} + \Delta t K_I e_{FB}. \tag{3}$$

The feedback pitch command (θ_{FB}) is determined through combination of P_{FB} and I_{FB}:

$$\theta_{FB} = P_{FB} + I_{FB}. \tag{4}$$

For floating turbines, an additional pitch command is provided through consideration of the nacelle rotational velocity to aid in overcoming negative aerodynamic damping. This is typically termed the floating feedback damping loop. The tower-top velocity ($\dot{x}_{tt}$) is measured and multiplied by a proportional feedback term ($K_{p,TFB}$) which is then combined with the command from the feedback pitch controller

$$\theta_{TFB} = K_{p,TFB}\dot{x}_{tt}, \tag{5a}$$

$$\theta_{FB} = \theta_{FB} + \theta_{TFB}, \tag{5b}$$

where θ_{TFB} is the tower-top feedback pitch command.

10991824, 2024, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.2891 by Test, Wiley Online Library on [10/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

4.2 | Feedforward collective pitch controller

The feedforward control approach used a five-beam LIDAR, with four beams focused around the rotor disc and one focused at hub height directly ahead of the turbine, as illustrated in Figure 1.

When using a multiple-beam LIDAR, as in this study, the REWS can be calculated. This was determined by averaging the longitudinal wind speeds at the LIDAR measurement points, which were evenly distributed across the rotor-swept area

$$REWS = \frac{\sum_{i=1}^{n_{beams}} U_{L,i}}{n_{beams}}, \tag{6}$$

where i is the beam number, $U_{L,i}$ is the longitudinal LIDAR-measured wind speed of a beam and n_{beams} is the number of LIDAR beams. The appropriate LIDAR REWS record to be used for the feedforward pitch command had to be recalled at the required time. This required accounting for the transport delay of the wind from the point of measurement to the turbine as well as time delays incurred by filtering. This was achieved through the use of a data storage row vector. Upon each new time step of the simulation, the values stored in the vector were shifted across to vacate the first space of the array for the latest LIDAR REWS measurement. The current LIDAR REWS measurement was then passed through a first-order low-pass filter with the following transfer function:

$$G_{LPF}(s) = \frac{2\pi f_{cut\text{-}off}}{s + 2\pi f_{cut\text{-}off}}, \tag{7}$$

where $f_{cut\text{-}off}$ is the cut-off frequency of the low-pass filter. Details on how this was obtained and the time delay that it incurred are provided in Section 4.5. The gain, G_{LPF}, was applied to filter the LIDAR REWS wind speed measurement, which was then assigned to the first position within the vector.

The approximate lead time (T_{lead}) of the current LIDAR measurement from the point of measurement to the turbine was

$$T_{lead,1} = \frac{\Delta x}{U_{ref}} \tag{8}$$

where Δx is the x-direction focal distance (equal for all beams) and U_{ref} is the reference wind speed. The time within the simulation at which the latest LIDAR wind speed measurement was expected to reach the turbine ($T_{impact,1}$) was

$$T_{impact,1} = T_{meas} + T_{lead,1}, \tag{9}$$

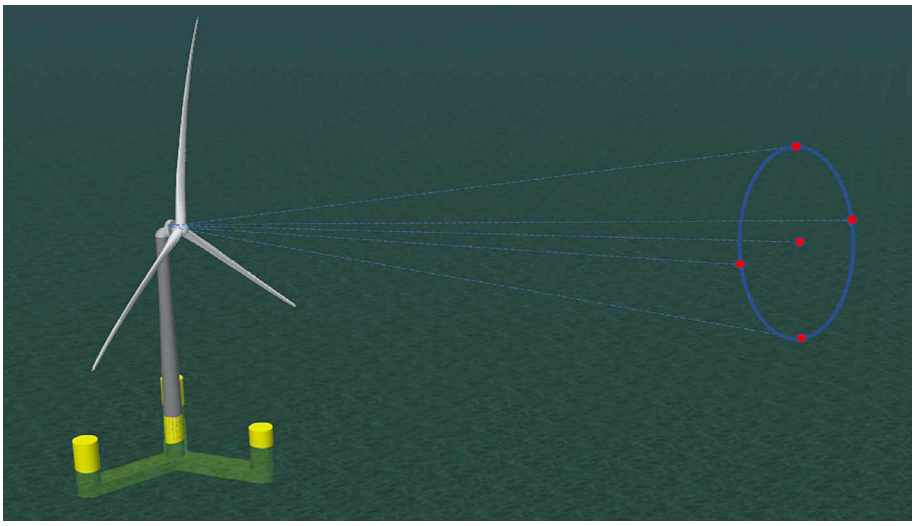


FIGURE 1 Illustrating a side view of the nacelle-mounted LIDAR beam trajectories, marked by blue dashed lines, and their focal positions, marked by red dots. Not to scale.

10991824, 2024, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.2891 by Test, Wiley Online Library on [10/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

where T_{meas} is the simulation time at the point when the measurement is taken. The impact time was also stored in a data storage row vector housing estimated impact times for each LIDAR REWS record. Similarly to the REWS records, the latest impact time record assumed the first position after the previous records were shifted across. The two data storage vectors were aligned in their positions of the expected time of impact and the filtered LIDAR REWS. The time delays incurred by the first-order low-pass filter of the LIDAR measurements (T_{filter}), the second-order low-pass filter of the pitch actuators (T_{pitch}), the LIDAR full-scan duration, T_{scan} (1 s) and the current simulation time, T, were then subtracted from each of the values in the impact time storage vector, such that values within the vector decreased as T increased. The remaining storage time ($T_{storage,j}$) of data point j at a certain T is defined as

$$T_{storage,j} = T_{impact,j} - T - T_{filter} - T_{pitch} - \frac{1}{2}T_{scan}. \tag{10}$$

The location within the $T_{storage}$ vector that had a value closest to zero was then selected, and the value at the same location within the filtered LIDAR REWS vector was selected as the REWS record to use when calculating the feedforward command for the current time step. The feedforward pitch angle (θ_{FF}) was then determined through interpolation of the wind speed-pitch angle curve, shown in Figure 2.

The difference between θ_{FF} and the feedforward command computed at the previous time step ($\theta_{FF,prev}$) was calculated and divided by the simulation time step duration, Δt, to give the feedforward pitch rate ($\dot{\theta}_{FF}$):

$$\dot{\theta}_{FF} = \frac{\theta_{FF} - \theta_{FF,prev}}{\Delta t}. \tag{11}$$

$\dot{\theta}_{FF}$ was then included within the calculation of I_{FB} (Equation 3):

$$I_{FF\text{-}FB} = I_{FF\text{-}FB,prev} + \Delta t\left(K_I e_{FB} + \dot{\theta}_{FF}\right), \tag{12}$$

where I_{FF-FB} is the feedforward modified feedback controller integral component and $I_{FF-FB,prev}$ is the value of I_{FF-FB} computed at the previous time step. The feedforward rate addition methodology has also been applied in previous studies.[1,3,24] P_{FB} (from Equation 2) and I_{FF-FB} were then combined to give the feedforward modified feedback pitch command:

$$\theta_{FF\text{-}FB} = (P_{FB} + I_{FF\text{-}FB}) + \theta_{TFB}. \tag{13}$$

4.3 | Feedback individual pitch controller

The FBIPC approach used in this study, as defined by Bossanyi[13] and Magar et al,[27] used the out-of-plane bending moment at the root of each blade and the rotor azimuth as its feedback inputs. The blade pitch angles are given as a function of the rotor azimuth angle, A, as follows[13]:

$$\theta_1 = \theta_{avg}\cos(A), \tag{14a}$$

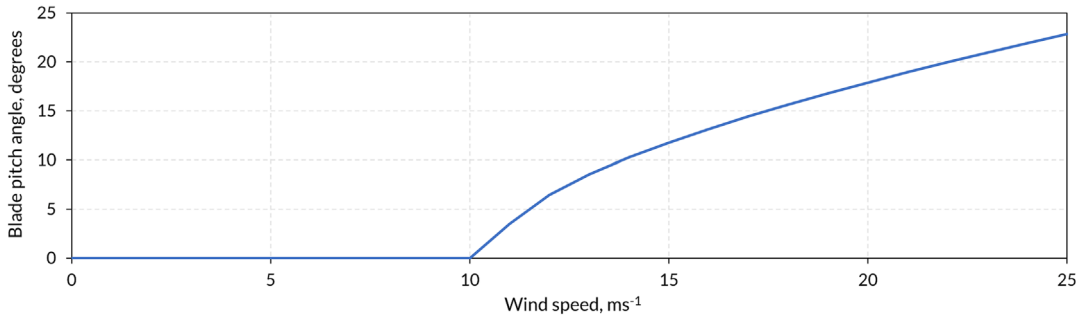


FIGURE 2 Wind speed–blade pitch angle curve.

10991824, 2024, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.2891 by Test, Wiley Online Library on [10/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

$$\theta_2 = \theta_{avg} cos\left(A + \frac{2\pi}{3}\right), \tag{14b}$$

$$\theta_3 = \theta_{avg} cos\left(A + \frac{4\pi}{3}\right), \tag{14c}$$

where θ_1, θ_2 and θ_3 are the pitch angles of each blade and θ_{avg} is the mean pitch angle.

The blade pitch angles are then transformed to the d–q axis using a multi-blade coordinate transformation, as was first demonstrated by Bossanyi.[13] This can be applied to convert the three-phase blade root bending moments onto a 2-D perpendicular axis

$$\begin{bmatrix} \theta_d \\ \theta_q \end{bmatrix} = \frac{2}{3} \begin{bmatrix} cos(A) & cos\left(A + \frac{2\pi}{3}\right) & cos\left(A + \frac{4\pi}{3}\right) \\ sin(A) & sin\left(A + \frac{2\pi}{3}\right) & sin\left(A + \frac{4\pi}{3}\right) \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}, \tag{15}$$

where θ_d and θ_q are the direct and quadrature axis outputs, respectively. The bending moments at the roots of the blades are assumed to be measured and then transformed to d–q axis using Equation (15). The controller is implemented during this transformation to keep the bending moments constant irrespective of changes in them due to variations in the wind speed. An inverse d–q transformation is then performed to convert the control outputs into individual pitch commands:

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = \begin{bmatrix} cos(A) & sin(A) \\ cos\left(A + \frac{2\pi}{3}\right) & sin\left(A + \frac{2\pi}{3}\right) \\ cos\left(A + \frac{4\pi}{3}\right) & sin\left(A + \frac{4\pi}{3}\right) \end{bmatrix} \begin{bmatrix} \theta_d \\ \theta_q \end{bmatrix}. \tag{16}$$

The individual pitch command for each blade was then combined with the command from the FFCPC, resulting in

$$\theta_{FF\text{-}FB,k} = (P_{FB} + I_{FF\text{-}FB}) + \theta_{TFB} + \theta_{FB,k}, \tag{17}$$

where $\theta_{FB,k}$ is the FBIPC command for an individual blade, k, and $\theta_{FF\text{-}FB,k}$ is the combined FFCPC + FBIPC pitch command for an individual blade.

4.4 | Feedforward individual pitch controller

In the present study, a novel individual pitch controller was designed to tune each blade's pitch angle depending on its azimuth position. It was used to provide an additional refinement to the feedforward collective pitch controller. The approach used the four outer beams of the five-beam LIDAR illustrated in Figure 1. These were focused at 90° azimuthal increments, as illustrated in Figure 3. Simulations were performed where the beams were positioned at various rotor radial span widths (30%, 50% and 70%) to determine the best-performing beam span positions for the FFCPC + FFIPC. This approach is most similar to that used by Dunne et al,[17,18] though differs in that the LIDAR beams are not rotating with the blades and the collective pitch controller is still used to provide a command to all blades.

Though the beams were focused at the specified points relative to the nacelle, the movement of the nacelle and its impact upon the measurement coordinates and beam azimuth angles were accounted for in real-time. The longitudinal wind speed was recorded at each LIDAR beam's measurement point. These were then individually low-pass filtered using the same cut-off frequencies as were used for the filter on the REWS for FFCPC. The pitch angle associated with the filtered wind speed recorded by each beam, $\theta_{setpoint}$, was then determined through interpolation of the wind speed-pitch angle curve (Figure 2). The azimuth positions of each blade (A_k) at the current time were determined by feedback of the OpenFAST azimuth output (A):

$$A_k = A + \frac{2\pi(k-1)}{3}, \tag{18}$$

where k is the blade number from 1 to 3. The current rotor rotational speed (Ω_r) was then used to predict the azimuth position of each blade ($A_{k,new}$) after the delay incurred by the low-pass filter that will be used on the individual commands (T_{IPC}). An example scenario of this is illustrated

10991824, 2024, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.2891 by Test, Wiley Online Library on [10/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

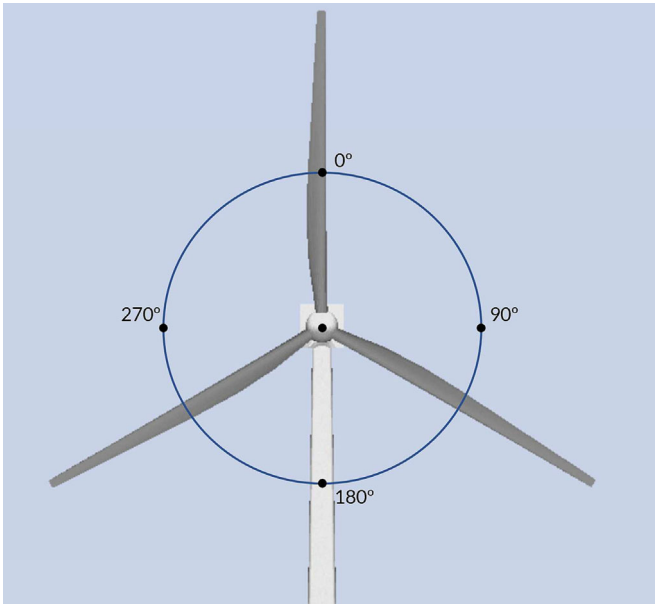


FIGURE 3 Rotor azimuth positions of the LIDAR focal positions within the rotor plane at 50% rotor radial span.

in Figure 4. This assumes that the rotor speed does not change over the delay period incurred by the low-pass filter on the individual pitch commands:

$$A_{k,new} = A_k + \Omega_r T_{IPC}. \tag{19}$$

The expected azimuth angle of each blade was then subtracted from the azimuth angle of each beam (A_i):

$$A_{k,dif} = A_i - A_{k,new}. \tag{20}$$

The beam associated with the smallest difference in azimuth angle ($A_{k,dif}$) was then selected to provide the individual pitch command to the individual blade. The error ($e_{FF,i}$) between the pitch angle setpoint associated with the beam, $\theta_{setpoint,i}$, and the current average blade pitch angle ($\theta_{c,avg}$) was then determined.

$$e_{FF,i} = \theta_{setpoint,i} - \theta_{c,avg}. \tag{21}$$

A low-pass filter with the following transfer function was used to prevent large variations in individual pitch commands:

$$G_{LPF,IPC}(s) = \frac{2\pi f_{cut\text{-}off,IPC}}{s + 2\pi f_{cut\text{-}off,IPC}}, \tag{22}$$

where $f_{cut\text{-}off,IPC}$ is the cut-off frequency of the low-pass filter applied to the feedforward individual pitch commands. Details on how this was determined and the time delay that it incurred are provided in Section 4.5. The gain, $G_{LPF,IPC}$, was applied to filter the feedforward individual pitch commands. This resulted in the feedforward pitch command for the individual blade, $\theta_{FF,k}$. Time series of an individual pitch command pre- and post-filtering are shown in Figure 5.

The time delay incurred by the filter required an additional term, T_{IPC}, to be considered within Equation (10) when accounting for the storage time when undertaking FFIPC such that

$$T_{storage,j} = T_{impact,j} - T - T_{filter} - T_{pitch} - \frac{1}{2}T_{scan} - T_{IPC}. \tag{23}$$

10991824, 2024, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.2891 by Test, Wiley Online Library on [10/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

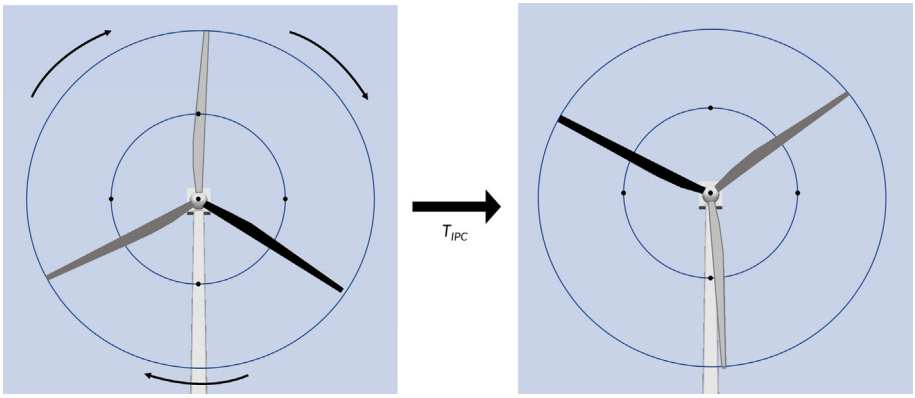


FIGURE 4 Illustrating an example scenario of the rotation of the turbine blades during the time delay incurred by the low-pass filter applied to the feedforward individual pitch commands.

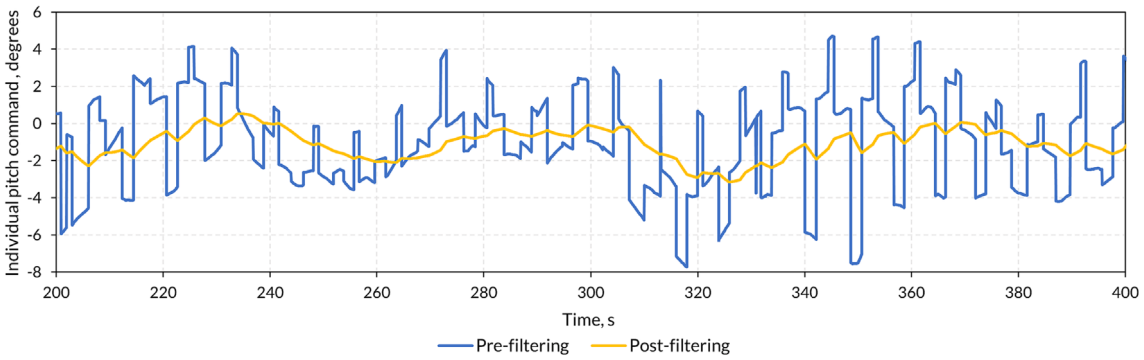


FIGURE 5 Excerpt of time series individual blade pitch commands pre- and post-low-pass filtering.

The individual component of the controller was designed to provide small adjustments to each blade in addition to the collective command that is provided to all of the blades:

$$\theta_{FF\text{-}FB,k} = (P_{FB} + I_{FF\text{-}FB}) + \theta_{TFB} + \theta_{FF,k}. \tag{24}$$

This resulted in an individual pitch variation as shown in Figure 6.

4.5 | Filter design and time delays

This section will outline how the cut-off frequencies and subsequent time delays of the low-pass filters used on the LIDAR REWS measurements, pitch actuators and feedforward individual pitch commands were determined. The time delays determined the minimum x-direction focal distance to ensure that the buffer time, T_{buffer}, in

$$T_{buffer} = T_{lead} - T_{filter} - T_{pitch} - \frac{1}{2}T_{scan} - T_{IPC} \tag{25}$$

was greater than zero at all reference wind speeds.

10991824, 2024, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.2891 by Test, Wiley Online Library on [10/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

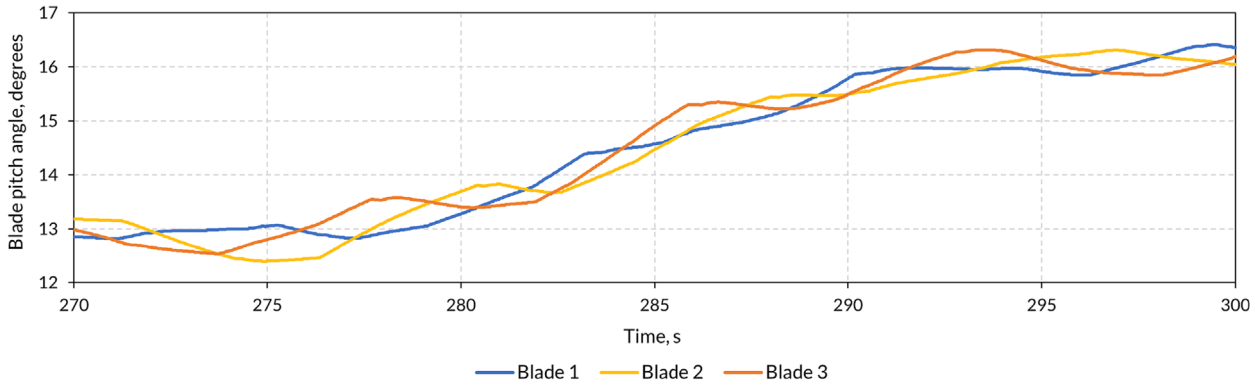


FIGURE 6 Excerpt of time series individual blade pitch variation when using feedforward individual blade pitch control.

4.5.1 | LIDAR REWS low-pass filter

A low-pass filter was applied to the LIDAR REWS time series measurements to ensure that frequencies above which there was a poor correlation between the LIDAR-measured REWS and the REWS experienced by the turbine were not considered. This point was defined by the cut-off frequency of the filter. Without filtering, or with the cut-off frequency set too high, uncorrelated frequencies would be used by the LIDAR-assisted controller, leading to incorrect control action. The cut-off frequency was determined through analysis of the transfer function between time series of the LIDAR-measured and turbine REWS, $|G_{RL}|$. $|G_{RL}|$ takes a value between 0 and 1 and indicates the quality of LIDAR preview.[24,28]

$$|G_{RL}(f)| = \frac{|S_{RL}(f)|}{S_{LL}(f)}, \tag{26}$$

where S_{LL} is the auto-spectrum of the LIDAR and estimated REWS and S_{RL} is the cross-spectrum of the LIDAR and turbine REWS estimates. This transfer function can also be derived theoretically if the turbulence spectra and coherence are known. From time series measurements, the spectra were calculated over a 10-min period. The resulting frequency domain (f) was converted into the wavenumber domain (k) using Taylor's frozen turbulence hypothesis[29]:

$$k = \frac{2\pi f}{U_{ref}}. \tag{27}$$

To compare to the LIDAR REWS measurements, the turbine-based REWS was determined by extracting outputs from multiple nodes within OpenFAST, discretized to 10-m increments across the rotor-swept area. These were then averaged to determine the REWS. S_{RL} and S_{LL} were determined using MATLAB's cross power spectral density (PSD)[30] and Welch's PSD estimate[31] functions, respectively. The spectra can also be determined analytically and details of this are provided within the literature.[28,29] The LIDAR and turbine REWS time series used in the analysis are shown in Figure 7. These resulted in a transfer function spectrum as shown in Figure 8.

The wavenumber at which $|G_{RL}|$ reaches -3 dB ≈ 0.7 is termed the maximum coherent wavenumber, $\hat{k}$. From Figure 8, this occurred at a wavenumber of 0.01563 rad m^{-1}. This process was repeated for beams positioned at 30% and 70% radial span, yielding a $\hat{k}$ of 0.0070 and 0.0208 rad m^{-1}, respectively. The cut-off frequencies for different reference wind speeds for each LIDAR beam configuration were then determined.[28]

$$f_{cut\text{-}off} = \frac{\hat{k}U_{ref}}{2\pi}. \tag{28}$$

Figure 9 shows the value of $f_{cut\text{-}off}$ at different reference wind speeds for the different beam configurations.

The low-pass filter for the REWS is first order and incurs a time delay (T_{filter}), which can be approximated at a certain frequency $\omega_{delay} = 2\pi f_{delay}$, $\omega_{cut\text{-}off} = 2\pi f_{cut\text{-}off}$ using equations from Schlipf[28]:

10991824, 2024, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.2891 by Test, Wiley Online Library on [10/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

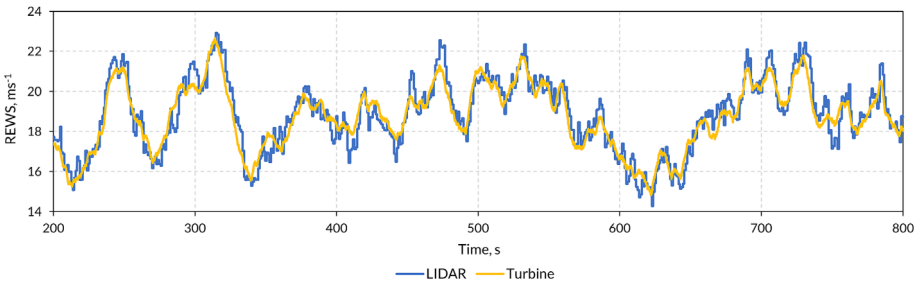


FIGURE 7 Ten-minute time series of LIDAR (radial span = 50%) and turbine-based REWSs used in the coherence analysis.

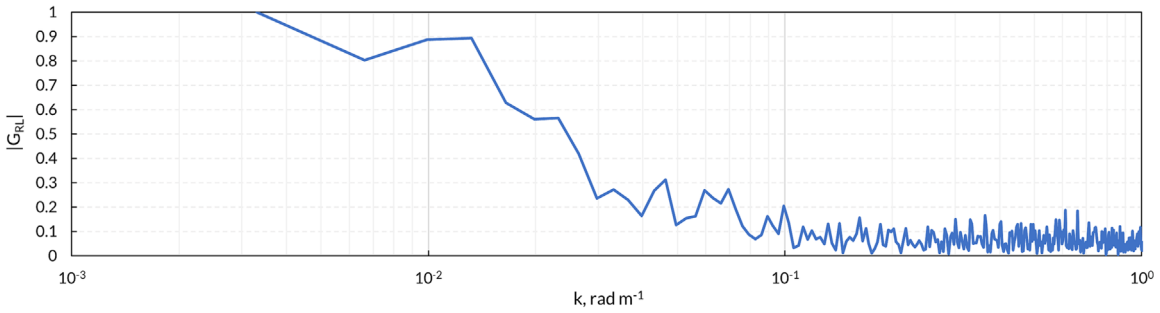


FIGURE 8 Transfer function between the LIDAR (radial span = 50%) and turbine REWS simulation data as a function of wave number.

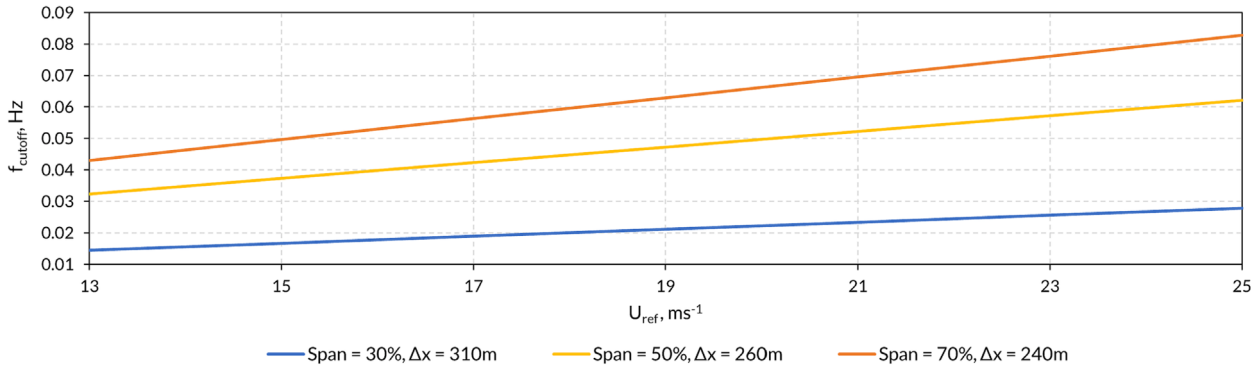


FIGURE 9 $f_{cut\text{-}off}$ at different average wind speeds for each of the radial span locations at their minimum x-direction focal distances.

$$T_{filter} = \frac{arctan(\omega_n)}{\omega_{delay}} with \omega_n = \frac{\omega_{delay}}{\omega_{cut\text{-}off}}. \quad (29)$$

The pitch actuator uses a second-order low-pass filter, whose time delay can be approximated using[28]

$$T_{pitch} = \frac{arctan\left(\frac{\sqrt{2}\omega_n}{1-\omega_n^2}\right)}{\omega_{delay}}. \quad (30)$$

In both Equations (29) and (30), f_{delay} was set to 0.025 Hz, as this is the critical frequency near where the IEA 15 MW turbine's rotor has higher fluctuations.[12]

10991824, 2024, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.2891 by Test, Wiley Online Library on [10/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

4.5.2 | FFIPC low-pass filter

As discussed in Section 4.4, a low-pass filter was required to smoothen and reduce the magnitude of the feedforward individual pitch commands. To achieve this, the cut-off frequency had to be set to a low level, which also mitigated interference with key turbine and substructure frequencies. In this work, $f_{cut\text{-}off}$ of the FFIPC filter was set to 0.0132 Hz, which is below the platform pitch rigid-body natural frequency (0.036 Hz[10]) as well as the frequency where the turbine's rotor has higher fluctuations (0.025 Hz[12]). There was not expected to be interference with the tower frequencies because, for this FOWT, the tower first natural frequencies were above the 3P range, as was explained in Section 2.1.3. Consequently, this filter resulted in an additional time delay (T_{IPC}) to be subtracted from T_{lead} in Equation (25), calculated using Equation (29). This resulted in the requirement for higher values of T_{lead} to ensure that T_{buffer} was greater than 0 at all wind speeds, as shown in Figure 10. This was achieved by setting the x-direction focal distance of the beams to 310, 260 and 240 m for beams focused at 30%, 50% and 70% radial span, respectively. The minimum x-direction focal distance decreased as the radial span increased because values of $\hat{k}$ and, therefore, $f_{cut\text{-}off}$ (Figure 9) were greater as the radial span increased. Higher values of $f_{cut\text{-}off}$ result in lesser time delays incurred by the low-pass filter on the LIDAR REWS. Therefore, less lead time was required and the LIDAR beams could be focused at positions closer to the turbine.

5 | SIMULATIONS

Turbulent wind fields were created using TurbSim,[32] with average wind speeds covering the above-rated wind spectrum, from 13 to 25 ms^{-1} in intervals of 2 ms^{-1}. These were generated for four random seeds of 1-h length (3800 s were simulated but the first 200 s were discarded to remove start-up transients) to provide a variety of turbulent wind profiles. Although the authors are aware of the challenges brought by the transition between the above- and below-rated control regions,[33] the behaviour of the controllers around this transition is not explicitly examined in this work. The turbulence was generated under the normal turbulence model (NTM), turbulence intensity (TI) Category B.[32] These conditions fall into design load case (DLC) 1.2, as defined in the IEC standards.[34] DLC 1.2 was used as this embodies the loads resulting from turbulence that occurs during the normal operation of a wind turbine throughout its lifetime. The FOWT was modelled in a water depth of 200 m and under irregular waves. The wind and waves were aligned in the same direction. The significant wave height (H_s) and peak spectral period (T_p) were adjusted to suit each average wind speed integer, with values interpolated from those defined by Allen et al,[10] and shown in Table 3.

5.1 | Statistical metrics

The benefits of LIDAR-assisted control can be assessed by statistical metrics, the choice of which varies depending on the desired outcome most relevant to the parameter. The metric most commonly used when assessing the performance of LIDAR-assisted control is the standard deviation,

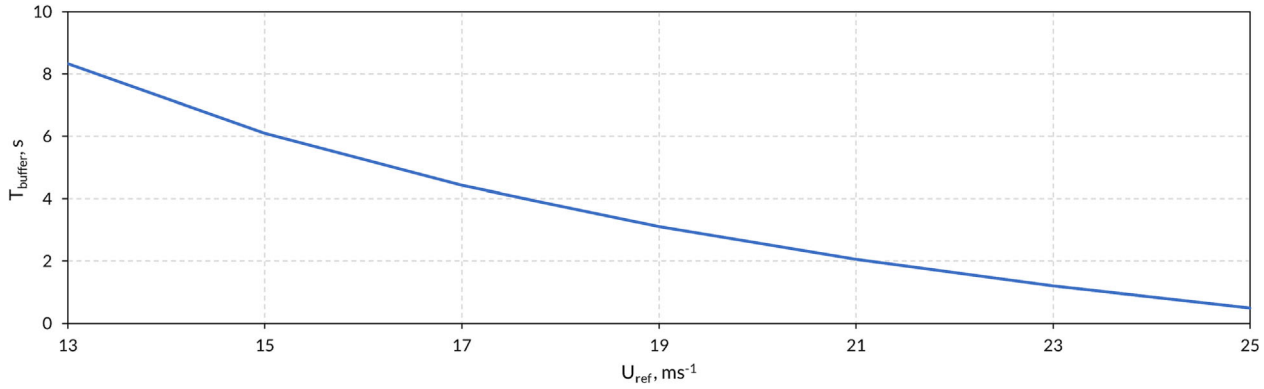


FIGURE 10 T_{buffer} at different reference wind speeds for beams positioned at 50% radial span at an x-direction focal distance of 260 m.

TABLE 3 IEC DLC 1.2 NTM wind and wave conditions.[10]

U_{hub} (ms^{-1})	13	15	17	19	21	23	25
H_s (m)	2.0	2.4	2.8	3.3	3.8	4.3	4.8
T_p (s)	7.5	7.6	7.8	8.3	8.8	9.2	9.7

10991824, 2024, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.2891 by Test, Wiley Online Library on [10/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 4 Detailing the key performance indicating metrics that were assessed for the specified parameter and their significance in demonstrating the efficacy of LIDAR-assisted control.

Parameter	Metric	Significance
Rotor speed	RMSE	This metric can assess how closely the rotor speed is maintained at its rated value (7.56 rpm). A lower error therefore indicates that the pitch controller is better able to deliver its control objective.
	Range	The range in rotor speed determines the 1P and 3P ranges, which are important for substructure and tower design. Allen et al[10] reported that, for the IEA 15 MW reference turbine on the VolturnUS-S, the constraining operational rotor speed range was 3.97 to 8.86 rpm, which impacted upon the tower design. Reductions to the 1P and 3P frequency ranges may allow for the use of a soft-stiff tower, which will be a lighter and cheaper option than a stiff-stiff tower.
	Maximum	Floating wind turbines are prone to over-speeding by up to 30% greater than the rated rotor speed,[4,6] which can lead to shutdowns and damage to turbine components.
Power	RMSE	This metric can assess how closely the generator power is maintained at its rated value (15 MW). Significant variation around the setpoint can lead to excessive stress on the generator.
Blade pitch	σ	Decreased standard deviations indicate a reduced degree of pitch actuation and, therefore, less demand on the pitch actuation mechanisms.
	Rate	This metric provides information on how quickly the blade pitch angle changes between measurements and can provide further indication of the demand on the pitch actuation mechanisms.
Thrust	σ	A reduced standard deviation in the thrust indicates a reduction in the variation on the force applied to the wind turbine.
Bending moments	σ	Reductions indicate that the loadings are more consistent, predictable and uniform across the structure and can result in reduced fatigue, failure rates and, therefore, lifetime extension.
	DEL	Similarly to σ, lower DELs indicate reduced stress on turbine components and structures, which can result in reduced fatigue and, ultimately, lifetime extension and cost savings.
Platform motions	σ	Reductions to the standard deviation indicate reduced magnitude and variation of platform motions, which can improve stability and lead to performance improvements.
	Acceleration (pitch only)	Access to the turbines can be limited by the acceleration of the platform motions, primarily in pitch and, therefore, it is favourable to reduce them.
	Maximum (pitch only)	The maximum magnitude of platform motions, particularly the platform pitch angle, is a design constraint for the turbine–substructure design as it plays a role in dictating the aerodynamic efficiency and the loads at the tower base.

σ, which provides knowledge of the degree of variation within the data set. The root mean squared error (RMSE) metric can be applied to assess the ability of the controller to maintain the control variables (viz., the rotor speed and power) at their desired set points. For bending moments, the damage equivalent loads (DELs) can also be compared. DELs equate to the load experienced by the turbine structure/component over a period of time that is equivalent to the damage caused by a specific event. The other metrics used to compare the time series results of the controllers were the ranges, maximum values, rates of change and accelerations. Table 4 highlights the key performance indicating metrics that were assessed for the various turbine and substructure parameters, with rationale explaining their significance.

6 | RESULTS AND DISCUSSION

This section presents the results of the OpenFAST simulations for the single turbine. Results are presented as normalized values, whereby the values were divided by that of the feedback-only controller.

$$X_{i,normalized} = \frac{X_i}{X_{FB}}, \tag{31}$$

where X_i is the value of a metric for a given parameter associated with a control strategy, i (FB, FFCPC, FFCPC + FBIPC or FFCPC + FFIPC), and $X_{i,normalized}$ is the normalized value of a metric for a given control strategy. This approach allows for direct comparison to the results of the feedback controller, which are normalized to 1. Therefore, a value that is less than one indicates a reduction (and therefore an improvement) compared to the baseline feedback controller and a value that is greater than one indicates an increase (and therefore a deterioration) compared to the baseline feedback controller.

10991824, 2024, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.2891 by Test, Wiley Online Library on [10/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

6.1 | Feedback versus feedforward control

This section outlines the results of the simulations. Time series results of a single seed with an average wind speed of 17 ms^{-1} will be first shown, to highlight the improvements to the turbine and substructure parameters when using the FFCPC or FFCPC + FFIPC compared to the feedback-only controller. Results of the FFCPC + FBIPC will also be presented to compare the effectiveness of the feedback and feedforward individual control elements. Four seed average results from across the above-rated spectrum will then be given to further demonstrate the capability of the feedforward controllers.

6.1.1 | Single wind field

Figure 11 presents an excerpt of time series results of the selected turbine and substructure parameters.

Table 5 provides a summary of the averaged normalized metric values for various parameters delivered by the feedforward control strategies relative to the baseline feedback controller across four random seed 1-h length simulations.

As can be seen from Figure 11 and Table 5, the collective and individual pitch control additions are able to deliver benefits to the key metrics of numerous turbine and substructure parameters. The rotor speed and generator power benefitted most significantly, with reductions of up to 54% in the RMSEs compared to the baseline feedback controller, demonstrating improved power tracking, and reduced ranges of 1P and 3P

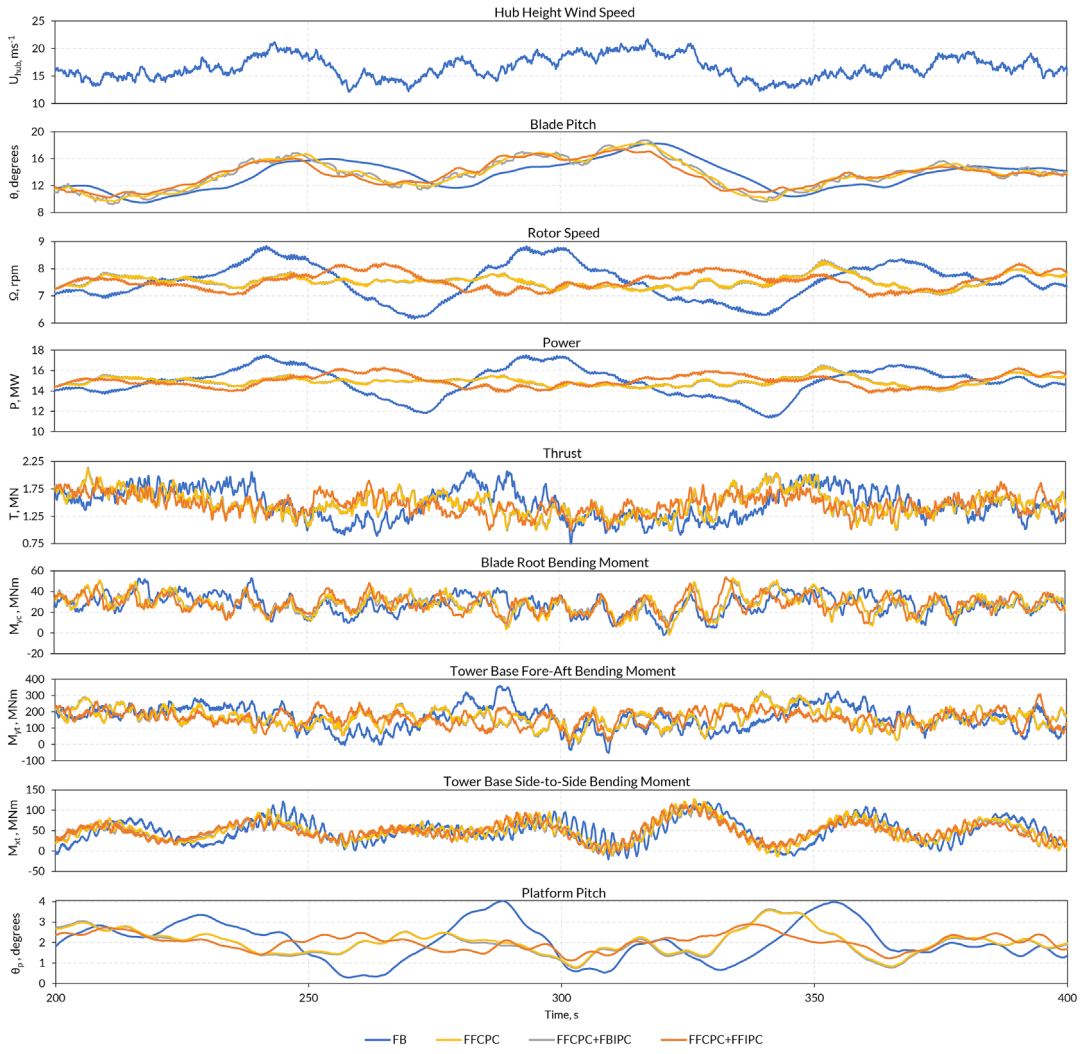


FIGURE 11 Two hundred-second excerpt time series results comparing the baseline FB pitch controller to the FF controllers. Simulations were performed under a turbulent wind field ($U_{avg} = 17\,ms^{-1}$, NTM, TI Category B) and irregular waves ($H_s = 2.83\,m$, $T_p = 7.85\,s$). The LIDAR beams were positioned at 50% rotor radial span for all feedforward control strategies.

10991824, 2024, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.2891 by Test, Wiley Online Library on [10/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 5 Normalized metric values of various parameters through the addition of feedforward pitch control versus baseline feedback control under a turbulent wind field ($U_{avg} = 17\,\mathrm{ms}^{-1}$, NTM, TI Category B) with irregular waves ($H_s = 2.83\,\mathrm{m}$, $T_p = 7.85\,\mathrm{s}$).

Parameter	Metric	Normalized FB	Normalized FFCPC	Normalized FFCPC + FBIPC	Normalized FFCPC + FFIPC
Performance					
Rotor speed	RMSE	1.00	0.48	0.47	0.48
	Range	1.00	0.54	0.54	0.56
	Maximum	1.00	0.93	0.93	0.92
Power	RMSE	1.00	0.47	0.46	0.46
Blade pitch	σ	1.00	1.02	1.03	0.95
	Rate	1.00	1.44	4.62	1.48
Thrust	σ	1.00	0.86	0.86	0.78
Bending moments					
Blade root	σ	1.00	0.97	0.89	0.94
	DEL	1.00	1.00	0.93	0.96
Tower base fore–aft	σ	1.00	0.85	0.85	0.80
	DEL	1.00	0.91	0.92	0.89
Tower base side-to-side	σ	1.00	1.00	0.96	1.02
	DEL	1.00	0.98	0.96	1.00
Platform motions					
Pitch	σ	1.00	0.74	0.73	0.64
	Acceleration	1.00	0.83	0.83	0.78
	Maximum	1.00	0.90	0.89	0.82
Roll	σ	1.00	1.01	0.96	1.03
Yaw	σ	1.00	1.01	0.79	0.94
Heave	σ	1.00	1.02	1.02	1.02
Surge	σ	1.00	0.88	0.88	0.76
Sway	σ	1.00	1.01	0.85	1.05

Note: The LIDAR beams were positioned at 50% rotor radial span for all feedforward control strategies. Results shown are average normalized values across the four random seed 1-h simulations at $U_{avg} = 17\,\mathrm{ms}^{-1}$.

frequencies by up to 46%. These benefits are attributed to the superior blade pitch preparation in advance of the wind's impact. This has a damping effect on the rotor's motion, thereby reducing its variation compared to the feedback-only controller. However, to achieve this, the average rate of change of the blade pitch was increased, especially when using FFCPC + FBIPC, which indicates an increased demand on the pitch actuation mechanisms. Despite the increases to the pitch rate, the average deviation of the blade pitch angle from its mean value was not affected significantly and was slightly reduced when FFCPC + FFIPC was employed. The standard deviations of the bending moments on the blade roots and tower were reduced by up to 20%, which led to reductions in DELs by up to 11% compared to the baseline feedback controller. The variation in the blade root loads were reduced most when combining the feedforward CPC with the traditional FBIPC (FFCPC + FBIPC) because maintaining the blade root bending moments as constant is a control objective of this approach, as was explained in Section 4.3. In terms of platform motions, the greatest beneficiary was the pitch degree of freedom, where 26% and 36% reductions in the standard deviation compared to the baseline were recorded for FFCPC and FFCPC + FFIPC, respectively. This was expected because the thrust force reported its greatest reduction in standard deviation with FFCPC + FFIPC, meaning that a more consistent force was applied to the rotor when using feedforward control instead of feedback-only control, resulting in reduced variation in the force pushing the turbine backwards. This led to the reduced variations seen in the pitching of the platform as well as the fore–aft bending moment acting on the tower base.

Overall, the FFCPC + FFIPC strategy further enhanced the FFCPC by providing additional tuning to the pitch of the individual blades. Performance parameters benefitted from up to 8% further reductions in the standard deviation compared to the baseline feedback controller, though the benefits to the rotor speed and power were not enhanced further. The blade root and tower base fore–aft bending moments benefitted from further 3% and 5% reductions in their standard deviations compared to the baseline, respectively. The most significant improvement delivered by FFCPC + FFIPC to the platform's motions was in surge, which achieved a further 12% reduction in standard deviation compared to the baseline feedback controller, while the platform pitch observed a 10% further reduction in its standard deviation and 5% and 8% further reductions to its acceleration and maxima, respectively.

10991824, 2024, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.2891 by Test, Wiley Online Library on [10/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

The FFCPC + FFIPC strategy was able to offer improvements upon the traditional FBIPC strategy, though both excelled in different areas. When combined with FFCPC, FFIPC gave greater improvements to the rotor thrust than when FBIPC was used, while also greatly benefiting the platform pitch and surge motions in addition to the tower base fore–aft bending moment. These benefits were also achieved without large increases to the blade pitching rate, as were observed with the FBIPC. However, FBIPC was able to offer benefits over the FFIPC to the blade root and tower side-to-side bending moment as well as the yaw and sway motions of the platform.

The improvements offered by the FFCPC + FFIPC strategy over the baseline feedback control can also be illustrated by PSD plots. Figure 12 presents PSD plots for the rotor speed, blade pitch, platform pitch and tower base fore–aft bending moment. The sampling frequency used in the simulations was 40 Hz, and this is the maximum frequency value of the plots shown in Figure 12.

The rotor speed, platform pitch and tower base fore–aft bending moment's spectra were reduced in the frequency range below the 1P range at 0.2 Hz, indicating a reduction in the influence of the wind disturbance, as was expected, and was consistent with the findings shown in Figure 11 and Table 5. Above this frequency, the spectra for the rotor speed and tower base fore–aft bending moment when using FFCPC + FFIPC were very similar to those of the baseline controller because the feedforward pitch commands were damped out by the low-pass filter. The spectrum of the blade pitch for FFCPC + FFIPC was greater than that of the baseline between the 1P and 3P ranges, which was also expected due to the increase to the average pitch rate.

6.1.2 | Above-rated wind spectrum

Simulations were performed across the above-rated wind spectrum from 13 to 25 ms^{-1} in intervals of 2 ms^{-1}. This aimed to capture the performance benefits attainable from FFCPC with FBIPC or FFIPC across the full operating range of the pitch controller to allow for the assessment of lifetime benefits. The four seed average wind speed series results for various turbine and substructure parameters are shown in Figure 13.

Figure 13 re-iterates the enhancements to FFCPC provided by the FFIPC and FBIPC controllers, as these were able to deliver greater standard deviation reductions for the majority of the above-rated wind speeds. In general, the improvements delivered by the feedforward controllers to the turbine and substructure parameters were least significant at lower above-rated average wind speeds closer to the above–below-rated

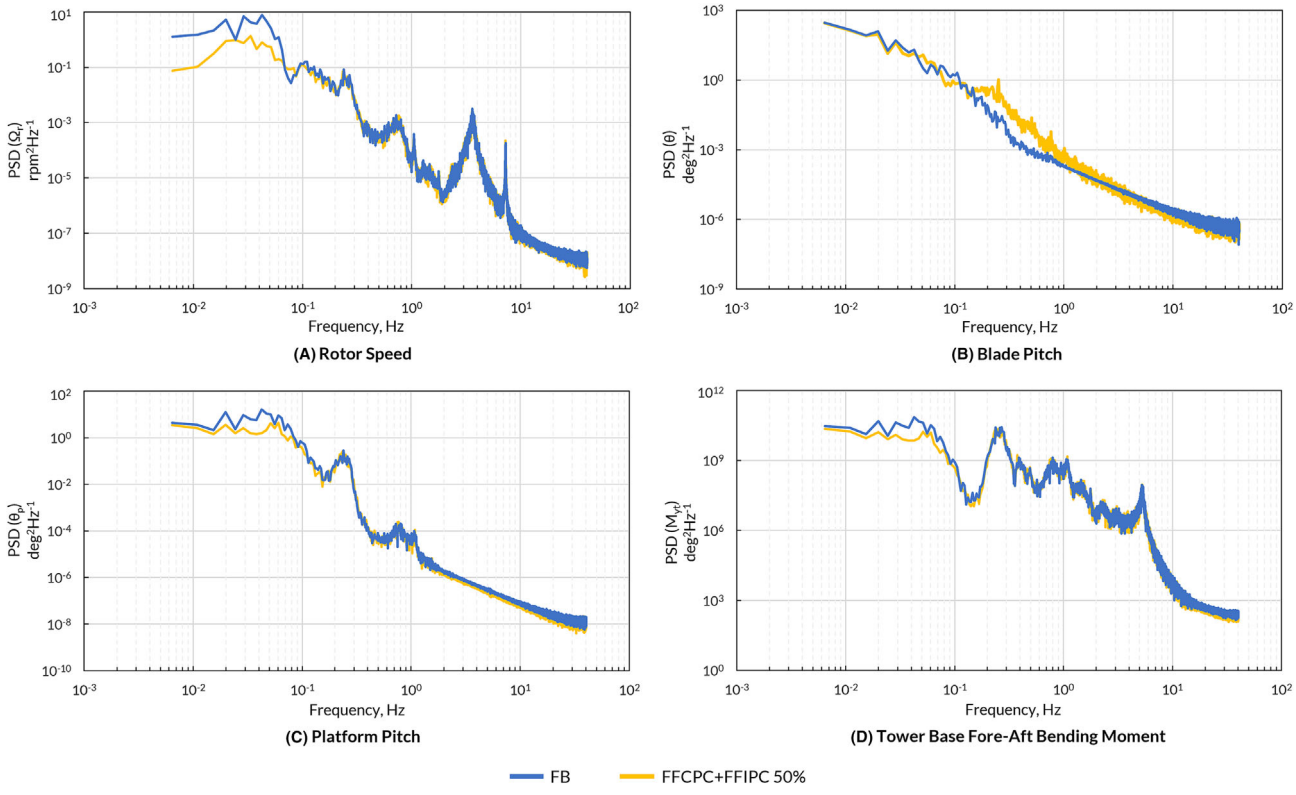


FIGURE 12 PSD plots of the rotor speed, blade pitch, platform pitch and tower base fore–aft bending moment generated from 1 h simulations with a turbulent wind field ($U_{avg} = 17\,ms^{-1}$, NTM, TI Category B) and irregular waves ($H_s = 2.83\,m$, $T_p = 7.85\,s$). The LIDAR beams were positioned at 50% rotor radial span for the FFCPC + FFIPC control strategy.

10991824, 2024, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.2891 by Test, Wiley Online Library on [10/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

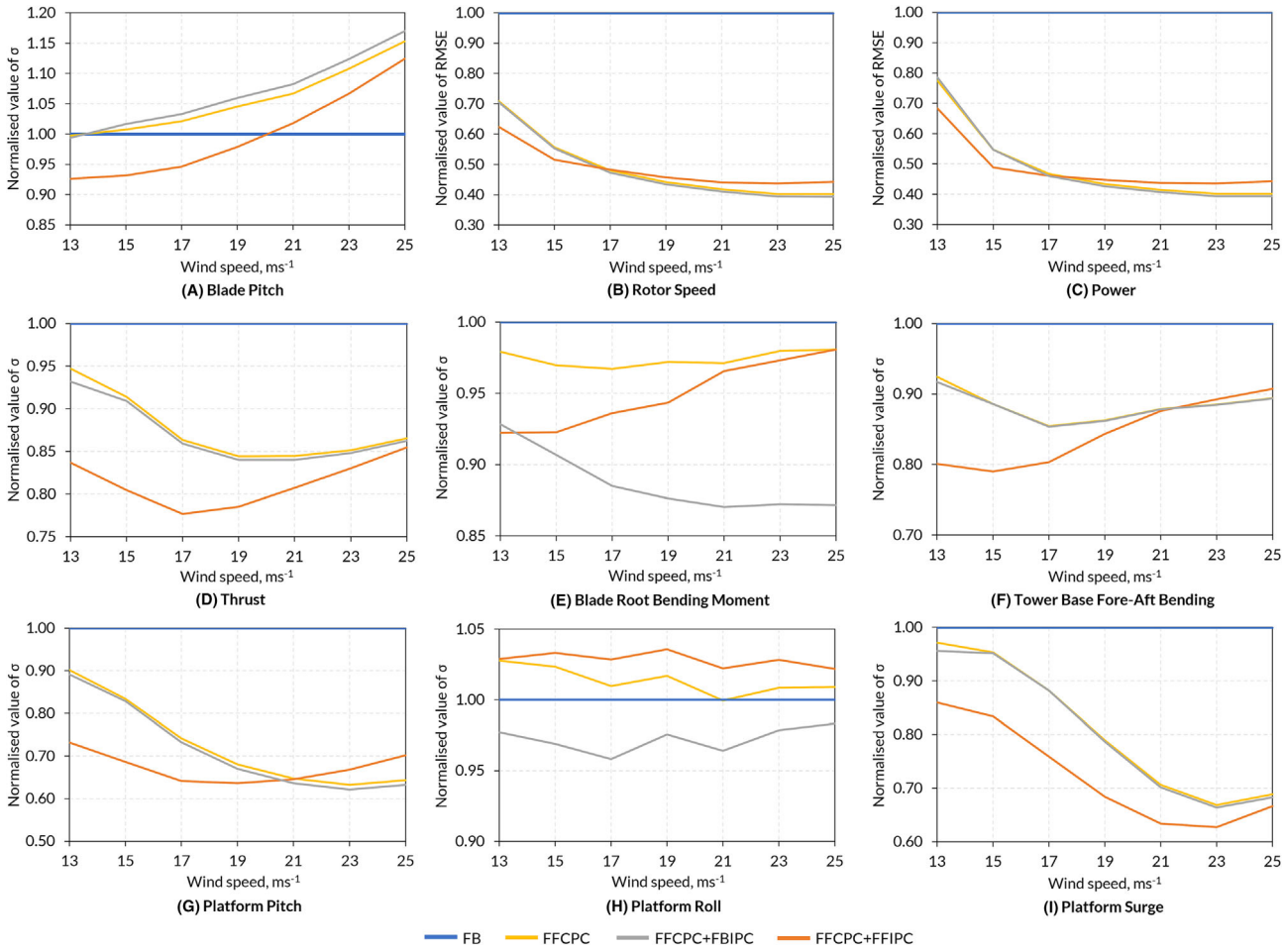


FIGURE 13 Normalized σ or RMSE values delivered by the feedforward controllers compared to the baseline FB-only pitch controller across the above-rated wind spectrum. The LIDAR beams were positioned at 50% rotor radial span for all feedforward control strategies. The results shown are the four seed average normalized values from the 1-h simulations at each average wind speed integer for each control configuration.

transition point (13 and 15 ms^{-1}). This is because, as these are average wind speeds, part of the simulations are spent with wind speeds below the rated speed where the feedforward controllers do not operate, meaning that responses were identical to the feedback-only controller, thereby reducing the differences between the datasets.

As the wind speeds increased, some parameters experienced greater variation and higher magnitude oscillations when using the baseline feedback controller and so this led to more room for improvement for the feedforward controllers. This was true for the rotor speed (Figure 13B) and power (Figure 13C), which benefitted most significantly, with reductions in their RMSEs by up to 60% at an average wind speed of 25 ms^{-1}. This came at the expense of increased variation in the blade pitch as the wind speed increased, by up to 17% greater than the baseline at an average wind speed of 25 ms^{-1}, though FFCPC + FFIPC was able to reduce this increase. The rotor thrust (Figure 13D) saw reductions in the standard deviation delivered by the feedforward controllers increase until approximately 17–19 ms^{-1} before decreasing at higher wind speeds due to diminishing variation in the rotor thrust as the wind speed increased. This was mirrored within the loadings and platform motions, as reductions to the standard deviations of the tower base fore–aft bending moment (Figure 13F), platform pitch (Figure 13G) and platform surge (Figure 13I) also diminished at higher wind speeds due to the decreased reduction to the standard deviation of the rotor thrust. The benefit to the blade root bending moment (Figure 13E) remained relatively consistent across the full above-rated wind spectrum when using FFCPC, though greater improvements were observed when pairing FFCPC with FBIPC, as maintaining the blade root bending moments constant is a control objective of this approach, as was explained in Section 4.3.

6.2 | Comparison of beam span locations

Figure 14 shows the above-rated wind spectra results of the simulations when using beams focused at 30%, 50% and 70% rotor radial span.

10991824, 2024, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.2891 by Test, Wiley Online Library on [10/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

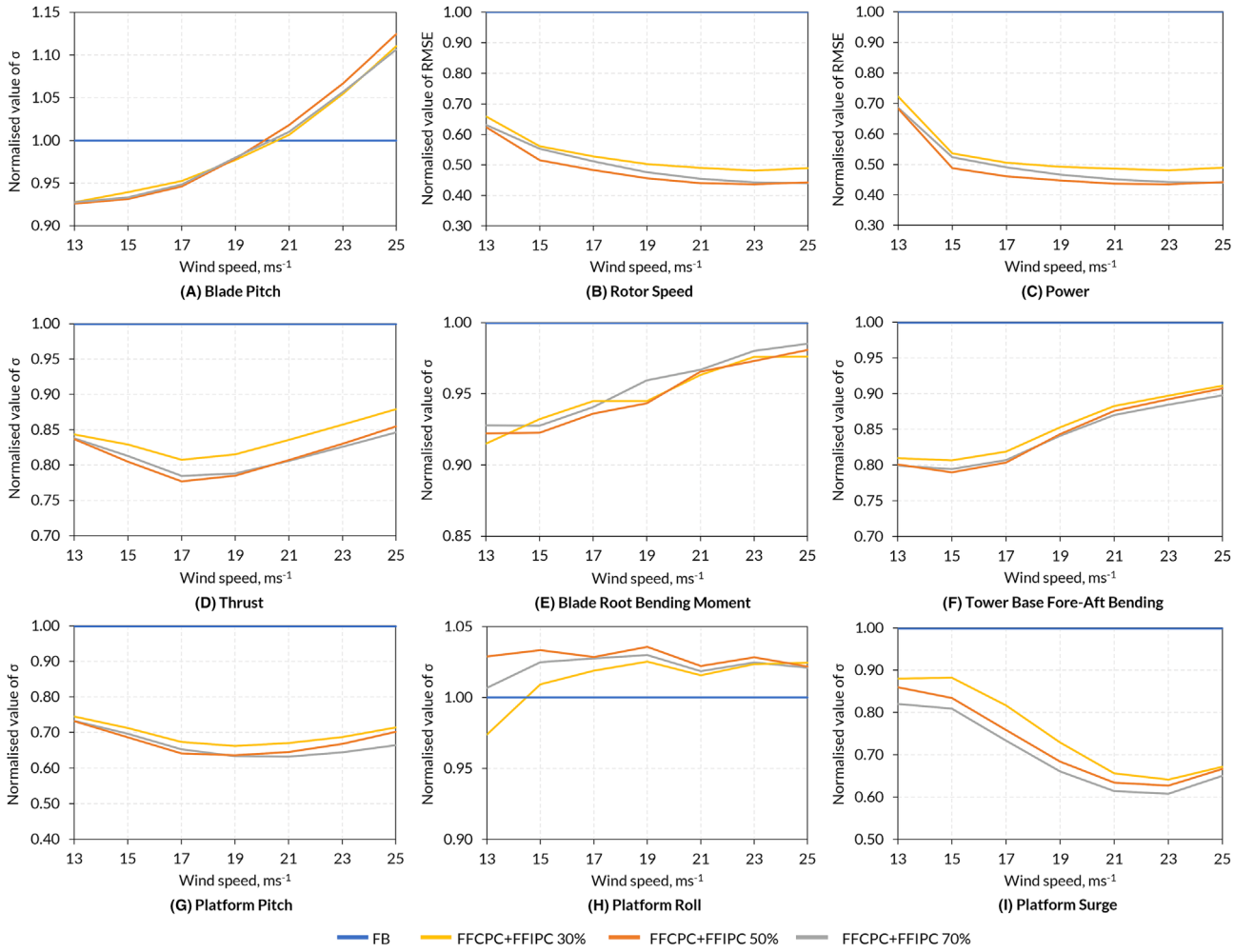


FIGURE 14 Normalized σ or RMSE values delivered by the FFCPC + FFIPC controller with beams positioned at 30%, 50% and 70% rotor radial span compared to the baseline feedback pitch controller across the above-rated wind spectrum. The results shown are the four seed average normalized values from the 1-h simulations at each average wind speed integer for each control configuration.

From Figure 14, the FFCPC + FFIPC performed marginally better across the above-rated wind spectrum when the beams were focused at 50% or 70% than at 30% span. At 30% span, the benefits to the rotor speed (Figure 14A), power (Figure 14B), rotor thrust (Figure 14D) and platform pitch (Figure 14G) were all slightly less significant. The FFCPC + FFIPC performed similarly with beams positioned at 50% and 70% span. However, there were marginally greater benefits delivered when the beams were focused at 50% span compared to at 70% span, namely, in the rotor speed (Figure 14B), power (Figure 14C) and blade root bending moment (Figure 14E), by up to 4%.

The differences seen in the performance of the feedforward IPC at various rotor radial spans can be explained through consideration of the variation in the wind speed across the rotor-swept area as the span positions increase. At 70% span, the distance between the opposite beams was approximately 170 m, compared to 120 m at 50% span and 75 m at 30% span. As a single average wind speed was used by the feedforward collective pitch controller, the potentially larger differences in wind speed at the measurement locations impact upon the REWS used and so may lead to sub-optimal commands issued by the feedforward CPC. Similarly, at the smallest span length simulated (30%), the variation in the measured wind velocities may not be sufficient to represent the full rotor-swept area. This was evidenced by the lower coherence between the LIDAR-estimated and turbine-based REWS, meaning that lower cut-off frequencies were required within the low-pass filter, limiting the feedforward control action. Furthermore, for the individual control element, it is pertinent to capture the variation across the rotor-swept area in order to provide the appropriate command to the individual blades. Therefore, with the beams positioned at 50% rotor radial span, the LIDAR was able to capture the wind velocities appropriate for the effective operation of both the collective and individual feedforward pitch control elements.

10991824, 2024, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.2891 by Test, Wiley Online Library on [10/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

6.3 | Study limitations

As was discussed in Section 3.1, the wind velocity is assumed to not evolve from the point of measurement to the turbine (therefore assuming Taylor's frozen hypothesis), 100% LIDAR measurement availability has been assumed and the impact of blade blockage of the LIDAR beams has not been considered. Moreover, the impacts of the velocities, accelerations and rotations of the LIDAR device were not accounted for. The assumptions made may have resulted in an overestimation of the LIDAR preview quality, which will be investigated in future work. Therefore, it is recommended that the results presented in this study be considered as the best-case scenario of what can be achieved through the implementation of the detailed LIDAR-assisted controllers used within this work. Furthermore, the simulations were only performed under normal operating conditions (DLC 1.2). Therefore, the scope of this study does not account for extreme operating conditions and gusts.

7 | CONCLUSIONS

This study has presented a novel LIDAR-assisted FFIPC approach in combination with a feedforward CPC strategy. This was applied to the IEA-Wind 15 MW reference wind turbine mounted on the UMaine VolturnUS-S semi-submersible floating substructure. The benefits of the feedforward IPC were investigated using modified versions of OpenFAST and ROSCO. LIDAR simulation was added to the OpenFAST program through modification to the source code of the InflowWind module. The changes made are now present in the latest release of OpenFAST, v3.5. LIDAR measurements were then interfaced to ServoDyn to allow for transfer to ROSCO for LIDAR-assisted control, where code for collective and individual feedforward pitch control was incorporated. The results presented demonstrate the efficacy of the LIDAR simulation and control additions to OpenFAST and ROSCO and can serve as a demonstration of the capabilities of the LIDAR simulation available in OpenFAST v3.5.

Simulations were performed on a single floating turbine, which yielded significant time series standard deviation, RMSE, range and maxima reductions for both collective and individual feedforward controllers over the baseline feedback collective and individual controllers. The individual feedforward pitch controller was able to provide enhancement to the collective feedforward pitch controller through increased standard deviation reductions over the baseline feedback controller by up to 12% (recorded in the platform surge motion) at an average wind speed of 17 ms^{-1}. Simulations were performed to cover the full above-rated wind spectrum. For some parameters, the standard deviation and RMSE reductions delivered by the feedforward controllers increased with increasing wind speed, while for others, the reductions increased up to an average wind speed of 19 ms^{-1} before diminishing at higher average wind speeds. The feedforward collective combined with the feedforward individual controller was found to give the greatest benefits over the baseline feedback controller, primarily in the rotor thrust, platform pitch, platform surge and tower base fore–aft bending moment, due to the wind speed-azimuth focused tuning of the blade pitch. The combined feedforward collective and individual pitch controller was studied at multiple rotor radial percentage span locations. The controller was found to perform best when the beams were focused at 50% span, offering the best balance between the rotor-effective average wind speed used by the collective control element while capturing the variation across the rotor-swept area used for the individual control component.

Future work aims to investigate the potential degree of overestimation in the LIDAR preview quality obtained from the detailed LIDAR simulator through comparison with an alternative simulator. Lifetime operations and maintenance cost reductions of floating offshore wind turbines will also be quantified through assessment of the lifetime failure rate and DEL reductions. Further work will also aim to investigate the potential for structural optimization of the tower given the LIDAR-assisted controller's ability to reduce the range in the rotor speed and therefore the 1P and 3P frequency ranges.

AUTHOR CONTRIBUTIONS

Andrew J. Russell conceived the concept, performed the modifications to the OpenFAST and ROSCO source codes, executed the simulations and prepared the paper. Maurizio Collu, Alasdair S. McDonald, Philipp R. Thies, Aidan Keane and Alexander R. Quayle provided supervision and suggestions and reviewed the paper.

ACKNOWLEDGEMENTS

This work was funded by UK Research and Innovation as part of the EPSRC and NERC Industrial CDT for Offshore Renewable Energy (IDCORE), Grant EP/S023933/1. The authors wish to acknowledge the support provided by NREL, namely, Andy Platt and Derek Slaughter, for facilitating the integration of the proposed changes into OpenFAST v3.5.

CONFLICT OF INTEREST STATEMENT

The authors declare no potential conflict of interest.

PEER REVIEW

The peer review history for this article is available at https://www.webofscience.com/api/gateway/wos/peer-review/10.1002/we.2891.

10991824, 2024, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.2891 by Test, Wiley Online Library on [10/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

DATA AVAILABILITY STATEMENT

The original modified version of OpenFAST v.3.4, which includes the LIDAR simulator used in this work can be accessed via https://github.com/Russell9798/OpenFAST-v3.4-Lidar-IfW-Original/releases/tag/2.0. The modified version of ROSCO v2.6 with the feedforward control additions can be accessed via https://github.com/Russell9798/ROSCO-v2.6-LAC-IPC/releases/tag/2.0. Simulation datasets used for this paper are available and can be assessed via https://doi.org/10.17605/OSF.IO/FE6YP.

ORCID

Andrew J. Russell https://orcid.org/0009-0008-2534-7664
Maurizio Collu https://orcid.org/0000-0001-7692-4988
Philipp R. Thies https://orcid.org/0000-0003-3431-8423

REFERENCES

1. Schlipf D, Simley E, Lemmer F, Pao L, Cheng PW. Collective pitch feedforward control of floating wind turbines using LIDAR. In: 25th ISOPE International Ocean and Polar Engineering Conference. ISOPE; 2015. ISOPE-I-15-755.
2. Navalkar ST, van Wingerden J, Fleming PA, Van Kuik G. Integrating robust LIDAR-based feedforward with feedback control to enhance speed regulation of floating wind turbines. In: 2015 American Control Conference (ACC). IEEE; 2015:3070-3075.
3. Schlipf D, Lemmer F, Raach S. Multi-variable feedforward control for floating wind turbines using LIDAR. In: 30th ISOPE International Ocean and Polar Engineering Conference. ISOPE; 2020. ISOPE-I-20-1174.
4. Larsen TJ, Hanson TD. A method to avoid negative damped low frequent tower vibrations for a floating, pitch controlled wind turbine. *J Phys: Conf Ser*. 2007;75(1):12073.
5. Yu W, Lemmer F, Schlipf D, et al. Evaluation of control methods for floating offshore wind turbines. *J Phys: Conf Ser*. 2018;1104(1):12033.
6. Fischer B. Reducing rotor speed variations of floating wind turbines by compensation of non-minimum phase zeros. *IET Renew Power Gen*. 2013;7(4):413-419.
7. Jonkman J, Butterfield S, Musial W, Scott G. Definition of a 5-MW Reference Wind Turbine for Offshore System Development. Tech. Rep. NREL/TP-500-38060, NREL; 2009.
8. MingYang. Leading innovation: MingYang Smart Energy launches MySE 16.0-242, the world's largest offshore hybrid drive wind turbine. URL: http://www.myse.com.cn/en/jtxw/info.aspx?itemid=825; 2021.
9. Gaertner E, Rinker J, Sethuraman L, et al. Definition of the IEA-Wind 15-Megawatt Offshore Reference Wind Turbine. Tech. Rep. NREL/TP-5000-75698, NREL; 2020.
10. Allen C, Viscelli A, Dagher H, et al. Definition of the UMaine VolturnUS-S Reference Platform Developed for the IEA-Wind 15-Megawatt Offshore Reference Wind Turbine. Tech. Rep. NREL/TP-5000-76773, NREL; 2020.
11. Schlipf D, Guo F, Raach S, Lemmer F. A tutorial on LIDAR-assisted control for floating offshore wind turbines. In: 2023 American Control Conference (ACC). IEEE; 2023:2536-2541.
12. Guo F, Schlipf D. Assessing LIDAR-assisted feedforward and multivariable feedback controls for large floating wind turbines. *Wind Energy Sci Discuss*. 2023;2023:1-25.
13. Bossanyi EA. Individual blade pitch control for load reduction. *Wind Energy*. 2003;6(2):119-128.
14. Namik H, Stol K. Individual blade pitch control of floating offshore wind turbines. *Wind Energy*. 2010;13(1):74-85.
15. Schlipf D, Schuler S, Grau P, Allgöwer F, Kühn M. *Look-Ahead Cyclic Pitch Control Using LIDAR*: University of Stuttgart; 2010.
16. Raach S, Schlipf D, Sandner D, Cheng PW. Nonlinear model predictive control of floating wind turbines with individual pitch control. In: 2014 American control conference (ACC). IEEE; 2014:4434-4439.
17. Dunne F, Pao L, Wright A, Jonkman B, Kelley N. Combining standard feedback controllers with feedforward blade pitch control for load mitigation in wind turbines. In: 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition. American Institute of Aeronautics and Astronautics; 2010:250.
18. Dunne F, Schlipf D, Pao L, et al. Comparison of two independent LIDAR-based pitch control designs. In: 50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition. American Institute of Aeronautics and Astronautics; 2012:1151.
19. NREL. OpenFAST v3.5.0. URL: https://github.com/OpenFAST/openfast/releases/tag/v3.5.0; 2023.
20. Abbas NJ, Zalkind DS, Pao L, Wright A. A reference open-source controller for fixed and floating offshore wind turbines. *Wind Energy Sci*. 2022;7(1):53-73.
21. NREL. OpenFAST v3.4.0. https://github.com/OpenFAST/openfast/tree/v3.4.0; 2022.
22. Schlipf D, Trujillo JJ, Basterra V, Kühn M. *Development of a Wind Turbine LIDAR Simulator*. University of Stuttgart; 2009.
23. Guo F, Schlipf D, Zhu H, Platt A, Cheng PW, Thomas F. Updates on the OpenFAST LIDAR simulator. *J Phys: Conf Ser*. 2022;2265(4):42030.
24. Guo F, Schlipf D, Cheng PW. Evaluation of LIDAR-assisted wind turbine control under various turbulence characteristics. *Wind Energy Sci*. 2023;8(2):149-171.
25. Taylor GI. The spectrum of turbulence. *Proc R Soc London Ser A-Math Phys Sci*. 1938;164(919):476-490.
26. Gaertner E, Rinker J, Sethuraman L, et al. IEA-15-240-RWT. https://github.com/IEAWindTask37/IEA-15-240-RWT; 2022.
27. Magar KT, Balas MJ, Frost S. Direct adaptive control for individual blade pitch control of wind turbines for load reduction. *J Intell Material Syst Struct*. 2015;26(12):1564-1572.
28. Schlipf D. *LIDAR-Assisted Control Concepts for Wind Turbines*. University of Stuttgart; 2016.
29. Held DP, Mann J. LIDAR estimation of rotor-effective wind speed—an experimental comparison. *Wind Energy Sci*. 2019;4(3):421-438.
30. MathWorks. cpsd. https://uk.mathworks.com/help/signal/ref/cpsd.html; 2023.
31. MathWorks. pwelch. https://uk.mathworks.com/help/signal/ref/pwelch.html; 2023.
32. Kelley ND, Jonkman BJ. Overview of the TurbSim Stochastic Inflow Turbulence Simulator. Tech. Rep. NREL/TP-500-36971, NREL; 2005.

10991824, 2024, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.2891 by Test, Wiley Online Library on [10/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

33. Pao LY, Johnson KE. A tutorial on the dynamics and control of wind turbines and wind farms. In: 2009 American Control Conference (ACC). IEEE; 2009:2076-2089.
34. IEC 61400-3-1:2019—Wind Energy Generation Systems—Part 3-1: Design Requirements for Fixed Offshore Wind Turbines, British Standards Institution; 2019. Tech. Rep.

How to cite this article: Russell AJ, Collu M, McDonald AS, Thies PR, Keane A, Quayle AR. LIDAR-assisted feedforward individual pitch control of a 15 MW floating offshore wind turbine. *Wind Energy*. 2024;27(4):341-362. doi:10.1002/we.2891

10991824, 2024, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.2891 by Test, Wiley Online Library on [10/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Check for updates

Wind Energy

WILEY

RESEARCH ARTICLE OPEN ACCESS

The Impact of LIDAR-Assisted Pitch Control on Floating Offshore Wind Operational Expenditure

Andrew J. Russell[1] | Jade McMorland[2] | Maurizio Collu[2] | Alasdair S. McDonald[3] | Philipp R. Thies[4] | Aidan Keane[5] | Alexander R. Quayle[6] | David McMillan[7] | James Carroll[7] | Andrea Coraddu[8]

[1]Industrial Doctoral Centre for Offshore Renewable Energy, School of Engineering, King's Buildings, University of Edinburgh, Edinburgh, UK | [2]Department of Naval Architecture, Ocean and Marine Engineering, University of Strathclyde, Glasgow, UK | [3]School of Engineering, King's Buildings, University of Edinburgh, Edinburgh, UK | [4]Faculty of Environment, Science and Economy Engineering, University of Exeter, Penryn Campus, Penryn, UK | [5]Wood Renewables, Glasgow, UK | [6]Flotation Energy Ltd, Edinburgh, UK | [7]Electronic and Electrical Engineering Department, University of Strathclyde, Glasgow, UK | [8]Faculty of Mechanical, Maritime and Materials Engineering, Delft University of Technology, Delft, The Netherlands

Correspondence: Andrew J. Russell (A.J.Russell-1@sms.ed.ac.uk) | Jade McMorland (jade.mcmorland@strath.ac.uk)

Received: 23 November 2023 | **Revised:** 9 August 2024 | **Accepted:** 16 August 2024

Funding: This work was funded by UK Research and Innovation as part of the EPSRC and NERC Industrial CDT for Offshore Renewable Energy (IDCORE), Grant number EP/S023933/1. The authors would also like to acknowledge EPSRC for funding this work through the Wind and Marine Energy Systems Centre for Doctoral Training under grant number EP/S023801/.

Keywords: failure rates | floating offshore wind | LIDAR-assisted pitch control | O&M | operational expenditure | wind turbine control

ABSTRACT

Floating offshore wind (FOW) is a renewable energy source that is set to play an essential role in addressing climate change and the need for sustainable development. However, due to the increasing threat of climate emergency, more wind turbines are required to be deployed in deep water locations, further offshore. This presents heightened challenges for accessing the turbines and performing maintenance, leading to increased costs. Naturally, methods to reduce operational expenditure (OpEx) are highly desirable. One method that shows potential for reducing OpEx of FOW is LIDAR-assisted pitch control. This approach uses wind velocity measurements from a nacelle-mounted LIDAR to enable feedforward control of floating offshore wind turbines (FOWTs) and can result in reductions to the variations of structural loads. Results obtained from a previous study of combined feedforward collective and individual pitch control (FFCPC + FFIPC) are translated to OpEx reductions via reduced component failure rates for future FOW developments, namely, in locations awarded in the recent ScotWind leasing round. The results indicate that LIDAR-assisted pitch control may allow for an up to 5% reduction in OpEx, increasing to up to 11% with workability constraints included. The results varied across the three ScotWind sites considered, with sites furthest from shore reaping the greatest benefit from LIDAR-assisted control. This work highlights the potential savings and reduction in the overall levelised cost of energy for future offshore wind turbine projects deliverable through the implementation of LIDAR-assisted pitch control.

1 | Introduction

Renewable energy plays a vital role in addressing the serious challenges of climate change and the need for sustainable development. Wind energy, specifically offshore wind, is set to play an essential role in the deployment of clean energy sources globally. Offshore wind has seen significant success in recent years, with 9.4 GW installed worldwide as of 2022 [1]. However, due to the increasing threat of climate emergency, there is a heightened demand for further deployment of wind turbines. At present, bottom-fixed wind (BFW) turbines are limited to a depth of water of 60 m [2]. Wind Europe estimates that around 80% of Europe's available wind resource lies within waters exceeding this depth [3], and FOW provides a viable solution for wind deployment in deeper areas.

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2024 The Author(s). *Wind Energy* published by John Wiley & Sons Ltd.

FOW technology benefits from the knowledge gained through the successful deployment of BFW, presenting a promising path towards commercially viable FOW projects. However, introducing floating systems brings new obstacles and limitations, primarily concerning the operational phase of the project. Key challenges for the technology include wave sensitivity, maintainability, anchor and mooring cost/complexity, and turbine motion, as identified by NREL [4]. With the increasing demand for FOW, deployment sites are being driven to greater offshore distances and depths, with associated harsher environmental conditions and heightened accessibility challenges.

Recently, there has been growing concern about the finance and profitability of future FOW projects. Challenges such as the risk of new technology, increased inflation rates globally, shortage of components/materials/vessels and lack of an established supply chain in remote deployment areas are set to put an additional financial strain on future FOW projects. Therefore, it is vital to reap cost-saving benefits where possible. O&M can account for up to 30% of the total cost of energy for BFW sites, making it a key area for cost reduction. Recent estimates for FOW cost of energy (CoE) indicate that OpEx for FOW projects could potentially exceed 40% due to challenges surrounding environmental conditions, turbine motion and major component replacements [5]. However, there are a number of methods to reduce this impact and to allow effective, efficient and economically feasible O&M campaigns for FOW.

The implementation of LIDAR-assisted pitch control is one method that shows promise for reducing OpEx of FOW sites. This method uses nacelle-mounted LIDAR to gather wind velocity measurements upstream of the turbine, to enable feedforward control of FOWTs. This allows FOWTs to actuate their pitch, torque or yaw control systems in advance of the wind's impact. The greatest benefits delivered by LIDAR-assisted control have been achieved when assisting with blade pitch control in above-rated wind speed conditions [6–9]. One of the key deliverables resulting from LIDAR-assisted pitch control is reductions in the variations of structural loads. This is significant because this gives rise to more consistent loads, which are predictable and uniform across the structure, resulting in reduced fatigue, failure rates and, consequently, lifetime extension, which can lead to reduced maintenance activities and OpEx. This paper aims to use the loading standard deviation reductions obtained from a previous study of LIDAR-assisted pitch control [9] to quantify how they translate to OpEx savings for FOW.

This paper is structured as follows. Section 2 provides a background overview of floating offshore wind and the O&M challenges specific to the technology. Section 2 also briefly overviews the advantages of LIDAR-assisted control within FOW. Section 3 provides a review of the LIDAR-assisted control strategy, implementation, and previous results. Section 4 introduces the O&M methodology for this analysis with details of the vessel fleet, support structure, and case study locations. Section 5 provides the OpEx results using a bottom-fixed wind farm as a baseline comparison. Finally, Section 6 presents conclusions from the results and offers recommendations for future work.

2 | Background

2.1 | Floating Wind Maintenance Overview

It is widely cited that O&M costs can account for up to 30% of the total cost of energy within offshore wind [10]. Yet recent studies have estimated that, for FOW technology, this figure could increase to almost 40% [5], highlighting the need for an effective O&M strategy. NREL identified wave sensitivity, maintainability, anchor cost/complexity, mooring cost/complexity and turbine motion as the key challenges posed by FOWT sites [11].

The deep depths of FOW sites pose challenges for conventional major component replacement vessels [2], which are limited to 60-m depth. This gives rise to the need for alternative methods such as floating-to-floating transfer, floating cranes, self-hoisting equipment and off-site solutions such as tow to shore. The limited availability of appropriate port and grid infrastructure hampers the wider adoption of FOW. Many European ports and harbours lack the necessary capacity to handle the installation and maintenance demands of such assets, particularly in terms of cranage, vessels and personnel, which is a growing concern due to the rapid pace of BFW and FOW site deployment.

As sites move into deeper waters, the average distance to shore is also expected to increase. The average distance to shore (mainland Scotland) of the proposed ScotWind projects exceeds 70 km, with this further increasing for the FOW sites, while the average distance to shore of the current operational offshore wind farms is 18.7 km [12]. The increase in distance to shore is expected to have a significant impact on the accessibility of the sites. Accessibility is determined based on both the weather conditions and the duration of suitable weather windows. Travel time to site and the required weather window increase with distance to shore. The weather window is the duration of access required to safely attend the site, complete the repair and then return to shore. As this duration increases, the access to the site decreases. Rowell et al. [13] compared the accessibility of future ScotWind sites against that of currently operating wind farms (e.g., Moray East). This work used significant wave height (H_s) as the limiting factor, which was kept consistent for both FOW and BFW sites. The results indicated that accessibility for FOW sites will be challenging, particularly for those using a crew transfer vessel (CTV)–based approach.

In addition to the existing access limits, the wave and tidal-induced motion of the turbine could potentially further limit turbine access. Turbine motion is frequently referenced in scholarly works as a major obstacle in FOW O&M procedures [14]. Recent studies have analysed the potential impact of turbine motion within a number of sea states [15–17]. Most of these studies have concentrated on ascertaining turbine motion under severe weather circumstances, aiming to establish the platform survivability and the impact of turbine motion on degradation and fatigue. Nevertheless, the exploration of turbine motion for maintenance-related oceanic conditions (e.g., situations where significant wave heights are below 4 m [18]) remains relatively constrained.

When modelling O&M processes, it is crucial to consider this additional motion in order to ensure the health and safety of

10991824, 2024, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.2951 by Danish Technical Knowledge, Wiley Online Library on [25/03/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

technicians. Assessing the workability of the turbine requires taking into account additional weather and environmental factors such as the peak wave period (T_p) and the direction of the waves [14]. The movement of the turbine is also expected to affect the degradation rate of components, especially within the drivetrain.

2.1.1 | Workability

The concept of workability has been explored within the literature by CoreWind [19] and Scheu et al. [20], identifying the workability index (WI) for several different floating platform designs. Workability is a measure of how safe it is for technicians to perform maintenance on the asset in a number of sea states, which are a combination of H_s, mean wind speed (U) and T_p. Currently, there is no definitive guidance to determine an acceptable WI for offshore wind operations. This introduces a trade-off between site accessibility and technician health and safety.

For one sea state condition, individual WIs from different motion directions (lateral, vertical and rotational) are calculated and multiplied to provide the WI of the specific H_s-T_p combination. The resulting WI is calculated for each H_s-T_p combination of interest. A WI of 1 indicates no impairment of working conditions due to motion, and a value of 0 means that work is not possible under the respective conditions. Time workable conditions are determined based on the specific response amplitude operator (RAO) of the selected offshore wind structure, under a specific set of H_s and T_p values. An acceptable motion threshold is then imposed based on the existing oil and gas threshold from Nordforsk [21]. This can then be normalised to allow for comparisons through the use of a workability index (WI):

$$WI = \frac{\sum T_{wc_i}}{T_t} \tag{1}$$

where T_{wc_i} is the number of workable hours and T_t is the total number of hours.

Scheu et al. [20] provided WI results for a 10-MW wind turbine mounted on four different platform designs (A, B, C, and D). Designs A–D include a spar, tension leg platform (TLP), semi-submersible and a barge design. However, the specific design details of A–D were not disclosed. Workability was found to be a slight limiting factor for sea states with a WI of $\leq 90\%$ and had a significant influence for WI $< 60\%$. The concept of WI for specific platform designs was also explored within the CoreWind project [19]. This project examined the impact of two specific floater designs, ActiveFloat (semi-submersible) and WindCrete (spar). Both platforms were connected to a 15-MW–rated IEA reference turbine [22].

While existing guidance addresses safe human exposure, specific limits for a safe WI in offshore operations are lacking, though Scheu et al. [20] categorised WI impact >90% as slight, 60%–90% as significant and <60% as major.

In terms of operations, workability limits access to site, which in turn increases downtime due to an increase in "waiting on weather" periods. Floating sites are typically in areas with already challenging conditions, therefore the addition of more access limiting factors will have a negative impact on OpEx. One solution to overcome this challenge within the operational phase is to reduce the *number* of waiting on weather periods, by reducing the number of transfers taking place through failure rate reduction mechanisms.

2.2 | LIDAR-Assisted Control

One method that shows promise for reducing OpEx for FOW is LIDAR-assisted control. Nacelle-mounted, forward-facing LIDAR technology can provide measurements of the incoming wind field, the characteristics of which can be used for feedforward turbine control. Traditionally, turbines use feedback control. However, the drawback of this type of control strategy is that the wind turbine provides a delayed reaction to the incoming wind. Feedforward control is able to overcome this drawback by enabling turbines to actuate their control systems in advance of the wind's impact upon the rotor.

The greatest benefits delivered by LIDAR-assisted control have been observed when assisting with blade pitch control in above-rated conditions wind speed [6–8]. LIDAR-assisted feedforward pitch control has demonstrated an ability to enhance turbine performance through improved rotor speed regulation, and therefore more stable power capture. It is also capable of reducing the variation in the structural loadings, motions, and the tensions of the mooring lines of FOWTs [6–8, 23, 24].

Furthermore, wind turbines can utilise individual blade pitch control (IPC), which can mitigate the cyclic loads resulting from variations in the wind profile across the rotor-swept area caused by wind shear, wind veer and turbulence. The concept of feedback IPC (FBIPC) was first explored by Bossanyi et al [25] and applied to floating turbines by Namik and Stol [26]. Alternatively, feedforward individual pitch control (FFIPC) is able to deliver individual pitch commands in advance of the wind's impact and has been studied using various implementations [27–30]. In previous work, the benefits of LIDAR-assisted combined feedforward collective and individual pitch control (FFCPC + FFIPC) for a 15-MW floating offshore wind turbine were investigated [9]. Significant benefits to the turbine performance, loading and motion parameters were observed.

Building on the results of the previous work [9], this paper focuses on translating the reductions in the variations of the structural loads to component failure rate reductions and, subsequently, investigating how these can lead to reduced OpEx. The reductions in OpEx are expected to stem from the decreased lost revenue resulting from reduced downtime of turbines, as well as reduced costs associated with the reduced frequency of maintenance operations, where vessel hire, personnel and material costs are incurred. Furthermore, the reduced failure rates achieved from LIDAR-assisted control may also alleviate the workability access constraints for FOWTs due to the reduced frequency of transfers to site.

10991824, 2024, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.2951 by Danish Technical Knowledge, Wiley Online Library on [25/03/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

2.3 | ScotWind Case Study

In 2022, Crown Estate Scotland launched ScotWind Leasing, aiming to secure 8–10 GW of additional offshore wind capacity, raising the Scottish target to 17–19 GW, mainly by 2030–2032. The 2022 ScotWind leasing results exceeded expectations by awarding 11 floating projects with a combined capacity of approximately 18 GW [31], showcasing the rapid growth of the floating offshore wind industry and Scotland's leading position in this sector. The number of projects is expected to lead to significant competition when securing a strike price through contracts for difference (CfD). In addition, margins are tightened due to inflation and rising costs of commodities. Therefore, any technologies with potential OpEx reduction are of vital importance.

This work uses three of the recent ScotWind allocation sites as the basis of the case study: E1, NE1 and NE3. These sites have been selected due to their spread of location, capacity and distance to shore. This study aims to assess how the cost-saving benefits brought by LIDAR-assisted control differ for the three sites due to these variables. The location of these zones, with details of distance to shore and met-ocean averages, is provided in Table 1.

Further distances from shore require larger weather windows to allow for the longer travel time require to get to and from the site, which results in greater costs for vessel hire and personnel. Furthermore, the weather window restrictions for these far offshore locations amplifies the impacts of turbine downtime, as the waiting on weather periods will be longer and more frequent. Therefore, E1 is expected to benefit most from the implementation of the LIDAR-assisted control strategy because the reduced failure rates will not only decrease the downtime of the turbines, but also the frequency of maintenance visits and therefore the OpEx compared to using the feedback-only control strategy.

TABLE 1 | Case study ScotWind FOW regions with detail of O&M base, distance to shore and average H_s.

ScotWind site	O&M base	Distance to shore, km	Average H_s, m
E1	Aberdeen	120	1.7
NE1	Lerwick	55	1.5
NE3	Wick	55	1.6

3 | Operation and Maintenance Methodology

OpEx is defined as the cost of the operational phase of a project [32]. This is one of the few costs which can be controlled once the FOW farm is operational and is estimated to contribute up to almost 40% of the total cost of energy for future floating wind [5].

The OpEx simulations within this work were performed using the Strath-OW O&M model [33]. This model has previously been validated and used to assist in the development of maintenance campaigns for currently operational offshore wind farms [34]. This model has been adapted to account for workability limits imposed based on the platform design of FOWTs, as discussed in Section 2.1.

The model uses a Monte Carlo time domain simulation approach and consists of four input modules: climate, vessel specifications and fleet configuration, wind farm/turbine and cost and failures. For this analysis, the model has been adapted for FOW day-to-day maintenance procedures, excluding major replacement procedures.

This model comprises of three main parts: climate modelling, turbine failure modelling and resource and cost modelling. Users input sample data, allowing the model to simulate wind speed and significant wave height time series over the defined wind farm lifetime using a multivariate autoregressive model. These values play a crucial role in calculating energy production and losses, as well as determining turbine accessibility for maintenance tasks.

Within the current model [33], access to site is limited by U and H_s, where the thresholds are vessel dependent. Within this adapted model, where workability limits are imposed, the combination of T_p and H_s form the limiting criteria, dependent on the specific platform design. An example of limitations imposed for Design A [20] is shown in Figure 1.

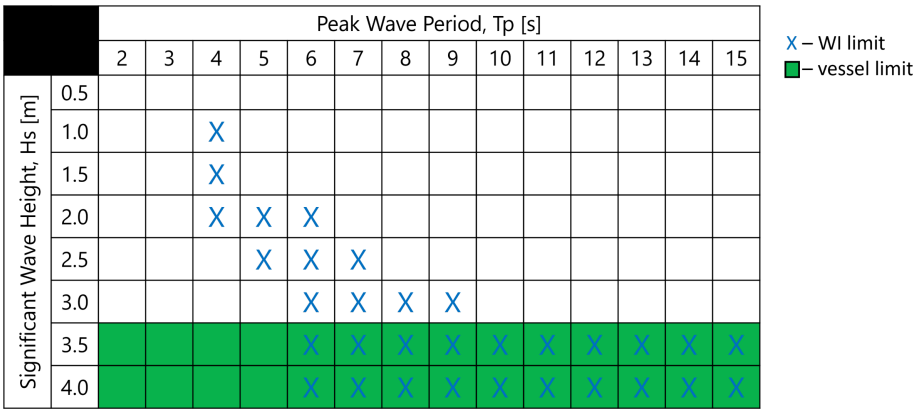


FIGURE 1 | Limitations imposed on accessibility for specific H_s-T_p combinations for Design A [20].

10991824, 2024, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.2951 by Danish Technical Knowledge, Wiley Online Library on [25/03/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

3.1 | Maintenance Strategy

Within offshore wind, maintenance actions can be categorised as minor, major or replacement operations [35, 36]. While there are significant challenges surrounding the major component replacement for FOW turbines, this work focused on the impact of reduced failure rates on minor/major repairs, which make up 90% of total failure rates [35]. Therefore, the use of a heavy lift vessel (HLV) was not included.

It was assumed that the cost of repair of components was identical for both BFW and FOW to allow for direct comparison. Each turbine had a 15-MW rating, based on the NREL 15-MW reference turbine [22]. All baseline failure rates, technicians required, time to repair and cost of components for different turbine subsystems were taken from Carroll et al. [35].

To allow for direct comparison between the ScotWind locations, each site had a capacity of 1 GW and followed the same maintenance strategy. Due to the scale of the sites, and distance to shore, the case study utilised a service operation vessel (SOV)–based maintenance strategy. Each site had a dedicated SOV, which was supported by daughter craft. A set of 6 daughter crafts was chosen to ensure that maintenance activities were not hindered due to resource limitations. The SOV and daughter craft had a H_s limit of 3.5 m and 2 m respectively. Since the goal of this study was to assess potential OpEx reductions via the implementation of LIDAR-assisted control, the optimisation of the vessel fleet was not the focus. Scheduled maintenance was set to 60 h per turbine per year. Maintenance activities were not limited by daylight working hours.

3.1.1 | OpEx Modelling

The adapted O&M model [33] is utilised within this analysis to determine potential OpEx savings of introducing LIDAR-assisted control, particularly within the context of floating wind. OpEx costs are commonly categorised into direct and indirect expenditures. Direct costs encompass fixed expenses, transportation fees, staffing outlays and repair costs. Conversely, indirect costs involve foregone earnings, termed as downtime, constituting a substantial portion of the overall OpEx. Downtime represents an opportunity cost, reflecting the revenue forgone due to asset failure, mirroring potential turbine-generated income [37].

Downtime is dependent on two factors, frequency of failure and accessibility. As the number of failures increase, the number of periods of downtime increase, therefore leading to an increase in lost revenue. However, the duration of downtime can be increased due to poor access to site. Due to safety limitations imposed on operation, turbines can only be repaired in periods of favourable weather.

This is particularly challenging for floating turbines, due to additional workability limitations imposed due to the floating platform but also due to their greater distances from shore making access more restricted by weather windows as personnel have to be offshore for longer periods of time compared to BFW. A reduction in failure rates will have a benefit for both BFW and FOW sites, due to reductions in downtime, in addition to repair costs, transfer costs and fuel costs. However, the most significant benefit is expected for sites with poor accessibility, which will be more common for FOW, where the cost of a single maintenance action is high due to waiting on weather periods. The failure rate reductions brought by LIDAR-assisted control can help to reduce OpEx costs by reducing the frequency of trips required to perform repairs, thereby leading to savings in vessel hire, personnel, fuel and repair costs.

4 | LIDAR-Assisted Control Methodology

The turbine studied within aero-servo-hydro-elastic simulations was the IEA-Wind 15-MW reference turbine, developed between NREL and the Technical University of Denmark (DTU), via the IEA, as defined by Gaertner et al. [22]. The turbine was mounted on the University of Maine's VolturnUS-S semi-submersible [38]. The FOWT was simulated using NREL's open-source Fatigue, Aerodynamics, Structures, and Turbulence (FAST) code. OpenFAST is the latest iteration of the FAST code, and version 3.4 was used for this study, with modifications made to allow for LIDAR configuration simulation within the InflowWind module, which are now present within OpenFAST, from version 3.5 onwards. The LIDAR measurements were interfaced to ServoDyn and sent to the Reference Open Source Controller (ROSCO) [39] where v2.6 was modified to enable LIDAR-assisted control. Full descriptions of the modifications made to OpenFAST and ROSCO were disclosed in previous work [9]. Due to its large rotor diameter (240 m), the turbine is subject to significant variations in structural loads due to differences in the wind profile across the rotor-swept area. In order to mitigate these variations, individual feedforward pitch control was also implemented within the ROSCO controller and worked by providing additional tuning to the pitch of each blade depending on its azimuth position. Full explanation of the feedforward collective and individual pitch control theory can be found within the previous work [9]. The combined feedforward collective and individual pitch controller (FFCPC + FFIPC) was found to deliver significant benefits to the performance, motions and loadings of the FOWT, by way of standard deviation (σ) reductions, compared to the baseline feedback-only (FB) controller, the collective pitch controller (FFCPC) and the FFCPC combined with a traditional feedback individual pitch controller (FFCPC + FBIPC) across the full above-rated wind spectrum, as shown in Figure 2.

From Figure 2, the addition of FFCPC + FFIPC was able to deliver significant benefits to turbine and substructure parameters across the above-rated wind spectrum. Key reductions were observed in the rotor speed and power, as seen in Figure 2a, due to the superior blade pitch preparation in advance of the wind impact. This resulted in up to 56% reductions in their standard deviations compared to the baseline. The superior turbine performance had positive benefits on the bending moments acting on the turbine (Figure 2b), with reductions in the standard deviations of over 15% recorded in the rotor total load and the tower base fore-aft bending moment compared to the baseline controller. The benefits to the

10991824, 2024, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.2951 by Danish Technical Knowledge, Wiley Online Library on [25/03/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

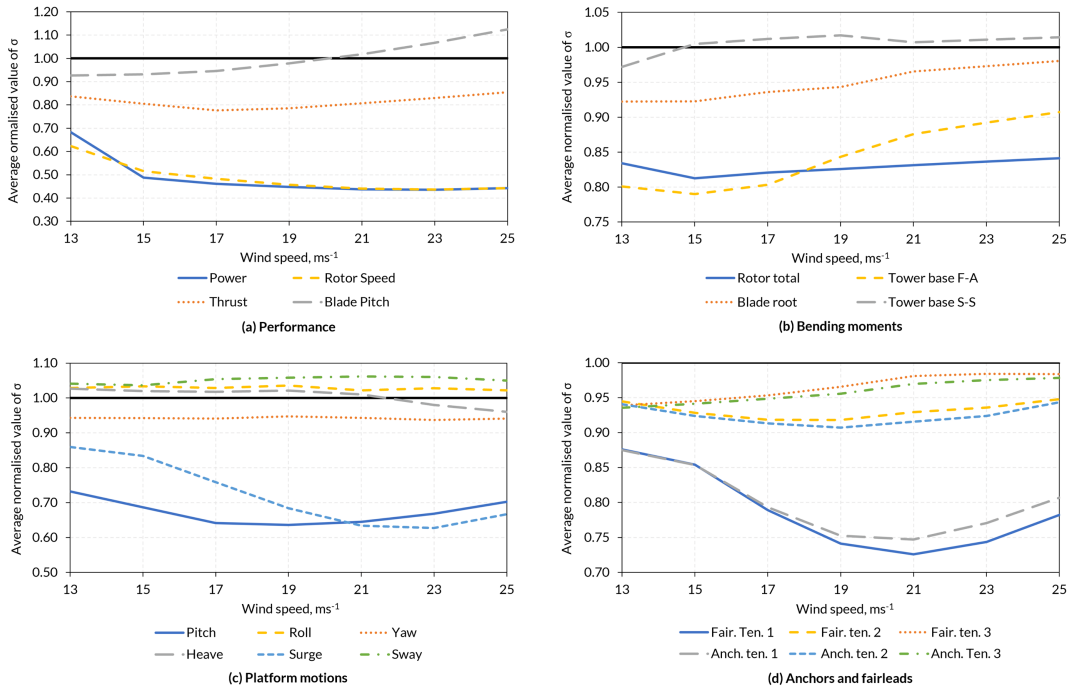


FIGURE 2 | Normalised values of σ for various turbine and substructure parameters when using FFCPC + FFIPC compared to the baseline feedback controller (denoted by black lines at 1.00). Values shown are averaged values from four random seed 1-h simulations at each average wind speed integer.

loadings were particularly relevant to this work, as the standard deviation reductions can lead to reduced failure rates and damage equivalent loads (DELs) of the turbine's components. Moreover, positive benefits were also observed in the platform motions (Figure 2c), namely, in pitch and surge, due to the reductions in the variation in the thrust force acting on the turbine. These benefits also translated to the mooring line tensions at the anchors and fairleads (Figure 2d).

4.1 | Failure Rate Determination

The results of the feedback-only controller and the best-performing feedforward controller examined in the previous work (FFCPC + FFIPC) were post-processed using MLife. MLife is a MatLab-based tool created by NREL to post-process results from wind turbine tests and aero-elastic, dynamic simulations, including from OpenFAST. MLife computes statistical information and fatigue estimates for one or more data files and follows the techniques outlined in the industrial standards [40]. Essential details and equations detailing MLife's operation are provided here. A complete theory manual by Hayman [41] outlining all details of MLife's operation is also available for reference.

MLife works by accumulating fatigue damage due to fluctuating loads over the design life of a wind turbine, which are broken down into individual hysteresis cycles by rainflow counting [41]. The damage is assumed to accumulate linearly with each cycle according to Miner's rule. The total damage from all cycles is given by

$$D = \sum_i \frac{n_i}{N_i\left(L_i^{RF}\right)} \tag{2}$$

where N_i is the number of cycles to failure, n_i is the cycle count and L_i^{RF} is the cycle's load range about a fixed load-mean value. The relationship between the load range and cycles to failure (the S-N curve) is modelled by

$$N_i = \left(\frac{L^{ult} - |L^{MF}|}{\frac{1}{2}\left(L_i^{RF}\right)}\right)^m \tag{3}$$

where L^{ult} is the ultimate design load of a component, L^{MF} is the fixed load-mean and m is the Wöhler exponent, which is specific to the component under consideration. In this study“ an m of 4 was used for steel components and an m of 10 was used for composite components.

As the simulations performed only cover a short period of the design lifetime, the time-series damage-cycle counts must be extrapolated over the design lifetime. To perform these extrapolations of the damage-cycle counts across the lifetime of the wind turbine, MLife requires the wind speed distributions of the sites. This enables MLife to model the wind with a Weibull distribution. The Weibull parameters and average wind speeds inputted to MLife for each of the ScotWind sites analysed are given in Table 2.

The wind speeds were then placed into bins, which were then assigned a probability of occurrence based on the Weibull distribution for each site. Following the lifetime extrapolation, the total damage was determined [41]. As failure occurs when D^{Life} equals one, the time until failure, T^{Fail}, is the ratio between the design lifetime and the accumulated damage.

$$T^{Fail} = \frac{T^{Life}}{D^{Life}} \tag{4}$$

10991824, 2024, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.2951 by Danish Technical Knowledge, Wiley Online Library on [25/03/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 2 | Average wind speeds and Weibull parameters of the ScotWind leasing locations assessed in this study [42].

ScotWind site	Average wind speed, $m\,s^{-1}$	Weibull scale parameter C	Weibull shape parameter K
E1	8.293	9.358	2.360
NE1	8.567	9.667	2.394
NE3	10.10	11.39	2.255

TABLE 3 | Turbine subsystems considered in this study and the OpenFAST outputs associated with them.

Sub-system	OpenFAST output	Description	Units
RNA[a]	RtFldMxh	Total rotor aerodynamic load (moment in x direction)	Nm
Pitch bearings	RootMxc#	Blade in-plane moment at the blade root[b]	kN m
	RootMyc#	Blade out-of-plane moment at the blade root[b]	kN m
Blade roots	RootMxb#	Blade edgewise moment at the blade root[b]	kN m
	RootMyb#	Blade flapwise moment at the blade root[b]	kN m
Tower base	TwrBsMxt	Tower base side-to-side bending moment	kN m
	TwrBsMyt	Tower base fore-aft bending moment	kN m
Moorings	FAIRTEN1	Tension in mooring line 1 at the fairlead	kN
	ANCHTEN1	Tension in mooring line 1 at the anchor	kN

[a]RNA is assumed to represent the rotor blades, hub and pitch system.
[b]# is the blade number from 1 to 3.

The failure rate, λ, is the inverse of the time until failure.

$$\lambda = \frac{1}{T^{Fail}} \tag{5}$$

Within the analysis performed in this study, the time series outputs for numerous loadings associated with different turbine components were inputted to MLife and the time to failure associated with each loading on a component was determined. The loading with the lowest time to failure was assumed to be the time to failure for the component.

4.2 | Assessed Parameters

OpenFAST allows for the output of time-series data of numerous loadings applied to the turbine's various subsystems. These include the rotor blades, tower, shaft, bedplate and pitch bearings. The data acquired can be processed through MLife, which can return the damage, time to failure and DEL occurring as a result of a particular load applied on the subsystem. As the OpenFAST load outputs are defined at various cross sections of the turbine, this process works well for structures composed of simple cylindrical shapes and isotropic materials such as the tower and shaft, as S-N curves are able to deliver a good indication of the time to failure at the cross sections of the components. Conversely, for non-cylindrical components, MLife is unable to accurately predict their times to failure as the S-N curves cannot be applied to these components. A detailed finite element analysis, with the load outputs from OpenFAST as an input, is required to determine the resulting state of stress for these components. However, for the purposes of this study, the total aerodynamic load-induced bending moments on the rotor-nacelle assembly (RNA) was assumed to represent the forces acting upon the rotor blades, hub and pitch system, as defined by Dao et al. [43]. The approximately 15% reductions to the standard deviation of these loads (Figure 2) were, therefore, used to estimate the reductions to the times to failure of the components within the RNA, which will be used within the O&M model.

Table 3 gives the OpenFAST outputs that represent the loadings applied to the relevant sub-system. Time-series data of these outputs obtained from average turbulent wind speeds of 11–25 $m\,s^{-1}$ in intervals of 2 $m\,s^{-1}$ were inputted to MLife for the turbine utilising the FB or FFCPC + FFIPC controllers. For the FFCPC + FFIPC, the same data as was used for the FB when average wind speeds below 11 $m\,s^{-1}$ were simulated as this was below the rated wind speed (10.69 $m\,s^{-1}$) and where the feedforward controller did not operate.

At each average wind speed integer from 3 to 25 $m\,s^{-1}$, four random seed 1-h simulations were performed. The turbulence was generated under the normal turbulence model (NTM), turbulence intensity (TI) category B using TurbSim [44]. These conditions fall into design load case (DLC) 1.2, as defined in the industrial standards [40]. DLC 1.2 was used as this embodies the loads resulting from turbulence that occurs during the normal operation of a wind turbine throughout its lifetime. The floating turbine was modelled in a water depth of 200 m and under irregular waves. The wind and waves were aligned

10991824, 2024, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.2951 by Danish Technical Knowledge, Wiley Online Library on [25/03/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

in the same direction. Though normally a joint probability distribution of wind and waves is considered within DLC 1.2 for lifetime fatigue estimates, this was not performed within this study because the OpEx reduction results presented in Section 5 focus solely on the loading reductions experienced by the rotor, which is assumed to be aligned with the wind direction at all times. However, a joint wind and wave probability distribution should be considered when further investigating the benefits to the tower and moorings. H_s and T_p were adjusted to suit each average wind speed integer, with values interpolated from those defined by Allen et al [38], and shown in Table 4.

When analysing the time-to-failure data for the subsystems with multiple relevant loading outputs outlined in Table 3, namely, the pitch bearings, the blade roots and the tower base, a first-to-fail approach was taken such that the loading with the lowest time-to-failure was used for comparison between the FB and FFCPC + FFIPC control strategies.

5 | Results and Discussion

Within this section, the reduction of failure rates through LIDAR implementation is discussed. The impact of these reductions is then quantified through simulation of OpEx over a lifetime of a given site.

5.1 | LIDAR-Assisted Control Failure Rate Reductions

The results of the MLife post-processing using the data from the four random seed 1-h simulations are shown in Table 5. Results are presented as normalised values of failure rate compared to the baseline feedback-only controller.

The loadings resulting in the lowest time-to-failure of subsystems in Table 5 occurred in the fore-aft direction due to the alignment of the FOWT with the co-directional wind and waves. Table 5 indicates that the FFCPC + FFIPC strategy was able to deliver significant reductions to the failure rates of turbine subsystems at all of the investigated sites. The reductions in the failure rate of the tower base may not be significant for O&M costs due to their relatively low failure rate, but indicate potential for structure optimisation, whereby a thinner, lighter tower may be possible. The results also highlighted the promising potential to enable failure rate reductions of the pitch bearing, the blade roots and the mooring system. While it was found within the previous work that the standard deviation of the blade pitch increased at higher above-rated wind speeds [9], as shown in Figure 2, this is believed to have been offset by the reduced standard deviations of the blade pitch at lower above-rated wind speeds, which represent a much greater proportion of the probability distribution of wind speeds than at higher wind speeds, ultimately resulting in the reduced failure rates of the pitch bearing. However, it is worth noting that the failure rate reduction reported here is solely based on the loadings imparted on the components and so does not account for failures arising from mechanical wear of the pitch system from control actuation.

For the purpose of this study, the most significant result was that of the rotor total load leading to failures in the RNA. For each of the sites, an approximately 20% reduction in the failure rate resulting from the rotor total load was recorded. The results also indicated marginally greater reductions in the failure rate at site NE3 than at sites NE1 and E1. This was due to its slightly higher average wind speed at the site (as shown in Table 2), meaning

TABLE 4 | IEC DLC 1.2 NTM wind and wave conditions [38].

U_{hub}, m s^{-1}	3	5	7	9	11	13	15	17	19	21	23	25
H_s, m	1.10	1.14	1.25	1.43	1.69	2.01	2.39	2.83	3.34	3.82	4.27	4.80
T_p, s	8.52	8.41	8.16	7.83	7.55	7.45	7.55	7.85	8.28	8.75	9.22	9.70

TABLE 5 | Normalised values of failure rates for turbine subsystems resulting from the loads recorded by the OpenFAST simulation outputs when using FFCPC + FFIPC relative to the baseline FB controller at each ScotWind site.

		Normalised failure rate			
Subsystem	**OpenFAST output**	**Baseline**	**E1**	**NE1**	**NE3**
RNA[a]	RtFldMxh	1.00	0.81	0.80	0.79
Pitch bearings	RootMyc#[b]	1.00	0.56	0.55	0.54
Blade roots	RootMyb#[b]	1.00	0.57	0.57	0.57
Tower base	TwrBsMyt	1.00	0.73	0.72	0.70
Moorings	FAIRTEN1	1.00	0.92	0.91	0.90
	ANCHTEN1	1.00	0.92	0.91	0.90

Note: Outputs correspond to the loadings with the lowest time-to-failure for each subsystem when using the baseline FB controller, hence the omission of the other outputs present in Table 3.
[a]RNA is assumed to represent the rotor blades, hub and pitch system.
[b]# is the blade number from 1 to 3. Values are averages.

10991824, 2024, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.2951 by Danish Technical Knowledge, Wiley Online Library on [25/03/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

that more of its lifetime will be spent with the feedforward controller operating in above-rated wind speeds where the loading reductions can be achieved.

5.2 | Baseline Comparison

As discussed in Section 5.1, the LIDAR simulations saw a reduction in degradation and failure rates for the RNA, pitch bearings, blade roots, tower base and mooring system. Within the literature surrounding offshore wind turbine failure rates, there is a variety of nomenclature used, making it difficult to draw comparison between the results. In the review by Dao et al. [43] the definition of RNA is made up of the subassemblies of blades, hub, air brake and pitch system. These elements were matched to the failure rates for the subassemblies detailed in the work of Carroll et al. [35]. For all FOW simulations, the failure rates for the RNA only are adjusted based on the normalised values presented in Table 5. The same failure rate reductions were assumed to apply to all wind turbines within the array. All other failure rates are that of the BFW equivalent site [35].

OpEx encompasses all expenses incurred from the moment of takeover, comprising both one-time and recurring costs associated with the wind farm, measured annually. These costs are split as direct and indirect costs. Direct costs are those which have a "direct" impact on the operation of the site finances such as cost of repair, transport, staff and other fixed costs such as insurance. Indirect costs, such as lost production costs are more difficult to predict, due to uncertainty in the prediction of unscheduled failures and metocean conditions. Lost revenue is viewed within industry as an "opportunity cost". An opportunity cost is defined as the revenue which could have been generated, had the turbine been operational.

The reduction in total OpEx, and the subcosts for a FOW farm with the reduced failure rates obtained from the implementation of LIDAR-assisted control are given in Table 6. This analysis is based on vessel limits only using the existing Strathclyde O&M model [33].

The study's outcomes indicate a notable 3%–5% reduction in operational expenditures, displaying variability among different sites. Particularly, NE3 emerges as the most advantageous because this exhibiting the most substantial decrease in RNA failure rates compared to the other locations (Table 5) due to its higher average wind speeds, meaning that LIDAR-assisted control is active for a greater proportion of its lifetime. The overall decline in OpEx is attributed to lowered lost revenue, due to reduced downtime, and repair expenses. The key OpEx costs savings result from the potential for transport costs to be reduced by decreasing the number of vessels (and associated hiring costs) due to the reduced required transfers resulting from the lifetime extension of the components. The relationship between the reduction of failure rates, and therefore number of transfers required, and the OpEx decrease is highlighted in Figure 3. This illustrates that as the reduction to the failure rate achieved from LIDAR-assisted pitch control increases, the OpEx savings increase because fewer transfers to the wind farm site are required during the lifetime, thereby reducing costs of vessel hire and personnel. The reduced failure rates also mean that components require repair less frequently, resulting in the repair cost savings shown in Table 6.

TABLE 6 | OpEx savings breakdown based on reduction of failure rates.

	E1	NE1	NE3
Lost revenue	5.38%	8.58%	1.70%
Repair costs	7.10%	10.3%	11.0%
Total OpEx	2.89%	4.29%	5.05%

These results are representative of the impact of the reduced failure rates on the WindCrete and ActiveFloat [19] designs, where workability would not impact operation using an SOV strategy with a H_s limit of 3 m.

5.3 | Workability Impact

As previously discussed in Section 2.1.1, FOW turbines are also subject to additional weather constraints due to human exposure to the motion of the asset. As shown in Figure 1, workability limitations are unique to each platform design and limits access to the turbine during specific combinations of T_p and H_s. Using designs A-D from Scheu et al. [20], the workability limitations are imposed on access to the site, in addition to the existing vessel limits. As previously discussed, CoreWind [19] also provides WI limitations for the ActiveFloat semi-sub and WindCrete spar designs. However, these limitations only exist for WI combinations of 3.5 m and above, which are already captured within the vessel limits; therefore, the OpEx results will be unchanged from those presented in Table 6. The methodology detailing the adaptations can be found in previous work [45].

Within this analysis, the WI limitations are placed on both the FOW baseline comparison and the FOW with reduced failure rates due to the LIDAR-assisted control implementation. It is assumed that any H_s-T_p combinations resulting in WIs less than 1 are inaccessible. The estimated OpEx savings are shown in Figure 4.

Unlike the results with no consideration of WI, the highest OpEx savings is not consistent with the site with the highest

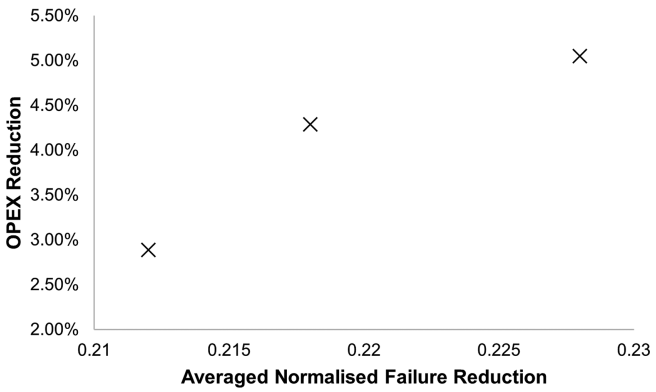


FIGURE 3 | Relationship between failure rate reduction of the RNA and OpEx savings.

10991824, 2024, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.2951 by Danish Technical Knowledge, Wiley Online Library on [25/03/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

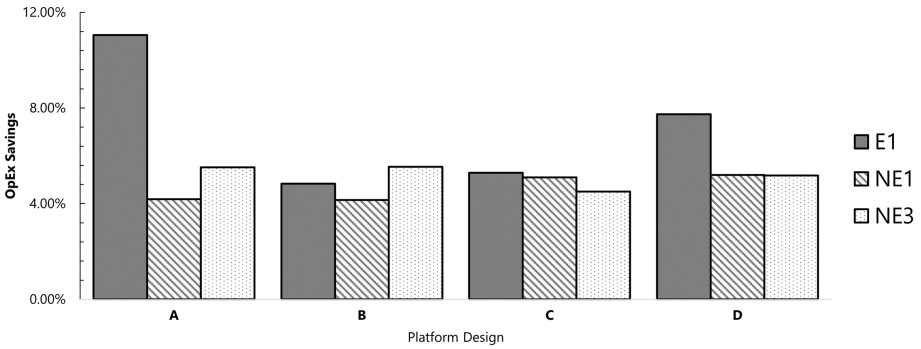


FIGURE 4 | OpEx savings resulting from reduced failure rates for FOW sites with imposed WI limitations.

TABLE 7 | Percentage savings in OpEx from FOW with WI limits and LIDAR-assisted control enabled failure rate reductions compared to the equivalent BFW baseline using feedback-only control, where "—" indicates that the BFW was cheaper than the FOW equivalent site.

ScotWind site	Design A	Design B	Design C	Design D
NE1	—	—	1.89%	—
NE3	4.41%	4.97%	4.57%	0.39%

Note: FOW was not more cost-effective at E1 for any of the designs.

reduction in failure rates. Site E1 has the highest OpEx savings for all platform designs.

Site E1 experiences the highest OpEx primarily because it undergoes the most significant reduction in accessibility resulting from the incorporation of workability limits. As a result, it reaps the most substantial advantage from the decrease in failure rates. This advantage translates into a reduction in the necessary transfers and subsequently minimises the demand for optimal weather conditions. This highlights that sites with accessibility challenges, and therefore predicted high OpEx, can see the highest benefit from the addition of LIDAR-assisted control.

The variability of the results for the 4 designs highlights the unique challenges of each specific design, and that there is no "one-fits-all solution." As with the comparison in Table 6, for LIDAR-assisted control with no WI considerations, the decrease in OpEx is attributed to the reduction of repair costs and lost revenue. However, for all sites and all designs, the savings in lost revenue was the main contributor to these savings, due to the challenges of accessibility and the link between access and OpEx [46].

When comparing the FOW with both LIDAR-assisted control and motion limitations against the BFW equivalent with feedback-only control (i.e. without LIDAR-assisted control) for specific designs, in some cases, the total OpEx for motion limited FOW with LIDAR-assisted control delivered failure reductions was less than that of the BFW with original failure rates and no motion limits. The percentage savings in OpEx for NE1 and NE3 are detailed in Table 7.

Site NE3 saw consistent savings across all designs, whereas NE1 was only more cost-effective than BFW equivalent for Design C. E1 was not found to be more cost-effective for any of the designs due to the additional workability restrictions of the FOW site, which resulted in larger costs than the BFW equivalent despite including the failure rate reductions brought by the LIDAR-assisted control.

The results show that while FOW the sites face additional challenges due to location and motion, the implementation of specific technologies and control strategies, such as LIDAR-assisted control, has the potential to bring certain aspects of the cost of some FOW sites below that of the equivalent BFW site using feedback-only control.

6 | Conclusions

This work has highlighted the potential OpEx benefits from the introduction of a LIDAR control strategy for future FOW developments. While implementing a LIDAR system incurs initial capital costs, its potential for substantial OpEx savings presents can justify the initial expenditure. In addition to the OpEx saving, there is the additional benefit of the reduction in the number of transfers within a health and safety context. Reducing the number of transfers, in turn, will reduce the total exposure to motion for technicians on FOW turbines, and therefore have a well-being impact. The reduction in transfers also allows for the vessel fleet to be optimised and potentially reduced, resulting in savings in transportation costs and in vessel emissions.

The key findings from this work can be summarised as follows:

- The LIDAR-assisted pitch control simulations performed within OpenFAST indicated the ability to deliver reductions in the failure rate of components in the rotor nacelle assembly by 20%, the tower base by up to 30% and the pitch bearings and blade roots by 46%. These translated to baseline OpEx savings of up to 5%.

10991824, 2024, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.2951 by Danish Technical Knowledge, Wiley Online Library on [25/03/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

- FOW sites most impacted by the addition of WI limits saw the largest OpEx savings from LIDAR-assisted control integration due to the reduction in downtime and lost revenue.
- For specific designs and sites, the overall OpEx for FOW sites with motion limits and LIDAR-assisted control were more cost-effective than the BFW equivalent site using feedback-only control.

Within this work, the focus has been surrounding the OpEx savings of the reduction of failures for the RNA for minor and major repairs using an SOV vessel supported by a fleet of daughter craft. However, as highlighted, the moorings, blade root, pitch bearing and tower also experience a reduction in fatigue due to the implementation of LIDAR-assisted pitch control, and therefore, the potential OpEx savings highlighted here could be increased.

While minor/major repairs can account for up to 90% of the total failure rates, the cost associated with major component replacements makes up a significant portion of the total OpEx for the full site lifecycle. It is expected that major component replacement for FOW has the potential to be highly costly due to the time involved in a tow to shore operation, or the cost of specialist vessels for in situ maintenance. Therefore, the reduction in failures for these maintenance events could have significant operational savings.

To summarise, the integration of LIDAR-assisted control in floating wind turbine projects emerges as a promising solution to mitigate failure rates and subsequently drive down OpEx. By accurately assessing environmental conditions and adjusting turbine operations in real-time, LIDAR technology offers a proactive approach to enhancing overall system reliability. The quantification of OpEx reduction resulting from decreased failure rates highlights the potential savings, and reduction in the overall levelised cost of energy for future FOW developments.

The results presented assume 100% LIDAR availability and therefore should be considered to be the best case scenario for the benefits delivered by this LIDAR-assisted control approach. Further work should seek to perform sensitivity analysis to investigate the impact of reduced LIDAR availability and LIDAR-assisted pitch control benefit on the failure rates and cost savings. Further work may also seek to address the potential for structural optimisation of the turbine and substructure through the implementation of LIDAR-assisted control, as well as how nacelle-mounted LIDAR wind measurement can help in understanding the loadings on FOWTs in their inactive (feathered blades) state.

Author Contributions

Andrew Russell performed the modifications to the OpenFAST and ROSCO source codes, executed the OpenFAST simulations, MLife post-processing and compiled the LIDAR-assisted control related aspects of the paper. **Jade McMorland** performed the O&M modelling and compiled the O&M related aspects of the paper. **Maurizio Collu, Alasdair McDonald, Philipp Thies, Aidan Keane, Alexander Quayle, David McMillan, James Carroll** and **Andrea Corradu** provided supervision, suggestions and reviewed the paper.

Acknowledgements

This work was funded by UK Research and Innovation as part of the EPSRC and NERC Industrial CDT for Offshore Renewable Energy (IDCORE), Grant number EP/S023933/1. The authors would also like to acknowledge EPSRC for funding this work through the Wind and Marine Energy Systems Centre for Doctoral Training under grant number EP/S023801/.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The original modified version of OpenFAST v.3.4, which includes the LIDAR simulator used in this work can be accessed via https://github.com/Russell9798/OpenFAST-v3.4-Lidar-IfW-Original/releases/tag/2.0. The modified version of ROSCO v2.6 with the feedforward control additions can be accessed via https://github.com/Russell9798/ROSCO-v2.6-LAC-IPC/releases/tag/2.0.

LIDAR-assisted control simulation datasets used for this paper are available and can be assessed via https://doi.org/10.17605/OSF.IO/FE6YP.

Peer Review

The peer review history for this article is available at https://www.webofscience.com/api/gateway/wos/peer-review/10.1002/we.2951.

References

1. World Forum Offshore Wind, "Global Offshore Wind Report," (2022), https://wfo-global.org/wp-content/uploads/2023/03/WFO_Global-Offshore-Wind-Report-2022.pdf.

2. D. Ahn, S. Shin, S. Kim, H. Kharoufi, and H. Kim, "Comparative Evaluation of Different Offshore Wind Turbine Installation Vessels for Korean West-South Wind Farm," *International Journal of Naval Architecture and Ocean Engineering* 9, no. 1 (2017): 45–54.

3. Wind Europe, "Floating Wind Vision Statement," https://windeurope.org/wp-content/uploads/files/about-wind/reports/Floating-offshore-statement.pdf.

4. S. Butterfield, W. Musial, J. Jonkman, and P. Sclavounos, "Engineering Challenges for Floating Offshore Wind Turbines," tech. rep. National Renewable Energy Lab.(NREL), Golden, CO (United States, 2007).

5. BVG, "Guide to a Floating Offshore Wind Farm," https://guidetofloatingoffshorewind.com/wind-farm-costs/.

6. D. Schlipf, E. Simley, F. Lemmer, L. Pao, and P. W. Cheng, "Collective Pitch Feedforward Control of Floating Wind Turbines Using LIDAR," in *25th ISOPE International Ocean and Polar Engineering Conference* (2015), ISOPE–I–15–755.

7. S. T. Navalkar, J. van Wingerden, P. A. Fleming, and G. Van Kuik, "Integrating Robust LIDAR-Based Feedforward With Feedback Control to Enhance Speed Regulation of Floating Wind Turbines," in *2015 American Control Conference (ACC)* (IEEE, 2015), 3070–3075.

8. D. Schlipf, F. Lemmer, and S. Raach, "Multi-Variable Feedforward Control for Floating Wind Turbines Using LIDAR," in *30th ISOPE International Ocean and Polar Engineering Conference* (2020), ISOPE–I–20–1174.

9. A. J. Russell, M. Collu, A. S. McDonald, P. R. Thies, A. Keane, and A. R. Quayle, "Lidar-Assisted Feedforward Individual Pitch Control of a 15 MW Floating Offshore Wind Turbine," *Wind Energy* 27, no. 4 (2024): 341–362.

10991824, 2024, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.2951 by Danish Technical Knowledge, Wiley Online Library on [25/03/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

10. Y. Feng, P. J. Tavner, and H. Long, "Early Experiences With UK Round 1 Offshore Wind Farms," *Proceedings of the Institution of Civil Engineers-energy* 163, no. 4 (2010): 167–181.

11. W. Musial, S. Butterfield, and A. Boone, "Feasibility of Floating Platform Systems for Wind Turbines," in *42nd AIAA Aerospace Sciences Meeting and Exhibit*, (2004): 1007.

12. H. Díaz and C. G. Soares, "Review of the Current Status, Technology and Future Trends of Offshore Wind Farms," *Ocean Engineering* 209 (2020): 107381.

13. D. Rowell, B. Jenkins, J. Carroll, and D. McMillan, "How Does the Accessibility of Floating Wind Farm Sites Compare to Existing Fixed Bottom Sites?," *Energies* 15, no. 23 (2022): 8946.

14. B. Jenkins, A. Prothero, M. Collu, J. Carroll, D. McMillan, and A. McDonald, "Limiting Wave Conditions for the Safe Maintenance of Floating Wind Turbines," in *Journal of Physics: Conference Series*, Vol. 2018, (IOP Publishing, 2021): 12023.

15. C. M. Wang, T. Utsunomiya, S. C. Wee, and Y. S. Choo, "Research on Floating Wind Turbines: A Literature Survey," *IES Journal Part A: Civil & Structural Engineering* 3, no. 4 (2010): 267–277.

16. S. Bashetty and S. Ozcelik, "Review on Dynamics of Offshore Floating Wind Turbine Platforms," *Energies* 14, no. 19 (2021): 6026.

17. A. Otter, J. Murphy, V. Pakrashi, A. Robertson, and C. Desmond, "A Review of Modelling Techniques for Floating Offshore Wind Turbines," *Wind Energy* 25, no. 5 (2022): 831–857.

18. I. B. Sperstad, M. Stålhane, I. Dinwoodie, O.-E. V. Endrerud, R. Martin, and E. Warner, "Testing the Robustness of Optimal Access Vessel Fleet Selection for Operation and Maintenance of Offshore Wind Farms," *Ocean Engineering* 145 (2017): 334–343.

19. CoreWind, "Floating Wind O&M Strategies Assessment," (2020), https://corewind.eu/wp-content/uploads/files/publications/COREWIND-D4.2-Floating-Wind-O-and-M-Strategies-Assessment.pdf.

20. M. N. Scheu, D. Kaufer, M.-A. Schwarzkopf, and A. Kolios, "Workability on Offshore Wind Turbines—A Comparative Study of Fixed-Bottom and Floating Applications," in *ISOPE International Ocean and Polar Engineering Conference* (ISOPE, 2018), ISOPE–I.

21. I. R. Nielsen, "Assessment of Ship Performance in a Seaway," (1987). Publisher: NORDFORSK, Sortedam Dossering 19, DK-200 Copenhage, Denmark, ISBN: 87-982637-1-4.

22. E. Gaertner, J. Rinker, L. Sethuraman, et al., "Definition of the IEA-Wind 15-Megawatt Offshore Reference Wind Turbine," Tech. Rep. NREL/TP-5000-75698. (NREL, Golden, CO, 2020).

23. D. Schlipf, F. Guo, S. Raach, and F. Lemmer, "A Tutorial on LIDAR-Assisted Control for Floating Offshore Wind Turbines," in *2023 American Control Conference (ACC)*, (2023): 2536–2541.

24. F. Guo and D. Schlipf, "Assessing LIDAR-Assisted Feedforward and Multivariable Feedback Controls for Large Floating Wind Turbines," *Wind Energy Science Discussions* 2023 (2023): 1–25.

25. E. A. Bossanyi, "Individual Blade Pitch Control for Load Reduction," *Wind Energy* 6, no. 2 (2003): 119–128.

26. H. Namik and K. Stol, "Individual Blade Pitch Control of Floating Offshore Wind Turbines," *Wind Energy* 13, no. 1 (2010): 74–85.

27. D. Schlipf, S. Schuler, P. Grau, F. Allgöwer, and M. Kühn, *Look-Ahead Cyclic Pitch Control Using LIDAR* (University of Stuttgart, 2010).

28. S. Raach, D. Schlipf, D. Sandner, and P. W. Cheng, "Nonlinear Model Predictive Control of Floating Wind Turbines With Individual Pitch Control," in *2014 American control conference (ACC)*, (2014): 4434–4439.

29. F. Dunne, L. Pao, A. Wright, B. Jonkman, and N. Kelley, "Combining Standard Feedback Controllers With Feedforward Blade Pitch Control for Load Mtigation in Wind Turbines," in *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, (2010): 250.

30. F. Dunne, D. Schlipf, L. Pao, et al., "Comparison of Two Independent LIDAR-Based Pitch Control Designs," in *50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, (2012): 1151.

31. Crown Estate Scotland, "Awards: Lead Applicants, Project Partners, Area, Capacity and Foundations," (2022), https://www.crownestatescotland.com/resources/documents/scotwind-list-of-successful-project-partners.

32. A. Ioannou, A. Angus, and F. Brennan, "Parametric Capex, Opex, and Lcoe Expressions for Offshore Wind Farms Based on Global Deployment Parameters," *Energy Sources, Part B: Economics, Planning, and Policy* 13, no. 5 (2018): 281–290.

33. I. Dinwoodie, F. Quail, and D. McMillan, "Analysis of Offshore Wind Turbine Operation and Maintenance Using a Novel Time Domain Meteo-Ocean Modeling Approach," *Turbo Expo: Power for Land, Sea, and Air*, Vol. 44724 (American Society of Mechanical Engineers, 2012), 847–857.

34. I. Dinwoodie, O.-E. V. Endrerud, M. Hofmann, R. Martin, and I. B. Sperstad, "Reference Cases for Verification of Operation and Maintenance Simulation Models for Offshore Wind Farms," *Wind Engineering* 39, no. 1 (2015): 1–14.

35. J. Carroll, A. McDonald, and D. McMillan, "Failure Rate, Repair Time and Unscheduled O&M Cost Analysis of Offshore Wind Turbines," *Wind Energy* 19, no. 6 (2016): 1107–1119.

36. ReliaWind, "Reliability Focused Research on Optimizing Wind Energy Systems Design, Operation and Maintenance: Tools, Proof of Concepts, Guidelines & Methodologies for a New Generation," (2011), https://cordis.europa.eu/project/id/212966/reporting.

37. J. McMorland, M. Collu, D. McMillan, and J. Carroll, "Operation and Maintenance for Floating Wind Turbines: A Review," *Renewable and Sustainable Energy Reviews* 163 (2022): 112499.

38. C. Allen, A. Viscelli, H. Dagher, et al., "Definition of the UMaine VolturnUS-S Reference Platform Developed for the IEA-Wind 15-Megawatt Offshore Reference Wind Turbine," NREL, Golden, CO (US, 2020).

39. N. J. Abbas, D. S. Zalkind, L. Pao, and A. Wright, "A Reference Open-Source Controller for Fixed and Floating Offshore Wind Turbines," *Wind Energy Science* 7, no. 1 (2022): 53–73.

40. "IEC 61400-3-1:2019 - Wind Energy Generation Systems–Part 3-1: Design Requirements for Fixed Offshore Wind Turbines," Tech. Rep. (British Standards Institution, London, UK, 2019).

41. G. J. Hayman, "Mlife Theory Manual for Version 1.00," Tech. Rep. NREL, Golden, CO (US, 2012).

42. LAUTEC ESOX, "ERA5 Data Point Map," https://esox.lautec.com/map/.

43. C. Dao, B. Kazemtabrizi, and C. Crabtree, "Wind Turbine Reliability Data Review and Impacts on Levelised Cost of Energy," *Wind Energy* 22, no. 12 (2019): 1848–1871.

44. N. D. Kelley and B. J. Jonkman, "Overview of the TurbSim Stochastic Inflow Turbulence Simulator," Tech. Rep. NREL/TP-500-36971. NREL, Golden, CO (US, 2005).

45. J. McMorland, M. Collu, D. McMillan, and A. Corradu, "The Impact of Limiting Sea States on Floating Offshore Wind Operation and Maintenance," *Applied Oceans Research [under review]* (2023).

46. G. J. W. Van Bussel and W. A. A. M. Bierbooms, "The DOWEC Offshore Reference Windfarm: Analysis of Transportation for Operation and Maintenance," *Wind Engineering* 27, no. 5 (2003): 381–391.

10991824, 2024, 11, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.2951 by Danish Technical Knowledge, Wiley Online Library on [25/03/2025]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License