

Electronic Thesis and Dissertation Repository

2-28-2022 8:45 AM

A computational fluid dynamics-based surrogate wind turbine blade aerodynamic model for hybrid simulation

Eric Lalonde, *The University of Western Ontario*

Supervisor: Bitsuamlak, Girma T., *The University of Western Ontario*

Joint Supervisor: Dai, Kaoshan, *Sichuan University*

Co-Supervisor: Lu, Wensheng, *Tongji University*

A thesis submitted in partial fulfillment of the requirements for the Doctor of Philosophy degree in Civil and Environmental Engineering

© Eric Lalonde 2022

Follow this and additional works at: <https://ir.lib.uwo.ca/etd>



Part of the [Aerodynamics and Fluid Mechanics Commons](#), [Dynamics and Dynamical Systems Commons](#), and the [Structural Engineering Commons](#)

Recommended Citation

Lalonde, Eric, "A computational fluid dynamics-based surrogate wind turbine blade aerodynamic model for hybrid simulation" (2022). *Electronic Thesis and Dissertation Repository*. 8431. <https://ir.lib.uwo.ca/etd/8431>

This Dissertation/Thesis is brought to you for free and open access by Scholarship@Western. It has been accepted for inclusion in Electronic Thesis and Dissertation Repository by an authorized administrator of Scholarship@Western. For more information, please contact wlsadmin@uwo.ca.

Abstract

Hybrid simulation (HS) is a promising technique for studying wind turbines due to the presence of scaling errors in wind tunnel testing. However, HS of wind-loaded structures is limited by the current practice of using lower-accuracy, “pre-calculated” aerodynamic loads, which uncouple the aerodynamic loading from the structural response. This thesis presents six stand-alone studies that collectively build towards a novel HS framework that employs a computational fluid dynamics (CFD) based surrogate model to generate higher-accuracy aerodynamic loads within the HS loop. An experimentally-validated residential wind turbine model equipped with an external damping system was used to illustrate the proposed framework.

Development of the proposed HS framework occurred in the following stages: firstly, the limitations of the existing HS framework were identified and quantified through a numerical case study of an industrial wind turbine. The aerodynamic surrogate model was developed through CFD simulations of airfoils to determine optimal test parameters, followed by the identification of an optimal convolutional neural network (CNN) architecture to act as the surrogate model. The test case for the HS was chosen based on a numerical study of semi-active tuned mass damper (STMD) systems for reducing turbine tower vibrations. A dynamic numerical model was developed of a 1.1 m residential wind turbine on a rotating base validated through a set of full-scale aeroelastic wind tunnel tests.

These components were ultimately combined for a series of twelve artificial HS to compare the effectiveness of the proposed surrogate model-based HS framework to the existing “pre-calculated” technique. A number of CFD simulations were performed to generate the training data for the CNN surrogate model, which was combined with the dynamic turbine model to act as the numerical HS substructure. This was paired with the optimized STMD acting as the virtual experimental substructure, including simulated equipment delays and measurement errors. The results of these HS indicate that the proposed framework has improved aerodynamic accuracy and aeroelastic fidelity compared to the existing technique, though it faces hurdles from computationally costly CFD data generation. This framework offers a promising tool for future HS of wind turbines and other wind engineering applications.

Keywords: Real-time hybrid simulation, semi-active damping systems, wind turbine tower vibrations, neural network-based surrogate model, full-scale aeroelastic wind tunnel testing, computational fluid dynamics

Summary for Lay Audience

Hybrid simulation (HS) is a structural engineering research technique that tests a structure as two components: a physical component in a laboratory and a simulated component on a computer. This allows complex, hard-to-simulate structural elements to be tested without having to recreate the entire structure. Wind turbines (WT) are a promising option for this technique because the standard method for testing wind-sensitive structures – wind tunnel testing – struggles to accurately capture the resulting wind forces, creating the need for an alternative physical testing method.

Unfortunately, existing HS of WTs have been limited. Since the HS components run in tandem with each other, the program hosting the simulated component must run quickly – delays in the program will cause de-syncing with the physical component and lead to inaccurate results. Thus, in an HS of a WT, the simulated component must be able to quickly predict the wind forces. Various methods exist to predict wind forces, however, they generally prioritize either speed or accuracy, not both. Therefore, HS of WTs have historically been limited to low-accuracy wind force predictions, reducing the technique’s utility.

This thesis explores a method to incorporate an improved simulated component into HS, using a tool called a *neural network* (NN) to act as a *surrogate model*. NNs are artificial intelligences that are trained to predict outputs based on inputs without performing the underlying calculations, acting as a high-speed surrogate to other models. *Computational fluid dynamics* (CFD) is a time-consuming method to simulate high-accuracy wind forces; by training a NN on CFD data, a program for predicting wind forces that is both high-speed and high-accuracy can be created and incorporated into HS of WTs.

This thesis presents six papers that detail the investigation and development of this surrogate model, culminating in a series of HS of a WT equipped with a vibration-suppressing damping system to investigate the effectiveness of the model. Ultimately, the upgraded HS technique presented here is identified as a promising option for effectively studying wind-loaded structures which warrants further investigation.

Co-Authorship Statement

This thesis has been prepared in accordance with the regulations for an Integrated Article format thesis stipulated by the School of Graduate and Postdoctoral Studies at the University of Western Ontario.

Eric Lalonde is the primary researcher and author of the studies presented in each chapter. Dr. Girma Bitsuamlak and Dr. Kaoshan Dai were responsible for supervision and editing of all research; the research presented in Chapters 2 and 5 was primarily supervised by Dr. Kaoshan Dai, while the research presented in Chapters 3, 4, 6, and 7 was primarily supervised by Dr. Girma Bitsuamlak. Dr. Wensheng Lu provided editing to Chapters 2 and 5 and has graciously acted as interim supervisor at Tongji University since Dr. Dai began his current position at Sichuan University.

Additional co-authorship for individual chapters are as follow: Benjamin Vischschraper was responsible for the initial research concerning MLP and LSTM networks in Chapter 4, as well as editing. Zhi Zhao was responsible for generating the loading data employed in Chapter 5, as well as editing.

Chapters 2, 4, and 5 of this thesis summarize published research as described in their summaries. The research presented in the other chapters has yet to be published as of the time of writing.

Acknowledgements

I would like to express my appreciation and sincere gratitude to my research supervisors, Dr. Girma Bitsuamlak and Dr. Kaoshan Dai. Kaoshan's generosity, support, and drive was instrumental to my development as a research and growth as an individual during my time at Tongji University. Girma's infectious positivity and eagerness has been life-changing; without his support I would not be where I am today nor would I have achieved all that I have.

I would also like to thank my friends and colleagues that have helped me throughout this journey. Thank you to Zhi and Ben for your collaboration as researchers, as well as your friendship and support. Thank you to Tristan Cormier for the support and aid during testing at the WindEEE Research Institute. I would also like to thank the huge number of people that have provided advice and helped me with editing including Anant, Dillon, Evaline, Gazia, Nick, Paul, Teddy, and Tibebu, as well as the rest of the research group.

I want to give special thanks to my parents, Cindy and Marc, my sister, Alex, and the rest of my family whose support and encouragement let me survive through the homesickness while abroad and the late nights since coming home.

Most importantly, I dedicate this thesis to my partner, Rayna, for the care and patience she has shown me throughout our time together. It was her love that gave me the courage to begin this journey and the strength to finish it. Thank you Rayna for being my rock, for following me around the world and back again, and for the years of joy that we have shared as well as those yet to come.

Contents

Abstract	i
Summary for Lay Audience	ii
Co-Authorship Statement	iii
Acknowledgements	iv
List of Figures	x
List of Tables	xiii
List of Appendices	xvi
List of Acronyms and Abbreviations	xvii
1 Introduction	1
1.1 Motivation	2
1.2 Objectives	3
1.3 Structure	4
1.3.1 Chapter 2: Wind turbines and hybrid simulation: an overview of re- search, opportunities and limitations	4
1.3.2 Chapter 3: Investigation of the drag and lift coefficient curves of the NACA 0012 airfoil using CFD and a novel synthesis method	6
1.3.3 Chapter 4: Optimizing the type and architecture of neural network- based surrogate aeroelastic wind turbine blade models	6
1.3.4 Chapter 5: A comparison of passive and semi-active tuned mass damper systems for wind turbines	6
1.3.5 Chapter 6: Development of a rotating-base residential wind turbine model based on aeroelastic wind tunnel tests	7
1.3.6 Chapter 7: Development of a novel hybrid simulation framework em- ploying a CFD-based surrogate model to study a semi-actively damped wind turbine	7
2 Wind turbines and hybrid simulation: an overview of research, opportunities and limitations	8
2.1 Introduction	9

2.2	Wind turbine parts, types and frames of reference	10
2.3	Numerical characterisation of wind turbines	12
2.3.1	Actuator disc model and momentum theory	13
2.3.2	Airfoil lift and drag	16
2.4	Traditional numerical turbine studies	17
2.4.1	Finite element modelling	18
2.4.2	Blade element momentum theory	18
2.4.3	Computational fluid dynamics	19
2.4.4	Summary of numerical tests	20
2.5	Traditional experimental turbine studies	23
2.5.1	Wind tunnel experiments	23
2.5.1.1	Airfoil scaling errors in wind tunnel testing	24
2.5.2	Shaking table experiments	25
2.5.3	Summary of experimental tests	26
2.6	Overview of hybrid simulation	26
2.6.1	Development and numerical explanation	28
2.6.2	Control in hybrid simulation	30
2.6.3	Example hybrid simulation applications	31
2.7	Hybrid simulation of wind turbines	32
2.8	A study on the limitations of HS of wind turbines	33
2.8.1	Methodology	34
2.8.2	Testing and results	37
2.8.3	Study conclusions	41
2.9	Overall conclusions	42
3	Investigation of the drag and lift coefficient curves of the NACA 0012 airfoil using CFD and a novel synthesis method	45
3.1	Introduction	46
3.2	Synthesis of previous studies of the NACA 0012 airfoil	47
3.2.1	Overview	48
3.2.2	Collection of previous studies	50
3.2.3	Average 3D lift coefficient curves	53
3.2.4	Average 2D lift coefficient curves	58
3.2.5	Average 3D drag coefficient curves	59
3.2.6	Average 2D drag coefficient curves	59
3.3	CFD simulations	63
3.3.1	Overview	63
3.3.2	Mesh specifications and sensitivity analysis	64
3.3.3	2D simulation results	68
3.3.4	3D simulation results	72
3.4	Conclusions	75
4	Optimizing the type and architecture of neural network-based surrogate aeroelastic wind turbine blade models	77
4.1	Introduction	78

4.2	Neural Network Theory and Application	79
4.2.1	Previous Applications	79
4.2.2	Neural network architecture	81
4.2.3	Training process	82
4.2.4	Multilayer perceptron (MLP) networks	83
4.2.5	Long short-term memory (LSTM) networks	84
4.2.6	Convolutional neural networks (CNNs)	84
4.3	Methodology	85
4.3.1	OpenFAST simulations	86
4.3.2	Data pre-processing	87
4.3.3	Training Procedure	88
4.4	Time-independent aerodynamic models	89
4.4.1	Input and output data	90
4.4.2	Network layers	90
4.4.3	Optimized hyperparameters	91
4.4.4	Results	91
4.5	Time-dependent aerodynamic model: LSTM	94
4.5.1	LSTM specifications	95
4.5.2	LSTM results	96
4.5.3	Pseudo-time-dependent MLP	97
4.6	Time-dependent aerodynamic model: CNN	98
4.6.1	CNN specifications	98
4.6.2	CNN results	99
4.6.3	Multi-time-step MLP	101
4.7	Conclusions	102
5	A comparison of passive and semi-active tuned mass damper systems for wind turbines	104
5.1	Introduction	105
5.2	Methodology	107
5.2.1	Turbine model	107
5.2.2	Applied load cases	111
5.2.2.1	Wind loading	112
5.2.2.2	Seismic loading	113
5.2.3	TMD models	114
5.2.3.1	Summary of test cases	114
5.2.3.2	Passive damping component	115
5.2.3.3	Passive stiffness component	116
5.2.3.4	Semi-active damping component via MR damper	116
5.2.3.5	Semi-active stiffness component via SAIVS device	118
5.2.3.6	Parameter optimization	118
5.2.4	Numerical algorithm	119
5.3	Results of TMD comparisons	121
5.4	Conclusions and recommendations	125

6	Development of a rotating-base residential wind turbine model based on aeroelastic wind tunnel tests	126
6.1	Introduction	127
6.2	Wind tunnel testing	128
6.2.1	Residential wind turbine specifications	128
6.2.2	Aeroelastic base design and construction	129
6.2.3	Testing facility and data acquisition	131
6.2.4	Wind tunnel testing: first round	133
6.2.5	Wind tunnel testing: second round	137
6.2.6	Conclusions and post-mortem	139
6.3	Dynamic structural model	139
6.3.1	Analysis of free vibration tests	139
6.3.2	Numerical blade approximation	141
6.3.3	Surrogate thrust coefficient model	146
6.3.4	1-DOF dynamic structural model	148
6.3.5	Finalization and validation of the structural model under fixed and non-fluctuating conditions	149
6.3.6	Damping model and dynamic validation	154
6.3.7	Summary of the 1-DOF structural model	161
6.4	Case study employing the structural model	161
6.5	Conclusions	164
7	Development of a novel hybrid simulation framework employing a CFD-based surrogate model to study a semi-actively damped wind turbine	165
7.1	Introduction	166
7.2	Methodology: structural model, TMD model and pre-calculated aerodynamic loads	167
7.2.1	Wind speed and tip speed ratio combinations	168
7.2.2	Aerodynamic models and HS frameworks	169
7.2.3	OpenFAST aerodynamic load time histories	171
7.2.4	MR-controlled semi-active tuned mass damper design	173
7.3	Methodology: CFD-based surrogate aerodynamic model	176
7.3.1	CFD fundamentals	176
7.3.2	Turbine blade model	177
7.3.3	CFD validation	178
7.3.4	Final CFD mesh	181
7.3.5	CFD results	184
7.3.6	Neural network specifications	191
7.3.7	Neural network optimization and training	193
7.4	Artificial hybrid simulations	196
7.4.1	Description	196
7.4.2	Results	198
7.4.3	Improvements offered by the proposed framework	204
7.4.4	Limitations in the proposed framework	206
7.5	Conclusions	208

8	Conclusions and recommendations	209
8.1	Summaries of the conclusions of each chapter	210
8.1.1	Chapter 2: Wind turbines and hybrid simulation: an overview of re- search, opportunities and limitations	210
8.1.2	Chapter 3: Investigation of the drag and lift coefficient curves of the NACA 0012 airfoil using CFD and a novel synthesis method	210
8.1.3	Chapter 4: Optimizing the type and architecture of neural network- based surrogate aeroelastic wind turbine blade models	211
8.1.4	Chapter 5: A comparison of passive and semi-active tuned mass damper systems for wind turbines	212
8.1.5	Chapter 6: Development of a rotating-base residential wind turbine model based on aeroelastic wind tunnel tests	213
8.1.6	Chapter 7: Development of a novel hybrid simulation framework em- ploying a CFD-based surrogate model to study a semi-actively damped wind turbine	213
8.2	Observations supporting the effectiveness of the proposed framework	214
8.3	Future research directions	216
	Bibliography	219
	A CFD data set sample for NN development	241
	B MATLAB code for CNN optimization and training	249
	C MATLAB code for artificial hybrid simulations	261
	Curriculum Vitae	272

List of Figures

1.1	Classification of chapters into four relevant research areas	5
2.1	Parts of a horizontal-axis wind turbine	11
2.2	Different types of turbines with possible bases and modelling methods	12
2.3	Standard frames of reference for structural analysis of wind turbines	13
2.4	Actuator disk model and the locations of the three cross-sections of interest	14
2.5	Max C_p vs TSR from Hau (2006)	15
2.6	Airfoil parameters and loads	16
2.7	Cd and Cl plots from Sheldahl and Klimas (1981)	17
2.8	Cd plots for NACA 0012 at different Re	24
2.9	Theoretical pseudo dynamic test of a water tower	29
2.10	Theoretical substructured HS of a braced frame	30
2.11	Comparison of ideal and real HS frameworks	34
2.12	Calculation process flowchart	38
2.13	Response time history segment of tower top under 15 m/s winds	39
3.1	NACA 0012 airfoil with a unit chord length	46
3.2	C_D on a cylinder as a function of Re	47
3.3	Airfoil behaviour patterns based on Re and AoA from Wu et al. (1998)	48
3.4	Maximum lift coefficient versus Re from McCroskey (1987)	49
3.5	Lift coefficient curves from sources detailed in Table 3.1	52
3.6	Drag coefficient curves from sources detailed in Table 3.1	52
3.7	Definition of the 12 traits to characterize the 3D lift coefficient curves	53
3.8	Predicted 3D lift coefficient curves at a range of Reynolds numbers	57
3.9	Definition of the 9 traits to characterize the 2D lift coefficient curves	60
3.10	Predicted 2D lift coefficient curves at a range of Reynolds numbers	60
3.11	Definition of the 9 traits to characterize the 3D drag coefficient curves	61
3.12	Predicted 3D drag coefficient curves at a range of Reynolds numbers	61
3.13	Definition of the 8 traits to characterize the 2D drag coefficient curves	62
3.14	Predicted 2D drag coefficient curves at a range of Reynolds numbers	62
3.15	A summary of the testing mesh, used in both 2D and 3D testing	65
3.16	Results of mesh sensitivity analysis	68
3.17	Flow behaviour around the NACA 0012 airfoil from 2D CFD simulations	70
3.18	Lift and drag coefficient curves predicted by the 2D CFD simulations	71
3.19	Flow behaviour around the NACA 0012 airfoil from 3D CFD simulations	73
3.20	Lift and drag coefficient curves predicted by the 3D CFD simulations	74

4.1	Generic NN architecture with two fully-connected hidden layers	82
4.2	Flowchart of the optimization and training process of a neural network	83
4.3	Layout of an LSTM NN with one LSTM layer and one fully-connected layer	84
4.4	CNN network architecture	85
4.5	Recorded wind turbine coordinates systems and data	89
4.6	MLP layer architecture and sizes	91
4.7	NRMSE of the a) Full or b) Reduced MLPs	92
4.8	Scatter plot of 10% of the normalized Full MLP results	93
4.9	Comparison of wind load time histories	94
4.10	LSTM layer architecture and sizes	95
4.11	NMRSE of the LSTM	96
4.12	CNN layer architecture and sizes	99
4.13	NRMSE of the CNN	100
5.1	Simplified models of TMDs connected to main structure	106
5.2	32-DOF FEM turbine model	109
5.3	Turbine tower mode shapes and the resulting TMD locations	110
5.4	PSD of blade load time histories under SW and HIW conditions	112
5.5	Mean seismic record response spectra	113
5.6	Approximating TMD restoring forces as F_D and F_S	114
5.7	Modified Bouc-Wen model: a numerical approximation of a MR damper	117
5.8	Flowchart of the stiffness component (F_S) control process	119
5.9	Numerical integration algorithm for a simplified 1-DOF system	120
5.10	Displacement time histories from EQ load case	121
6.1	Photo of the 1.1 m residential wind turbine	129
6.2	Side-view of proposed aeroelastic turbine response	130
6.3	Photos and schematic of the below-tunnel portion of the aeroelastic base	130
6.4	Comparison of open profile recorded at WindEEE	131
6.5	Photograph of aeroelastic turbine setup for wind tunnel testing	132
6.6	Schematic of setup of the rotating-base turbine in the wind tunnel	133
6.7	Frequency content of the recorded wind speed and tower rotation	135
6.8	Internal view of pneumatic actuator	136
6.9	Frequency content of the recorded tower rotation	138
6.10	Numerical model of spring system in rotating base turbine	140
6.11	Normalized airfoil profiles	141
6.12	Lift and drag coefficient curves	143
6.13	Lift and drag coefficient curves of the entire blade	144
6.14	Relationship between the angle of attack and TSR at the blade tip	145
6.15	Plots of the thrust coefficient at various TSRs and wind speeds	146
6.16	Diagram of the architecture of the surrogate thrust MLP NN	147
6.17	Contour plot of the thrust coefficient predicted by the NN	147
6.18	Simplified 1-DOF model of the rotating base turbine	149
6.19	Free vibration response and linear decay curve	155
6.20	Free vibration response and four decay curves	158

6.21	1-DOF structural model with STMD	162
7.1	Three wind speed time histories generated using TurbSim	169
7.2	The four AHS frameworks	170
7.3	Comparison of the pre-calculated aerodynamic rotor loads	172
7.4	3D rotor model preparation stages	178
7.5	Mesh properties of single-blade CFD simulations	179
7.6	Flow and vorticity snapshot of the single-blade CFD sims	180
7.7	Predicted drag and lift coefficient curves for the single-blade CFD sims	181
7.8	Mesh properties of full-rotor CFD simulations	183
7.9	Explanation of changing angle of wind in CFD simulations	185
7.10	Wind speed node locations	185
7.11	Full rotor CFD results	187
7.12	CNN architecture	192
7.13	CFD data range	195
7.14	Process used to predict aerodynamic thrust using the surrogate model	198
7.15	Pre-calculated and surrogate model-generated rotor thrust time histories	200
7.16	Base rotation time histories from AHS	202
7.17	Scatter plots comparing CFD data and AHS results	207
A.1	Blade identification	242

List of Tables

2.1	A collection of numerical wind turbine studies	21
2.2	A collection of experimental wind turbine studies	27
2.3	TMD properties	36
2.4	Tower FE properties	37
2.5	Other FE model properties	37
2.6	Nacelle response from OpenFAST-only simulations	40
2.7	Nacelle response from simulations of undamped wind turbine	41
2.8	Nacelle response from simulations of damped wind turbine	41
3.1	Aerodynamic studies of the NACA 0012 airfoil in Re flows of 10^5 to 10^6	51
3.2	Collection of recorded 3D drag and lift coefficient curves	51
3.3	Collected traits that define 3D lift coefficient curve	54
3.4	Twelve curve traits of the 3D lift coefficient curves	55
3.5	Boundary conditions of the 3D lift coefficient curves	56
3.6	Collected traits that define 2D lift coefficient curves	60
3.7	Boundary conditions of the 2D lift coefficient curves	60
3.8	Collected traits that define 3D drag coefficient curves	61
3.9	Boundary conditions of the 3D drag coefficient curves	61
3.10	Collected traits that define 2D drag coefficient curves	62
3.11	Boundary conditions of the 2D drag coefficient curves	62
3.12	Summary of parameters for the mesh sensitivity analysis	65
3.13	Summary of mesh sensitivity analysis results at angles of attack of 8° and 17°	67
3.14	2D CFD simulation results	69
3.15	3D CFD simulation results	74
4.1	Rotor speed and blade pitch specifications	86
4.2	Nodes along the turbine blade where data was recorded	87
4.3	Recorded data from OpenFAST simulations	88
4.4	Input and output data groups for the Full and Reduced MLPs	90
4.5	Hyperparameter optimization results for time-independent MLPs	91
4.6	Optimization, training and run times for the Full and Reduced MLPs	94
4.7	Input and output data groups for the LSTM	95
4.8	Hyperparameter optimization results for the LSTM	96
4.9	Optimization, training and run times for the LSTM	97
4.10	Input and output data groups for the Pseudo-time-dependent MLP	97
4.11	Hyperparameter optimization results for the Pseudo-time-dependent MLP	98

4.12	Input and output data groups for the CNN	99
4.13	Hyperparameter optimization results for the CNN	100
4.14	Optimization, training and run times for the CNN	100
4.15	Hyperparameter optimization results for the Multi-time-step MLP	101
4.16	Summary of the six NNs	102
5.1	Summary of previous STMD WT studies	107
5.2	Turbine tower properties used in FEM model	110
5.3	Natural frequency comparisons	110
5.4	List of applied load cases	111
5.5	Summary of IEC wind field parameters for the 1.5 MW wind turbine	112
5.6	Selected scaled ground motion records	113
5.7	Summary of different TMDs	115
5.8	Optimized parameters for the 3% modal mass ratio TMDs	119
5.9	List of CR integration parameters of the STMDs	120
5.10	Average % improvement compared to uncontrolled turbine under EQ loading .	123
5.11	Average % improvement compared to uncontrolled turbine under HIW loading	123
5.12	Average % improvement compared to uncontrolled turbine under SW loading .	123
5.13	Optimized parameters for the 1.5% modal mass ratio TMDs	124
5.14	Average % improvement compared to uncontrolled turbine under EQ loading .	124
5.15	Comparison of 3% and 1.5% mass ratio TMDs	124
6.1	Test cases studied in the first round of wind tunnel testing	134
6.2	Results of the first round of wind tunnel testing	134
6.3	Predicted damping ratio from simplified Bernoulli model	137
6.4	Results of the second round of wind tunnel testing	138
6.5	Airfoils in the numerical approximation of the wind turbine blade	142
6.6	Comparison of the first round of wind tunnel tests and the numerical model . .	151
6.7	Difference in rotations of wind tunnel and numerical tests	152
6.8	Comparison of the second round of wind tunnel tests and the numerical model .	153
6.9	Calculation of aerodynamic damping for the 1.1 m residential wind turbine . . .	156
6.10	Calculation of aerodynamic damping for the NREL 5 MW wind turbine	156
6.11	Optimal linear damping ratios	158
6.12	Optimal linear, quadratic, and Coulomb damping properties	160
6.13	Summary of 1-DOF dynamic structural model	161
6.14	Summary of TMD vs STMD simulations	163
7.1	Wind speed, turbulence intensity, and TSR testing combinations	168
7.2	Comparison of thrust time histories predicted with OpenFAST	172
7.3	Optimal passive TMD properties	174
7.4	Modified Bouc-Wen model properties	175
7.5	CFD domain scaling factors	177
7.6	Mesh sizes used for single-blade CFD simulations	179
7.7	Recorded drag and lift values from single-blade CFD simulations	180
7.8	Final mesh dimensions	183

7.9	Eight wind angles tested in CFD simulations	184
7.10	Mean CFD simulation results	189
7.11	Comparison of mean thrust and torque coefficients	190
7.12	CNN inputs and outputs	193
7.13	Hyperparameter optimization results	194
7.14	Trained neural network accuracy	194
7.15	The twelve artificial HS test cases	196
7.16	Comparison of two surrogate models	199
7.17	Mean results of the twelve AHS simulations	201
7.18	Summary of nine simulations of the residential turbine without STMD	204
A.1	Columns 1-12 of processed CFD data for the NN	243
A.2	Columns 13-24 of processed CFD data for the NN	244
A.3	Columns 25-36 of processed CFD data for the NN	245
A.4	Columns 37-48 of processed CFD data for the NN	246
A.5	Columns 49-60 of processed CFD data for the NN	247
A.6	Columns 61-64 of processed CFD data for the NN	248

List of Appendices

Appendix A: Full CFD data set for NN development	241
Appendix B: MATLAB code for CNN optimization and training	249
Appendix C: MATLAB code for artificial hybrid simulations	261

List of Acronyms and Abbreviations

The following table identifies acronyms, and abbreviations used in this document.

Acronym	Meaning	Explanation
AHS	Artificial hybrid simulation	HS that simulates the experimental substructure numerically
AR	Artificial	Refers to artificial harmonic or pulse loading condition
ATMD	Active TMD	TMD that uses actuators to add additional force during operation
BEM	Blade element momentum	FE technique used to determine aerodynamic loads on a turbine blade
CFD	Computational fluid dynamics	Technique for modelling fluid flow
CNN	Convolutional NN	NN that uses convolution and pooling to draw out optimal data, often used for image processing
CR	Chen-Ricles	Numerical algorithm for solving the EOM of a structure
CWE	Computational wind engineering	Method that derives wind loads from numerical simulations
DES	Detached eddy simulation	CFD turbulence model that combines RANS and LES
DOF	Degree of freedom	A possible response of a numerical model
EOM	Equation of motion	Differential equation describing the dynamic behaviour of an object
EQ	Earthquake	Refers to the seismic loading condition
FA	Fore-aft	Along-wind response of a turbine in nacelle reference frame
FAST	Fatigue, aerodynamics, structures, and turbulence	Previous name of OpenFAST, a wind turbine analysis software by NREL
FE	Finite element	Technique used to model complex behaviour as a group of simplified elements
FLX	Fully-flexible	Blade aeroelasticity level used in Chapter 2

Acronym	Meaning	Explanation
FSI	Fluid-structure interaction	Studies that combine dynamic FE structural models with CFD
HIW	High-intensity wind	Extreme wind speeds where a wind turbine would be parked
HS	Hybrid simulation	Testing technique that combines experimental and numerical techniques
LES	Large eddy simulation	CFD turbulence model that filters small-scale turbulence for true time-history analysis
LQR	Linear-quadratic regulator	Optimized feedback controller for dynamic systems
LSTM	Long short-term memory	NN with built in memory, for time-history analysis
MAPE	Mean absolute percent error	Error quantification equation
MBW	Modified Bouc Wen	Hysteresis model for MR dampers
MGH	Modified ground-hook	Simple control technique for damping component of an STMD
MLP	Multi-layer perceptron	Vanilla NN that connects inputs to outputs via fully-connected layers of nodes
MR	Magnetorheological	A fluid whose viscosity changes when exposed to a magnetic field
MTMD	Multiple TMD	A group of TMDs tuned to different frequencies to increase robustness
NACA	National Advisory Committee for Aeronautics	NASA predecessor, source of airfoil designs
NN	Neural network	Learning computation technique that can characterize systems without modelling the underlying behaviour
NND	No numerical damping	Pre-calculated wind load condition in Chapter 2
NREL	National Renewable Energy Laboratories	American research laboratory that has authored research tools such as FAST and standard wind turbines specifications
NRMSE	Normalized RMSE	Normalized modification to the RMSE equation
PDT	Pseudo dynamic testing	Form of HS where experimental structure is loaded very slowly
PND	Perfect numerical damping	Pre-calculated wind load condition in Chapter 2
PTMD	Passive TMD	TMD with fixed properties tuned to a single vibration frequency
RANS	Reynolds-averaged Navier-Stokes	Time-averaged CFD turbulence model
RGD	Rigid	Blade aeroelasticity level used in Chapter 2
RMSE	Root mean square error	Error quantification equation

Acronym	Meaning	Explanation
RTHS	Real-time HS	HS performed in real time allowing for velocity- and acceleration-based behaviour to be captured experimentally
SAIVS	Semi-active and independently variable stiffness	Configuration of four springs and an actuator that creates a semi-active stiffness element
SDA	Small-deflection assumed	Blade aeroelasticity level used in Chapter 2
SS	Side-to-side	Across-wind response of a turbine in nacelle reference frame
SSI	Soil-structure interaction	Flexible behaviour of a structure's foundation
STMD	Semi-active TMD	TMD that can modify its properties in real time
SW	Service wind	Operational wind speeds of a wind turbine
TLD	Tuned liquid damper	Damper systems consisting of a tank filled with water
TMD	Tuned mass damper	Additional flexible mass added to structure to reduce vibrations
TSR	Tip speed ratio	Ratio of the tip speed of a rotating turbine blade to the upwind wind speed
URANS	Unsteady RANS	Applied version of RANS to approximate a time-history analysis
VAL	Validation	Pre-calculated wind load condition in Chapter 2
WT	Wind turbine	Structure used to convert kinetic energy in the wind to electrical energy via rotating blades
WV	Wave	Refers to wave loading condition

Chapter 1

Introduction

With the ongoing threat of climate change driving the need for renewable energy sources, wind turbines have seen a marked increase in adoption around the world. According to the latest reports (Lee and Zhao, 2021), 2020 saw a remarkable 14.3% increase in the global cumulative wind power capacity, including a 1.2% increase in Canadian capacity to 13.6 GW and a 22.0% increase in Chinese capacity to 288.3 GW. While these increases in adoptions are undoubtedly a positive trend, current wind turbine designs continue to be limited to a relatively short service life due to fatigue loading, and are at risk of failure in high-intensity wind events. Future engineering research could aid in pushing back against these limitations, and thus should be facilitated as much as possible.

A promising technique for such research is hybrid simulation (HS): a joint experimental-numerical testing technique. HS excels at facilitating experimental-level results in highly non-linear test cases such as complex external damping systems or member failure. Wind turbines in particular are a promising candidate because, unlike most structures, they experience scaling-based errors in small-scale wind tunnel tests, meaning that an alternative experimental-level testing technique is particularly valuable. However, current HS frameworks face limitations when studying wind-loaded structures, and thus an improved HS framework that incorporates advanced computational wind engineering (CWE) methods is needed to truly maximize the technique's potential for studying wind-loaded structures, particularly for wind turbines.

This thesis presents a collection of six research papers that each present unique, stand-alone research while collectively building towards an proposed, novel HS framework that incorporates a surrogate model for real-time, high-accuracy, in-the-loop aerodynamic calculations. These chapters define the need for the proposed framework and the components required to investigate it, including the development of both the aerodynamic thrust model using neural networks (NN) trained on computational fluid dynamics (CFD) data, an appropriate test case for the HS in the form of a semi-active tuned mass damper, and an aeroelastic structural model trained on wind tunnel data. Finally, artificial HS are performed using the proposed framework to investigate its efficacy. This chapter goes into more detail of the motivations of this research, the objectives, and the structure of this thesis including a description of each of the six research papers.

1.1 Motivation

The motivation for this research is fully detailed in Chapter 2, but a summarized version is presented here. The need for a new hybrid simulation framework that includes improved aerodynamic modelling for studying wind turbines can be identified via four observations:

1. Hybrid simulation is able to improve upon existing types of studies as well as facilitate new ones. By combining experimental and numerical models into a single simultaneous, real-time test, the scope and scale can be improved compared to traditional experimental tests while achieving improved accuracy compared to traditional numerical tests. This technique is particularly effective in highly non-linear scenarios where the accuracy of numerical testing would be particularly low, such as complex external dampers or member failure. The main limitation of HS is the requirement for testing to be in real-time, meaning that any calculations in the numerical side of the test must be able to be run extremely quickly.
2. There are unique challenges faced when implementing HS for wind-loaded structures, due to the fact that high-accuracy aerodynamic load calculation is a very time-intensive process. While simplified aerodynamic models such as the standard drag equation are very simple and quick to compute, they have been shown to be notably lower in accuracy compared to intensive CWE techniques such as CFD. Since running these intensive CWE techniques is significantly time-consuming, they are too slow to be paired directly into a real-time HS. As a result, existing HS of wind-loaded structures often use lower-accuracy aerodynamic models which are used to “pre-calculate” the wind loading time history prior to the HS itself. Pre-calculating the wind loads refers to the technique of performing aerodynamic analysis of the structure prior to the HS using a simplified numerical model, then using the resulting load time history within the HS. This technique is computationally efficient and allows for real-time aerodynamic loads to be easily applied to the structure during the HS, but these pre-calculated loads are not able to react to the true structural responses that occur during testing. This means that the relationship between the aerodynamics and the structural response is severed, and thus aeroelasticity can not be properly modeled. For a highly-flexible wind-loaded structure such as a wind turbine, this limitation can affect results significantly – as will be quantified in Section 2.8, pre-calculating wind loads can result in errors of 10-70% depending on how the pre-calculation process is performed and the response index in question.
3. Neural networks (NNs) are trained machine learning models that can be used to relate input and output data without directly performing the underlying computations. For example, when properly trained on a large, diverse set of sample data, a NN could be used to accurately predict the aerodynamic loads on a turbine blade based on the wind speed without the use of an aerodynamic model. Such a trained NN can be referred to as a surrogate model, that is, a model that acts as a surrogate to the direct calculation. The major advantage that these surrogate models offer for this application is that the NN calculation is computationally light and can be run very quickly. Thus, NN-based surrogate models offer a method for time-consuming aerodynamic calculations to be pre-calculated, while still allowing them to be directly integrated into a real-time HS loop.

4. Wind tunnel testing is the most common technique for experimentally studying wind-loaded structures, where the models for these structures must typically be scaled-down significantly to fit in the wind tunnel. The Reynolds number (Re) is a nondimensional parameter that describes the turbulence in a given flow and is proportional to the structure size and the wind speed. If a structure is scaled down by 50%, the wind speed must be doubled to maintain the same Re . This means that it is essentially impossible to maintain Re -similitude in most structural wind tunnel tests due to very large geometric scaling factors. Unfortunately, wind turbine blade airfoils are very sensitive to changes in Re , meaning that geometrically-scaling an industrial turbine down to wind tunnel size will result in errors in the prediction of thrust and power production of the rotor. There are techniques to minimize these errors, but none completely eliminate them. Therefore, it can be said that traditional experimental studies of wind turbines are limited by scaling errors in wind tunnel tests, and thus an alternative testing technique capable of generating experimental-level results without the use of the wind tunnel would be very beneficial.

Based on these four observations, a clear direction for research can be seen: to use high-accuracy CFD data to train a neural network-based surrogate aerodynamic model to study wind turbines using HS. To perform such testing, the surrogate aerodynamic model will need to be paired with a structural turbine model as well as a test case for the HS, such as an external damping system.

1.2 Objectives

Given the clear need for a modified HS framework that incorporates improved aerodynamics modelling, the overall objectives of this research can be specified. There are four research areas that must be studied to be able to create and test the proposed surrogate model-based HS framework:

1. A proper understanding of background theory is required.
2. The surrogate aerodynamic model of the turbine rotor must be developed and validated. This entails the generation of a robust data set using validated and optimized CFD simulations of the turbine rotor, followed by the training of a NN with optimal type and architecture to create the surrogate model.
3. The test case for the HS must be chosen and validated. It was decided that the test case employed in this research would be a complex external damping system for the wind turbine, therefore this damper must be validated then built or modelled. In a true HS, this damper would be tested experimentally, however in the artificial HS that were ultimately performed here, the damper was instead modelled numerically.
4. A structural model of the turbine tower and blades must be developed and validated, ideally using experimental results. In a true HS, the structure could be modelled experimentally, however for the proposed test case it would typically be modelled numerically, as was done for the artificial HS performed in this research.

With these four areas in mind, the goals of this research are as follow:

1. The errors caused by pre-calculating wind loads must be quantified to justify the need for the proposed surrogate model-based HS framework.
2. The complex aerodynamic behaviour of airfoils that introduce the scaling errors in wind tunnel testing also makes them challenging to test using CFD. Using well-documented experimental data, an optimal set of CFD settings must be found such that accurate drag and lift loads can be predicted by the simulations.
3. The process of developing a NN-based surrogate aerodynamic blade model must be developed and validated. This includes generating the large amount of required testing data, and determining an optimal type and architecture for the NN.
4. The optimized CFD settings must be used to generate aerodynamic rotor data of the test turbine, which will then be used to train a NN with the optimal type and architecture. This will be the surrogate model ultimately used in the final HS.
5. A complex external damping system must be modelled and optimized for its effectiveness at controlling wind turbine vibrations, with the goal of applying this damping system as the test case for the final HS.
6. A dynamic structural model of a turbine must be developed. For maximal accuracy, a set of aeroelastic wind tunnel tests will be performed and the results will be used to define the properties of the turbine model, which will then be applied in the final HS.
7. The final HS will be performed of the structural model equipped with the optimized damper, with loading determined both using the existing HS framework that employs pre-calculated wind loads from a lower-accuracy aerodynamic model, as well as the novel, proposed framework that incorporates improved aerodynamic modelling via the NN-based, CFD-trained surrogate model.

1.3 Structure

Figure 1.1 identifies the six research papers presented in this thesis, and organizes them according to the four research areas identified in the previous section. While each paper stands on its own as independent research, Chapters 2-6 can also be classified as pertaining to one of the four research areas that directly contribute to the final artificial HS simulation performed in Chapter 7. Finally, Chapter 8 will summarize the findings of the previous six chapters and propose directions for future research. A brief summary is presented here for each of the six research papers.

1.3.1 Chapter 2: Wind turbines and hybrid simulation: an overview of research, opportunities and limitations

This chapter presents an introduction to wind turbine research, followed by a literature review of traditional experimental and numerical turbine testing, hybrid simulation in structural en-

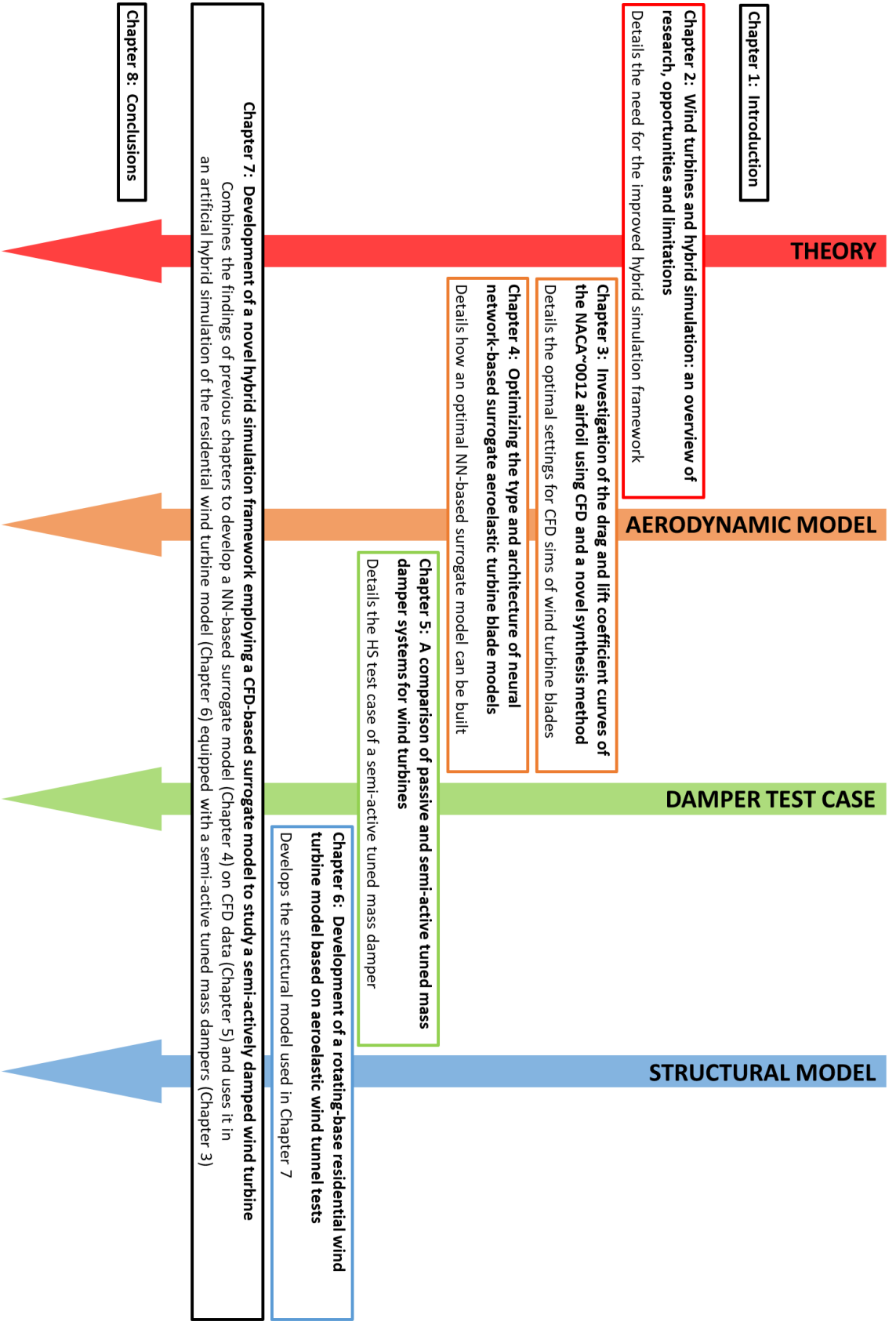


Figure 1.1: Classification of Chapters 2-6 as they relate to the four research areas building towards Chapter 7. These research areas are the underlying theory, the aerodynamic rotor model, the test case of a complex external damping system, and the dynamic structural model

gineering, and HS of wind turbines. Finally, the limitations of the current HS framework for wind turbines are shown via a numerical study that gauges the impact of aeroelasticity in the response of wind turbine blades. It is shown that the current technique of pre-calculating wind loads in hybrid simulations can introduce notable error into the results.

1.3.2 Chapter 3: Investigation of the drag and lift coefficient curves of the NACA 0012 airfoil using CFD and a novel synthesis method

This chapter studies the pre- and post-stalling behaviour of the NACA 0012 airfoil in flows where the Reynolds number ranges from 10^5 to 10^6 . Due to complexity of stalling behaviour on an airfoil in these conditions, a previous consensus has not been reached about the exact expected behaviour. This chapter firstly collects the drag and lift coefficient curves reported by existing experimental wind turbine studies and numerical simulations, and develops a series of piecewise polynomial systems to synthesize the average predicted curves as a function of the Re of the flow. This chapter secondly performs a number of CFD simulations to attempt to identify the optimal simulation parameters to generate results that match the synthesized average drag and lift coefficient curves. This includes the comparison of 2D and 3D CFD simulations, turbulence models (steady-state RANS, URANS and LES), and RANS closure equations (Spalart–Allmaras, $k-\epsilon$, $k-\omega$). Overall it was found that 3D URANS simulations with $k-\omega$ was most effective by balancing accuracy and computational requirements.

1.3.3 Chapter 4: Optimizing the type and architecture of neural network-based surrogate aeroelastic wind turbine blade models

This chapter trains and compares six different NNs using numerical aeroelastic blade data. The NNs include four multilayer perceptron (MLP) configurations, a long short-term memory (LSTM) network, and a convolutional neural network (CNN). Over 30 million data points of aeroelastic time history information of an operational 5 MW wind turbine were numerically generated for training these networks. The architecture, input and output data, and the optimization and training processes are provided in detail for each network, and the resulting accuracy and computation times are analyzed and compared. Ultimately, it was found that both the trained full-input MLP and the CNN were particularly accurate surrogate models, with average normalized root mean square errors of 1.11% and 0.66% respectively. While the CNN surpasses the full-input MLP in accuracy, the latter is simpler to train and faster to run, thus both are compelling options for future researchers.

1.3.4 Chapter 5: A comparison of passive and semi-active tuned mass damper systems for wind turbines

This chapter investigates a 1.5 MW wind turbine controlled by eight different types of tuned mass damper systems using a lumped-mass 3D finite element model. The turbines are subjected to loading and operational case combinations including service wind loads on operational turbines, seismic loading with service wind on operational turbines, and high-intensity storm wind loads on parked turbines. Ultimately, it was found that all the semi-active TMD

systems outperformed the passive systems. It was also shown that by reducing the mass of the TMD and adding a second smaller TMD below, the vibrations near the mid-point could be greatly reduced at the cost of a slightly lower reduction in vibrations at the tower top.

1.3.5 Chapter 6: Development of a rotating-base residential wind turbine model based on aeroelastic wind tunnel tests

This chapter develops an experimentally-validated numerical structural turbine model to create an ideal testbed for future HS. Firstly, two rounds of wind tunnel testing are described and catalogued of a full-scale 1.1 m residential wind turbine installed on a custom aeroelastic rotating base. A 1-DOF numerical structural model is then developed and validated based on the observed experimental behaviour of the turbine. The basic structural properties, the sources of loading, and the damping properties are then derived and validated from the experimental data. Finally, the developed structural model is applied in a representative numerical study that tests the effectiveness of passive and semi-active tuned mass dampers at controlling the structural rotation. The detailed structural model is shown to match the experimental results well and is effective in the representative case study.

1.3.6 Chapter 7: Development of a novel hybrid simulation framework employing a CFD-based surrogate model to study a semi-actively damped wind turbine

This chapter investigates a novel HS framework incorporating a CFD-based surrogate aerodynamic model, replacing the existing technique of pre-calculating aerodynamic loads. This study combines the findings of the previous chapters to perform twelve artificial HS of a residential wind turbine with an aeroelastic rotating base equipped with a STMD. Firstly, the development of the surrogate aerodynamic turbine blade model employing the findings of Chapters 3 and 4 is detailed including the description and validation of 24 URANS CFD simulations at a variety of wind speeds, tip speed ratios (TSRs) and base rotations. These CFD results are then used to train a CNN as a surrogate model capable of quickly estimating the blade thrust based on the wind speed, rotor speed and blade position. The novel framework employing this surrogate model is combined with the structural model developed in Chapter 6 and the optimal STMD developed in Chapter 5 to perform the AHS under three wind speed/TSR combinations. This is compared to AHS employing pre-calculated thrust time histories of varying aeroelastic fidelity. The predicted rotor thrust, base rotation, and effectiveness of the STMD across the twelve AHS are then compared to evaluate the effectiveness of the proposed framework; these observations indicate that the proposed surrogate model-based framework is feasible and a likely improvement over the existing framework.

Chapter 2

Wind turbines and hybrid simulation: an overview of research, opportunities and limitations

Summary

This chapter presents an overview of wind turbine research techniques including the recent application of hybrid simulation. Wind turbines are complex due to their large size, slenderness, and dynamic behaviour, which limits applicable research techniques. Traditionally, numerical simulation is widely used to study turbines while experimental tests are rarer and often face cost and equipment restrictions. Hybrid simulation is a relatively new simulation method that combines numerical and experimental techniques to accurately capture unknown or complex behaviour by modelling portions of the structure experimentally while numerically simulating the remainder. This can allow for increased detail, scope, and feasibility in wind turbine tests. Hybrid simulation appears to be an effective tool for future wind turbine research, and the few studies that have applied it have shown promising results, though they are not without their own limitations. This paper presents an introduction to turbine research, followed by a literature review of experimental and numerical wind turbine testing, hybrid simulation in structural engineering, and hybrid simulation of wind turbines. Finally, the limitations of current implementations of hybrid simulation for wind turbines are shown via a numerical study that gauges the impact of aeroelasticity in the response of wind turbine blades. It is shown that the current technique of pre-calculating wind loads in hybrid simulations can introduce notable error into the results.

This chapter is heavily based on updated reproductions of the author’s published papers “Wind turbine testing methods and application of hybrid testing: A review” (Lalonde et al., 2019) and “Effect of aeroelastic blade deflections on the overall response of a 5 MW wind turbine tower” (Lalonde et al., 2020a).

2.1 Introduction

Driven by the threat of climate change, wind energy has seen a major expansion of use in recent years (GWEC, 2017). One consequence of this large expansion of wind power is the required use of higher risk locations for wind farms. Near-shore and offshore turbines typically face higher typhoon and hurricane risks, and onshore turbines are prevalent in seismic-prone areas such as northern China, California, and Japan. High-intensity wind events such as typhoons are responsible for the majority of the numerous wind turbine failures each year (Chou and Tu, 2011) and are likely to increase in intensity in the future due to climate change (Haldar and Basu, 2016). Research has also shown that earthquake loading may govern for turbines in high-seismic areas (Diaz and Suarez, 2014; Mardfekri and Gardoni, 2015). Furthermore, since identical turbines are often used in a given wind farm, mass failure is risked in a single extreme event. Many of the most common wind turbine design codes (IEC, 2005; Riso, 2002; GL, 2010) lack explicit guidelines for certain aspects of design (Diaz and Suarez, 2014; Katsanos et al., 2016) and the structural design codes of some countries will result in unsafe wind turbine designs (Song et al., 2013; Stamatopoulos, 2013). Overall, the wind turbine industry faces serious challenges from structural failures, and given the growth of this industry, further studies of wind turbines will be of great value.

From an engineering perspective, wind turbines are challenging structures to study, which limits the types of testing available to researchers. Wind turbines are large, slender, hollow, and uniquely shaped structures with little redundancy and whose behaviour will greatly vary depending on operational states as well as loading type and direction. Due to their high flexibility, their lifespan is limited by fatigue failure from dynamic wind and (for offshore turbines) wave loading, which is difficult behaviour to model and test. Additional structural damping is often added to turbine towers to combat fatigue loading – a summary of the mainstream vibration control methods for wind turbines can be found in Rezaee and Aly (2016) – which can further complicate testing.

There are two primary categories of testing performed in structural engineering: experimental tests and numerical tests. Experimental (or physical) testing analyzes full-sized or scaled-down physical models to determine structural behaviour. This type of testing can accurately capture nonlinear or poorly-understood behaviour by testing real replicas of the structure, but the maximum scale of the model is often restricted by test budgets and equipment limitations, and smaller-scale models often struggle to accurately simulate structural properties. Alternatively, numerical (or analytical) testing, typically performed via computational analysis, allows for structures to be modelled at full scale with no real cost or equipment restrictions, however unknown or complex behaviour is difficult to capture as incorrect or flawed models will produce inaccurate outputs. Due to this, complimentary experimental or field research is often used to validate new numerical models, and cheaper and repeatable numerical testing can be used to design experimental tests. It should be noted that numerical and experimental testing are not mutually exclusive: an experimental study may employ numerical models for prototyping and validation, while a numerical study may employ models that were derived from experimental results. Generally, however, most studies can be classified as being primarily experimental or numerical.

For wind turbine studies, the advantages and challenges of both experimental and numerical testing are present: the large size of turbines means that experimental tests are either extremely

costly or heavily scaled down, and the complex behaviour of turbines can present challenges in numerical modelling. However, an alternative research method exists that appears quite promising for wind turbine studies: hybrid simulation. Hybrid simulation (or hybrid testing) combines experimental and numerical testing by modelling portions of the structure that undergo nonlinear or poorly-understood behaviour experimentally while the remainder of the structure is captured in numerical models. As a result, hybrid simulation allows for accurate physical simulation of complex behaviour, such as nonlinear damping or failure, at large scale with reduced costs and equipment requirements. Hybrid simulation may enable wind turbine research that is poorly suited for traditional numerical or experimental studies.

This overview will explore wind turbine research and the possible application of hybrid simulation via literature review as well as a numerical study: the following two sections introduce the fundamentals of wind turbines and wind turbine modelling. The following sections will then present reviews of numerical and experimental wind turbine studies respectively and discuss the advantages and challenges of each method. The development and capabilities of hybrid simulation for general structural engineering research will then be presented, followed by an examination of hybrid simulation for wind turbines in particular. This review will highlight the limitations in existing hybrid simulations of wind turbines: the use of reduced aeroelastic fidelity and pre-calculated wind loads. To quantify the effect of these limitation, a numerical study is performed on a tuned mass damper-equipped wind turbine. Finally, this chapter will finish with a summary of the drawn conclusions and propose future research directions.

2.2 Wind turbine parts, types and frames of reference

Here, the fundamentals of wind turbines that will be referenced throughout this thesis are presented. This consists of the parts of a turbine, the different types of turbines, and the various frames of references when performing structural analyses of wind turbines. The turbines discussed here are all horizontal-axis wind turbines – the most common type of turbines used for large-scale applications. More complex aspects of wind turbine design such as pre-coning, tilting, and teetering are excluded here for simplicity.

Figure 2.1 lists the main components of a wind turbine: the tower, nacelle, and rotor. The tower is hollow to allow workers to reach the nacelle for maintenance, includes a door at ground level, and is in a fixed position. The nacelle houses the generator system and is capable of rotating (or yawing) towards the wind to maximize energy production, though this rotation can be braked in high-intensity wind situations. The rotor consists of the hub and the blades, and rotates about the rotor axis, which can also be braked. Wind turbine blades are typically hollow and made from balsa frames covered in fiberglass.

The azimuth, twist, and pitch are all common angle measurements: the azimuth refers to the position of the blade within the rotor plane, which is equal to 0° when the blade points straight up. The blade twist refers to the rotation of the airfoils along the length of the blade, which is fixed when the blades are built. The blade pitch refers to the ability of the blade to rotate axially, the optimal pitch angle will depend on the blades and wind speed. During high-intensity wind events, the blades are “feathered” which means they are rotated and locked to a pitch of 90° to minimize wind loads and protect the turbine.

Figure 2.2 shows the different types of wind turbines. While onshore turbines are most com-

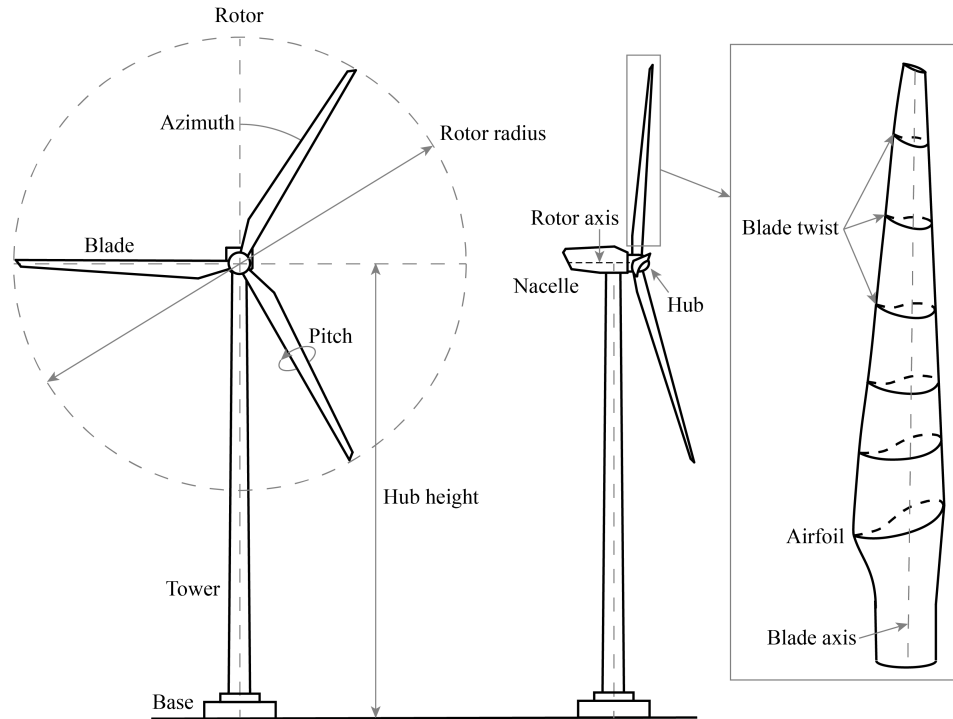


Figure 2.1: Parts of a horizontal-axis wind turbine

mon, offshore wind turbines are seeing increased implementation due to higher wind speeds and greater available space. Depending on the depth of the water where offshore turbines are built, they can either be fixed or floating. Fixed offshore wind turbines are generally cheaper, however are impractical when water depths exceed 40-50 m. Floating offshore turbines include barge, buoy, and tension leg systems. Offshore turbines must consider wave loading in addition to the standard loads on wind turbines, while floating offshore turbines must also consider mooring loads and much greater deflections. When performing numerical or experimental analysis of onshore or fixed offshore turbine, the soil-structure interaction (SSI) of the base may or may not be considered. While SSI is often ignored, it can be accounted for numerically using relatively simple flexible bases (shown in Figure 2.2) and experimentally using soil tanks.

Figure 2.3 shows the different frames of reference when studying wind turbines. Firstly there is the fixed frame of reference from the turbine base that is relative to cardinal directions. More importantly, however, is the nacelle frame of reference, which is relative to the rotor axis and the wind direction. Since the nacelle yaws to face the direction of the wind during operational conditions, this is the most important frame of reference for most turbine analysis. The nacelle frame of reference yields the following loading directions: fore-aft/along-wind loading is parallel to the rotor axis, side-side/across-wind loading is horizontal and perpendicular to the rotor axis, and axial/vertical loads is perpendicular to the ground. Thrust (T) and torque (Q) capture the sum of the rotor loads in the nacelle frame of reference, where thrust is the along-wind force, while torque is the moment about the rotor axis and is directly related to the power production of the turbine.

Three different frames of reference are also commonly used when analyzing turbine blades.

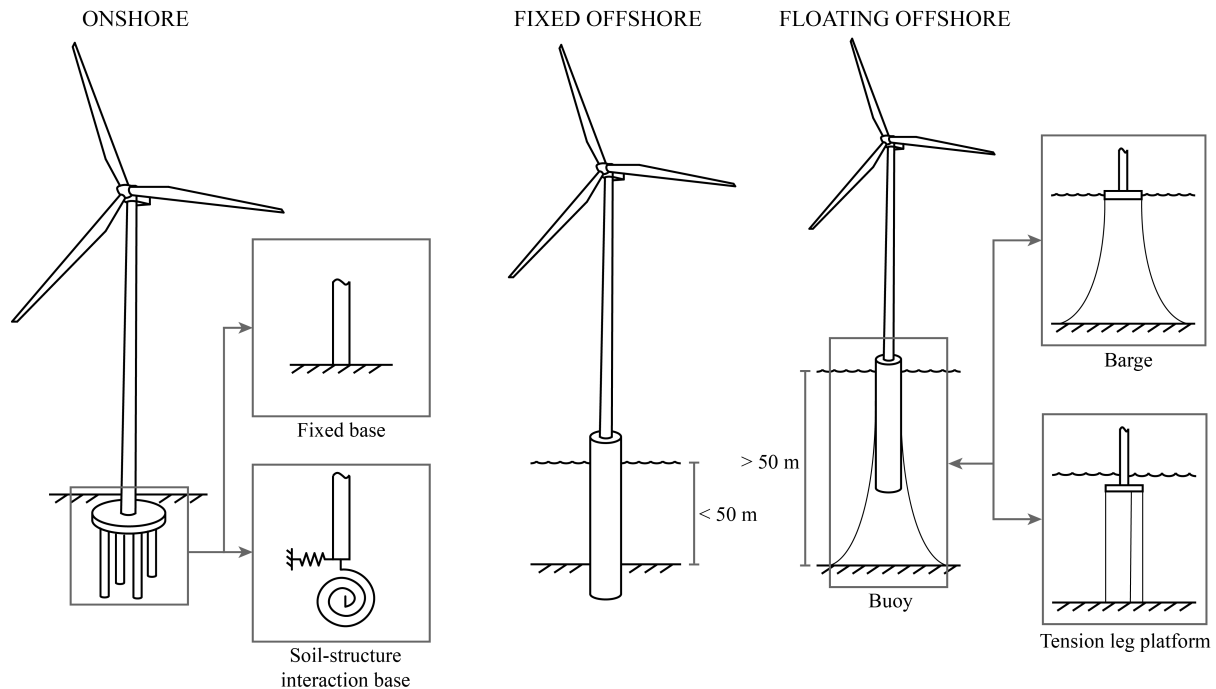


Figure 2.2: Different types of turbines with possible bases and modelling methods

The global blade frame of reference is used to determine the thrust and torque loads. Here, the loading directions are the along-wind, the rotor and the axial directions, where the along-wind loads are used to calculate the thrust, the rotor loads are used to calculate torque, and the axial direction follows the axis of the blade as it rotates. The 2D airfoils that run along the wind turbine blades each have two local frames of reference: one for the drag, lift and axial loads and one for the edgewise, flapwise and axial loads. In both, the axial loads are identical, while the drag and lift loads are relative to the relative wind direction (which includes both the natural wind plus the relative wind from the blade rotation). Conversely, the edgewise and flapwise loads are relative to the chord of the airfoil. Should the relative wind angle line up with the chord line of the airfoil, these values are identical. Drag and lift loads are used to characterize the airfoils (see Section 2.3.2) while the edgewise and flapwise loads are more easily converted to the other frames of reference since the blade twist is fixed while the relative wind angle changes with time.

2.3 Numerical characterisation of wind turbines

This section will briefly present the fundamental behaviour of wind turbines – how they convert wind flow into mechanical energy – in numerical form. Understanding this is key for understanding and designing experimental studies, and these numerical representations are often directly employed in numerical studies. Betz’s actuator disk model and the power coefficient of a turbine from momentum theory, as well as the lift and drag coefficients of an airfoil will be discussed herein.

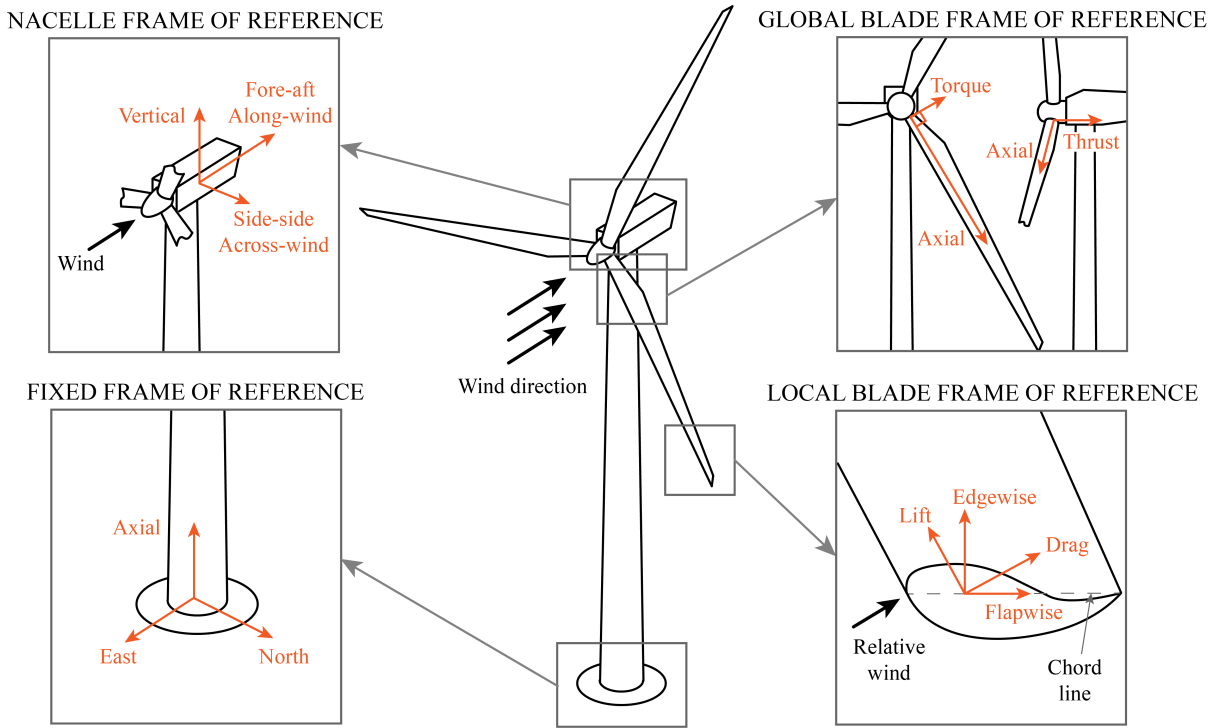


Figure 2.3: Standard frames of reference for structural analysis of wind turbines

2.3.1 Actuator disc model and momentum theory

The actuator disc model, whose development is usually attributed to Betz (1926), can be used to characterize the power production of a wind turbine rotor as the loss of power in wind flow from upstream to downstream. For a more detailed explanation of the concept, the reader is directed to Refan (2009) or any of the introductory textbooks on wind turbine behaviour.

This model assumes an ideal turbine in incompressible, steady, uniform flow, with an infinite number of blades and no friction. Consider a circular stream of wind of varying radius that maintains a constant mass flow rate as shown in Figure 2.4. At any cross-section of the stream, the mass flow rate can be calculated as ρAU where ρ is the air density, A is the cross-sectional area, and U is the velocity of the flow. With conservation of mass flow rate and incompressible flow, it can be written that:

$$\rho A_u U_u = \rho A_r U_r = \rho A_d U_d \quad (2.1)$$

Here, the subscript u refers to the upstream region; r refers to the stream at the rotor (where A_r is the swept area of the rotor); and d refers to the downstream region. Finally, let a refer to the axial induction factor such that:

$$U_r = U_u(1 - a) \quad (2.2)$$

Based on the decrease in flow velocity from the upstream flow, the power production of the turbine thus be determined.

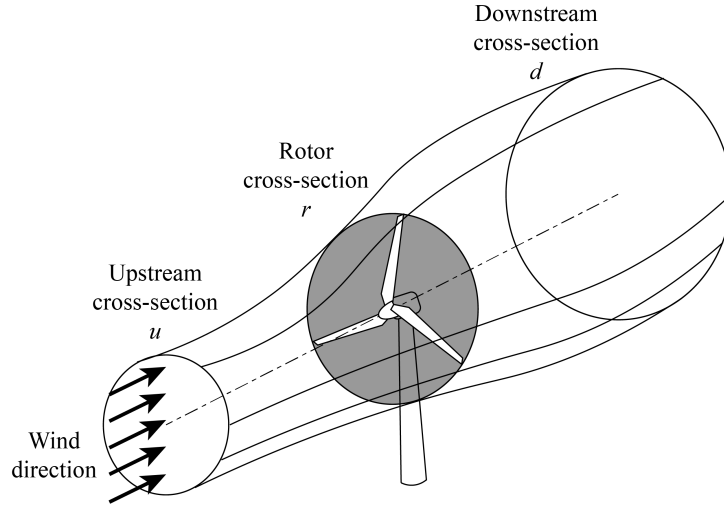


Figure 2.4: Actuator disk model and the locations of the three cross-sections of interest

Firstly, the thrust (the along-wind force) T on the rotor can be calculated as the mass flow rate times the drop in wind velocity between the upstream and downstream sections:

$$T = \rho A_r U_r \cdot (U_u - U_d) \quad (2.3)$$

The source of the thrust can equally be considered as the sum of a positive pressure on the upwind face of the rotor plane and a negative pressure of the downwind face, the difference between which is shown as Δp_r :

$$T = \rho A_r \cdot \Delta p_r \quad (2.4)$$

Secondly, by modelling the upstream and downstream portions of the wind stream using Bernoulli's equation and subtracting these two equations, it can be shown that:

$$\Delta p_r = \frac{1}{2} \rho (U_u^2 - U_d^2) \quad (2.5)$$

By combining Equations 2.2 - 2.5:

$$U_d = U_u (1 - 2a) \quad (2.6)$$

Finally, the power P extracted from the stream by the rotor is equal to the thrust times the air velocity, using Equations 2.2 and 2.6:

$$P = T U_r = \rho A_r U_r^2 (U_u - U_d) = 2 \rho A_r U_u^3 a (1 - a)^2 \quad (2.7)$$

The power coefficient C_p is typically used to nondimensionalize the power production of a turbine. It is the ratio of power production over the total power in the wind:

$$C_p = \frac{P}{\frac{1}{2} \rho U_u^3 A_r} \quad (2.8)$$

By combining Equations 2.7 and 2.8 we find that:

$$C_p = 4a(1 - a)^2 \quad (2.9)$$

By analyzing C_p as a function of a , it can be found that the maximum possible power coefficient is achieved when $a = \frac{1}{3}$, which yields $C_p = 0.593$. This is known as the Betz limit – the maximum possible efficiency of a propeller. Note that in reality it is impossible for a turbine to achieve this level of efficiency, as there will always be losses from unsteady flow, mechanical friction, etc. The goal of wind turbine design is to maximize the power coefficient thus maximizing the efficiency at which a turbine can extract power from the wind.

For a non-idealized turbine, the maximum possible C_p will depend on the design of the turbine as well as the tip speed ratio the turbine operates under. The tip speed ratio (TSR) relates the wind speed to the rotation speed of a turbine, as shown in Equation 2.10:

$$\text{TSR} = \frac{\omega R}{U} \quad (2.10)$$

Here, ω is the rotation speed of the turbine and R is the radius of the turbine rotor, thus ωR is the tip speed of the given rotor; and U is the wind speed. For a given size of turbine in a given wind speed, the optimal tip speed ratio will identify the ideal rotation speed of the turbine. There is a classic figure in many wind turbine textbooks that compares the maximum power coefficients of various turbines versus their tip speed ratio to Betz's theoretical max C_p – it is reproduced here in Figure 2.5 from Hau (2006). From this figure, it can be seen that of the designs, the modern 3-bladed turbine is able to achieve the maximum possible C_p at a tip speed ratio of about 7.

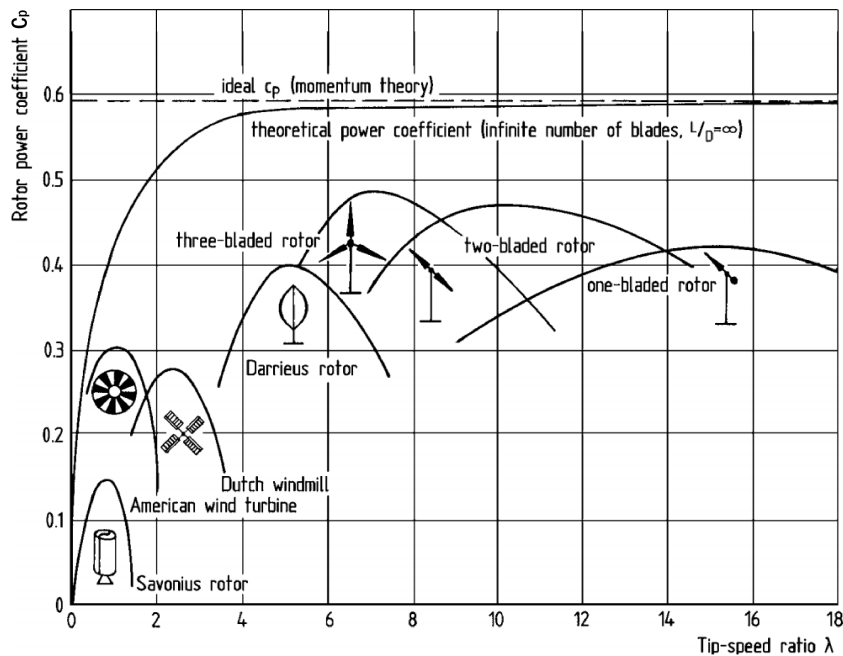


Figure 2.5: The maximum power coefficients as a function of tip speed ratio (λ) of various turbine designs. Reproduced from Hau (2006) with permission from Springer Nature ©

2.3.2 Airfoil lift and drag

In reality, the power of a turbine must be determined by studying the airfoils that make up the turbine blades. Airfoils are curved cross-sections that generate lift from incoming wind, famously employed in airplane wings, propellers, and turbine blades.

Figure 2.6a shows the relevant parameters of an airfoil in air flow: the wind passing over the airfoil generates a parallel drag force (F_D) and a perpendicular lift force (F_L). In a wind turbine, the lift force is generally responsible for driving the rotor. The amount of lift and drag produced by an airfoil is a function of the wind speed, the airfoil shape, the chord length (c) and the angle of attack (α). Figure 2.6b shows a very simple explanation of how lift is generated by an airfoil using Newton's third law: the flow being deflected downwards applies an equal upwards reaction on the airfoil. Figure 2.6 also shows the two types of airfoils: symmetrical and cambered. The main differences between these two is that a cambered airfoil will generate lift at an angle of attack of 0° and are capable of achieving higher maximum lift, thus they are typically employed in wind turbine applications.

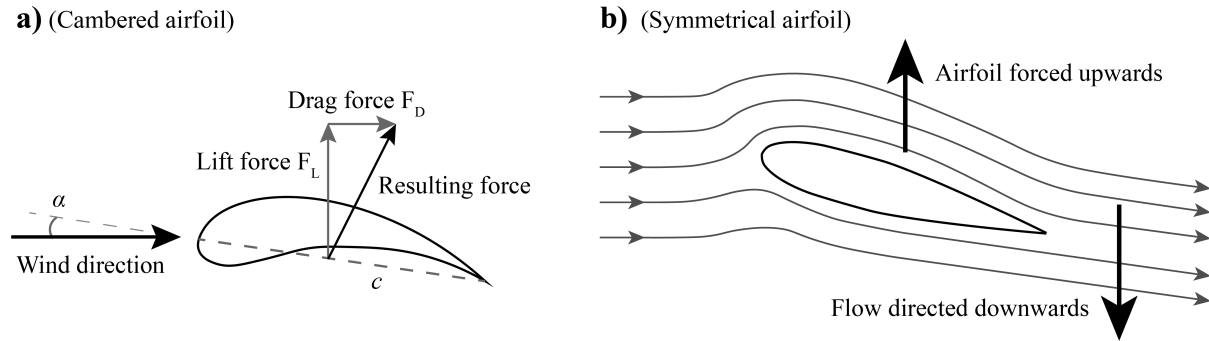


Figure 2.6: a) Behaviour and parameters of an airfoil. b) Lift force explanation

As the power was nondimensionalized previously, the same can be done to the lift and drag forces on an airfoil. Equations 2.11 and 2.12 define the drag and lift coefficients of an airfoil:

$$C_d = \frac{F_D}{\frac{1}{2}\rho U^2 A} \quad (2.11)$$

$$C_l = \frac{F_L}{\frac{1}{2}\rho U^2 A} \quad (2.12)$$

Here, C_d and C_l are the drag and lift coefficients, respectively, while F_D and F_L are the respective drag and lift forces. ρ is the density of the air, U is the wind speed, and A is the surface area – for an airfoil the chord length (c) times the airfoil width is used.

The drag and lift forces acting on an airfoil are difficult values to calculate directly. Instead, plots of the drag and lift coefficients of an airfoil as a function of the angle of attack are derived via experimental or numerical simulation. Example plots of the drag and lift coefficients of the symmetrical NACA 0012 airfoil are reproduced in Figure 2.7 from Sheldahl and Klimas (1981). These plots are dependent on the turbulence of the flow and the magnitude of the wind speed, which are characterized by the nondimensional parameters the Reynolds number

(Re) and the Mach number (Ma). For wind turbines, the Mach number is of little concern, however the impact of the Reynolds number is notable, as will be discussed in greater detail in Section 2.5.1.1.

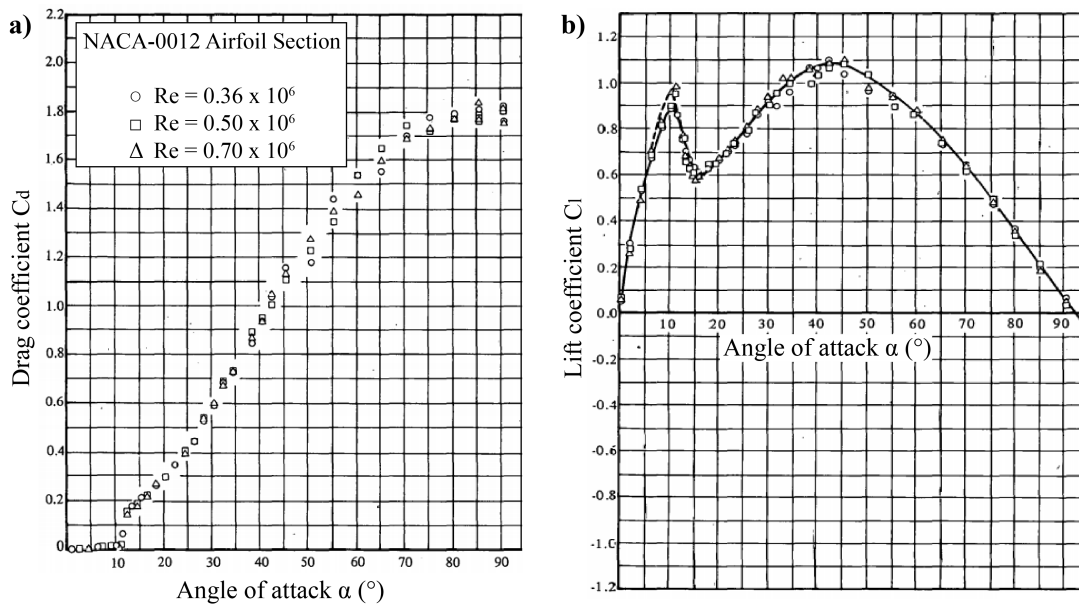


Figure 2.7: a) Drag coefficient and b) Lift coefficient versus angle of attack plots of the NACA0012 airfoil. Reproduced from Sheldahl and Klimas (1981) with permission from U.S. Gov. ©; with clarified axes and legend

By studying Figure 2.7, the relationship between the lift coefficient and the angle of attack can be studied. It can be seen that as α increases, C_l also increases, up to an angle of attack around 10-15°, where the C_l peaks. This is known as the stall angle of the airfoil and occurs when the flow separates from the top of the airfoil. The post-stall behaviour of the airfoil sees the C_l decreasing then eventually increasing and decreasing again as the angle of attack continues to increase, however due to the transient, separated flow, it is more challenging to capture this complex behaviour in testing. Further, while industrial turbine blades can experience very high angle of attacks in high winds, these angles are rarely encountered in aeronautics. These together mean that the post-stall behaviour of airfoils is often under-reported in literature – this topic is investigated further in Chapter 3. In wind turbine design, the rotation speed, airfoils, blade twist, and pitch of the blades are carefully chosen in an attempt to control the angle of attack thus optimizing the resulting lift on the blade.

2.4 Traditional numerical turbine studies

Numerical testing of wind turbines consists of simulating the behaviour of wind turbines using a series of numerical approximations. A simple example could be modelling the turbine tower using Euler-Bernoulli beam theory, a model which has been rigorously validated in previous literature. Unlike experimental testing, numerical simulations have no need of expensive lab equipment beyond a computing platform and are particularly suited for investigatory and

parametric analyses where models are cheap to iterate and have no risk of being permanently damaged. An additional advantage of numerical testing is the ability to control the required fidelity based on the scope of the project – degrees of freedom (DOF) that are not relevant to a given study will often be excluded from the model to reduce computational costs. The major limitation of numerical simulations is that results are only as accurate as the models used to generate them, for example, should the turbine tower be subjected to significant twisting or plastic yielding, the Euler-Bernoulli model would become invalid and fail to accurately capture the tower's behaviour. Thus, particularly complex and nonlinear behaviour can often be challenging to accurately capture using numerical simulations.

The following subsections will introduce some of the fundamental techniques used in numerical simulations of wind turbines, including finite element (FE) modelling, the blade element momentum theory (BEM) and computational fluid dynamics (CFD). Finally, a summary of recent examples of numerical studies of wind turbines will be presented and analyzed.

2.4.1 Finite element modelling

Finite element (FE) modelling is a technique used in numerical simulation to solve complex behaviour. By taking a system that in reality is governed by a series of extremely complex equations and splitting it into a series of small, interconnected elements, these behavioural equations can be solved instead as a series of simple, linear equations. For example, a long beam with complex 2D behaviour can be simplified into a series of small beam elements with four degrees of freedom each (vertical displacement and rotation at each end), allowing the overall behaviour to be found as a simple system of linear algebraic equations. Section 5.2.1 in this document uses this technique to model the tower of a wind turbine, and for a more thorough explanation of the technique the reader is directed to one of the many textbooks that provide an introduction to this topic such as Reddy (1993).

FE analysis includes a number of types of elements of varying complexity – the more complex an element is, the more DOF it has and thus the more computationally intensive it is to solve the resulting system of linear equations. For the purposes of this section, two types of elements will be discussed: simpler beam elements used in elastic analysis of beam-like structural members such as towers and blades, as well as more complex shell elements which are used for stress and strain analysis of surfaces such as the walls of a turbine tower or the shell of a blade. Generally speaking, for wind turbine analysis, beam element models will be used to study the overall structural response while shell elements will be used to investigate local damage from fatigue or other sources. The numerical studies presented in Section 2.4.4 that use FE models will be presented as either beam element-based or shell element-based.

2.4.2 Blade element momentum theory

Calculating the wind loads on turbine blades represents the most challenging aspect of wind turbine simulation. Industrial wind turbine blades have significant variation in their chord length and twist, and are often built of different airfoils with different drag and lift coefficient curves. When aeroelasticity is considered by allowing the blades to flex under the wind loads, it becomes an even more complex process as turbine blades are irregular, composite structures made from fiberglass and wood, and the resulting deformations will modify the angle of attack

of the wind, thus affecting the loads. In practice, the principles of FE analysis are carried over into the calculations; by dividing the blade into a series of uniform airfoil cross-sections, the drag and lift can be determined on each element and summed along the length of the blade to determine the overall loading. This technique is referred to as the blade element momentum (BEM) theory.

BEM builds upon the momentum theory discussed previously and can be used to estimate the wind loads on a turbine blade and the resulting power generation. Originally developed by Glauert (1935) for use in aeronautics, the derivation of this technique is somewhat lengthy, thus it is excluded here for brevity in favour of a brief summary, but the derivation can be easily found in literature (Refan, 2009) or online. The technique consists of using an iterative method to solve for the axial induction factor a and the angular induction factor a' , which in turn can be used to estimate the true angle of attack of the wind on a blade. Then, using the lift and drag coefficient curves, the lift and drag forces as well as the resulting thrust and torque on the blade section can be found. By performing this analysis along the length of the blade, the total forces can be found. However, this theory has a number of limitations stemming from the fact that it is fundamentally a 2D analysis of the forces on a turbine blade. To account for this, there are a number of typical correction factors that are applied to account for tip losses and stall delays, among other effects. Modern researchers are still attempting to improve this method (Maheri et al., 2006; Madsen et al., 2007; Macquart et al., 2012), including developing unified models that combine the numerous proposed corrections (Liu and Janajreh, 2012).

BEM theory is highly valued for its simplicity and fast computational times. While direct aeroelastic analysis is extremely computational intensive, BEM approximation is significantly faster to perform and easy to incorporate into numerical models. As a result, BEM theory is used in the majority of numerical wind turbine studies, which will be shown in Section 2.4.4.

2.4.3 Computational fluid dynamics

Computational fluid dynamics (CFD) offers an alternative to BEM for aerodynamic simulations of turbine blades. CFD is a modern numerical analysis technique that can simulate the flow of fluid around objects and determine the resulting loads. CFD uses an FE-like process of splitting the fluid domain into a number of small elements and solving for the behaviour of each element to characterize the behaviour of the full, complex system. CFD has seen a large amount of use for structural analysis of wind-loaded structures – a review of which can be found in Dagnev and Bitsuamlak (2013). Martinez-Tossas and Leonardi (2012) have likewise prepared a review of CFD used for wind turbine studies; such studies will be explored in more detail in Section 2.4.4.

If CFD analysis is performed correctly, it can yield extremely precise results, often comparable to the results from wind tunnel experiments, which greatly exceeds the accuracy of other numerical aerodynamics models such as BEM. Lee et al. (2017) found that when comparing CFD and BEM results to an experimental wind turbine test, that the accuracy of the CFD simulations was at least 5% greater than the BEM simulations. The main limitations of CFD are its relative difficulty to learn, and its extreme computational intensity. In fact, large CFD simulations often require days or weeks to run on super-computing platforms, hampering it as an option for less-equipped researchers.

CFD functions by iteratively solving the flow and pressure behaviour within each element

of the fluid domain. The specific equations to solve depend on the turbulence model used in the analysis – Reynolds-averaged Navier-Stokes (RANS), unsteady RANS (URANS), large eddy simulation (LES) and detached eddy simulation (DES) are all common models. Deriving these turbulence models is a lengthy process, thus they are excluded for brevity, but the derivations can be easily found in literature (Versteeg and Malalasekera, 2007) or online. Instead, each of these models are summarized here:

- **RANS:** The RANS equations are the most common approach to turbulence modelling. They consist of a series of time-averaged equations of motions for fluid flow, which means that RANS is best suited to steady-state problems. The RANS equations introduce Reynolds stresses into the equation of motions, which in turn must be solved using additional turbulence models including Spalart-Allmaras (SA), k - ϵ and k - ω models. In addition to being the foundation for all mainstream CFD turbulence models, the main advantage of RANS simulations are that they are computationally light compared to the alternatives.
- **URANS:** For unsteady simulations, RANS can be used to approximate a time-history analysis as URANS. For situations where the time scale of the turbulence is much smaller than the time scale of the flow, URANS can find a time history of steady-state solutions that approximates the true unsteady time history while remain relatively lightweight computationally. An airfoil in the post-stall regime is an example of a situation where a URANS simulation would be preferable to RANS.
- **LES:** LES modifies the RANS equation by performing low-pass filtering to remove very small-scale turbulence, allowing true time-history simulations to be performed in a feasible amount of time. LES is more computationally intensive than RANS, particularly because it requires a more refined grid in the fluid domain to ensure accuracy, but is capable of achieving time-history results with a greater level of accuracy.
- **DES:** DES combines RANS and LES by performing a RANS simulation that switches to LES in regions with very fine grids, reducing the overall computational costs compared to LES. However, it is more challenging to generate the fluid domain grid for DES, and to transition the turbulence between the RANS and LES models.

Chapters 3 and 7 of this thesis explore the use of CFD for determine aerodynamic loads on a turbine rotor and simulating the aeroelastic response of a wind turbine.

2.4.4 Summary of numerical tests

A collection of modern numerical wind turbines is presented in Table 2.1. It would be impossible to provide a thorough summary of the thousands of turbine studies in print, thus this table has been curated to emphasize the general trends the author finds notable. Table 2.1 presents the study, a general synopsis, the type of turbine tested (see Section 2.2), the types of applied loads, the structural and aerodynamic models used (if any), the testing indices, and finally a general category that the study falls in to. This section will discuss the trends presented by this table.

Paper	Summary	Turbine type	Load types	Dynamic structural model	Aerodynamic model	OpenFAST?	Testing indices	Category
Adhikari and Bhatnagya (2011)	Develops a method to find natural frequencies of turbines considering SSI	Onshore with SSI	AR	FE beam model	—		Modal frequencies	Model evaluation
Li et al. (2012)	Validates CFD sim of turbine using wind tunnel data	Turbine blades	SW	—	CFD-RANS and DES		Base shear, base moment	Validation
Wang et al. (2013)	Determines response of turbines under typhoon loads	Onshore	HW, RN	FE shell model	BEM		Top response, base stress	Behaviour evaluation
Yelmule and Ajuiri (2013)	Validates CFD sim of turbine using wind tunnel data	Turbine blade	SW	—	CFD-RANS		Power, thrust, axial blade loads	Validation
Diaz and Suarez (2014)	Evaluates turbine under seismic loading	Onshore	SW, HW, EQ	FE beam model	Simple drag equation		Blade stress, tower stress	Behaviour evaluation
Van der Woelde and Narasimhan (2014)	Evaluates turbine with vibration isolation	Fixed offshore	SW, EQ	FE beam model	BEM	Y	Top response, base stress	Vibration control
Zhang et al. (2014a)	Evaluates turbine under typhoon	Onshore	HW, RN	FE shell model	Simple drag equation		Base stress	Behaviour evaluation
Chen et al. (2015a)	Validates field investigation of turbine failures from typhoon	Onshore	HW	FE beam model	Dynamic amplification		Blade failure, tower failure	Validation
Dai et al. (2015a)	Develops assessment model for turbines under seismic loads	Onshore	EQ	FE beam model	—		Modal frequencies, base shear	Model evaluation
Do et al. (2015a)	Develops a FE model for fatigue analysis	Onshore	SW	FE beam model	Simple drag equation		Fatigue damage	Model evaluation
Mardikri and Gardoni (2015)	Evaluates fragility under multi-hazard loading	Fixed offshore	SW, HW, WV, EQ	FE beam model	BEM	Y	Base shear, base moment	Behaviour evaluation
Berry-Brandt and Ruiz (2016)	Develops probabilistic model of turbine tower fatigue	Onshore	SW	FE shell model	BEM		Fatigue damage	Model evaluation
Butkala et al. (2016)	Develops practical modelling methodology	Residential	SW	FE beam model	BEM	Y	Power generation	Model evaluation
Chen and Xu (2016)	Validates field investigation of turbine failures from typhoon	Onshore	HW	FE beam model	BEM	Y	Blade failure, tower failure	Validation
Smith and Mahmoud (2016)	Evaluates behaviour under multi-hazard loading	Onshore	SW, EQ	FE beam and shell model	Simple drag equation		Shear, power, stress, fatigue	Behaviour evaluation
Amrithi and Jung (2017)	Evaluates turbines under hurricane loads using improved wind time history model	Onshore	HW	FE beam model	BEM	Y	Base moment	Behaviour evaluation
Dai et al. (2017b)	Evaluates collapse of turbine tower	Onshore	EQ	FE shell model	BEM		Top response, shear, stress	Behaviour evaluation
Nunez-Casado et al. (2017)	Proposes assembly strategies based on evaluated fatigue damage behaviour	Onshore	SW	FE beam model	Simple drag equation		Fatigue damage	Behaviour evaluation
Rong et al. (2017)	Develops analytical method for determining natural frequency of monopiles	Fixed offshore	AR	FE beam model	—		Modal frequencies	Model evaluation
Langer-Moeller et al. (2017)	Validates CFD sim of turbine using wind tunnel data	Turbine blades	SW	—	CFD-RANS		Power, thrust, torque	Validation
Lee et al. (2017)	Evaluates turbine behaviour using FSI	Onshore	SW	FE shell model	CFD-RANS		Thrust, torque	Model evaluation
Ebrahimi and Mardani (2018)	Evaluates designs to minimize noise caused by turbine blades	Turbine blade	SW	—	CFD-LES		Blade velocity and drag	Behaviour evaluation
Sun and Jahangiri (2018)	Evaluates turbine with 3D pendulum damper	Fixed offshore	SW, EQ, WV	FE beam model	BEM		Top response	Vibration control
Wang et al. (2018)	Develops and tests control scheme	Onshore	SW	FE beam model	BEM	Y	Power generation	Control
Park et al. (2019)	Evaluates offshore turbines with MR dampers	Fixed and floating offshore	SW	FE beam model	BEM	Y	Tower shear, tower moment	Vibration control
Alkhoury et al. (2020)	Develops model for modal analysis of offshore turbines including SSI	Fixed offshore with SSI	AR	FE beam and shell model	—		Modal frequencies	Model evaluation
Ali et al. (2020)	Investigates impact of model simplifications on seismic turbine studies	Fixed offshore	SW, EQ	FE beam model	Simple drag equation		Tower stress	Model evaluation
Chapman and Aly (2020)	Evaluates turbine with pendulum pounding TMD	Onshore	AR	FE beam model	—		Top response	Vibration control
Colberthaus et al. (2020)	Evaluates turbine with pendulum TMD	Fixed offshore	SW, EQ	FE beam model	BEM		Top response, base stress	Vibration control
Lalonde et al. (2020b)	Evaluates a turbine with different TMDs	Onshore	SW, HW, EQ	FE beam model	BEM	Y	Top response	Vibration control
Liu et al. (2020b)	Evaluates best method for modelling turbine wake using LES results	Onshore	SW	—	CFD-LES		Wind flow, vorticity	Model evaluation
Zhang and Fitzgerald (2020)	Evaluates turbine with inerter TMD	Onshore	SW	FE beam model	BEM		Blade vibrations	Vibration control
Griderstev et al. (2021)	Evaluates aeroelastic response using FSI	Rotor	SW	FE beam model	CFD-DES and LES		Blade loads	Behaviour evaluation

Table 2.1: A collection of numerical wind turbine studies. Load types are artificial (AR); service wind (SW); high-intensity wind (HW); wave (WV); and rain (RN)

Many of these tests use the open-source turbine simulator OpenFAST Jonkman et al. (2021), which employs a beam-element-based FE model for the turbine, including an improved beam element for the blades (Wang et al., 2014), and the BEM theory for aerodynamic blade loads. Further, its partner-program TurbSim (Kelley and Jonkman, 2012) can be used to easily generate wind field for use by OpenFAST. Studies that used OpenFAST are listed in Table 2.1 in the OpenFAST? column. OpenFAST can be used solely for load generation (Lalonde et al., 2020b), but is more typically used for both load evaluation and dynamic structural analysis.

From Table 2.1, the following trends can be drawn:

- It can be seen that numerical simulation is very flexible in what research it applies to. However, we see that the literature can be classified into a few broad categories: **behaviour evaluation**, the evaluation of the response of a turbine under certain conditions; **control**, the evaluation of a control technique for turbines, often vibration control via tuned mass damper (TMD) or similar system; **model evaluation**, the development and testing of new numerical representations of physical phenomena; and **validation**, using numerical simulation to validate observed physical results. The different numerical simulations performed throughout this thesis meet all four of these categories, including Section 2.8 for behaviour evaluation, Chapter 5 for vibration control, Chapter 4 for model evaluation, and Chapter 6 for validation.
- While many researchers study offshore turbines, the default structure in these tests are onshore wind turbines. While soil-structure interaction (SSI), i.e. the understanding that a turbine foundation in soil is not truly fixed, has been shown to have an impact on the response behaviour of both onshore (Adhikari and Bhattacharya, 2011) and fixed offshore (Alkhoury et al., 2020) turbines, most researchers opt to use the simpler model of a fixed foundation.
- It can be seen that service wind loads, the most common type of loading a wind turbine will experience during its lifetime and the most relevant to power production, is the most common test case in numerical turbine simulations. High-intensity wind and earthquake loading appear more often in vibration control tests as well as tests where member stress is being evaluated to assess the fatigue or ultimate limit states. Offshore wind turbines are often also subjected to wave loading. Studies that evaluate the natural frequencies of turbines typically use artificial harmonic loading. It is clear that a numerical simulation only requires an aerodynamic model if the turbine is subjected to service or high-intensity wind loads.
- FE analysis is generally used for dynamic turbine models. A clear line can be drawn between the type of FE models and the performance indices in this table: when simpler beam models are used, generally only the response of the turbine (the displacement, acceleration, etc.) is of interest, whereas when researchers are interested in element stress more complex shell models are employed. Sometimes models combine both beam and shell elements, such as by modelling the blades as beams and the tower as a shell (Smith and Mahmoud, 2016).
- BEM theory is the most common aerodynamic model employed in numerical simulations, partially due to its inclusion in OpenFAST, and particularly due to its simplicity

and accuracy compared to the alternative methods. Such alternative methods include performing a simple drag calculation, which fails to account for the angular momentum of the rotating turbine blades as well as coming with all the limitations of the 2D-based BEM theory.

- CFD is employed in a minority of wind turbine studies, though its popularity is increasing as more researchers gain access to sufficiently powerful computer systems. It can also be seen that RANS is the most common type of CFD, though LES and DES are increasing in popularity with time as well. For nearly all the CFD studies, they are not paired with a dynamic structural model. To include a flexible structure in a CFD simulation requires a moving mesh and a coupled dynamic model, greatly increasing the complexity and overhead of an already complex process. Pairing CFD with a dynamic model is commonly referred to as fluid-structure interaction (FSI) studies, which are fairly uncommon (Lee et al., 2017; Grinderslev et al., 2021).
- The limitations of traditional numerical turbine simulations can be inferred from the table. It can be seen that there are very few dynamic analyses of turbine blades in particular due to the limitations in BEM as well as the relative material complexity of the wood and fiberglass that makes up the blades. Further, there are very few numerical studies of turbines with complex vibration damper systems until recently. These both represent scenarios where numerical simulations are the weakest: when complex non-linearity lacks an effective numerical model thus the models are prone to a large amount of error.

2.5 Traditional experimental turbine studies

While it is employed in only a minority of wind turbine studies (Katsanos et al., 2016), experimental testing is a potent tool for researchers. It is well suited for studying nonlinear and unknown behaviour. However, given the large size of modern wind turbines, test budgets and equipment limitations mean that models typically require significant scaling, which makes it difficult to accurately maintain the dynamic characteristics of the structure. The problem of cost is further exacerbated when performing failure testing, as it limits the number of allowable tests.

The following subsections will introduce wind tunnels and shaking tables – two common experimental testing environments for wind turbine studies, and will discuss the advantages and limitations of each. Finally a representative collection of recent examples of experimental wind turbine studies will be presented and analyzed.

2.5.1 Wind tunnel experiments

Wind tunnel testing is the standard experimental technique for testing a wind-loaded object, including wind turbines. Wind tunnels are typically the most accurate method for studying wind loads; only well-done CFD simulations can hope to achieve a similar level of accuracy. Wind tunnels are very effective for testing wind turbines where scaling is not required, such as testing a segment of a blade (Sheldahl and Klimas, 1981; Selig and McGranahan, 2004) or a full-sized residential turbine (Shirazadeh et al., 2021). The largest full-scale wind tunnel study

of a turbine was performed at the Ames Research Center tunnel (which houses the largest wind tunnel in the world) on a 20 kW turbine (Leishmann, 2002).

Unfortunately, due to the large size of industrial wind turbines, model scaling is required to test them. The airfoils of the wind turbine blades present a unique challenge for wind tunnel studies in particular, which is detailed below. Furthermore, by requiring model scaling, aeroelastic simulation becomes more challenging as the smaller the scale the more difficult it becomes to match the dynamics of the full-sized structure. Ultimately, the issues around wind tunnel testing of wind turbines could be alleviated if turbines could be tested at a larger scale, or if accurate wind loads could be generated separately and the wind tunnel avoided entirely.

2.5.1.1 Airfoil scaling errors in wind tunnel testing

As detailed previously in Section 2.3.2, the lift and drag on an airfoil is a function of the Reynolds (Re) number, a nondimensional value that characterizes the turbulence of a flow. Equation 2.13 shows the calculation for the Reynolds number:

$$Re = \frac{UL}{\nu} \quad (2.13)$$

Here, U is the wind speed, ν is the kinematic viscosity of the air, and L is the characteristic length. For airfoils, the characteristic length is the chord length.

The Re -sensitivity of airfoils is easily shown in a plot of angle of attack versus the drag coefficient for a number of flows with different Reynolds numbers. Figure 2.8 plots the drag coefficient curve of the NACA-0012 airfoil in a number of different flows using data generated by the program XFOIL. Here it can be seen that there are large differences between the predicted drag coefficient at a given angle of attack depending on the Reynolds number of the flow.

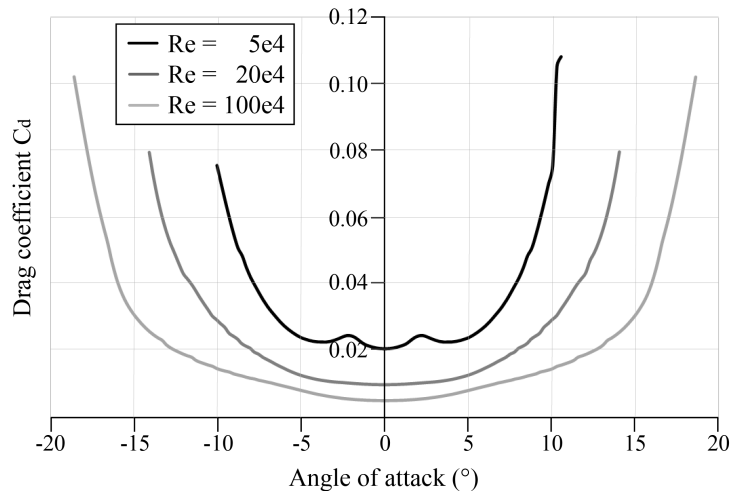


Figure 2.8: Drag coefficient plots for the NACA 0012 airfoil in flows of varying Re number

Now consider the effect scaling has on the characteristic length in Equation 2.13. If a 1:10 scale model is used, L will be reduced by a factor of ten, meaning that Re will likewise be reduced by a factor of 10. For airfoils this 10x reduction in the Re will result in different drag

and lift behaviour on the model airfoil, introducing error into the simulation. Alternatively, to keep Re at the same value, the wind speed U could be increased tenfold. This would maintain Re similitude, however if the true turbine is in wind speeds of 20 m/s, this would require generating 200 m/s winds in the tunnel, which is not feasible. Likewise, modifying the viscosity of the flow is impractical. Thus, this error is unavoidable when placing a geometrically-scaled turbine model in a wind tunnel.

Wind tunnel studies of traditional bluff bodies avoid this issue as they lack the Re -sensitivity that airfoils have. Provide the Re number remains above a certain value, the drag on a rectangular building is independent of the Reynolds number. Instead, when wind tunnel tests of scaled wind turbine models are performed, the tip speed ratio (defined previously in Equation 2.10) is used for scaling (Stein and Kaltenbach, 2016) and blades with different airfoils which are designed for lower Reynolds number flow are used to approximately simulate thrust and torque loads (McTavish et al., 2013), a technique used in many of the studies presented in the following section.

Further consideration must be given to nondimensional parameters when studying offshore wind turbines in wind tunnels equipped with wave basins. Hydrodynamic similitude is achieved in such a test by scaling according to the Froude (Fr) number, defined in Equation 2.14:

$$Fr = \frac{U}{\sqrt{gL}} \quad (2.14)$$

Here, U is the flow speed, g is gravitational acceleration, and L is the characteristic length of the turbine. By comparing Equations 2.13 and 2.14, it can be seen that u and L occupy opposite sides of the division sign, making Fr -scaling fundamentally incompatible with Re -scaling. Thus, redesigned blades must once again be used, with an emphasis on matching lift coefficients while accepting error from drag coefficient mismatching (Kimball et al., 2014).

2.5.2 Shaking table experiments

Shaking tables are typically used to subject experimental wind turbine models to seismic loads. Shaking tables consist of a base that can be moved back and forth, replicating ground motion during an earthquake. This is typically performed using smaller-scale turbine models, however Prowell et al. (2009) famously tested a full-scale 65 kW wind turbine on an outdoor shaking table. An additional advantage of shaking table tests are that since wind loading is typically not considered, these turbine models need not worry about scaling airfoils, making small-scale tests more feasible provided the dynamics of the turbine can be accurately captured.

The main limitation of shaking table tests for wind turbines is that seismic loading is generally of secondary concern to researchers; most wind turbine studies are primarily interested in wind loading. This limits the breath of possible testing that shaking tables offer, however some researchers have attempted to push these bounds by applying approximations of wind loads via the shaking table (Chen et al., 2015b).

2.5.3 Summary of experimental tests

A collection of modern experimental wind turbine studies is presented in Table 2.2. This table has been partially curated by the author to emphasize certain trends which will be discussed. Table 2.2 presents the study, the experimental model and the testing environment, the type of turbine (see Section 2.2), the types of applied loading, the testing indices and a general classification category.

From Table 2.2, the following trends can be drawn:

- It can be seen that in comparison to numerical testing, there is a smaller number of collected categories of experimental tests. **Behaviour evaluation** is by far the most common, but **control**, and **material/model evaluation** are also applications of experimental testing. Despite it being a very popular research topic numerically, it is difficult to perform vibration control studies on a full turbine model, primarily due to scaling limitations.
- We can see that a number of studies use full-sized wind turbines or blades for wind tunnel studies to side-step the scaling issues, though there remain some small scale tests that employ redesigned, low-Re blades.
- Some wind tunnel studies are paired with additional equipment to test floating offshore turbines, either directly using a wave basin (Kimball et al., 2014; Bayati et al., 2016) or indirectly using a surging table (Mancini et al., 2020), which is essentially a rotating shaking table. If scaling issues can be addressed, this appears to be an effective method of testing multi-loaded turbines.
- Shaking table tests are more limited in their applicability and categories of tests compared to wind tunnel tests. They are mostly restricted to earthquake loading.
- Some studies (Sim et al., 2014; Ma et al., 2015) directly apply loading to the turbine tower via actuator to avoid scale-limitations imposed by shaking tables and wind turbines, however this type of testing is limited to applying numerically-generated point loads to the top of the tower.
- There exist very few experimental studies of turbines subjected to high-intensity wind loading, again likely stemming from the difficulty in modelling aeroelasticity in small-scale wind tunnel models.

2.6 Overview of hybrid simulation

Hybrid simulation is a relatively recent research technique that combines numerical and experimental methods simultaneously with the goal of making use of the advantages of each systems while minimizing their limitations. As previously presented, experimental testing excels at capturing complex nonlinear behaviour, but full-scale testing is often infeasible and small-scale testing presents challenges. Opportunely, complex structural behaviour is typically restricted to small portions of the structure, which allows for more affordable numerical simulations to be used to model the remainder of the structure. This is the conceptual basis for

Paper	Model	Testing environment	Turbine type	Load types	Testing indices	Category
Leishmann (2002)	Full-scale 12 m turbine	Wind tunnel	Onshore	SW	Blade and base response and loads	Behaviour evaluation
Selig and McGranahan (2004)	Large-scale blade sections	Wind tunnel	Turbine blade	SW	Blade loads	Behaviour evaluation
Powell et al. (2009)	Full-scale 22 m turbine	Shaking table	Onshore	EQ	Modal frequencies, top response	Behaviour evaluation
Refan and Hangan (2012)	Full-scale 1.9 m turbine	Wind tunnel	Residential	SW	Power	Behaviour evaluation
Imraan et al. (2013)	0.5 m telescopic blades	Wind tunnel	Turbine blades	SW	Blades loads	Behaviour evaluation
Kimball et al. (2014)	1:50-scale 90 m turbine	Wind tunnel with wave basin	Floating offshore	SW, WV	Blade loads, platform response	Behaviour evaluation
Sim et al. (2014)	Full-scale 22 m turbine tower	Loads applied by actuator	Onshore	AR	Top response	Behaviour evaluation
Chen et al. (2015b)	1:1.3-scale 100 m turbine	Shaking table	Fixed offshore	SW, WV, EQ	Top response	Vibration control
Ma et al. (2015)	1:1.5-scale 100 m prestressed concrete turbine tower	Loads applied by actuator	Onshore	AR	Top response, tower stress	Material evaluation
Navalkar et al. (2015)	Small-scale 120 m turbine	Wind tunnel	Onshore	SW	Blade loads	Control
Abdelkader et al. (2016)	1:150-scale 90 m turbine	Wind tunnel	Onshore	HW	Base loads	Behaviour evaluation
Bayati et al. (2016)	1:75-scale 120 m turbine	Wind tunnel with wave basin	Floating offshore	SW, WV	Top response	Model evaluation
Campagnolo et al. (2016)	Six 1.1 m turbines	Wind tunnel	Onshore	SW	Power	Control
Mao et al. (2018)	1:20-scale 65 m turbine	Shaking table	Onshore	EQ	Top response, base loads	Behaviour evaluation
Dai et al. (2020)	1:100-scale 65 m turbine	Shaking table	Onshore with SSI	EQ	Top response	Behaviour evaluation
Mancini et al. (2020)	1:75-scale 75 m turbine	Wind tunnel with surging table	Floating offshore	SW, WV	Base loads	Behaviour evaluation
Shirazadeh et al. (2021)	Full-scale 1.9 m turbine	Wind tunnel	Residential	SW	Base loads	Behaviour evaluation

Table 2.2: A collection of experimental wind turbine studies. Load types are artificial (AR); service wind (SW); high-intensity wind (HW); and wave (WV)

hybrid simulation. The section presents the development of hybrid simulation, along with a mathematical explanation of the different types of HS, the importance of control for this type of testing, and finally a curated list of previous hybrid simulations that emphasize the trends of this technique. HS as applied in wind turbine studies will be discussed later in this chapter.

2.6.1 Development and numerical explanation

The historical development summarized here is an abridged version. For a more detailed description, readers are directed to McCrum and Williams (2016). To fully describe hybrid simulation, an understanding of the numerical integration methods used for structural analysis is required. In numerical structural analysis, a structure is simplified into a series of DOFs based on the scope of the research. Connections between DOFs represent structural properties. A simple example could be to approximate a water tower as a single degree of freedom lumped mass structure, connected to a fixed base via a beam with stiffness and damping. To solve for structural response due to loading, the standard equation of motion (EOM) of a structure is used, shown here in Equation 2.15:

$$M\ddot{x}_i + C\dot{x}_i + Kx_i = F_i \quad (2.15)$$

Here, M , C , and K are the matrices that describe the mass, damping, and stiffness of the structure, respectively; \ddot{x}_i , \dot{x}_i , and x_i are the acceleration, velocity, and displacement vectors of the structure at time step i respectively; and F_i is the vector of the applied force on the structure at time step i . By employing a numerical integration technique, by knowing the state of the system at time step i and the applied load at time step $i + 1$, the structural response can be predicted at step $i + 1$, which can then be used to predict the response at $i + 2$, etc. This process is continued for all time steps of the analysis to determine the time history response of the structure.

Pseudo dynamic testing (PDT) was the original form of hybrid simulation. It was conceived by Hakuno et al. (1969) and developed by Takanashi (1974), as explained in a summary by Takanashi and Nakashima (1987). Originally, the entirety of the structure was modelled both experimentally and numerically. The same numerical integration process described previously occurs, however, the equation of motion of the structure is slightly modified, as shown in Equation 2.16:

$$M\ddot{x}_i + C\dot{x}_i + F_{Ri} = F_i \quad (2.16)$$

Here, compared to Equation 2.15, the new term F_{Ri} has been added, which refers to the restoring force vector. In PDT, the same numerical process is used to predict structural displacement x_i , but then this predicted displacement is applied to the experimental model using actuators. The measured force of the structure resisting this displacement, F_{Ri} is then returned to the numerical model and used to predict x_{i+1} . This process is repeated for all time steps. Figure 2.9 illustrates this process using the previous water tower example – the numerical model predicts the displacement that is applied to the experimental model which, in turn, returns the restoring force. Pseudo dynamic testing has improved structural stiffness estimation as F_{Ri} is more accurate than Kx_i , particularly for plastic behaviour. Since this loading is performed quasi-statically (i.e. loads are applied slowly such that the velocity and acceleration of the physical structure

are essentially zero), the inertial and damping forces of the structure are exclusively simulated in the numerical model, and thus the experimental component of pseudo dynamic testing is not appropriate for testing structures with velocity- or acceleration-dependent behaviour such as vibration dampers.

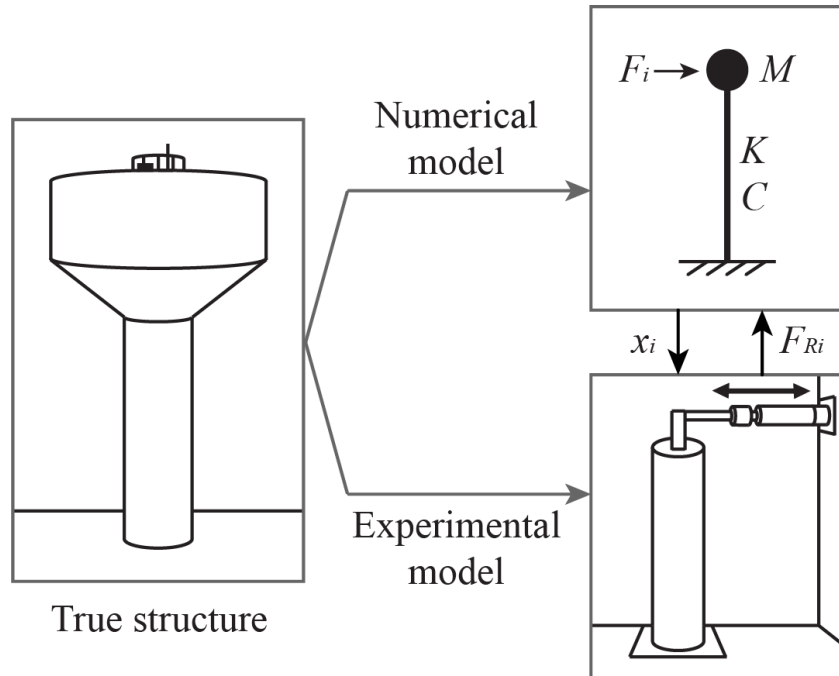


Figure 2.9: Theoretical pseudo dynamic test of a water tower

The development of real-time hybrid simulation (RTHS) was an important leap for hybrid simulation as it greatly expanded the capabilities of the research method. Since this type of testing is run at real-time, velocity- and acceleration-dependent behaviour can be captured in the physical model which improves accuracy and allows for additional types of structures to be tested. In RTHS, the restoring force measured from the physical model may include stiffness, damping and/or inertial forces, depending on the test. Historically, the first example of real-time hybrid simulation was performed by Nakashima et al. (1992). Real-time hybrid simulation faces new challenges compared to PDT, primarily that the numerical model must now be sufficiently lightweight and on sufficiently powerful equipment to run in real time. Both real-time hybrid simulation and pseudo dynamic testing still see wide use in structural engineering research, as will be shown below.

The final major evolution of hybrid simulation was the development of substructuring by Dermitzakis and Mahin (1985). Substructuring allows multiple partial sections of the structure to be tested separately from one another to optimize testing. Typically, this means that the experimental substructures will consist of sections of interest of the structure where nonlinear behaviour will occur, while the rest of the structure is numerically modelled. Figure 2.10 shows a theoretical substructured hybrid simulation, similar to the test performed by McCrum and Broderick (2013), consisting of a multi-bay steel frame with a single braced bay, where the braced bay is physically modelled and the rest of the frame is modelled numerically. Both substructures are tested simultaneously and information is passed between them. Substructuring

allows hybrid simulation to be used to perform large-scale experimental tests without physically modelling the entire structure, thus unlocking the full potential of the testing technique. As a result, substructuring is employed in nearly all examples of modern hybrid simulation. Distributed hybrid simulation is a specialized form of substructuring that tests multiple substructures simultaneously in separate facilities, which allows researchers to take advantage of specialized equipment available at multiple different sites (Watanabe et al., 2001; Spencer et al., 2004; Wang et al., 2008; Ojaghi et al., 2014). However, significant timing issues are introduced with distributed hybrid simulation, and as such it sees only occasional use.

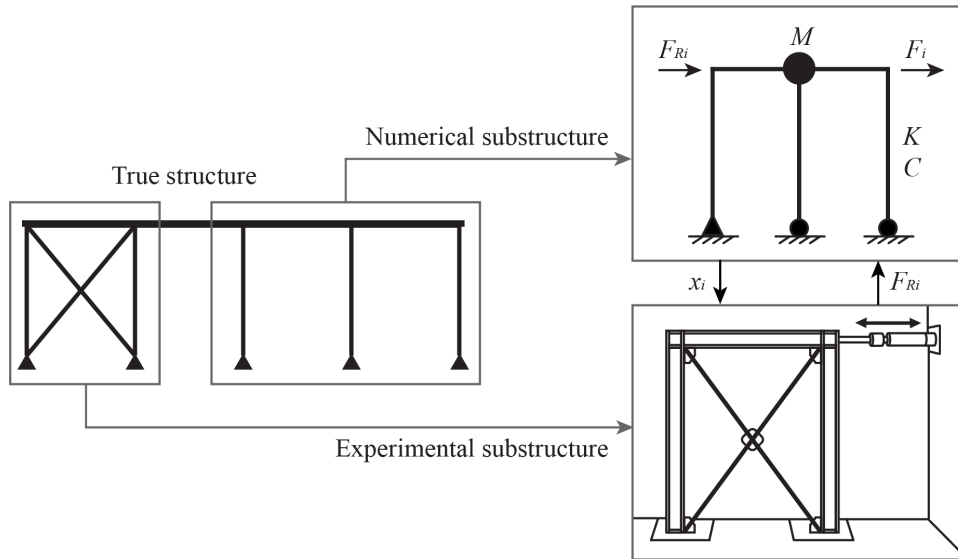


Figure 2.10: Theoretical substructured hybrid simulation of a one-story braced frame

The hybrid simulation process shown in Figure 2.10 is the more common technique known as displacement-control; target displacements are applied to the experimental substructure and the resulting restoring force is measured and used by the numerical model. However, hybrid simulation can also be performed via force-control: where a target force is applied to the experimental substructure via actuators and the resulting displacement is measured and returned to the numerical model. A comparison of the equations of motion of the structure using these two methods can be found in Plummer (2006). Force-control can result in more precise control of actuators (Yalla and Kareem, 2007) which is critical for fields such as robotics, and is applied in civil engineering such as in the control of some shaking tables (Shao et al., 2011) and barges (Ueland et al., 2018; Vilsen et al., 2019). Still, the vast majority of hybrid simulation research uses the simpler and more stable displacement-control, as will be explored in Section 2.6.3.

2.6.2 Control in hybrid simulation

Controlling error is critical for accurate simulations. Due to the severe time requirements of real-time hybrid simulation (RTHS), numerical algorithms for the the numerical substructure as well as controllers for the experimental actuators must be carefully selected. Experimental and systematic errors risk compounding over the course of a test, ruining the results.

Numerical integration algorithms are used to solve the differential equation of motion of a structure in dynamic analysis, as discussed in Section 2.6.1. These algorithms can either be implicit or explicit: implicit algorithms iterate to find a solution while explicit algorithms calculate a solution directly. Explicit algorithms have lower accuracy and stability but are much quicker to compute, and as a result, RTHS almost exclusively employs explicit numerical algorithms. However, since errors from these methods must be minimized and stability must be guaranteed as much as possible, older explicit algorithms (Newmark, 1959) see infrequent use in favour of more modern and complex algorithms (Ahmadizadeh and Mosqueda, 2007; Chen et al., 2009; Kolay et al., 2015; Tang and Lou, 2017; Kolay and Ricles, 2017). The CR algorithm by Chen et al. (2009) is highlighted as simpler method with generally acceptable accuracy that generates an integration parameter based on the structural properties to determine the explicit solution. While some have argued that this CR-algorithm risks high-frequency noise (Chang, 2012), the authors have convincingly defended the accuracy of their algorithm (Chen and Ricles, 2012). The MKR- α algorithm by Kolay and Ricles (2017) is likewise highlighted as a more complex but more robust explicit algorithm that employs a number of integration parameters from the structural parameters.

Controllers are also required for the experimental actuators in a RTHS. All equipment has some amount of time delay between sending a command signal and the equipment responding. Since RTHS is run at such small time steps, the impact of this response lag error can be significant. As a result, many researchers have developed actuator controllers to minimize the response lag error (Mosqueda et al., 2007; Lim et al., 2007; Phillips and Spencer, 2012a,b; Chae et al., 2013). Unlike the numerical algorithm, however, the choice of controller will depend on the experiment equipment used in the testing, and as a result, no single controller has become the standard.

2.6.3 Example hybrid simulation applications

There have been hundreds of examples of the application of hybrid simulation in the field of structural engineering. As the focus of this chapter is on hybrid simulation as it relates to wind turbine studies, a collection of non-turbine hybrids simulations has not been collected. Rather, the main categories of hybrid simulations will be highlighted with representative, modern examples. Hybrid simulation of wind turbines will be explored in Section 2.7.

The author has chosen to categorize the typical uses of hybrid simulation as **behaviour evaluation**, **control**, and **material evaluation**. These categories are identical to the most common categories found in the collection of experimental tests in Table 2.2. Thus, at its most basic level, hybrid simulation increases the scope or scale of experimental techniques where a pure-numerical simulation would be less effective. A representative study is provided here for each category:

- HS is used for behaviour evaluation for structures with highly non-linear behaviour, such as member failure. Ramos et al. (2016) studied the failure behaviour of a four-story steel frame under seismic loads using pseudo dynamic testing, and concluded that hybrid simulation was an effective way of capturing this behaviour with limited budget and equipment.

- HS is used for control, particularly vibration control, when the control system is challenging to model, such as a magnetorheological (MR) damper which is known for its esoteric hysteresis curve. Friedman et al. (2014) studied the effect of MR dampers on the vibration response of a nine-story steel frame subjected to seismic loading, where a three-story damped steel frame acted as the physical substructure while the remainder was numerically simulated.
- HS is used for material evaluation when the material properties are not well known and thus can not be modelled numerically. Hashemi et al. (2017) used pseudo dynamic hybrid simulation to test carbon fiber reinforced polymer (CFRP) repair of seismically damaged reinforced concrete columns. The base column of a five-story moment-resisting frame was loaded to failure and repaired, then its fragility was analyzed.

As hybrid simulation was originally developed for the field of seismic engineering, there exist very few example of HS that consider other forms of loading. In fact, the vast majority of hybrid simulations considering wind-loaded structures study wind turbines specifically, as will be seen in the following section. However, the author wishes to highlight the recent RTHS of an aeroelastic building model in a wind tunnel by Moni et al. (2020). Here, the entire structure was modelled numerically in the wind tunnel, but a numerical model was used to predict base rotation which was applied in real time to the wind tunnel model. While the scaling issue of wind turbines would still need to be accounted for, this type of testing could potentially be applied in future wind turbine studies, though this direction is not pursued in this thesis.

Ultimately, hybrid simulation can improve the scalability and fidelity of experiments, reduce equipment requirements, and validate numerical models. It excels at testing nonlinear behaviour, such as from complex vibration dampers or member failure, as well as unknown behaviour. Hybrid simulation remains a promising candidate for future wind turbine research.

2.7 Hybrid simulation of wind turbines

Hybrid simulation has been applied to study wind turbines in a few instances. As previously discussed, experimental testing of offshore turbines is particularly challenging due to a scaling mismatch between aerodynamic and hydrodynamic forces. Partially as a result of this, the majority of turbine research employing hybrid simulation has studied offshore turbines.

There are several examples of studies on the development and optimization of hybrid simulation techniques for turbines. Two studies (Bachynski et al., 2015; Karimirad and Bachynski, 2017) examined the effect of limited actuation of wind loading in hybrid simulations of floating and fixed offshore wind turbines, finding which DOFs were required to accurately model wind response and which could be safely ignored. Hall et al. (2014) ran numerical simulations in OpenFAST of various floating turbine designs under different loading conditions to predict the required performance specifications of test equipment to ensure minimal error in theoretical hybrid simulations of offshore turbines. Additionally, Koukina et al. (2015) developed a custom loading device for applying numerical wind loads to a floating offshore wind turbine during a RTHS, allowing for more accurate loading in this specific case.

Along-side this preparatory research, there are a few examples of actual hybrid simulation of floating offshore wind turbines. Azcona et al. (2014) generated wind loads numerically

and applied them to a physical floating turbine model using a fan, though the accuracy of this loading method was quite low. Additionally, Chabaud (2016) developed a framework to test floating offshore wind turbines. Numerically calculated wind loads were applied using a custom 6-DOF actuator while wave loads were applied using a wave basin, which avoided the scaling problem between hydrodynamic and aerodynamic loads. Song et al. (2020) performed a thorough study of an offshore turbine tested in a artificial/conceptual hybrid simulation: a purely-numerical simulation that approximates a true HS by separating the structure into multiple numerical substructures and applies artificial delay and noise error. In this study, they modelled fixed offshore turbine with an “experimental” tower and numerical rotor evaluating its sensitivity to delays, noise, misalignment and other types of testing error.

There also exist two particularly notable cases of HS of fixed offshore turbines. Brodersen et al. (2016) performed RTHS of a shallow water offshore turbine equipped with a hybrid damper. The goal of the study was to determine the vibration reduction effect of hybrid dampers, which involved the development of a simplified numerical turbine model. The physical substructure consisted of the hybrid damper, thus real-time testing was used for accurate loading. The hybrid damper was shown to surpass passive dampers when tuned correctly, and the results of the hybrid simulation showed good agreement with a concurrent numerical test. Similarly, a research group has released two similar, detailed papers analyzing large wind turbine with tuned liquid dampers (TLDs) using RTHS (Zhang et al., 2016, 2017). These papers detail the real-time hybrid simulation of TLDs aligned to reduce the across-wind vibration of the nacelle and compares the results of these hybrid simulations to those from an equivalent numerical simulations. The TLD was physically built in a lab while a previously developed (Zhang et al., 2014b, 2015) turbine model was numerically simulated. TLDs are very challenging to model numerically due the nonlinear behaviour of liquids, such as waves breaking or splashing, though models with reasonable accuracy have been developed using simplified assumptions (Tait et al., 2008). It was shown that the TLDs were effective in reducing across-wind vibrations, and that there was good agreement with the numerical model for less nonlinear behaviour but less agreement when behaviour was more nonlinear – such as when the wind speed was high and the TLDs included screens.

All of the studies presented here fall into the categories of **behaviour evaluation** and **vibration control** that have been discussed previously. HS has been applied to a handful of types of turbine research to good effect. Both conceptually and in practice, hybrid simulation allows for research whose accuracy would otherwise suffer in numerical testing, and would be infeasible using experimental testing. That being said, there exists a major limitations to the research presented here: the simplified aeroelastic models and the act of pre-calculating wind loads. These errors are defined and quantified in the following section.

2.8 A study on the limitations of HS of wind turbines

The existing HS of wind turbines presented in the previous section have two major limitations: they employ simplified aeroelastic models and pre-calculated wind loads. Simplified aeroelastic models refers to the blade models used in many numerical wind turbines simulations (as well as the numerical substructures in HS) that are lower accuracy compared to robust 3D models. The reduced accuracy of the more common BEM technique compared to CFD is an

example of this trend, which was touched upon in Section 2.4. “Pre-calculating” wind loads consist of numerically predicting the structural loads from the test wind field prior to the HS itself, then applying the time history of these loads in the RTHS. Doing so allows the time-consuming aeroelastic calculations to be performed ahead of time, which facilitates real-time testing, but by doing so, the aeroelasticity of the turbine is fundamentally uncoupled from the experimental model in the HS proper. Figure 2.11 compares the components and processes of a theoretical RTHS in both ideal conditions and the conditions employed in current studies: the wind field time history V_w applies structural loads F_s which cause the structural response x_s . In the ideal case, the F_s calculation would be performed using a model with high aeroelastic fidelity within the HS loop, but historically they have been performed with simplified models outside of the loop ahead of time. It is plain to see that both limitations will inject some amount of error into a simulation, the question is how much error.

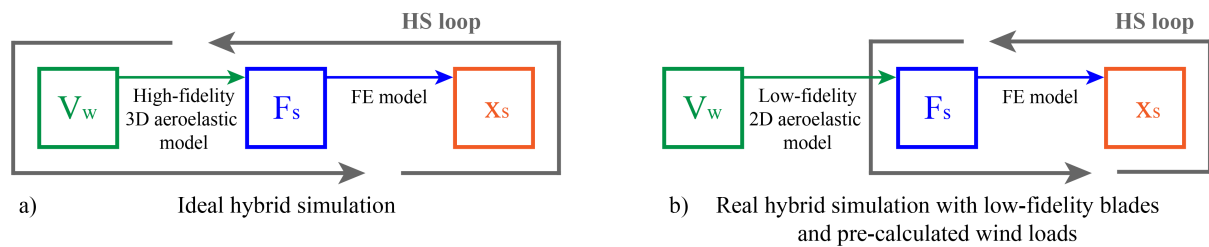


Figure 2.11: Comparison between an ideal hybrid simulation and a real example with simplified aeroelastic model using pre-calculated wind loads

Unfortunately, there is a lack of research that attempts to quantify the error caused by these limitations, which provides the motivation of this study. To explore the limitations present in previous HS of wind turbines, a short numerical study of the effects of aeroelastic blade deflections on the overall response of a wind turbine was performed. This study presents a representative case study that serves to quantify the error caused by simplifying or decoupling aeroelasticity in wind turbine blades as well as by pre-calculating wind loads. These tests do not employ true hybrid simulations, rather the results of these numerical tests can be used to quantify errors in the existing hybrid testing framework.

For convenience, aeroelastic blade models are typically simplified by assuming that only small deflections occur in turbine blades, this includes the default settings in common turbine modelling software such as OpenFAST. However, it has been shown that the small-deflection assumption is invalid for large multi-megawatt wind turbine blades (Rasmussen et al., 2003; Wang et al., 2016). When investigating the impact of this simplification, Wang et al. (2014) found that notable error ($>10\%$) occurred at high wind speeds in numerical simulations of a turbine blade due to the small deflection assumption. As far as the author is aware, there have been no studies into quantifying the impact of pre-calculating wind loads.

2.8.1 Methodology

The goal of the parametric study summarized here was to investigate the effect of aeroelastic deflection of wind turbine blades in a multi-megawatt turbine equipped with a tuned mass

damper (TMD) on the accuracy of the response of the overall structure. Specifically, the effects of two modelling practices were studied: the level of blade deflection assumed and the use of pre-calculated wind loads.

To study the effects of aeroelastic blade simplifications, three levels were considered during testing:

- **Rigid blade models (RIG):** by ignoring blade deflections entirely, the computational requirements of blade models can be greatly reduced. As seen in table 2.1, rigid blades are often employed in studies of wind turbine farms where the far-field wake is of greater importance, as well as modern CFD studies of turbine blades where aeroelastic analysis is not feasible.
- **Small-deflection-assumed blade models (SDA):** by assuming deflections in the wind turbine blades are small, the axial wind effects can generally be ignored as well as enabling the use of small-angle approximation in numerical models. Such techniques have been widely-used historically but may overestimate power production and flapwise blade deflection, as well as underestimate pitch moments at the blade roots (Rasmussen et al., 2003).
- **Fully-flexible blade models (FLX):** modelling full flexibility in wind turbine blades increases the complexity of the numerical model but eliminates possible error from the previous approximations and allows for consideration of 2nd-order effects. These models are becoming more commonly applied for studies of multi-megawatt turbines (Liu et al., 2018) and is an option in popular software such as OpenFAST (Wang et al., 2017).

To study the effects of pre-calculating wind loads, this research considered two extreme cases to find the bounds of error this may cause:

- **No numerical damper (NND):** in these cases the numerical model used in the pre-calculation of the wind loads does not include the TMD, and thus the effect of the TMD on the aeroelastic behaviour is totally decoupled in this HS. This represents the scenario with the greatest expected error, where the predicted aeroelastic response should deviate from the true response the most.
- **Perfect numerical damper (PND):** in these cases the numerical model includes a perfectly-accurate numerical representation of the TMD, and thus the exact effect of the TMD on the aeroelastic behaviour of the structure is captured in the hybrid simulation. This case does not exist in reality, as the existence of a PND model would preclude the need for the hybrid simulation in the first place, but it will be used as a baseline to gauge the effect of the NND model case.

NREL's reference 5 MW wind turbine (Jonkman et al., 2009) was used as the test structure in this research. This turbine has a hub height of 88 m and a blade length of 62 m. It has cut-in, target, and cut-out hub height wind speeds of 3 m/s, 11.4 m/s, and 25 m/s respectively, and a rated rotor speed of 12.1 rpm. While this structure is typically studied as an offshore turbine, for the purposes of this study it was assumed to be onshore in open terrain with a fixed foundation.

Mass (kg)	10200
Stiffness (N/m)	142000
Damping coefficient (Ns/m)	9860

Table 2.3: TMD properties

A passive 2D TMD was placed at the top of the turbine tower to act as an external damping system for much of the testing. TMDs add an additional mass to the structure capable of moving independently to reduce the overall response of the main structure. The natural frequency of the TMD is tuned to match that of the main structure by adjusting its mass and stiffness. The TMD parameters shown in Table 2.3 were selected according to TMD design guidelines from Bakre and Jangrid (2007), based on a mass ratio between the TMD and the wind turbine of 4%, a natural frequency of 3.73 rad/s, and a TMD damping ratio of 10%.

The wind fields used in this testing were generated using the open-source program TurbSim (Kelley and Jonkman, 2012) following standard wind turbine design guidelines (IEC, 2005). Service-level wind fields with mean hub height wind speeds of 5 m/s, 10 m/s, 15 m/s, and 20 m/s were generated using the Kaimal turbulence model with turbulence intensity of 0.16 and a power law profile with an exponent of 0.2, over open terrain with a roughness length of 0.05 m. To ensure stability of all blade models, a small time step was used; the generated wind field time histories were 90 s long with 0.0005 s time steps.

The open-source turbine modelling software OpenFAST was employed for studying the effect of aeroelasticity of blade models, as well as used to pre-calculate wind loads. The differing aeroelastic fidelities of the blade models were captured using the following modules: rigid RGD blades were modelled by disabling the DOFs of each blade within the standard ElastoDyn module. The small deflection SDA blade models used the standard ElastoDyn module, which is based on straight Euler-Bernoulli beams limited to moderate deflections. The fully-flexible FLX blades were modelled with the BeamDyn module (Wang et al., 2017). The BeamDyn module uses the geometrically exact beam theory which is more effective at capturing the behaviour of the highly flexible, composite wind turbine blades compared to the ElastoDyn module.

For the pre-calculated wind load cases, the time histories of the structural turbine tower loads were extracted from OpenFAST and applied to a custom FE turbine model. This technique simulates the two-step process used in real-time hybrid simulations with pre-calculated wind loads, and can be considered a simplified version of an artificial hybrid simulation, such as was performed in Song et al. (2020). This custom-built 80-DOF FE turbine model was run in MATLAB. The FE model of the turbine tower was built of twenty 8-DOF beam elements. The mass of the nacelle and turbine blades was lumped at the top of the tower, and the TMD in the nacelle was programmed to act against the fore-aft and side-side lateral displacements of the top of the tower.

Table 2.4 lists the properties of the beam elements in the turbine tower, and Table 2.5 lists the other assorted properties of the turbine. The model used Rayleigh damping tuned to the first two modes, plus additional damping in fore-aft direction to capture the aerodynamic damping caused by rotating turbine blades, a technique employed by Zhao et al. (2019b) and Lalonde et al. (2020b). This setup was employed to approximate the 5% fore-aft and 1% side-side damping identified in previous research (Mardfekri and Gardoni, 2015). To eliminate the

Tower section	Element length (m)	Flexural rigidity (10^3 Nm^2)	Linear density (10^9 kg/m)
1 (top)	4.38	2.66	1.29
2	4.38	2.79	1.43
3	4.38	2.92	1.57
4	4.38	3.06	1.73
5	4.38	3.19	1.89
6	4.38	3.33	2.08
7	4.38	3.47	2.26
8	4.38	3.62	2.47
9	4.38	3.77	2.69
10	4.38	3.92	2.92
11	4.38	4.07	3.16
12	4.38	4.23	3.43
13	4.38	4.39	3.71
14	4.38	4.55	4.01
15	4.38	4.72	4.31
16	4.38	4.89	4.65
17	4.38	5.06	4.99
18	4.38	5.24	5.37
19	4.38	5.41	5.75
20 (bottom)	4.38	5.50	5.94

Table 2.4: Tower FE properties

Structural damping ratio	0.01
Fore-aft aerodynamic damping coefficient (Nm)	35500
Combined nacelle and hub mass (kg)	76800

Table 2.5: Other FE model properties

greater number of structural modes considered in the FE model compared to OpenFAST as a source of error, a version of the FE model that used modal analysis of the first two modes was compared to the full FE model. No significant difference in response was seen, thus OpenFAST's assumption that the first two modes govern the response appears valid, as is the use of the FE model.

2.8.2 Testing and results

The different trials performed in this study are summarized in Figure 2.12. A total of twenty wind fields (five each at 5 m/s, 10 m/s, 15m/s and 20m/s) were generated and fifteen simulations were performed for each. In Figure 2.12:

- The first level of OpenFAST simulations considered the three levels of blade aeroelasticity with and without a TMD. Both the time history of the structural loads and the

structural response were recorded from these tests. The time history of these structural responses was primarily used to study the impact of blade aeroelasticity fidelity.

- The second level of simulations using the FE model in MATLAB was performed by applying the time history of the structural loads measured from OpenFAST. This was performed in three combinations per blade model to compare the best case (the PND case where OpenFAST and MATLAB both model the TMD), the worst case (the NND case where only MATLAB models the TMD), and a case for validating the OpenFAST and MATLAB models to one another (the VAL case where neither model the TMD). The time histories of these structural responses were primarily used to study the impact of pre-calculating wind loads.

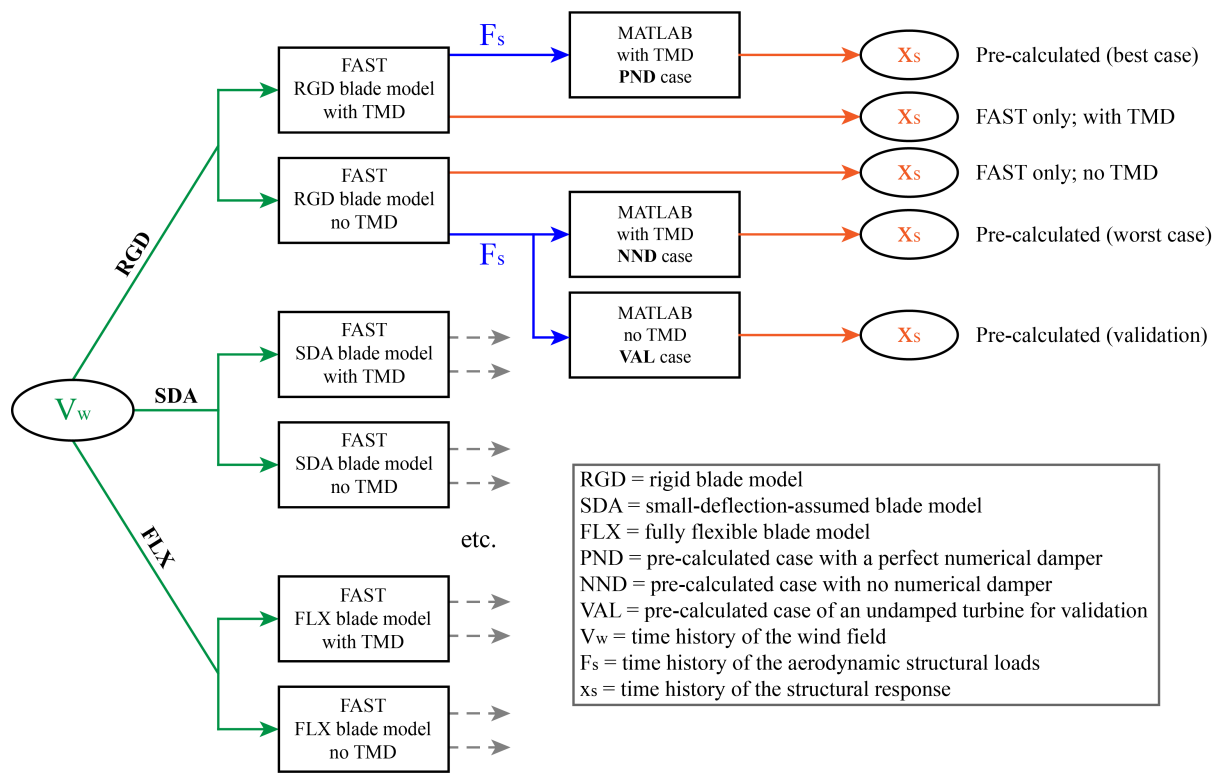


Figure 2.12: Flowchart of calculations for a given wind field time history. Note that the MATLAB simulations are only shown for the RGD blade model, but these are similarly performed for the SDA and FLX models

Tower top displacement was chosen as the performance index for this study. The time histories of the tower top displacement in the fore-aft and side-side directions were recorded for all tests and compared below in Tables 2.6-2.8. Figure 2.13 presents a selected segment of the tower top responses under a single wind load case and blade model. The values tabulated below are the mean of the five time histories for the given wind speed. Tables 2.6-2.8 use symbols due to space restrictions, they are defined as follows: \bar{v}_w is the mean hub height wind speed for the given time histories; μ is the mean displacement of the turbine tower top in

the given DOF, σ is the mean standard deviation of the displacement time histories; \bar{R} is the mean root mean square error (RMSE) between the specified time histories and \bar{R}_N is the mean normalized root mean square error (NRMSE), which are defined in the following equations:

$$\bar{R} = \sqrt{\frac{\sum_{t=1}^T (\hat{X}_t - \hat{Y}_t)^2}{T}} \quad (2.17)$$

$$\hat{X}_t = X_t - \text{mean}(X) \quad (2.18)$$

$$\hat{Y}_t = Y_t - \text{mean}(Y) \quad (2.19)$$

$$\bar{R}_N = \frac{\bar{R}}{(\max(X) - \min(X))} \quad (2.20)$$

Here, \hat{X}_t and \hat{Y}_t are the fluctuating components of the first and second specified time histories at time step t .

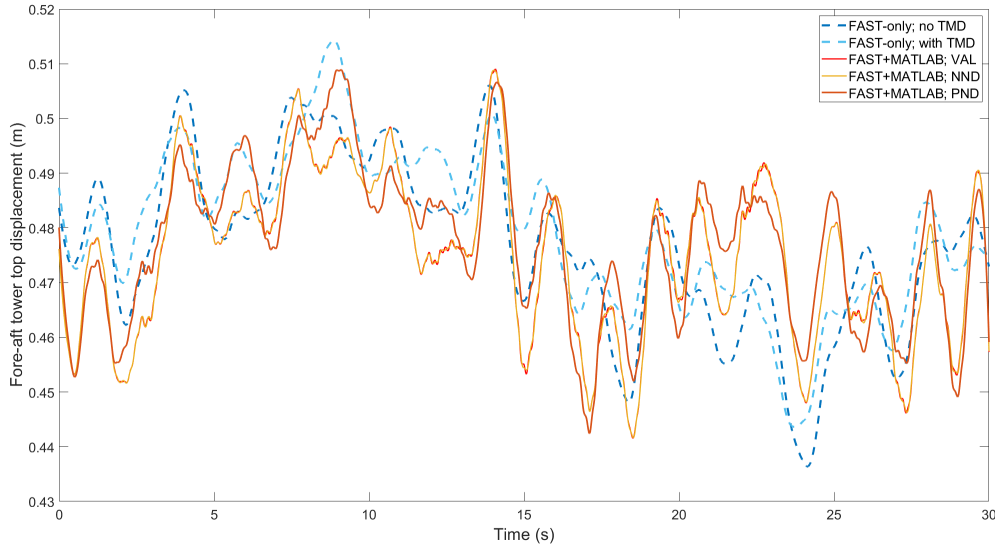


Figure 2.13: Response time history segment of tower top under 15 m/s winds with fully-flexible blade models from OpenFAST and OpenFAST+MATLAB models (see Figure 2.12)

Table 2.6 summarizes the results of the OpenFAST-only simulations, which can be used to investigate the effect of the aeroelastic fidelity of the wind turbine blade models. The tower top response with rigid (RGD), small-deflection-assumed (SDA), and fully-flexible (FLX) blade models are presented, and the RMSEs between the FLX and the other models were calculated, assuming that the FLX models provided the most accurate results. It can be seen that the RGD blade models tend to result in overestimation of the tower top response, with NRMSEs around 5-20%. The SDA blade models were more accurate comparatively with NRMSEs around 1-10%. Normalized error was greater in the side-side direction due to heightened sensitivity from the smaller vibrations, though the SDA model again outperformed the RGD model. The presence of a TMD did not have a notable effect on the results. Due to larger error, the

DOF	TMD	\bar{V}_w	RESPONSE SUMMARY						DATA COMPARISON			
			RGD		SDA		FLX		FLX vs RGD		FLX vs SDA	
			μ (m)	σ (m)	μ (m)	σ (m)	μ (m)	σ (m)	\bar{R} (m)	\bar{R}_N	\bar{R} (m)	\bar{R}_N
Fore-aft	No	5 m/s	0.103	0.011	0.101	0.011	0.091	0.010	0.004	8%	0.002	4%
		10 m/s	0.331	0.025	0.323	0.023	0.319	0.024	0.007	8%	0.002	2%
		15 m/s	0.522	0.029	0.508	0.025	0.512	0.025	0.012	9%	0.005	4%
		20 m/s	0.606	0.063	0.592	0.045	0.595	0.048	0.027	9%	0.007	2%
	Yes	5 m/s	0.104	0.011	0.102	0.011	0.091	0.010	0.003	8%	0.001	3%
		10 m/s	0.333	0.024	0.324	0.023	0.321	0.024	0.006	7%	0.001	1%
		15 m/s	0.524	0.020	0.511	0.019	0.515	0.019	0.006	6%	0.002	2%
		20 m/s	0.609	0.037	0.596	0.032	0.598	0.033	0.009	6%	0.003	2%
Side-side	No	5 m/s	-0.002	0.003	-0.001	0.003	-0.002	0.003	0.002	12%	0.001	6%
		10 m/s	-0.038	0.016	-0.034	0.015	-0.035	0.014	0.006	11%	0.004	7%
		15 m/s	-0.096	0.039	-0.088	0.034	-0.089	0.034	0.024	19%	0.009	7%
		20 m/s	-0.119	0.056	-0.112	0.033	-0.112	0.043	0.048	34%	0.014	10%
	Yes	5 m/s	-0.002	0.001	-0.001	0.001	-0.002	0.001	0.001	13%	0.001	7%
		10 m/s	-0.038	0.008	-0.034	0.007	-0.035	0.007	0.004	12%	0.001	3%
		15 m/s	-0.096	0.012	-0.088	0.012	-0.089	0.012	0.008	13%	0.002	3%
		20 m/s	-0.119	0.013	-0.112	0.009	-0.112	0.010	0.010	22%	0.003	7%

Table 2.6: Nacelle response from OpenFAST-only simulations of undamped and damped wind turbine with rigid (RGD), small-deflection-assumed (SDA), and fully-flexible (FLX) blade models

RGD model is not recommended for in-depth structural analysis, but the SDA could be a valid simplification in the right circumstances.

To validate the MATLAB model to the OpenFAST model, Table 2.7 collects the results of the VAL tests, where OpenFAST+MATLAB were combined to model the undamped wind turbine. The results were then compared to the OpenFAST-only results of the undamped turbine presented in Table 4 to generate the RMSE and the NRMSE. The results were separated between the three blade models, but it can be seen that the error between the OpenFAST+MATLAB results and the OpenFAST-only results were understandably independent of the aeroelastic fidelity of the blade model employed. Overall, very good agreement can be seen in the side-side response of the turbine, but there are some differences in the fore-aft response, particularly at lower wind speeds. This is likely due to the different methods of modelling aerodynamic damping between the models, where the aerodynamic damping in OpenFAST is a function of the velocity of the entire tower rather than just the nacelle, which reduces the damping at low speeds. Still, the behaviour of the models was deemed sufficiently similar to move forward with the second level of testing.

Table 2.8 shows the results of the OpenFAST+MATLAB tests of the damped turbine, using both the no-numerical-damping (NND) scenario and the perfect-numerical-damping (PND) scenario, which should correspond to the worst and best cases respectively. The RMSE and NRMSE were calculated between the PND and NND response time histories, assuming the PND was correct as it more closely matched the OpenFAST-only results in Table 4. Testing was performed using all three levels of blade aeroelasticity, but only the SDA and FLX results are presented here for brevity. Table 2.8 shows very clearly that the fore-aft response is similar between both techniques at low wind speeds but the error increases to over 10% at the upper limit of service wind loads. In the side-side direction, a similar trend can be seen, but there

DOF	\bar{V}_w	RGD				SDA				FLX			
		μ (m)	σ (m)	\bar{R} (m)	\bar{R}_N	μ (m)	σ (m)	\bar{R} (m)	\bar{R}_N	μ (m)	σ (m)	\bar{R} (m)	\bar{R}_N
Fore-aft	5 m/s	0.106	0.014	0.011	26%	0.107	0.015	0.011	21%	0.095	0.014	0.010	24%
	10 m/s	0.298	0.026	0.023	24%	0.296	0.025	0.020	22%	0.288	0.026	0.021	22%
	15 m/s	0.483	0.031	0.016	10%	0.474	0.027	0.014	11%	0.473	0.027	0.015	11%
	20 m/s	0.578	0.058	0.013	3%	0.562	0.043	0.012	4%	0.563	0.046	0.012	4%
Side-side	5 m/s	-0.002	0.003	0.000	1%	-0.001	0.003	0.001	4%	-0.002	0.003	0.001	5%
	10 m/s	-0.039	0.017	0.001	1%	-0.035	0.016	0.001	1%	-0.036	0.014	0.001	1%
	15 m/s	-0.098	0.041	0.002	1%	-0.089	0.036	0.002	1%	-0.090	0.036	0.002	1%
	20 m/s	-0.121	0.059	0.003	2%	-0.114	0.035	0.002	2%	-0.113	0.045	0.002	2%

Table 2.7: Nacelle response from OpenFAST+MATLAB simulations of undamped wind turbine, compared to equivalent OpenFAST-only results from Table 2.6

DOF	\bar{V}_w	SDA BLADE MODEL						FLX BLADE MODEL					
		NND		PND		PND vs NND		NND		PND		PND vs NND	
		μ (m)	σ (m)	μ (m)	σ (m)	\bar{R} (m)	\bar{R}_N	μ (m)	σ (m)	μ (m)	σ (m)	\bar{R} (m)	\bar{R}_N
Fore-aft	5 m/s	0.107	0.015	0.108	0.014	0.001	2%	0.095	0.014	0.096	0.014	0.001	1%
	10 m/s	0.296	0.025	0.298	0.025	0.002	2%	0.288	0.026	0.289	0.026	0.002	2%
	15 m/s	0.474	0.027	0.476	0.022	0.010	9%	0.473	0.027	0.476	0.023	0.010	8%
	20 m/s	0.562	0.043	0.565	0.032	0.017	11%	0.563	0.046	0.566	0.033	0.020	12%
Side-side	5 m/s	-0.001	0.003	-0.001	0.001	0.002	38%	-0.002	0.003	-0.002	0.001	0.002	35%
	10 m/s	-0.035	0.016	-0.035	0.008	0.011	34%	-0.036	0.014	-0.036	0.007	0.010	31%
	15 m/s	-0.089	0.036	-0.090	0.012	0.027	42%	-0.090	0.036	-0.091	0.012	0.027	43%
	20 m/s	-0.114	0.035	-0.114	0.009	0.029	69%	-0.113	0.045	-0.113	0.010	0.038	76%

Table 2.8: Nacelle response from OpenFAST+MATLAB simulations of damped wind turbine with SDA and FLX blade models comparing no-numerical-damping (NND) and perfect-numerical-damping (PND)

exists notable error between the NND and PND results even at the minimum wind speeds in the order of 30-40%. The larger magnitude of the NRMSE in this direction likely stems from the smaller magnitude of overall response; since the side-side response is orders of magnitude smaller than the fore-aft response, a small amount of error causes a larger change in the NRMSE. The error between the techniques appears to be independent of the level of aeroelastic fidelity of the blade models.

2.8.3 Study conclusions

The purpose of this study was to investigate the importance of aeroelastic blade behaviour on the overall response of a wind turbine to study limitations in current hybrid simulations of wind turbines. Two techniques in particular were investigated: the use of various levels of aeroelasticity in turbine blade models, and the act of pre-calculating the structural wind loads on a turbine prior to full simulation, which inherently uncouples the blade aeroelasticity. This testing was performed by simulating a 5 MW turbine under service wind conditions using a combination of the turbine modelling software OpenFAST and a custom finite element model in MATLAB.

A series of OpenFAST simulations were performed to study wind turbines equipped with rigid (RGD), small-deflection-assumed (SDA) and fully-flexible (FLX) blade models. The following conclusions were drawn from these simulations:

- The response of the SDA model more closely matched the FLX model's response than the RGD model did, suggesting an increase in accuracy as the aeroelastic fidelity increases.
- The normalized error of the SDA model compared to the FLX model was typically 1-10%, thus the use of the simpler SDA model is not unreasonable under service wind conditions, though better results would likely be achieved by including fully-flexible blade models.
- The normalized error of the RGD model compared to the FLX model was typically 5-20%, thus rigid blade models are not recommended for in-depth structural analysis of wind turbines.
- The fore-aft response error was generally unaffected by wind speed, but the error in the side-side response increased as the wind speed increased.

A series of combined OpenFAST+MATLAB simulations investigated the impact of pre-calculating wind loads on a structure in hybrid simulation. Two scenarios were compared: no-numerical-damping (NND) where the OpenFAST model that generates the structural wind loads has no TMD, and perfect-numerical-damping (PND) where the OpenFAST model has a perfect match of the TMD applied in the MATLAB model, which roughly correspond to the worst case and ideal cases, respectively. The following conclusion was drawn from these simulations:

- At low wind speeds, the difference between the NND and PND cases was very small, however the normalized error increased up to 10% for the fore-aft response at higher service wind speeds. The side-side response was even more sensitive, with errors of 70% being reached between the NND and PND simulations. Thus, future hybrid simulations should aim to replicate the PND condition as closely as possible to maximize accuracy.

In addition to highlighting current limitations in hybrid simulations of wind turbines, these values should prove useful to future researchers for designing numerical studies, finding a balance between a simple numerical model and level of accuracy sought.

2.9 Overall conclusions

This chapter has presented an introduction to wind turbine research as well as literature collections of past wind turbine research and hybrid simulations with the goal of exploring the potential application of hybrid simulation to future wind turbine research. Modern wind turbines are unique structures which can be challenging to study. Despite this, previous wind turbine studies have successfully applied both numerical and experimental research techniques. Each of these forms of testing have advantages and disadvantages such as experimental models capturing nonlinear behaviour well but facing financial and equipment limitations, while numerical testing can struggle to model this same behaviour but excels in terms of cost, versatility, and repeatability. Since nonlinear behaviour is often restricted to small sections of structures, substructured hybrid simulation can be used to physically model only these areas while the

remainder is numerically modelled. Alternatively, structures subjected to multi-hazard loading can apply certain loads experimentally and others numerically. Several examples of the capabilities of hybrid simulation, including modelling of structural failure, damped structures, and materials with unknown structural properties, have been presented.

The complexity and scaling issues of wind turbines make them prime candidates for the application of hybrid simulation for analysis. Other researchers agree with this observation as both preparatory research and actual hybrid simulations of wind turbines have been recently performed. Two sets of studies were presented of hybrid simulation of damped fixed offshore turbines where the damper was physically substructured; these were excellent examples of the utility and feasibility of hybrid simulation for wind turbine research.

Unfortunately, these studies were limited by the use of simplified aeroelastic models and the use of pre-calculating wind loads. The impact of these limitations has not been well quantified previously, thus a small numerical study is presented that attempts to evaluate the impact of this error. By studying a TMD-equipped wind turbine in a conceptual hybrid simulation, it was shown that insufficient aeroelastic fidelity can result in errors exceeding 10% and that pre-calculated wind loads can result in errors exceeding 70%. As a result, there are significant improvements to be made in hybrid simulation of wind turbines if a framework can be developed that minimizes or avoids these limitations.

Additionally, as hybrid simulation of wind turbines remains in its infancy and thus there remain many possible applications and avenues of future research of wind turbines using hybrid simulation. Several of these are proposed here:

- **Offshore turbines:** Floating offshore turbines are strong candidates for the use of hybrid simulation as it addresses the issue of the fundamental scaling mismatch between aerodynamic and hydrodynamic loads. Preparatory research, including determining performance specifications and actuation requirements, has been performed, but there are few published examples of actual hybrid simulation of these structures.
- **Multi-hazard loading:** Hybrid simulation can be used to subject wind turbines to multi-hazard loading, including wind, wave, and seismic. Previous numerical research has shown that coupling between wind and seismic loading (Asareh and Volz, 2013), as well as wind and wave loading in offshore turbines (Calderer et al., 2014; Tran and Kim, 2017) plays an important role in determining the structural response of turbines but also introduces unique challenges. Experimental testing can struggle to simultaneously apply multiple loading types, but hybrid simulation could allow for some loading to be experimentally modelled for greater accuracy while other loading is applied to the numerical model, as has been performed previously for some floating structures (Koukina et al., 2015; Ueland et al., 2018; Vilsen et al., 2019).
- **Damped turbines:** Dampers are often used to improve the service life and ultimate resistance of turbines. With the growing use of more complex, nonlinear dampers such as tuned liquid or magnetorheological, hybrid simulation can be used to physically test the damper while numerically modelling the turbine. Tests such as these are very common uses of real-time hybrid simulation as the simulation technique allows for cost-effective experimental damper research.

- Turbine failures: Hybrid simulation can be used to further study blade or tower failures of turbines, as it allows for failure testing that uses precise numerical loading and applies it to generate accurate experimental failure behaviour, while using a minimalist physical model that is less costly to replace than a purely experimental test.

In addition to these specific proposals, hybrid simulation can also be used to validate complex numerical turbine models. These possible avenues of future research are in no way all-encompassing, but merely propose some of the more conspicuous applications of hybrid simulation for wind turbine research. This is not to say that hybrid simulation should be applied in all future wind turbine research; it is merely one more highly useful tool to add to a researcher's repertoire. Future research of wind turbines will be used to help design safer and more efficient turbines.

Chapter 3

Investigation of the drag and lift coefficient curves of the NACA 0012 airfoil using CFD and a novel synthesis method

Summary

Stalling occurs on an airfoil when the incoming wind strikes it a steep angle causing flow separation. As a result, the stall and post-stall aerodynamic behaviour of an airfoil is significantly more complex compared to its pre-stall behaviour, and thus much more challenging to capture using numerical simulations. Stalled airfoils see comparatively little study as they occur infrequently in aeronautic applications. Conversely, wind turbine blades often experience stall as their maximum efficiency occurs when the incoming wind on the blades strikes at around the stall angle, and as such researchers must attempt to capture this complex aerodynamic behaviour. This study attempts to do so for the widely-studied NACA 0012 airfoil in flows with a Reynolds number of 10^5 to 10^6 – common values for industrial wind turbines. The first section of this study collects previous experimental wind turbine studies and numerical simulations that report the drag and lift coefficients curves of the NACA 0012 airfoil in the specified range of Reynolds numbers. Numerical piecewise polynomial systems are then developed based on the reported results to synthesize the average predicted drag and lift coefficients as a function of the Reynolds number. In the second section of this study, 112 CFD simulations are performed to find an optimal combination of mesh fidelity, dimensionality (2D, 3D), turbulence models (steady-state RANS, URANS), and RANS closure models (Spalart–Allmaras, k - ϵ , k - ω) for calculating coefficient curves that agree with the average predicted curves from the first section. These simulations also included a number of preliminary LES simulations; computational limitations restricted the allowable mesh density and thus the ultimate accuracy of these simulations. It was found that an effective, efficient model with decent matching to experimental results can be generated using a combination of 3D simulation, URANS momentum equations, and the k - ω turbulence closure model. This effective CFD configuration can thus be extended to future turbine airfoil simulations.

3.1 Introduction

The most basic characterization of the aerodynamic behaviour of an airfoil is through its lift and drag coefficient curves. The lift and drag coefficients (C_L and C_D , respectively) quantify the relationship between the wind speed, the geometry of a structure, and the resulting aerodynamic loads as defined in Equations 2.11 and 2.12. The lift and drag coefficient curves plot these values as a function of the angle of attack of the wind, that is, the angle between the chord line of the airfoil and the direction of the wind. While these curves are normalized across wind speeds and chord lengths, they are heavily dependent on the Reynolds number and Mach number of the flow. Both are dimensionless parameters that characterize the turbulence and speed of fluid flows respectively. While the Mach number is critical in aeronautics, its effect is often insignificant for other airfoil applications. The Reynolds number, alternately, is always relevant. The Reynolds number is a nondimensional parameters that relates the wind speed and viscosity to structure size, as defined in Equation 2.13. In the case of airfoils the chord length is used as the characteristic length for calculating the Reynolds number.

The lift and drag coefficient curves of an airfoil can be assessed through a number of experimental and numerical means. By placing a prismatic airfoil section in a wind tunnel, the drag and lift loads can be measured allowing the coefficient curves to be calculated. By rotating this airfoil, a number of angles of attack can be tested sequentially. Numerical methods specifically for characterizing 2D airfoils are also available, such as XFOIL (Drela and Youngren, 2000), though these methods tend to lose accuracy at higher angles of attack. General numerical wind engineering tools such as computational fluid dynamics (CFD) can also be used to accurately simulate flow around objects, and can be used to characterize airfoils in 2D and 3D simulations.

The NACA 0012 airfoil is perhaps the most widely studied airfoil in literature, known for its symmetrical profile and favourable lift to drag ratio. Developed by the National Advisory Committee for Aeronautics, the “00” refers to the amount of the camber in the airfoil (none) and the “12” refers to the maximum thickness of the airfoil relative to its chord length (12%). Equation 3.1 describes the normalized coordinates of the airfoil surface:

$$y = \pm 0.6 * (0.2969 \sqrt{x} - 0.1260x - 0.3516x^2 + 0.2843x^3 - 0.1015x^4) \quad (3.1)$$

Here, the x axis represents the chord line of the airfoil. The airfoil curve described by this equation is shown in Figure 3.1.

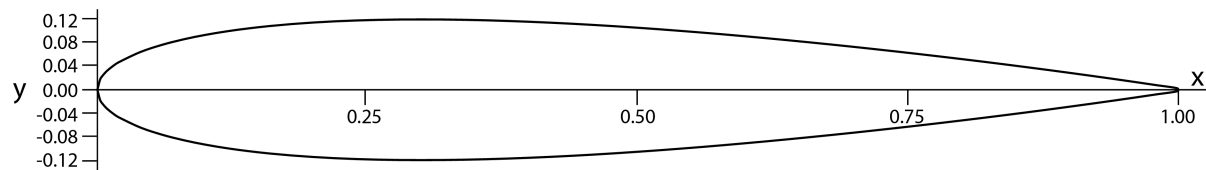


Figure 3.1: NACA 0012 airfoil with a unit chord length

This paper focuses on the behaviour of the NACA 0012 airfoil in flows with Reynolds numbers ranging from 10^5 to 10^6 . This range in particular is chosen for two reasons: firstly, the flow around industrial wind turbine blades often fall into this region. Secondly, this range captures the transition from the subcritical to the supercritical flow regimes around a cylinder,

which consists of a rapid change of drag coefficient depending on the surface roughness of the cylinder, as shown in Figure 3.2 which has been reproduced from Simiu and Scanlan (1986). This figure shows that the expected drag on cylinders can vary substantially within Reynolds numbers of 10^5 to 10^6 , and this behaviour extends from cylinders to curved airfoils as well. As will be shown, despite the NACA 0012 airfoil having a robust amount of experimental and numerical tests at a wide range of Reynolds numbers, the results from these previous studies often fail to strongly agree. If a consensus on the lift and drag coefficient curves cannot be reached for perhaps the most widely-tested airfoil, this is problematic for researchers attempting to validate new testing techniques against older results.

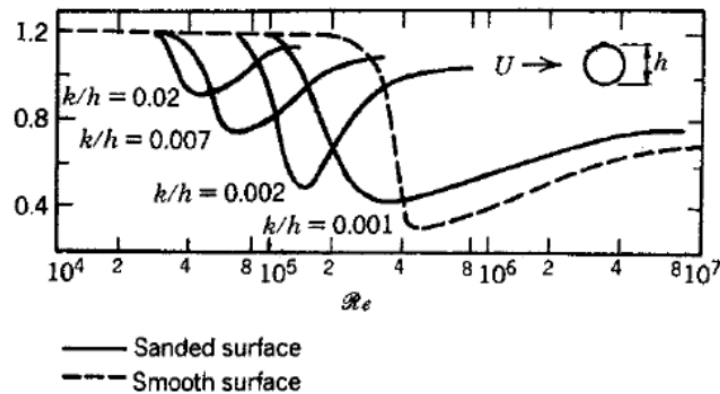


Figure 3.2: C_D on a cylinder as a function of Re , where the y -axis is C_D and k/h is a measure of the surface roughness of the cylinder. Reproduced from Simiu and Scanlan (1986) with permission from Wiley ©

This paper investigates this behaviour further in two stages: firstly, a collection of previous studies of the NACA 0012 airfoil in flows with Reynolds numbers ranging from 10^5 to 10^6 are collected and compared in Section 3.2. The drag and lift coefficient curves reported in these studies are synthesized to derive average predicted curves as a function of the Reynolds number. This is performed by quantifying and averaging trends in the plotted coefficient curves and using them to generate piecewise polynomial systems. Secondly, a number of new CFD simulations of the NACA 0012 airfoil are detailed and presented in Section 3.3. These CFD simulations were performed in both 2D and 3D using RANS, URANS and LES CFD models, and the results were then compared to the predicted average curves. By comparing the results of these two sections, conclusions are drawn and recommendations for future CFD simulations are made, which are detailed in Section 3.4.

3.2 Synthesis of previous studies of the NACA 0012 airfoil

This section presents an overview of the aerodynamic behaviour of the NACA 0012 airfoil. It subsequently collects the drag and lift coefficients of the airfoil in flows where Re is between 10^5 and 10^6 from previous wind tunnel tests and numerical simulations. Finally, a framework to develop average 2D and 3D lift and drag coefficient curves for the NACA 0012 airfoil as a function of the Reynolds number based on the existing research is presented. This consists

of identifying and quantifying trends in the coefficient curves as a function of the Reynolds number and using them to develop piecewise polynomial systems. This framework is then employed to generate four predicted coefficient curves for average lift in 3D simulations, average lift in 2D simulations, average drag in 3D simulations, and average drag in 2D simulations at $Re = 5 \times 10^5$. These are referred to as the “3D lift”, “2D lift”, “3D drag”, and “2D drag” curves, respectively.

3.2.1 Overview

Stall occurs when the flow separates from the side of the airfoil. For standard airfoils in flows ranging from $Re = 10^5$ to 10^6 , it is generally seen that low-frequency stall begins at an angle of attack that ranges from 12° to 18° . Low-frequency stall is characterized by a decrease in lift coefficient and an increase in the drag coefficient. Beyond 18° , deep or post-stall occurs, which is characterized by an increase in both lift and drag coefficients.

Wu et al. (1998) used a small number of studies of the NACA 0012 and similar airfoils to characterize their flow response at a given Re flow and angle of attack. This figure has been reproduced here in Figure 3.3. As can be seen in this figure, at Reynolds numbers of 10^5 to 10^6 , as the angle of attack increases the flow transition from steady attached to low-frequency steady separated (this transition occurs at higher angles of attack as Re increases) and then finally to turbulent vortex shedding. It can be seen that these flow transitions coincide with the stall angle and the deep stall angle where the slope of the lift coefficient curve changes from positive to negative and back to positive.

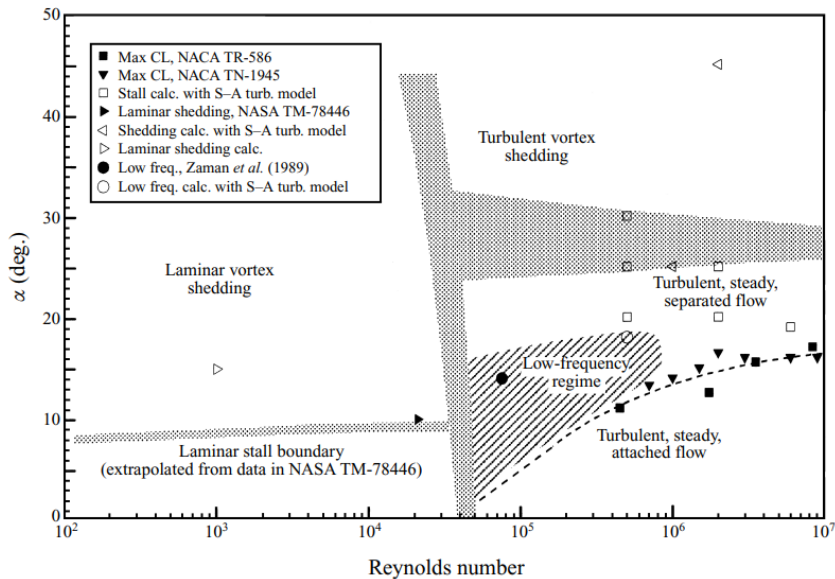


Figure 3.3: Collection of airfoil behaviour patterns based on Reynolds number and angle of attack (α). Reproduced from Wu et al. (1998) with permission from Cambridge University Press ©

The “low-frequency” behaviour identified in Figure 3.3 is adapted from Zaman et al. (1989). Low-frequency separated flow is characterized by vortex shedding at stall occurring at a Strouhal

number of 0.02 rather than the typical value of 0.15-0.20. The Strouhal number is a non-dimensional parameter that characterizes the frequency of vortex shedding:

$$St = \frac{fL}{U} \quad (3.2)$$

Here, St is the Strouhal number, f is the vortex shedding frequency, L is the characteristic chord length, and U is the flow velocity. The reproducibility of the low-frequency behaviour found by Zaman et al. (1989) was not consistent, it was only achieved by artificially increasing the tunnel turbulence to above 0.4% or by adding acoustic excitement to the flow, thus this behaviour is not guaranteed to be seen in all testing.

McCroskey (1987) has previously prepared a collection of studies of the NACA 0012 airfoil, though they focused on pre-stall studies mostly performed at higher Reynolds numbers than what is discussed in this paper. They likewise report the lack of consensus among various studies, and support the general trends reported by Wu et al. (1998). Figure 3.4, reproduced here from McCroskey (1987), reports the maximum lift coefficient at stall achieved by the NACA 0012 airfoil in different Reynolds number flows. As can be seen between this and Figure 3.3, the Reynolds number of the flow controls the angle of attack where stall occurs and, as a result, the maximum lift coefficient achievable by the airfoil.

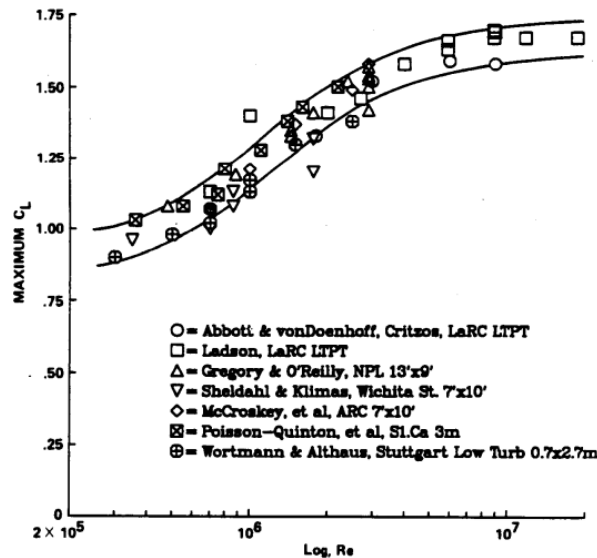


Figure 3.4: Averaged trend of maximum lift coefficient versus Reynolds number from a number of NACA 0012 studies. Reproduced from McCroskey (1987) with permission from U.S. Gov. ©

An important consideration for numerical lift and drag studies of airfoils is the difference in behaviour that is seen between 2D and 3D simulations. A 2D simulation assumes that no out-of-plane flow occurs which makes computation much more efficient, but models a less-realistic, idealized flow scenario. In reality, 3D flow along the length of the blade is a fundamental part of aerodynamic turbulence and can strongly affect the stall angle. Compared to 3D simulations, 2D simulations often achieve higher maximum lift coefficients at higher stall angles due to the idealized flow staying attached to the airfoil more readily. In 3D simulations, the aspect ratio of

the airfoil must also be considered; the aspect ratio is the ratio of the span length of the airfoil segment to its chord length. In general, as the aspect ratio increases, the results will approach the idealized 2D results, though they are typically rather small due to being limited by the wind tunnel dimensions in experimental tests and domain sizes in numerical tests.

While wind tunnel studies have historically been the most popular method of studying airfoils, computational fluid dynamics (CFD), a powerful numerical tool for aerodynamic simulations, has seen use in recent years for the same purpose. However, some researchers have argued that RANS-based CFD is not adequately equipped to completely capture the complex behaviour of an airfoil in stall. When Eleni et al. (2012) compared the standard closure models used in RANS simulations – Spalart-Allmaras, $k-\epsilon$, and $k-\omega$ – for accurately simulating the NACA 0012 airfoil, they concluded all three models failed to capture the stall behaviour well, though the $k-\omega$ model fared the best of the three. Other studies including Ren et al. (2009); Atkins (2015); Matyushenko et al. (2017) also support these general trends when studying other airfoils.

3.2.2 Collection of previous studies

Table 3.1 collects the drag and lift coefficient curves reported by previous studies of the NACA 0012 airfoil tested at high angles of attack in Reynolds number flows of 10^5 to 10^6 . The majority of the collected studies are from wind tunnel tests, though some CFD studies are included as well as numerical 2D predictions of the low-angle of attack behaviour from XFOIL. Note that a significant number of studies were excluded from this collection due to being restricted to low angles of attack, falling outside of the Re range, performing dynamic tests where the angle of attack is varied in real-time as opposed to steady-state testing, or studies where a finite turbine blade rather than a constant airfoil section was tested. Related studies of the NACA 0012 airfoil that met some or all of these exclusion criteria include McAlister et al. (1978); Shida et al. (1987); Ladson (1988); Laitone (1997); Martinat et al. (2008); Douvi et al. (2012); Petrillu et al. (2013); Nordanger et al. (2015); and Martinez-Aranda et al. (2016).

As can be seen in Table 3.1, available lift and drag coefficient curves were ultimately extracted from seven sources. These sources consist of six published papers and reports, as well as curves reported by Airfoil Tools (2021), which were generated using the well-documented, 2D-based XFOIL algorithm (Drela and Youngren, 2000). Table 3.1 lists the data sources, the type of test, the testing parameters (Reynolds number, wind speed, angles of attack), and the physical dimensions of the test including the chord length, the span length and aspect ratio between the span and chord length in 3D tests, and the domain/wind tunnel height.

The lift coefficient curves from the sources detailed in Table 3.1 are plotted in Figure 3.5, the exact values of which are listed in Table 3.2. In Figure 3.5, 3D tests are plotted using solid lines while 2D tests use dashed lines. Multiple tests from the same source use the same symbol and colour in a different shade. As can be seen, there is a significant amount of disagreement between the various tests, even when accounting for Reynolds number and aspect ratio (which is very similar for all 3D tests except Benyahia et al. (2003)). Still, certain general trends can be identified, particularly for the higher-angle tests from Critzos et al. (1955) and Sheldahl and Klimas (1981). There is an initial positive linear relationship between the angle of attack and lift coefficient, which reaches a peak at the stall angle at $8-15^\circ$. This is followed by a negative linear section that reaches a minimum post-stall point at $10-17^\circ$, then a linear positive

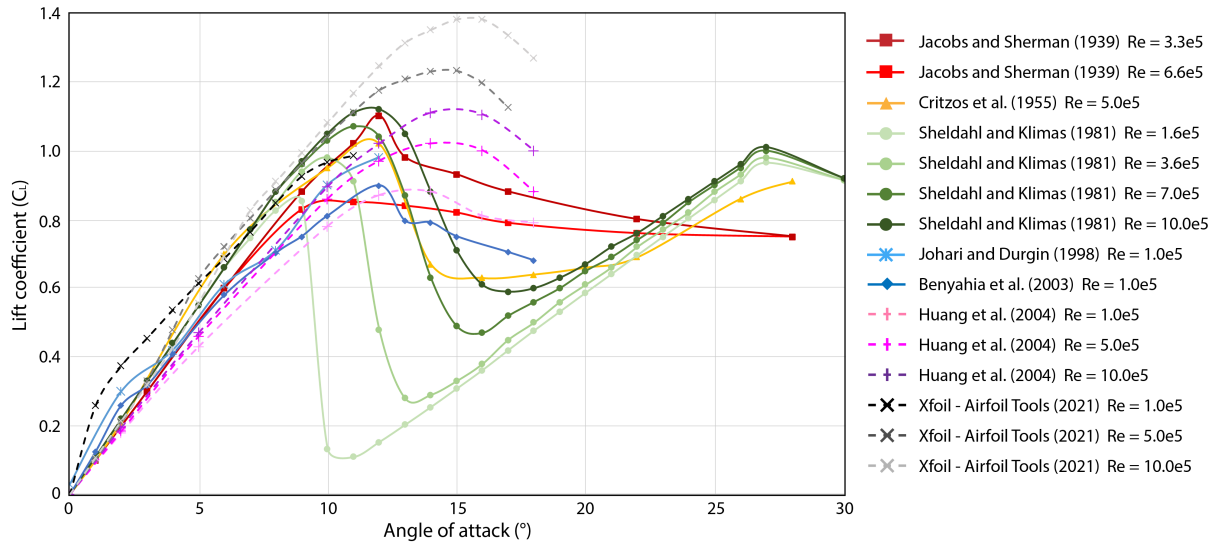


Figure 3.5: Lift coefficient curves from sources detailed in Table 3.1

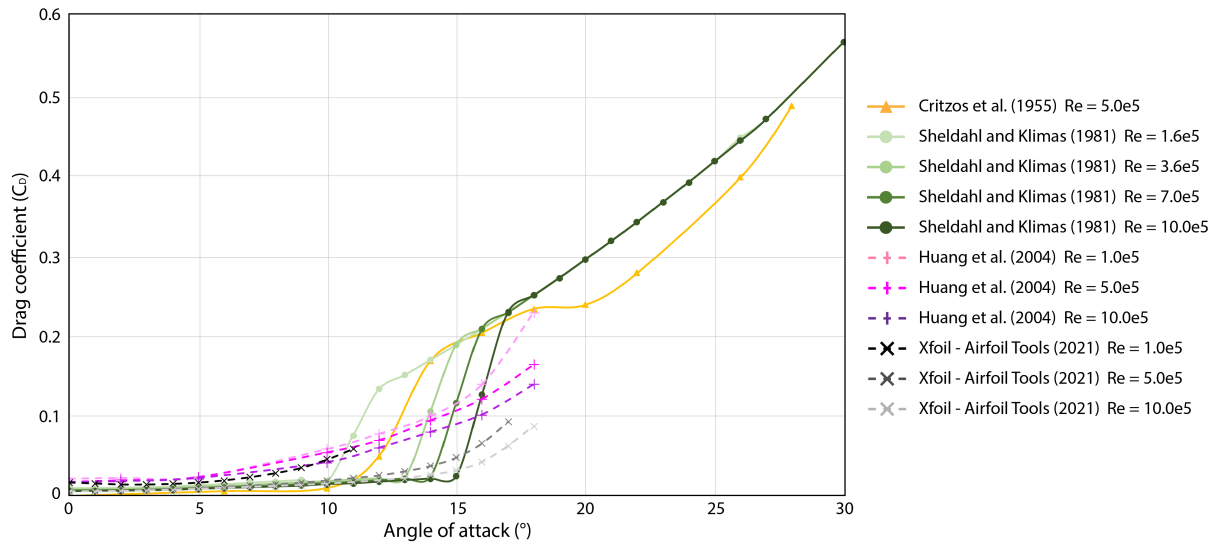


Figure 3.6: Drag coefficient curves from sources detailed in Table 3.1

deep-stall section. However, it can be seen that the post-stall behaviour recorded by Jacobs and Sherman (1939) does not match these general trends seen in more modern studies. Given that the two other studies that test beyond the stall angle (Critzos et al., 1955; Sheldahl and Klimas, 1981) agree with one another and are more recent, the post stall-behaviour reported by Jacobs and Sherman (1939) will not be considered in the following section. These trends are used to generate empirical models that synthesize average curves based on the collected responses, a process which is detailed in the following sections.

Comparing the 3D and 2D tests in Figure 3.5, it can be seen that the 2D tests predict larger maximum lift coefficients and stall angles due to ignoring the axial effects of wind loads. This phenomenon has been well-documented previously. As a result, separate 2D and 3D average curves will be generated in the following sections.

The drag coefficient curves from the sources in Table 3.5 are plotted in Figure 3.6, the exact values of which are listed in Table 3.2, though some studies did not report their drag coefficient curves. As can be seen, all curves show exponential growth in the drag coefficient as the angle of attack increases, however a large difference between curves is the presence of a significant jump in the drag coefficient at the stall angle, which is seen in the 3D tests but not in the idealized 2D environment. The results from Sheldahl and Klimas (1981) also suggests that the drag coefficient curve is relatively independent of the Reynolds number of the flow, except for it controlling the stall angle and thus when the jump occurs.

3.2.3 Average 3D lift coefficient curves

An average model of the 3D lift coefficient curves as a function of Re would be useful point of comparison for future tests, given the lack of strong agreement between existing tests and the limited amount of data available for an individual Reynolds number. To synthesize this predicted coefficient curve, the traits that define the shape of the lift coefficient, which are described above, can be quantified as function of Re. A piecewise polynomial curve based on these curve traits can then be used to generate the average predicted curve for a given Re.

The quantification of the 3D lift coefficient curve is performed by extracting the 12 curve traits shown in Figure 3.7, including the angle of attack (x), the lift coefficient (y), and the slope (s) at the initial linear (subscript 0), the stall angle (subscript 1), the minimum post-stall peak (subscript 2), and the post-stall linear (subscript 3) sections of the curve. While many of the traits vary between the different curves shown in Figure 3.5, a handful of these values are constant for all curves ($x_0 = y_0 = s_1 = s_2 = 0$), while $x_3 = 22^\circ$ is chosen as a functional, but arbitrary point to characterize the linear post-stall region. The remaining seven curve traits are then tabulated from each source in Table 3.3. Note that not all curves were tested at high enough angles of attack to report each of these traits.

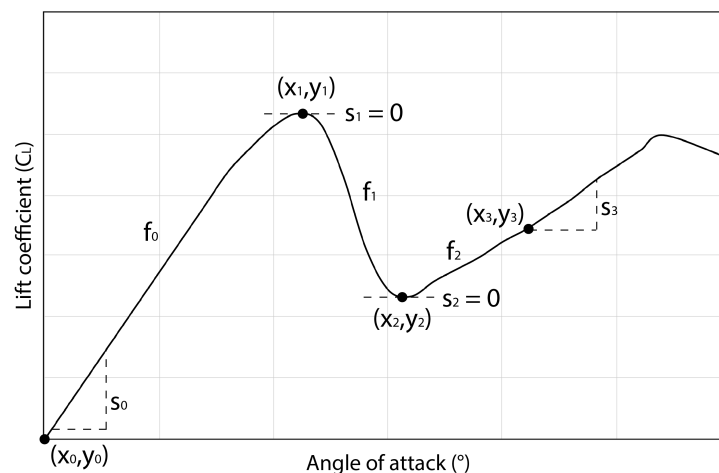


Figure 3.7: Definition of the 12 traits to characterize the 3D lift coefficient curves

Paper	Jacobs and Sherman (1939)		Critzos et al. (1955)	Sheldahl and Klimas (1981)				Johari and Durgin (1998)	Benyahia et al. (2003)	Slope S_i	Intercept I_i	Std dev
Re	$3.3*10^5$	$6.6*10^5$	$5.0*10^5$	$1.6*10^5$	$3.6*10^5$	$7.0*10^5$	$10.0*10^5$	$1.0*10^5$	$1.0*10^5$			
s_0	0.100	0.100	0.095	0.110	0.110	0.110	0.110	0.102	0.097	$6.64 * 10^{-9}$	0.101	0.006
y_1	0.86	1.10	1.04	0.86	0.98	1.08	1.13		0.90	$3.31 * 10^{-7}$	0.836	0.043
x_1	10.0	12.0	11.8	8.8	10.0	11.0	12.0		12.0	$2.06 * 10^{-6}$	9.969	0.985
y_2			0.62	0.10	0.27	0.47	0.59			$5.46 * 10^{-7}$	0.113	0.120
x_2			15.0	10.5	13.5	16.0	17.0			$7.39 * 10^{-6}$	10.378	0.773
s_3			0.025	0.054	0.055	0.050	0.048			$-5.70 * 10^{-9}$	0.050	0.011
y_3			0.690	0.696	0.720	0.740	0.760			$7.68 * 10^{-8}$	0.679	0.014

Table 3.3: Collected traits that define the 3D lift coefficient curves from Figure 3.7 and the resulting slopes and intercepts of the linear trendlines

For the purposes of these models, only the difference in Reynolds number will be considered while all other differences between the tests (chord length, aspect ratio, etc.) are considered to have a minimal impact on the resulting lift coefficient curve. Ultimately, there is an insufficient amount of data to account for any variable other than Reynolds number. Based on the data collected in Table 3.3, a linear relationship can be found for between each variable (V) and the Reynolds number of the flow, which is characterized by a slope (S) and intercept (I):

$$V_i = S_i * Re + I_i \quad (3.3)$$

Here, V_i is the i th variable ranging from s_0 to y_3 in Figure 3.7. The slope and intercept are calculated using the following equations:

$$S_i = \frac{n \sum (v_{ij} * Re_j) - \sum v_{ij} \sum Re_j}{n \sum Re_j^2 - (\sum Re_j)^2} \quad (3.4)$$

$$I_i = \frac{\sum v_{ij} - S_i \sum Re_j}{n} \quad (3.5)$$

Here, S_i and I_i are calculated from the data samples v_{ij} , where j ranges from 1 to n and n is the number of samples for that curve trait. Thus for calculating S_7 , the slope for the seventh variable y_3 , it can be seen that $n = 5$ and v_{ij} are 0.690, 0.696, 0.720, 0.740, and 0.760 as listed in the seventh row of Table 3.3.

Table 3.3 lists the calculated slopes and intercepts of the seven curve traits, which are summarized in Table 3.4. For example, this table shows that the average stall angle (x_1) for the 3D lift coefficient curve at a given Reynolds number is $x_1 = 2.06 * 10^{-6} * Re + 9.969$. The standard deviation of the experimental data compared to the linear approximation is also reported in Table 3.3, which will be used to plot uncertainty ranges for the predicted curves.

A piecewise polynomial curve can then be built from these 12 defined curve traits. Piecewise polynomial systems, also known as spline curves, are a technique from computer graphics for generating complex best-fit curves based on a limited number of inputs. These systems are built of a number of polynomial curves defined as:

$$y_i = a_i + b_i x + c_i x^2 + \dots \quad (3.6)$$

Here a_i , b_i , etc., are the unknown coefficients of the curve, and i refers to the individual piecewise polynomial curve within the system. These individual polynomials are related and solved

Variable	Curve trait	Value	Standard deviation
x_0	AoA = 0°	0	± 0
y_0	C_L at AoA = 0°	0	± 0
s_0	Slope at AoA = 0°	$6.64 * 10^{-9} * Re + 0.101$	± 0.006
x_1	Stall angle (°)	$2.06 * 10^{-6} * Re + 9.969$	± 0.985
y_1	Stall C_L	$3.31 * 10^{-7} * Re + 0.836$	± 0.043
s_1	Slope at stall = 0	0	± 0
x_2	AoA at minimum post-stall C_L (°)	$7.39 * 10^{-6} * Re + 10.378$	± 0.773
y_2	Minimum post-stall C_L	$5.46 * 10^{-7} * Re + 0.113$	± 0.120
s_2	Slope at minimum post-stall $C_L = 0$	0	± 0
x_3	AoA = 22°	22	± 0
y_3	C_L at AoA = 22°	$7.68 * 10^{-8} * Re + 0.679$	± 0.014
s_3	Slope at AoA = 22°	$-5.70 * 10^{-9} * Re + 0.050$	± 0.011

Table 3.4: Twelve curve traits defined in Figure 3.7 for the 3D lift coefficient curves

using boundary conditions to generate a continuous complex curve using only a limited number of inputs. Readers are directed to House (2014) for further explanation of this technique.

For the predicted 3D lift coefficient curve, the piecewise polynomial system is made of three curves (f_0 , f_1 , f_2) which connect x_0 to x_1 , x_1 to x_2 , and x_2 to x_3 , respectively, as shown in Figure 3.7. f_0 , f_2 , and their derivatives are defined as:

$$y = f_i(x) \quad (3.7)$$

$$f_i = a_i + b_i x + c_i x^2 + d_i x^3 \quad (3.8)$$

$$f_i' = b_i + 2c_i x + 3d_i x^2 \quad (3.9)$$

$$f_i'' = 2c_i + 6d_i x \quad (3.10)$$

Here, x and y refer to the angle of attack and lift coefficient respectively, $a_i - d_i$ are the coefficients of the polynomial curve, i is 0 or 2, and f_i' and f_i'' are the slope and curvature of the polynomial. As will be justified below, two additional terms are added for f_1 such that:

$$f_1 = a_1 + b_1 x + c_1 x^2 + d_1 x^3 + e_1 x^4 + g_1 x^5 \quad (3.11)$$

$$f_1' = b_1 + 2c_1 x + 3d_1 x^2 + 4e_1 x^3 + 5g_1 x^4 \quad (3.12)$$

$$f_1'' = 2c_1 + 6d_1 x + 12e_1 x^2 + 20g_1 x^3 \quad (3.13)$$

To solve for the unknown coefficients, a number of boundary conditions are collected, as summarized in Table 3.5. Since 14 boundary conditions were collected, 14 unknown polynomial coefficients are required for the system to remain singular. As a result, f_1 is chosen to have six coefficients (see Equation 3.11) while f_0 and f_2 were assigned four each (see Equation 3.8). This distribution was found to produce the best splines for this case. The equations in Table 3.5 are reorganized into matrix form to solve for the unknown polynomial coefficients:

$$\{y\} = [M] \{c\} \quad (3.14)$$

$$[M]^{-1} \{y\} = \{c\} \quad (3.15)$$

Boundary condition	Equation
1. f_0 passes through (x_0, y_0)	$y_0 = a_0 + b_0x_0 + c_0x_0^2 + d_0x_0^3$
2. Slope of f_0 ($= f_0'$) is defined at x_0	$s_0 = b_1 + 2c_1x_0 + 3d_1x_0^2$
3. f_0 passes through (x_1, y_1)	$y_1 = a_0 + b_0x_1 + c_0x_1^2 + d_0x_1^3$
4. f_0' is defined at x_1	$s_1 = b_1 + 2c_1x_1 + 3d_1x_1^2$
5. Curvature matches between f_0 and f_1 ($f_0'' = f_1''$) at x_1	$2c_0 + 6d_0x_1 = 2c_1 + 6d_1x_1 + 12e_1x_1^2 + 20g_1x_1^3$
6. f_1 passes through (x_1, y_1)	$y_1 = a_1 + b_1x_1 + c_1x_1^2 + d_1x_1^3 + e_1x_1^4 + g_1x_1^5$
7. f_1' is defined at x_1	$s_1 = b_1 + 2c_1x_1 + 3d_1x_1^2 + 4e_1x_1^3 + 5g_1x_1^4$
8. f_1 passes through (x_2, y_2)	$y_2 = a_1 + b_1x_2 + c_1x_2^2 + d_1x_2^3 + e_1x_2^4 + g_1x_2^5$
9. f_1' is defined at x_2	$s_2 = b_1 + 2c_1x_2 + 3d_1x_2^2 + 4e_1x_2^3 + 5g_1x_2^4$
10. $f_1'' = f_2''$ at x_2	$2c_1 + 6d_1x_2 + 12e_1x_2^2 + 20g_1x_2^3 = 2c_2 + 6d_2x_2$
11. f_2 passes through (x_2, y_2)	$y_2 = a_2 + b_2x_2 + c_2x_2^2 + d_2x_2^3$
12. f_2' is defined at x_2	$s_2 = b_2 + 2c_2x_2 + 3d_2x_2^2$
13. f_2 passes through (x_3, y_3)	$y_3 = a_2 + b_2x_3 + c_2x_3^2 + d_2x_3^3$
14. f_2' is defined at x_3	$s_3 = b_2 + 2c_2x_3 + 3d_2x_3^2$

Table 3.5: Boundary conditions of the 3D lift coefficient curves based on Figure 3.7

$$\text{inv} \begin{bmatrix} 1 & x_0 & x_0^2 & x_0^3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2x_0 & 3x_0^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & x_1 & x_1^2 & x_1^3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2x_1 & 3x_1^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 6x_1 & 0 & 0 & -2 & -6x_1 & -12x_1^2 & -20x_1^3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & x_1 & x_1^2 & x_1^3 & x_1^4 & x_1^5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2x_1 & 3x_1^2 & 4x_1^3 & 5x_1^4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & x_2 & x_2^2 & x_2^3 & x_2^4 & x_2^5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2x_2 & 3x_2^2 & 4x_2^3 & 5x_2^4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 6x_2 & 12x_2^2 & 20x_2^3 & 0 & 0 & -2 & -6x_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & x_2 & x_2^2 & x_2^3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2x_2 & 3x_2^2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & x_3 & x_3^2 & x_3^3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2x_3 & 3x_3^2 \end{bmatrix} \begin{bmatrix} y_0 \\ s_0 \\ y_1 \\ s_1 \\ 0 \\ y_1 \\ s_1 \\ y_2 \\ s_2 \\ 0 \\ y_2 \\ s_2 \\ y_3 \\ s_3 \end{bmatrix} = \begin{bmatrix} a_0 \\ b_0 \\ c_0 \\ d_0 \\ a_1 \\ b_1 \\ c_1 \\ d_1 \\ e_1 \\ g_1 \\ a_2 \\ b_2 \\ c_2 \\ d_2 \end{bmatrix} \quad (3.16)$$

Here, $\{c\}$ is a vertical array of the 14 unknown polynomial coefficients, $\{y\}$ is a vertical array of the left side of the equations from Table 3.5, and $[M]$ is the matrix of the right side of the equations from Table 3.5. Readers are encouraged to note the relationship between Equation 3.16 and Table 3.5. This matrix equation can be solved to determine the 14 polynomial coefficients for a given Reynolds number. For example, when $\text{Re} = 10^5$ the curve coefficients are found as:

$$\begin{aligned} a_0 &= 0 & a_1 &= 7.80 \cdot 10^5 & a_2 &= 1.76 \\ b_0 &= 0.102 & b_1 &= -3.67 \cdot 10^5 & b_2 &= -0.335 \\ c_0 &= 0.005 & c_1 &= 6.91 \cdot 10^4 & c_2 &= 0.021 \\ d_0 &= -6.67 \cdot 10^{-4} & d_1 &= -6.50 \cdot 10^3 & d_2 &= -3.87 \cdot 10^{-4} \\ e_1 &= 305 & & & & \\ g_1 &= -5.73 & & & & \end{aligned} \quad (3.17)$$

Using these polynomial coefficients the average 3D lift coefficient curve can be plotted using the following equation, where x is the angle of attack in degrees:

$$C_L = \begin{cases} a_0 + b_0x + c_0x^2 + d_0x^3 & x_0 \leq x \leq x_1 \\ a_1 + b_1x + c_1x^2 + d_1x^3 + e_1x^4 + g_1x^5 & x_1 < x \leq x_2 \\ a_2 + b_2x + c_2x^2 + d_2x^3 & x_2 < x \leq x_3 \\ y_3 + s_3(x - x_3) & x_3 < x \leq 25^\circ \end{cases} \quad (3.18)$$

To account for the error in the linear approximations performed in Table 3.3, additional curves are likewise generated using the largest and smallest possible curve traits within one standard deviation. The standard deviation between the true values and the linear approximations of the curve traits were calculated using the mean root square error equation and are reported in Table 3.4. The standard deviation curves were then generated using $x_1 \pm \sigma_{x_1}$, $y_1 \pm \sigma_{y_1}$, etc. where σ_{x_1} is the reported standard deviation of the curve variable x_1 . Generally, the upper curve used a positive value for the standard deviation while the lower curve used the negative value, though this was reversed in a few instances to prevent curve crossings. The additional curves represent the error in the predicted curve within a single standard deviation. Modelling the error using this method shows a very narrow range of values in the pre-stall and late-post-stall regions, while the regions at the stall angle and immediately after have a much wider range of values – this agrees with the general understanding that the stalling behaviour of an airfoil is the most complex aerodynamically and thus would have the greatest variability.

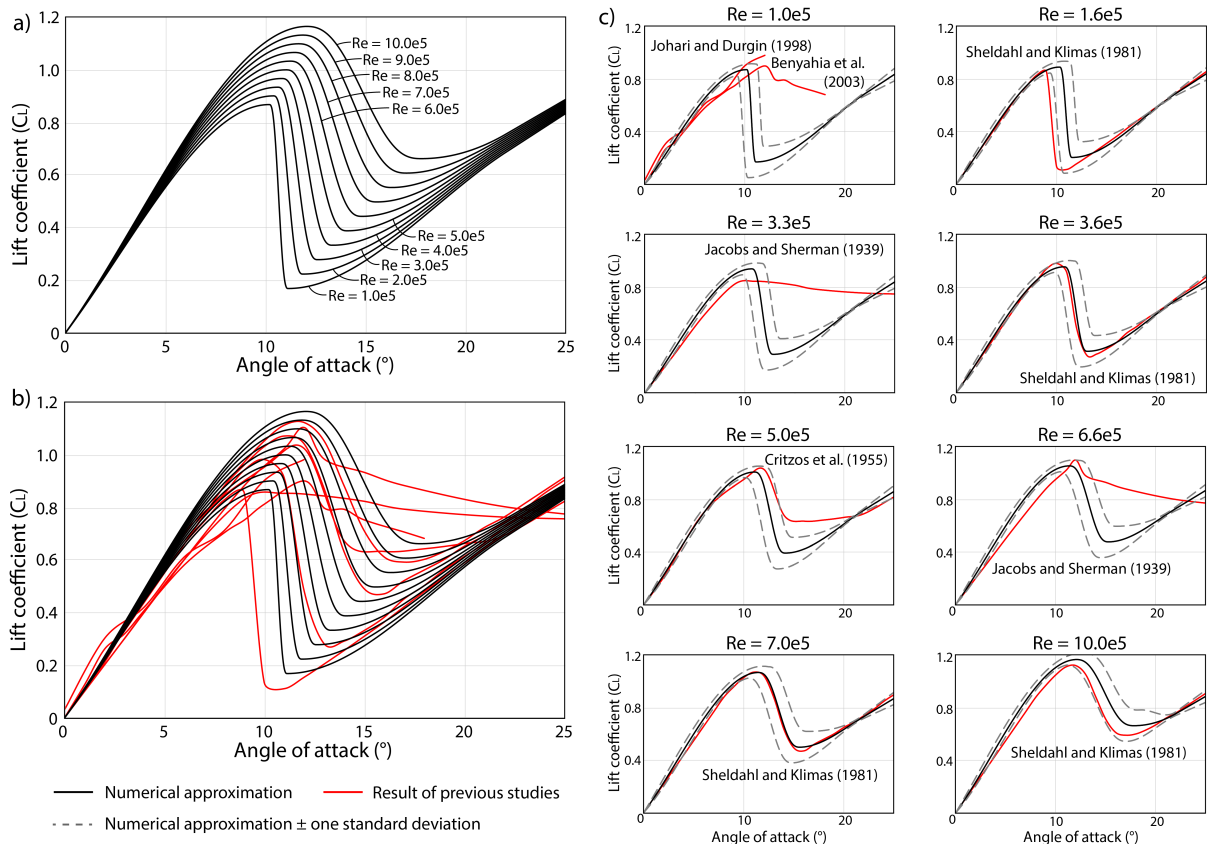


Figure 3.8: Predicted 3D lift coefficient curves at a range of Reynolds numbers, compared to the collected literature described in Table 3.1 at the same Reynolds number

Figure 3.8 plots the predicted 3D lift coefficient curves generated using the piecewise polynomial system technique. It reports lift coefficient curves at Reynolds numbers ranging from 10^5 to 10^6 and overlays them on the data from the previous studies shown in red. Then, for each Reynolds number at which lift coefficient curve data exists, a predicted curve at the same Reynolds number is generated and compared, including the one-standard-deviation curves (the dashed lines). Comparing the existing data with the predicted curves, it can be seen that the matching is quite good for curve aspects that were able to be considered. For example, no stall angle was quantified from Johari and Durgin (1998), and no post-stall behaviour was quantified from Jacobs and Sherman (1939), explaining the differences between that data and the predicted curves. Despite being included in the linear approximations, it can be seen that the post-stall decrease in lift coefficient is over-estimated compared to Critzos et al. (1955) at $Re = 5.0 \cdot 10^5$, and the stall angle is over-estimated compared to Sheldahl and Klimas (1981) at $Re = 1.6 \cdot 10^5$ and under-estimated compared to Benyahia et al. (2003) at $Re = 1.0 \cdot 10^5$. Beyond these exceptions, matching is quite good, including a nearly perfect estimation compared to the curve reported by Sheldahl and Klimas (1981) at $Re = 7.0 \cdot 10^5$. One could argue that these predicted curves over-value the data reported by Sheldahl and Klimas (1981), considering that these curves make up four of the five sources of post-stall information, but given the small amount of data to work with, this approach was deemed best.

3.2.4 Average 2D lift coefficient curves

The process described in Section 3.2.3 to generate a predicted lift coefficient is likewise performed using 2D data. For brevity, only changes from the procedure outlined in the previous section are reported here.

Figure 3.9 shows the nine curve traits used to define the two polynomial system. Here, $x_0 = y_0 = s_1 = 0$ and $x_2 = 17^\circ$, while the rest are defined on a per-curve basis. The piecewise polynomial system is made of two fourth-order equations where x is the angle of attack, and a_0 to d_1 are the eight unknown curve coefficients:

$$C_L = \begin{cases} a_0 + b_0x + c_0x^2 + d_0x^3 & x_0 \leq x \leq x_1 \\ a_1 + b_1x + c_1x^2 + d_1x^3 & x_1 < x \leq x_2 \end{cases} \quad (3.19)$$

Table 3.6 lists the collected traits from the recorded 2D lift coefficient curves from Huang et al. (2004) and Airfoil Tools (2021), as well as the resulting slopes, intercepts and standard deviations. Table 3.7 lists the eight boundary conditions used to solve for the curve coefficients. Finally, Figure 3.10 presents the predicted curves and compares them to the results from the existing studies. It can be seen that while the general trends are captured, due to the disparity between the data samples and the low number of curves to work with, the true data often lies just outside of one standard deviation from the predicted curve, a slightly worse accuracy compared to the averaged 3D lift coefficient curves. Still, the predicted curves accurately pass between the two data sets at each Reynolds number making this an effective enough model of data with less-than-perfect agreement.

3.2.5 Average 3D drag coefficient curves

The process described in Section 3.2.3 to generate average 3D lift coefficient curves is likewise performed here for 3D drag coefficient curves. For brevity, only changes from the procedure outlined in the previous section are presented here.

Figure 3.11 lists the nine curve traits used to define the three polynomial system. Here, $x_0 = 0^\circ$ and $x_3 = 25^\circ$, while the rest are defined on a per-curve basis. The piecewise polynomial system is made of a first-order, a fifth-order, and a second-order equation where x is the angle of attack, and a_0 to c_2 are the ten unknown curve coefficients:

$$C_D = \begin{cases} a_0 + b_0x & x_0 \leq x \leq x_1 \\ a_1 + b_1x + c_1x^2 + d_1x^3 + e_1x^4 & x_1 < x \leq x_2 \\ a_2 + b_2x + c_2x^2 & x_2 < x \leq x_3 \end{cases} \quad (3.20)$$

Table 3.8 lists the collected traits from the recorded 3D drag coefficient curves from Critzos et al. (1955) and Sheldahl and Klimas (1981), and the resulting slopes, intercepts and standard deviations. Table 3.9 lists the ten boundary conditions used to solve for the curve coefficients. Finally, Figure 3.12 summarizes the results of the averaging and compares them to the results from the existing studies. It can be seen that, with the exception of the dip in the drag coefficient reported by Critzos et al. (1955) around an angle of attack of 20° , the average curve includes nearly all data within one standard deviation, and matched extremely well to the data from Sheldahl and Klimas (1981) at Reynolds numbers of $1.6 \cdot 10^5$ and 10^6 .

3.2.6 Average 2D drag coefficient curves

The process described in Section 3.2.3 to generate average 3D lift coefficient curves is likewise performed here for 2D drag coefficient curves. For brevity, only changes from the procedure outlined in the previous section are presented here.

Figure 3.13 lists the eight curve traits used to define the two polynomial system. Here, $x_0 = 0^\circ$ and $x_2 = 17^\circ$, while the rest are defined on a per-curve basis. The piecewise polynomial system is made of a third-order and a second-order equation where x is the angle of attack, and a_0 to c_1 are the seven unknown curve coefficients:

$$C_D = \begin{cases} a_0 + b_0x + c_0x^2 + d_0x^3 & x_0 \leq x \leq x_1 \\ a_1 + b_1x + c_1x^2 & x_1 < x \leq x_2 \end{cases} \quad (3.21)$$

Table 3.10 lists the collected traits from the recorded 2D drag coefficient curves from Huang et al. (2004) and Airfoil Tools (2021), and the resulting slopes, intercepts and standard deviations. Table 3.11 lists the seven boundary conditions used to solve for the curve coefficients. Finally, Figure 3.14 summarizes the results of the averaging and compares them to the results from the existing studies. It can be seen that the overall matching is well given the disparity between recorded curves, though it struggles the most at low and high angles of attack.

Paper	Huang et al. (2004)			XFOIL data (Airfoil Tools, 2021)			Slope S_i	Intercept I_i	Std dev
	Re	$1.0 \cdot 10^5$	$5.0 \cdot 10^5$	$10.0 \cdot 10^5$	$1.0 \cdot 10^5$	$5.0 \cdot 10^5$			
s_0	0.086	0.092	0.094	0.187	0.104	0.107	$-3.80 \cdot 10^{-8}$	0.132	0.031
x_1	13.0	14.5	15.0	11.5	14.5	15.5	$3.26 \cdot 10^{-6}$	12.26	0.63
y_1	0.89	1.03	1.12	0.99	1.24	1.39	$3.45 \cdot 10^{-7}$	0.926	0.106
y_2	0.80	0.95	1.06	-	1.13	1.34	$4.19 \cdot 10^{-7}$	0.796	0.109
s_2	-0.023	-0.060	-0.053	-	-0.070	-0.067	$-3.06 \cdot 10^{-8}$	-0.035	0.013

Table 3.6: Collected traits that define the 2D lift coefficient curves from Figure 3.9 and the resulting slopes, intercepts, and standard deviation of the linear trendlines

Boundary condition	Equation
1. f_0 passes through (x_0, y_0)	$y_0 = a_0 + b_0x_0 + c_0x_0^2 + d_0x_0^3$
2. f_0' is defined at x_0	$s_0 = b_1 + 2c_1x_0 + 3d_1x_0^2$
3. f_0 passes through (x_1, y_1)	$y_1 = a_0 + b_0x_1 + c_0x_1^2 + d_0x_1^3$
4. f_0' is defined at x_1	$s_1 = b_0 + 2c_0x_1 + 3d_0x_1^2$
5. $f_0'' = f_1''$ at x_1	$2c_0 + 6d_0x_1 = 2c_1 + 6d_1x_1$
6. f_1 passes through (x_1, y_1)	$y_1 = a_1 + b_1x_1 + c_1x_1^2 + d_1x_1^3$
7. f_1' is defined at x_1	$s_1 = b_1 + 2c_1x_1 + 3d_1x_1^2$
8. f_1' is defined at x_2	$s_2 = b_1 + 2c_1x_2 + 3d_1x_2^2$

Table 3.7: Boundary conditions of the 2D lift coefficient curves based on Figure 3.9 which are used to solve for the curve coefficients for a given Reynolds number

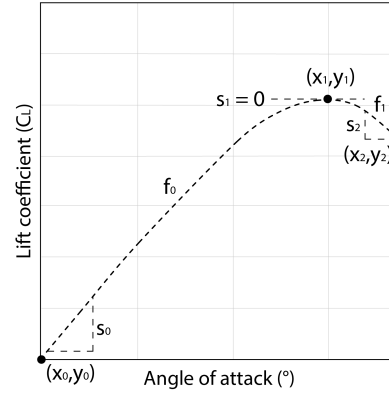


Figure 3.9: Definition of the 9 traits to characterize the 2D lift coefficient curves

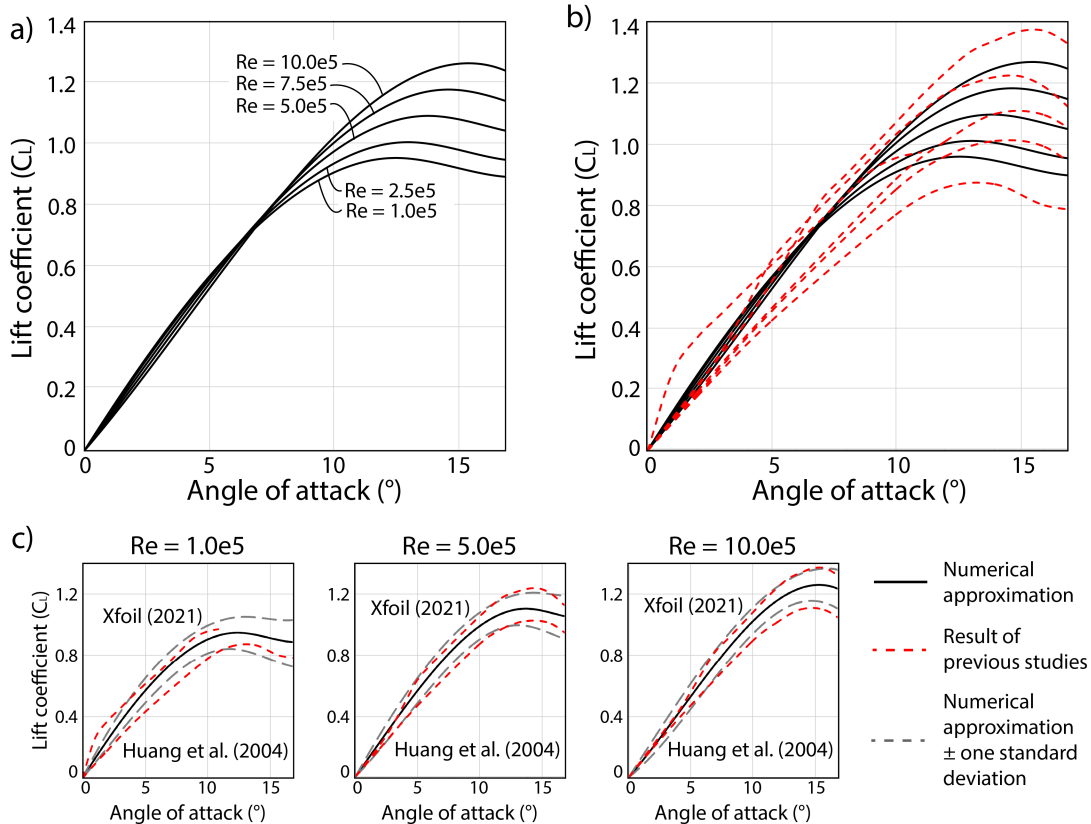


Figure 3.10: Predicted 2D lift coefficient curves at a range of Reynolds numbers, compared to the collected literature describe in Table 3.1 at the same Reynolds number

Paper	Critzos et al. (1955)	Sheldahl and Klimas (1981)				Slope S_i	Intercept I_i	Std dev
	Re	$1.6 \cdot 10^5$	$3.6 \cdot 10^5$	$7.0 \cdot 10^5$	$10.0 \cdot 10^5$			
y_0	0.000	0.010	0.008	0.007	0.006	$-3.92 \cdot 10^{-9}$	0.008	0.003
x_1	10.0	10.0	13.0	14.0	15.0	$5.66 \cdot 10^{-6}$	9.32	1.26
y_1	0.010	0.018	0.022	0.024	0.025	$8.77 \cdot 10^{-9}$	0.015	0.005
x_2	14.0	12.0	15.0	16.0	17.0	$5.45 \cdot 10^{-6}$	11.84	0.70
y_2	0.170	0.134	0.190	0.210	0.231	$1.06 \cdot 10^{-7}$	0.129	0.013
y_3	0.370	0.420	0.420	0.420	0.420	$5.29 \cdot 10^{-9}$	0.407	0.020
s_3	0.030	0.026	0.026	0.026	0.026	$-4.24 \cdot 10^{-10}$	0.027	0.002

Table 3.8: Collected traits that define the 3D drag coefficient curves from Figure 3.11 and the resulting slopes, intercepts, and standard deviation of the linear trendlines

Boundary condition	Equation
1. f_0 passes through (x_0, y_0)	$y_0 = a_0 + b_0 x_0$
2. f_0 passes through (x_1, y_1)	$y_1 = a_0 + b_0 x_1$
3. $f_0' = f_1'$ at x_1	$b_0 = b_1 + 2c_1 x_1 + 3d_1 x_1^2 + 4e_1 x_1^3$
4. $f_0'' = f_1''$ at x_1	$0 = 2c_1 + 6d_1 x_1 + 12e_1 x_1^2$
5. f_1 passes through (x_1, y_1)	$y_1 = a_1 + b_1 x_1 + c_1 x_1^2 + d_1 x_1^3 + e_1 x_1^4$
6. f_1 passes through (x_2, y_2)	$y_2 = a_1 + b_1 x_2 + c_1 x_2^2 + d_1 x_2^3 + e_1 x_2^4$
7. $f_1' = f_2'$ at x_2	$b_1 + 2c_1 x_2 + 3d_1 x_2^2 + 4e_1 x_2^3 = b_2 + 2c_2 x_2$
8. $f_1'' = f_2''$ at x_2	$2c_1 + 6d_1 x_2 + 12e_1 x_2^2 = 2c_2$
9. f_2 passes through (x_3, y_3)	$y_3 = a_2 + b_2 x_3 + c_2 x_3^2$
10. f_2' is defined at x_3	$s_3 = b_2 + 2c_2 x_3$

Table 3.9: Boundary conditions of the 3D drag coefficient curves based on Figure 3.11 which are used to solve for the curve coefficients for a given Reynolds number

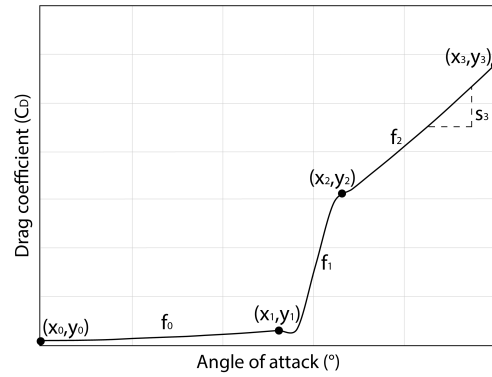


Figure 3.11: Definition of the 9 traits to characterize the 3D drag coefficient curves

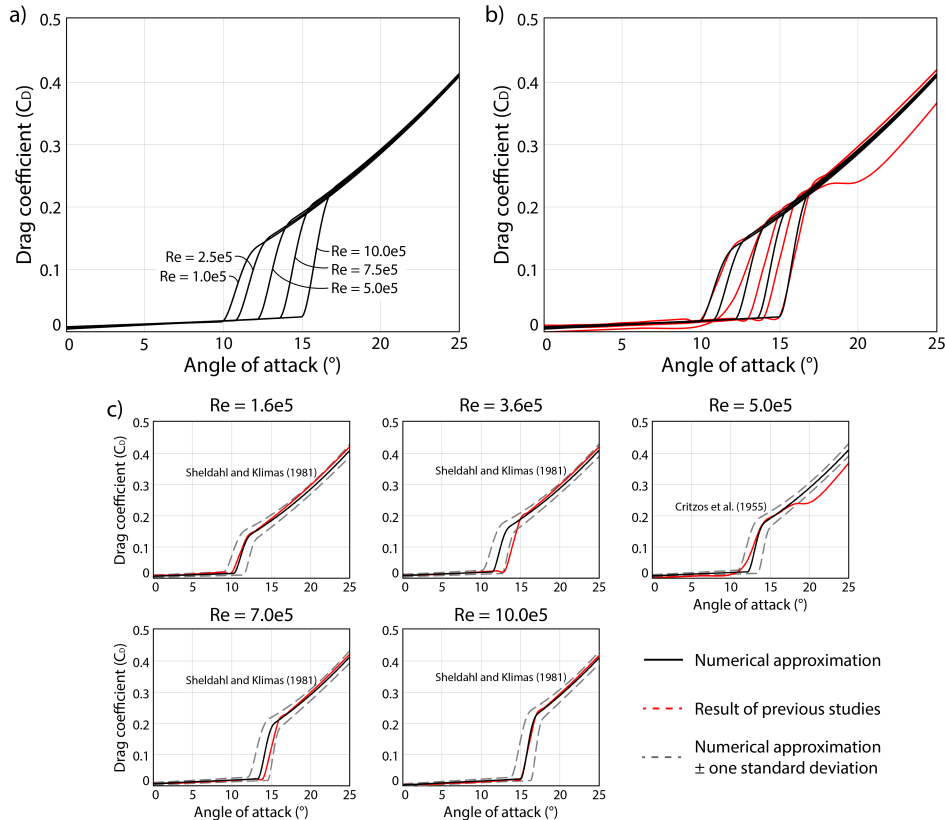


Figure 3.12: Predicted 3D drag coefficient curves at a range of Reynolds numbers, compared to the collected literature describe in Table 3.1 at the same Reynolds number

Paper	Huang et al. (2004)			XFOIL data (Airfoil Tools, 2021)			Slope S_i	Intercept I_i	Std dev
	Re	$1.0 \cdot 10^5$	$5.0 \cdot 10^5$	$10.0 \cdot 10^5$	$1.0 \cdot 10^5$	$5.0 \cdot 10^5$			
y_0	0.022	0.018	0.017	0.017	0.006	0.005	$-8.88 \cdot 10^{-9}$	0.019	0.0053
s_0	0.0005	0.0010	0.0005	-0.0012	0.0002	0.0001	$6.57 \cdot 10^{-10}$	-0.0002	0.0006
x_1	14.0	14.0	14.0	-	14.0	15.0	$6.46 \cdot 10^{-7}$	13.80	0.33
y_1	0.100	0.095	0.080	-	0.038	0.026	$-4.70 \cdot 10^{-8}$	0.097	0.026
y_2	0.167	0.129	0.109	-	0.093	0.062	$-8.19 \cdot 10^{-8}$	0.163	0.021
s_2	0.045	0.022	0.019	-	0.027	0.023	$-2.27 \cdot 10^{-8}$	0.041	0.005

Table 3.10: Collected traits that define the 2D drag coefficient curves from Figure 3.13 and the resulting slopes, intercepts, and standard deviation of the linear trendline

Boundary condition	Equation
1. f_0 passes through (x_0, y_0)	$y_0 = a_0 + b_0 x_0 + c_0 x_0^2 + d_0 x_0^3$
2. f_0' defined at x_0	$s_0 = b_0 + 2c_0 x_0 + 3d_0 x_0^2$
3. f_0 passes through (x_1, y_1)	$y_1 = a_0 + b_0 x_1 + c_0 x_1^2 + d_0 x_1^3$
4. $f_0' = f_1'$ at x_1	$b_0 + 2c_0 x_1 + 3d_0 x_1^2 = b_1 + 2c_1 x_1$
5. $f_0'' = f_1''$ at x_1	$2c_0 + 6d_0 x_1 = 2c_1$
6. f_1 passes through (x_1, y_1)	$y_1 = a_1 + b_1 x_1 + c_1 x_1^2$
7. f_1 passes through (x_2, y_2)	$y_2 = a_1 + b_1 x_2 + c_1 x_2^2$

Table 3.11: Boundary conditions of the 2D drag coefficient curves based on Figure 3.11 which are used to solve for the curve coefficients for a given Reynolds number

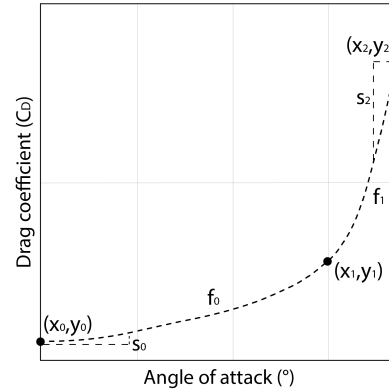


Figure 3.13: Definition of the 8 traits to characterize the 2D drag coefficient curves

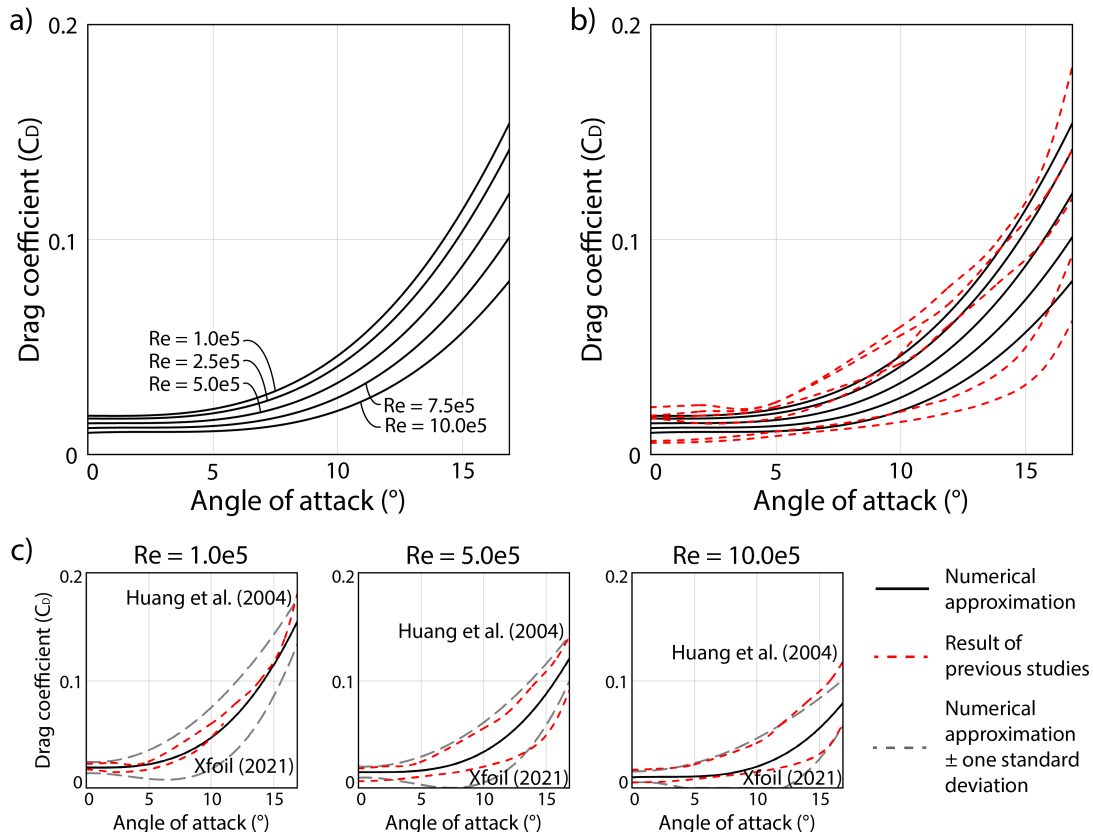


Figure 3.14: Predicted 2D drag coefficient curves at a range of Reynolds numbers, compared to the collected literature describe in Table 3.1 at the same Reynolds number

3.3 CFD simulations

3.3.1 Overview

CFD is a modern aerodynamic simulation technique that can model flow through a domain and determine aerodynamic loads on surfaces from pressure and drag. CFD divides the fluid domain into a number of small elements and solves the local flow behaviour of each element to collectively characterize the full, complex behaviour of the flow. Inexperienced users can easily perform CFD simulations incorrectly, however a well-constructed CFD simulation can yield extremely precise results, comparable to the accuracy offered by wind tunnel experiments. As Section 2.4.3 has already offered an in-depth discussion of CFD, only a broad summary is presented here.

Two steps in CFD simulations are notably computationally intensive: the first is the creation of the domain mesh which is often performed using an auto-meshing algorithm. The second is running the CFD simulation itself; the resolution of the flow through the meshed domain is performed by iteratively solving the flow and pressure behaviour within each element of the fluid domain. The specific equations solved during this process depend on the turbulence model used in the analysis, including options such as Reynolds-averaged Navier-Stokes (RANS), unsteady RANS (URANS), and large eddy simulation (LES). The derivation of each model has been well-documented previously, but to summarize each model:

- RANS simulations are a widely-used series of time-averaged equations of motions for fluid flow, meaning they are best-suited to steady-state problems. The RANS equations introduce Reynolds stresses into the Navier-Stokes equations of motion of the fluid, which in turn must be solved using additional closure models such as S-A, k - ϵ , and k - ω . The main advantage of RANS is its higher speed due to fewer equations and lower required mesh density compared to the alternatives.
- URANS simulations employ the RANS equations to generate a time history of steady-state solutions, allowing it to approximate unsteady behaviour. Provided that the time scale of the turbulence is much smaller than the time scale of the flow, this technique is relatively accurate and maintains the higher speeds offered by RANS simulations.
- LES modifies the Navier-Stokes equation by filtering out very small-scale turbulence, allowing true time-history simulations to be performed in a feasible amount of time. LES is more computationally intensive than RANS, particularly because it requires a more refined grid in the fluid domain to capture smaller-scale turbulence effects, but is capable of achieving time-history results with a greater level of accuracy. LES is not applicable for 2D simulations.

Note that the term “turbulence model” can be used to describe the type of simulation performed (RANS, LES, etc.) as well as the set of closure equations used in RANS simulations (S-A, k - ω , etc.) depending on context. As both will be compared in this study, the term “closure model” is adopted to refer to the latter sets of equations. Regardless of which turbulence model is employed, additional requirements must be met for accurate simulations such as mesh sensitivity analyses as well as y^+ and Courant number criteria, which will be detailed below. In

this study, the commercial software STAR-CCM+ was used to perform the simulations. Due to the high computational requirements of CFD and the number of simulations performed, this study would be infeasible to run on a personal computer, and as such the CFD simulations presented here were performed on supercomputing clusters provided by Compute Canada. In general, these simulations were meshed and run on 32-CPU cores, taking between one to seven days depending on mesh density, turbulence model, number of simulation steps and number of testing parameters.

3.3.2 Mesh specifications and sensitivity analysis

The dimensions of the experimental tests by Sheldahl and Klimas (1981) were replicated in this study, thus the CFD simulations were performed on an airfoil with a chord length of 0.15 m and, in 3D tests, a 0.91 m across-wind span. All simulations were performed in smooth flows at a Reynolds number of 5×10^5 , the midpoint of the testing range collected in Table 3.1. To achieve this value, assuming an air density of 1.225 kg/m^3 and a kinematic viscosity of $1.48 \times 10^{-5} \text{ m}^2/\text{s}$, a mean wind speed of 49.3 m/s was used during testing.

To simulate equivalent test conditions compared to the existing studies, a uniform cross-section of the NACA 0012 airfoil was placed in a semi-circular-upwind and rectangular-downwind domain. By rotating the airfoil around its windward edge while keeping wind flow directly along the center axis, the airfoil could be tested at a range of angles of attack. Figure 3.15 shows the specifications of the mesh used in the 2D and 3D simulations. A curved inlet boundary, the typical form used in airfoil simulations, supplies the inflow which travels across the airfoil and is released at the flat face on the far side of the domain. The dimensions of the domain are shown relative to the chord length c , these values are similar in size to previous CFD simulations of airfoils. The domain mesh is defined by a base mesh size, which is refined in a cross-shaped domain in the along-wind and vertical directions around the airfoil, and is further refined at the surface of the airfoil. The airfoil surface includes 30 prism layers with a growth rate of approximately 1.2, highlighted in Figure 3.15d. To accurately resolve the sub-scale turbulence, it is critical for the dimensionless wall distance (y^+) to be ≤ 1 ; y^+ is a nondimensional parameter that controls the wall friction caused by the flow and is critical for accurately modelling the drag and lift on a surface in a CFD simulation. y^+ can be estimated using the following equations:

$$y^+ = \frac{y_{nearwall} u^*}{\nu} \quad (3.22)$$

$$u^* = \sqrt{\frac{\tau_w}{\rho}} \quad (3.23)$$

$$\tau_w = 0.5 \rho U^2 C_f \quad (3.24)$$

$$C_f = (2 \log_{10}(\text{Re}) - 0.65)^{-2.3} \quad \text{for } \text{Re} < 10^9 \quad (3.25)$$

Here, $y_{nearwall}$ is the thickness of the near-wall prism layer, u^* is the friction velocity, ν is the kinematic viscosity of the air, τ_w is the wall shear stress, ρ is the air density, U is the freestream wind speed, C_f is the skin friction, and Re is the Reynolds number of the flow. Using the properties specified above, to achieve a y^+ value of 1, the first prism layer thickness needs to be approximately $\leq 6 \times 10^{-6} \text{ m}$. While Equations 3.22-3.25 are only approximations used to

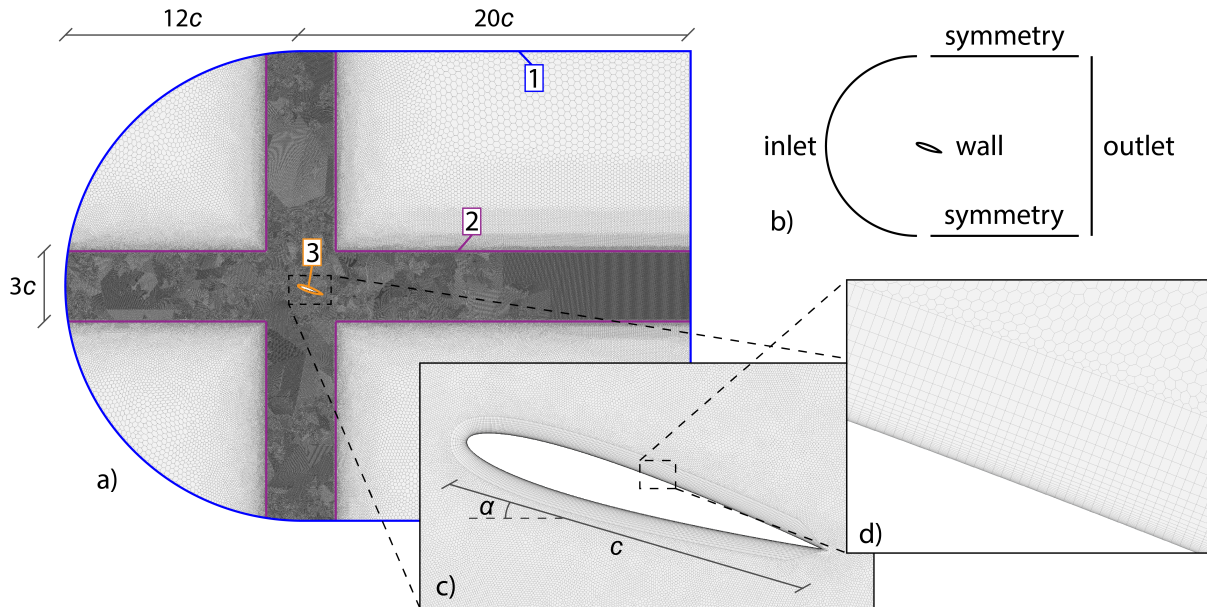


Figure 3.15: A summary of the testing mesh, used in both 2D and 3D testing. c is the chord length of 0.15 m and in the 3D tests the depth of the domain is 0.91 m. a) The shape and size of the domain as well as Boundaries 1, 2, and 3 used to refine the mesh, as explained in Table 3.12. b) A summary of the CFD domain boundaries. c) A close-up of the airfoil boundary, defining the chord length c and the angle of attack α . d) A close-up of the prism layers around the airfoil

Boundary and property	Value	Base mesh size B (m)							
		0.1000	0.0750	0.0500	0.0250	0.0100	0.0075	0.0060	0.0050
1. Outer boundary - target size (m)	$2B$	0.2000	0.1500	0.1000	0.0500	0.0200	0.0150	0.0120	0.0100
2. Refinement volume - target size (m)	$0.2B$	0.0200	0.0150	0.0100	0.0050	0.0020	0.0015	0.0012	0.0010
3. Airfoil boundary - target size (m)	$0.08B$	0.0080	0.0060	0.0040	0.0020	0.0008	0.0006	0.0005	0.0004
3. Airfoil boundary - first prism layer thickness (m)	$6 * 10^{-6}$	-	-	-	-	-	-	-	-
3. Airfoil boundary - prism layers total thickness (m)	0.007	-	-	-	-	-	-	-	-
3. Airfoil boundary - number of prism layers	30	-	-	-	-	-	-	-	-
Resulting cell count (2D):		28.8K	48.1K	90.3K	289K	1.58M	2.42M	3.92M	5.68M
Resulting cell count (3D):					7.20M				

Table 3.12: Summary of base and refined mesh sizes compared in the sensitivity analysis and subsequent simulations, with the final cell counts used in the 2D and 3D simulations highlighted. For values which are a function of the mesh’s base size, the resulting sizes are tabulated

estimate the y^+ , post-simulation analysis confirmed that $y^+ \leq 1$ in over 95% of cells on the surface of the airfoil.

Though detailed investigation of the wall treatment is beyond the scope of the paper, it will be briefly mentioned that the use of a dimensionless wall distance of $30 \leq y^+ \leq 100$ alongside a wall function was explored. Employing this so-called “high y^+ wall treatment” reduces the number of required cells by approximating the sub-scale turbulence using a log-law function.

This technique has been shown to be effective for less turbulent scenarios, but for stalled airfoils it was found that this technique damped vortex shedding, resulting in reduced drag compared to cases where vortex shedding was simulated to occur.

Based on the near-wall prism layer thickness as well as the desired number of prism layers and the scale between adjacent layers, the total thickness of the prism layers can be estimated using the following equations:

$$y_{outer} = y_{nearwall}(s^{n-1}) \quad (3.26)$$

$$y_{total} = y_{nearwall} \sum_{i=0}^{n-1} s^i \quad (3.27)$$

Here, $y_{nearwall}$ is the thickness of the near-wall prism layer, n is the number of prism layers, s is the scaling factor between prism layers, and y_{total} is the combined thickness of all the prism layers. y_{outer} is the thickness of the outermost prism layer, which should be comparable in size to the adjacent mesh to prevent large jumps in size. In this case, assuming 30 prism layers and a scaling factor of 1.2, the total thickness of the layers was estimated at about 0.007 m.

An additional parameter that must be considered in unsteady CFD simulations is the Courant number, which controls the maximum allowable time step length during a simulation. The Courant number is defined as:

$$C = \frac{u\Delta t}{\Delta x} \leq 1 \quad (3.28)$$

Here, C is the Courant number, U is the wind speed, Δt is the time step length, and Δx is the mesh size. Simply put, this parameter ensures that the flow is unable to travel farther than one adjacent cell between time steps, improving the stability and convergence of the analysis. In reality, the Courant number is the sum of three terms in the x-, y-, and z-directions of the flow, however, in this case $U_x \gg U_y, U_z$ and thus the other terms can be disregarded. It was found that this parameter can be oppressive at the finest mesh points along the airfoil, recommending time steps of below 10^{-6} s. Instead the time step for unsteady analyses was selected at 10^{-4} s using the general properties of the refined cross-section of the domain (see Figure 3.15) where $U=49.3$ m/s and $\Delta x=0.006$ m.

A mesh sensitivity analysis was also performed to confirm that the flow was modelled properly beyond the surface of the airfoil. As shown in Table 3.12, this was done by setting the mesh size of certain surfaces and regions as a function of the base mesh size of the simulation, applying a conservative growth rate of 1.05 to the automatic mesher, and varying the base size of the mesh between 0.100 m to 0.005 m. By comparing the recorded lift and drag coefficients predicted by each simulation, once the mesh became fine enough that consistent results were obtained, it was deemed that the results had become mesh-insensitive.

Eight different base sizes were tested via 2D URANS simulations using the k- ω closure model at angles of attack of 8° and 17° for the mesh sensitivity analysis as described in Table 3.12. These angles of attack were chosen such that the former would capture the pre-stall flow and the latter the post-stall flow. Table 3.13 and Figure 3.16 summarize the resulting drag and lift coefficients for the sixteen simulations and compare them to the predicted 2D lift and drag coefficient curves calculated in Sections 3.2.4 and 3.2.6. As one would expect, it can be seen that the simulations converge more readily in the simpler flow at the pre-stall angle of

AoA = 8°		Drag			Lift		
Base size (m)	Force (N)	Coefficient	% change	Force (N)	Coefficient	% change	
0.1000	20.1	0.099	–	113.1	0.56	–	
0.0750	18.4	0.091	8.9%	101.5	0.50	11.5%	
0.0500	13.6	0.067	35.2%	123.6	0.61	17.9%	
0.0250	7.2	0.035	90.8%	155.1	0.76	20.3%	
0.0100	4.0	0.020	77.5%	173.0	0.85	10.3%	
0.0075	3.8	0.019	6.7%	172.8	0.85	0.1%	
0.0060	3.6	0.018	4.0%	173.1	0.85	0.2%	
0.0050	3.5	0.017	3.7%	174.2	0.86	0.6%	
Predicted		0.026 ± 0.019			0.84 ± 0.18		

AoA = 17°		Drag			Lift		
Base size (m)	Force (N)	Coefficient	% change	Force (N)	Coefficient	% change	
0.1000	50.5	0.25	–	121.4	0.60	–	
0.0750	48.4	0.24	4.4%	116.1	0.57	4.6%	
0.0500	45.7	0.22	5.8%	113.0	0.56	2.8%	
0.0250	43.7	0.21	4.7%	107.2	0.53	5.4%	
0.0100	64.9	0.32	32.7%	178.1	0.88	39.8%	
0.0075	70.8	0.35	8.4%	193.6	0.95	8.0%	
0.0060	77.0	0.38	8.1%	215.3	1.06	10.1%	
0.0050	75.2	0.37	2.5%	211.9	1.04	1.9%	
Predicted		0.122 ± 0.021			1.05 ± 0.14		

Table 3.13: Summary of mesh sensitivity analysis results at angles of attack of 8° and 17°, including the relative percent change in the coefficient as the mesh is refined, performed using 2D URANS simulations. The predicted results come from Sections 3.2.4 and 3.2.6

attack. It can also be seen that the converged results match the predicted model quite well for both instances of lift and for the pre-stall drag.

It is shown in Figure 3.16 that the CFD-generated drag coefficient far exceeds the value predicted by the synthesized curve from Section 3.2.6. The reason for this discrepancy likely stems from the data used to generate the predicted drag coefficient curve in Section 3.2.6: the XFOIL simulations in Airfoil Tools (2021) use a simplified pressure panel method (Drela, 1989) that can struggle to model airfoil stall, and the CFD simulations by Huang et al. (2004) used a relatively coarse mesh of only 210K cells which, as shown here, may not predict accurate drag results. This latter phenomenon can be seen in the mesh sensitivity analysis; at the higher base cell sizes the flow over the airfoil remains attached and the drag is lower. As the mesh is refined, the flow remains attached and the drag coefficient value approaches the value predicted by the synthesized curve, but once the base size reaches 0.01 m – the same mesh size where the

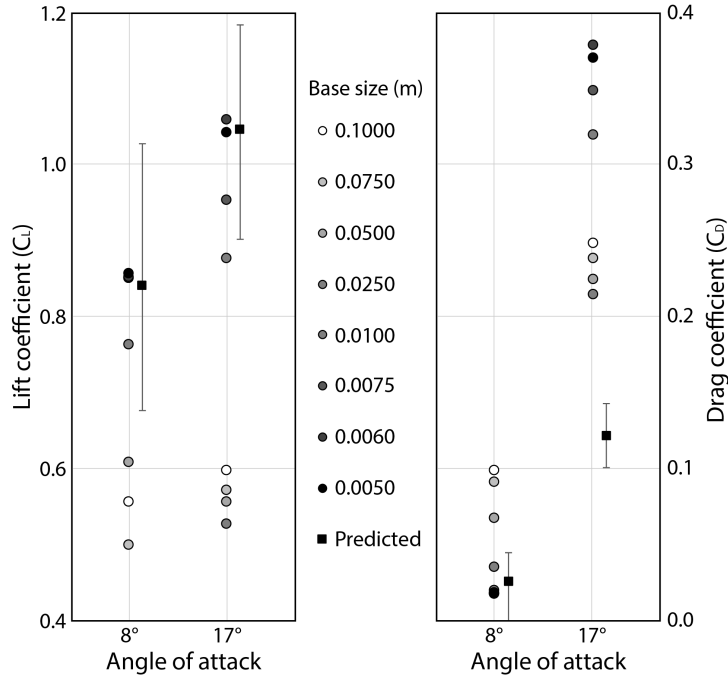


Figure 3.16: Results of mesh sensitivity analysis, with predicted results shifted slightly for visibility. Note the mismatch between synthesis- and CFD-predicted drags at the 17° angle of attack

cell count significantly exceeds the simulations by Huang et al. (2004) – the flow detaches and the drag increases significantly. This is further supported by the high wall treatment simulation discussed above, when a coarser prism layer mesh with a $y^+ \approx 30$ was tested, the flow remained attached and a drag coefficient of 0.14 was found at this angle of attack. It appears clear that despite the airfoil being beyond its stall angle, the vortex suppression that is caused by 2D simulations and coarse meshes restricted vortex shedding in the simulations by Huang et al. (2004) and thus the predicted drag coefficient curve will be incorrectly lower than calculated by the detailed CFD simulations.

Despite the discrepancy between the predicted and calculated drag coefficients, the mesh sensitivity results can be used to identify a target mesh size. By judging the percentage change between predicted coefficients shown in Table 3.13 as the mesh is refined, it appears that mesh sensitivity is mostly eliminated at a base mesh size of 0.006 m across all four criteria. Thus all subsequent 2D simulations will use a base mesh size of 0.006 m, as specified in Table 3.12.

3.3.3 2D simulation results

Using the validated mesh, a series of 2D simulations were performed of the airfoil at sixteen angles of attack ranging from 0° to 26°, with a focus on the angles near stall condition. While LES can not be performed in a 2D environment, four other types of CFD simulations were performed for a total of 64 simulations, out of which the resulting lift and drag coefficient curves were extracted. These conditions consisted of a RANS simulation using the $k-\omega$ closure model and three URANS simulations using the Spalart-Allmaras, $k-\epsilon$, and $k-\omega$ closure models.

Angle of Attack	RANS (k- ω)				URANS (k- ω)				URANS (k- ϵ)				URANS (Spalart-Allmaras)			
	Drag (N)	Lift (N)	C_D	C_L	Drag (N)	Lift (N)	C_D	C_L	Drag (N)	Lift (N)	C_D	C_L	Drag (N)	Lift (N)	C_D	C_L
0	2.6	0.0	0.012	0.00	2.7	0.0	0.012	0.00	2.9	0.0	0.013	0.00	2.8	0.0	0.012	0.00
4	3.6	74.5	0.016	0.33	3.0	91.8	0.013	0.41	3.3	92.9	0.015	0.42	3.0	92.8	0.014	0.42
8	5.3	148.3	0.024	0.66	4.0	178.0	0.018	0.80	4.3	182.0	0.019	0.82	4.1	181.0	0.018	0.81
10	6.1	186.0	0.027	0.83	5.0	216.0	0.022	0.97	5.3	222.0	0.024	0.99	5.0	221.0	0.022	0.99
11	6.6	211.7	0.029	0.95	5.7	233.0	0.026	1.04	6.0	241.0	0.027	1.08	5.7	239.0	0.025	1.07
12	7.4	218.8	0.033	0.98	6.6	248.0	0.030	1.11	6.9	258.0	0.031	1.16	6.5	256.0	0.029	1.15
13	8.9	218.9	0.040	0.98	7.8	258.0	0.035	1.16	7.9	272.0	0.035	1.22	7.5	269.0	0.034	1.20
14	11.8	213.0	0.053	0.95	9.6	262.0	0.043	1.17	9.2	283.0	0.041	1.27	9.1	279.0	0.041	1.25
15	18.8	192.8	0.084	0.86	59.3	190.9	0.266	0.85	11.1	288.0	0.050	1.29	11.7	279.0	0.052	1.25
16	56.1	213.5	0.251	0.96	67.0	200.8	0.300	0.90	14.0	281.0	0.063	1.26	18.3	252.0	0.082	1.13
17	79.2	227.3	0.355	1.02	76.9	214.8	0.344	0.96	20.8	238.0	0.093	1.07	65.0	202.7	0.291	0.91
18	80.8	206.3	0.362	0.92	81.0	212.2	0.363	0.95	55.9	209.3	0.251	0.94	71.8	198.0	0.321	0.89
20	79.2	188.1	0.355	0.84	101.1	248.8	0.453	1.11	64.4	189.5	0.288	0.85	85.3	208.6	0.382	0.93
22	104.0	227.0	0.466	1.02	114.3	254.7	0.512	1.14	90.7	225.8	0.406	1.01	97.3	217.1	0.436	0.97
24	111.4	229.7	0.499	1.03	131.8	270.5	0.590	1.21	115.1	257.5	0.515	1.15	109.4	223.5	0.490	1.00
26	143.4	274.7	0.642	1.23	149.8	285.2	0.671	1.28	144.8	296.9	0.648	1.33	153.1	292.7	0.686	1.31

Table 3.14: 2D CFD simulation results

This allows for a comparison between RANS and URANS in the 2D environment as well as a comparison of three three main turbulence models used in CFD.

Snapshots of the instantaneous flow speeds around the airfoils at angles of attack of 8° and 17° were captured for each of the four types of simulation, as shown in Figure 3.17. As can be seen, the behaviour is quite consistent across all presented simulations. At angles of attack of 8° , it can be seen that the flow remains attached to the airfoil and there is very little downstream turbulence as a result. However, when the stall angle is exceeded as shown in the 17° sims, the flow separates and causes oscillating vortex shedding in the downwind flow. The four simulations predict essentially identical behaviour in the pre-stall simulation, however in post-stall simulation there is a clear difference between the k- ω model and the other options. Both the Spalart-Allmaras and k- ϵ models appear to suppress vortex shedding as the separated flow remains smooth downwind with very little vorticity in the wake. Conversely, the k- ω model simulates significant vortex shedding around the airfoil in the post-stall regime. The zoomed-out views in Figure 3.17 show the vortices as the downstream areas of increased flow. The vortex shedding appears to be more regular for the URANS simulation compared to the RANS simulation, stemming from URANS' ability to approximate unsteady flow whereas the RANS simulation reports the time average.

The average lift and drag on each airfoil during each simulation was recorded and used to calculate the resulting drag and lift coefficients. These results are listed in Table 3.14, and plotted in Figure 3.18, where they are compared to the predicted 2D drag and lift coefficient curves that were generated in Sections 3.2.4 and 3.2.6. Despite the differences in predicted flows in Figure 3.17, relatively good matching can be seen between all four simulations, with the biggest difference being when stall occurs.

The RANS model, unsurprisingly, see the largest amount of variation as its curves struggle to converge in the post-stall regime, due to attempting to capture time-dependent behaviour using a time-averaging solver. It can also be seen to underestimate the pre-stall lift coefficient compared to the predicted curve more than the other simulations.

Between the three URANS simulation, they can all be seen to perform fairly similarly. From the flows captured in Figure 3.17, it is known that at an angle of attack of 17° the wake is much smoother for the k- ϵ and S-A simulations, resulting in their higher predicted lift and

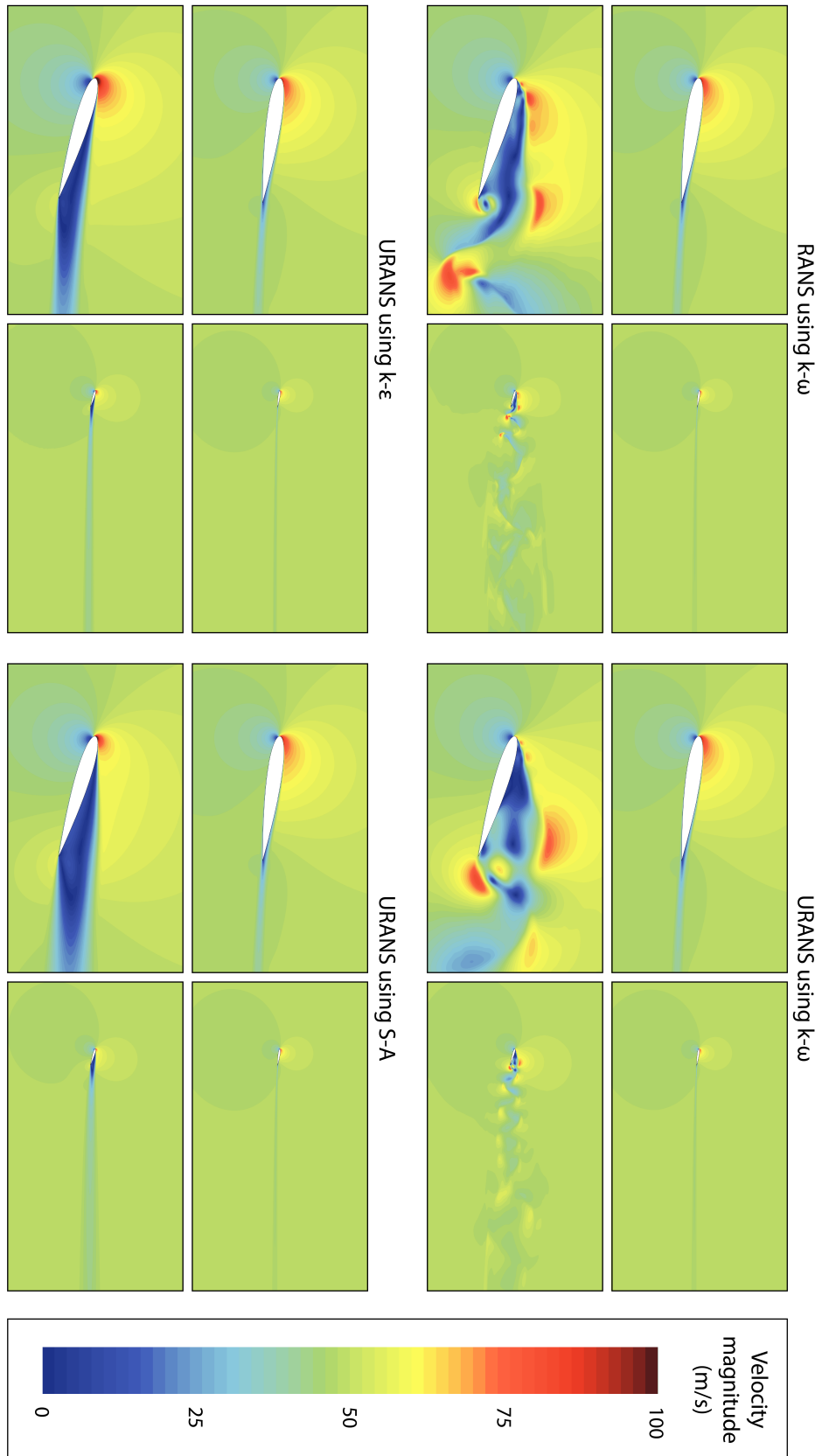


Figure 3.17: Instantaneous flow behaviour around the NACA 0012 airfoil at 8° and 17° for different 2D CFD simulations

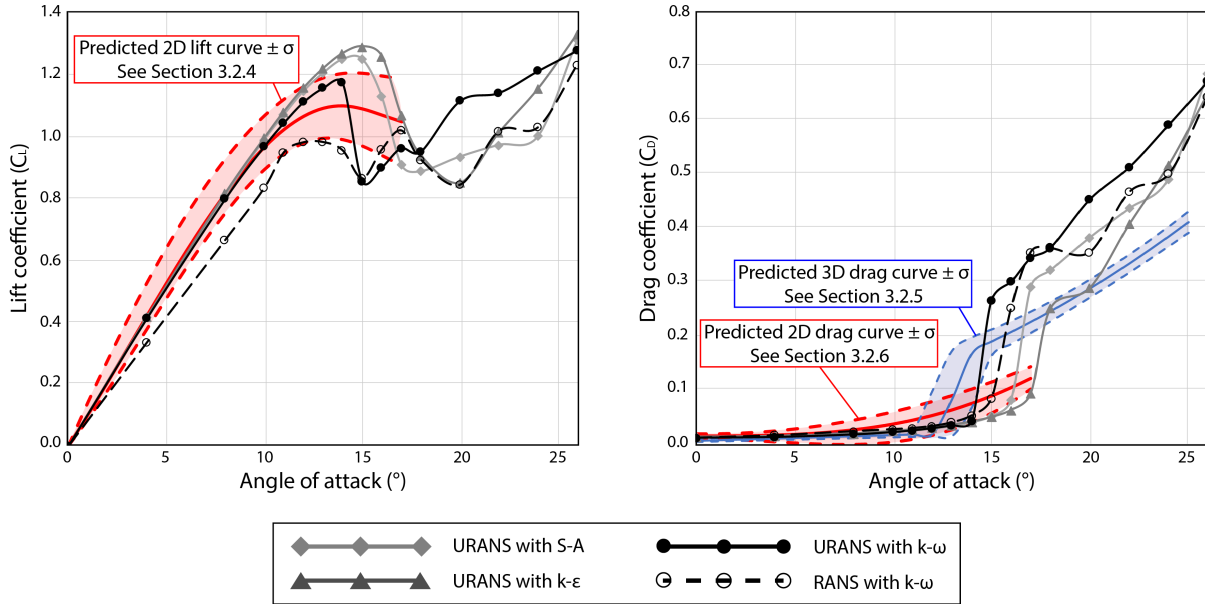


Figure 3.18: Lift and drag coefficient curves predicted by the 2D CFD simulations, including RANS and URANS simulations using Spalart-Allmaras, $k-\epsilon$, and $k-\omega$ closure models, compared to the predicted coefficient curves \pm one standard deviation (σ)

lower predicted drag compared to the $k-\omega$ simulations. Based on the flow snapshots and the recommendations by previous researchers, it can be argued that the $k-\omega$ model plots the smoothest curves and remains closest to the predicted model. Of the three, it also begins to stall the earliest and thus has the lowest maximum lift coefficient. Recall that delayed stall can be a sign of unintentional numerical damping occurring in the fluid simulation.

Comparing the predicted lift and drag curves to the generated CFD simulations, it can be seen that matching is decent, particularly for the $k-\omega$ simulations, though as the previous 2D simulations typically ended at the stall angle, the post-stall behaviour can not be compared. The matching of the predicted drag curve to the simulations is much worse than the lift curve. As discussed in Section 3.3.2, the previous 2D studies used to generate the predicted curve may have unintentionally restricted vortex shedding around the airfoils, which greatly reduces the resulting drag loads. It can be seen that matching is good until the jump at the stall angle, which is not captured by the 2D predicted model. If the predicted 3D drag coefficient curve is instead compared to these four simulations, matching is improved, but still not perfect. Compared to the 3D curve, these 2D curves stall later and jump to a higher drag value after stall occurs. As stated in Section 3.2.2, the tendency for 2D simulations to stall later with higher lift and drag forces has been documented in previous research, and since the URANS $k-\omega$ simulation stalls the earliest as well as the other listed advantages, it is recommended as the most accurate model compared to both predicted curves.

3.3.4 3D simulation results

Using the 3D mesh described in Section 3.3.2 with an across-wind span of 0.91 m as in Sheldahl and Klimas (1981), three types of 3D CFD simulations of the NACA 0012 airfoil at sixteen angles of attack ranging from 0° to 26° were performed, for a total of 48 simulations. Complete 3D RANS and URANS simulations using the $k-\omega$ closure model were performed, as well as preliminary 3D LES simulations. The total mesh size of approximately 7.2M cells was achieved using a coarser mesh than the 2D simulations: this mesh used a base size of 0.025 m as shown in Table 3.12. This reduction was required to create a mesh with the desired dimensions that could be run relatively quickly considering the high number of tests and available computational resources. Since LES simulations in particular require an extremely refined mesh to model sub-scale turbulence, this mesh was insufficient for accurate LES simulations of the NACA 0012 airfoil; as such the reported LES results are preliminary in nature and are presented to explore the broad trends of the technique as applied for stalled airfoils.

Instantaneous flow behaviour for each of the RANS, URANS, and preliminary LES simulations at an angle of attack of 14° are reported in Figure 3.19. In addition to a 2D flow velocity snapshot, the 3D vortex shedding is shown via isosurface of the Q-criterion. The Q-criterion is a function of the difference between the vorticity and viscous stress in the flow. In general, lower Q-criterion values will capture more of the vortex shape; a value of 10 s^{-1} was found to effectively display the vortices in this case.

As can be seen, all three models show turbulent flow in the wake of the airfoils, but the regular vortex shedding differs from the 2D simulations, possibly in part due to the coarser mesh used in the 3D tests and well as the removal of the flow simplifications found in 2D simulations. Overall, the following observations can be made by comparing the three models: firstly, while all three simulations capture flow separation around the inclined airfoils, the resulting turbulence differs between the simulations. In the time-averaged RANS simulation, the turbulence settles into regular vortex shedding; in the URANS simulation milder vortex shedding can be seen around the airfoil that quickly dissipates downwind; and in the LES simulation the flow moves even further away from regular vortex shedding, showing irregular turbulence due to smaller vortices on the upper edge of the airfoil, resulting in an irregular, turbulent wake. The 3D plots of the vorticity isosurfaces around the airfoil corroborate these observations from the 2D flow fields: the RANS simulation shows regular vortex shedding that detaches non-uniformly along the length of the airfoil and continues downstream a long distance; the URANS simulation predicts smaller vortices that dissipate quite quickly; and the LES simulations show much more complex behaviour made up of a number of smaller sub-scale vortices, which result in long, along-wind vortices in the wake. This latter observation shows the ability for LES to simulate improved near-wall flow behaviour which can result in more accurate simulations, but also its notable increase in computational intensity.

The resulting average drag and lift coefficient curves were extracted from the simulations. Note that as opposed to the 2D simulations where a unit length was included in the denominator of the calculation, the 0.91 m domain width is taken to calculate the resulting coefficients. Table 3.15 lists the recorded drags and lifts and the resulting coefficients from the simulation, using the same methods as Section 3.3.3. Figure 3.20 plots these results and compares them to the 3D drag and lift curves predicted in Sections 3.2.3 and 3.2.5.

Figure 3.20 shows many of the same trends as were seen in Section 3.3.3; both the URANS

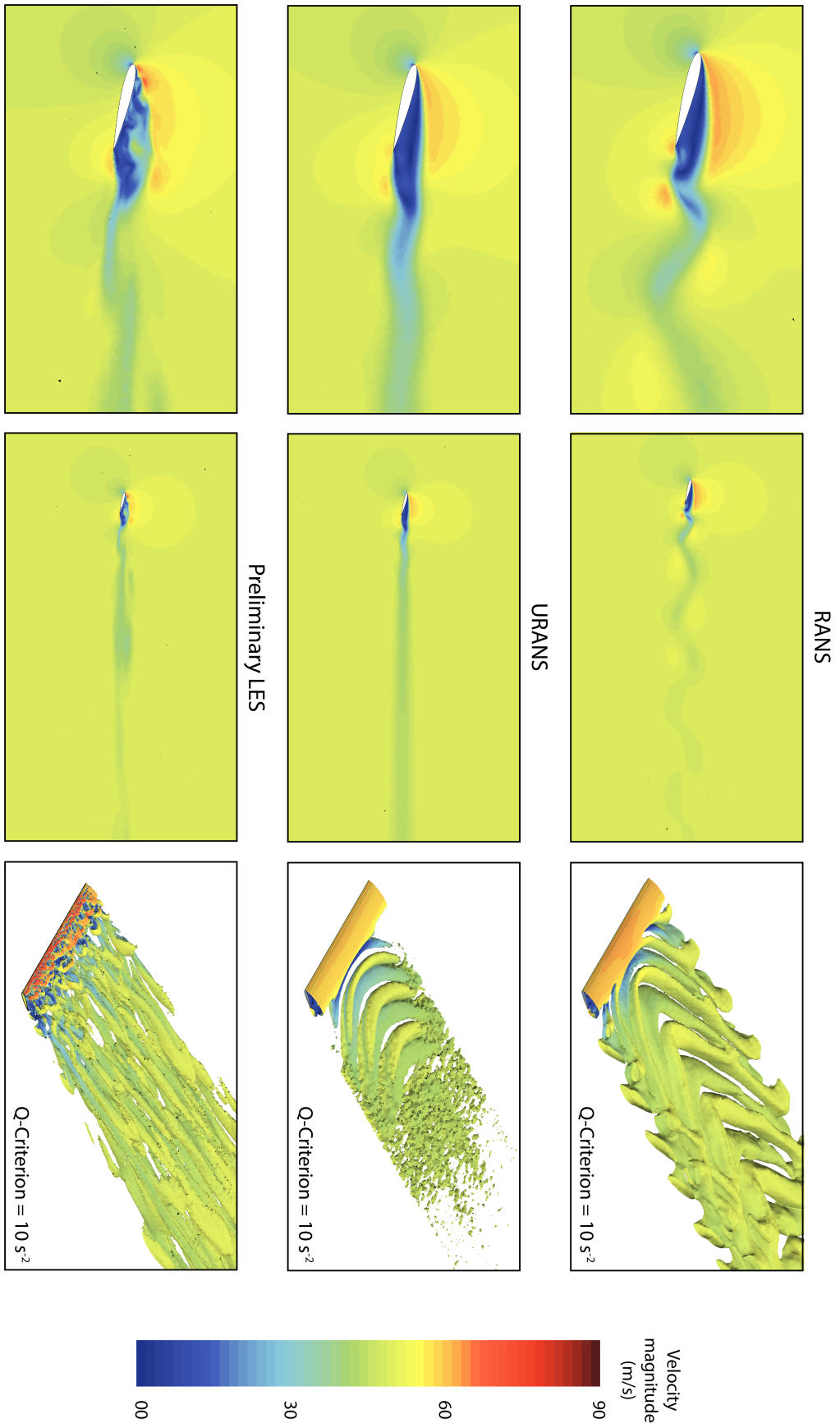


Figure 3.19: Instantaneous flow behaviour around the NACA 0012 airfoil at 14° from different 3D CFD simulations. 2D flow velocities were taken at the center of the span of the 3D airfoil

Angle of attack	RANS				URANS				Preliminary LES			
	Drag (N)	Lift (N)	C_D	C_L	Drag (N)	Lift (N)	C_D	C_L	Drag (N)	Lift (N)	C_D	C_L
0	0.81	0.0	0.012	0.00	1.70	0.0	0.013	0.00	1.91	-0.2	0.009	0.00
4	0.91	27.6	0.014	0.41	1.94	54.5	0.015	0.41	2.72	75.6	0.013	0.37
8	1.26	53.6	0.019	0.80	3.29	103.1	0.025	0.77	6.10	99.5	0.030	0.49
10	1.59	65.2	0.024	0.97	4.54	120.8	0.034	0.90	9.52	112.4	0.047	0.55
11	1.87	69.7	0.028	1.04	5.84	126.3	0.044	0.94	22.28	139.7	0.110	0.69
12	2.22	73.1	0.033	1.09	7.52	123.2	0.056	0.92	28.73	142.9	0.141	0.70
13	2.73	73.7	0.041	1.10	16.07	91.9	0.120	0.69	29.44	139.9	0.145	0.69
14	3.62	68.1	0.054	1.02	19.91	83.5	0.149	0.62	38.16	135.7	0.188	0.67
15	4.18	55.2	0.062	0.82	22.19	72.8	0.166	0.54	39.63	122.9	0.195	0.60
16	14.95	46.1	0.223	0.69	24.55	67.3	0.183	0.50	43.49	123.9	0.214	0.61
17	15.63	40.7	0.233	0.61	26.42	66.8	0.197	0.50	44.05	106.6	0.217	0.52
18	18.06	45.2	0.270	0.67	28.40	67.9	0.212	0.51	45.97	104.3	0.226	0.51
20	20.64	48.8	0.308	0.73	32.39	70.9	0.242	0.53	47.49	101.9	0.234	0.50
22	22.02	46.9	0.329	0.70	36.62	74.3	0.273	0.55	50.83	100.2	0.250	0.49
24	23.45	46.0	0.350	0.69	41.32	78.3	0.308	0.58	54.36	103.3	0.268	0.51
26	31.45	59.3	0.470	0.89	45.99	81.4	0.343	0.61	57.78	109.5	0.284	0.54

Table 3.15: 3D CFD simulation results

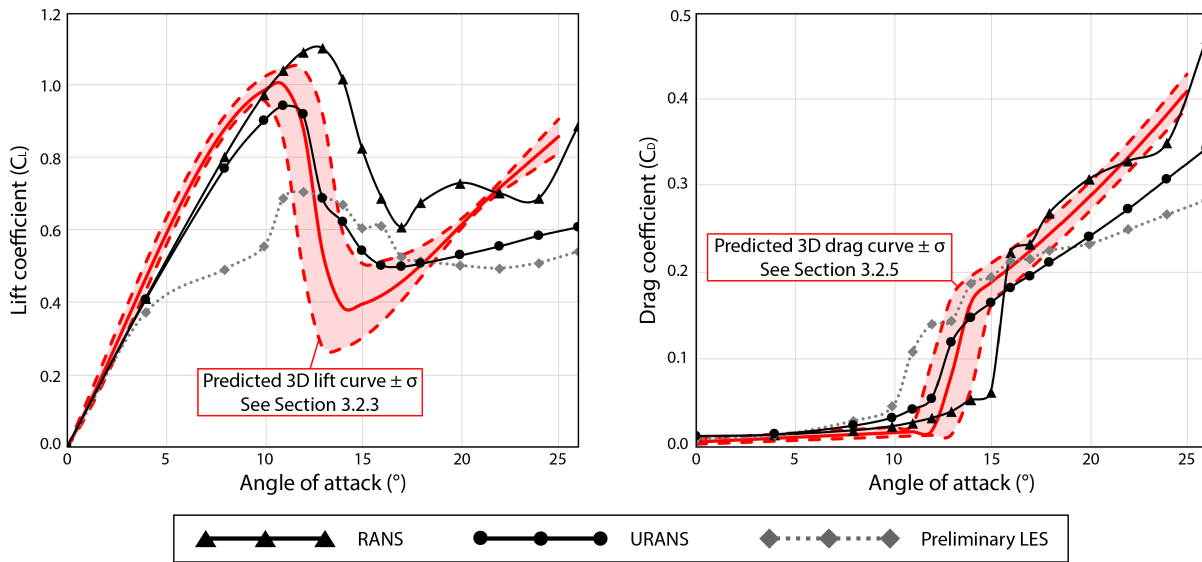


Figure 3.20: Lift and drag coefficient curves calculated by the 3D CFD simulations, including RANS and URANS simulations using the $k-\omega$ closure model and preliminary LES simulations, compared to the predicted coefficient curves \pm one standard deviation (σ)

and RANS simulations follow the expected shape of the lift and drag curves, though they do not necessarily fall in to the narrow bounds of the synthesized curves. Once again the steady-state RANS simulation has greater variance in the post-stall region compared to the URANS simulations, as well as being seen to stall at a higher angle of attack. However, in this case it can be seen that the RANS simulations match the predicted curves on average more closely than the URANS simulations in the post-stall region, despite the URANS simulations once again plotting extremely smooth coefficient curves that show better matching pre- and at-stall.

Due to the relative coarseness of the mesh compelled by the large number of simulations and the available computational resources, the results of the preliminary LES simulations matched the expected results relatively poorly. As can be seen in Figure 3.20, the preliminary LES simulations strongly underestimate the lift as the airfoil approaches the stall angles when compared to the predicted curves as well as the URANS and RANS simulations. The post-stall behaviour matches the URANS simulations more closely, and the predicted drag curve matches decently with the exception of the early stall jump and a lower predicted drag at very high angles of attack. It is clear that a finer mesh is required by the LES simulations to accurately model stall behaviour around the airfoil in future simulations.

Comparing the coefficient curves in Figure 3.20 to the captured flow behaviour in Figure 3.19 provides explanations for the differences in the predicted lifts and drags in the RANS, URANS, and preliminary LES simulations. The URANS and LES simulations match quite closely at the angle of attack of 14° , while the RANS simulations predicts much higher lift and much lower drag comparatively. Based on the vorticity isosurfaces, it can be seen that the large, lasting vortices that are regularly shed by the airfoil in the RANS simulations are the cause of this high lift and low drag; as the vortices become smaller and weaker in the URANS simulations, or are subsumed by the smaller sub-scale vorticities in the LES simulation, the lift drops and the drag increases notably. Thus, Figure 3.19 interestingly captures the broad flow behaviour as the airfoil transitions from at-stall to post-stall behaviour.

Based on these results, it must be concluded that the URANS simulations were overall the most effective type of CFD simulation in this study given the available computational resources. URANS was capable of capturing the time-dependent stall behaviour more effectively than steady RANS simulations, and its reduced computational intensiveness compared to LES potentially resulted in improved stall simulation for the given mesh size limited by the available computational resources. It is recommended that the URANS simulations be performed in 3D which has been shown to capture more realistic behaviour of an airfoil compared to idealized 2D simulations.

3.4 Conclusions

This study aimed to characterize the aerodynamic loads on the NACA 0012 airfoil in Reynolds numbers (Re) flows of 10^5 to 10^6 . Due to the complex aerodynamic behaviour of airfoils, particularly within this Re range, this characterization is more difficult to achieve than it would initially seem. The first section of this study collects and examines existing experimental and numerical studies of the NACA 0012 airfoil in flows of the given Re range performed in both 2D and 3D. It then attempts to synthesize numerical averages of these collected results by identifying common trends which are used to develop piecewise polynomial systems. This is

performed for lift and drag coefficient curves generated using 2D and 3D testing, for a total of four predicted curves. The second section of this study details and reports on a series of 2D and 3D CFD simulations, comparing the effectiveness of a variety of turbulence and closure models. Based on these results, a series of optimal CFD parameters can be recommended.

Seven previous studies of the NACA 0012 airfoil in Re flows of 10^5 to 10^6 were examined, including five sets of experimental studies and two sets of 2D numerical simulations. The predicted drag and lift coefficient curves from these studies were collected, reported, and plotted. They were then synthesized to generate piecewise polynomial systems by identifying and quantifying similar trends between the curves such as the peak lift coefficient value. Despite the relatively low number of existing studies, four curves for the 3D lift, 2D lift, 3D drag, and 2D drag coefficients were generated that matched the existing data well. This synthesis method was shown to be an effective way to combine a number of different reported curves into an average for easier comparisons to new results, and to help predict curves at Re where existing data is limited, provided tests at Re above and below can be found.

A total of 112 CFD simulations were run in 2D and 3D at various angles of attack using RANS and URANS with Spalart–Allmaras, $k-\epsilon$, $k-\omega$ closure models, as well as preliminary 3D LES simulations. The results of these CFD simulations are compared to the synthesized lift and drag curves from the literature analysis. Overall, matching between the CFD-generated curves and the predicted curves is not perfect. While the general shapes match quite well, the CFD simulations often lie outside the one-standard-deviation range plotted for the predicted curves. This limitation likely stems from a combination of the limited number of previous examples for the predicted curves as well as limitations in the mesh size or simulation length of the CFD simulations performed here. LES in particular requires a much finer mesh than was tested to truly capture the stalled airfoil behaviour.

Ultimately, it was shown that much like the existing studies, airfoil stall in this range of Reynolds numbers is a very complex phenomenon and there was disagreement between predicted curves and those generated by the CFD simulations. Still, provided a mesh sensitivity study is performed and criteria involving the y^+ values and the Courant number are met, URANS simulations using the $k-\omega$ closure model were overall the most effective simulation technique. The effectiveness of the $k-\omega$ closure model over other options has been noted previously, agreeing with the results here. Further, the URANS turbulence model was simpler and less intensive compared to the LES simulations allowing the optimized mesh to effectively perform the simulation given the available computational resources, while still approximating the unsteady post-stall vortex shedding on the airfoil better than the steady-state RANS simulations. URANS likewise has the advantage of being simpler to perform for novice CFD users compared to LES simulations. Previous studies have identified that the idealized 2D airfoil simulations tend to hinder vortex shedding and over-predict the maximum lift and the stall angle. The simulations here corroborate these observations, meaning that 3D simulations are recommended over 2D simulations given the availability of sufficient computational resources. These observed best-practices are thus recommended for future CFD simulations of airfoils including of full wind turbine blades.

Chapter 4

Optimizing the type and architecture of neural network-based surrogate aeroelastic wind turbine blade models

Summary

Current numerical aerodynamic wind turbine blade models – models that calculate the aerodynamic loads on a blade based on the incoming wind – are generally computationally intensive. However, a surrogate aerodynamic blade model consisting of a neural network (NN) trained on high-accuracy data should be capable of achieving both high accuracy and fast computation times. In this study, six different NNs were trained using numerical data from aeroelastic blade simulations, then compared for the highest possible prediction accuracy while maintaining high speeds. The NNs include two time-independent multilayer perceptron (MLP) networks with a full and reduced set of input data. A time-dependent long short-term memory (LSTM) network was also trained, along with a similar Pseudo-time-dependent MLP. Finally, a time-dependent convolutional neural network (CNN) and a similar Multi-time-step MLP were trained and compared. Over 30 million data points of aeroelastic time history information of an operational 5 MW wind turbine across 115 time histories and 23 different mean wind speeds were numerically generated for training the networks. The architecture, input and output data, and the optimization and training processes are provided in detail for each network, and the resulting accuracy and computation times are analyzed and compared. Ultimately, it was found that both the trained full-input MLP and the CNN were particularly accurate surrogate models, with average normalized root mean square errors of 1.11% and 0.66% respectively. While the CNN surpasses the full-input MLP in accuracy, the latter is simpler to train and faster to run, thus both are compelling options for future researchers. Such a surrogate model could be used

to predict the aerodynamic loads on a wind turbine blade when computation speed is a priority, such as during design optimization or hybrid simulations.

This chapter is an updated reproduction of the author’s published paper “Comparison of neural network types and architectures for generating a surrogate aerodynamic wind turbine blade model” (Lalonde et al., 2021).

4.1 Introduction

As the global threat of climate change continues to grow, the widespread adoption of wind turbines for energy generation is critical. The global installed capacity of wind energy was over 650 GW as of 2020 (Lee and Zhao, 2021), and this continues to increase. Further engineering refinement will help increase the safety and efficiency of these structures, thus enabling more complex engineering studies of wind turbines is of great value.

In numerical analyses of wind turbines, the aerodynamic calculations of the blades are the most intensive part. The aerodynamic calculation consists of determining the loads on a blade from the incoming wind, which is affected by the blade’s position, the airfoil geometry, and the rotation speed of the rotor. Several numerical methods exist for performing these type of calculations, but they tend to either compute quickly with lower accuracy, such as the blade element momentum (BEM) technique (Glauert, 1935), or to be computationally time-consuming but high in accuracy, such as computational fluid dynamics (CFD) (Dagnev and Bitsuamlak, 2013). However, surrogate aerodynamic blade models in the form of a trained neural network (NN) have the possibility of achieving both high speed and high accuracy provided that quality training data is available.

Neural networks (NNs) are powerful machine learning tools that can be trained to model relationships accurately without directly simulating the underlying behaviour. Input data is connected to output data via neurons, which are trained to accurately predict the output from the input. Thus, a properly trained neural network could be used to predict the resulting loads on a wind turbine blade from a given wind field without performing the time consuming aerodynamic calculations that would be required using a direct model.

Such a surrogate model would be advantageous due to its high speed and ability to be combined into other testing. Expedited testing is often quite valuable during the preliminary stages of any engineering project – a high-accuracy, trained surrogate blade model would allow for quick testing of wind turbines with greater flexibility than the more-restrictive programs that would be used to generate the network’s training data in the first place. Expedited simulations are also useful when performing bulk analyses, such as during parametric optimization. Additionally, the high speed of the model would allow it to be combined into types of testing with strict physical requirements, such as real-time hybrid simulation (RTHS) (see McCrum and Williams (2016) for an introduction to this topic). Lalonde et al. (2019) have previously examined the value that RTHS offers for studying wind turbines, and have noted that the promising technique is currently limited by the lower-accuracy modelling techniques currently used in this system (Lalonde et al., 2020a). A neural network-based surrogate aerodynamic wind turbine model is thus a promising option for improving the accuracy of RTHS of wind turbines.

However, neural network design and training is a complex process: there are a huge number of possible architectures and types of neural networks, and if trained improperly, a surrogate

model is likely to be of low accuracy and value. In this study, the open-source turbine modelling program OpenFAST is used to quickly generate a large number of response time histories of a 5 MW wind turbine under operational conditions. This data is used in MATLAB 2019b to train six different NNs to act as surrogate models for the wind turbine blades, including multilayer perceptron (MLP), long short-term memory (LSTM) and convolutional (CNN) neural networks. The accuracy and speed at which these surrogate models can predict the blade loads under given wind speeds at each step of a time history are evaluated and contrasted to determine the feasibility of this technique as well as the optimal neural network type and architecture. Section 4.2 of this paper presents a summary of background information about neural networks for readers unfamiliar to the topic. Section 4.3 details the generation of the training data and the general training process. Section 4.4 trains and analyzes two time-independent MLP networks – one that was supplied the full set of aerodynamic data as the input (the Full MLP) and on that was supplied a subset of data for improved usability (the Reduced MLP). Section 4.5 investigates a time-dependent LSTM network and a similar Pseudo-time-dependent MLP. Section 4.6 trains and studies a time-dependent CNN and a similar Multi-time-step MLP. Finally, Section 4.7 summarizes the effectiveness of the different NNs and draws conclusions.

4.2 Neural Network Theory and Application

Neural networks are a form of machine learning that mimics nodes in the human brain to indirectly model the relationship between input and output data using an arbitrary number of nonlinear equations. These equations are optimized based on the provided training data – the networks learn to accurately predict the supplied output from the supplied inputs during this training process. NNs can be used to model any continuous function (see Fortuner (2017) and Nielsen (2019) for simple proofs of this concept), however, care must be taken by designers when training NNs, as it is easy to develop misleading and inaccurate models if done poorly. The various networks employed in this study are all supervised regression models.

This section of the paper presents previous applications of neural networks for structural engineering applications, with particular focus given to their application in wind turbine studies, followed by a foundational explanation of neural network architecture, training, and the three types of networks employed in this study: MLPs, LSTMs, and CNNs.

4.2.1 Previous Applications

NNs (and other forms of machine learning) have seen applications in an enormous number of fields, often for classification or prediction purposes, and have been receiving greater focus by structural and wind engineers in recent years. MLPs are the most common type of NN used in these fields, but examples of more complex networks will be highlighted here.

Previous application of NNs in structural engineering and fluid mechanics have been well documented; Salehi and Burgueño (2018) have prepared a thorough review of artificial intelligence methods (including MLPs, LSTMs, CNNs and other types of machine learning) in structural engineering. Amezcuita-Sanchez et al. (2016) have prepared a literature review of NN use in civil engineering and identified the five main areas of application: structural system identification, structural health monitoring, structural control, material property prediction, and

design/optimization of structures. For a historical perspective, Adeli (2002) provides a detailed literature review of neural networks in civil engineering in the previous century.

Some notable examples of complex NNs used in structural engineering include Kao and Loh (2011) using an LSTM to predict the static deformation of a dam based on water levels and temperature. Luo and Xiong (2019) used an LSTM-like NN to relate external loading and fatigue loading in structural components, and Bieker et al. (2020) used a similar network for predictive control of a flow field. Further, Kiranyaz et al. (2019) have composed a summary of the use of CNNs for seismic damage prediction of structures, and Sony et al. (2021) have composed a summary of the use of CNNs for structural health monitoring.

NNs have likewise seen a wide variety of applications in wind engineering; Kareem (2020) offers a recent overview of machine learning in this field. Similarly, Brunton et al. (2020) offers a review of machine learning in fluid mechanics. Other notable applications include using neural networks for studying the interference effect from neighbouring buildings (Khandui et al., 1997; Hu et al., 2020), for modelling the effect of topography on wind flow (Bitsuamlak et al., 2007), for predicting the response of bridge decks under wind loading (Wu and Kareem, 2011; Li et al., 2018), for aerodynamic optimization of buildings (Elshaer et al., 2017; Elshaer and Bitsuamlak, 2018), and for predicting wind speeds (Madhiarasan and Deepa, 2017). Wind-loaded cylinders have been studied using CNNs (Jin et al., 2018) and other machine learning techniques (Hu and Kwok, 2020). The pressure loads on residential builds have been likewise been studied using MLPs (Bre et al., 2019; Tian et al., 2020).

NNs have also seen some applications for studying wind turbines from both structural and wind engineering perspectives. Where applicable, the architecture and error of the network will be reported, however note that different studies quantify the error of the network using different metrics (normalized root mean square error (NRMSE), mean absolute percentage error (MAPE), and R^2 are all common metrics), so consider these errors as rough approximations that should not be precisely compared.

- CNNs have been applied for structural health monitoring of turbines; see the review by Stetco et al. (2020) and the modern study by Yang et al. (2021). In a similar vein, Jimenez et al. (2017) used an MLP with 15 inputs and 6 outputs to classify delamination of wind turbines blades, with a success rate of 91%.
- A variety of NNs have been used to predict the power production of wind turbines; see reviews by Colak et al. (2012) and Table 1 of Lin and Liu (2020). Modern examples of this include Yin and Zhao (2019) using an LSTM with 6 inputs and 2 outputs to predict power production with an error of $< 1\%$. Lin and Liu (2020) used an MLP to predict power production with 11 inputs, 1 outputs and an error of 9%.
- Material property prediction of the stiffness and fatigue life of a turbine blade was performed using an LSTM and a CNN by Liu and Leng (2020). With 30 inputs and 5 outputs, both models were able to predict the stiffness curve of the blade with an extremely low amount of error. However, due to the very low amount of training data and the value of the resulting error ($< 0.001\%$), the authors believe that network overfitting may be an issue with this study.
- Wake prediction of wind farms has been explored using NNs. Richmond et al. (2020) used an MLP (among other machine learning techniques) to predict the downwind wind

speed. Their MLP with 2 inputs and 1 output was only able to achieve an error of 12%. Ti et al. (2020) developed an impressive set of MLPs capable of predicting a 2D vertical cross-section of the wake behind turbine(s). Using only 2 inputs, they were able to predict the resulting wind speed at 240,000 outputs with an average error of only 2-5%.

- NNs have also been used in optimization of wind turbines. Optimizing blade and airfoil geometries of wind turbines blades using NNs has been attempted by Ribeiro et al. (2012) who achieved an error of 6% using an MLP with 9 inputs and 2 outputs, and by Sessarego et al. (2020) who achieved an error of 1% using an MLP with 6 inputs and 2 outputs. Zhao et al. (2019) used a MLP to predict the optimal parameters of a damping device for wind turbines, achieving an accuracy of 1% with 2 inputs and 2 outputs.

It can be seen that NNs offer a number of applications in wind turbine research. Some general trends that can be seen are that while most research is capable of developing a network of acceptable accuracy (error $\leq 5\%$), NNs with more than a tiny number of inputs and outputs struggle to achieve very high accuracy (error $\leq 1\%$). Further, of the five applications of NNs for structural engineering, NNs for structural system identification are underrepresented when studying wind turbines. This paper will detail the development of high-accuracy NNs with a large number of inputs and outputs for developing a surrogate aerodynamic blade model, which represents a type of structural system identification.

4.2.2 Neural network architecture

As stated previously, neural networks are trained to connect input data to output data without performing the underlying equations that connect them. The architecture of NNs is built out of a series of layers as shown in Figure 4.1. The input layer of the network is connected to the output layer by an arbitrary number of *hidden layers*, and each layer is built of a certain number of *nodes*. Each node in the input layer represents an individual value in the layer, thus a network with ten inputs will have ten nodes in the input layer. NNs with multiple hidden layers are sometimes referred to as *deep* networks.

The most common type of hidden layer in a NN is a *fully-connected layer*. The network shown in Figure 4.1 is made entirely of fully-connected layers. In a fully-connected layer, each node is connected to every node in the layer before and after it, and the value of the node is a weighted sum of the nodes in the previous layer, plus an internal bias, passed through a nonlinear activation function such as tanh or ReLU to produce the overall output of the node, as defined in Equation 4.1:

$$x_{ij} = g \left(b_{ij} + \sum_{k=1}^{K_{i-1}} x_{(i-1)k} W_{(i-1)kj} \right) \quad (4.1)$$

Here, x_{ij} is the value of the j th node on the i th hidden layer, $g()$ is the activation function, b_{ij} is this node's bias, $x_{(i-1)k}$ is the value of node k on the previous layer, $W_{(i-1)kj}$ is its respective weight, and K_{i-1} is the total number of nodes on the previous layer.

MLP NNs are built entirely of fully-connected layers as shown in Figure 4.1, which is the cause of their relative simplicity. Conversely, LSTM and CNN include additional complexities that will be discussed in Sections 4.2.5 and 4.2.6.

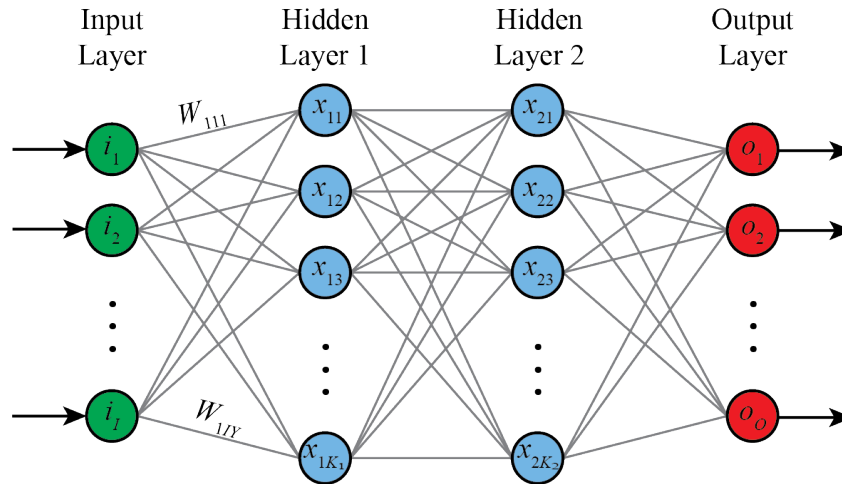


Figure 4.1: Generic NN architecture with two fully-connected hidden layers

4.2.3 Training process

Training a neural network refers to optimizing the weights and biases in the network, as well as the *hyperparameters*, such that the predicted output is as accurate as possible. This is done by supplying the NN with a set of training data that includes input data with its associated output data. The network then learns to predict the output accurately using the training data as a guide. As a rule, the longer a NN is allowed to train and the more training data it is provided, the more accurate the network will be. This also means that the neural network is only valid for predicting outputs of inputs that fall within the bounds of the training data, as it will not have learned how to accurately predict outputs beyond those bounds.

Training the NN occurs in two stages: first the hyperparameters are optimized, then the final NN is trained using the full training data set with these optimized hyperparameters. The term hyperparameter refers to any parameters of the network that are not directly optimized during the training, which includes values that define the architecture and training process such as the number of hidden layers, the size of the hidden layers, the learning rate of the network, and the number of epochs for training. Since the hyperparameters are not optimized during the training itself, an external optimization process (typically the Bayesian optimization algorithm – see Frazier (2018) for an introduction to this technique) is employed instead. Figure 4.2 shows the full optimization and training process employed in this study. This process involves specifying possible ranges of values for the hyperparameters, then during each step of the optimization process a NN is quickly trained on a subset of the training data using some combination of possible values for the hyperparameters, and the accuracy of each NN is recorded. After some number of rounds of optimization, optimal hyperparameters can be estimated, which are employed in the final network training.

During the final network training, a neural network is trained on the full training set using the optimal hyperparameters. Training of the neural network is done using a back-propagation algorithm – in this study the stochastic gradient descent with momentum (SGDM) method was used.

The optimization target of both processes is to lower the overall average normalized root

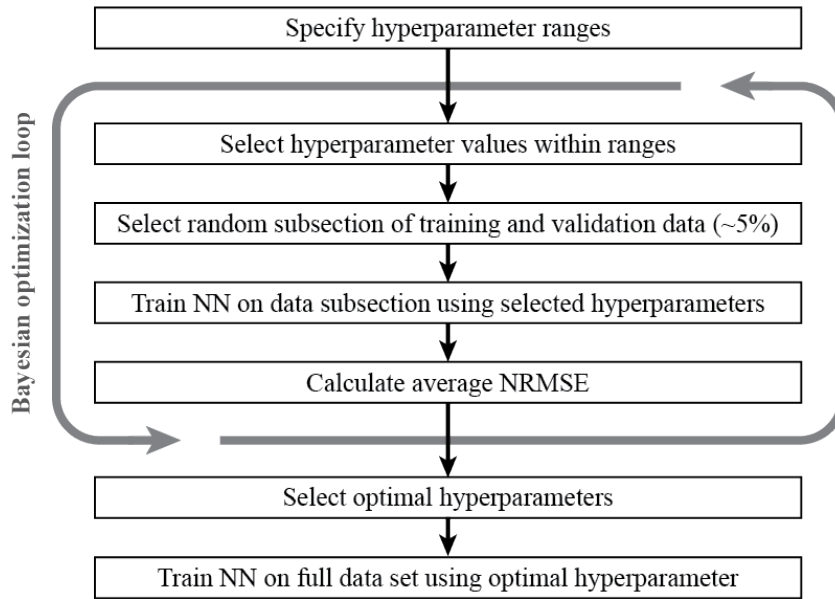


Figure 4.2: Flowchart of the optimization and training process of a neural network

mean square error (NRMSE) between the predicted and true outputs for all time histories for all outputs. After generating the predicted outputs for a given time history, the RMSE is calculated using Equation 4.2:

$$\text{RMSE} = \sqrt{\frac{\sum_{t=1}^T (\hat{o}_t - o_t)^2}{T}} \quad (4.2)$$

Here t is the current time step which ranges from 1 to T ; \hat{o}_t is the predicted output value at the given time step; and o_t is the true output at the given time step. The NRMSE calculation is then performed using Equation 4.3:

$$\text{NRMSE} = \frac{\text{RMSE}}{o_{max} - o_{min}} \quad (4.3)$$

Here o_{max} and o_{min} are the maximum and minimum values of o for the given time history, respectively. This NRMSE value is then averaged across all outputs and time histories to return a single value to be reduced by optimization.

4.2.4 Multilayer perceptron (MLP) networks

As discussed previously, MLPs are one of the most basic form of NNs, built entirely of fully-connected layers – an example of this architecture is shown in Fig 4.1. MLPs are a type of feed-forward neural network. Being quite generic, MLPs can be applied to nearly any machine learning problem, and are easier to train and implement while still capable of developing highly accurate models. However, this means that more specialized types of neural networks may be able to outperform the MLP, which is why LSTMs and CNNs are also investigated. In this study the MLPs are limited to studying each time step of a time history as an independent data

sample, thus its ability to consider time-based phenomena is very restricted, which was thought to pose an issue given the nature of the data used in this study.

4.2.5 Long short-term memory (LSTM) networks

LSTMs are a form of recurrent neural network (RNN), which are designed to process time series data. Rather than considering each time step as an independent data point, RNNs are capable of remembering data between time steps, and LSTMs in particular are designed to have long memory. This memory, also referred to as the *cell state*, allows the networks to consider previous data when predicting the current output, and thus LSTMs are theoretically able to achieve higher accuracy for time history processes than MLPs.

Figure 4.3 shows the layout of a LSTM network. As can be seen, the main difference between the LSTM and the MLP is the addition of the LSTM layer. Each LSTM layer contains the cell state that remembers data – at each time step, based on the input data, some data is forgotten by the cell state, some data is added to the cell state, then the cell state and the input data are used together to predict the layer output, using similar nodal equations to those employed in a fully-connected layer. For a very clear explanation of the internal workings of the LSTM layer, readers are directed to Phi (2018).

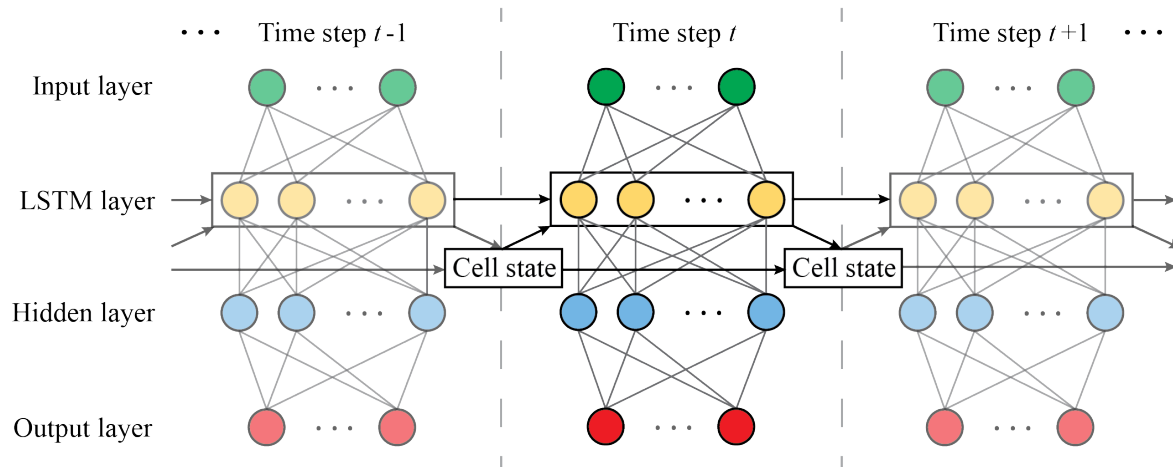


Figure 4.3: Layout of an LSTM NN with one LSTM layer and one fully-connected layer

The limitations of LSTMs are that the added complexity increases the time and computational requirements of training the network. Furthermore, if the temporal relationship between neighbouring data is not significant, the added complexity and variables to optimize can result in a network with reduced accuracy compared to modelling each time step independently.

4.2.6 Convolutional neural networks (CNNs)

CNNs are designed to process image data by passing *filters* (also known as *detectors*) over an image to compress salient features, applying an activation function, and then *pooling* the results. These steps are referred to as *convolution*, activation, and pooling respectively. By

performing this process multiple times, neurons farther down the network are connected to large areas of the original image, and the number of calculations needed versus an equivalent MLP is strongly diminished.

Figure 4.4 shows the architecture of a typical CNN. The convolution layers slide multiple small 2D windows over the input image and apply a filter (which can be considered as similar to the weights in a fully-connected layer). This is performed a number of times, then the filtered images are passed through the activation function and are pooled, which consists of reducing areas of images into their max or average values, i.e. if a 2x2 section of the image is pooled, then the resulting pixel's intensity will be the average intensity of the four pooled pixels, and the size of the image will be reduced by 3/4th. Pooling is performed to control the number of variables in the network and increase the receptive field of deeper layers. This convolution-activation-pooling process is often performed a number of times and followed by fully-connected layers to maximize the accuracy of the CNN.

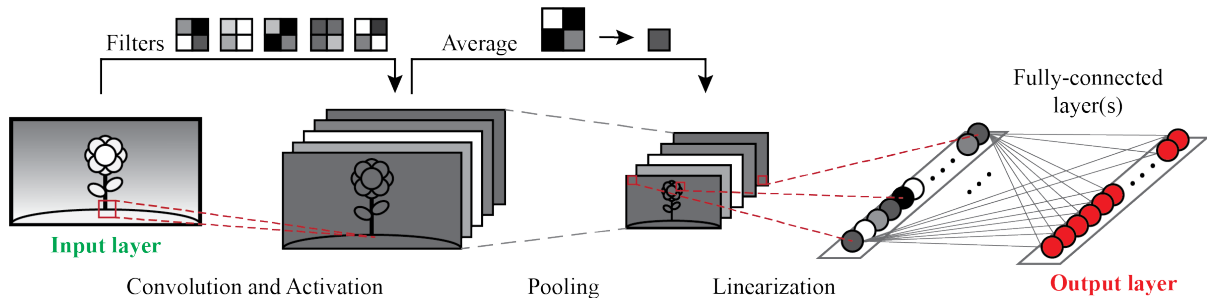


Figure 4.4: CNN network architecture

CNNs can be applied to time history data as a 1D CNN, in which time history data is normalized and arranged as a 1D grayscale image, where each pixel is one time step and the intensity of the pixel its corresponding value (i.e. white is equal to zero and black is equal to one). Passing an image such as this through a CNN will allow neurons deeper in the network to have a view of a number of adjacent time steps when training the network. However, in this study, since each time step has multiple inputs, rather than a grayscale image that has a single channel, a 1D image with a number of channels equal to the number of inputs is provided to the network to predict the resulting blade loads at each time step. Section 4.6.1 of this paper delves into this configuration in further detail. This form of CNN has similar advantages to LSTMs: the possibility of improved accuracy for analyzing time history data, however due to the size reduction of the network caused by the convolution and pooling, the risk of reduced accuracy due to increased complexity is lowered. There is no consensus within the machine learning community as to whether LSTMs or CNNs are superior for time-history analysis, thus both are considered in this study.

4.3 Methodology

The open-source wind turbine modelling software OpenFAST (Jonkman et al., 2021) was used in this study to generate the training data for the NNs.

Hub height wind speed (m/s)	Rotor speed (rpm)	Blade pitch (°)	Hub height wind speed (m/s)	Rotor speed (rpm)	Blade pitch (°)
3*	6.9	0	15	12.1	10.5
4	6.9	0	16	12.1	12.1
5	6.9	0	17	12.1	13.5
6	6.9	0	18	12.1	14.9
7	7.4	0	19	12.1	16.2
8	8.5	0	20	12.1	17.5
9	9.6	0	21	12.1	18.7
10	10.6	0	22	12.1	19.9
11**	11.7	0	23	12.1	21.2
12	12.1	3.8	24	12.1	22.4
13	12.1	6.6	25***	12.1	23.5
14	12.1	8.7	<i>*Cut-in; **Approximate rated; ***Cut-out</i>		

Table 4.1: Specified rotor speed and blade pitch for the given hub height wind speeds. Five simulations were run at each wind speed

Numerical simulation of an operational 5 MW wind turbine under a range of wind speeds was used in this study to generate the training data for the NNs. MATLAB was used to optimize and train the various NNs which was run on a supercomputing platform. This section specifies the details of the generated data and of the NN training processes.

4.3.1 OpenFAST simulations

The open-source wind turbine modelling software OpenFAST (Jonkman et al., 2021) was used to simulate the aeroelastic response of NREL’s standard 5 MW wind turbine (Jonkman et al., 2009). This turbine has a hub height of 126 m and a blade length of 63 m. The cut-in and cut-out hub height wind speeds for this turbine are 3 m/s and 25 m/s, respectively, and the turbine was operational for all simulations. Table 4.1 summarizes the operational conditions of the wind turbine at each wind speed considered in this study.

The wind fields used during these simulations were generated using the software TurbSim (Kelly and Jonkman, 2012), which is capable of generating wind field time histories according to the conditions specified by the IEC turbine design codes (IEC, 2005). In this case, the wind fields had the following specifications: a duration of 600 s, a von Karman turbulence model of normal-B IEC turbulence levels, a power law wind profile with exponent of 0.2 and a surface roughness of 0.05. For each of the wind speeds listed in Table 4.1, five distinct time history simulations were performed, each with a distinct random seed for generating the wind field. Thus a total of 115 OpenFAST simulations were performed – five trials each for the 23 distinct wind speeds.

OpenFAST performs aerodynamic calculations using the blade element momentum (BEM) technique (Glauert, 1935), which has seen wide usage and validation previously. The BEM technique consists of dividing the turbine blade into 2D airfoils, determining the drag and lift loads on the airfoil based on the wind speed and angle of attack, summing these loads along the length of the blade, and finally multiplying by some factors to account for 3D effects such

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
r (m)	63.0	61.8	58.7	56.2	52.5	48.9	44.6	40.2	36.6	32.3	27.9	24.3	20.0	15.6	12.0	8.3	5.8	2.7
r/R	1.0	.98	.93	.89	.83	.77	.70	.63	.57	.50	.43	.37	.30	.23	.17	.11	.07	.02

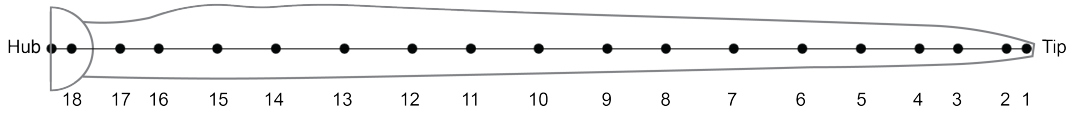


Table 4.2: Nodes along the turbine blade where data was recorded

as tip loss. More details about the BEM technique as well as examples of its use can be found in Section 2.4. BEM theory can be described as high-speed but low-accuracy. Since this study is exploratory in nature, lower-accuracy training data was deemed sufficient for finding the optimal NN types and architectures, which can be trained in future studies with higher-accuracy data.

The individual OpenFAST simulations were run at a time step of 0.02 s for the full 600 s wind field time histories, thus each time history consisted of 300,000 time steps. Across all 115 time histories, this resulted in a total of 34.5 million data points in the full data set. The ElastoDyn module was used for aeroelastic calculations of the blades. The input wind speed as well as the deflection and force responses were extracted at each time step at the hub and 18 nodes along one of the turbine blades. Since OpenFAST is only capable of extracting data from 9 points on a blade per simulation, each simulation was run twice to extract the data for the full 18 points. Table 4.2 specifies the location of each node along the 63 m blade.

Table 4.3 lists the various data points recorded at each step from the analysis, some at the hub and some at 18 points along one of the blades. Note that the velocity and acceleration are the instantaneous values at each time step. These data points have been grouped into five broad categories: **Geometry**, **Wind Speed**, **Deflected Shape**, **Structural Velocity/Acceleration**, and **Loads**. Thus, the Wind Speed category includes the wind speed recorded at the hub in the fore-aft and side-to-side directions, as well as the wind speed in the edgewise and flapwise directions at 18 points along the blade, resulting in 38 points of data per time step. These data groups will be used either as inputs or outputs to the NNs; the specific usages are discussed in Sections 4.4-4.6.

Figure 4.5 explains the different data and their coordinate systems recorded from the OpenFAST simulations. Fore-aft and side-side responses are measured at the hub, while flap and edgewise are measured at the blade and are relative to the orientation of the airfoil.

4.3.2 Data pre-processing

Data pre-processing consisted of normalizing the data and splitting it into training groups. Data must be normalized prior to training to prevent higher magnitude data from being overvalued. Data was normalized against its absolute maximum, which preserves the zero-mean nature of the structural vibrations and restricted the data to a range of $[-1, 1]$. Because this range was used rather than a more typical $[0, 1]$ range, the tanh activation function was employed in all neural networks rather than the more typical ReLU function.

After normalization, data was split into training, validation and testing sets. The training

Data group name	Recorded at hub (1 per time step)	Recorded at blade nodes (18 per time step)
Geometric (input)	Blade azimuth ($^{\circ}$)	-
	Rotor speed (rpm)	-
	Blade pitch ($^{\circ}$)	-
Wind Speed (input)	Wind speed, fore-aft (m/s)	Wind speed, edgewise (m/s)
	Wind speed, side-to-side (m/s)	Wind speed, flapwise (m/s)
Deflected Shape (input*)	Deflection, fore-aft (m)	Deflection, edgewise (m)
	Deflection, side-to-side (m)	Deflection, flapwise (m)
Structural Velocity/	Velocity, fore-aft (m/s)	Velocity, edgewise (m/s)
	Velocity, side-to-side (m/s)	Velocity, flapwise (m/s)
Acceleration (input**)	Acceleration, fore-aft (m/s^2)	Acceleration, edgewise (m/s^2)
	Acceleration, side-to-side (m/s^2)	Acceleration, flapwise (m/s^2)
Loads (output)	Shear force, fore-aft (N)	Shear force, edgewise (N)
	Shear force, side-to-side (N)	Shear force, flapwise (N)
	Moment, fore-aft (Nm)	-
	Moment, side-to-side (Nm)	-

*Not used in the *Reduced MLP* in Section 4.4.1; **Only used in the *PTD MLP* in Section 4.5.3

Table 4.3: Recorded data from OpenFAST simulations

data set is used to train the network. The validation data set is used to validate the training, ensuring that overfitting does not occur. The testing set is used to evaluate the trained network. In this case, a 60-20-20 distribution between the three sets, respectively, was chosen. Since each mean hub wind speed had five associated time histories, three per wind speed were given to the training set, one was given to the validation set, and one was given to the testing set. This ensured that there was an even distribution of mean wind speeds between each set. This meant that ultimately, 69 time histories were used to train each neural network, 23 were used for validation and 23 we used for testing.

4.3.3 Training Procedure

The Deep Learning Toolbox in MATLAB 2019b (Beale et al., 2019) was used for the construction and training of the NNs. While open-source tools such as the Python-based TensorFlow, PyTorch and Keras are more typical and more robust for machine learning research, MATLAB offers a straight-forward platform for NN applications, which is why it was chosen for this research as well as by Zhao et al. (2019). NNs are trained using the *trainNetwork* command, using the stochastic gradient descent with momentum (SGDM) method in the back-propagation algorithm due to its somewhat improved convergence speed compared to the other available options (Dogo et al., 2018). Network layers were specified in a vertical array – the specific layers used for each NN are detailed in Sections 4.4-4.6.

There are a numerous number of hyperparameters that define a NN, but the greater the number of optimization variables, the longer the Bayesian optimization process will take to perform. Thus, in this study a number of hyperparameters were chosen to have fixed, conservative values. These chosen hyperparameters affect the training time of the network rather than the accuracy, thus sub-optimal values for them would be unlikely to influence the accuracy of

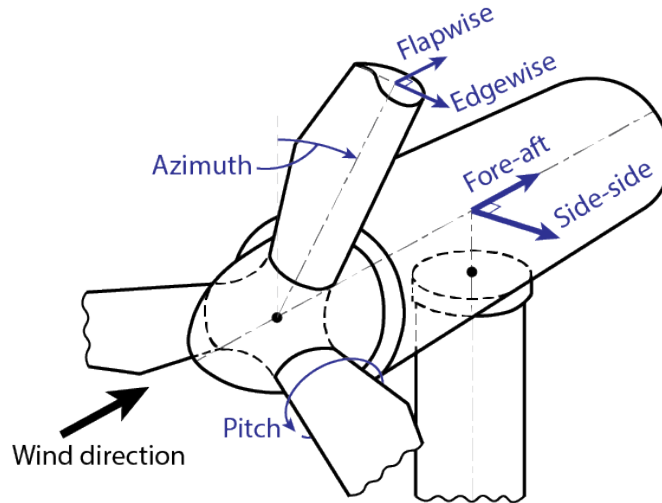


Figure 4.5: Recorded wind turbine coordinates systems and data

the final results. To reduce computation time, the networks trained during the hyperparameter optimization process were supplied with random sections of 5% of the training and validation data, which were evenly distributed across the mean wind speed time histories. The hyperparameter optimization process was performed twice per network: in the first round a very large range of values for each hyperparameter was given; and in the second round the range was reduced significantly, centered on the optimal value predicted in the first round. All of the values used for the hyperparameters are specified for each NN in the following sections. Bayesian optimization was performed in MATLAB using the *bayesopt* command, and 300 rounds of optimization was performed for each network.

Network optimization and training was performed in MATLAB on SHARCNET’s Graham supercomputing cluster using 16GB NVIDIA T4 Turing GPUs. The optimization and training time of each NN is reported in the following sections have been adjusted to as if they were run on a single GPU. As using a supercomputer for operation of a surrogate model is not feasible, the run times of the networks were instead measured on a local machine with a 4GB NVIDIA GTX 1050 Ti GPU. As the run times can vary slightly round-to-round depending on the values of the input data, the reported run times are the average value of five runs each.

4.4 Time-independent aerodynamic models

This section of the paper explores two MLP NNs trained as a surrogate aerodynamic turbine blade models. These models are considered time-independent as each point of training data is independent from all others; neighbouring time steps have no influence on the models. The two models developed in this section differ based on the provided inputs: a model with a full set of aerodynamic inputs was trained and contrasted to a model with a reduced set of inputs, the latter of which was explored due to its improved usability. This sections details the specifics of the input and output data of the models, the hyperparameter ranges, and the resulting accuracy and time requirements of the networks.

Full MLP		Reduced MLP	
Input groups	Output groups	Input groups	Output groups
Geometry Wind Speed Deflected Shape	Loads	Geometry Wind Speed	Loads
Total number of inputs per time step: 79	Total number of outputs per time step: 40	Total number of inputs per time step: 41	Total number of outputs per time step: 40

Table 4.4: Input and output data groups for the Full and Reduced MLPs, including total number of data points per time step. Table 4.3 details the contents of each data group

4.4.1 Input and output data

The two NNs developed in this section differ based on the input and output data that is supplied to the networks. For each time step, the input data and output data is arranged according to the data groups defined in Table 4.3.

The first network is referred to as the **Full MLP** network; it was supplied the full set of aerodynamic inputs to predict the resulting aerodynamic loads. Thus, the Geometry, Wind Speed, and Deflected Shape groups of data from Table 4.3 were supplied as inputs, which were used to predict the Loads group as outputs. By using a simpler, widely-used type of NN and receiving all the information required for aerodynamic calculations, the Full MLP model represents the baseline NN in this study, against which all other models will be judged.

The second network trained here is referred to as the **Reduced MLP** network. Compared to the Full MLP, the Reduced MLP was not provided the deflected shape of the turbine blade as an input to predict the resulting aerodynamic loads. The Reduced MLP was explored as an option as it would have a number of advantages if it could predict aerodynamic loads as accurately as the Full MLP. This would greatly reduce the scope of the simulations required to generate the training data, and the trained network would likely run faster due to its reduced number of inputs as well as be simpler to implement in a time-history simulation. However, it was expected that the Reduced MLP would have lower accuracy compared to the Full MLP due to the reduced number of inputs, the question was whether the drop in accuracy would be small enough to justify the increased usability. Table 4.4 summarizes the input and output data groups used in this section.

4.4.2 Network layers

Figure 4.6 lists the vertical array of NN layers used in MATLAB to define the architecture of the network, where the NN figure uses the same colour scheme as the MLP previously shown in Figure 4.1. In Figure 4.6: I refers to the number of inputs to the network; O refers to the number of outputs; and the *Layer Sizes* are hyperparameters defined in Section 4.4.3.

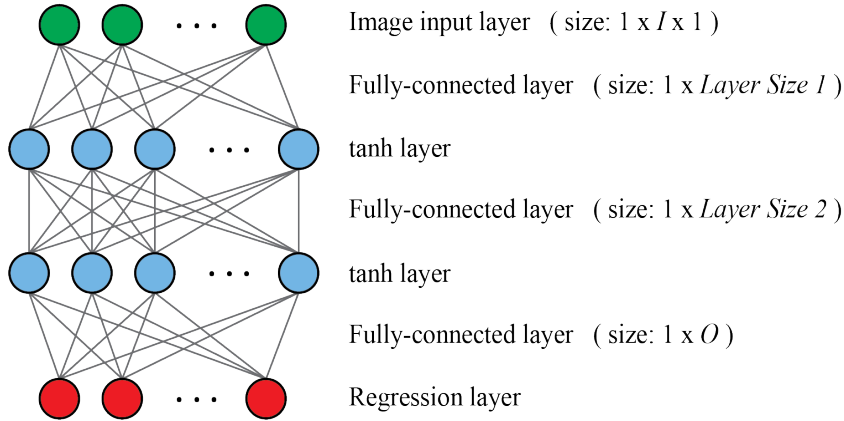


Figure 4.6: MLP layer architecture and sizes

Hyperparameter	Full MLP			Reduced MLP		
	Momentum	Layer Size 1	Layer Size 2	Momentum	Layer Size 1	Layer Size 2
Round 1 low	0.900	10	10	0.900	10	10
Round 1 high	0.999	700	700	0.999	700	700
Round 2 low	0.995	450	400	0.993	200	120
Round 2 high	0.998	600	550	0.996	280	200
Optimal	0.997	541	466	0.994	240	150

Table 4.5: Hyperparameter optimization results for time-independent MLPs

4.4.3 Optimized hyperparameters

As detailed in Section 4.2.3, two rounds of hyperparameter optimization were performed during training. The first round used a large range of possible hyperparameters, while the second round used a smaller range centered on the optimal value found in the first test. For the MLP networks, it was decided that the momentum, the size of the first hidden layer, and the size of the second hidden layer would be optimized, while the remaining hyperparameters were assigned fixed conservative values. These fixed values included using two fully-connected layers, performing 100 epochs of training during optimization and 500 epochs during the final training, using a batch size of 100, and an initial learning rate of 0.0001. Table 4.5 summarizes the hyperparameter optimization results.

4.4.4 Results

The average NRMSE of the trained MLPs across all outputs and time histories is presented in the upper right corners of Figure 4.7. It can be seen that the Full MLP achieves a notably low error of 1.11%. Recall that it was shown in Section 4.2.1 that MLPs applied in wind turbine studies with a large (>10) number of inputs typically have error of 5-10% (Ribeiro et al., 2012; Jimenez et al., 2017; Lin and Liu, 2020). Thus the accuracy of the Full MLP greatly exceeds the accuracy of previous large MLPs applied in wind turbines studies. Conversely,

the Reduced MLP achieves an average NRMSE of only 5.78% – a value roughly comparable to those previous NNs. The drop in accuracy stems from the reduced number of inputs to the network as well as not providing all the information that would be required to perform the aerodynamic calculations directly. It is reasonable to say that approximately tripling the error is not worth the increased usability offered by the Reduced MLP, thus the Full MLP is the recommended configuration of these options.

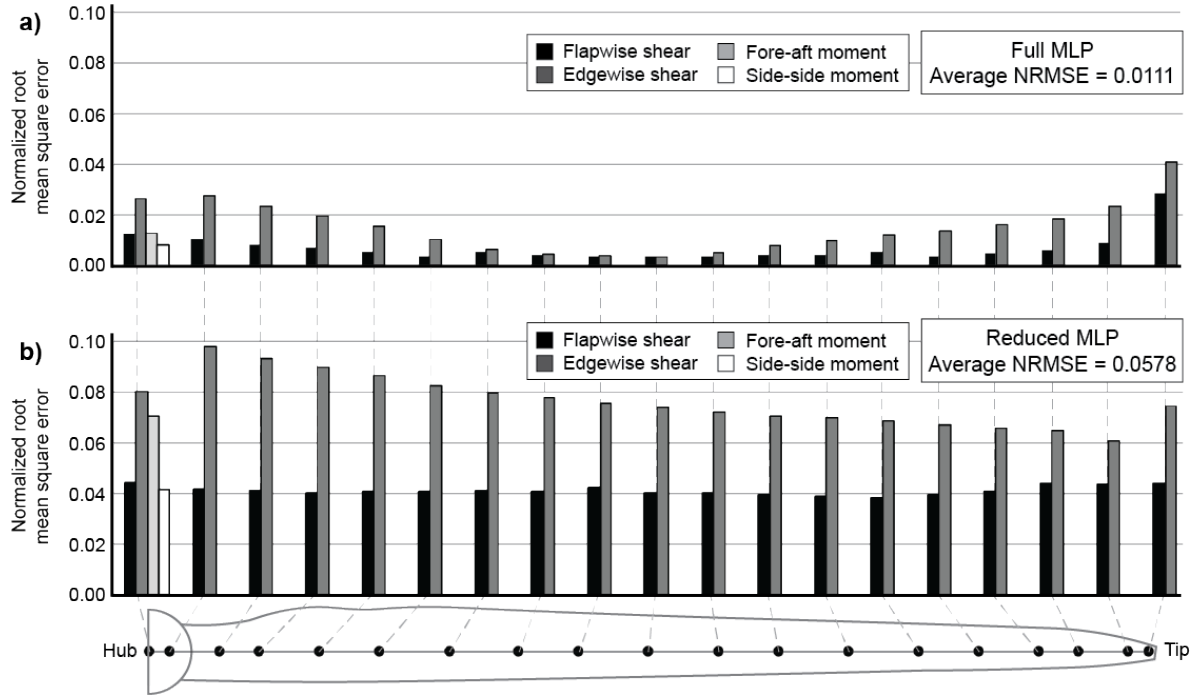


Figure 4.7: NRMSE of each response output averaged across all time histories for a) Full or b) Reduced MLPs

To examine the results in further detail, Figure 4.7 also compares the average NRMSE of the Full and Reduced MLPs across each output response using the same scale. The error in predicting the edgewise drag and flapwise lift loads at each node, as well as the fore-aft and side-to-side loads and moments at the hub averaged across all time histories are presented. The values on the left of the figure refer to the loads at the hub and continue to the right along the blade, as shown in the bottom of the figure.

Figure 4.7a presents the average prediction error for the Full MLP. As can be seen there, the prediction error is greater for edgewise loads than for flapwise loads (note that the fore-aft moment is governed by the edgewise loads) – this increase in error is likely due to the greater sensitivity of the lower-magnitude edgewise loads. It can also be seen that error was greatest at the tip of the turbine blade (where the deflections are greatest) and at the root (where the incoming turbulent wind is a larger component of the overall wind), which may explain this increase in sensitivity.

Figure 4.7b presents the average NRMSE of the Reduced MLP for each output. It can be seen that the trends described for the Full MLP are true here as well, in addition to the increase

in error across the board due to reduced number of inputs.

Figure 4.8 presents a scatter plot of the normalized Full MLP results, comparing the predicted and true output values normalized by the maximum positive true value. To prevent overplotting, 90% of the data was randomly removed from the results before generating the figure. Here, the outputs have been categorized into six broad groups based on whether it is the edgewise or flapwise response and which section of the blade it falls on – the Tip group consists of the outputs from nodes 1-6, the Middle group consists of the outputs from nodes 7-12, and the Root group consists of the outputs from nodes 13-18 and the hub. Consult Table 4.2 for the specific location of each of these nodes. The diagonal line on Figure 4.8 represents a perfect prediction, and it can be seen that the prediction of the Full MLP is generally quite good with an R^2 of 0.991. Further, the most noticeable outliers are from the edgewise responses at the tip and root and the flapwise responses at the root, which generally agree with the results seen in Figure 4.7.

Figure 4.9 shows the accuracy of the Full MLP from the perspective of a time history analysis. Both the wind loads predicted by the NN and the true wind loads supplied in the training data are compared in three examples which vary by blade node and mean wind speed. Note that the scales of the vertical axes differ for each time history. These example time histories were chosen arbitrarily across wind speeds and locations on the blade to show the consistency of the trends. It can be seen that irrespective of the blade node and wind speed, the time histories predicted by the Full MLP match the true values quite well, though it can be seen that the NN tends to slightly under-predict the loads at the peaks.

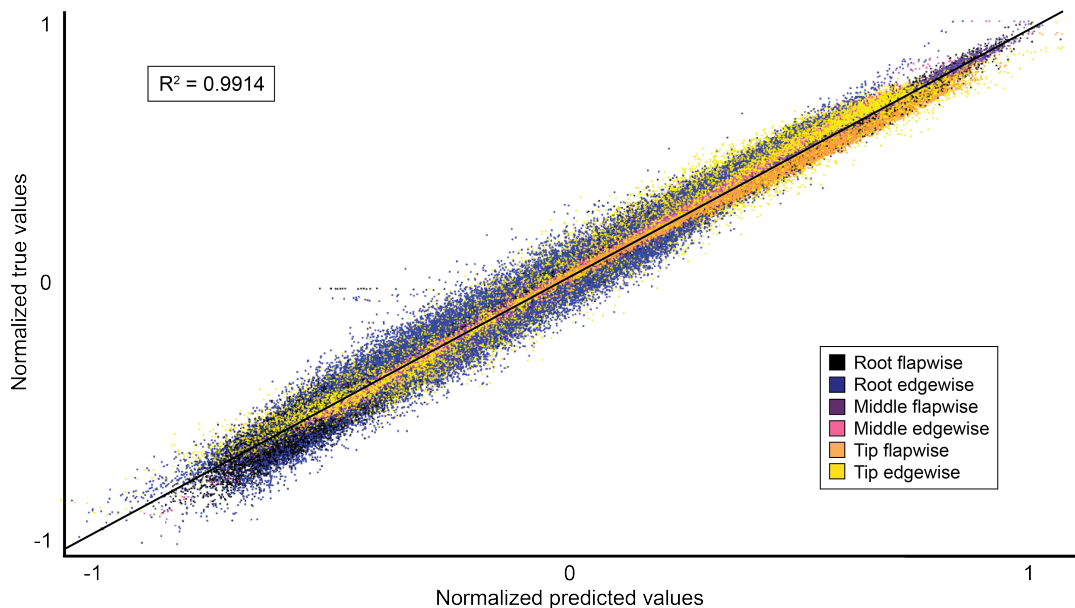


Figure 4.8: Scatter plot of 10% of the normalized Full MLP results, categorized by colour into six broad output groups

The time required to optimize, train and run both NNs was also extracted using the *tic* and *toc* functions in MATLAB. The results of this are shown in Table 4.6. Here, the run time refers to the time required to run the trained network for a single time step, i.e. how long it takes to predict a set of outputs from of a single set of inputs. As can be seen, the Reduced MLP

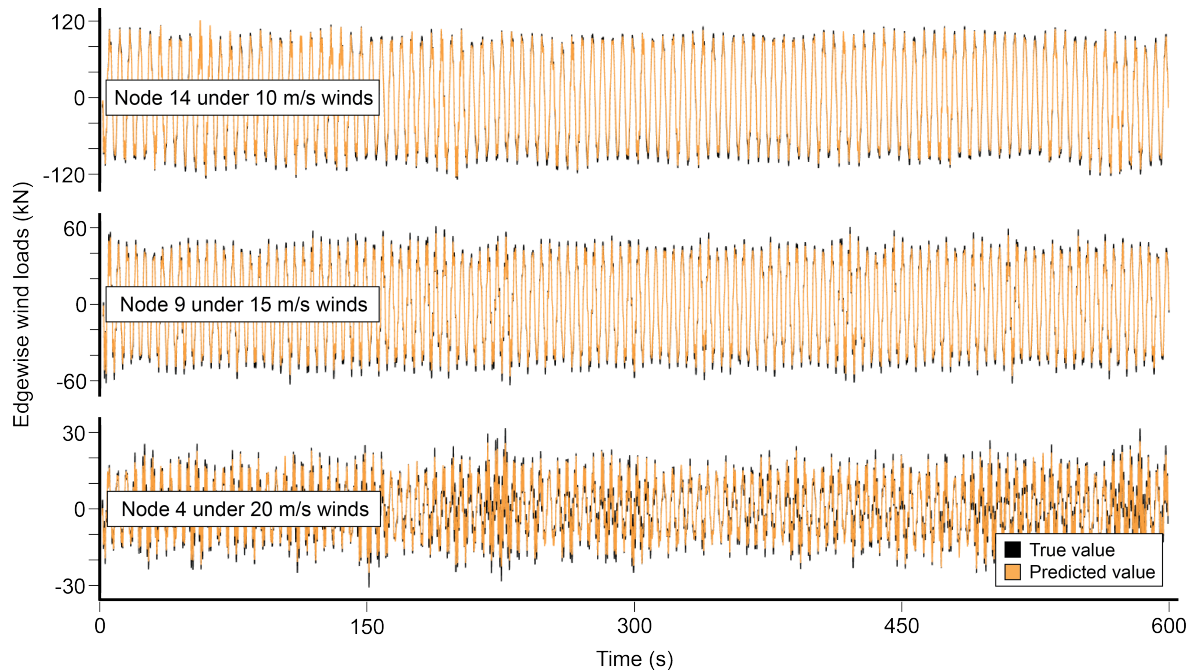


Figure 4.9: Comparison of arbitrary wind load time histories between outputs predicted by Full MLP and the true values supplied in the testing data. Each of the three examples was recorded from a different blade node and at a different mean wind speed

	Full MLP	Reduced MLP
Optimization time (h)	11.6	11.1
Training time (h)	21.8	10.4
Run time (ms)	2.78	2.91

Table 4.6: Optimization, training and run times for the Full and Reduced MLPs

is quicker to optimize and train than the Full MLP due to the reduced number of inputs and smaller layers, but the run times of the two networks are nearly identical. Both NNs take about 3 ms to predict the output of a single time step on the tested hardware, which is a feasible speed for incorporating these models into real-time testing.

4.5 Time-dependent aerodynamic model: LSTM

The previous section of this paper details the accuracy of a time-independent MLP-based surrogate aerodynamic blade model referred to as the Full MLP, which was able to achieve an average NRMSE of 1.1% between the predicted and true output values. This section explores possible accuracy improvements by employing a specialized type of NN designed for time history data. In this section an LSTM network is trained and compared to the Full MLP. For further comparison, a Pseudo-time-dependent MLP is additionally trained using structural velocity and acceleration data as inputs.

Input groups	Output groups
Geometry	
Wind Speed	Loads
Deflected Shape	
Total number of inputs per time step: 79 Total number of outputs per time step: 40	

Table 4.7: Input and output data groups for the LSTM, including the total number of inputs and outputs per time step. Table 4.3 details the contents of each data group

4.5.1 LSTM specifications

The **LSTM** network uses the same groups of data as the Full MLP as shown in Table 4.7: Geometry, Wind Speed and Deflected Shape are provided as inputs, and the Loads are obtained as outputs.

Figure 4.10 lists the vertical array of layers for the LSTM network, using the same colour scheme as the LSTM previously shown in Figure 4.3. Here, I refers to the number of inputs to the network; T refers to the number of time steps in a single time history; N refers to the number of time histories; O refers to the number of outputs; and the *Layer Sizes* refers the hyperparameters defined below. Note that the sequence data must be supplied as a vertical cell array of 2D matrices.

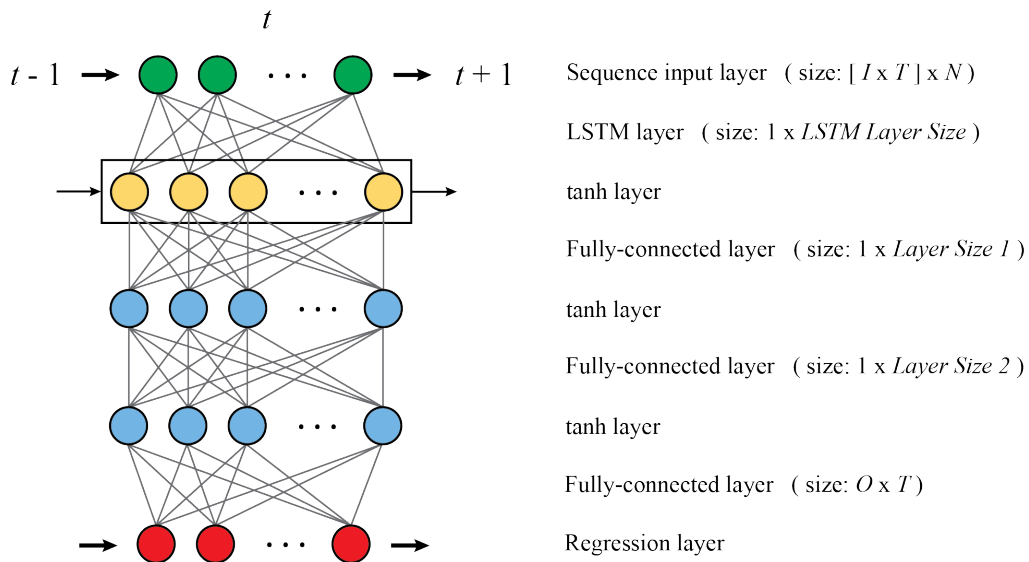


Figure 4.10: LSTM layer architecture and sizes

Table 4.8 details the hyperparameters optimized for the LSTM. Fixed conservative values were applied to the remaining hyperparameter as follows: one LSTM layer and two fully-connected layers were used, 100 epochs of training during optimization and 500 epochs during the final training were performed, a batch size of 100 was used, as well as an initial learning rate of 0.0001. Note that as the LSTM has higher memory requirements than a fully-connected layer, the maximum layer sizes of the network had to be reduced.

Hyperparameter	Momentum	LSTM Layer Size	Fully-Connected Layer Size 1	Fully-Connected Layer Size 2
Round 1 low	0.880	10	10	10
Round 1 high	0.999	120	400	400
Round 2 low	0.880	110	260	380
Round 2 high	0.920	120	300	420
Optimal	0.890	110	280	395

Table 4.8: Hyperparameter optimization results for the LSTM

4.5.2 LSTM results

The average NRMSE of the trained LSTM across all outputs and time histories is shown in Figure 4.11 to be 2.45%, which is greater than the prediction error of Full MLP network (1.11%). Figure 4.11 further details the NRMSE values of each output variable. It can be seen that compared to the Full MLP results, the edgewise response error of the LSTM is comparable, but that the error in predicting the flapwise loads is much higher. In fact, unlike either of the MLP networks, the LSTM seems to be equally effective at predicting the flapwise and the edgewise responses. Similar to previous networks, accuracy was the greatest near the midpoint of the blade and was lower at the hub and tip.

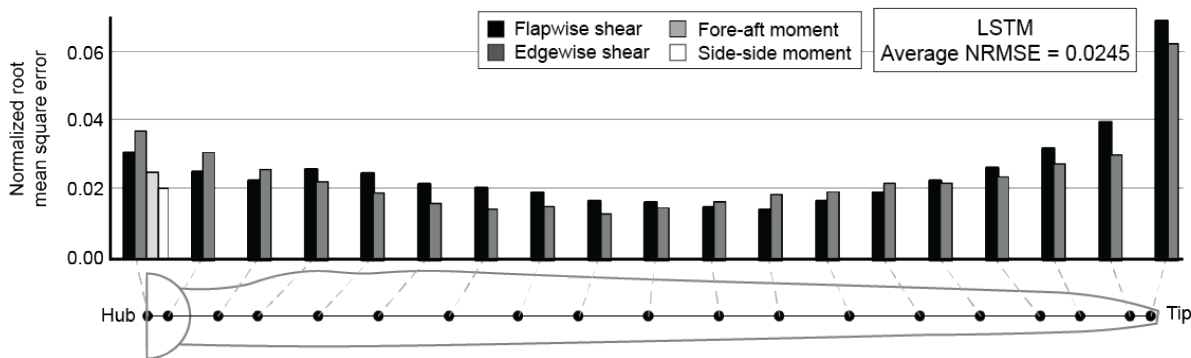


Figure 4.11: NRMSE of each response output averaged across all time histories for the trained LSTM

Table 4.9 presents the optimization, training and run times of the network. Here it can be seen that the optimization and training time exceeds those of the Full MLP due to the increased number of variables within the LSTM to optimize. This increased number of variables also significantly increases the run time of the LSTM which exceeds 10 ms, approximately three times the speed of the Full MLP. This longer run time would limit the applicability of this surrogate model in real-time testing scenarios.

These results suggest that the additional temporal considerations offered by the LSTM are not significant compared to the increased optimization difficulties by using this more complex type of neural network. It has been proven mathematically that MLPs can exceed the accuracy

Optimization time (h)	51.2
Training time (h)	22.6
Run time (ms)	11.71

Table 4.9: Optimization, training and run times for the LSTM

Input groups	Output groups
Geometry	
Wind Speed	
Deflected Shape	Loads
Structural Velocity/Acceleration	
Total number of inputs per time step: 151 Total number of outputs per time step: 40	

Table 4.10: Input and output data groups for the Pseudo-time-dependent MLP, including total number of inputs and outputs per time step. Table 4.3 details the contents of each data group

of RNNs depending on the type of time history data being modelled (Dreyfus, 2005). In this case, these results imply that the instantaneous inputs to the MLP (the Geometry, Wind Speed, and Deflected Shape data groups) are sufficient to achieve acceptable accuracy when determining the aerodynamic response, and that any improvement from considering the adjacent time steps does not justify the increase in network complexity. This hypothesis is investigated in the following subsection by training a Pseudo-time-dependent MLP network.

4.5.3 Pseudo-time-dependent MLP

To further investigate the impact of time dependency on the NN's ability to predict aerodynamic loads, the Full MLP from Section 4.4 is modified to include some time-dependent data by providing the instantaneous structural velocity and acceleration of the blade as inputs to the network. The addition of the velocity and acceleration offers the network an indirect view of the state of the system in the previous time step, which is conceptually similar to an LSTM's indirect view of previous time steps. This new network is referred to as the **Pseudo-time-dependent MLP**, the development of which is briefly summarized here.

Table 4.10 specifies the provided inputs and outputs to the network at each of the 300,000 time steps: to generate the Loads, the Geometry, Wind Speed, Deflected Shape and Structural Velocity/Acceleration wind groups are provided as inputs, resulting in nearly double the number of supplied inputs to the network compared to the Full MLP.

The list of layers used by the Pseudo-time-dependent MLP is the same as the previous MLPs, which is detailed in Section 4.4.2.

Table 4.11 details the hyperparameter optimization for this NN. The fixed hyperparameters include using two hidden layers, performing 100 epochs of training during optimization and 800 epochs during the final training, a batch size of 100, as well as an initial learning rate of 0.0001. Note that the number of final training epochs was increased to account for the additional inputs.

Hyperparameter	Momentum	Layer Size 1	Layer Size 2
Round 1 low	0.900	10	10
Round 1 high	0.999	700	700
Round 2 low	0.988	270	620
Round 2 high	0.995	350	700
Optimal	0.991	299	685

Table 4.11: Hyperparameter optimization results for the Pseudo-time-dependent MLP

The average NRMSE of the trained Pseudo-time-dependent MLP was found to be **1.34%** across all outputs and time histories. This error is greater than the Full MLP (1.11%) but less than the LSTM (2.45%). By providing the structural velocity and acceleration data to the MLP network, the accuracy was not notably increased and may in fact have been decreased; this implies that the instantaneous structural velocity and acceleration are not significant factors in determining the instantaneous aerodynamic loads, and thus the indirect effect of neighbouring time steps may likewise not affect it significantly. Therefore, the use of an LSTM network does not appear to notably improve accuracy while increasing the complexity and optimization difficulties of the model.

The trends of the full breakdown of the NRMSE and the training time for the Pseudo-time-dependent MLP do not vary notably from the Full MLP, thus they are excluded for brevity.

4.6 Time-dependent aerodynamic model: CNN

To further explore the time-dependency of the surrogate models, a CNN network was also investigated to conclusively confirm the findings of the previous section. This section details the development and analysis of a CNN aerodynamic model, as well as a Multi-time-step MLP network for further investigation.

4.6.1 CNN specifications

The CNN network used the same groups of data as the Full MLP: Geometry, Wind Speed and Deflected Shape were provided as inputs, and the Loads were obtained as outputs. However, in this case the CNN analyzes whole segments of time history as an image, thus to obtain each set of outputs, the w previous time steps were provided as inputs. This is summarized in Table 4.12, where the CNN is designed to consider w number of previous time steps.

Figure 4.12 lists the vertical array of layers for the CNN. Here, O refers to the number of outputs for the single predicted time step; I refers to the number of inputs to the network per time step; the hyperparameter w refers to the number of time steps provided as an input. Thus, each horizontal pixel in the input image represents a single time step, and each channel in the image represents one input. The *Layer Sizes* hyperparameters control the size of the fully-connected layers, and the *Filter Size* hyperparameter control the size of the convolutional filters and the resulting pooling layer size. These filters convolute every channel across *Filter*

Input groups	Output groups
Geometry	
Wind Speed	Loads
Deflected Shape	
Total number of inputs per time step: $79 \times w$	Total number of outputs per time step: 40

Table 4.12: Input and output data groups for the CNN, including the total number of inputs and outputs per time step. Table 4.3 details the contents of each data group

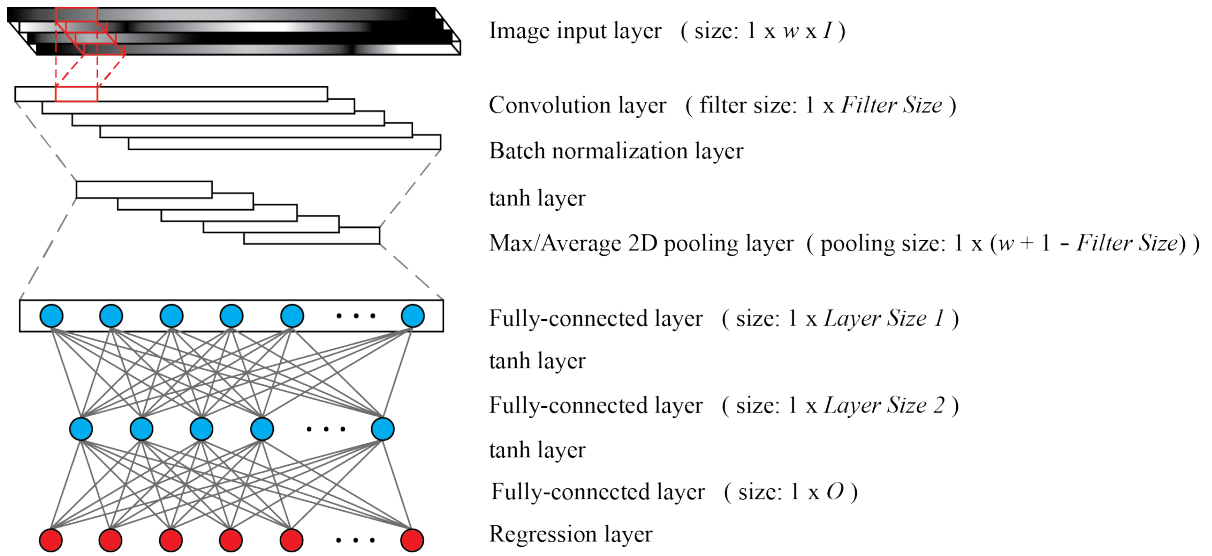


Figure 4.12: CNN layer architecture and sizes

Size time steps, moving across the time history one pixel at a time. The pooling layer then pools the entire convolution row, using either max or average pooling depending on the *Max or Average Pooling* hyperparameter, resulting in a number of condensed nodes equal to the *Number of Filters* hyperparameter with a wide view of the input data.

Table 4.13 details the hyperparameters optimized for the CNN. Fixed conservative values were applied to the remaining hyperparameters as follow: one round of convolution-activation-pooling was performed followed by two fully-connected layers, 50 epochs of training during optimization and 100 epochs during the final training were performed, a batch size of 100 was used, as well as an initial learning rate of 0.0001. Note that due to increased size of this network compared to others, less epochs for optimization and training were used due to greatly increased time requirements.

4.6.2 CNN results

The average NRMSE of the trained CNN was found to be 0.66% across all outputs and time histories as shown in Figure 4.13. This error is notably lower than the Full MLP (1.11%).

Hyperparameter	Momentum	Convolution Filter Size	Number of Filters	Max or Average Pooling	Number of Time Steps (w)	Fully-Connected Layer Size 1	Fully-Connected Layer Size 2
Round 1 low	0.900	1	1	0 (Max)	5	10	10
Round 1 high	0.999	5	10	1 (Avg.)	10	500	500
Round 2 low	0.990	3	1	0 (Max)	7	150	350
Round 2 high	0.996	5	5	1 (Avg.)	9	250	450
Optimal	0.993	3	1	0 (Max)	8	161	432

Table 4.13: Hyperparameter optimization results for the CNN

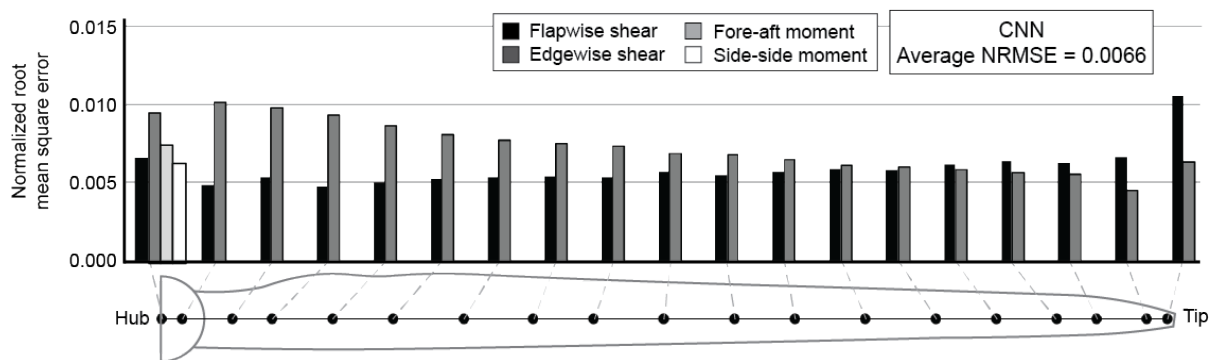


Figure 4.13: NRMSE of each response output averaged across all time histories for the trained CNN

These results suggest that by providing the network with a number of previous time steps (8, in this case), the error between the predicted and true outputs can be nearly halved.

Figure 4.13 also breaks down the NRMSE on a per-output basis. Studying this, it can be seen that like the LSTM and unlike the MLP networks, there is not a significant disparity in error between the prediction of the flapwise and edgewise loads; it struggles the most at predicting the flapwise loads at the tip and the edgewise loads at the root.

Table 4.14 presents the optimization, training and run times of the network. Here it can be seen that both the optimization and the training time far exceeds the previous networks, due to the increased computations involved with the convolution process as well as the increased number of hyperparameters. This represents an inconvenience for researchers, however since training occurs ahead of time, it is an acceptable limitation. The run time of the CNN is only slightly longer than the Full MLP, meaning that the model is fast enough to be used in real-time testing environments.

Optimization time (h)	98.8
Training time (h)	85.1
Run time (ms)	3.16

Table 4.14: Optimization, training and run times for the CNN

Hyperparameter	Momentum	Number of Time Steps	Fully-Connected Layer Size 1	Fully-Connected Layer Size 2
Round 1 low	0.900	1	10	10
Round 1 high	0.999	10	500	500
Round 2 low	0.970	5	360	210
Round 2 high	0.980	7	440	290
Optimal	0.979	6	384	246

Table 4.15: Hyperparameter optimization results for the Multi-time-step MLP

These results suggest that by providing additional previous time steps to the network, the accuracy can be increased. Further, the convolution and pooling process to limit the size of the fully-connected layers while still providing access to the full set of time steps and inputs seems to aid in this process. While Section 4.5 seemed to indicate the temporal sensitivity of the aerodynamic calculations was low, it appears that the convolution process allows for it to be considered in a way that does not significantly increase the complexity of the network. However, to ensure that the additional accuracy does not solely stem from the additional amount of input data, a Multi-time-step MLP network was also developed and tested.

4.6.3 Multi-time-step MLP

To investigate whether the convolution process of the CNN is significant for the network's accuracy, the Full MLP from Section 4.4 is modified to receive the input data from w previous time steps for predicting the aerodynamic loads on the turbine blade. This new network is referred to as the **Multi-time-step MLP**, the development of which is briefly summarized here.

The Multi-time-step MLP uses the same combinations of inputs and outputs as the CNN network, as summarized in Table 4.12. The list of layers is the same as the previous MLPs, which is presented in Section 4.4.2, except that the size of the input layer is $1 \times (I * w)$. Thus, this network is essentially the same as the CNN with the convolution and pooling layers removed.

Table 4.15 details the hyperparameter optimization for this NN. The fixed hyperparameters include using two fully-connected layers, performing 50 epochs of training during optimization and 200 epochs during the final training, as well as using a batch size of 100 and an initial learning rate of 0.0001 – identical initial values to the CNN, but it was found that extra epochs compared to the CNN were required during the final testing due to the larger number of connections in the first fully-connected layer.

The average NRMSE of the trained Multi-time-step MLP was found to be **1.60%** across all outputs and time histories. This error is greater than both the CNN (0.66%) and the Full MLP (1.11%). This shows that the size reduction provided by the convolution process of the CNN results is key to accurately training the network, allowing the large increase in inputs to the network to be more effectively considered given the same amount of training.

The trends of the full breakdown of the NRMSE and the training time for the Multi-time-step MLP do not vary notably from the Full MLP, thus they are excluded for brevity.

Neural network	Average NRMSE	Input Groups	Output Groups
Full MLP	1.11%	G, W, D	L
Reduced MLP	5.78%	G, W	L
LSTM	2.45%	G, W, D	L
Pseudo-time-dependent MLP	1.34%	G, W, D, S	L
CNN	0.66%	G, W, D	L
Multi-time-step MLP	1.60%	G, W, D	L

Table 4.16: Summary of the six NNs, their average NRMSE across all outputs and time steps, and their input and output groups. Groups are described in Table 4.3 and include (G)eometry, (W)ind Speed, (D)eflected Shape, (S)tructural Velocity/Acceleration, and (L)oads

4.7 Conclusions

Six neural networks (NNs) were investigated as surrogate aerodynamic blade models for a 5 MW wind turbine blade with the advantage of quick computational times. Previous NNs applied to wind turbine research have achieved errors of 1-5% – errors of $\leq 1\%$ have typically only been achieved by networks with less than five inputs and outputs.

The training and validation data for these neural networks were generated in OpenFAST, where an operational 5 MW wind turbine was subjected to mean hub wind speeds ranging from 3 to 25 m/s. Data of this quality was deemed sufficient for the optimization performed in this study.

NN types considered in this study include multilayer perceptron (MLP), long short-term memory (LSTM), and convolutional neural networks (CNN). After hyperparameter optimization and final training, the accuracy of the network was determined by calculating the average normalized root mean square error (NRMSE) between the predicted outputs of the network to the supplied testing data outputs. The inputs, outputs, and total average NRMSE of each of the six trained networks is summarized in Table 4.16. The run time of most networks was found to be quite low at ≤ 5 ms.

The Full MLP represents the baseline time-independent network of the study. It was supplied with the geometry, wind speed and deflected shape information to predict the aerodynamic loads. The Full MLP’s average NRMSE of 1.11% represents a highly accurate network that surpasses the accuracy of previous NNs of this size used in wind turbine studies.

The Reduced MLP received a reduced number of inputs relative to the Full MLP for a possible improvement in usability, but the resulting decrease in accuracy means that this network is not recommended.

The LSTM and Pseudo-time-independent networks were unable to match the accuracy of the Full MLP despite being better equipped for time-sensitive data. This suggests that the time-sensitivity of the data was not sufficiently high to justify the increase in network size required by these models.

The CNN provided a segment of neighbouring time steps as an image to predict the output of a single time step. This represents an extremely large increase in the amount of input data, but the convolution-activation-pooling process was used to reduce the size of the network while keeping the most important information. The average NRMSE of the CNN was found to

be only 0.66%, which was the most accurate surrogate model developed in this study. The necessity of the convolution process is highlighted by the reduced accuracy of the Multi-time-step MLP which received the same inputs as the CNN but did not convolute. The size reduction of the convolution process explains why the CNN surpassed the Full MLP while the rest of the time-dependent models failed to do so.

Overall, the CNN was found to be most accurate, while the Full MLP also achieved a high accuracy using a simpler network. Simplicity of development is important for the neural networks to be extended into structural engineering, and the Full MLP surpasses the CNN in this regard. Thus, either of these network architectures could be recommended, depending on the requirements of the researchers.

This research was limited in several ways; even though supercomputing was employed, the hardware still had an upper limit on the number of variables that could be stored in memory at once. This required that the generated data set be reduced to 300,000 time steps per time history, meaning that these networks could potentially have been further optimized with more training data. Furthermore, neural network-based surrogate models are inherently limited to only being valid within the bound of the training data. Thus the developed surrogate models cannot be used to accurately study a parked turbine, a turbine under wind loads exceeding the cut-out speed, or a different size of turbine, etc. Finally, while reducing noise in input data has been previously shown to improve NN performance (Cornford et al., 2000), this technique was not explored in this study.

Ultimately, the CNN and Full MLP surrogate wind turbine blade models offer promising options for time sensitive applications, either when the amount of testing is large such as during parametric optimization, or when high computation times are required such as during real-time hybrid simulation. Future research will look to train a surrogate model on higher-accuracy aerodynamic data to further explore the advantages these systems offer.

Chapter 5

A comparison of passive and semi-active tuned mass damper systems for wind turbines

Summary

Robust semi-active vibration control of wind turbines using tuned mass dampers (TMDs) is a promising technique. This study investigates a 1.5 megawatt wind turbine controlled by eight different types of tuned mass damper systems of equal mass: a passive TMD, a semi-active damping-component TMD, a semi-active stiffness-component TMD, a semi-active semi-active damping-and-stiffness-component TMD, as well as these four damper systems paired with an additional smaller passive TMD near the mid-point of the tower. The mechanism and controller for each of these TMD systems is explained, such as employing magnetorheological dampers for the damping-component control cases. The turbine is modelled as a lumped-mass 3D finite element model. The uncontrolled and controlled turbines are subjected to loading and operational case combinations including service wind loads on operational turbines, seismic loading with service wind on operational turbines, and high-intensity storm wind loads on parked turbines. The displacement and acceleration responses of the tower at the first and second mode maxima were used as the performance indicators. Ultimately, it was found that while all the semi-active TMD systems outperformed the passive systems, it was the semi-active damping-and-stiffness-component control system that was found to be the most effective overall – capable of controlling vibrations about as effectively with only half the mass as a passive TMD. It was also shown that by reducing the mass of the TMD and adding a second smaller TMD below, the vibrations near the mid-point could be greatly reduced at the cost of slightly increased vibrations at the tower top.

This chapter is an updated reproduction of the author’s published paper “Comparison of semi-active and passive tuned mass damper systems for vibration control of a wind turbine” (Lalonde et al., 2020b).

5.1 Introduction

With the ongoing threat of climate change, renewable energy technologies such as wind turbines continue to see increased implementation worldwide. As of 2019, the global installed wind energy capacity was 592 GW (GWEC, 2017) and this number continues to grow. The lifespan of wind turbines is traditionally governed by fatigue caused by dynamic wind loading due to the flexibility of the structure. However, there has also been increased erection of wind turbines in areas with high wind or seismic risks such as parts of the USA, Japan and China, which risks structural failures (Mardfekri and Gardoni, 2015; Diaz and Suarez, 2014; Chou and Tu, 2011). Ensuring long service lives and the safety of wind turbines is critical to keeping their costs low and encouraging further implementation of this technology. For a more thorough introduction to this topic, the reviews of modern vibration control methods for wind turbines by Rezaee and Aly (2016) and Rahman et al. (2015) are recommended.

A very common structural vibration control method is the tuned mass damper (TMD). The response of the primary structure can be reduced by adding a secondary mass within the structure that can displace relative to the main structure. The TMD is typically connected to the main structure by spring, damper, and/or pendulum systems. The natural frequency of the TMD is tuned by changing the properties of this connection to improve the vibration control effect. TMDs controls include passive, active, and semi-active systems, as shown in Figure 5.1. Passive tuned mass dampers (PTMDs) are tuned to a single frequency, typically the natural frequency of the primary structure. PTMDs have been applied to wind turbines in many studies (Sun and Jahangiri, 2018; He et al., 2017; Lackner and Rotea, 2010; Murtagh et al., 2008; Argyriadis et al., 2004), which have generally concluded that while PTMDs are simple and effective at controlling vibration at their tuned frequency, they tend to have low robustness and lose effectiveness for wideband loading processes. In particular, seismic loading can excite the higher modes of a structure that often are not a concern under wind loading (Zhao et al., 2019b). If the natural frequency of the primary structure changes over time due to damage or through soil-structure effects, the PTMD will lose effectiveness in turn. Multiple tuned mass dampers (MTMDs) can be used to offset the low robustness of individual PTMDs by tuning each to a different frequency. Additionally, it is often more practical to install multiple small masses within a structure than a single large one, and MTMDs have improved redundancy compared to single TMDs. When applied to wind turbines, passive MTMDs have been shown (Hussan et al., 2018, 2017; Zuo et al., 2017) to improve the response of the structure when subjected to seismic loading which excites the higher modes of the structure. However, this may come at the cost of reduced vibration control under service wind loading compared to a single PTMD.

Active tuned mass dampers (ATMDs) include actuators that apply an active force to the TMD mass to improve the vibration control effect. There have been multiple studies of ATMDs used in wind turbines (Brodersen et al., 2017; Stewart, 2012; Lackner and Rotea, 2011) which all have shown a strong capability for vibration control, however ATMDs are often limited by the large electrical cost of running the actuators as well as lower reliability compared to PTMDs.

Semi-active tuned mass dampers (STMDs) combine the benefits of passive and active systems by modifying their stiffness or damping values in real-time to improve their vibration control capabilities. In traditional structures, STMDs have been shown to be about as effective

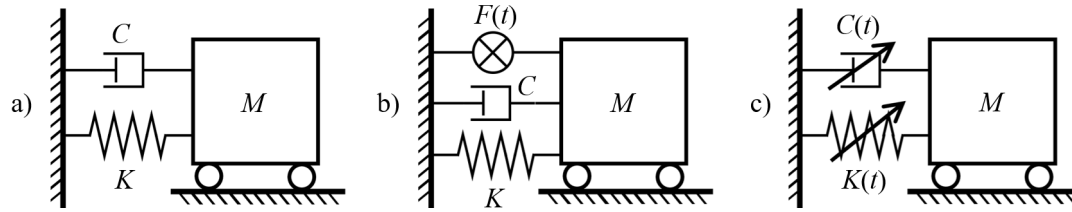


Figure 5.1: Simplified models of TMDs connected to main structure. a) PTMD b) ATMD c) STMD

as ATMDs, but with better reliability and reduced electrical costs (Kaveh et al., 2015; Sun et al., 2014; Eason et al., 2013; Chung et al., 2013; Esteki et al., 2011; Owji et al., 2011; Chey et al., 2009; Nagarajaiah and Sonmez, 2007; Yang et al., 2002; Pinkaew and Fujino, 2001; Ricciardelli et al., 2000; Hrovat et al., 1983). There are two main methods of controlling an STMD: by modifying the TMD stiffness in real-time to change the tuning, increasing its robustness; or modifying the damping in real-time to increase the amount of energy absorbed by the TMD. While both techniques have been shown to improve effectiveness compared to PTMDs, some studies (Nagarajaiah, 2009) recommend employing the variable stiffness dampers due to the improved robustness and the ability to tune the damper to the structural loading rather the natural frequency of the structure. Some studies also employ both variable stiffness and damping simultaneously to control their STMDs (Sun and Nagarajaiah, 2013). There are several instances of single and multiple STMDs with various control methods being employed in wind turbines, all of which conclude that the improved robustness of the STMDs makes them more effective than traditional PTMDs - these papers are summarized in Table 5.1.

These previous studies make it clear that STMDs can be effective at controlling vibrations in wind turbines. However, Table 5.1 shows there exists limitations in the previous research: they frequently test a smaller number of load cases using models that simplify the tower as a single beam, which does not give the second mode response full consideration. With a few exceptions (Park et al., 2019), the effectiveness of the proposed STMD is only compared to an equivalent PTMD, not to other STMDs. Frequently these studies do not approach the design of the TMDs from a pragmatic perspective: the physical systems used to control the stiffness or damping are not modelled but merely assumed, which may result in the semi-active behaviour of these models being unrealistically precise. Additionally, many studies place the TMD in the wind turbine nacelle without acknowledging that there is limited space there for such a system in modern wind turbine designs. A practical TMD design must allow workers to climb the tower to maintain the turbine, and if placed within the nacelle it must not interfere with the generator and other internal systems.

This paper aims to build upon the previous literature by studying the effectiveness of various passive and semi-active single and multiple TMD systems with a focus on addressing the gaps in previous research by comparing a wide array of equivalent, fully-detailed TMDs. It compares 3D turbine models equipped with eight equivalent-mass TMD systems – a passive TMD, an STMD that controls the TMD damping, an STMD that controls the TMD stiffness, an STMD that controls both damping and stiffness, as well as MTMD configurations of these four – to an uncontrolled turbine under service wind loads, high intensity wind loads, and seismic loads with service wind. Firstly, the design of the wind turbine, the loading cases, and the TMD

Paper	Turbine structure	STMD location(s)	STMD control method and mechanism*	Does model capture 2nd mode?	2D or 3D model?	Load types	Performance indices
Huang et al. (2010)	Floating offshore turbine	Nacelle, blades, platform	Stiffness control (no mechanism)	No	2D	Wind, wave	Nacelle displacement
Arrigan et al. (2011)	Onshore turbine	Nacelle, blades	Stiffness control (no mechanism)	No	2D	Wind	Nacelle and blade displacement
Martynowicz (2015)	Onshore turbine	Nacelle	Damping control (numerical and experimental MR damper)	Yes	2D loading	Harmonic, chirp	Tower top and midpoint displacement
Dinh et al. (2016)	Floating offshore turbine	Nacelle, blades, buoy	Stiffness control (no mechanism)	No	2D	Wind, wave, mooring	Nacelle, blade and buoy displacement
Park et al. (2016)	Fixed offshore turbine	Tower top	Damping control (no mechanism)	Yes	3D	Wind, wave	Fore-aft bending, base moment
Tsouroukdissian et al. (2016)	Fixed and floating offshore turbine	Nacelle	Damping control (no mechanism)	Yes	3D	Wind, wave	Nacelle acceleration
Martynowicz (2016)	Onshore turbine	Nacelle	Damping control (experimental MR damper)	Yes (experimental)	2D loading	Harmonic, impulse	Tower top and midpoint displacement
Martynowicz (2017)	Onshore turbine	Nacelle	Damping control (experimental MR damper)	Yes (experimental)	2D loading	Harmonic	Tower top and midpoint displacement
Sun (2017)	Fixed offshore turbine	Nacelle	Stiffness and damping control (no mechanism)	No	3D	Wind, wave	Nacelle displacement, base rotation
Hemmati and Oterkus (2018)	Fixed offshore turbine	Nacelle	Stiffness and damping control (no mechanism)	Yes	2D	Wind, wave, seismic	Nacelle displacement, base shear and moment
Park et al. (2019)	Fixed and floating offshore turbine	Tower top	Damping control (numerical MR damper)	Yes	3D	Wind, wave	Nacelle displacement, base moment

*Here the term “mechanism” refers to whether the paper presents a mechanical explanation of how the stiffness and / or damping of the TD is varied, such as modelling a damping-component controller using a magnetorheological damper.

Table 5.1: Summary of previous studies of wind turbines with semi-active tuned mass dampers

models is presented. The effectiveness of the various TMD systems is compared by examining the response of the turbines at the maxima of the first two modes. Finally, a pragmatic TMD for modern wind turbines is proposed.

5.2 Methodology

5.2.1 Turbine model

The 3D turbine model employed in this testing was a lumped-mass, 32-degree of freedom (DOF) finite element (FE) model built in MATLAB. The tower was built of standard beam

elements and had a fixed base, which allowed for multiple modes to be analyzed in both the fore-aft (FA) and side-to-side (SS) directions. The turbine's nacelle and blades were modelled as a lumped mass at the top of the tower with an applied moment to account for the normal eccentricity of the center of mass. This simplification has been previously applied in numerous studies (Zhang et al., 2019; Hussan et al., 2017; Dai et al., 2017a; Brodersen et al., 2016; Dai et al., 2015a; Prowell et al., 2009). MATLAB was used for interconnectivity with the surrogate models detailed later in Chapters 4 and 7.

This FEM model was made of eight prismatic beam elements in the tower above a fixed foundation, as shown in Figure 5.2. Each beam element had eight DOF, which correspond to the lateral displacements and rotations at the ends of the beams. Each lumped-mass node was thus capable of displacement and rotation in both the FA and SS directions (which correspond to x_{FA} , θ_{FA} , x_{SS} , and θ_{SS} respectively in Figure 5.2c). Axial deformation and torsion were not modelled in this study. The lumped mass consisting of the nacelle and blades is described in Figure 5.2b: c_a is the modelled aerodynamic damping and M_s is the static moment caused by the mass eccentricity of the nacelle – the values of the parameters are quantified below.

Equations 5.1-5.2 were used to calculate the stiffness and distributed mass element matrices for the beams in this model (Reddy, 1993):

$$[K_e] = \frac{2EI_e}{L_e^3} \begin{bmatrix} 6 & & & & & & & & \\ -3L_e & 2L_e^2 & & & & & & & \\ 0 & 0 & 6 & & & & & & \\ 0 & 0 & -3L_e & 2L_e^2 & & & & & \\ -6 & 3L_e & 0 & 0 & 6 & & & & \\ -3L_e & L_e^2 & 0 & 0 & 3L_e & 2L_e^2 & & & \\ 0 & 0 & -6 & 3L_e & 0 & 0 & 6 & & \\ 0 & 0 & -3L_e & L_e^2 & 0 & 0 & 3L_e & 2L_e^2 & \end{bmatrix} \quad (5.1)$$

$$[M_e] = \frac{LD_e \cdot L_e}{210} \begin{bmatrix} 78 & & & & & & & & \\ 11L_e & 2L_e^2 & & & & & & & \\ 0 & 0 & 78 & & & & & & \\ 0 & 0 & 11L_e & 2L_e^2 & & & & & \\ 27 & 7.5L_e & 0 & 0 & 78 & & & & \\ -7.5L_e & -4.5L_e^2 & 0 & 0 & -11L_e & 2L_e^2 & & & \\ 0 & 0 & 27 & 7.5L_e & 0 & 0 & 78 & & \\ 0 & 0 & -7.5L_e & -1.5L_e^2 & 0 & 0 & -11L_e & 2L_e^2 & \end{bmatrix} \quad (5.2)$$

Here, $[K_e]$ and $[M_e]$ are the symmetrical stiffness and mass element matrices respectively, L_e is the element length, EI_e is the flexural rigidity of the element, and LD_e is its linear density. Note that since turbine towers are built from circular hollow steel sections, the simplified mass, stiffness and modal properties are the same in both the FA and SS directions.

Previous research has shown that when the aerodynamic damping provided by the spinning rotor is considered, operational turbines have a damping ratio of approximately 5% in the FA direction and a value of 0.5% (Katsanos et al., 2016) to 1% (Mardfekri and Gardoni, 2015) in the SS direction – the latter of these values was used in this study. This damping behaviour was captured using Rayleigh damping based on the first two FA natural frequencies and a 1% structural damping ratio. The additional aerodynamic damping was simulated using a linear

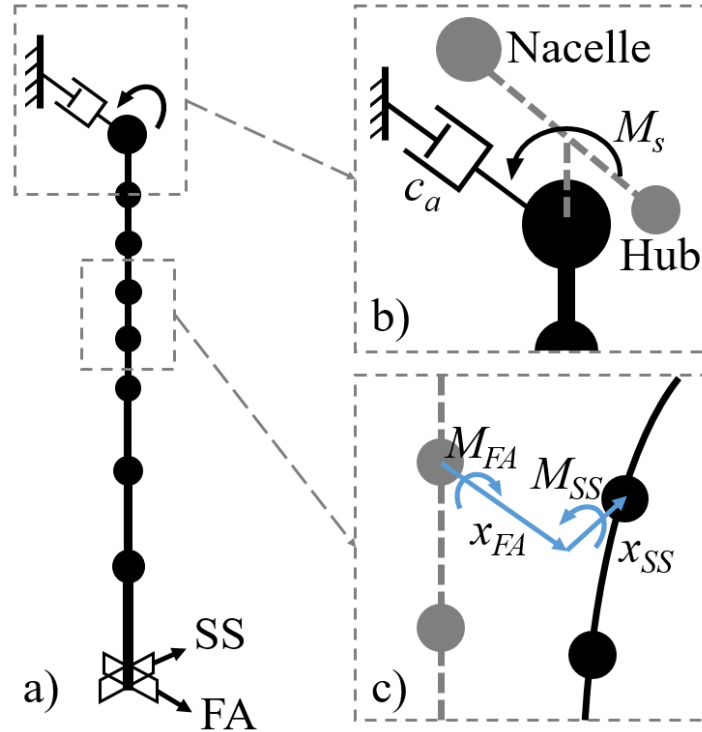


Figure 5.2: a) The 32-DOF FEM turbine model. b) Eccentricity moment and aerodynamic damping of the top lumped-mass node. c) DOFs of a single lumped-mass node

damper attached to the top of the turbine and oriented in the FA direction, a technique suggested by Valamanesh and Myers (2014). The damping coefficient for this damper was calculated using Equation 5.3 based on the assumption that the first mode will govern the response at the top of the tower:

$$c_a = 2 \cdot m_{m1} \cdot f_n \cdot \xi_a \quad (5.3)$$

Here, m_{m1} is the first modal mass; ω_1 is the first natural frequency in the FA direction; and ξ_a is the aerodynamic damping ratio of 4% (this plus the 1% structural damping reached the target value of 5%). When the turbine was in the parked condition, c_a was set to zero.

The turbine tested in this study was a standard 1.5 MW three-blade horizontal axis wind turbine with a hub height of 65 m, which has previously been tested a number of times (Zhang et al., 2019; Zhao et al., 2019a,b; Dai et al., 2017a; Sadowski et al., 2017). Table 5.2 lists the tower properties used in Equations 5.1-5.2. As shown in Figure 5.2, the hub was modelled as a 16.83 tonne lumped mass 1.5 m above and 2.5 m in front of the center of the tower top, and the nacelle was modelled as a 60 t mass 1.5 m above and 1 m behind. Sadowski et al. (2017) found the first and second modal mass contributions for this turbine to be 64% and 18% respectively, thus $m_{m1} \approx 100$ t and $m_{m2} \approx 28$ t. The damping coefficient of the aerodynamic hub damper was calculated using Equation 5.3 to be 28,000 Ns/m. Modal analysis identified the maximum of the first mode shape at the top of the tower (at a height of 61.7 m) and the maximum of the second mode shape at the fifth node from the top (at a height of 39.8 m), which were chosen as the installation locations for the upper and lower TMDs [see Figure 5.3].

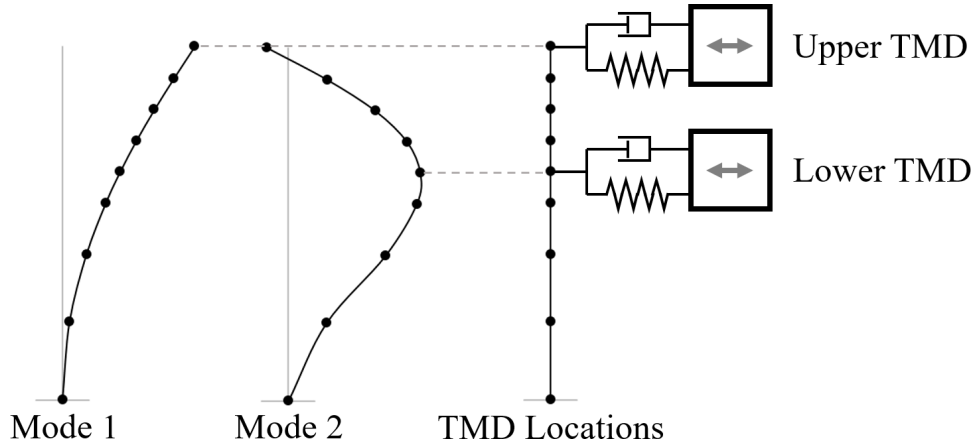


Figure 5.3: Turbine tower mode shapes and the resulting TMD locations

Tower Section	Element Length (m) L_e	Flexural Rigidity (Nm ²) EI_e	Linear Density (kg/m) LD_e	Tower Section	Element Length (m) L_e	Flexural Rigidity (Nm ²) EI_e	Linear Density (kg/m) LD_e
1 (Top)	5.60	2.22E+10	739	5	5.40	5.15E+10	1043
2	5.46	2.73E+10	792	6	9.04	6.68E+10	1248
3	5.44	3.31E+10	844	7	11.76	8.19E+10	1529
4	5.42	3.96E+10	896	8 (Bottom)	13.61	1.12E+11	2084

Table 5.2: Turbine tower properties used in FEM model

	1st FA freq. (Hz)	2nd FA freq. (Hz)	1st SS freq. (Hz)	2nd SS freq. (Hz)
MATLAB FEM model	0.558	5.03	0.558	5.03
SAP 2000 model	0.557	4.93	0.557	4.93
% difference	0.06%	1.9%	0.06%	1.9%
OpenFAST model	0.511	4.82	0.511	4.82
% difference	8.3%	4.1%	8.3%	4.1%
Field measurements	0.490	3.85	0.481	4.07
% difference	12.1%	23.5%	13.8%	18.9%

Table 5.3: Natural frequency comparison between SAP 2000 model, OpenFAST model, and field measurements to this study’s FEM model

Load case	Loading direction	Additional loading	Operation state	Number of time histories
SW - 10 m/s	FA	—	Operational	10
SW - 15 m/s	FA	—	Operational	10
SW - 20 m/s	FA	—	Operational	10
SW - 25 m/s	FA	—	Operational	10
HIW - 35 m/s	SS	—	Parked	10
HIW - 40 m/s	SS	—	Parked	10
HIW - 45 m/s	SS	—	Parked	10
HIW - 50 m/s	SS	—	Parked	10
HIW - 55 m/s	SS	—	Parked	10
HIW - 60 m/s	SS	—	Parked	10
EQ 1	FA, SS, FA+SS	SW - 10 m/s	Operational	3
EQ 2	FA, SS, FA+SS	SW - 10 m/s	Operational	3
EQ 3	FA, SS, FA+SS	SW - 10 m/s	Operational	3
EQ 4	FA, SS, FA+SS	SW - 10 m/s	Operational	3
EQ 5	FA, SS, FA+SS	SW - 10 m/s	Operational	3
EQ 6	FA, SS, FA+SS	SW - 10 m/s	Operational	3
EQ 7	FA, SS, FA+SS	SW - 10 m/s	Operational	3

Table 5.4: List of applied load cases

This FEM model of the uncontrolled turbine was validated by comparing the first two natural frequencies in the FA and SS directions to an identical 8-element model built in SAP 2000, as well as a more detailed FEM model in the open-source turbine modelling software OpenFAST (Jonkman et al., 2021). It was also verified against field measurements of a real specimen of the test turbine near Shanghai (Dai et al., 2017b). Table 5.3 shows that the first and second modes in the FA and SS directions were almost identical between the MATLAB and SAP 2000 model, with an increased error compared to the OpenFAST and field measurements due to simplifications present in the 8-element turbine model.

5.2.2 Applied load cases

Three general loading conditions were considered in this study: multi-directional earthquake (EQ) loading combined with service wind loads to an operational wind turbine, high-intensity wind (HIW) loads applied in the SS direction to a parked wind turbine (which has been shown to be the worst extreme wind loading case (Zhang et al., 2019; Wang et al., 2013), and service wind (SW) loads applied in the FA direction to an operational turbine. Table 5.4 summarizes the time histories used in this testing, with full explanations in the following sections. The speeds following the wind load cases refer to the mean hub wind speed. Test durations were kept consistent across all tests, thus an overall duration of 85 s was used. In seismic cases the ground motion began 15 s into the test. A total of 40 SW, 60 HIW, and 21 EQ load cases were applied to the uncontrolled turbine as well as the eight TMD types for a total of 1089 trials.

5.2.2.1 Wind loading

Both the 50-year HIW loads and the SW loads were generated using the following technique, which is described in greater detail in Zhao et al. (2019b). First, wind fields were generated in the software TurbSim (Kelley and Jonkman, 2012) using random seeds, with parameters selected according to IEC guidelines (IEC, 2005) using the normal wind and extreme turbulent wind models. Table 5.5 summarizes the parameters and equations, which are based on the mean hub wind speeds (V_{hub}) listed in Table 5.4. TurbSim then uses the Sandia method (Veers, 1988) to generate the wind fields. Secondly, these newly generated wind fields were applied to the uncontrolled test turbine in OpenFAST (Jonkman et al., 2021), which uses the BEM theory (Glauert, 1935) to calculate the resulting blade loads on the tower. The time history of these blade loads were applied to the FEM model of the controlled wind turbine during testing. This TurbSim-OpenFAST-FEM process to model blade loads has been applied in several other studies (Mo et al., 2017; Asareh et al., 2016; Mardfekri and Gardoni, 2015). Tower and nacelle loads were applied as concentrated loads at the appropriate DOF, which were likewise generated using OpenFAST.

Parameter	Normal wind model (SW)	Extreme turbulent wind model (HIW)
Turbulence intensity	$I_{ref} = 0.16$	(same)
Turbulence standard deviation	$\sigma_1 = I_{ref}(0.75V_{hub} + 5.6)$	$\sigma_1 = 0.11V_{hub}$
Wind profile (z is height (m) and $z_{hub} = 65$ m)	$V(z) = V_{hub}(\frac{z}{z_{hub}})^{0.2}$	$V(z) = V_{hub}(\frac{z}{z_{hub}})^{0.11}$
Equivalent 10 m design wind speed	$V_{10} = 0.688V_{hub}$	$V_{10} = 0.841V_{hub}$
Spectrum	$S(f) = 0.05\sigma_1^2(\frac{42}{V_{hub}})^{-\frac{2}{3}}f^{-\frac{5}{3}}$	(same)

Table 5.5: Summary of IEC wind field parameters for the 1.5 MW wind turbine

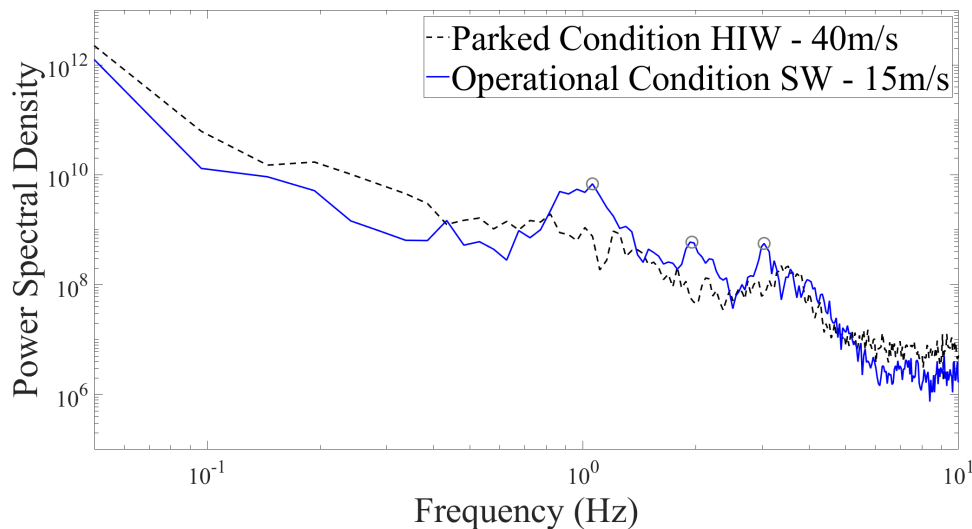


Figure 5.4: Power spectrum density of blade load time histories under SW and HIW conditions

The SW conditions ranged from a mean hub wind velocity of 10 m/s to 25 m/s, and the HIW conditions ranged from 35 m/s to 60 m/s. Since this testing was limited to elastic turbine response, the maximum design wind speed of 60 m/s for the test turbine was not exceeded.

Figure 5.4 shows the power spectrum density of the concentrated FA blade loads calculated in OpenFAST under representative operational and parked conditions. The peaks at 1, 2 and 3 Hz in the operational condition due to rotation of the blades are highlighted, which disappear in the parked condition.

5.2.2.2 Seismic loading

As suggested by the Chinese seismic design code (MHURD, 2010), seven ground motion records were scaled such that their mean spectra matched the 5%-damping design response spectrum for rare earthquakes shown in Figure 5.5. Table 5.6 lists the chosen ground motion records and scaling factors, which were selected from the PEER database (Ancheta et al., 2013). Both the north-south and east-west components were applied during testing, where it was randomly selected which direction would correspond to the FA direction of the turbine. Seismic loading was applied to operational turbines, which were simultaneously subjected to 10 m/s SW loads in the FA direction. Three cases were derived from each scaled time history: one where the only loading was in the FA direction, one where the loading was in the SS direction, and one where both the FA and SS components were applied.

No.	Earthquake name and station	Scaling factor
1	Imperial Valley-02 (1940) — El Centro Station #9	1.55
2	Imperial Valley-06 (1979) — El Centro Array #12	1.53
3	Superstition Hills-02 (1987) — Westmorland Fire Station	1.38
4	Manjil Iran (1990) — Abbar	0.51
5	Chi-Chi Taiwan (1999) — TCU122	1.97
6	Iwate Japan (2008) — IWT010	3.17
7	Darfield New Zealand (2010) — Christchurch Cashmere HS	0.86

Table 5.6: Selected scaled ground motion records

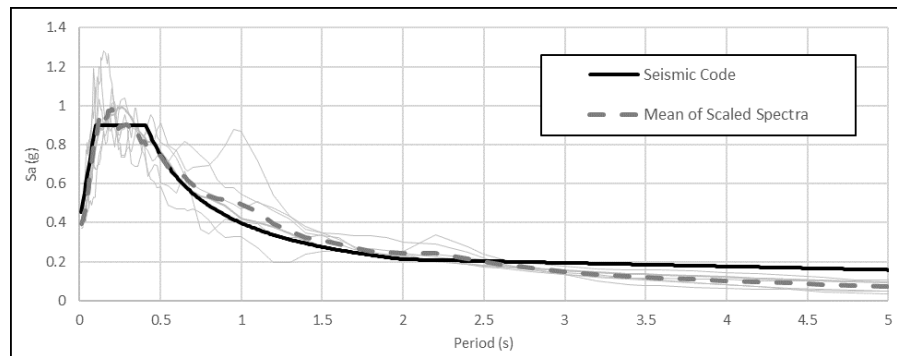


Figure 5.5: Mean of seven scaled seismic record response spectra compared to the seismic code requirement for rare earthquakes

5.2.3 TMD models

TMDs reduce vibrations in structures by applying a restoring force in response to movement, as shown in Figure 5.6. This restoring force can be deconstructed into two components: a velocity-based damping component (F_D) and a displacement-based stiffness component (F_S). This is demonstrated by Equations 5.4-5.5, which are the equations of motion of the main structure and the TMD for the 1-DOF case shown in Figure 5.6:

$$m_1\ddot{x}_1 + c_1\dot{x}_1 + k_1x_1 = F - F_D - F_S \quad (5.4)$$

$$m_2\ddot{x}_2 = F_D + F_S \quad (5.5)$$

Here m_1 , c_1 , k_1 are the mass, damping and stiffness of the main structure; m_2 is the mass of the TMD; x_1 , \dot{x}_1 , \ddot{x}_1 are the displacement, velocity and acceleration of the main structure; x_2 , \dot{x}_2 , \ddot{x}_2 are the displacement, velocity and acceleration of the TMD; $F(t)$ is the applied force on the main structure from wind and/or seismic loading; F_D is the damping component of the restoring force which is primarily a function of $\dot{x}_1 - \dot{x}_2$; and F_S is the stiffness component of the restoring force which is primarily a function of $x_1 - x_2$.

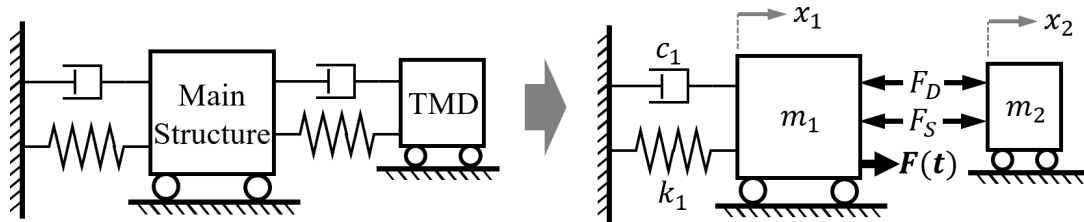


Figure 5.6: Approximating TMD restoring forces applied to main structure in 1-DOF case as F_D and F_S

In this study, the response of turbines equipped with eight different TMD systems were compared to an equivalent uncontrolled turbine. This paper investigates a TMD placed within the top section of the turbine tower. Since such a system would not be able to rotate with the nacelle, a 2D TMD capable of displacing in both the FA and SS directions was used. In the FEM model, the resulting forces from the upper TMD were applied to the lateral FA and SS DOFs at the top node of the tower. In some cases, a secondary TMD was also placed at the fifth node of the model from the top (see Figure 5.3), which was approximately 40 m above the ground. This location represents the maximum response of the second mode shape of the tower and has been found to be a failure point of turbine towers under seismic loading (Zhao et al., 2019a). All eight damper systems used the same collective modal mass ratio of 3%. Damper optimization was performed using the same performance indices analyzed in Section 3 of this paper. Additionally, a maximum TMD stroke limit of ± 1 m was placed on all TMDs, which was assumed as a reasonable upper limit due to the 3 m interior diameter of the top section of the turbine tower. The specifics of the eight TMDs, their controllers, and how they calculate F_D and F_S are detailed in the following sections.

5.2.3.1 Summary of test cases

Table 5.7 summarizes the eight TMD cases studied under each load case. The tests consisted of the uncontrolled turbine; four single-TMD cases (one passive and three semi-active that control

Test case:	Upper TMD (in all cases)			Lower TMD (in MTMD cases)		
	TMD mass (modal ratio μ_R)	Damping component (F_D) control	Spring component (F_S) control	TMD mass (modal ratio μ_R)	Damping component (F_D) control	Spring component (F_S) control
No TMD	0	—	—	0	—	—
Single-PTMD	m_T (3%)	Passive: Section 5.2.3.2	Passive: Section 5.2.3.3	0	—	—
Single- F_D control	m_T (3%)	Semi-active: Section 5.2.3.4	Passive	0	—	—
Single- F_S control	m_T (3%)	Passive	Semi-active: Section 5.2.3.5	0	—	—
Single- F_D & F_S control	m_T (3%)	Semi-active	Semi-active	0	—	—
Multi-PTMD	$0.9m_T$ (2.7%)	Passive	Passive	$0.1m_T$ (1.1%)	Passive	Passive
Multi- F_D control	$0.9m_T$ (2.7%)	Semi-active	Passive	$0.1m_T$ (1.1%)	Passive	Passive
Multi- F_S control	$0.9m_T$ (2.7%)	Passive	Semi-active	$0.1m_T$ (1.1%)	Passive	Passive
Multi- F_D & F_S control	$0.9m_T$ (2.7%)	Semi-active	Semi-active	$0.1m_T$ (1.1%)	Passive	Passive

Table 5.7: Summary of different TMDs. See Figure 5.3 for TMD locations and specified sections for controller details

F_D , F_S , or both) and four similar MTMD cases consisting of a large upper TMD and a smaller lower TMD. In these MTMD cases the upper TMD was either passive or semi-active, however the lower TMD was always passive as preliminary testing found that semi-active control of this smaller TMD did not resulting in significant response improvement. In Table 5.7, m_T is the mass of the TMD in the single-TMD cases, which was equal to 3% of the first modal mass of the turbine, as defined by the modal mass ratio μ_R . The combined mass of the two TMDs in the MTMD cases were made equal to the mass of the single TMD cases so that the effectiveness of the two methods could be roughly compared, and a 90-10 split between upper and lower masses was found to be near optimal – this results in a 2.7% mass ratio compared to the first mode for the upper TMD and a 1.1% mass ratio compared to the second mode for the lower TMD. The optimized parameters for the TMD controllers are summarized in Section 5.2.3.6.

5.2.3.2 Passive damping component

The passive damping component was modelled in the PTMD and F_S control cases, and was tuned following the Den Hartog guidelines presented in Connor (2002). Equations 5.6 - 5.9 show the calculation procedure for the damper force F_D based on the modal mass ratio (μ_R):

$$\xi_j = \sqrt{\frac{\mu_R(3 - \sqrt{0.5\mu_R})}{8(1 + \mu_R)(1 - 0.5\mu_R)}} \quad (5.6)$$

$$Q_j = 1 - 1.2\mu_R \quad (5.7)$$

$$c_j = 2(\mu_R m_{m_j})(Q_j f_{n_j})\xi_j \quad (5.8)$$

$$F_D = c_j \cdot (\dot{x}_1 - \dot{x}_2) \quad (5.9)$$

Here, Q_j is the optimal natural frequency ratio between the TMD and turbine, ξ_j is the damping coefficient, m_{mj} is the modal mass, f_{nj} is the natural frequency, c_j is the damping coefficient, and $(\dot{x}_1 - \dot{x}_2)$ is the relative velocity of the TMD (see Figure 5.6). $j = 1$ refers to the first mode and $j = 2$ refers to the second mode; recall that the upper TMD is always tuned to the first mode and the lower TMD is tuned to the second.

5.2.3.3 Passive stiffness component

The passive stiffness component was modelled in the PTMD and F_D control cases, using the same guidelines and nomenclature from Section 5.2.3.2. Equations 5.10-5.11 show the calculation procedure for the spring force (F_S) based on the modal mass ratio (μ_R):

$$k_j = (\mu_R m_{mj})(Q_j f_{nj})^2 \quad (5.10)$$

$$F_S = k_j \cdot (x_1 - x_2) \quad (5.11)$$

Here, k_j is the stiffness, and $(x_1 - x_2)$ is the relative displacement of the TMD [Figure 5.6].

5.2.3.4 Semi-active damping component via MR damper

The semi-active damping component was modelled in the F_D and F_D & F_S Control cases, and consisted of a magnetorheological (MR) damper — MR dampers are filled with a fluid whose viscosity changes when subjected to an electric current, allowing their properties to be controlled in real time. MR dampers have been frequently studied for turbine vibration control (Martynowicz, 2017, 2016; Caterino et al., 2016; Martynowicz, 2015; Caterino et al., 2014; Caterino, 2014). These dampers can be numerically simulated using the Modified Bouc-Wen (MBW) model (Talatahari et al., 2012; Caterino et al., 2011; Spencer et al., 2004), which is shown in Figure 5.7. Though there have been criticisms of the accuracy of the MBW model for modelling very powerful dampers (Chae et al., 2012), it should be accurate for the relatively small damper modelled in these tests. Equations 5.12-5.18 list the calculations used for determining the damper force (F_D):

$$F_D = \alpha_{MR} \cdot z + c_{MR0} \cdot (\dot{x}_R - \dot{y}) + k_{MR0} \cdot (x_R - y) + k_{MR1} \cdot (x_R - x_{R0}) \quad (5.12)$$

$$\dot{z} = -\gamma_{MR} \cdot |\dot{x}_R - \dot{y}| \cdot z \cdot |z|^{n_{MR}-1} - \beta_{MR} \cdot (\dot{x}_R - \dot{y}) \cdot |z|^{n_{MR}} + A_{MR} \cdot (\dot{x}_R - \dot{y}) \quad (5.13)$$

$$\dot{y} = \frac{1}{c_{MR0} + c_{MR1}} \cdot [\alpha_{MR} \cdot z + c_{MR0} \cdot \dot{x}_R + k_{MR0} \cdot (x_R - y)] \quad (5.14)$$

$$\alpha_{MR}(u_{MR}) = \alpha_{MRa} + \alpha_{MRb} \cdot u_{MR} \quad (5.15)$$

$$c_{MR0}(u_{MR}) = c_{MR0a} + c_{MR0b} \cdot u_{MR} \quad (5.16)$$

$$c_{MR1}(u_{MR}) = c_{MR1a} + c_{MR1b} \cdot u_{MR} \quad (5.17)$$

$$\dot{u}_{MR} = -\eta_{MR} \cdot (u_{MR} - v_{MR}) \quad (5.18)$$

Here, u_{MR} is the efficient voltage; v_{MR} is the applied voltage; z is the hysteretic displacement; x_R is the relative displacement of the MR damper to the turbine (equal to $(x_1 - x_2)$ from Equation 5.4); y , k_{MR0} , k_{MR1} , c_{MR0} , c_{MR1} are the displacement, stiffnesses, and damping shown in Figure 5.7; and α_{MRa} , α_{MRb} , c_{MR0a} , c_{MR0b} , c_{MR1a} , c_{MR1b} , k_{MR0} , k_{MR1} , x_{R0} , γ_{MR} , β_{MR} , A_{MR} , n_{MR} , η_{MR} are the fourteen parameters of the MBW model for a given MR damper which are derived

from experimental testing. This paper reused the MBW model properties given in Li et al. (2017) for a 2.4 kN RD-8041-1 MR damper. For simplicity, the backwards difference method was used to solve the differentials of z and y .

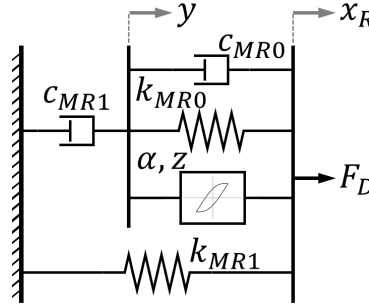


Figure 5.7: Modified Bouc-Wen model: a numerical approximation of a MR damper

Martynowicz (2017) studied several types of varying-damper STMD controllers and found that a modified ground-hook (MGH) controller was highly effective and simple to implement, thus it was applied here. The MGH controller is defined by Equation 5.19:

$$v_{MR} = \begin{cases} v_{max}, & x_1 \cdot F_D \geq 0 \\ v_{min}, & x_2 \cdot F_D < 0 \end{cases} \quad (5.19)$$

Here v_{MR} is the voltage applied to the MR damper, which was either v_{max} or v_{min} , depending on the state of the system. A simplified explanation of this controller is that it maximizes resistance force by applying the maximum voltage to the MR damper when the structure moves away from its neutral position and minimizes resistance force by applying the minimum voltage when the structure returns towards its neutral position. The MGH controller can be displacement-based or velocity-based, each of which have been shown to have their own advantages (Park et al., 2019), but the displacement-based method was used here as that was the method employed by Martynowicz (2017). For thoroughness, a more complex LQR controller (Hrovat et al., 1983) for the MR damper was also tested, but the MGH controller was ultimately found to be equally effective while also being simpler to implement. During testing, an artificial 10 ms delay was added to the model between calculating and applying the desired voltage to more closely simulate a real mechanical system (Caterino et al., 2013).

Note that as opposed to a linear damper, the MBW model includes a varying stiffness component (see Equation 5.12). Previous research has shown that the stiffness component contributes very little to the overall resistance force (Caterino et al., 2011) – in this study the optimized MR damper temporarily increased the stiffness of the TMD by a maximum of 2%. Thus this additional stiffness was ignored in the F_D Control case where the passive spring system cannot easily account for this increase in stiffness, but was accounted for in the F_D & F_S Control case [see Section 5.2.3.5] using k_{MR} : the equivalent MR stiffness at a given time step which was calculated using Equation 5.20, which uses the same symbols defined above:

$$k_{MR} = \frac{k_{MR0} \cdot (x_R - y) + k_{MR1} \cdot (x_R - x_{R0})}{x_R - x_{R0}} \quad (5.20)$$

5.2.3.5 Semi-active stiffness component via SAIVS device

The semi-active varying-stiffness controller was used in the F_S and F_D & F_S Control cases, and employs the SAIVS system (Sun and Nagarajaiah, 2013; Nagarajaiah, 2009; Nagarajaiah and Sonmez, 2007) to modify the stiffness of the TMD in real-time. This system uses an actuator to adjust an array of four springs to actively change the equivalent stiffness of the TMD. Equation 5.21 shows the equation for the equivalent stiffness as a function of the actuator displacement d_{SAIVS} in meters:

$$k_2(d) = k_{SAIVS} \cdot \cos^2 \left[\frac{\pi}{2} \cdot \sin \left(\frac{\pi(0.3 - d_{SAIVS})}{0.6} \right) \right] \quad (5.21)$$

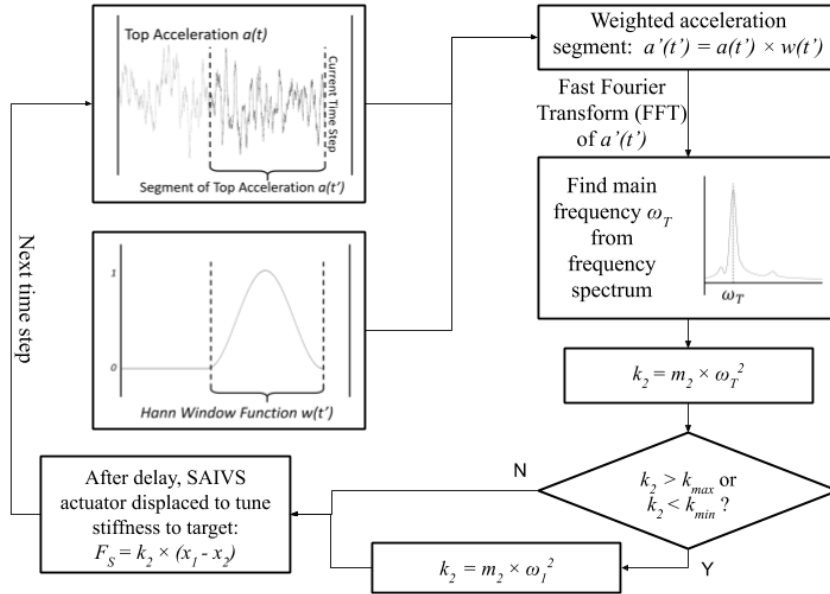
Here d_{SAIVS} could range from 0 m (the minimum stiffness) to 0.3 m (the maximum stiffness). It was found that the total stiffness range required could be achieved when k_{SAIVS} was equal to 160 kN/m.

The SAIVS was tuned by using a short-term Fourier transform (STFT) to determine the dominant frequency of the recent lateral acceleration of the top of the turbine tower and tuned the TMD to match that frequency: a widely-used technique (Park et al., 2019; Hemmati and Oterkus, 2018; Sun, 2017; Dinh et al., 2016; Arrigan et al., 2011; Huang et al., 2010). Some studies tune the varying-spring to the dominant frequency of the displacement of the turbine tower (Sun, 2017), but acceleration was used in this study due to the improved practicality as in reality an accelerometer may be used to easily capture the acceleration response. This STFT controller applied a Hann window function (Equation 5.22) to a segment of the previous top lateral acceleration response to extract the dominant frequency, which the TMD was then tuned to. If this target frequency was outside of the range of the SAIVS system, the TMD was instead tuned to the main structural frequency using Equations 5.10-5.11. In either case, the target stiffness at the given time step was reduced by k_{MR} [Equation 5.20]: the stiffness added by the varying-damper system, though $k_{MR} = 0$ in all configurations but in the F_D & F_S Control cases. A tight allowable frequency band was found to be the overall most effective at controlling the behaviour of the structure – Dinh et al. (2016) similarly used an STFT variable stiffness controller with a tight allowable band to control their turbine. The entire controller process is detailed in Figure 5.8. An artificial 20 ms delay was inserted between calculating the target stiffness and implementing it to model the time required for the physical system to adjust.

$$w(t) = \sin^2\left(\frac{\pi t}{T}\right); \quad t \in [0, T] \quad (5.22)$$

5.2.3.6 Parameter optimization

Parameter optimization was performed manually via parametric study of the response of the TMD-equipped turbine subjected to fifteen random load cases (five each of the EQ, HIW, and SW load time histories). The averaged displacement and acceleration response of the turbine (which is fully detailed in Section 5.3) was used as the optimization index for the four parameters. Initial values of each parameter were estimated based on previous research, and the value of each parameter was individually studied in 0.05 V, 0.5 s, and 0.05 increments for the

Figure 5.8: Flowchart of a single time step of the stiffness component (F_S) control process

Optimized parameters	Load type:		
	EQ	HIW	SW
F_D Control: Max voltage v_{max} (V) [Eq. 5.19]	3	1.5	0.75
F_D Control: Min voltage v_{min} (V) [Eq. 5.19]		0	
F_S Control: Window time (s) [Fig. 5.8]		2	
F_S Control: stiffness limit k_{max} (N/m) [Fig. 5.8]		$1.5m_2f_n^2$	

Table 5.8: Optimized parameters for the 3% modal mass ratio TMDs

voltages, window times, and stiffness limit, respectively. The final combination of optimized parameters was confirmed to achieve superior results compared to all other tests performed during the parametric study. The v_{max} parameter was optimized on a per-load-type basis for increased effectiveness, while the rest of the parameters converged on the same optimal value regardless of loading condition. The ability to change parameters based on the measured loading type is an advantage of a semi-active vibration control system, and could be achieved in reality using the existing equipment for wind speed measurement and an accelerometer at the base of the turbine for EQ load cases. Table 5.8 lists the optimized parameters for the 3% modal mass ratio TMDs used in the final testing.

5.2.4 Numerical algorithm

The explicit Chen-Ricles (CR) numerical algorithm (Chen et al., 2009) was used in this testing, which is summarized in Equations 5.23-5.25:

TMD case	Stiffness value K_2 (N/m)	Damping value C_2 (Ns/m)
Single TMD	36700	2200
Upper MTMD	33300	1900
Lower MTMD	302000	1400

Table 5.9: List of approximate stiffness and damping values used for calculating the CR integration parameters of the semi-active TMDs

$$\dot{x}_{j,i+1} = \dot{x}_{j,i} + \Delta t \cdot \alpha_j \cdot \ddot{x}_{j,i} \quad (5.23)$$

$$x_{j,i+1} = x_{j,i} + \Delta t \cdot \dot{x}_{j,i} + \Delta t^2 \cdot \alpha_j \cdot \ddot{x}_{j,i} \quad (5.24)$$

$$\alpha_j = \frac{4 \cdot M_j}{4 \cdot M_j + 2 \cdot \Delta t \cdot C_j + \Delta t^2 \cdot K_j} \quad (5.25)$$

Here, $j = 1$ refers to the turbine and $j = 2$ refers to the TMD as in Figure 5.6; i refers to the time step; x_j , \dot{x}_j , \ddot{x}_j are the displacement, velocity and acceleration of the given mass; and Δt is the length of the time step in this analysis. α_j refers to the integration parameters for the CR algorithm, which is calculated using Equation 5.25 where M_j , C_j , K_j refers to the mass, damping, and stiffness matrices of the given mass. In the MTMD cases, separate integration parameters were calculated for each TMD. In the semi-active cases where the stiffness and damping are not constant, approximate values were used in these calculations; the original authors (Chen et al., 2009) as well as testing here have found that the sensitivity of the integration parameter is very low, thus using these approximated values did not introduce notable error into the final results. Table 5.9 lists a summary of the stiffness and damping values used for each damper system ($j = 2$) for calculating the integration parameter. The CR algorithm was employed to accommodate future real-time testing.

Figure 5.9 details the full CR algorithm process used in this testing for a simplified 1-DOF system (using nomenclature from Figure 5.6), where x_1 , \dot{x}_1 , \ddot{x}_1 are the time histories of the displacement, velocity and acceleration responses of the main mass and x_2 , \dot{x}_2 , \ddot{x}_2 are likewise for the TMD; n is the total number of time steps in the analysis; and F is the time history of the loading applied to the main mass.

<p>Inputs: $F, K_1, K_2, C_1, C_2, M_1, M_2$</p> <p>Outputs: $x_1, \dot{x}_1, \ddot{x}_1, x_2, \dot{x}_2, \ddot{x}_2$</p> <ol style="list-style-type: none"> 1. Calculate α_1, α_2 using Eq. 5.25 2. For $i = 2$ to n <ol style="list-style-type: none"> 3. Calculate $x_1(i), \dot{x}_1(i)$ using Eqs. 5.23-5.24 4. Calculate $x_2(i), \dot{x}_2(i)$ using Eqs. 5.23-5.24 5. Calculate F_D and F_S depending on TMD case – [see Section 5.2.3] 6. Calculate $\ddot{x}_2(i)$ using TMD's equation of motion – Eq. 5.5 7. Calculate $\ddot{x}_1(i)$ using TMD's equation of motion – Eq. 5.4 8. End
--

Figure 5.9: Numerical integration algorithm for a simplified 1-DOF system

5.3 Results of TMD comparisons

Each of the 121 load cases were applied to the uncontrolled wind turbine as well as turbines with the eight different TMD cases. Figure 5.10 shows a selection of response time histories under EQ loading. The effectiveness of the various TMD systems were evaluated based on the FA and SS accelerations and displacements at the top of the turbine tower as well as at the maximum of the second mode of the structure where the lower TMD was placed, which are hereafter referred to as the **Top** and **Mid** points respectively. Since the TMDs only affect the fluctuating component of the response, the static component of the response was disregarded during analysis. The absolute mean of the displacements and accelerations over the entire 85 s time history were compared to find the improvement of the response due to the TMDs. This process for a given load case is summarized in Equations 5.26-5.27:

$$\chi_{avg,n} = \text{mean} \{ | \chi_n(\cdot) - \text{mean}\{\chi_n(\cdot)\} | \}, \quad n = 1, 2, \dots, 9 \quad (5.26)$$

$$\text{imp}_{\chi_{avg,n}} = \frac{\chi_{avg,1} - \chi_{avg,n}}{\chi_{avg,1}}, \quad n = 2, 3, \dots, 9 \quad (5.27)$$

Here n refers to the TMD case where the results of the uncontrolled turbine are 1 and the Single-PTMD, Single- F_D Control, Single- F_S Control, Single- F_D & F_S Control, Multi-PTMD, Multi- F_D Control, Multi- F_S Control, and Multi- F_D & F_S Control case results are 2 through 9 respectively (see Table 5.7); $\chi_n(\cdot)$ is the entire time history of the response of interest – displacement or acceleration in the FA or SS directions at the Top or Mid point of the turbine; $\chi_{avg,n}$ is the absolute average mean of the fluctuating component used for comparison; $\text{imp}_{\chi_{avg,n}}$ is the percent improvement of the given TMD case compared to the uncontrolled turbine subjected to the same loading.

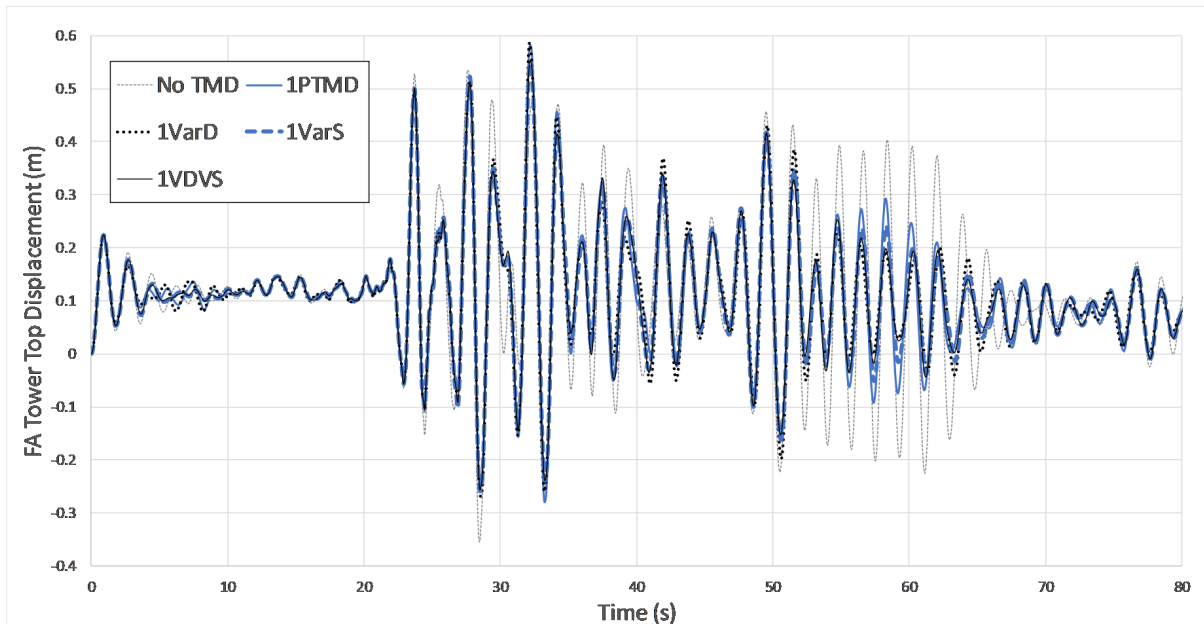


Figure 5.10: Displacement time histories from EQ6 load case for single TMD cases

As both acceleration and displacement have been used as performance indices in previous testing [see Table 5.1], both were considered here. Acceleration is important to consider as the large inertial mass of the nacelle can cause large accelerations in the middle of the tower. The comparative improvements of each TMD case over the uncontrolled turbine for all the EQ, HIW, and SW load cases have been averaged and are shown in Tables 5.10-5.12. The eight response indices are averaged together across each TMD case to provide a broad comparison of the effectiveness of each TMD system, though this number may overvalue reductions in certain response indices depending on design goals.

These results clearly show that in all instances the addition of a TMD improved the response of the turbine compared to the uncontrolled case. The additional damping provided by the TMD had a much greater impact on controlling displacements in the SS direction where the structure lacked aerodynamic damping compared to the FA direction. Since the structure was parked in the HIW load cases and lacked aerodynamic damping, the TMDs were notably more effective in the FA direction here as well. In general, the TMDs were more effective at reducing the average displacement of turbines under HIW loading compared to EQ and SW loading.

The PTMD cases were shown to perform nearly as effectively as the semi-active cases. While the STMDs surpassed the PTMD in all cases, the difference in improvement of the average response reductions was only 2-3%. This difference was greatest when controlling the acceleration and displacement of the Mid point of the turbine, particularly in the Single-PTMD case, compared to the most effective semi-active cases. The presence of realistic mechanical delays within the semi-active systems may play a part in why the mechanically-simpler PTMD performed so comparatively well. The F_S Control was overall the least effective semi-active controller, typically only showing a 0.5-1% difference in improvement in average response reduction compared to the PTMD. Due to the narrow tuning range of the F_S controller, it was very effective at controlling the response of the top of the tower (generally surpassing the F_D Control cases) but was less effective at controlling the Mid response of the tower – similar to the PTMD cases.

The F_D Control cases consistently reduced the turbine response compared to the PTMDs and the F_S Control systems, particularly when looking at the response of the Mid point of the tower where it was up to 4 times as effective, indicating that this system was more effective at controlling the second mode response of the structure due to a greater robustness compared to the narrowly-tuned F_S Control cases. It suffered compared to the F_S Control cases at controlling the top of the tower, possibly due to the passive spring system not accounting for the additional stiffness provided by the MR damper in the varying-damper system.

The F_D & F_S Control cases showed even further improvement in response reduction compared to the F_D Control and F_S Control cases individually – combining the varying-spring and varying-damper systems tended to result in improvements that were roughly cumulative of the improvements provided by the individual semi-active systems compared to the PTMDs. This improvement was slightly less than cumulative under EQ loading and slightly more under SW loading. The F_D & F_S controller was the most effective of all tested.

The introduction of the second passive TMD in the MTMD cases improved the overall average response reduction for all cases. The improvement was slightly larger under EQ loading (where the second mode of the structure was more heavily excited) and the SW loading (where the single TMDs were least effective at controlling the Mid response). It can be plainly seen that the MTMD systems were extremely effective at controlling the response of the Mid point

$\mu_R = 3\%$ TMD case:	Displacement				Acceleration				Overall average:
	Fore-aft		Side-side		Fore-aft		Side-side		
	Top	Mid	Top	Mid	Top	Mid	Top	Mid	
Single-PTMD	9.6%	9.8%	33.4%	34.0%	9.7%	1.0%	25.7%	4.9%	16.0%
Single-F_D	9.0%	9.2%	34.8%	34.8%	11.2%	2.6%	28.5%	6.9%	17.2%
Single-F_S	9.2%	9.4%	35.2%	35.2%	12.2%	1.2%	29.5%	5.7%	17.1%
Single-F_D & F_S	10.2%	10.3%	36.7%	36.7%	12.2%	2.9%	30.3%	7.1%	18.3%
Multi-PTMD	8.2%	8.7%	32.3%	32.3%	11.6%	25.6%	26.1%	27.5%	21.8%
Multi-F_D	8.4%	8.5%	34.9%	34.9%	13.0%	26.9%	29.4%	29.7%	23.3%
Multi-F_S	9.4%	9.2%	35.1%	35.1%	13.3%	25.7%	30.8%	28.5%	23.2%
Multi-F_D & F_S	9.7%	10.0%	36.2%	36.2%	13.4%	26.9%	31.4%	29.6%	24.2%

Table 5.10: Average % improvement compared to uncontrolled turbine under EQ loading

$\mu_R = 3\%$ TMD case:	Displacement				Acceleration				Overall average:
	Fore-aft		Side-side		Fore-aft		Side-side		
	Top	Mid	Top	Mid	Top	Mid	Top	Mid	
Single-PTMD	49.5%	49.7%	38.5%	37.5%	42.0%	2.5%	48.9%	3.9%	34.0%
Single-F_D	49.0%	50.4%	38.7%	37.7%	41.3%	5.5%	50.2%	6.2%	34.9%
Single-F_S	50.2%	49.2%	39.2%	38.1%	44.7%	2.5%	50.7%	4.0%	34.8%
Single-F_D & F_S	51.0%	51.2%	38.9%	37.9%	44.8%	5.6%	51.4%	6.1%	35.9%
Multi-PTMD	47.6%	47.9%	37.8%	36.8%	41.3%	10.7%	49.3%	20.1%	36.5%
Multi-F_D	48.4%	48.9%	37.4%	36.5%	42.4%	14.0%	50.4%	22.7%	37.6%
Multi-F_S	48.6%	48.7%	37.6%	37.3%	44.6%	10.7%	50.5%	20.2%	37.3%
Multi-F_D & F_S	49.3%	49.6%	38.3%	36.7%	44.9%	14.0%	51.1%	22.7%	38.3%

Table 5.11: Average % improvement compared to uncontrolled turbine under HIW loading

$\mu_R = 3\%$ TMD case:	Displacement				Acceleration				Overall average:
	Fore-aft		Side-side		Fore-aft		Side-side		
	Top	Mid	Top	Mid	Top	Mid	Top	Mid	
Single-PTMD	8.6%	8.6%	39.8%	40.3%	6.0%	0.4%	30.0%	2.2%	17.0%
Single-F_D	9.0%	9.1%	39.8%	41.1%	6.0%	1.8%	30.8%	4.0%	17.7%
Single-F_S	9.0%	9.1%	40.6%	40.3%	6.0%	0.4%	30.9%	2.3%	17.3%
Single-F_D & F_S	9.2%	9.2%	41.9%	42.5%	6.6%	1.9%	31.5%	4.0%	18.3%
Multi-PTMD	8.5%	8.6%	38.5%	39.0%	7.7%	19.8%	31.5%	20.3%	21.7%
Multi-F_D	8.6%	8.9%	38.6%	40.5%	7.7%	21.5%	32.0%	22.3%	22.5%
Multi-F_S	8.8%	8.7%	39.8%	39.1%	8.1%	19.8%	32.1%	20.4%	22.1%
Multi-F_D & F_S	8.8%	8.9%	41.2%	41.9%	8.6%	21.5%	33.1%	22.3%	23.3%

Table 5.12: Average % improvement compared to uncontrolled turbine under SW loading

Optimized parameters	Load type:		
	EQ	HIW	SW
F_D Control: Max voltage v_{max} (V) [Eq. 5.19]	1.5	0.5	0.3
F_D Control: Min voltage v_{min} (V) [Eq. 5.19]		0	
F_S Control: Window time (s) [Fig. 5.8]		2	
F_S Control: stiffness limit k_{max} (N/m) [Fig. 5.8]		$1.5m_2f_n^2$	

Table 5.13: Optimized parameters for the 1.5% modal mass ratio TMDs

Load type:	TMD case:	$\mu_R = 3\%$				Displacement				Overall average:
		Fore-aft		Side-side		Fore-aft		Side-side		
		Top	Mid	Top	Mid	Top	Mid	Top	Mid	
EQ	Single- F_D	9.6%	9.8%	33.4%	34.0%	9.7%	1.0%	25.7%	4.9%	16.0%
	Single- F_D & F_S	9.0%	9.2%	34.8%	34.8%	11.2%	2.6%	28.5%	6.9%	17.2%
HIW	Single- F_D	9.2%	9.4%	35.2%	35.2%	12.2%	1.2%	29.5%	5.7%	17.1%
	Single- F_D & F_S	10.2%	10.3%	36.7%	36.7%	12.2%	2.9%	30.3%	7.1%	18.3%
SW	Single- F_D	8.2%	8.7%	32.3%	32.3%	11.6%	25.6%	26.1%	27.5%	21.8%
	Single- F_D & F_S	8.4%	8.5%	34.9%	34.9%	13.0%	26.9%	29.4%	29.7%	23.3%

Table 5.14: Average % improvement compared to uncontrolled turbine under EQ loading

of the tower – in some cases 50 times as effective – compared to the single TMD cases, though typically at the cost of a slight reduction of control of the top of the turbine tower.

To further analyze the advantages of the STMDs over the PTMDs, parameter optimization was performed (as per Section 5.2.3.6) for turbines equipped with 1.5% modal mass ratio TMDs (as opposed to the 3% mass ratio used previous), which is summarized in Table 5.13. These new TMD-equipped turbines were subjected to the same 121 load cases; Table 5.14 lists the resulting average response improvement of a limited number of controlled turbines under the three loading types. It can be seen that, as before, the average reduction effect of the Single- F_D & F_S Control case surpasses the Single- F_D Control case in all cases.

Table 5.15 compares the ratio of the average response reduction by the 1.5% modal mass ratio Single- F_D Control and Single- F_D & F_S Control cases from Table 5.14 to the average response reduction of the 3% modal mass ratio Single-PTMD system from Tables 5.10-5.12. It can be seen that the response reduction provided by the 1.5% modal mass ratio Single- F_D & F_S controller is only slightly reduced (by about 7%) compared to the response reduction provided by the 3% modal mass ratio Single- F_D Control case. Since a smaller TMD mass can result in

Load type:	$\mu_R = 3\%$	$\mu_R = 1.5\%$			
	Average reduction by Single-PTMD	Average reduction by Single-PTMD	Ratio vs 3% Single-PTMD	Average reduction by F_D & F_S Control	Ratio vs 3% Single-PTMD
EQ	16.0%	11.6%	0.73	15.2%	0.95
HIW	34.0%	27.1%	0.80	30.9%	0.91
SW	17.0%	14.5%	0.85	15.9%	0.94
		Average:	0.79	Average:	0.93

Table 5.15: Comparison of average % improvement between 3% mass ratio Single-PTMD to 1.5% mass ratio Single-PTMD and Single- F_D -and- F_S -Control STMD

a reduced P-delta effect on the tower, smaller space requirements, and simpler installation, it is a reasonable design goal to achieve the target response reduction with as small of a TMD mass as possible. This example shows that a notably lighter semi-active TMD can achieve a nearly equivalent response reduction effect compared to a heavier passive damping system.

5.4 Conclusions and recommendations

This paper presents a comparative study of eight tuned mass damper (TMD) systems used for vibration control of an onshore wind turbine subjected to service wind, high-intensity wind and seismic loading. A single passive TMD, three single semi-active TMDs (damping control, stiffness control, and damping-and-stiffness control), a passive multiple TMD (MTMD) system, and three mixed passive-and-semi-active MTMDs (where the upper TMD was semi-actively controlled using the three previous methods and the lower TMD was passive) were compared to an uncontrolled turbine. The design of these TMD systems were approached from a practical perspective using specific mechanical systems to achieve the variable properties and including realistic mechanical delays and limits. Overall, all the TMDs improved the response of the wind turbine, but the Multi- F_D & F_S Control case was the most effective when considering all performance indices. The semi-active F_D Control and F_S Control cases also showed improved effectiveness compared to passive TMDs (PTMDs) cases, with the former showing improved control of the middle of the tower and the latter showing improved control of the top of the tower. The use of a MTMD system with the same collective mass as a single TMD resulted in improved control of higher mode responses at the cost of a slight reduction in control of the first mode. The improvement of the STMDs versus the PTMD cases was relatively small depending on the load case and index of interest, and must be weighed against the increased cost of implementing the semi-active systems, though in this specific case, a semi-active damper with half the mass was found to be almost equally effective as an optimized passive damper. Overall, the absolute best damper will depend on the desired control for each performance index, the expected loading, and a cost-benefit analysis.

Future testing could build on this research by applying these dampers to a more widely-used wind turbine size such as NREL's standard 5MW wind turbine (Jonkman et al., 2009). Research could likewise be expanded by considering the effects of non-stochastic wind loads such as tornados (Gairola and Bitsuamlak, 2019) or downbursts Aboshosha et al. (2015a), potentially through the use of more robust simulation methods such as computational fluid dynamics (Dagnev and Bitsuamlak, 2013). Results could be refined by employing more modern wind spectrum generation techniques, as the methods suggested in the IEC code (IEC, 2005) have been found lacking in other research (Aboshosha et al., 2015b). Furthermore, it is possible that the controllers used here could be further improved which would directly impact the comparative effectiveness of the different semi-active systems.

The TMD system shown here to best balance simplicity and robustness was the single multi-directional TMD equipped with the F_D & F_S Control system, which would be located at the top of the turbine tower below the nacelle with a braking system to allow maintenance workers to pass safely. Further study of this STMD using a more detailed FEM model as a practical damping system for controlling wind turbine vibrations is warranted.

Chapter 6

Development of a rotating-base residential wind turbine model based on aeroelastic wind tunnel tests

Summary

The development of improved hybrid simulation frameworks for studying wind turbines is a promising endeavour due to inherent limitations in small-scale wind tunnel tests of these structures. For use in investigations of such a framework, an experimentally-validated numerical structural turbine model must be developed that both avoids the existing scaling issues and incurs aeroelastic deflection to create an ideal testbed for future hybrid simulations. This study details the development and validation of a structural turbine model that meets the listed requirements. Firstly, two rounds of wind tunnel testing are described and catalogued of a full-scale 1.1 m residential wind turbine installed on a custom aeroelastic rotating base. Secondly, using existing research as well as the results for the aeroelastic wind tunnel tests, a 1-DOF structural model is developed and validated that replicates the behaviour of the 1.1 m rotating-base turbine. Part of this development includes the creation of a simple surrogate thrust model using a trained multi-layer perceptron neural network. The basic structural properties, the sources of loading, and the damping properties are then derived and validated from the experimental data. Finally, the developed structural model is applied in a representative numerical study that tests the effectiveness of passive and semi-active tuned mass dampers at controlling the structural rotation. The detailed structural model is shown to match the experimental results well and is effective in the representative case study, opening the door for applying it in future hybrid simulations.

6.1 Introduction

Structural and wind engineering-based upgrades to wind turbines are critical for improving their efficiency and longevity. A relatively new testing technique called hybrid simulation (HS) improves existing research as well as enables new types of studies. HS has seen widespread use in seismic engineering but is just beginning to see applications for wind-loaded structures. As described in detail in Chapter 2, HS works by simultaneously testing an experimental model and a numerical model together to “fill out” the experimental model with cheaper numerical data. This allows for decreased cost, increased scope, and even new types of testing compared to traditional experimental tests, but with higher accuracy than purely numerical simulations. To capture damping and inertial effects, hybrid simulations are frequently run in real time, that is, where one second of simulation time is calculated within one second of real time. This has been the main limiting factor for the adoption of HS in wind engineering: compared to calculating seismic loads, calculating the aerodynamic loads on a structure is significantly more time consuming. Beyond the simplest drag models, the estimation of pressure and shear on the surface of the structure is a complex process, and the highest-accuracy numerical method, computational fluid dynamics (CFD), potentially takes weeks in real time for just seconds of simulation time. Therefore the current HS framework faces the burden of requiring lower-accuracy, high-speed aerodynamic models. Future testing to develop an improved wind engineering HS framework that better facilitates the use of this technique in wind engineering is thus required.

Wind turbines in particular stand as a promising option for HS. This is because the typical method of experimental study of wind-loaded structures, wind tunnel testing, runs into unique challenges when studying wind turbines. Wind tunnel testing of structures typically requires significant geometric scaling of the model, but this scaling affects facets of the analysis. The Reynolds number (Re), a non-dimensional parameter that describes the wind turbulence, is a factor of both the wind speed and the size of the structure; if a structure is reduced to half the size, the wind speed must double to maintain the same Re flow over it. Since large structures are potentially scaled down hundreds of times to fit in the wind tunnel, it is clear that maintaining the same Re flow is impossible for many wind tunnel tests. As the aerodynamic drag and lift on the wind turbine blades are extremely sensitive to changes in Re , industrial-sized wind turbine cannot merely be geometrically scaled down for wind tunnel tests. Instead, researchers must totally redesign the model turbine blades to approximate the lift and drag properties of the full-sized blade at these lower Re . Examples of this are presented in Section 2.5.1.1, where researchers are generally able to match the lift properties of the blade while sacrificing the accuracy of the simulated drag, which can affect the predicted power production notably. However, if an improved HS framework could be developed that would allow wind turbines to be studied with experimental-level accuracy while side-stepping wind tunnel scaling issues, it would open the door to a number of new research avenues – a very valuable proposition.

Towards developing a framework that remedies these issues, an experimentally-validated structural model is required on which to test any new HS frameworks. The model must be experimentally-validated to offer a baseline against which the existing and any proposed HS frameworks may be compared. This chapter presents a set of wind tunnel testing designed to side-step the scaling issues of typical wind tunnel tests of turbines while offering an appropriate testbed with which to perform future hybrid simulations. These wind tunnel tests, as well as previous testing on the same wind turbine, are then used to develop a robust, widely-validated

1-DOF dynamic structural model that can be used in future wind turbine HS. Finally, a short numerical case study is performed of the structural model equipped with a non-linear tuned mass damping (TMD) system to show the utility of the proposed model and its potential for future studies.

6.2 Wind tunnel testing

To offer an appropriate testbed for future HS, the wind tunnel tests that will be used to generate the structural model must meet the following requirements:

- The tests must be able to avoid inaccuracies arising from the geometric scaling of the wind turbine blades. A direct way of accomplishing this is by performing a full-scale wind tunnel test of a smaller residential wind turbine, eliminating scaling-induced inaccuracies as no scaling is performed.
- The tests must be significantly aeroelastic. Since the main limitation of existing HS of wind turbines is that the aerodynamic and structural calculations are uncoupled, any testbed to investigate this must have significant wind-induced structural deflections so that this limitation can be explored. Further, external damping systems, a common HS target, require larger deflections to cause a significant reduction in vibrations.

6.2.1 Residential wind turbine specifications

The turbine studied in these wind tunnel tests is the 900 W three-bladed horizontal-axis wind turbine with a rotor radius of 1.1 m shown in Figure 6.1. The average diameter of the tower is approximately 0.08 m, and while the tower height can be adjusted, it has typically been tested at a hub height of 1.9 m. The manufacturer reports a rated wind speed of 12.9 m/s and a maximum wind speed of 17 m/s with a furling system to reduce rotor speed and wind loads at higher wind speeds. It has an optimal tip speed ratio (TSR) of 5.2 which results in a rotor speed of 580 rpm at the rated wind speed. The turbine generator is connected to a transformer that in operation would be connected to a battery, however by replacing the battery with a series of resistors, the electrical resistance of the system can be controlled which in turn controls the friction on the rotor from the generator. In practice, this means that by adjusting the electrical resistance in the circuit, the friction in the rotor can be controlled, allowing for a specific friction level to be chosen to control the TSR of the turbine during a wind tunnel test.

The 1.1 m residential wind turbine has previously been studied with a fixed tower base at the Boundary Layer Wind Tunnel Laboratory (Refan, 2009; Refan and Hangan, 2012) and the WindEEE Research Institute (Shirazadeh et al., 2021), both at Western University in London, Canada. Refan (2009) first tested one of its turbine blade in a wind tunnel, then the entire turbine, however the relatively high blockage ratio of these entire-turbine tests raises questions about their accuracy. The blockage ratio issue was avoided by Shirazadeh et al. (2021) by testing in a larger wind tunnel, where they subjected the residential wind turbine to uniform wind loads and a number of extreme wind effects. In both instances, the turbine was tested at much lower wind speeds and TSRs than the reported rated conditions; due to the extremely high rotor speeds that occur in residential wind turbines, the rated conditions cannot be tested without risking damage to the turbine or other wind tunnel equipment.

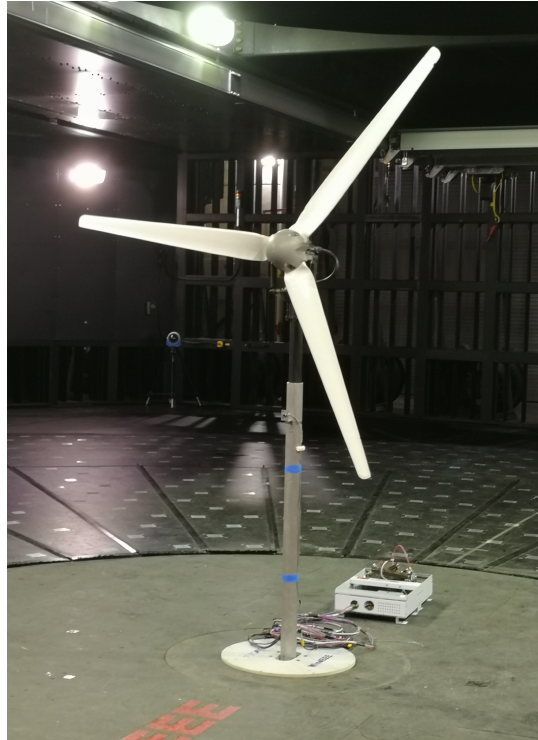


Figure 6.1: Photo of the 1.1 m residential wind turbine with the connected transformer

6.2.2 Aeroelastic base design and construction

The existing fixed base on the 1.1 m residential wind turbine can be replaced with a new base that the existing tower can slot into. This allows for the turbine to be modified to enable aeroelastic response without requiring any permanent modifications to the turbine itself. With the goal of developing a simple structural model for future HS testing, an aeroelastic rotating base for the tower was designed and built. It allows the turbine to rotate in the fore-aft direction at ground level while the tower remains essentially rigid as shown in in Figure 6.2. This type of response mimics the pitch rotation in floating offshore wind turbines.

Figure 6.3 shows a schematic of the developed aeroelastic rotating base. It shows that the tower axle rotates in ball bearing mounts, with a fixed frame beneath the floor of the wind tunnel that is connected to the tower via springs as well as pneumatic dampers in some tests. These springs and dampers control the structural dynamic properties of the rotating wind turbine without disrupting its aerodynamics. The detailed specifications of this rotating base include:

- The dimensions of the aeroelastic base are shown in Figure 6.3. The tower was built from an aluminum pipe, with a welded aluminum rod to act as the axle. The ground-level element is built from medium-density fibreboard, with aluminum slotted frame elements and a plywood support piece at the bottom.
- The rotational stiffness of the system was controlled by four linear springs. The springs were four No. 5108N762 extension springs from McMaster-Carr with the following dimensions: 3.5" length, 1" outer diameter, and 0.095" wire diameter with a rated stiffness of 1524 N/m.

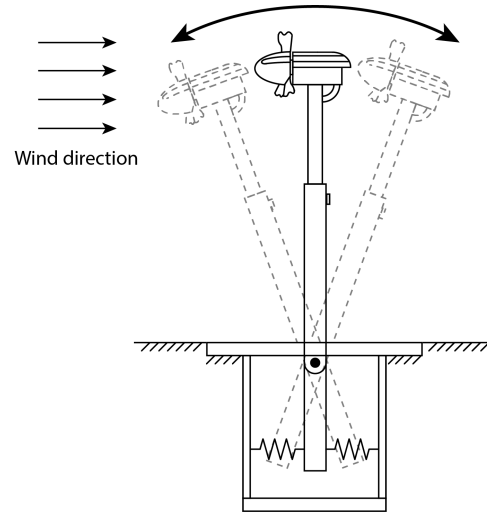


Figure 6.2: Side-view of proposed aeroelastic turbine response

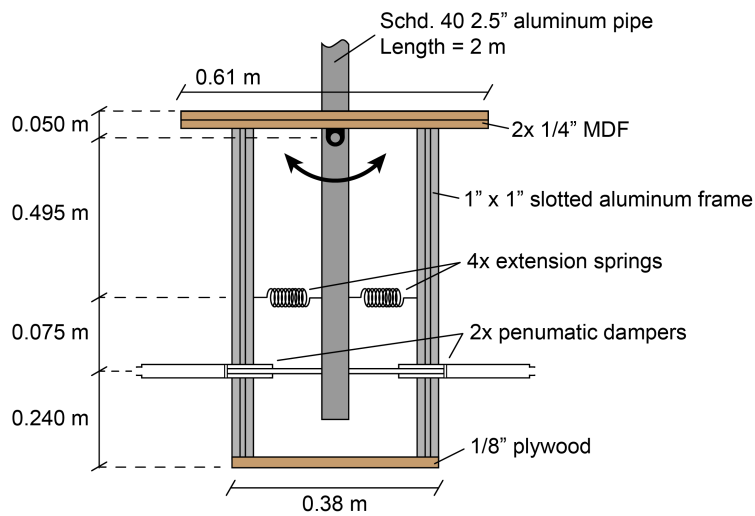
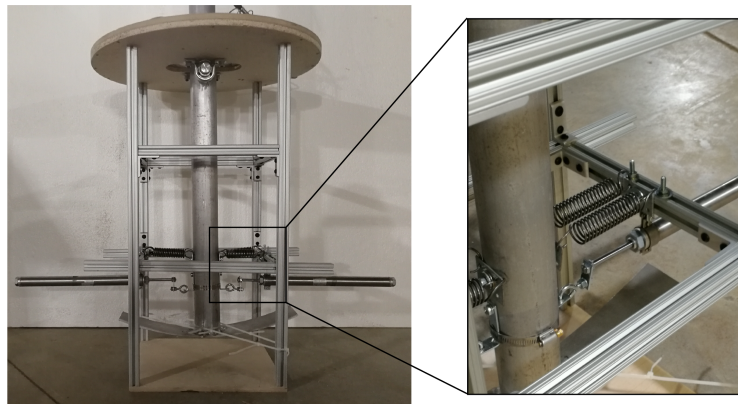


Figure 6.3: Photos and schematic of the below-tunnel portion of the aeroelastic base with the stiffness and damping systems highlighted

- The rotational damping of the system was controlled in initial tests by using two pneumatic dampers. These pneumatic dampers were two NCME 106-1200-DUPO3176 double-action air cylinders from SMC with the following dimensions: $1-\frac{1}{16}$ " (26.99 mm) interior bore size, 12" stroke, and a $\frac{3}{8}$ " (9.35 mm) aperture diameter. The apertures of the cylinders were left open so they acted as pneumatic dampers rather than pneumatic actuators, though additional fixtures with openings of 4.25, 3.13, and 0.63 mm were screwed into the aperture to further increase the damping level.

6.2.3 Testing facility and data acquisition

As was done previously by Shirazadeh et al. (2021), the WindEEE Research Institute was used to study the rotating-base turbine, avoiding any problems with high blockage ratios. The WindEEE Research Institute houses a 25 m diameter, 3.8 m high hexagonal wind tunnel. The facility is equipped with 106 fans to enable complex wind studies such as tornadoes and downbursts, however for traditional wind tunnel tests such as this only the 60 fans in the front wall are used to generate inflow.

While the tunnel is equipped with adjustable blocks to generate a variety of wind profiles, complex wind profiles were not necessary for this testing, and thus an open profile was used. From previous testing at a mean uniform wind speed of 6.6 m/s, the open profile in the wind tunnel assuming all fans are blowing at equal power matches very closely to a power law profile with an exponent of 0.05, as seen in Figure 6.4. It is possible that this profile does not match perfectly at the lower wind speeds studied here, however this approximation is sufficient for use in the development of the structural model.

The final turbine setup in the wind tunnel can be seen in Figure 6.5, including the measurement devices used to capture the wind speed, tower rotation, and rotor speed during testing.

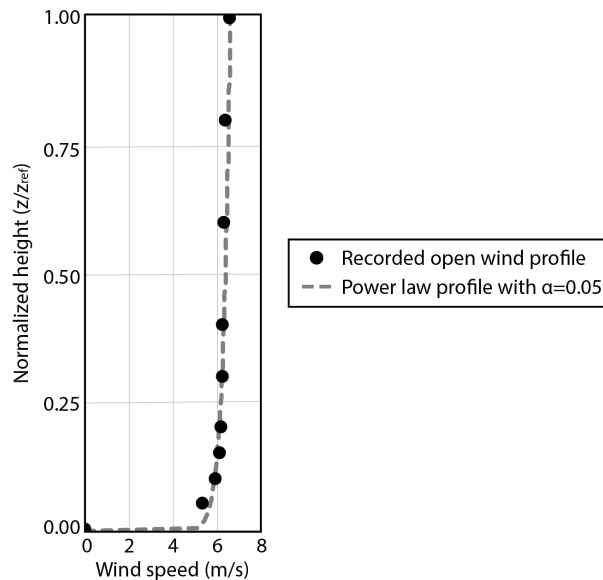


Figure 6.4: Comparison of open profile recorded at WindEEE with a reference wind speed of 6.6 m/s to a power law profile with exponent (α) of 0.05

The wind speed measurement was performed using four TFI cobra probes that capture the time histories of all three velocity directions as well as the air pressure. These four cobra probes were placed at the corners of the square that bound the rotor plane of the turbine, i.e. at ± 1.1 m above and ± 1.1 m across from the hub, with the tips of the cobra probes 1.05 m upwind of the rotor plane. This configuration is shown in Figure 6.6 and is the same as was used by Shirazadeh et al. (2021).

The rotor speed was captured using an Autonotics BR200-DDTN proximity sensor attached to the turbine tower. The voltage from the sensor peaks when the blade passes in front of the sensor, and thus by determining the frequency of the peaks the rotor speed and the resulting TSR can be derived. Finally, the rotation of the turbine tower was recorded using a HL-G112-A-C5 laser distance sensor from Panasonic which has a measurement range of ± 0.06 m, the center of which is about 0.12 m from the sensor. Since the laser sensor measures linear displacement, the measured displacement was converted into a rotation by multiplying by the laser height of 0.60 m (assumption small-angle rotation). Both the proximity and laser sensors are shown in Figures 6.5 and 6.6.

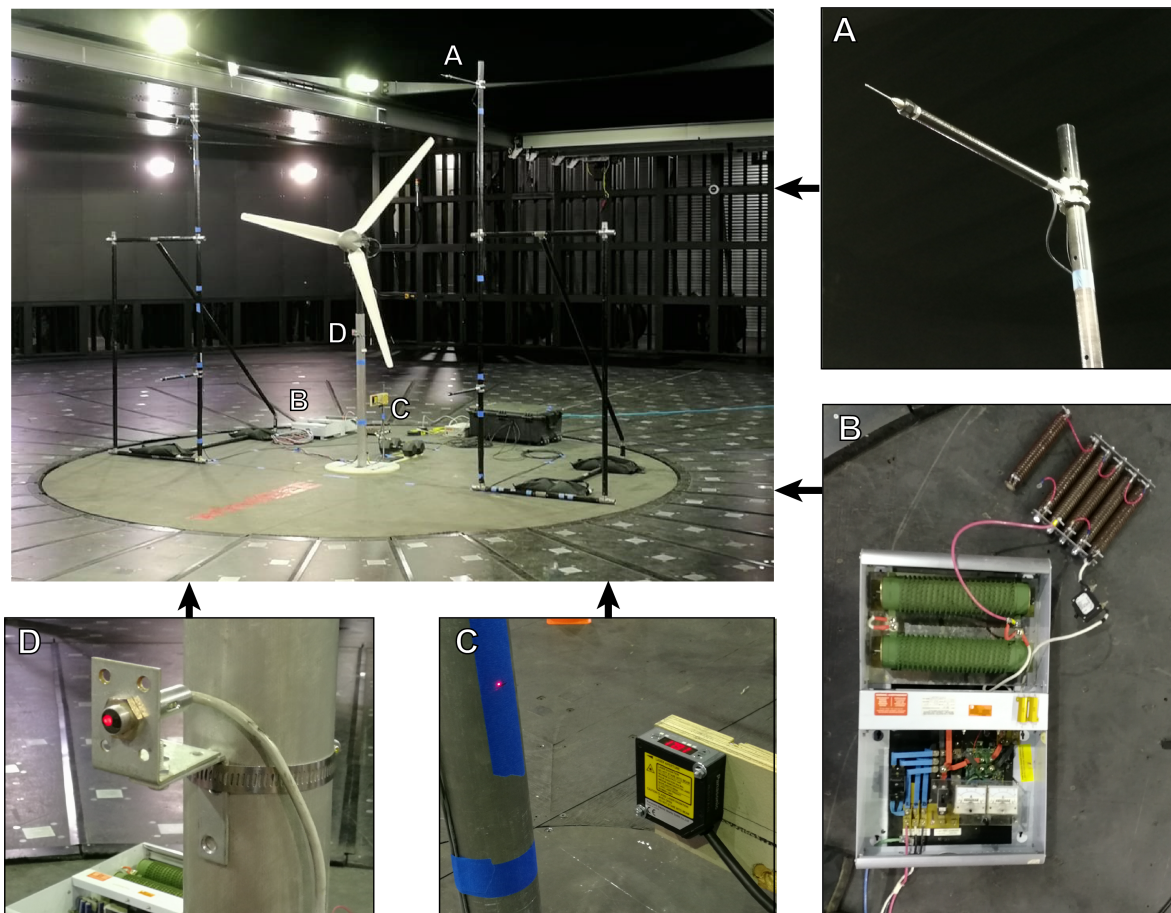


Figure 6.5: Photograph of aeroelastic turbine setup for wind tunnel testing. Equipment of note is highlighted: A) one of the four cobra probes to measure wind speed, B) the generator transformer to control TSR, C) the laser sensor to measure base rotation, D) the proximity sensor to measure TSR

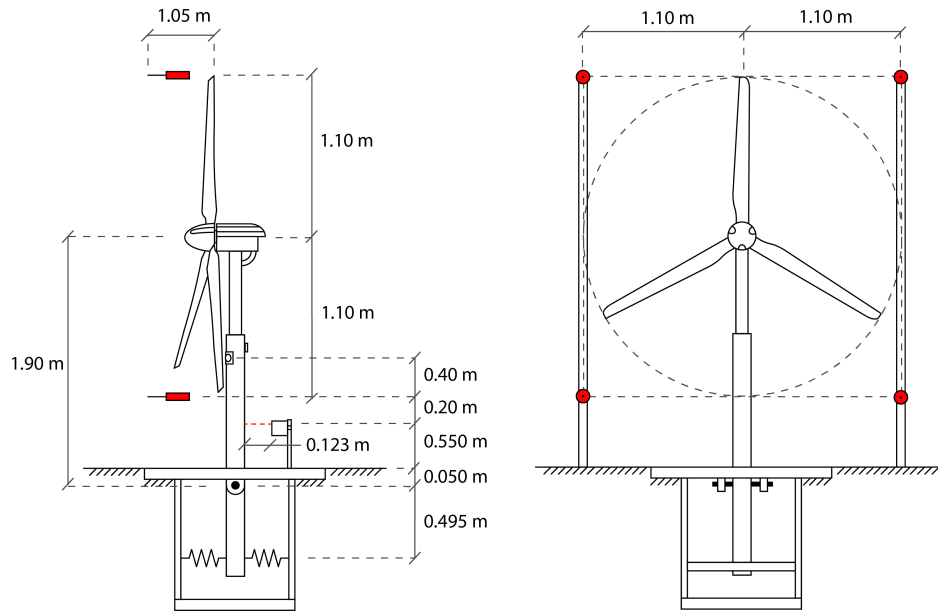


Figure 6.6: Schematic of setup of the rotating-base turbine in the wind tunnel. Cobra probes and laser sensor are highlighted

As will be discussed in the following sections, two rounds of testing were performed: the first round of testing included all of the components discussed in this section, while the second round of testing removed the pneumatic actuators and was performed without the cobra probes. For both of these rounds, free vibration tests were performed in addition to the wind tunnel tests, in which the only measurement of interest is the rotation recorded by the laser sensor.

6.2.4 Wind tunnel testing: first round

Three parameters were varied in this first round of wind tunnel tests:

- The mean hub height wind speed was varied between approximately 3.63, 4.07, and 4.85 m/s.
- The electrical resistance in the generator transformer was varied between 7.1, 12.0, and 16.9 Ω , which in turn controlled the TSR. Recall that the electrical resistance is inversely proportional to the friction in the rotor, meaning that a high resistance results in a higher TSR.
- The aperture size of the pneumatic dampers was varied between 9.35, 4.25, 3.13, and 0.63 mm. The smaller the aperture size, the higher the damping effect.

Due to the requirement to keep rotor speeds low to protect the turbine and surrounding equipment, the combinations of these parameters were limited: for each of the four aperture sizes, six configurations were tested as shown in Table 6.1. Each of the 24 tests had a duration of 180 s and a sampling frequency of 1000 Hz. Between test cases, time was given such that the turbine could reach a steady state in an attempt to avoid capturing transient behaviour.

Configurations	Wind speed (m/s)	Electrical resistance (Ω)
1	3.63	7.1
2	4.07	7.1
3	4.85	7.1
4	3.63	12.0
5	4.07	12.0
6	3.63	16.9

Aperture sizes: 0.63, 3.18, 4.76, 9.53 mm

Table 6.1: Test cases studied in the first round of wind tunnel testing; the six configurations are tested at each damper aperture size

Aperture (m)	Variable	Unit	Config 1	Config 2	Config 3	Config 4	Config 5	Config 6
0.63	Mean wind speed	m/s	3.64	4.10	4.86	3.65	4.07	3.64
	Turbulence intensity	–	6.6%	6.9%	6.8%	6.6%	6.7%	6.6%
	Electrical resistance	Ω	7.1	7.1	7.1	12.0	12.0	16.9
	TSR	–	0.91	1.13	1.43	1.36	1.72	1.84
	Rotor speed	rad/s	3.02	4.22	6.30	4.51	6.37	6.08
	Non-zeroed mean base rotation	rad	0.00636	0.00695	0.01007	0.00634	0.00738	0.00686
	Std. dev. of non-zero base rotation	rad	0.00050	0.00104	0.00121	0.00075	0.00062	0.00060
3.18	Mean wind speed	m/s	3.59	4.05	4.84	3.65	4.11	3.65
	Turbulence intensity	–	6.8%	6.9%	6.6%	6.4%	6.5%	6.5%
	Electrical resistance	Ω	7.1	7.1	7.1	12.0	12.0	16.9
	TSR	–	0.93	1.14	1.44	1.33	1.70	1.88
	Rotor speed	rad/s	3.03	4.18	6.35	4.40	6.37	6.25
	Non-zeroed mean base rotation	rad	0.00285	0.00389	0.01009	0.00891	0.00959	0.00931
	Std. dev. of non-zero base rotation	rad	0.00044	0.00111	0.00121	0.00088	0.00074	0.00059
4.76	Mean wind speed	m/s	3.60	4.05	4.82	3.62	4.04	3.61
	Turbulence intensity	–	6.8%	6.7%	6.8%	6.7%	6.8%	6.7%
	Electrical resistance	Ω	7.1	7.1	7.1	12.0	12.0	16.9
	TSR	–	0.97	1.16	1.41	1.38	1.81	2.20
	Rotor speed	rad/s	3.18	4.28	6.19	4.54	6.67	7.22
	Non-zeroed mean base rotation	rad	-0.00504	-0.00431	0.00020	-0.00645	-0.00437	-0.00641
	Std. dev. of non-zero base rotation	rad	0.00064	0.00128	0.00158	0.00098	0.00075	0.00064
9.53	Mean wind speed	m/s	3.64	4.09	4.85	3.59	4.02	3.59
	Turbulence intensity	–	6.5%	6.6%	6.8%	7.0%	7.1%	7.0%
	Electrical resistance	Ω	7.1	7.1	7.1	12.0	12.0	16.9
	TSR	–	0.90	1.10	1.45	1.50	1.87	2.10
	Rotor speed	rad/s	2.99	4.08	6.38	4.90	6.83	6.87
	Non-zeroed mean base rotation	rad	0.00154	0.00209	0.00450	0.00333	0.00401	0.00377
	Std. dev. of non-zero base rotation	rad	0.00042	0.00081	0.00100	0.00065	0.00059	0.00048

Table 6.2: Mean results of the first round of wind tunnel testing. The test configurations are defined in Table 6.1

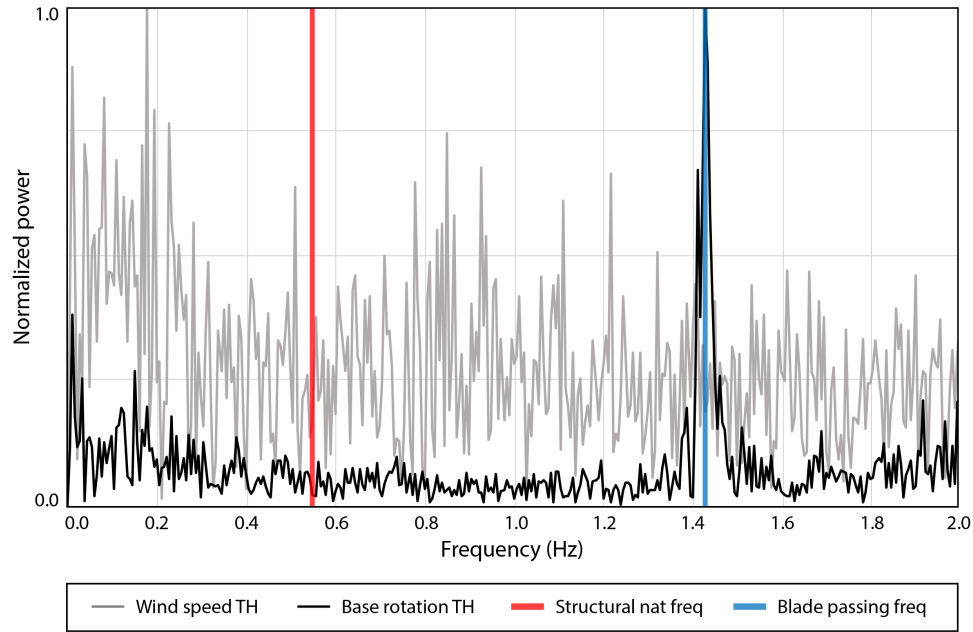


Figure 6.7: Normalized frequency content of the recorded wind speed and tower rotation time histories, compared to the natural frequency of the structure and blade passing frequency. These results are from the test with an aperture size of 9.53 mm, a wind speed of 3.63 m/s, and an electrical resistance of 7.1 Ω

Table 6.2 summarizes the results of these tests. The extremely low standard deviation of the base rotation is notable – these measurements clearly show that the tower fluctuations are very small, in the order of less than 0.001 rad. To investigate further, the frequency content of the recorded wind speed and tower response from the 9.35 mm aperture size, 4.07 m/s wind speed, and 12.0 Ω electrical resistance case was analyzed in MATLAB using the fast Fourier transform algorithm. Additionally, from the free vibration tests, the natural frequency of the rotating base turbine was estimated at 0.548 Hz, and based on the recorded TSR, the frequency of the blade passing the turbine tower is approximately 1.43 Hz. This analysis is summarized in Figure 6.7; as can be seen, despite the wind record containing the frequency content necessary to excite the rotating-base turbine at its natural frequency, the recorded base rotation sees essentially no response at that frequency. Instead, there is a large response spike at the frequency at which the blades pass the tower. This strongly suggests that the minuscule recorded fluctuations are a result of the operational rotor and there are essentially no aeroelastic vibrations caused by the wind loading.

Suppressed vibrations are a phenomenon seen in over-damped structures. To investigate if this was the case for the rotating-base turbine, the damping properties of the pneumatic actuators can be estimated numerically. Referencing Figure 6.8, by assuming incompressible flow and ignoring any friction effects the Bernoulli equation can be used to estimate the damping force applied to the moving plunger:

$$p_A + 0.5\rho V_A^2 + \rho gh_A = p_B + 0.5\rho V_B^2 + \rho gh_B \quad (6.1)$$

Here, A refers to the area immediately next to the plunger and B to the area adjacent to the damper aperture. p is the air pressure, ρ is the air density, and V is the flow velocity, V_A being

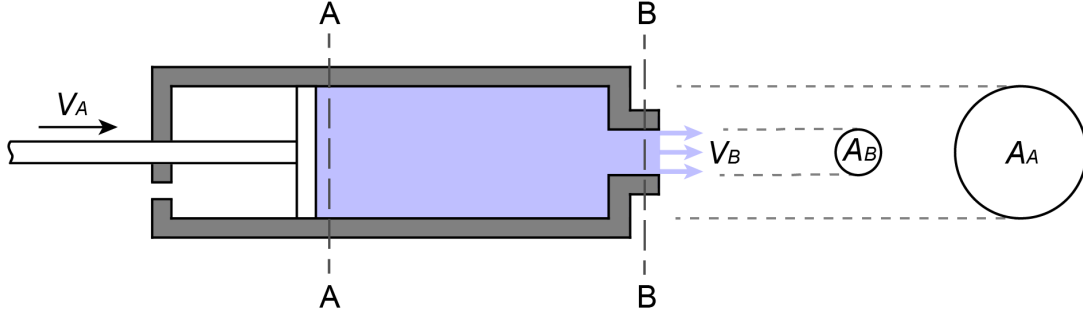


Figure 6.8: Internal view of pneumatic actuator showing sections A and B and their related air velocities (V_A and V_B) and cross-sectional areas (A_A and A_B)

equal to the plunger speed. To simplify, $h_A = h_B$, $p_A = p_0 + \Delta p_A$ (where p_0 is the atmospheric pressure and Δp_A is the difference between p_0 and p_A), $p_B = p_0$, therefore:

$$\Delta p_A + 0.5\rho V_A^2 = 0.5\rho V_B^2 \quad (6.2)$$

Due to flow rate conservation:

$$V_A A_A = V_B A_B \quad (6.3)$$

Here, A_A and A_B is the cross-sectional area at the respective section. By subbing Equation 6.3 into Equation 6.2:

$$\Delta p_A = 0.5\rho V_A^2 \left(\frac{A_A^2}{A_B^2} - 1 \right) \quad (6.4)$$

Since the force acting on the plunger opposite the direction of motion ($F_{plunger}$) can be calculated as the additional pressure times the plunger area, it can also be found that:

$$F_{plunger} = \Delta p_A A_A \quad (6.5)$$

$$F_{plunger} = 0.5\rho A_A V_A^2 \left(\frac{A_A^2}{A_B^2} - 1 \right) \quad (6.6)$$

The damping moment in this rotational system caused by the pneumatic actuators can be quantified using a quadratic damping model, that is a model where the damping moment (M_C) is proportional to the square of the rotational velocity ($\dot{\theta}$):

$$M_C = C_{quad} \dot{\theta} |\dot{\theta}| \quad (6.7)$$

Here, C_{quad} is the quadratic damping coefficient. The quadratic damping model is less commonly used than the traditional linear damping model, however it is appropriate for pneumatic damping systems since, as has been shown, aerodynamic drag force is proportional to the wind velocity squared. This quadratic damping model will be explored in greater detail in Section 6.3.6. Based on Equation 6.7 while using a small-angle assumption, where the vertical distance of the linear damper relative to the rotation point of the turbine is defined as h_C , it can be found that:

Aperture size	–	mm	0.63	3.18	4.76	9.53
Aperture area	A_b	m^2	$1.25 * 10^{-6}$	$3.18 * 10^{-5}$	$7.12 * 10^{-5}$	$2.85 * 10^{-4}$
Damping coefficient	C_{quad}	Nms/rad	1534	2.36	0.47	0.029
Damping moment	M_C	Nm	767	1.18	0.24	0.014

Table 6.3: Predicted damping ratio from simplified Bernoulli model. M_C is calculated assuming $\dot{\theta} = 0.5$ rad/s

$$C_{quad} = 0.5\rho A_A \left(\frac{A_A^2}{A_B^2} - 1 \right) h_C^2 \quad (6.8)$$

Finally, with the knowledge that $A_A = \pi 0.02699^2$ and $h_C = 0.57$ m in the 1.1 m wind turbine, C_{quad} can be determined depending on the aperture size of the pneumatic damper. The damping moment (M_C) can also be calculated assuming an roughly estimated rotational velocity of 0.5 rad/s. These results are shown in Table 6.3.

Table 6.3 reveals that the pneumatic actuators with smaller apertures induced a significant amount of damping into the structure, however it fails to account for the lower amount of vibrations seen in the larger aperture cases as well. Keep in mind that this Bernoulli-based model ignores compressibility, but this assumption may not be valid for the smaller aperture cases as the Mach number of the flow increases. This also fails to account for the friction within the pneumatic damper, observations of the damper indicate that there was a small initial friction to overcome before the piston would begin to move.

Thus, it was decided that the pneumatic dampers added too much uncertainty to the test results. Additionally, it was found that an accurate zero-measurement of the turbine had not been taken in this round of testing, limiting the ability to compare even the mean results to the structural model. Overall this round of testing failed to meet the initial requirements described at the beginning of Section 6.2 due to both the lack of aeroelastic fluctuations and limited applicability of the recorded data, and thus it was decided to perform a second round of wind tunnel tests to attempt to address these inadequacies.

6.2.5 Wind tunnel testing: second round

The second round of wind tunnel testing was run after removing the pneumatic dampers from the aeroelastic base, thus the only sources of damping within the system are the passive structural and aerodynamic values. The system was also tested in two spring configurations: the “full-stiffness” configuration with four springs that were used in the first round of tests, as well as the “half-stiffness” configuration that used only two springs, one on each side of the tower, to encourage greater vibrations in the structure.

Because of limited tunnel time available for the second round of testing, the tests were run without cobra probes at the same 3.83 and 4.07 m/s wind speeds as the first round of testing, such that the mean values used in the first round of testing could be applied to the second round of testing. Similarly, all tests were run with the electrical resistance in the transformer circuit at 12.0Ω to encourage high TSRs. Tests were 180 s in length and data was captured using the proximity and laser sensors at 1000 Hz as was done in the first round of testing.

Variable	Unit	Test 2-1	Test 2-2	Test 2-3	Test 2-4
Mean wind speed (from first round of tests)	m/s	4.07	3.63	4.07	3.63
Turbulence intensity (from first round of tests)	–	6.8%	6.7%	6.8%	6.7%
Electrical resistance	Ω	12.0	12.0	12.0	12.0
TSR	–	1.32	0.95	1.33	0.95
Rotor speed	rad/s	4.88	3.14	4.92	3.14
Stiffness case (number of springs)		Full (4)	Full (4)	Half (2)	Half (2)
Non-zeroed mean base rotation	rad	0.01112	0.00602	0.04005	0.02150
Standard deviation of non-zero base rotation	rad	0.00091	0.00078	0.00324	0.00206

Table 6.4: Results of the second round of wind tunnel testing. Reported wind results are the average values from the first round of testing at the same tunnel settings

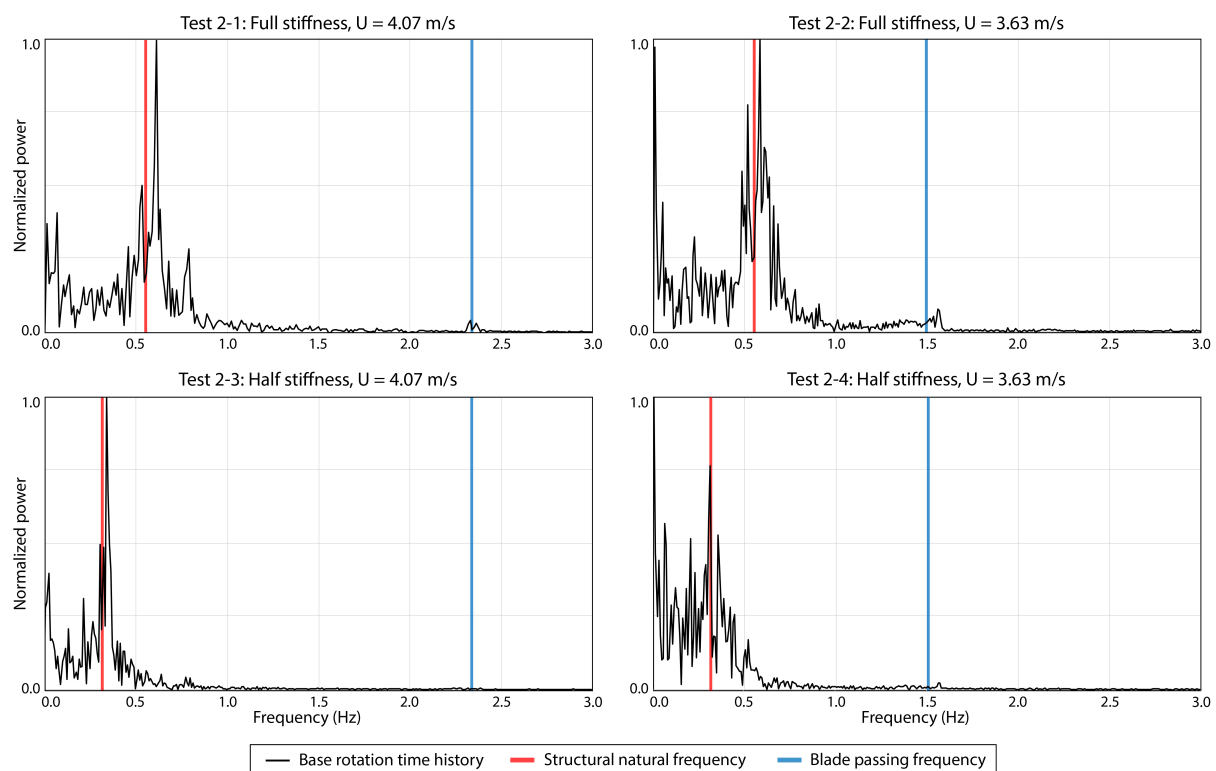


Figure 6.9: Normalized frequency content of the recorded tower rotation time histories, compared to the natural frequency of the structure and the blade passing frequency

Table 6.4 shows the results from this second round of testing. It can be seen that in the full-stiffness case, the standard deviation of the tower rotation has not increased substantially compared to the first round of testing, suggesting that in addition to the strong damping from the pneumatic dampers, the aeroelastic base itself contains a large amount of internal damping. However, when the frequency content of the response is analyzed in Figure 6.9, it can be seen that the structure is now vibrating primarily at its natural frequency rather than its blade frequency – though the response is small it appears to be genuine wind-induced vibrations. The vibrations are greater in the half-stiffness case, where the standard deviation of the response

is several times greater. As seen in Figure 6.9, the vibrations in the half-stiffness case are likewise greatest near the structural frequency. Therefore, while these results may have smaller vibrations than would be ideal, they are sufficient to meet the original testing requirements stated at the beginning of this section and thus will be used to help develop and validate the 1-DOF structural model.

6.2.6 Conclusions and post-mortem

The aeroelastic rotating base ultimately enabled testing that met the requirements for this study: a wind tunnel test where a turbine experiences aeroelastic vibrations that also avoids scaling-induced errors. Using this recorded data, a structural dynamic wind turbine model will be developed.

Unfortunately, there were notable issues with the amount of damping in the system, owing in large part to limited capabilities of pre-tunnel testing as well as the restrictive wind speeds and TSR limits. Had higher wind speeds been allowable, the higher structural damping would have been less impactful on the results.

Future testing of this rotating-base turbine is recommended to be run at higher wind speeds with a greater amount of turbulence, as well as adding a significantly increased amount of electrical resistance in the transformer circuit. The higher wind speeds would encourage greater mean and fluctuating rotations in the tower, the greater turbulence would encourage more fluctuations as well, and the increased electrical resistance would keep the rotor speed at a safe level despite the increased wind speed.

6.3 Dynamic structural model

This section details the derivation of a dynamic structural model of the wind turbine based on the results of the wind tunnel testing detailed in Section 6.2 as well as tests previously performed by other researchers (Refan, 2009; Shirazadeh et al., 2021). As part of this design and validation process, a simplified thrust coefficient model is also developed using a neural network, however in the following chapter of this thesis, a higher-accuracy CFD-based aerodynamic model will be developed and paired with this structural model. Firstly, free vibration tests will be used to extract the dominant structural features of the turbine, followed by the validation of a numerical blade approximation and the development of the surrogate thrust model using a neural network, and finally static and dynamic validations versus the generated wind tunnel data.

6.3.1 Analysis of free vibration tests

The free vibration tests performed in the second round of testing described in Section 6.2.5 allows for the natural frequency, the rotational stiffness, and the rotational inertia to be quantified for an equivalent structural model. Recall that free vibrations tests were performed for both the full-stiffness and half-stiffness cases, which used four and two springs respectively to restrain the tower during rotation. The stiffness of each spring was measured separately, with the average of the four found to be 1557 N/m.

By measuring the average period between peaks in the free vibration results, the natural frequency of the rotational response can be determined for both the full- and half-stiffness cases. These were calculated to be 3.62 rad/s (0.576 Hz) and 2.06 rad/s (0.328 Hz), respectively. It is well documented that in a system dominated by a single mode, as this rotating turbine system is, the natural frequency can be calculated as $\sqrt{k/m}$ where k is the structural stiffness and m is the structural mass. Since this is a rotational system, the variables k_{rot} and m_{rot} are used to designate their rotational frame of reference. It can further be assumed that the rotational mass of the turbine constant between the two tests. Conversely, the system has two significant sources of structural stiffness that vary between the tests: the positive stiffness provided by the spring system and the negative stiffness inherent in an inverted pendulum system. The positive stiffness is quite straightforward, and Figure 6.10 shows how the linear stiffness of the springs can be converted into the rotational stiffness required for the model. The negative stiffness stems from the fact that this most closely resembles an inverted, rigid pendulum, and thus its natural stability is with the pendulum hanging below the fulcrum. The fact that without the springs, the turbine would immediately fall over is the physical manifestation of the base system's negative stiffness. Thus the rotational stiffness of the system is a sum of two terms: the positive spring stiffness ($k_{s,rot}$) and the negative passive stiffness ($k_{p,rot}$).

Therefore, the following set of equations are declared and solved to identify the structural properties of the system:

$$\omega_1 = 3.617 = \sqrt{k_{1,rot}/m_{rot}} \quad (6.9)$$

$$\omega_2 = 2.060 = \sqrt{k_{2,rot}/m_{rot}} \quad (6.10)$$

$$k_{1,rot} = k_{p,rot} + 2k_{s,rot} \quad (6.11)$$

$$k_{2,rot} = k_{p,rot} + k_{s,rot} \quad (6.12)$$

Where subscript 1 refers to the full-stiffness case and subscript 2 refers to the half-stiffness case. From Figure 6.10, if:

$$k_{s,rot} = 2 * 1557 * 0.495^2 = 763.0 \text{ Nm/rad} \quad (6.13)$$

then:

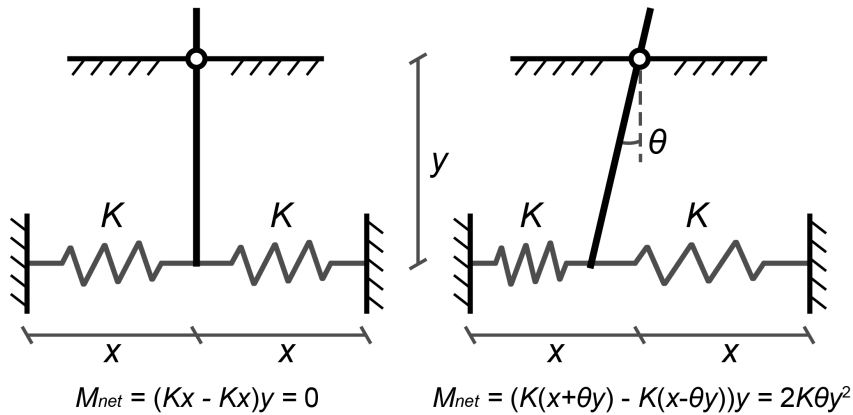


Figure 6.10: Numerical structural model of spring system in rotating base turbine. $M_{rot} = k_{s,rot}\theta$ therefore $k_{s,rot} = 2Ky^2$

$$k_{p,rot} = -397.0 \text{ Nm/rad} \quad (6.14)$$

$$m_{rot} = 86.32 \text{ kgm}^2 \quad (6.15)$$

$$k_{1,rot} = 1129 \text{ Nm/rad} \quad (6.16)$$

$$k_{2,rot} = 366.0 \text{ Nm/rad} \quad (6.17)$$

By simplifying the rotational model as a lumped mass at the end of a mass-less tower, the rotational inertia m_{rot} can be used to estimate the nacelle mass m as:

$$m = m_{rot}/1.9^2 = 23.9 \text{ kg} \quad (6.18)$$

6.3.2 Numerical blade approximation

The blades of the 1.1 m residential wind turbine were previously investigated by Refan (2009). A single blade was tested in Western University's Boundary Layer Wind Tunnel Laboratory on a force-balance base at several wind angles, which allowed lift and drag coefficient curves to be generated for the blades as a whole. Based on these experiments, Refan (2009) also approximated the unknown airfoils of the blade so that it could be numerically simulated. They concluded that the real turbine blades could be approximated using a combination of the NACA 6515 airfoil at the root of the blade and the Wortmann FX 63 137 airfoil at the tip, the profiles of which are shown in Figure 6.11. This design would have been chosen because the FX 63 137 has higher lift at low angles of attack but higher drag at higher angles of attack compared to the NACA 6515, making the former more effective for the tips and the latter more effective at the root. Using published lift and drag coefficient curves of the NACA 6515 airfoil (Jacobs et al., 1932) and the FX 63 137 airfoil (Selig and McGranahan, 2004), they optimized their numerical turbine blade approximation and generated lift and drag coefficient curves for the entire blades using the blade element momentum (BEM) technique (Glauert, 1935). However, these approximations were not without limitations. The previous experimental data used to characterize the airfoils was generated at much higher Re flows compared to those experienced by the 1.1 m turbine in the wind tunnel; the NACA and FX airfoils were previously tested at Reynolds numbers of $3 \cdot 10^6$ and $1 \cdot 10^5$ respectively, while the turbine experiences flows of $2 \cdot 10^4$ to $6 \cdot 10^4$. Furthermore, while the approximation was able to match the lift coefficient curve with some accuracy, it was less successful at modelling the drag coefficient curve. Such prioritization is often seen in turbine blade approximations (Kimball et al., 2014) because the lift force on a blade generally has a greater effect on the structural behaviour of the turbine.

Table 6.5 shows the specifications of the blade approximation used in this study, which is

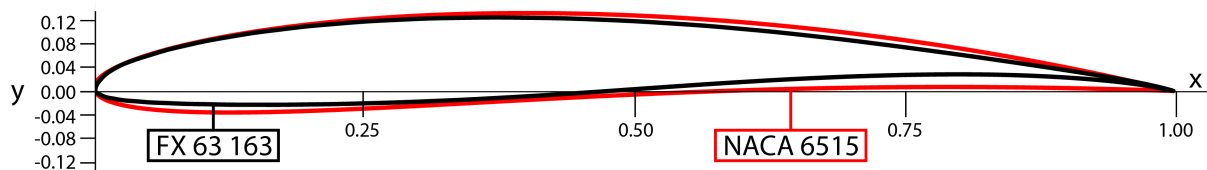


Figure 6.11: Normalized profiles of the NACA 6515 airfoil and the Wortmann FX 63 137 airfoil

derived from the data reported by Refan (2009). In this table, a pitch of 0° refers to the chord line parallel to the rotor plane.

While Refan (2009) directly employed the BEM technique to numerically study this rotor, the more-robust, BEM-based open-source turbine modelling software OpenFAST (Jonkman et al., 2021) was used here. OpenFAST is capable of performing a full aeroelastic analysis of an operational turbine, making it an effective tool for loading and response modelling. This robustness requires a larger amount of input data compared to other techniques – unlike the BEM simulations by Refan (2009) which used experimental data with angles of attack ranging from -6° to 30° , OpenFAST requires full 360° lift and drag coefficient curves of its airfoils. While such data exists for very common airfoils such as the NACA 0012 airfoil (Sheldahl and Klimas, 1981), as far as the author is aware 360° experimental data does not exist for the NACA 6515 and FX 63 137 airfoils. Instead, the open-source 2D airfoil analysis software XFOIL (Drela and Youngren, 2000) was used to generate lift and drag coefficient curves for angles of attack from -80° to 80° for these airfoils. XFOIL employs the simplified panel method (Drela, 1989) to approximate the aerodynamic surface pressure on an airfoil, deriving the lift and drag forces from this. Numerically generating the lift and drag curves using this method had the added benefit of being able to generate the curves for a flow with Reynolds number of 40800, a value which is equal to the median flow used by Refan (2009) and notably lower

Section	Airfoil	r (m)	r/R	Chord (m)	Twist ($^\circ$)	Center
1	NACA 6515	17%	0.19	0.1220	22.6	25%
2	NACA 6515	22%	0.24	0.1180	16.8	25%
3	NACA 6515	26%	0.29	0.1140	12.7	25%
4	NACA 6515	31%	0.34	0.1100	9.7	25%
5	NACA 6515	35%	0.39	0.1060	7.6	25%
6	NACA 6515	40%	0.44	0.1020	5.8	25%
7	NACA 6515	45%	0.49	0.0980	4.3	25%
8	NACA 6515	49%	0.54	0.0940	3.0	25%
9	NACA 6515	54%	0.59	0.0900	2.0	25%
10	Transition*	58%	0.64	0.0860	1.1	25%
11	FX 63 137	63%	0.69	0.0820	0.4	25%
12	FX 63 137	67%	0.74	0.0780	-0.1	25%
13	FX 63 137	72%	0.79	0.0740	-0.6	25%
14	FX 63 137	76%	0.84	0.0700	-1.1	25%
15	FX 63 137	81%	0.89	0.0660	-1.7	25%
16	FX 63 137	85%	0.94	0.0620	-2.1	25%
17	FX 63 137	90%	0.99	0.0580	-2.5	25%
18	FX 63 137	95%	1.04	0.0540	-2.8	25%
19	FX 63 137	99%	1.09	0.0500	-3.0	25%
20	FX 63 137	100%	1.10	0.0375	-3.0	15%

Table 6.5: Specifications of the airfoils lofted to make the numerical approximation of the wind turbine blade, where r is the local radius and R is the tip radius of 1.1 m. The transition airfoil is a radial mean of both airfoils to smooth the transition

than the existing experimental data. The accuracy and stability of XFOIL become quite poor at angles of attack beyond $\pm 90^\circ$, thus an alternative method was required to generate the lift and drag coefficient curves at these points. Instead, the existing 360° lift and drag curves for the NACA 64 airfoil, which are included in OpenFAST, were scaled and shifted to smoothly extend the XFOIL data. The limitation of using scaled data from a different airfoil was deemed to be acceptable as the angle of attack reaching values beyond 90° would rarely occur in this testing. Figure 6.12 shows the generated 360° lift and drag coefficient curves for the NACA 6515 and FX 63 137 airfoils, highlighting where values were generated using XFOIL and where they are scaled NACA 64 data. As would be expected, the FX 63 137 has higher lift at low angles of attack but also higher drag at high angles of attack compared to the NACA 6515. It should be noted that, as seen in Section 3.3.3, there are concerns of the accuracy of XFOIL in stall and post-stall conditions, particularly for drag prediction, thus this approximated blade model requires validation.

While OpenFAST simulates an entire turbine, the single-blade wind tunnel test performed by Refan (2009) can be replicated using OpenFAST by applying a uniform wind field to a parked wind turbine. By setting the blade pitch to near 90° , the leading edge of the blade can be pointed upwind, allowing for the single-blade wind tunnel tests to be recreated and the lift and drag loads at the root of the blade to be extracted. These loads are then converted into lift and drag coefficients for the entire blade using the standard equations, where the area in the

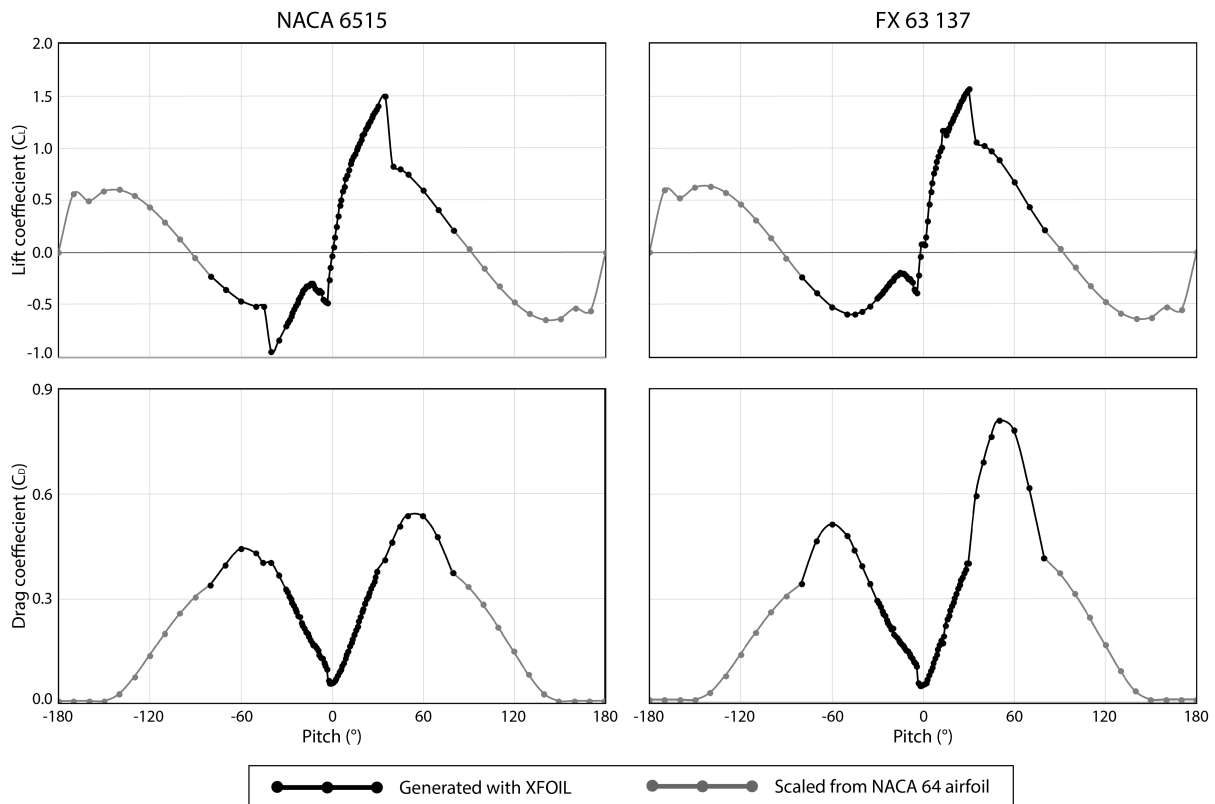


Figure 6.12: Lift and drag coefficient curves for the NACA 6515 and FX 63 137 airfoils at a Reynolds number of 40800

denominator is taken as the blade length (0.91 m) times the average chord length (0.089 m). Curves were generated at wind speeds of 5, 7, and 9 m/s at a turbulence intensity of 1% to match the testing conditions from Refan (2009). Figure 6.13 compares these curves generated in OpenFAST to the experimental and BEM-based numerical curves reported in Refan (2009).

As can be seen in Figure 6.13, there is a large discrepancy between the previous author’s experimental and BEM curves. Conversely, the OpenFAST sims match the experimental results much better, particularly the lift predictions, despite also employing the BEM technique. This improvement is likely due to a number of causes: firstly, as was mentioned, XFOIL was used to generate lift and drag coefficient curve for the airfoils near the Reynolds number of the experimental tests, as opposed to the much larger Reynolds number used by the previous author. XFOIL struggles at predicting drag in particular, which is likely why the lift coefficient curves match the experimental results much better than the drag curves. Secondly, the BEM technique is an older model and has seen numerous improvements by a number of authors including models to account for tip and hub losses – Liu and Janajreh (2012) offer a recent summary of these improvements. OpenFAST employs BEM with these improvements by default, while they are not used by Refan (2009) to generate the single-blade drag and lift coefficient curves.

While the discrepancy between drag coefficient curves is still significant in Figure 6.13, the error can be accepted. The response of the structural model developed in this study will be governed by the rotor thrust, which in turn is primarily governed by the airfoil lift given the fixed pitch of these turbine blades. As a result, the error in drag prediction should have

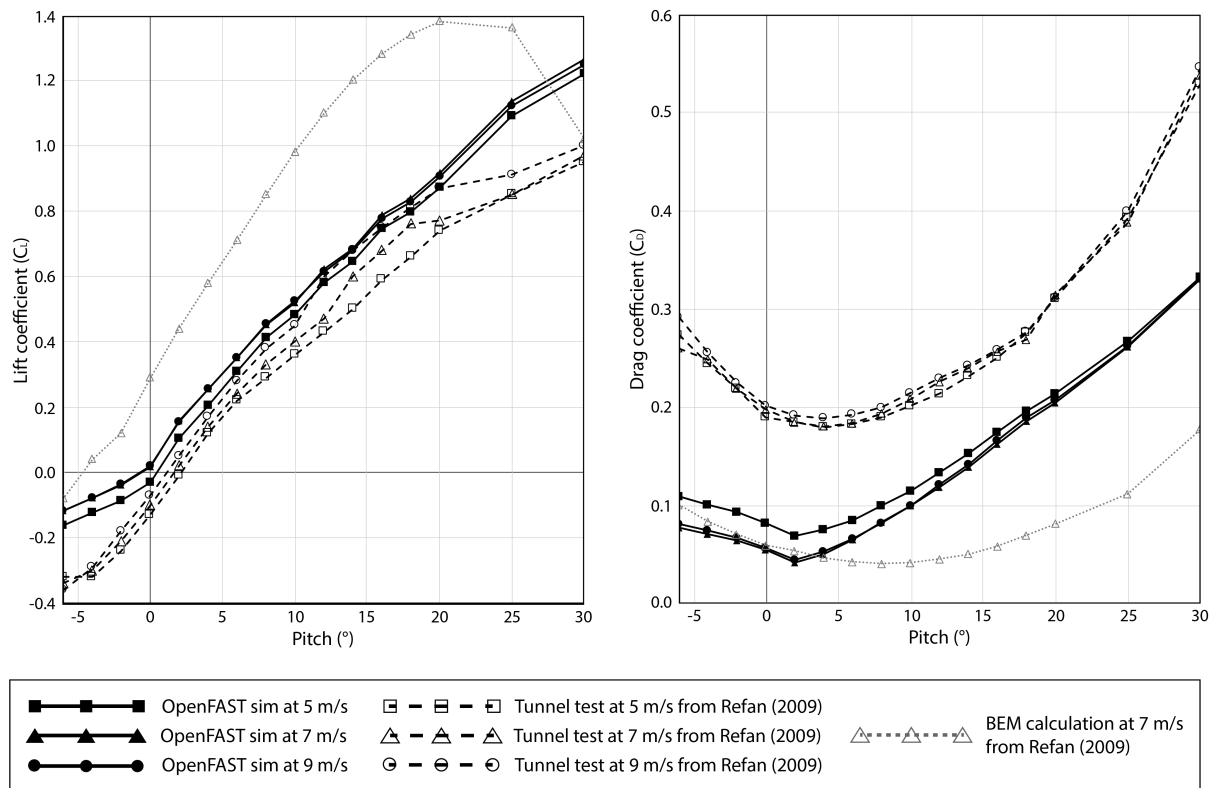


Figure 6.13: Lift and drag coefficient curves of the entire 1.1 m turbine blade generated with OpenFAST compared to wind tunnel and BEM tests from Refan (2009)

a negligible impact on the final model. On the other hand, the small discrepancy in the lift coefficient curves should be considered. By calculating the NRMSE between the OpenFAST and experimental curves at their respective wind speeds and averaging the results, it is found that the lift prediction by the OpenFAST implementation of the numerical approximation is 88% accurate for angles of attack between -6° and 30° .

It must be emphasized that the accuracy of the numerical rotor approximation is not validated at angles of attack beyond 30° . While in normal operational conditions this would not be a major issue as the blade would be unlikely to ever see an angle of attack that high, at the low TSRs seen in the experimental tests, the angle of attack on the turbine blade risks exceeding this 30° validation threshold. The angle of attack on the turbine blade can be roughly defined as:

$$\alpha = \tan^{-1} \left(\frac{U_{wind}}{U_{rot}} \right) \quad (6.19)$$

Here α is the angle of attack, U_{wind} is the wind speed, and U_{rot} is the rotation speed of the blade which is equal to the product of the distance from the hub and the rotor speed. Since the wind speed is relatively constant along the length of the blade, while the rotation speed is not, this means that the angle of attack will be closest to zero at the tip where it will be equal to:

$$\alpha = \tan^{-1} \left(\frac{1}{\text{TSR}} \right) \quad (6.20)$$

Here the angle of attack is the inverse tangent of the inverse of the TSR, which is also shown visually in Figure 6.14. This relationship carries forward logically: as the TSR increases for a given wind speed, so too does the rotor speed, thus the rotation component of the relative wind velocity on the blade will grow and the angle of attack will approach zero. This means that the TSR can be used to directly calculate the lowest angle of attack seen on the blade (ignoring airfoil twist which is very low at the tip). Based on Equation 6.20, with a TSR of 1.0 – which is close to the value specified in many of the experimental tests here – the angle of attack at the tip will be equal to 45° . This exceeds the validation angle of the surrogate thrust model and thus the accuracy of the thrust prediction can not be entirely validated. This limitation mainly affects the surrogate thrust model developed in the following section, not the structural model which is the main goal of this study. However, it does mean that when validating the structural model using this surrogate thrust model at lower TSRs, the thrust model is a likely source of any identified error.

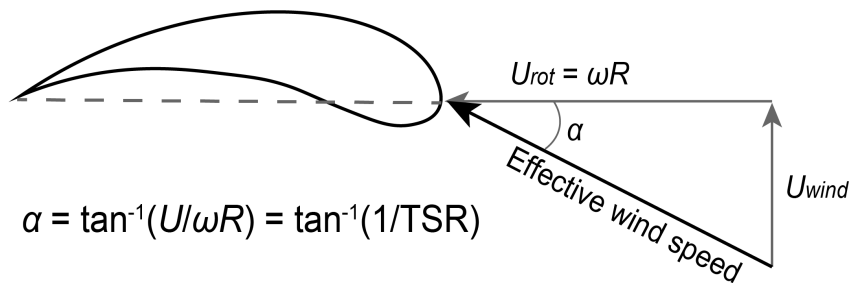


Figure 6.14: Relationship between the angle of attack and TSR at the blade tip

6.3.3 Surrogate thrust coefficient model

The thrust coefficient is a critical value for the structural design of wind turbines. The thrust is the along-wind force experienced by a turbine perpendicular to its rotor plane. While the torque experienced by a rotor is critical for the power production of the turbine, the thrust is critical for the tower design. Much like the other non-dimensional coefficients discussed previously, the thrust coefficient is calculated as:

$$C_T = \frac{T}{0.5\rho A_T U^2} \quad (6.21)$$

Here, C_T is the thrust coefficient, T is the recorded thrust, ρ is the air density, U is the hub-height wind speed, and A_T is the swept area of the rotor which is equal to πR^2 where R is the blade length.

For turbines where the blade pitch is fixed, such as the 1.1 m residential turbine studied here, the lift of the blades' airfoil will primarily govern the rotor thrust. Since the thrust coefficient is normalized by the wind velocity, for an operational turbine with negligible nacelle displacement under typical conditions the only factor that will affect the thrust coefficient is the tip speed ratio (TSR). However, should the nacelle experience notable rotational displacement due to tower deflections, this will also impact the thrust coefficients and in turn, the thrust loading on the tower that is responsible for the tower deflections – as a result there is a non-linear coupling between the wind speed and the thrust coefficient. Consider Figure 6.15 which compares plots of the thrust coefficient as a function of the wind speed and tip ratio for a fixed-base turbine and a rotating-base turbine. It is clearly shown that in the fixed case the thrust coefficient is independent of the wind speed but dependent on the TSR, while in the rotating case it is dependent on the TSR as well as the wind speed which controls the rotor deflection. The final plot shows the relationship between thrust coefficient and base rotation.

The values reported in Figure 6.15 were generated for the 1.1 m residential turbine with a rotating base using OpenFAST. A rigid turbine tower with a linearly rotating base can be

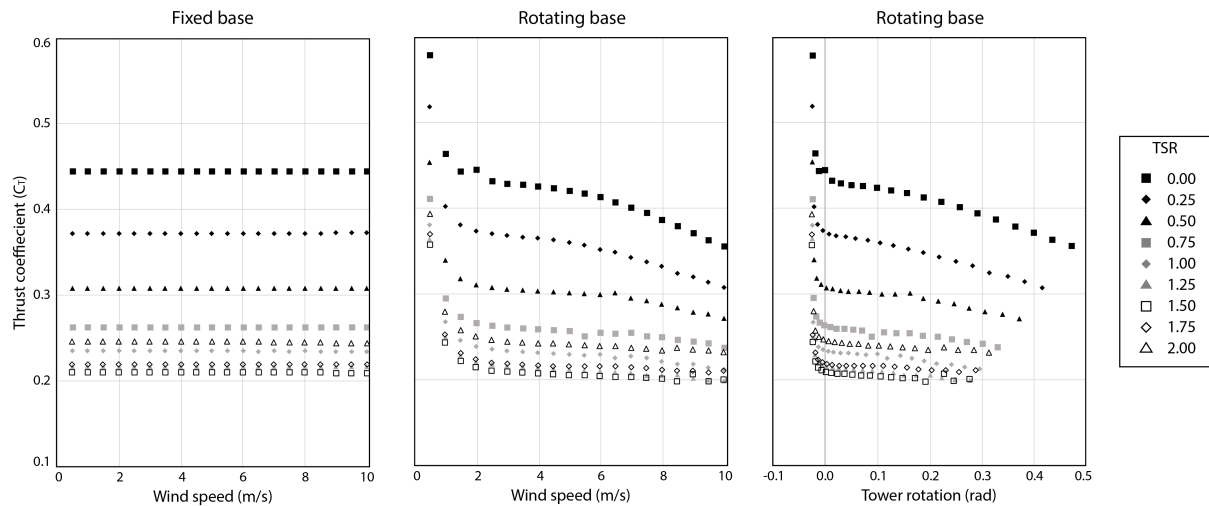


Figure 6.15: Plots of the thrust coefficient at various TSRs and wind speeds for fixed-base and rotating-base turbines

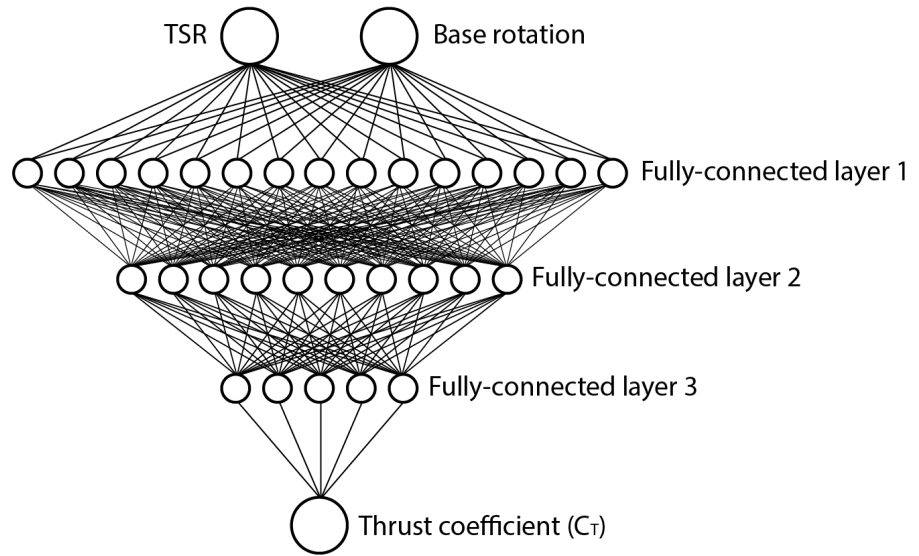


Figure 6.16: Diagram of the architecture of the surrogate thrust MLP NN

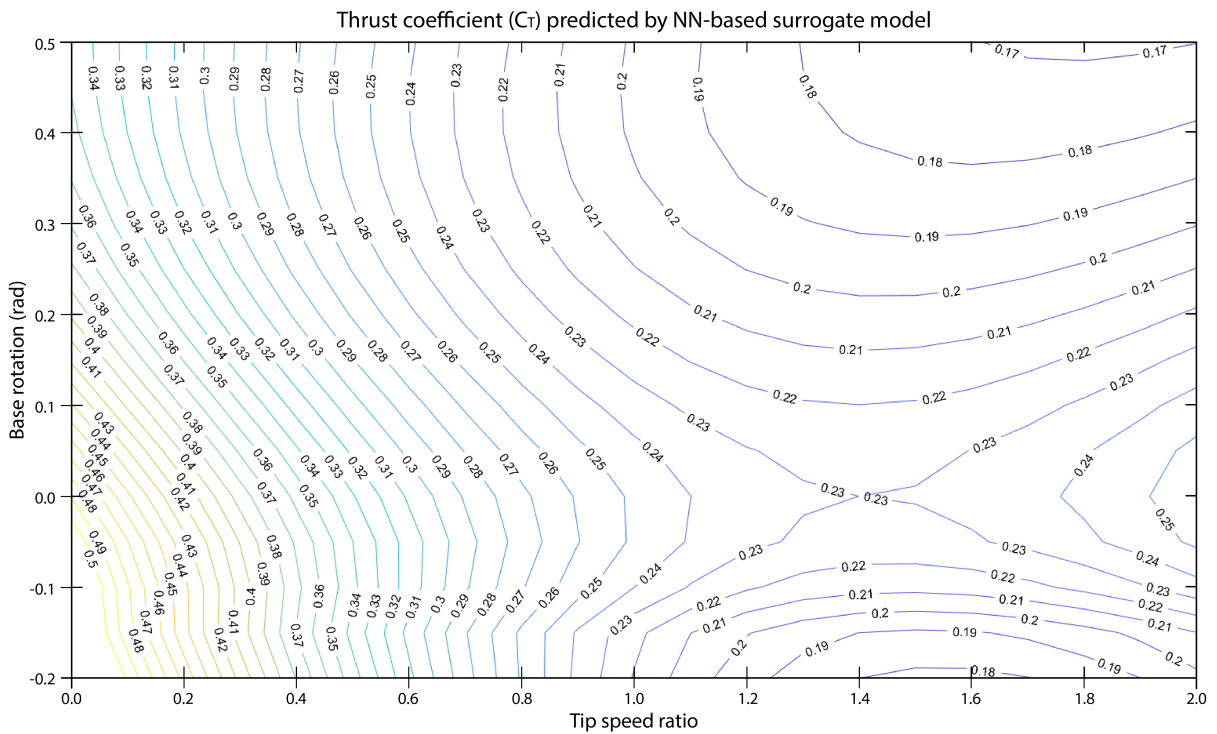


Figure 6.17: Contour plot of the thrust coefficient predicted by the NN-based surrogate model for a given TSR and base rotation

modelled in OpenFAST by restricting the tower deflections in the ElastoDyn module and importing a custom turbine platform into the SubDyn module. The turbine was assigned the rotational inertia and rotational stiffness derived in Section 6.3.1. By adjusting the rotational inertia and testing the turbine under a number of wind speeds and TSRs, then extracting the resulting thrust, the thrust coefficient for these varied conditions can be tabulated. Steady-state simulations were performed in OpenFAST at TSRs of 0 through 2 in increments of 0.25, and at wind speeds of 0.5 to 10 m/s in increments of 0.5 m/s. As seen in Figure 6.15, the relationship between the thrust coefficient, TSR, and base rotation is complex, and a numerical model can not be easily synthesized directly from the collected data.

Instead, the technique presented in Chapter 4 of generating a surrogate aerodynamic model using a trained neural network (NN) was employed here as well. This network, shown in Figure 6.16, was much simpler than the surrogate blade models developed in Chapter 4. It received two inputs – the TSR and the base rotation – and predicted one output – the thrust coefficient. Due to the simplicity, a three-layer multi-layer perceptron (MLP) with layer sizes of 15, 10, and 5 was trained on the 360 OpenFAST samples. In MATLAB, samples were normalized and randomized into the training data set (80% of samples) and testing data set (20%). Using the *trainNetwork* function, the MLP was trained on the training data set for 20,000 epochs with a learning rate of 0.01, resulting in a network with an NRMSE of 3.31% when evaluated using the testing data set. This accuracy of >96% was deemed sufficient for the planned use of this network in this study.

Figure 6.17 shows the resulting contour plot of the thrust coefficient as a function of the TSR and the rotation angle, plotted for a range of TSRs of 0 to 2 and -0.2 to +0.5 rad base rotation, which was generated using the trained neural network.

6.3.4 1-DOF dynamic structural model

Using a combination of basic structural and wind analysis, a simplified 1-DOF model can be developed to capture the structural dynamics of the rotating-base residential wind turbine. Figure 6.18 shows the 1-DOF model used to generate the structural equation of motion. Note the frame of reference that defines downwind rotation as positive and upwind rotation as negative, which will be used throughout the following chapters.

The 1-DOF model can be represented in equation form consisting of two sides of the equation of motion:

$$m_{rot}\ddot{\theta} + M_C(\dot{\theta}) + k_{rot}\theta = -mgd + 0.5\rho A_T C_T U^2 H + 0.5\rho A_t C_t (C_{tU} U)^2 C_{tH} H \quad (6.22)$$

Here, the left side of the equation represents the response of the structure, where θ , $\dot{\theta}$, and $\ddot{\theta}$ are the rotation, the rotational velocity, and the rotational acceleration about the base in rad, rad/s, and rad/s². m_{rot} and k_{rot} are the rotational inertia (kgm²) and the rotational stiffness (Nm/rad), and $M_C(\dot{\theta})$ is the damping moment in the structure which is a function of the rotational velocity; the damping moment requires additional calculations that are detailed in Section 6.3.6. On the right side of the equation are the three sources of moment acting on the turbine: the mass eccentricity moment, the rotor thrust moment, and the tower drag moment. The mass eccentricity moment consists of m , the lumped nacelle mass of 23.9 kg, g , the gravitational acceleration, and d , the eccentric distance between the center of mass of the nacelle and the tower center.

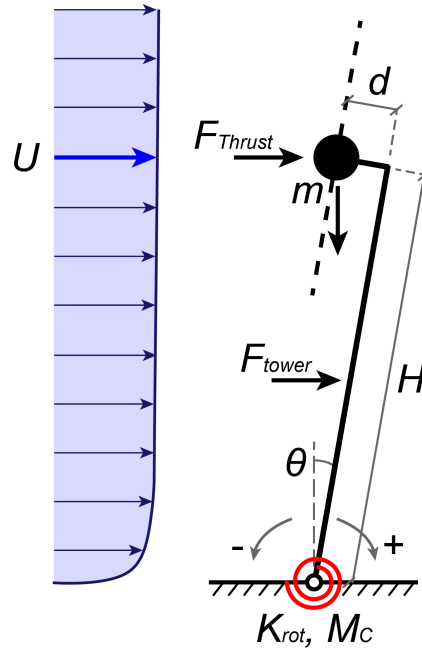


Figure 6.18: Simplified 1-DOF model of the rotating base turbine, with the rotation stiffness and damping shown in red and the wind profile and hub height wind speed in blue. Dashed lines representing the blades are included for visual clarity but are part of the lumped mass in the model

The thrust moment term consists of the thrust (see Equation 6.21) times the hub height (H). The tower drag moment consists of the standard drag equation (see Equation 2.11) times the hub height, where the cross-sectional area (A_t) is equal to the tower diameter (0.08 m) times the tower height (1.9 m), the tower drag coefficient (C_{dt}) is equal to 1.2, the typical subcritical drag value of a cylinder in low-to-mid Reynolds flow. Further, two additional coefficients have been added to account for the non-uniform wind profile along the tower length, C_{tU} is the mean wind speed adjustment factor and C_{tH} is the mean drag load height. Assuming a power law wind profile with a coefficient of 0.05, basic calculus can be used to derive $C_{tU}=0.953$ and $C_{tH}=0.517$.

The small-angle approximation is applied liberally in this model, which states that $\cos\theta \approx 1$ (applied in the loading terms) and $\sin\theta \approx \theta$ (applied in the structural terms). Provided that the structure does not rotate beyond approximately ± 0.4 rad, this simplification should have no impact on the accuracy of the structural model.

6.3.5 Finalization and validation of the structural model under fixed and non-fluctuating conditions

The remaining properties of the 1-DOF structural model, when paired with the NN-based surrogate thrust model, can be determined and validated using experimental results. Using data from Shirazadeh et al. (2021), this turbine model can be validated for the case of a fixed base. Likewise, using the results from the experimental tests described in Section 6.2, the non-fluctuating

response of the rotating-base turbine under mean wind loads can be validated.

Shirazadeh et al. (2021) previously tested the 1.1 m wind turbine under uniform loads of roughly 4.91 m/s at a TSR of 1.30. While this TSR value is lower than the residential turbine's rated value of 5.2, the maximum rotor speed was kept at this lower value to prevent any possible damage to the turbine or the testing instruments. The turbine was fixed to a force balance and the mean along-wind base shear of 14.36 N and base moment of 20.51 Nm were extracted.

Firstly, the surrogate thrust model from Section 6.3.3 can be validated by applying the 1-DOF model to the recorded base shear:

$$V_{b,exp} = F_T + F_t \quad (6.23)$$

$$F_T = 0.5\rho A_T C_T(\text{TSR}, \theta) U^2 = 0.5(1.225)(\pi 1.1^2)(0.231)(4.91^2) = 12.97 \text{ N} \quad (6.24)$$

$$F_t = 0.5\rho A_t C_t C_{tU}^2 U^2 = 0.5(1.225)(1.9 * 0.08)(1.2)(0.953 * 4.91)^2 = 2.86 \text{ N} \quad (6.25)$$

$$V_{b,exp} = 14.36 \text{ N} \quad (6.26)$$

$$F_T + F_t = 15.83 \text{ N} \quad (6.27)$$

Here, the experimental base shear ($V_{b,exp}$) is the sum of the thrust force (F_T) and the tower drag force (F_t). All other variables have been defined previously (see Equation 6.22), but note that $C_T(\text{TSR}, \theta)$ refers to the thrust coefficient predicted by the surrogate model from Section 6.3.3 for a given value of TSR and θ , which returns a coefficient of 0.231 when supplied with a TSR of 1.30 and a base rotation angle of 0 rad. The predicted base shear of the turbine using the structural model with the surrogate thrust model is equal to 15.83 N, which matches decently well with the recorded experimental value of 14.36 N with an error of 10.2%. Thus, it can be said that the numerical model appears valid for the fixed turbine in conditions similar to those described here.

The experimental data from Shirazadeh et al. (2021) also allows the eccentric distance of the nacelle to be calculated by applying the 1-DOF model to the recorded overturning moment:

$$M_{b,exp} = -mgd + M_T + M_t \quad (6.28)$$

$$M_T = 0.5\rho A_T C_T(\text{TSR}, \theta) U^2 H \quad (6.29)$$

$$M_T = 0.5(1.225)(\pi 1.1^2)(0.201)(4.91^2)(1.9) = 24.64 \text{ Nm} \quad (6.30)$$

$$M_t = 0.5\rho A_t C_t C_{tU}^2 U^2 C_{tH} H \quad (6.31)$$

$$M_t = 0.5(1.225)(1.9 * 0.08)(1.2)(0.953 * 4.91)^2(0.517)(1.9) = 2.80 \text{ Nm} \quad (6.32)$$

$$20.51 = -23.9(9.81)d + 21.49 + 2.80 \quad (6.33)$$

$$d = 0.0296 \text{ m} \quad (6.34)$$

Here, the experimental base moment ($M_{b,exp}$) is the sum of the mass eccentricity moment, the thrust moment (M_T), and the tower drag moment (M_t). As the only unknown in this equation is the eccentric nacelle distance (d), this value can be solved as 0.0296 m.

With a validated surrogate thrust model and an identified nacelle eccentricity, the dynamic structural model can be compared to the rotating-base turbine tests described in Section 6.2. As explained previously, it was found that the experimental system was overly-damped for most-to-all test cases, limiting the usefulness of the recorded results. However, the mean results recorded in these tests should be controlled primarily by the structure's rotational stiffness and

thus the over-damping should not strongly impact the mean response. In terms of the model, this condition is captured by removing the time-dependent components of Equation 6.22:

$$k_{rot}\theta = -mgd + 0.5\rho A_T C_T \bar{U}^2 H + 0.5\rho A_t C_t C_{tH}^2 \bar{U}^2 C_{tH} H \quad (6.35)$$

Here, the inertial and damping terms have been removed, and \bar{U} is the mean wind speed. Using this equation, the turbine's mean overturning moment and resulting mean base rotation can be determined for any combination of mean wind speed and TSR. Since the thrust coefficient is used to calculate the base rotation but is also a function of the base rotation, this equation must be solved iteratively: an initial value is assigned for the base rotation which is then used to calculate a new thrust coefficient, then a new base rotation, etc. This iterative equation tends to converge very quickly, typically only requiring five or fewer iterations.

As described in Section 6.2, two rounds of wind tunnel testing were performed. The structural model is compared to both rounds, including the non-zeroed first round and higher-accuracy second round. Beginning with the data collected in the first round of testing: due to the failure to capture a correct zero during testing, the first round can not be directly compared to the structural model on a test-by-test basis. However, the relative displacement between tests can be directly compared to the numerical. In other words, while the total displacement of a given test is unknown, the difference in base rotation between a test at $\bar{U}=3.59$ m/s and TSR=1.50 and a test at $\bar{U}=4.02$ m/s and TSR=1.86 is known. In the structural model, this can be modelled as:

$$k_{rot}(\theta_1 - \theta_2) = 0.5\rho A_T H (C_{T1} \bar{U}_1^2 - C_{T2} \bar{U}_2^2) + 0.5\rho A_t C_t C_{tH} H C_{tH}^2 (\bar{U}_1^2 - \bar{U}_2^2) \quad (6.36)$$

Here, subscripts 1 and 2 refer to the two tests being compared. Table 6.6 lists the measured wind speed and TSR of the six tests from the first round of testing with the 9.53 mm aperture pneumatic damper (the smallest amount of damping), as well as the resulting response calculated using the numerical structural model with the surrogate thrust model. Table 6.7 then compares the difference in predicted base rotations between each combination of tests, for both the experimental and numerical results. In Table 6.7, $\Delta\theta_{exp}$ is the difference in base rotation between the two listed experimental tests, $\Delta\theta_{num}$ is the difference in base rotation between the numerical simulations of the same cases, and ΔM is the equivalent moment that would

Variable	Symbol	Unit	Test 1-1	Test 1-2	Test 1-3	Test 1-4	Test 1-5	Test 1-6
Mean wind speed	U	m/s	3.64	3.59	3.59	4.09	4.02	4.85
Tip speed ratio	TSR	-	0.90	1.50	2.10	1.10	1.87	1.45
Rotor speed	Ω	rad/s	2.98	4.90	6.85	4.09	6.83	6.39
Approximate rotational stiffness	k_{rot}	rpm	28.4	46.7	65.4	39.1	65.3	61.1
Non-zeroed mean experimental rotation	θ_{exp}	rad	0.0015	0.0033	0.0038	0.0021	0.0040	0.0045
Tower drag moment	M_t	Nm	2.43	2.37	2.37	3.07	2.97	4.32
Thrust coefficient	C_T	-	0.256	0.230	0.261	0.238	0.245	0.227
Thrust moment	M_T	Nm	14.99	13.12	14.91	17.58	17.53	23.64
Mass eccentricity moment	$-mgd$	Nm	-6.94	-6.94	-6.94	-6.94	-6.94	-6.94
Predicted base moment	$M_{b,num}$	Nm	9.37	7.46	9.25	12.30	12.19	19.04
Predicted base rotation	θ_{num}	rad	0.0083	0.0066	0.0082	0.0109	0.0108	0.0169

Table 6.6: Comparison of non-fluctuating response between the first round of wind tunnel tests and the 1-DOF structural model

	Vs Test 1-2		Vs Test 1-3		Vs Test 1-4		Vs Test 1-5		Vs Test 1-6	
Test 1-1	$\Delta\theta_{exp}$ (rad)	0.0018	$\Delta\theta_{exp}$ (rad)	0.0022	$\Delta\theta_{exp}$ (rad)	0.0006	$\Delta\theta_{exp}$ (rad)	0.0025	$\Delta\theta_{exp}$ (rad)	0.0030
	$\Delta\theta_{num}$ (rad)	0.0017	$\Delta\theta_{num}$ (rad)	0.0001	$\Delta\theta_{num}$ (rad)	0.0026	$\Delta\theta_{num}$ (rad)	0.0025	$\Delta\theta_{num}$ (rad)	0.0086
	$\Delta\theta_{exp} - \Delta\theta_{num}$	0.0001	$\Delta\theta_{exp} - \Delta\theta_{num}$	0.0021	$\Delta\theta_{exp} - \Delta\theta_{num}$	0.0020	$\Delta\theta_{exp} - \Delta\theta_{num}$	0.0000	$\Delta\theta_{exp} - \Delta\theta_{num}$	0.0056
	ΔM (Nm)	0.11	ΔM (Nm)	2.41	ΔM (Nm)	2.31	ΔM (Nm)	0.03	ΔM (Nm)	6.33
	$\Delta M/M_T$	1%	$\Delta M/M_T$	16%	$\Delta M/M_T$	10%	$\Delta M/M_T$	0%	$\Delta M/M_T$	27%
Test 1-2	$\Delta\theta_{exp}$ (rad)		$\Delta\theta_{exp}$ (rad)	0.0022	$\Delta\theta_{exp}$ (rad)	0.0012	$\Delta\theta_{exp}$ (rad)	0.0007	$\Delta\theta_{exp}$ (rad)	0.0012
	$\Delta\theta_{num}$ (rad)		$\Delta\theta_{num}$ (rad)	0.0016	$\Delta\theta_{num}$ (rad)	0.0043	$\Delta\theta_{num}$ (rad)	0.0042	$\Delta\theta_{num}$ (rad)	0.0103
	$\Delta\theta_{exp} - \Delta\theta_{num}$		$\Delta\theta_{exp} - \Delta\theta_{num}$	0.0007	$\Delta\theta_{exp} - \Delta\theta_{num}$	0.0031	$\Delta\theta_{exp} - \Delta\theta_{num}$	0.0035	$\Delta\theta_{exp} - \Delta\theta_{num}$	0.0091
	ΔM (Nm)		ΔM (Nm)	0.74	ΔM (Nm)	3.45	ΔM (Nm)	3.96	ΔM (Nm)	10.26
	$\Delta M/M_T$		$\Delta M/M_T$	5%	$\Delta M/M_T$	15%	$\Delta M/M_T$	23%	$\Delta M/M_T$	43%
Test 1-3	$\Delta\theta_{exp}$ (rad)		$\Delta\theta_{exp}$ (rad)		$\Delta\theta_{exp}$ (rad)	0.0017	$\Delta\theta_{exp}$ (rad)	0.0002	$\Delta\theta_{exp}$ (rad)	0.0007
	$\Delta\theta_{num}$ (rad)		$\Delta\theta_{num}$ (rad)		$\Delta\theta_{num}$ (rad)	0.0027	$\Delta\theta_{num}$ (rad)	0.0026	$\Delta\theta_{num}$ (rad)	0.0087
	$\Delta\theta_{exp} - \Delta\theta_{num}$		$\Delta\theta_{exp} - \Delta\theta_{num}$		$\Delta\theta_{exp} - \Delta\theta_{num}$	0.0010	$\Delta\theta_{exp} - \Delta\theta_{num}$	0.0024	$\Delta\theta_{exp} - \Delta\theta_{num}$	0.0079
	ΔM (Nm)		ΔM (Nm)		ΔM (Nm)	1.15	ΔM (Nm)	2.68	ΔM (Nm)	8.97
	$\Delta M/M_T$		$\Delta M/M_T$		$\Delta M/M_T$	5%	$\Delta M/M_T$	15%	$\Delta M/M_T$	38%
Test 1-4	$\Delta\theta_{exp}$ (rad)		$\Delta\theta_{exp}$ (rad)		$\Delta\theta_{exp}$ (rad)		$\Delta\theta_{exp}$ (rad)	0.0019	$\Delta\theta_{exp}$ (rad)	0.0024
	$\Delta\theta_{num}$ (rad)		$\Delta\theta_{num}$ (rad)		$\Delta\theta_{num}$ (rad)		$\Delta\theta_{num}$ (rad)	0.0001	$\Delta\theta_{num}$ (rad)	0.0060
	$\Delta\theta_{exp} - \Delta\theta_{num}$		$\Delta\theta_{exp} - \Delta\theta_{num}$		$\Delta\theta_{exp} - \Delta\theta_{num}$		$\Delta\theta_{exp} - \Delta\theta_{num}$	0.0018	$\Delta\theta_{exp} - \Delta\theta_{num}$	0.0036
	ΔM (Nm)		ΔM (Nm)		ΔM (Nm)		ΔM (Nm)	2.07	ΔM (Nm)	4.02
	$\Delta M/M_T$		$\Delta M/M_T$		$\Delta M/M_T$		$\Delta M/M_T$	12%	$\Delta M/M_T$	17%
Test 1-5	$\Delta\theta_{exp}$ (rad)		$\Delta\theta_{exp}$ (rad)		$\Delta\theta_{exp}$ (rad)		$\Delta\theta_{exp}$ (rad)		$\Delta\theta_{exp}$ (rad)	0.0005
	$\Delta\theta_{num}$ (rad)		$\Delta\theta_{num}$ (rad)		$\Delta\theta_{num}$ (rad)		$\Delta\theta_{num}$ (rad)		$\Delta\theta_{num}$ (rad)	0.0061
	$\Delta\theta_{exp} - \Delta\theta_{num}$		$\Delta\theta_{exp} - \Delta\theta_{num}$		$\Delta\theta_{exp} - \Delta\theta_{num}$		$\Delta\theta_{exp} - \Delta\theta_{num}$		$\Delta\theta_{exp} - \Delta\theta_{num}$	0.0056
	ΔM (Nm)		ΔM (Nm)		ΔM (Nm)		ΔM (Nm)		ΔM (Nm)	6.30
	$\Delta M/M_T$		$\Delta M/M_T$		$\Delta M/M_T$		$\Delta M/M_T$		$\Delta M/M_T$	27%

Table 6.7: Difference in rotations in pairs of first round of wind tunnel testing listed in Table 6.6 compared to the equivalent pair of differences in numerical predictions

cause the difference between $\Delta\theta_{exp}$ and $\Delta\theta_{num}$. In an attempt to contextualize the value of these discrepancies, ΔM is compared to the thrust moment predicted by the second numerical simulation in the comparison, the ratio of which is reported as $\Delta M/M_T$. Assuming that any major discrepancies in the numerical prediction stem from error in the predicted thrust coefficient (see Section 6.3.2), $\Delta M/M_T$ thus quantifies the error in the thrust prediction of the numerical simulations. As can be seen in Table 6.7, the “thrust error” ($\Delta M/M_T$) is generally below 20% for most cases, though comparisons with Test 1-6 typically lie outside of this range. The error is likely due to the limitations in the thrust model (as assumed), any errors in predicting the thrust coefficient would scale equal to the wind speed squared, thus it would be expected that error would be greatest in the test case with the highest wind speed. Still, given the limitations within the data captured in the first round of wind tunnel testing as well as in the surrogate thrust model, the overall matching between the experimental results and the results predicted by the structural model are satisfactory.

While the results from the first round of wind tunnel testing are positive indicators of the accuracy of the structural model, the results from the second round of testing are of higher accuracy and worth greater consideration. These comparisons are also more straightforward; Table 6.8 shows the conditions and results of four tests from the second round of testing, labeled 2-1 through 2-4, which vary by mean wind speed, rotational stiffness, and TSR. The table also includes the predicted mean base rotation according to the structural model and compares the predicted results by using a typical error equation with the denominator as a function of the static rotation (θ_{static}) of 0.00615 rad or 0.0190 rad for the full- and half-stiffness cases, respectively:

$$\text{error} = \frac{\theta_{exp} - \theta_{num}}{\theta_{exp} - \theta_{static}} \quad (6.37)$$

As can be seen in Table 6.8, matching is decent between the experimental and numerical results, with the largest error of -17% occurring for Test 2-4. This error is greater than ideal, and

Variable	Symbol	Unit	Test 2-1	Test 2-2	Test 2-3	Test 2-4
Mean wind speed	U	m/s	4.07	3.63	4.07	3.63
Tip speed ratio	TSR	–	1.32	0.95	1.33	0.96
Rotor speed	ω	rad/s	4.88	3.14	4.92	3.17
Approximate rotational stiffness	k_{rot}	Nm/rad	1129	1129	366	366
Mean experimental base rotation	θ_{exp}	rad	0.0111	0.0060	0.0401	0.0215
Tower drag moment	M_t	Nm	1.65	1.31	1.65	1.31
Thrust coefficient	C_T	–	0.231	0.252	0.229	0.251
Thrust moment	M_T	Nm	16.93	14.71	16.78	14.64
Mass eccentricity moment	$-mgd$	Nm	-6.94	-6.94	-6.94	-6.94
Predicted base moment	$M_{b,num}$	Nm	11.64	9.08	11.49	9.01
Predicted base rotation	θ_{num}	rad	0.0103	0.0080	0.0314	0.0246
Base rotation error	–	–	4.7%	-16.6%	14.7%	-7.73%

Table 6.8: Comparison of non-fluctuating response between the second round of wind tunnel tests and the 1-DOF structural model

likely stems from limitations in the OpenFAST-generated data used to train the surrogate thrust model. This is supported by the fact that the model underestimates rotation at high wind speed/TSR and overestimates at lower wind speed/TSR. If the error were due to structural properties, it would be expected that the error would be consistently over- or underestimated. Conversely, error in predicting the thrust coefficient could over- or underestimate depending on the wind speed/TSR combination. Ultimately, the OpenFAST-generated thrust predictions were fairly accurate for the upright rotor, but the rotation of the rotor plane is likely injecting inaccuracies. Recall however, that the NN used to predict the thrust coefficient will not be carried forward into testing beyond this chapter, only the structural model will. Therefore, based on the overall accuracy between the non-fluctuating structural model and both rounds of experimental tests, the developed structural model is deemed sufficiently accurate to carry forward onto true dynamic analyses.

6.3.6 Damping model and dynamic validation

In a typical structure, the vibration damping that occurs can be approximated as a single linear term:

$$C_{lin} = 2m\omega\xi_{lin} \quad (6.38)$$

$$M_C = C_{lin}\dot{\theta} \quad (6.39)$$

Where C_{lin} is the linear damping coefficient, m is the modal mass, ω is the natural frequency of the structure, and ξ_{lin} is the linear damping ratio. Using this linear model, the damping moment (M_C) from Equation 6.22 is the product of the damping coefficient and the structure's rotational velocity ($\dot{\theta}$). While in reality the damping in a structure comes from a combination of linear and non-linear terms, the purely-linear approximation is typically sufficiently accurate for the purposes of structural analysis, particularly because it is often difficult to identify the damping ratio in a real structure. The most common technique to do so is the logarithmic decrement technique which uses the free vibration response to estimate the linear damping ratio in a structure. The full derivation can be found in Newth (2004), but to summarize: it can be derived from the standard equation of motion using Equation 6.38 that when a structure with a damping ratio $\ll 1$ is in free vibration, the decay curve plotted along the amplitudes of the free vibration oscillations (a) can be modelled as:

$$a = a_0 e^{-\omega\xi_{lin}t} \quad (6.40)$$

$$\frac{a_r}{a_{r-1}} = e^{-2\pi\xi_{lin}} \quad (6.41)$$

$$\xi_{lin} = \frac{\ln(a_{r-1}/a_r)}{2\pi} \quad (6.42)$$

Therefore, the decay curve of the oscillations is defined as the initial amplitude (a_0) times a function of the natural frequency (ω), the damping ratio (ξ_{lin}), and the time (t). This equation can be used to determine the structural damping ratio based on the ratio between subsequent amplitudes of oscillations a_r/a_{r-1} where a_r refers to the amplitude at $t = 2\pi r/\omega$. Thus Equation 6.42 can be used for each subsequent pair of amplitudes to estimate the damping ratio, the average of which across the entire free vibration time history can be used for the structure.

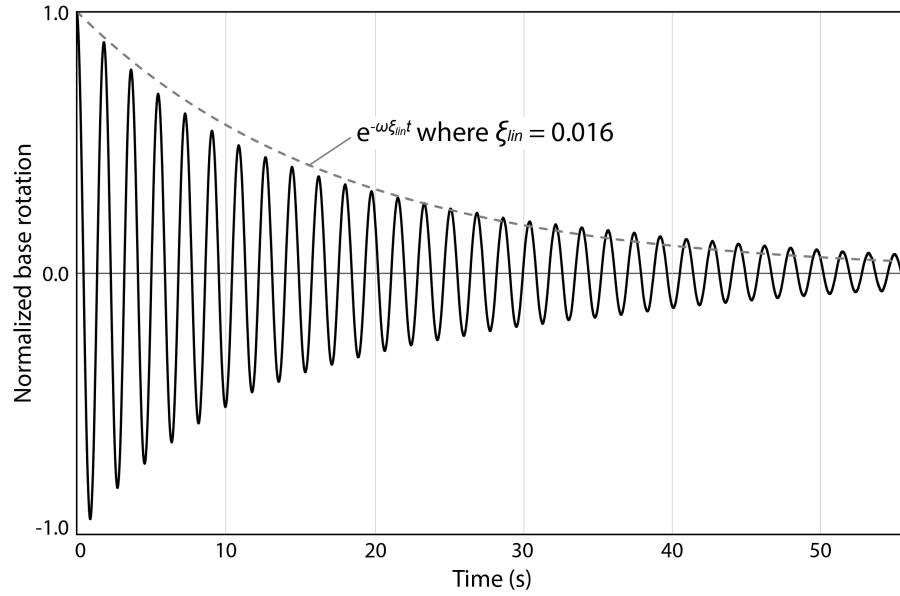


Figure 6.19: Normalized free vibration response of the full-stiffness rotating-base turbine and the decay curve predicted by the linear damping model assuming a damping ratio of 1.6%

Though, if the structure is closely following a linear damping response, there should be little variation in the predicted ratio from amplitude to amplitude. Figure 6.19 compares this linear model to the recorded free vibration results of the rotating-base turbine from the wind tunnel testing; it can be seen that the linear model fit is imperfect, but matches decently well at a damping ratio of 1.6%.

In industrial wind turbines, it is well documented that while there is a passive structural damping effect, there is a much larger damping effect in the fore-aft direction due to the aerodynamic damping caused by the operational rotor. As a rule-of-thumb, an industrial turbine will typically have a passive structural damping ratio of 0.5-1%, but with an additional 4-10% aerodynamic damping ratio. This effect is quite significant in industrial turbines, thus its impact must be considered for this residential turbine – recall that the rotor is not operational during the free vibration tests and thus the predicted structural damping ratio does not include this aerodynamic effect. The numerical Kühn model (Kühn, 2001) can be used to estimate the aerodynamic damping ratio of an operational wind turbine. The Kühn model states:

$$\xi_{aero} = \frac{N_b \rho \Omega}{4\omega m_{rot}} \int_{r_{root}}^R \frac{dC_l}{d\alpha} c r dr \quad (6.43)$$

Here, ξ_{aero} is the aerodynamic damping ratio, N_b is the number of blades, ρ is the air density, Ω is the rotor speed in rad/s, ω is the natural frequency of the turbine in rad/s, and m_{rot} is the rotational inertia of the turbine in kgm^2 . The final term is the integral of the linear pre-stall slope of the lift coefficient curve ($\frac{dC_l}{d\alpha}$) in rad^{-1} , multiplied by the chord length c in m, and times the radial distance from the hub r in m, ranging from the base of the blade (r_{root}) to the tip (R). Solving the integral yields:

Section	Airfoil	Radius (m)	Chord length (m)	C_L slope	Kühn variable
1	NACA 6515	0.19	0.1220	5.1700	0.0068
2	NACA 6515	0.24	0.1180	5.1700	0.0081
3	NACA 6515	0.29	0.1140	5.1700	0.0093
4	NACA 6515	0.34	0.1100	5.1700	0.0104
5	NACA 6515	0.39	0.1060	5.1700	0.0114
6	NACA 6515	0.44	0.1020	5.1700	0.0123
7	NACA 6515	0.49	0.0980	5.1700	0.0130
8	NACA 6515	0.54	0.0940	5.1700	0.0137
9	NACA 6515	0.59	0.0900	5.1700	0.0143
10	NACA 6515	0.64	0.0860	5.1700	0.0148
11	FX 63 137	0.69	0.0820	5.5558	0.0163
12	FX 63 137	0.74	0.0780	5.5558	0.0166
13	FX 63 137	0.79	0.0740	5.5558	0.0168
14	FX 63 137	0.84	0.0700	5.5558	0.0168
15	FX 63 137	0.89	0.0660	5.5558	0.0168
16	FX 63 137	0.94	0.0620	5.5558	0.0166
17	FX 63 137	0.99	0.0580	5.5558	0.0164
18	FX 63 137	1.04	0.0540	5.5558	0.0160
19	FX 63 137	1.09	0.0500	5.5558	0.0030
20	FX 63 137	1.10	0.0375	–	–
Sum of Kühn variables:					0.2492
Predicted aerodynamic damping ratio:					0.0036

Table 6.9: Calculation of numerically predicted aerodynamic damping for the 1.1 m residential wind turbine using Equation 6.43

Section	Airfoil	Radius (m)	Chord length (m)	C_L slope	Kühn variable
1	Cylinder1	2.8667	3.542	0.00	0
2	Cylinder1	5.6	3.854	0.00	0
3	Cylinder2	8.3333	4.167	0.00	0
4	DU40 A17	11.75	4.557	7.92	2043
5	DU35 A17	15.85	4.652	7.46	2547
6	DU35 A17	19.95	4.458	7.46	3000
7	DU30 A17	24.05	4.249	7.49	3405
8	DU25 A17	28.15	4.007	6.85	3397
9	DU25 A17	32.25	3.748	6.85	3609
10	DU21 A17	36.35	3.502	6.41	3532
11	DU21 A17	40.45	3.256	6.41	3634
12	NACA64 A17	44.55	3.01	6.31	3631
13	NACA64 A17	48.65	2.764	6.31	3628
14	NACA64 A17	52.75	2.518	6.31	2958
15	NACA64 A17	56.1667	2.313	6.31	2297
16	NACA64 A17	58.9	2.086	6.31	2170
17	NACA64 A17	61.6333	1.419	–	–
Sum of Kühn variables:					39850
Predicted aerodynamic damping ratio:					0.0710

Table 6.10: Calculation of numerically predicted aerodynamic damping for the NREL 5 MW industrial wind turbine using Equation 6.43

$$\int_{r_{root}}^R \frac{dC_l}{d\alpha} cr dr = 0.5 \sum_{i=1}^n \frac{dC_l}{d\alpha} c_i (r_{ui}^2 - r_{li}^2) \quad (6.44)$$

Here, the blade is divided into i segments of consistent airfoil shape and chord length. $\frac{dC_l}{d\alpha}$ is again the linear lift coefficient curve slope, c_i is the chord length of the blade section, and r_{ui} and r_{li} are the upper and lower radial distance of the segment respectively.

Using the Kühn model, the predicted aeroelastic damping ratio for the 1.1 m residential turbine can be estimated using a rotor speed of 4.88 rad/s which results in a TSR of 1.32 in 4.07 m/s wind. Table 6.9 shows the calculation process, separating the turbine blade into 20 segments and solving ξ_{aero} as 0.36%. As could be expected intuitively, due to the smaller rotor size and the relatively low rotor speed, there is not a significant amount of aerodynamic damping predicted. However, given that the predicted structural damping ratio is around 1.6%, this value would still impact the structural response, representing approximately a 25% increase in damping.

To confirm that the Kühn model works as expected, it was also used to solve for the aerodynamic damping of the NREL 5 MW industrial wind turbine (Jonkman et al., 2009). This turbine has three 62 m blades, with an operational rotor speed of 1.27 rad/s, a first modal frequency of 1.83 rad/s, and a first modal mass of approximately 360 tonnes. Table 6.10 shows the calculation process by dividing the blade into 17 segments, resulting in a predicted aerodynamic damping ratio of 7.1%. This value falls into the expected value range of 4-10%, supporting the effectiveness of the Kühn model and the relative accuracy of the predicted damping ratio for the 1.1 m turbine.

This damping model must also be validated against the wind tunnel tests. Here, the fluctuations in the rotation of the turbine tower are captured as the standard deviation of the rotation response. The stiffness of the system controls the mean rotation, but the vibration about the mean is controlled primarily by the damping of the system. Thus, if the dynamic structural model achieves a similar predicted standard deviation, it can be said that the damping model is capturing the behaviour effectively. To this end, numerical simulations of the structural model combined with the surrogate thrust model were performed for both the full- and half-stiffness cases under the recorded 3.63 m/s and 4.07 m/s wind speed time histories, and the standard deviations of the results were extracted. An optimization process was then used to determine the damping ratio required to match the recorded experimental standard deviation. The results of this process are reported in Table 6.11. Recall that since over-damping was an issue seen in many of the experimental tests, the recorded standard deviations of the tower rotation are relatively small. As can be seen, infeasibly high damping ratios ranging from 300-1500% are required to match the experimental results, compared to the values of 1.6% predicted by the free vibration tests. This difference can not be accounted for by the small amount of aerodynamic damping, which was predicted to be in the order of <1%. These results show that the purely-linear damping model is unable to truly capture the recorded behaviour of this rotating-base turbine.

To expand the damping model to satisfy both the free vibration and wind tunnel results, one needs to consider all the physical sources of damping in the system. This includes damping from the structural response (e.g. small deformations in the frame), damping from air resistance, and damping from friction (e.g. between the tower axle and the frame). These are

Variable	Test 2-1	Test 2-2	Test 2-3	Test 2-4
Mean wind speed (m/s)	4.07	3.64	4.07	3.64
Tip speed ratio	1.32	0.95	1.33	0.96
Rotor speed (rad/s)	4.88	3.14	4.92	3.18
Experimental standard deviation of rotation (rad)	0.00091	0.00078	0.00324	0.00206
Mean base moment (Nm)	18.54	16.11	18.44	15.94
Mean base rotation (rad)	0.0164	0.0143	0.0504	0.0436
Thrust coefficient	0.231	0.252	0.229	0.250
Optimized linear damping ratio	9.15	14.75	3.36	8.97
Numerical standard deviation of rotation (rad)	0.00091	0.00071	0.00320	0.00210

Table 6.11: Optimal linear damping ratios such that the standard deviation of base rotations matches recorded wind tunnel results

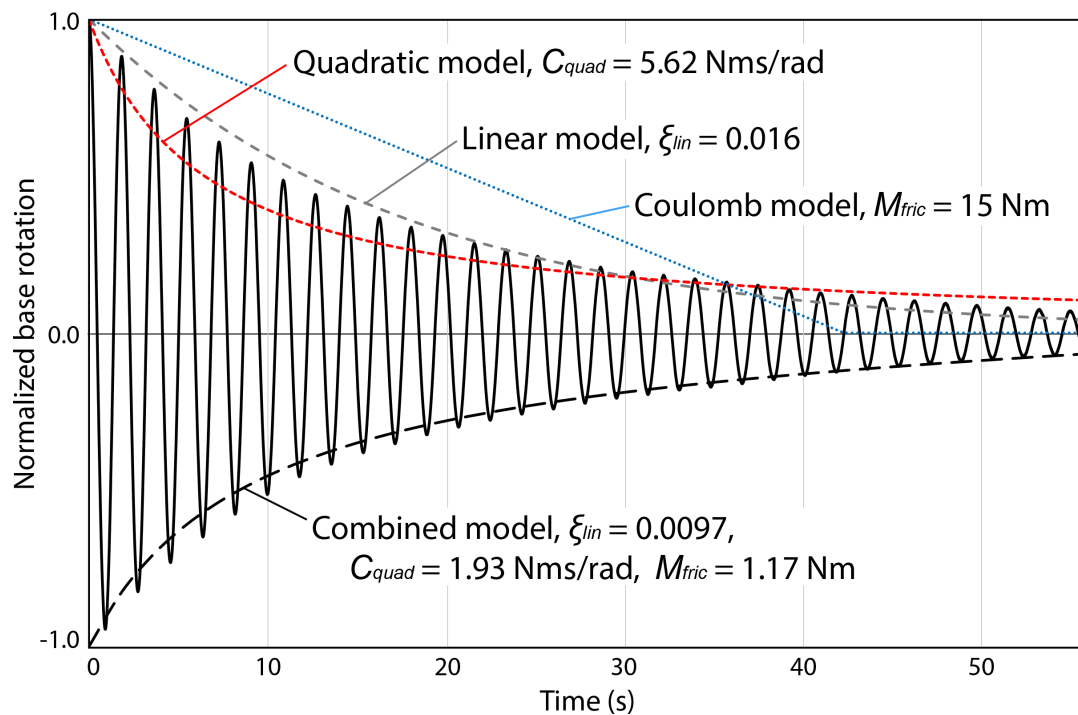


Figure 6.20: Normalized free vibration response of the full-stiffness rotating-base turbine and the decay curves predicted by the linear, quadratic and Coulomb damping models. The optimized decay curve from the combined model is also shown

referred to as structural damping, air damping, and friction damping; each of which can be described using a different damping model, as shown in Figure 6.20. The structural damping can be represented using linear damping, as discussed previously.

The air damping is modelled using quadratic damping:

$$M_C = C_{quad} \dot{\theta} |\dot{\theta}| \quad (6.45)$$

As can be seen, the damping force is equal to the quadratic damping coefficient (C_{quad}) multiplied by the velocity squared, where one of the terms is the absolute value to preserve directionality. Newth (2004) characterized the quadratic damping model, and found the free vibration decay curve to be equal to:

$$\frac{a_r}{a_{r-1}} = \frac{3}{4C_{quad}a_0 + 3} \quad (6.46)$$

Here, a_r/a_{r-1} is the ratio of adjacent amplitudes as previously defined in Equation 6.41. Quadratic damping is the appropriate model for air drag because, as defined in Equation 2.11, the aerodynamic drag force is a function of the wind velocity squared. As one would expect due to the squared term, it is seen in Figure 6.20 that the quadratic model dampens vibrations more strongly at higher amplitudes and less so at lower amplitudes.

The friction damping can be modelled using Coulomb damping:

$$M_C = M_{fric} * \text{sign}(\dot{\theta}) \quad (6.47)$$

Here, the damping moment models kinetic friction and is equal to a fixed frictional moment (M_{fric}) that opposes the direction of structural motion. The sign function is equal to either 1 or -1 depending on the directionality of $\dot{\theta}$. As a result, this applies a significant amount of damping to small amplitude vibrations while having limited effectiveness on high amplitude vibrations. As described by Rizcallah (2019), the decay curve associated with Coulomb damping is:

$$a = a_0 - \frac{M_{fric}\omega}{2k_{rot}}t \quad (6.48)$$

Where k_{rot} is the structural stiffness. As can be seen in Figure 6.20, the Coulomb damping model results in a linear reduction in amplitude until oscillations end; of the three damping models presented here, the Coulomb model is most effective at suppressing low-amplitude vibrations.

The improved damper model for the turbine includes the structural, air, and friction components, thus by combining Equations 6.38, 6.39, 6.45, and 6.47:

$$M_C = 2m\omega\xi_{lin}\dot{\theta} + C_{quad}\dot{\theta}|\dot{\theta}| + M_{fric}\text{sign}(\dot{\theta}) \quad (6.49)$$

In this combined model there are three unknowns that control the overall damping: ξ_{lin} , C_{quad} , and M_{fric} . There is no reasonable way to solve for these values explicitly, instead an optimization process will be used such that the assigned values for the unknowns result in a model that matches both the free vibration and wind tunnel tests. This process was performed in MATLAB using the Bayesian optimization algorithm previously detailed in Section 4.2.3. The full- and half-stiffness structural models combined with the surrogate thrust model are run under wind tunnel and free vibrations conditions, then the resulting response curves are extracted. The optimization process seeks to minimize the average error between the predicted oscillation peaks in the free vibration response and the standard deviation of the base rotation in the wind tunnel response. Note that this process ignores any additional aerodynamic damping added by the operational rotor, which was deemed an acceptable assumption. In addition to this three-term combined model, three two-term models consisting of each possible pair of the three terms

Variable	Test 2-1	Test 2-2	Test 2-3	Test 2-4
Mean wind speed (m/s)	4.07	3.64	4.07	3.64
Tip speed ratio	1.32	0.95	1.33	0.96
Rotor speed (rad/s)	4.88	3.14	4.92	3.18
Experimental standard deviation of rotation (rad)	0.00091	0.00078	0.00324	0.00206
Optimized linear damping ratio	0.0097	0.0078	0.0140	0.0111
Optimized quadratic damping coefficient (Nms ² /rad ²)	1.934	2.618	0.063	1.149
Optimized friction value (Nm)	1.172	0.865	0.831	0.765
Numerical standard deviation of rotation (rad)	0.00091	0.00078	0.00320	0.00200

Table 6.12: Optimal linear damping ratio, quadratic damping coefficient, and friction moment such that the standard deviation of base rotations matches recorded wind tunnel results as well as the free vibration results

were investigated as well, but the three-term model was the most effective and thus the results of these additional tests have been excluded.

Table 6.12 shows the results of this optimization process including the optimized damping parameters and the resulting error for both the free vibration and wind tunnel results. As can be seen, all four cases converge on similar values that result in the numerical model accurately matching both the standard deviation of the tower rotation, as well as the free vibration response as shown in Figure 6.20. As the rotational velocity of the turbine is relatively low in these cases, the quadratic damping term is the least impactful, while the Coulomb damping term is the most critical as it is instrumental in replicating the very small amplitude of fluctuation recorded in the wind tunnel tests. It can be seen while there is decent agreement between the three damping parameters between tests, there is some variation possibly stemming for the optimization process and the different TSRs affecting the aerodynamic damping. However, for the purposes of this study, it would be deemed sufficiently accurate to model the experimental damping as the average values across all four cases, thus $\xi_{lin}=1.07\%$, $C_{quad}=1.44$ Nms²/rad², and $M_{fric}=3.63$ Nm.

While this combined model serves to accurately capture the response of the experimental rotating-base turbine, it carries forward the oppressive amount of damping found in the experimental model. While it was important to showcase that the true damping behaviour of the experimental model could be identified and included in the structural model, for further application of this structural model it would be beneficial to remedy this limitation. Thus, it was decided to simplify the damping of the structural model for future tests. As was shown in Figure 6.19, a linear damping model of 1.6% matches the free vibration results well, and using Equation 6.43 the Kühn model predicted an operational aerodynamic damping ratio of 0.4%. Thus, a total linear damping ratio of 2.0% is chosen to be used in future tests, neglecting any quadratic or friction damping terms, thus allowing for the turbine model to experience larger base deflections in future simulations.

Property	Symbol	Unit	Value
Rotational inertia	m_{rot}	kgm ²	86.32
Rotational stiffness	k_{rot}	Nm/rad	1129 or 366
Natural frequency	ω	rad/s	3.617 or 2.060
Operational damping ratio	ξ	–	0.02
Rotational damping coefficient	c_{rot}	Nms/rad	12.49 or 7.11
Hub height	H	m	1.9
Nacelle mass	m	kg	23.9
Nacelle overhang	d	m	0.0296
Rotor area	A_T	m ²	3.80
Tower area	A_t	m ²	0.152
Tower drag coefficient	C_t	–	1.2
Density of air	ρ	kg/m ³	Per case
Wind speed	U	m/s	Per case
Thrust coefficient	C_T	–	Per case
Mass eccentricity moment	M_e	Nm	$-mgd\cos\theta$
Thrust moment	M_T	Nm	$0.5\rho A_T C_T (U\cos\theta)^2 H$
Tower drag moment	M_t	Nm	$0.5\rho A_t C_t (0.953U\cos\theta)^2 (0.517H)$
Equation of motion			$m_{rot}\ddot{\theta} + c_{rot}\dot{\theta} + k_{rot}\theta = M_e + M_T + M_t$

Table 6.13: Summary of 1-DOF dynamic structural model. Entries where two values are listed refers to the full-stiffness and half-stiffness models respectively

6.3.7 Summary of the 1-DOF structural model

The summarized properties of the 1-DOF dynamic structural model are listed in Table 6.13 including updating the loading terms to remove the small-angle approximation. When combined with the surrogate thrust model detailed in Section 6.21 or some other aerodynamic model, this model can be subjected to any wind speed time history and the resulting tower rotations can be tracked. This time history analysis can be performed using any number of numerical integration techniques, including the Chen-Ricles-based method described in Section 5.2.4.

6.4 Case study employing the structural model

As an example of the utility of the developed dynamic structural model, the following short numerical case study was performed. Since existing HS of wind turbines (see Section 2.7) have primarily been used to study non-linear external damping systems such as tuned mass dampers (TMDs), a numerical test was performed on a similar test case. In this case study, the structural model is paired with the surrogate thrust model from Section 6.3.3 to evaluate the effectiveness of passive and semi-active TMDs at reducing vibrations in this turbine system. This research is heavily inspired by the study presented in Chapter 5 and serves as both a proof-of-concept of the dynamic model and an additional validation of the findings from Lalonde et al. (2020b). This study subjects the full-stiffness dynamic model to a range of wind speeds recorded from

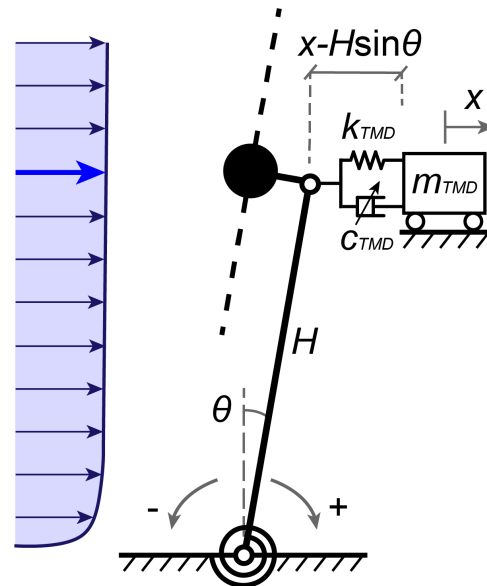


Figure 6.21: Modification of the 1-DOF structural model shown in Figure 6.18 to include a linear, semi-active tuned mass damper

the wind tunnel tests in Section 6.2 and TSRs to evaluate the relative effectiveness of passive and semi-active TMDs for vibration suppression.

This study places a linear TMD in the nacelle of the turbine acting against the rotation of the turbine tower. The sine of the tower rotation is converted into a linear displacement by multiplying by the tower height, as shown in Figure 6.21. Similarly, the sine of the base rotation speed times the tower height is added to the wind speed to determine the relative wind speed on the translating blades. The properties of the tuned mass damper are derived using the method laid out by Connor (2002) as described in Sections 5.2.3.2 and 5.2.3.3. To summarize, based on an assumed mass ratio between the TMD mass and the lumped nacelle mass of the turbine, the optimal TMD stiffness, and the damping values can be derived. In this case, it was decided to use a relatively common mass ratio of 1% resulting in a TMD mass (m_{TMD}) of 0.239 kg, a damping coefficient (c_{TMD}) of 0.0285ω Ns/m, and a stiffness (k_{TMD}) of $0.233\omega^2$ N/m, where ω is the natural frequency of the turbine (3.617 rad/s for the full-stiffness model).

For the semi-active TMD, it was decided to use a variable damping controller which was previously found in Chapter 5 to be highly-effective while also simple to implement. It uses the ground-hook controller as described in Section 5.2.3.4, which alternates the damping coefficient between a high and low value depending on the physical state of the turbine. In this case, when the turbine is moving away from its neutral position, the damping coefficient defined above is used, while when it is moving back towards its neutral position this value is reduced by 60%, thus encouraging the turbine to remain as close to its neutral position as possible.

To offer a relatively broad range of tests, the dynamic structural model with the surrogate thrust model was run by itself and while equipped with the passive TMD and the semi-active TMD in wind speeds of 3.63, 4.07, and 4.85 m/s using the recorded wind speed time histories from the first round of wind tunnel tests in Section 6.2. At each wind speed, the turbines were tested at TSRs of 1.0 and 2.0. From each simulation, the mean and standard deviation of the tower rotation was recorded, ignoring the first 40% of the time history to minimize start-

up effects. Further, the percent differences between the results recorded from the PTMD and STMD cases and the TMD-less case were calculated.

Note that the mass ratio and the value of the secondary (lower) damping coefficient in the ground-hook controller will control the response improvement offered by the STMD. For example, the STMD will more effective relative to the PTMD at lower wind speeds with a higher mass ratio and lower secondary damping coefficient. Conversely, a lower mass ratio and higher secondary damping coefficient will mean that the STMD will outperform the PTMD at higher wind speeds. A full parametric analysis of this behaviour is beyond the scope of this case study, which is why the singular mass ratio and secondary damping coefficient were selected to be analyzed here.

The results of the simulations are reported in Table 6.14, which shows that despite the low mass ratio, in all instances both TMDs were effective at reducing the response of the turbine, particularly at higher wind speeds and lower TSRs. The standard deviation of the turbine tower rotation was reduced by approximately 12% in 3.63 m/s wind speeds, and up to 17% in 4.85 m/s wind speeds. It can also be seen that the semi-active TMD matches or outperforms the passive TMD in all cases. In the 3.63 and 4.07 m/s cases, the STMD does not result in a substantial reduction in the standard deviation of the base rotation compared to the PTMD,

Wind speed	Tip speed ratio	TMD type	Mean rot. (10^{-3} rad)	St. dev. rot. (10^{-3} rad)	St. dev. reduction
3.63 m/s	1.00	None	6.02	3.223	-
		Passive	6.02	2.833	12.1%
		Semi-active	6.02	2.831	12.2%
	2.00	None	6.41	3.307	-
		Passive	6.41	2.919	11.8%
		Semi-active	6.41	2.917	11.8%
4.07 m/s	1.00	None	9.16	4.063	-
		Passive	9.16	3.587	11.7%
		Semi-active	9.16	3.585	11.8%
	2.00	None	9.66	4.171	-
		Passive	9.66	3.698	11.3%
		Semi-active	9.66	3.696	11.4%
4.85 m/s	1.00	None	16.81	3.641	-
		Passive	16.82	3.050	16.2%
		Semi-active	16.82	3.004	17.5%
	2.00	None	17.58	3.716	-
		Passive	17.59	3.138	15.6%
		Semi-active	17.59	3.094	16.7%

Table 6.14: Summary of simulations comparing TMD and STMD using the dynamic structural model. Mean rotation (rot.) and the standard deviation (st. dev.) are reported, and are used to compare the reduction in standard deviation compared to the cases with no TMD

but in the 4.85 m/s case the STMD reduces the response by $>1\%$ compared to the no-TMD case at both TSRs, which represents a relative improvement compared to the PTMD of 8.1 and 7.1% respectively. Overall, these results match the conclusions found in Lalonde et al. (2020b): both forms of TMD are quite effective, the semi-active TMD is both more efficient and more effective at response reduction, though there are instances where the improvement is quite minor. This short study also shows how the validated dynamic structural model can be applied for numerical turbine studies, which will see further use in the following chapter using a novel HS framework that incorporates an improved surrogate aerodynamic model.

6.5 Conclusions

This study details the development of an experimentally-validated structural wind turbine model to act as a testbed for future hybrid simulation studies. The experimental data used to build such a model was required to meet two criteria: that it avoids the scaling issues seen in traditional wind tunnel tests of wind turbines, and that it experiences measurable aeroelastic deflection. Thus, two rounds of wind tunnel testing were performed on a 1.1 m residential wind turbine with a custom-built aeroelastic rotating base. The rotating base mimics the pitch rotation in buoy-supported floating wind turbines and was achieved without affecting the aerodynamics of the turbine by placing stiffness and damping systems beneath the floor of the tunnel. Two rounds of testing were performed at a number of wind speeds, tip speed ratios, structural damping, and structural stiffness levels. There were issues in some tests with insufficient vibrations, a lack of aeroelastic excitation, and non-zeroed measurements, but ultimately the gathered data was sufficient to meet the two requirements.

A 1-DOF numerical structural model of the rotating-base turbine was built using the wind tunnel test results. This involved the development of a multi-layer perceptron neural network-based surrogate thrust model that predicted the thrust coefficient of the rotor based on the hub-height wind speed and the tip speed ratio. This model was trained on OpenFAST data using a previously-validated numerical approximation of the turbine rotor and was predicted to have an accuracy of about 96%. Subsequently, the basic structural properties were determined based on free vibration tests and validated based on previous fixed-base tests. The loading models were then validated against the mean results from the new wind tunnel tests, with an accuracy of greater than 80%, which was the maximum achievable due to the uncertainty in the surrogate thrust model at the test conditions. Finally, the structural damping was modelled based on the second round of wind tunnel testing, which was best-captured using a combined linear-quadratic-Coulomb damping model - this amount of damping was found to be overly-strong based on the wind tunnel testing, and thus it was intentionally reduced in the structural model. A condensed summary of the structural model properties was then provided.

Finally, the developed and validated 1-DOF structural model and the surrogate thrust model were used to study a non-linear vibration damping system. Specifically, passive and semi-active tuned mass dampers were compared for their vibration reducing capabilities on the rotating-base turbine. Using optimized damper parameters, vibrations were able to be reduced by 12-18%, and the structural model was shown to be effective in hybrid simulation-like test cases. Future testing will employ this structural model in a study to validate a novel hybrid simulation framework for wind turbines.

Chapter 7

Development of a novel hybrid simulation framework employing a CFD-based surrogate model to study a semi-actively damped wind turbine

Summary

Current applications of hybrid simulation (HS) for wind-loaded structures such as wind turbines are limited by the reliance on lower-accuracy “pre-calculated” aerodynamic data. This study investigates the effectiveness of a novel HS framework that incorporates a computational fluid dynamics (CFD) based surrogate model to generate higher-accuracy aerodynamic data within the simulation loop. The investigation is performed across twelve artificial HS of a residential wind turbine with an aeroelastic rotating base equipped with a semi-active tuned mass damper (STMD). Firstly, the development of the surrogate aerodynamic turbine blade model is detailed covering the training of a convolutional neural network using CFD-generated data. 24 URANS CFD simulations were performed at a variety of wind speeds, tip speed ratios (TSRs) and base rotations, the results of which were used to optimize and train the CNN. The resulting surrogate model is capable of quickly estimating the blade thrust based on the wind speed, rotor speed and blade position. The novel HS framework that incorporates this trained surrogate model is applied in the artificial hybrid simulations and contrasted against the existing technique of pre-calculating the aerodynamic loads, which were generated using OpenFAST. By comparing the results of the twelve AHS considering the predicted rotor thrust, base rotation, and STMD effect, the impact of the proposed framework can be assessed. While an exact quantification of its effectiveness can not be drawn, validation of the CFD simulations and observations of the HS indicate that the framework is feasible and should offer notable improvement over the existing framework in both accuracy and aeroelastic fidelity.

7.1 Introduction

Chapters 1 and 2 have detailed the promise of HS for wind turbine research and other wind engineering applications, as well as the need for an updated HS framework to improve the accuracy of aerodynamic load prediction. Existing HS of wind turbines were presented in detail in Section 2.7, including two notable studies (Brodersen et al., 2016; Zhang et al., 2017) which applied real-time HS to study industrial wind turbines equipped with novel non-linear damping systems. In these HS, the dampers were modelled experimentally while the turbines were modelled numerically. Aerodynamic loads were applied from time histories that were generated using blade element momentum (BEM) based methods and/or generated prior to the real-time HS. In other words, the aerodynamic loading was generated using lower-accuracy numerical techniques and since they were calculated prior to testing, they were unable to take into account the true structural response of the damper-equipped turbine, limiting the accuracy of the modelled aeroelasticity. While the existence of these limitations is clear, these studies fail to acknowledge them and often do not even report the specifics of the numerical model used to perform the aerodynamic pre-calculations. While the impact of these simplifications can not be quantified from these studies, Lee et al. (2017) has reported that simplified aerodynamic model such as BEM introduce errors of $\geq 5\%$ compared to advanced models such as CFD. Additionally, Section 2.8 summarizes a numerical study of the impact of limiting the aeroelastic fidelity in an HS-like simulation, finding that depending on the numerical model used to perform the pre-calculation and the response index of interest, errors of up to 70% can be seen. Thus it is clear that the existing HS framework has the potential to introduce a non-negligible amount of error in wind engineering applications.

The novel HS framework proposed in this study is intended to address both of these observed limitations in the existing framework by employing a surrogate model for aerodynamic calculations that is both comparable in accuracy to CFD while being sufficiently lightweight computationally to be run within the real-time HS loop. While these two improvements seem at odds with each other, a surrogate model consisting of a neural network (NN) trained on CFD-generated data should be able to achieve them. Neural networks are a form of machine learning that learn to relate input data to output data without performing the underlying simulations, such as relating wind speed to aerodynamic loads. A well-trained neural network can achieve comparable accuracy to the techniques used to generate its training data, and due to high computational speed of these networks, they have the potential to be able to be incorporated directly into a real-time hybrid simulation. This study details the development of a convolutional NN (CNN) trained on CFD-generated aerodynamic data, creating a surrogate model that is capable at producing near-CFD-level results at real-time computation speeds. The proposed HS framework that incorporates this surrogate model will then be compared to the existing technique of pre-calculating wind loads in a series of artificial hybrid simulations.

Artificial hybrid simulation (AHS), also known as conceptual hybrid simulation (Song et al., 2020), is a technique used to study aspects of HS without the specialized lab equipment required for a true real-time HS. It is achieved by approximating the experimental substructure as an additional numerical model, while still modelling the realistic transfer of data between the numerical and “experimental” substructure by introducing artificial, realistic delays and errors. Artificial HS are thus an approximation of true HS in a purely numerical environment, allowing aspects of the HS (frameworks, timing, control systems, etc.) to be initially studied

without requiring the use of the comparatively expensive experimental components. AHS was chosen to be applied here to investigate the effectiveness of the proposed HS framework due to limited access to the experimental equipment needed to perform a true HS.

This study presents a series of twelve artificial hybrid simulations of a rotating-base residential turbine equipped with a semi-active tuned mass damper (STMD), using both the existing technique of pre-calculating aerodynamic loads as well as a novel HS framework that incorporates a NN-based, CFD-trained surrogate model. In these AHS, the STMD is modelled numerically as the artificial “experimental” substructure while the turbine and aerodynamic models make up the numerical substructure. The wind turbine was tested at three combinations of wind speeds and tip speed ratios (TSR), using three pre-calculated aerodynamic load time histories and well as the optimized trained surrogate aerodynamic model. Section 7.2 presents the turbine model, the wind speed cases, and the pre-calculated aerodynamic time histories. Section 7.3 details the validation and results of 3D CFD simulations of the residential turbine rotor, as well as the training and validation of the resulting NN to act as the surrogate model. Finally, Section 7.4 lists the results of the artificial hybrid simulations, and Section 7.5 draws relevant conclusions. Overall, while an exact quantification of the improvement offered by the proposed framework could not be drawn, it was shown to be feasible to implement and a number of indicators of its effectiveness were identified.

7.2 Methodology: structural model, TMD model and pre-calculated aerodynamic loads

This study presents a series of AHS of a 1.1 m residential wind turbine with a rotating base equipped with a semi-active TMD controlled by a magnetorheological (MR) damper. Three different combinations of wind conditions were tested, as well as four different aerodynamic models resulting a total of twelve AHS. The three different wind combinations match cases in previous experimental tests, while the four aerodynamic models consist of three pre-calculated aerodynamic load time histories of varying aeroelastic fidelity as well as the NN-based, CFD-trained surrogate model employed by the novel HS framework proposed by this study. The test case of a wind turbine equipped with a non-linear external damping system for reducing aeroelastic deflections was chosen as it is the same case that has been previously studied in HS of wind turbines.

The residential wind turbine used for this testing was a numerical recreation of a real 900 W three-bladed horizontal-axis wind turbine with a rotor radius of 1.1 m, a photo of which can be seen in Figure 6.1. This wind turbine was studied in wind tunnel tests in Chapter 6 while mounted on a rotating base allowing for aeroelastic displacement in the fore-aft direction. This rotational response is similar to the base pitch that occurs in floating offshore turbines. The numerical 1-DOF rotational model of this wind turbine was derived from wind tunnel testing as described in Section 6.3 – this model is presented in Table 6.13 and is shown equipped with a linear, nacelle-based STMD in Figure 6.21. To summarize: a linear STMD is attached to the nacelle and assumed to move horizontally along the fore-aft direction as the base rotates. The equation of motion controls the response of the turbine, and loading on the turbine is applied by the passive eccentric mass moment, the tower drag moment, the restoring moment from

the TMD, and the aerodynamic rotor moment from the pre-calculated time histories or the CFD-based surrogate model.

These components are detailed in the following sections: Section 7.2.1 presents the three wind condition combinations; Section 7.2.2 presents fundamentals of the nine pre-calculated aerodynamic time history histories, which are generated using OpenFAST as presented in Section 7.2.3; finally the STMD is described in detail in Section 7.2.4. The development and validation of the surrogate model is presented separately in Section 7.3.

7.2.1 Wind speed and tip speed ratio combinations

Three combinations of mean hub-height wind speeds, turbulence intensities and TSRs were applied to the artificial HS in this study. These three cases, labelled A-C, are listed in Table 7.1. They are adapted from previous experimental studies of this wind turbine: the first combination at a mean wind speed of 3.63 m/s matches the most common test conditions in the wind tunnel testing performed in Chapter 6. The second combination at a mean wind speed of 5.15 m/s matches the non-extreme wind tunnel conditions reported by Shirazadeh et al. (2021). The third combination is run at the same mean speed of 5.15 m/s, but at the optimal TSR of 4.98 predicted by Refan (2009). Geometric scaling applied in the CFD simulations of this rotor prevented testing at higher tip speed ratios in higher wind speeds. While the wind tunnel tests inspiring these wind combinations had turbulence intensities in the range of 6-10%, the turbulence in these time histories was limited to 1.1% due to the eventual robustness of the surrogate model (see Section 7.3.6).

In all the combinations, the wind profiles are derived from the open wind field at the WindEEE Research Institute wind tunnel, as described in Section 6.2.3. This profile was best matched using a power law profile with an exponent of 0.05. Flow coherence is captured using the von Karman model, as recommended by wind turbine design codes (IEC, 2005). More details of this model and the IEC flow parameters were presented in Section 5.2.2.1. Flow fields were generated using the open-source flow generation software TurbSim (Kelley and Jonkman, 2012), which is designed to work with OpenFAST. Using TurbSim, the specifications and models for the flow can be chosen and the resulting flow can be output over a specified mesh. 360 second time histories were generated for each of the wind combinations in Table 7.1 at a 20x20 grid centered on the hub with node spacing of 0.15 m. To diversify the training data, two different wind time histories were generated for cases B and C using different random seeds. The

Variable	Unit	A	B	C
Mean hub-height wind speed	m/s	3.63	5.15	5.15
Turbulence intensity	–	0.011	0.011	0.011
Tip speed ratio	–	0.95	1.08	4.98
Rotor speed	rad/s	3.14	5.06	23.32
Rotor speed	rpm	30.0	48.3	222.7

Table 7.1: Wind speed, turbulence intensity, and TSR testing combinations

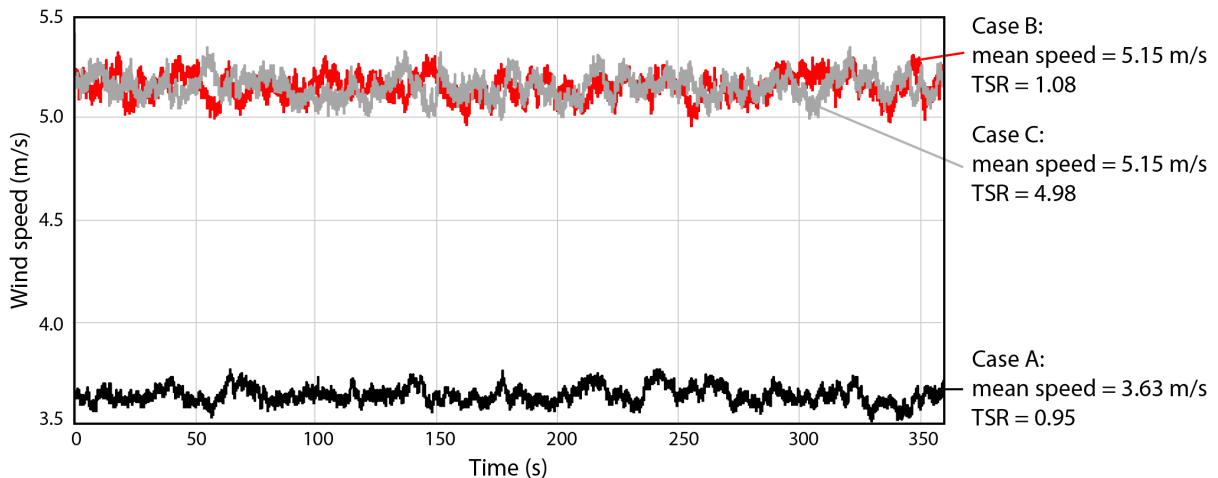


Figure 7.1: Three wind speed time histories generated using TurbSim and the related TSR for that wind case

i-direction (fore-aft) wind speed time histories recorded at the hub are presented in Figure 7.1. These wind field time histories will be used both for pre-calculating aerodynamic loads as well as in tandem with the CFD-based surrogate model during testing.

7.2.2 Aerodynamic models and HS frameworks

As discussed in Chapter 2, aeroelastic analysis of a structure consists of two sets of calculations. Analysis begins with determining the drag loads caused by the wind field. This can be referred to as the aerodynamic admittance. The second set of calculations determines the resulting structural deflections from the aerodynamic loads, sometimes referred to as the mechanical admittance. The mechanical model is identical for all the artificial HS performed in this study, however there are four different aeroelastic models considered across these tests as shown in Figure 7.2. These aerodynamic models consist of three different sets of pre-calculated aerodynamic loads as well as the surrogate model used in the proposed HS framework. Figure 7.2 shows that an ideal HS performs the aerodynamic calculations within its testing loop, which pre-calculated loads fail to achieve. The proposed HS framework, on the other hand, succeeds at this integration.

Three different sets of pre-calculated aerodynamic loads are generated to investigate the effect of the fidelity of the model used to perform the aerodynamic pre-calculations. It is clear that the aerodynamic loading and the resulting structural response of a structure (i.e. its aeroelasticity) can be strongly coupled. The deflection of a structure will affect its projected area to the wind, thus affecting the resulting aerodynamic loads, which in turn affects the deflection, etc. Therefore, failing to model this relationship will introduce some amount of error into the simulation. The degree to which this pre-calculation process impacts the results of the HS depends on the fidelity and accuracy of the numerical model used to perform the pre-calculation. Consider the hypothetical scenario of a pre-calculation model that perfectly simulates the HS test and predicts the response with the same accuracy as the experimental substructure. In this scenario the aerodynamic loads would be pre-calculated with perfect

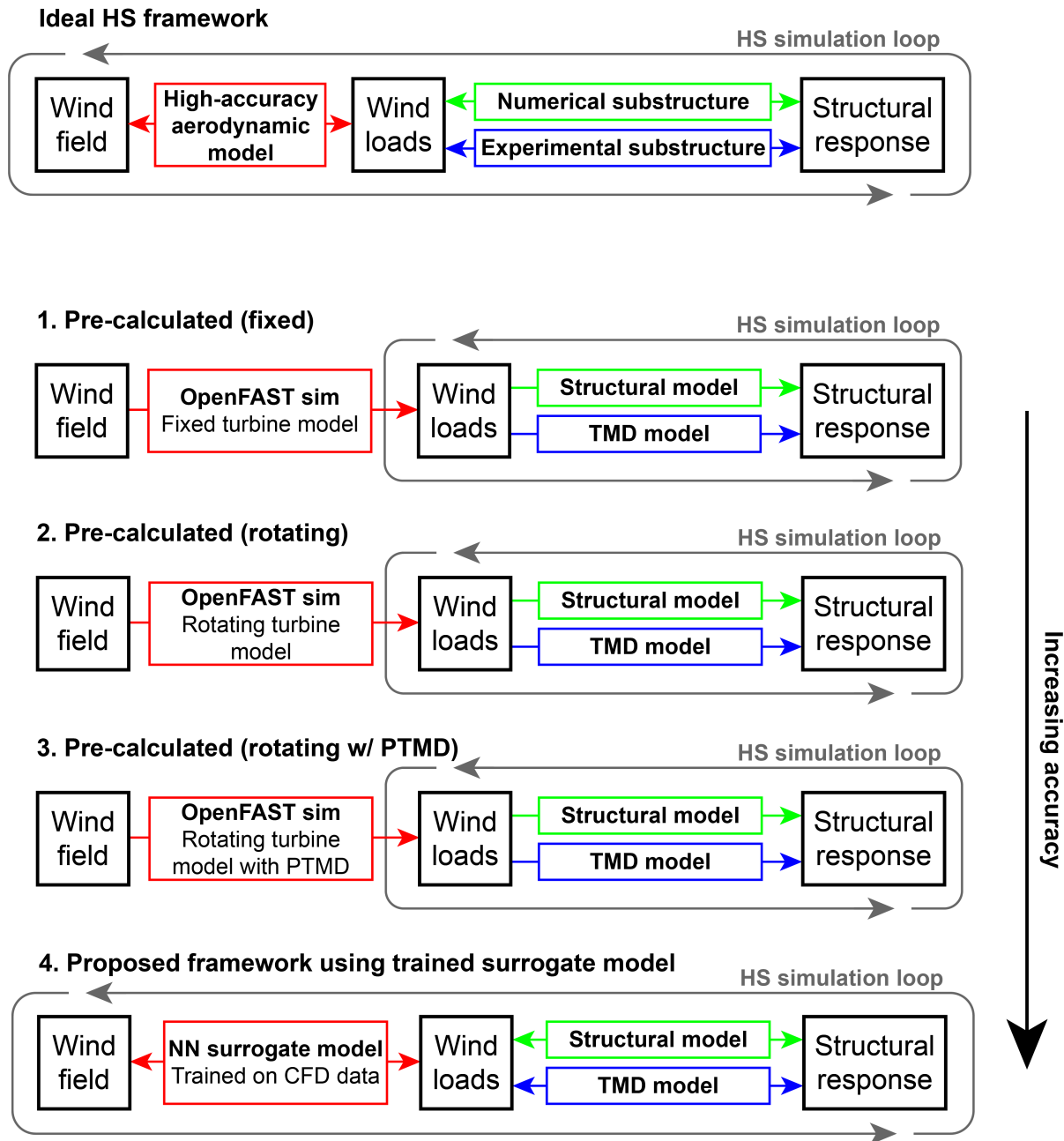


Figure 7.2: The four frameworks employed in the AHS compared to an idealized HS framework; all three wind case combinations are simulated by each framework

accuracy, introducing no error during the HS. However, as the pre-calculation model strays further from reality, the amount of error introduced to the HS will increase. Therefore, the better the numerical pre-calculation model replicates the HS model, the smaller the negative effect of using pre-calculated aerodynamic loads should be. Therefore, three different pre-calculation models of varying fidelity are used to generate the nine pre-calculated load time histories. In order of increasing accuracy, these models are:

1. Fixed wind turbine model: the turbine is modelled with a rigid tower and a fixed base, resulting in no aeroelastic deflection.
2. Rotating wind turbine model: the turbine is modelled with a rigid tower and a rotating base that matches the values determined in Chapter 6.
3. Rotating wind turbine with passive TMD (PTMD): the wind turbine is modelled with a rigid tower a rotating base, assuming a PTMD in the nacelle that operates in the fore-aft direction and uses the optimized properties that will be detailed in Table 7.3.

These pre-calculation models are also shown in Figure 7.2, and the resulting pre-calculated wind loads were generated using OpenFAST as described in Section 7.2.3.

7.2.3 OpenFAST aerodynamic load time histories

The open-source wind turbine testing software OpenFAST (Jonkman et al., 2021) is used to pre-calculate the aerodynamic loads. OpenFAST employs the 2D BEM method, a widely-used technique for predicting the aerodynamic loads on a rotor. BEM is computationally efficient, but due to simplifications of aerodynamic effects it is of lesser accuracy than advanced aerodynamic simulations such as CFD.

The three turbine models of varying fidelity listed in Figure 7.2 are separately modelled in OpenFAST, and are each subjected to the three sets of wind conditions listed in Table 7.1. These models are implemented in OpenFAST by restricting response DOFs in the ElastoDyn module, modelling the rigid tower base using a custom SubDyn base, modelling the PTMD using the ServoDyn module, and recording the resulting aerodynamic blade loads from the AeroDyn module. Load time histories are generated for each model subjected to each wind case at the same 0.001 s time step and 180 s simulation length as the TurbSim wind fields. The resulting thrust time histories are reported in Figure 7.3; during the AHS these aerodynamic loads are applied to the structural model during the simulation as an overturning moment by multiplying them by the hub height. The reported time histories generated by the two rotating turbine models under wind case C have been edited slightly to remove a small number of signal spikes erroneously generated by OpenFAST as the base rotation approached the built-in limit of 0.4 rad. The spikes occurred for less than 0.01 s at a time and were replaced by the local mean which should not impact the temporal and spectral properties of the time histories.

It was found in the case study in Section 2.8 that limiting the aeroelastic response of members in the pre-calculation model (such as generating wind loads on a fixed-base turbine instead of a rotating-base) introduced up to 5% error between the predicted results. However the test case in that study, which compared a fixed turbine blade to a flexible turbine blade, differs from the one here. Thus, the discrepancies between the pre-calculated load predicted by the fixed turbine model and the rotating models must be analyzed. The time histories predicted by OpenFAST are compared in Figure 7.3. As can be seen, the loads predicted by the three models closely match in wind cases A and B when the TSR and resultant thrusts are low, but a larger discrepancy is seen in wind case C due to the higher TSR and thrust. Additionally, the importance of TSR on blade loads can be clearly seen – increasing the TSR between cases B and C results in an approximately 7 times increase in the resulting thrust due to the increased rotor speeds. Table 7.2 reports the root mean square error (RMSE) between the fixed model results

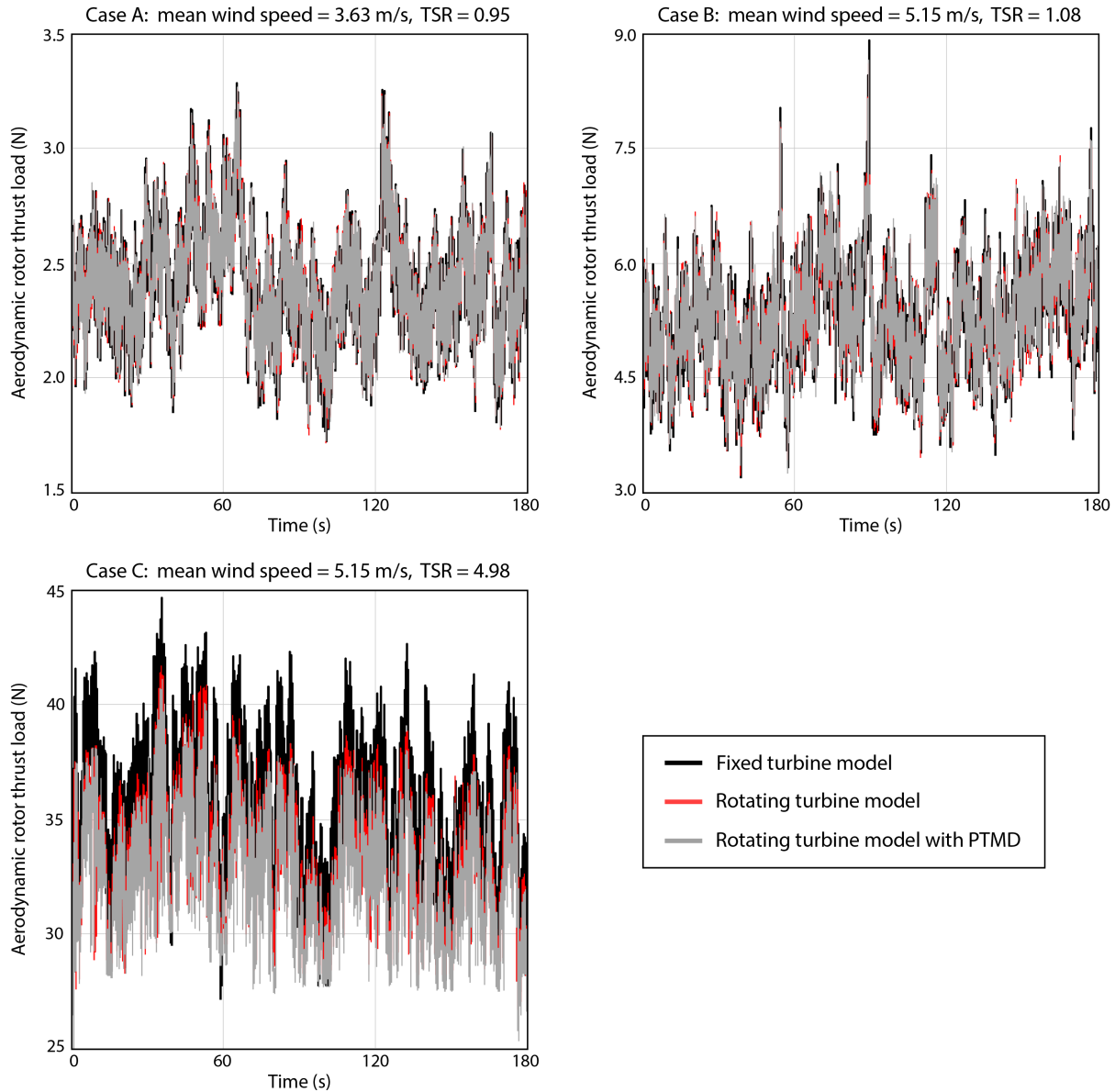


Figure 7.3: Comparison of the pre-calculated aerodynamic rotor loads using fixed, rotating and rotating-with-PTMD numerical models at mean hub height speeds of 3.63 m/s and 5.15 m/s

Difference measurement:	RMSE (N)			NRMSE		
	Wind case:	A	B	C	A	B
Rotating vs fixed models:	0.047	0.184	2.731	2.96%	3.23%	15.65%
Rotating-with-PTMD vs fixed models:	0.052	0.204	3.387	3.33%	3.57%	19.41%

Table 7.2: Differences between the OpenFAST-predicted thrust time histories generated using the fixed, rotating, and rotating-with-PTMD models, under the wind cases described in Table 7.1

and the results from two rotating models, as well as the normalized RMSE (NRMSE) where the RMSE is divided by the absolute range of the fixed model results – see Equations 4.2 and 4.3. The reported RMSEs match the observed trend that there is a greater difference between the thrust predicted by the rotating-with-PTMD model compared to the fixed model than there is between the rotating and fixed models. In other words, the predicted thrust changes as the accuracy of the model is increased. Additionally, it can be seen that the cases with lower wind speeds and TSRs that result in less base rotation match the fixed model better than case 3 where the thrust is significantly increased – larger base rotations result in greater discrepancies between the rotating- and fixed-base rotors. Overall, it can be seen that the results found in Section 2.8 are roughly replicated here, in that simplified dynamics of the pre-calculation model may result in a small but noticeable amount of error. This justifies the application of the three different sets of pre-calculated load time histories in the HS.

7.2.4 MR-controlled semi-active tuned mass damper design

Based on the findings of Chapter 5, which compared the effectiveness of a number of different passive and semi-active TMDs, the external damping system used in these AHS was chosen. The selected system of a variable-damping STMD controlled by a MR damper combines non-linear behaviour, modelling simplicity, and efficiency. As detailed in Section 5.2.3.4, MR dampers are filled with magnetically-sensitive fluid that changes viscosity when exposed to a magnetic field. Thus, a circuit connected to an electromagnet can control the MR damper to adjust its damping properties in real-time. As was shown in Chapter 5, the ground-hook controller that reduces the amount of damping between the turbine and the TMD when the turbine is moving towards its neutral position, the overall magnitude of vibrations can be reduced. This section offers a summarized explanation of the damper design process and control technique, though additional information can be found in Chapter 5, particularly in Section 5.2.3.

The STMD consists of a secondary mass attached to a primary structure that is capable of independent displacement. This secondary mass reacts in opposition to the vibration of the structure, damping the response. This mass is typically modelled as being connected to the main structure through stiffness and damping components (see Figure 5.6), which allows it to be modelled numerically using the standard structural equation of motion (see Equations 5.4-5.5). The main advantage of TMDs is their efficiency: using a relatively small secondary mass and relatively weak stiffness and damping members, notable reductions in vibrations can be achieved. To achieve the same vibration reduction through direct increases in the stiffness or damping of the main structure would require significantly stronger stiffness and damping elements compared to those connected to the TMD. A limitation in traditional passive TMDs is that they must be tuned to reduce vibrations of a specific frequency, and their effectiveness diminishes quickly at frequencies beyond this. Semi-active TMDs attempt to broaden the robustness of TMDs by modifying their dynamic properties during operation. As was shown in Chapter 5, controlling the damping vibration via a MR damper was successful at increasing the effectiveness of the TMD and could be realistically modelled numerically.

Effective TMD design firstly requires the dynamic properties to be optimized, the exact equations to do so are presented in Sections 5.2.3.2 and 5.2.3.3 based on the Den Hartog guidelines that are reproduced in Connor (2002). This optimization process begins with selecting the desired mass ratio between the turbine and the TMD. In a realistic scenario, this would be con-

Parameter	Symbol	Unit	Value
Turbine mass	m	kg	23.9
Turbine natural frequency	ω	rad/s	3.62
TMD mass ratio	μ	–	0.01
TMD mass	m_{TMD}	kg	0.24
TMD natural frequency	ω_{TMD}	rad/s	3.57
TMD stiffness	k_{TMD}	N/m	3.05
TMD damping ratio	ξ_{TMD}	–	0.060
40% of TMD damping ratio	$0.4\xi_{TMD}$	–	0.024
TMD damping coefficient	c_{TMD}	Ns/m	0.103

Table 7.3: Optimal passive TMD properties

trolled by the available space and desired damping levels, however in this test a modest mass ratio value of 1% was selected based on the observed effectiveness in the case study presented in Section 6.4. Based on the mass ratio, an optimal natural frequency of the TMD can be found, which will be very close to the natural frequency of the main structure. The stiffness connection between the main structure and TMD is then derived from this natural frequency. The mass ratio also determines the optimal damping ratio for the connection between the structure and TMD, which is used to determine the damping coefficient. Using Equations 5.6-5.11, the optimal passive properties for the TMD can be found, which are reported in Table 7.3. These same properties were derived for the simpler TMD model employed in Section 6.4.

During operation, the modified ground-hook controller recommended by Martynowicz (2017) will be used to control the semi-active damping. Equation 5.19 describes this control-scheme, but to summarize, when the turbine is moving away from its neutral position, the damping will be maximized to its recommended passive value, but will be reduced significantly when the turbine is moving back towards its neutral position. This controller thus encourages the turbine to stay near its neutral position, reducing the magnitude of fluctuations. For the MR damper, this means that the controller will alternate between applying a high and low voltage to the MR damper to achieve a high and low damping coefficient respectively. In this testing, a low damping value equal to 40% of the high damping value was used, which was shown to be effective in the testing summarized in Section 6.4.

The modified Bouc-Wen model (see Figure 5.7) was used to numerically model the MR damper. The Bouc-Wen model approximates the damper as a combination of dampers, springs and hysteretic elements which are all a function of the applied current, and has been shown to match the physical behaviour of an MR damper well (Talatahari et al., 2012). This model includes consideration of the delay between applying the voltage to the damper and the resulting change in properties. The model is rather complex with a number of parameters; it is detailed in Equations 5.12-5.18. A summarized version is that the internal voltage is derived, used to determine the properties of a number of stiffness, damping, and hysteretic members, then internal responses are resolved and finally the resulting damping force can be derived based on the displacement and velocity applied to the MR damper. There are fourteen different parameters used in the modified Bouc-Wen model which must be derived experimentally; the testing in

Parameter	Symbol	Unit	Value	Factor*
Initial MR displacement	x_0	m	0	–
Hysteretic factor 1	γ	m^{-2}	$3.62 \cdot 10^{-8}$	10^{-8}
Hysteretic factor 2	β	m^{-2}	$3.62 \cdot 10^{-8}$	10^{-8}
Hysteretic factor 3	A	–	154.6	–
Voltage application rate	η	s^{-1}	60	–
Hysteretic exponent	n	–	2	–
Base hysteretic stiffness	α_a	N/m	0.00358	10^{-4}
Hysteretic stiffness rate	α_b	(N/m)/V	0.00830	10^{-4}
Base inner damping coefficient	c_{0a}	Ns/m	0.00015	10^{-4}
Inner damping coefficient rate	c_{0b}	(Ns/m)/V	0.00328	10^{-4}
Base outer damping coefficient	c_{1a}	Ns/m	0.00995	10^{-4}
Outer damping coefficient rate	c_{1b}	(Ns/m)/V	0.04039	10^{-4}
Inner stiffness	k_0	N/m	0.00059	10^{-4}
Outer stiffness	k_1	N/m	$16 \cdot 10^{-7}$	10^{-4}
MR damper displacement	x	m	Calculated	–
Artificial internal displacement	y	m	Calculated	–
Hysteretic displacement	z	m	Calculated	–
Total MR damper force	F_D	N	Calculated	–

Table 7.4: Modified Bouc-Wen model properties. Factor refers to the factors applied to the values reported by Li et al. (2017) to reduce the damper strength, which were applied as a 10^4 times reduction in the internal displacement values

Chapter 5 of an industrial wind turbine used the values for a 2.4 kN MR damper reported by Li et al. (2017). Since the MR damper required for this smaller turbine must be much weaker, but lacking in literature that reports these values for a weaker damper, the values from Li et al. (2017) are adjusted by reducing all distance-based measurements by a factor of 10^4 , approximating it as a 0.24 N MR damper. The Bouc-Wen parameters, including their reduction factor, are summarized in Table 7.4.

The applied voltages that achieved the target damping ratios for the TMD were also evaluated. From Table 7.3, it is known the the optimal damping ratio for the TMD is 6.0% and thus the target damping ratio for when the low voltage is applied is 2.4%. The STMD was modelled in MATLAB using the properties described in Tables 7.3 and 7.4, and free vibration tests were run with different applied voltages. Based on the free vibration results, the approximate damping ratio of the system can be derived using the simplified logarithmic decrement method:

$$\xi = \frac{\ln \frac{a_T}{a_{T+1}}}{2\pi} \quad (7.1)$$

Here, ξ is the damping ratio, while a_T and a_{T+1} refer to the amplitudes of adjacent free vibration response peaks with T referring to the natural period of the TMD. This simplified model is valid provided $\xi \ll 1$. The free vibration response of the TMD matched extremely well to a linear damping model. Using trial and error, it was found that $\xi_{TMD}=0.060\%$ was found at 2.32 V and $\xi_{TMD}=0.024\%$ was found at 0.79 V. These voltages will be used to control the MR damper.

As the STMD will be the stand in for the experimental substructure in this artificial HS, there must be communication delays between the STMD and the rest of the simulation to capture the realistic limitations of experimental testing. Artificial delays and noise will thus be added to the STMD model to replicate these experimental effects, as follow:

- The signal travelling from the numerical model to the STMD is assigned a delay of 10 ms. This signal represents the time delay between deciding the desired voltage for the MR damper and the damper receiving the voltage, and was quantified by Caterino et al. (2013).
- The delay from displacing the TMD, recording the resulting restoring forces, and returning them to the numerical model is assigned a value of 20 ms. This delay length will depend on the magnitude of the requested displacement and the equipment being used; Chae et al. (2013) measured these delays in a HS as ranging from 20 to 40 ms. As discussed in Section 2.6.2, actuator controllers are critical for minimizing the propagation of this error in a true HS, but implementation of a numerical controller is beyond the scope of this study. Instead, the lower end value from Chae et al. (2013) is used to approximate the improved accuracy and stability provided by these controllers.
- The equipment used to measure the assigned displacement and resulting force has built-in measurement error. While the precise amount of error will vary depending on the piece of equipment, lab equipment attempts to keep measurement error at less than 0.5% of its maximum value. In light of this, factors are applied to both the measured displacements and resulting force at each time step to approximate the noise stemming from measurement error. These factors are equal to $1+0.005\text{rand}_{\text{norm}}$ where $\text{rand}_{\text{norm}}$ is a normally-distributed random number.

7.3 Methodology: CFD-based surrogate aerodynamic model

The novel framework proposed by this study incorporates a high-accuracy surrogate aerodynamic model of the rotor into the testing itself, with the goal of improving both the accuracy and aeroelastic fidelity compared to the existing frameworks. This surrogate model is a CNN trained on CFD data, which is capable of achieving both high accuracy and high speed. The following section details the generation of the training data using 24 time histories from URANS simulations, followed by the development of the surrogate model using this training data. Readers are directed to Chapter 3 for more information on CFD simulations of wind turbine blades and to Chapter 4 for more information on NN-based surrogate aerodynamic models.

7.3.1 CFD fundamentals

The CFD simulations were performed using the commercial software STAR-CCM+. Due to the intensive computation required for these simulations, Compute Canada supercomputing platforms were used. Generally speaking, simulations were run on 192 GB 32-CPU nodes for a duration of several hours to several days depending on the simulation type, mesh densities, and number of iterations. To keep the testing times as short as possible given the large number

Scaling factor	Symbol	Value	Example
Geometry/Length	λ_L	λ_L	1:4
Wind speed	λ_V	λ_L^{-1}	4:1
Rotor speed	λ_R	$\lambda_V \lambda_L^{-1} = \lambda_V^2$	16:1

Table 7.5: CFD domain scaling factors to maintain Re and TSR similitude. Example values are given for λ_V and λ_R assuming $\lambda_L=0.25$

of required tests, some domains were geometrically scaled down to reduce the required number of mesh cells. Even if the mesh sizes in the outer regions of the domains are scaled down the same rate, this still results in an overall reduction in cell count given that the finest mesh around the blades is controlled by the y^+ criterion rather than the geometric scale directly.

Turbine scaling requires that similitude is maintained in both the Reynolds number and the TSR between the scaled-down and full-scale model. Both the Re (Equation 2.13) and TSR (Equation 2.10) have been discussed in detail throughout this document; by maintaining Re-similitude the lift and drag behaviour of the turbine is modelled correctly and by maintaining TSR-similitude the overall rotor thrust and torque is modelled correctly. Assuming that the fluid properties remain constant, there are thus three physical properties that must be scaled to maintain this similitude as shown in Table 7.5: the geometric scale, the velocity scale, and the rotor speed scale. As can be seen, $\lambda_V = 1/\lambda_L$ and $\lambda_R = \lambda_V^2$ according to the Re and TSR respectively.

A final limitation on the scaling is the Mach number (Ma) which is the ratio of air speed to the speed of sound. It is known that at low Ma, air experiences essentially no compression. This incompressibility simplification is applied liberally in structural engineering applications including greatly simplifying CFD simulations. However, as Ma increases, the compression increases and this assumption ceases to be valid. In incompressible simulations the rule of thumb is that wind speeds should not exceed 100 m/s which roughly corresponds to $Ma = 0.3$. This ensures that any errors from the incompressibility assumption remain below 10%. Ultimately this limitation controls the maximum tip speed of the rotor which scales proportionally to λ_V , which in turn controls the maximum allowable reduction in the domain geometry.

7.3.2 Turbine blade model

To perform a CFD simulation of the 1.1 m turbine rotor, a 3D model of the turbine blade is required. The author did not have access to a true 3D scan of the blade, thus the numerical approximation proposed by Refan (2009) was used. This approximation was previously employed in the OpenFAST simulations in Sections 6.3.2 and 7.2.3, and is detailed in Table 6.5. This approximation consists of a series of NACA 6515 and FX 63 137 airfoils at a given length along the blade with a given pitch, which can be combined to derived a 3D model.

Using the commercial 3D modelling software SolidWorks, airfoil curves could be imported into the modelling environment as a table of 3D coordinates, then lofted together to form the blade. Finally, based on visual investigations of the blade, the base curve was extracted and the tip was rounded. This process is shown in Figure 7.4.

To implement this blade model in CFD, the blade was either tested alone or rotationally-

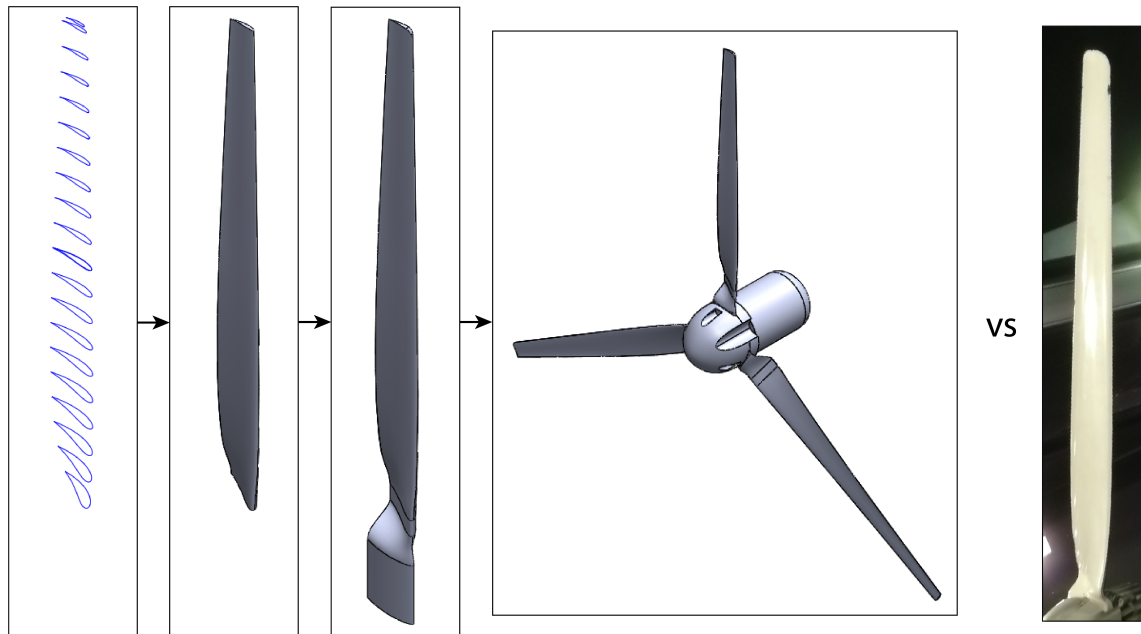


Figure 7.4: 3D rotor model preparation stages compared to photo of real residential wind turbine rotor

arrayed and combined with a nose cone to create the full rotor. The desired domain was then prepared around the blade(s), which were then subtracted from this domain to create a negative. This domain was then imported into the commercial CFD software STAR-CCM+ to be meshed and tested.

7.3.3 CFD validation

To validate the simulations of the 1.1 m turbine blade numerical approximation, a series of URANS CFD simulations were performed of a single blade at a number of pitches, replicating the single-blade wind tunnel tests from Refan (2009) which were presented in Section 6.3.2. URANS was identified in Chapter 3 as the best type of CFD simulation for this application by balancing accuracy and computational requirements. In the previous experimental tests, the mean lift and drag on the turbine blade under 7 m/s wind speeds were extracted and used to determine the overall lift and drag coefficients of the blade for a given pitch angle. This was replicated in a 1:10 scale CFD simulation under 70 m/s wind speeds to preserve Reynolds number-similitude, and since the blade was stationary there was no TSR-similitude to maintain.

Figure 7.5 shows the rectangular, meshed domain for these single-blade simulations, where the overall blade length is 0.11 m due to the scaling. Table 7.6 lists the mesh sizes used in these simulations, sized proportionally to the base size for the different regions. Testing was performed using URANS at 0.001 s time steps and 5 iterations per time step, for a total 1.5 s of analysis per pitch angle, of which only the second half was used to determine the mean result. The y^+ and Courant number requirements discussed in Section 3.3.2 were met in these tests.

Flow snapshots and the isosurface of the blade wake are shown in Figure 7.6 for a pitch of 30° . Isosurface plotting can be used to visually model the vorticity in the flow at a selected

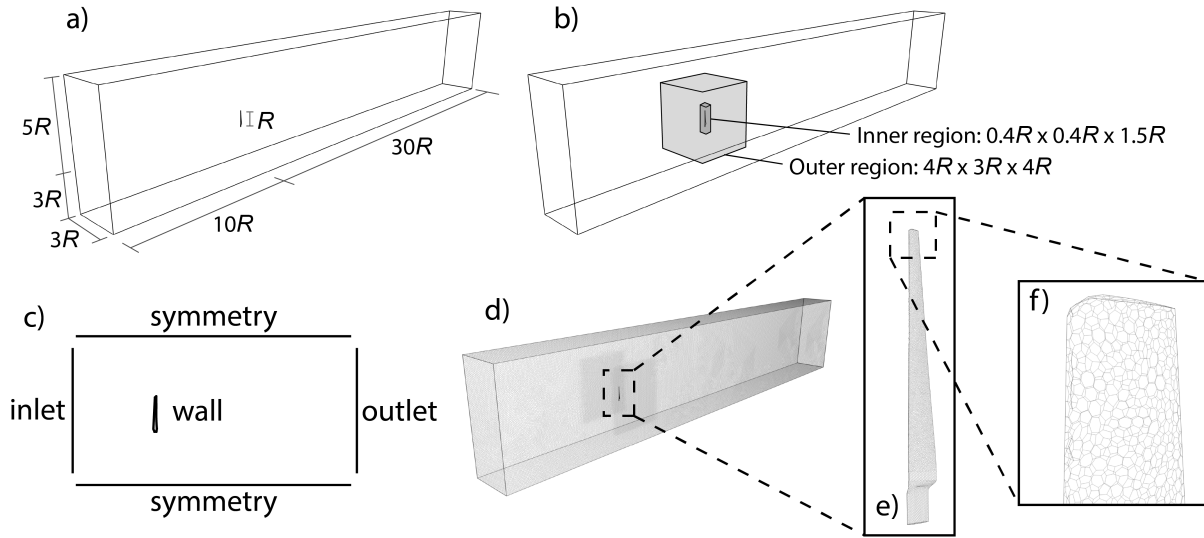


Figure 7.5: Mesh properties of single-blade CFD simulations: a) domain dimensions; b) mesh refinement regions; c) domain boundary specifications; d) meshed domain; e) meshed turbine blade; f) meshed blade tip

Region	Equation	Value
Base mesh size (m)	B	0.01
Outer region mesh size (m)	$0.5B$	0.005
Inner region mesh size (m)	$0.125B$	0.00125
Blade surface mesh size (m)	$0.05B$	0.0005
Number of prism layers	–	30
Prism layers total thickness (m)	–	0.007
Prism layer initial thickness (m)	–	7×10^{-6}

Table 7.6: Mesh sizes used for single-blade CFD simulations. The specified prism layers were along the surface of the turbine blade

Q-criterion. The Q-criterion is a function of the difference between the vorticity and viscous stress in the flow; higher Q-criterion values will result in less of the vortices being modelled. Examining the captured flow and vortices, the results are in line with what would be expected for this test, with smaller vortices at the mid span of the blade and large vortices streaming from the top and bottom. The extracted drag and lift on the blade as well as the resulting coefficient curves are listed in Table 7.7. The predicted coefficients should be unaffected by the geometric scaling process, which is captured in the drag equation with a reduced blade area of $0.82 \times 10^{-3} \text{ m}^2$, an unchanged air density of 1.225 kg/m^3 , and an increased wind speed of 70 m/s.

The results from Table 7.7 are compared to the experimental and numerical results found by Refan (2009) as well as the predictions by OpenFAST from Section 6.3.2 in Figure 7.7. As can

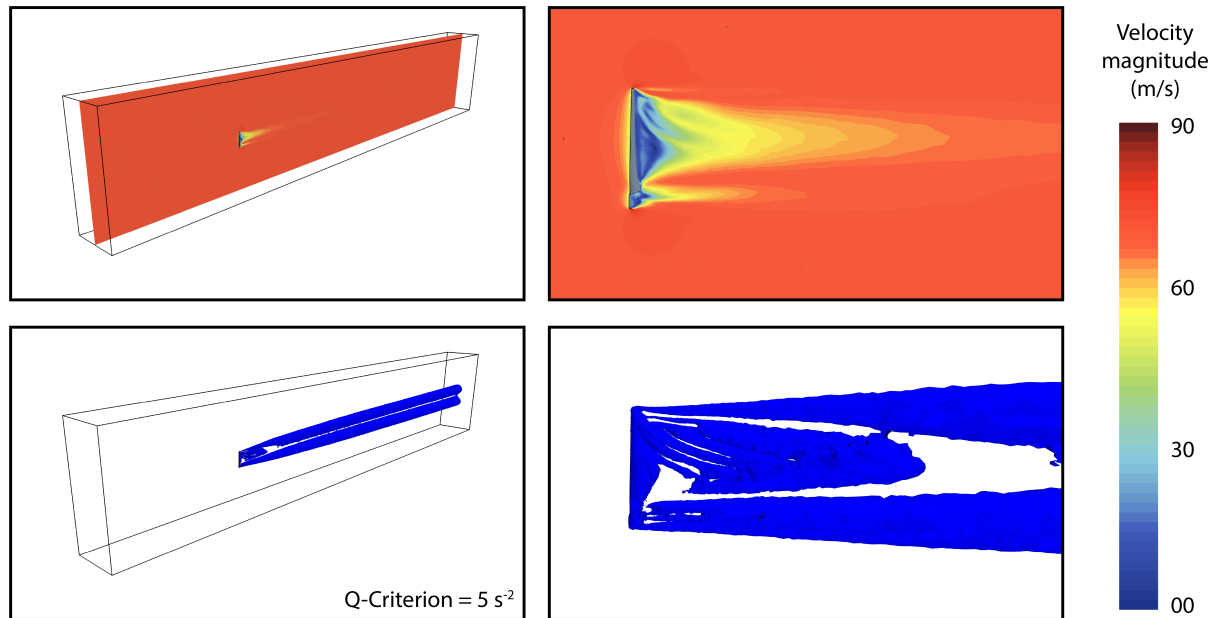


Figure 7.6: Flow and vorticity snapshot of the single-blade CFD sims at a pitch of 30°

Pitch ($^\circ$)	Drag (N)	Lift (N)	Drag coefficient	Lift coefficient
-6	0.397	-0.43	0.161	-0.17
-3	0.328	-0.08	0.133	-0.03
-2	0.311	0.05	0.126	0.02
0	0.288	0.28	0.117	0.11
3	0.280	0.71	0.113	0.29
6	0.307	1.03	0.124	0.42
9	0.367	1.26	0.148	0.51
12	0.447	1.46	0.181	0.59
15	0.545	1.77	0.221	0.72
20	0.728	2.02	0.294	0.82
25	0.971	2.14	0.393	0.87
30	1.270	2.33	0.514	0.94

Table 7.7: Recorded drag and lift values and resulting drag and lift coefficients for 1:10 scale single-blade CFD simulation

be seen, the CFD results are roughly equivalent to OpenFAST at predicting the lift coefficient at lower pitches. However, the drag predictions, which is where BEM typically struggles the most, is significantly improved at all pitches. At higher pitches, the CFD outperforms the OpenFAST predictions at both indices with very good matching occurring at pitches of 20-30°. This is a promising trend as at the lower TSRs investigated in Chapter 6, the angle of attack between the blade pitch and the wind direction will tend to be higher. It must also be noted that since the CFD model is based on a numerical approximation of the turbine blade, even a flawless CFD simulation would be unlikely to perfectly match the experimental results.

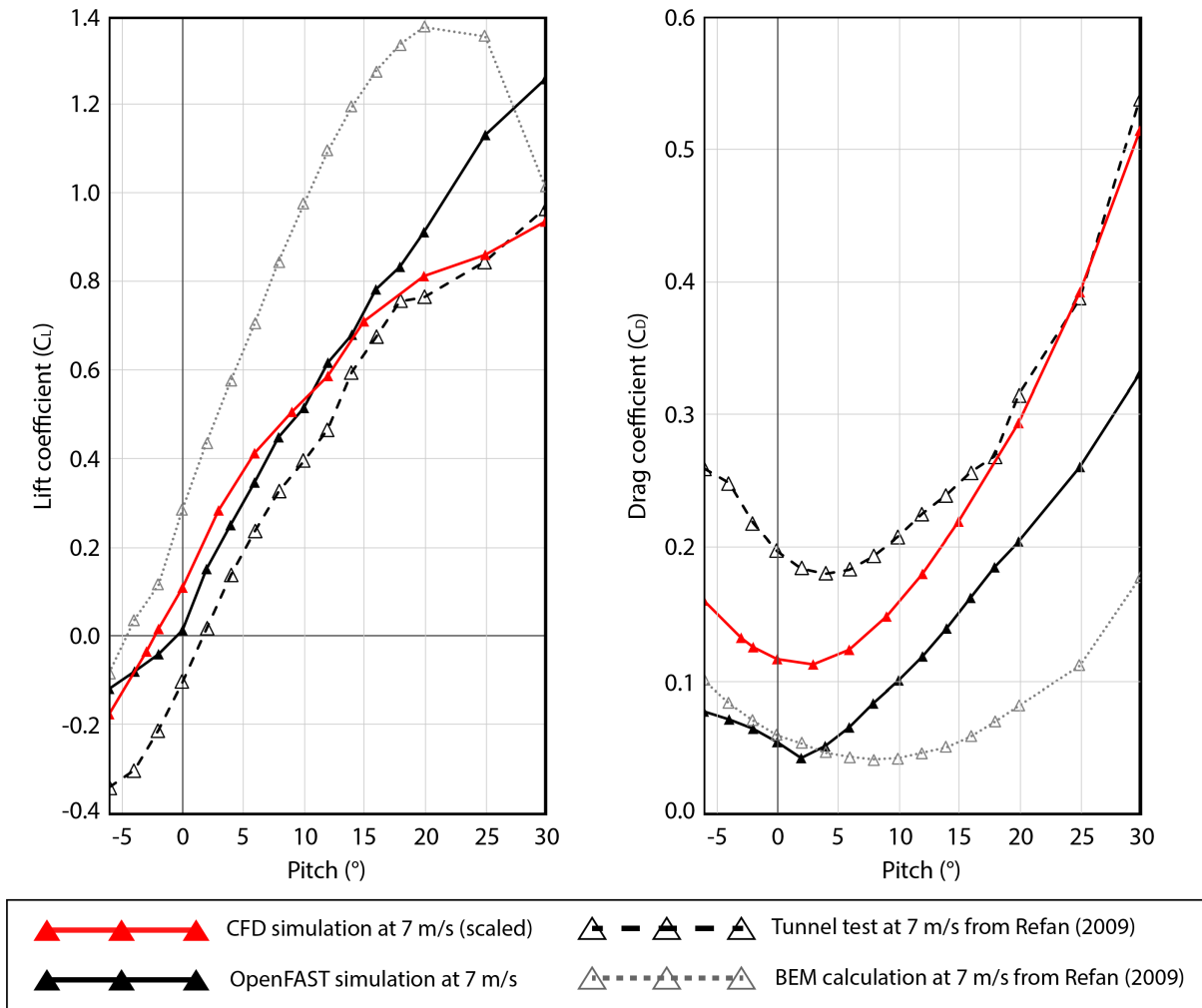


Figure 7.7: Predicted drag and lift coefficient curves for the single-blade CFD sims compared to experimental and numerical testing from Refan (2009) and OpenFAST sims from Section 6.3.2

Overall, this set of simulations help validate the 3D model of the turbine blade developed for this testing as well as serves as an example of the possible value offered by incorporating CFD into HS.

7.3.4 Final CFD mesh

The CFD simulations used to train the neural network require additional complexity compared to the single-blade sims. These complexities and the resulting modelling techniques used to capture them are as follow:

1. The model must include the entire rotor including the nose cone and all three blades to properly capture the effect of non-orthogonal wind on the rotor. This can be accomplished by creating a rotational array of the turbine blades in the 3D model and modelling an approximation of the nose cone, as shown in Figure 7.4. The domain must be modified for the rotor model, which will be detailed below.

2. The turbine blades must rotate in the wind. While it is possible to model moving meshes in CFD simulations, such a process is very computationally expensive on top of CFD's already high computational requirements. Instead, these simulations employ a rotating frame of reference around the rotor axis, which applies a rotational component to the flow in the domain without having to modify the mesh. In essence, this technique rotates the wind around a stationary rotor which generates the same results as a rotor rotating through the wind field. This technique is typically employed for CFD simulations of wind turbines (such as Wang et al. (2016)) due to its computational efficiency.
3. Since in this study the turbine tower rotates about its base in the fore-aft direction, this will change the angle between the rotor axis and the wind direction. Again, while it is possible to create a moving mesh that progressively models varying rotor deflections, it is significantly more computationally efficient to study static meshes at a series of deflection angles. This process can be further optimized by using a single domain mesh for all tests, and instead changing the angle of the wind flow at the inlet, greatly reducing the amount of meshing required for these tests. The rotating wind direction can be facilitated using a spherical inlet.

To decrease the mesh size and computational time required for the testing, these CFD simulations were run at a scale of 1:4 while maintaining both Re- and TSR-similitude, as detailed in Section 7.3.1. To achieve this, the freestream wind speed was increased by a factor of 4 and the rotation speed was increased by a factor of 16. For wind case C where the wind speed and TSR are highest, this results in a tip speed of 102.6 m/s, which is the upper limit of the incompressibility assumption of the fluid at $Ma = 0.3$. Due to the number of simulations required, the mesh had to be limited to below roughly 40M cells. However, even with the geometric scaling further simplifications were required to meet this mesh limit. It was intended to use the same mesh as was employed in Section 7.3.3, but a coarser mesh was ultimately required to keep the cell count reasonable, as well as using a high y^+ wall treatment using fewer, thicker prism layers – this concept is discussed in more detail in Section 3.3.2. While the accuracy of the simulations was likely to suffer due to this coarser mesh, it was required to achieve a mesh size that could be feasibly run the 24 simulations in a reasonable amount of time.

Figure 7.8 shows the specifications of the 3D mesh used in the full-rotor simulations. This includes a 3D view of the domain as well as the mesh refinement zones and related sizes as a function of the blade length, $R = 1.1/4 = 0.275$ m. As described previously, the inlet is a hemisphere to facilitate varying the mean wind direction during simulations to approximate the rotor displacement due to the rotating base. Table 7.8 lists the mesh sizes and details of the prism layers used in the domain, which aimed to match the mesh sizes previously employed in Table 7.6 and Table 3.13 as closely as possible while keeping the cell count reasonable. As previously detailed in Section 3.3.2, prism layers around the rotor are normally controlled by the requirement of keeping the dimensionless wall distance (y^+) below 1, however for a high y^+ wall treatment which employs a wall function to approximate the sub-scale behaviour, it can instead be kept to $30 < y^+ < 100$. When testing at 1:4 scale: based on the highest free stream velocity of 20.6 m/s and the average chord length of 0.0223 m, a near-wall prism layer thickness of 0.005 m corresponds to a y^+ of 41 according to Equation 3.22.

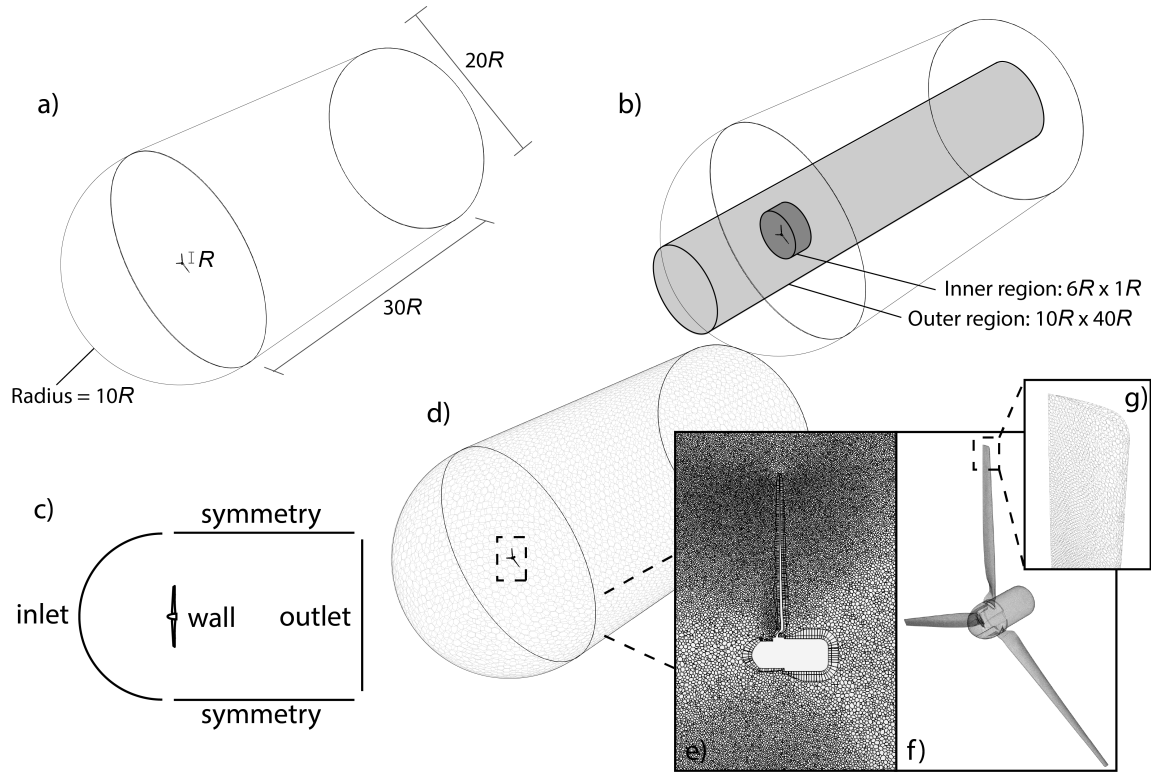


Figure 7.8: Mesh properties of full-rotor CFD simulations: a) domain dimensions; b) mesh refinement regions; c) domain boundary specifications; d) meshed domain; e) mesh around rotor; f) meshed rotor; g) meshed blade tip

	Region	Ratio	Value (m)
	Base size	B	0.2600
	Outer region	$0.250B$	0.0650
	Inner region	$0.100B$	0.0260
	Blade surface target size	$0.070B$	0.0182
	Blade surface minimum size	$0.035B$	0.0091
	Blade trailing edge target size	$0.030B$	0.0078
	Blade trailing edge minimum size	$0.010B$	0.0026
	Blade leading edge target size	$0.003B$	0.0008
	Blade leading edge minimum size	$0.0015B$	0.0004
	Prism layer thickness	–	0.005
	Number of prism layers	2 (high y^+ wall treatment)	
	Total meshed size	26.4M cells	

Table 7.8: Final mesh dimensions as a function of an optimized base size B , with reference to the domain regions shown in Figure 7.8

Using the stiffness of the numerical 1.1 m wind turbine model developed in Chapter 6, as summarized in Table 6.13, and the aerodynamic loads predicted by OpenFAST in Section 7.2.3, a range of expected base rotations can be generated. Using a base stiffness of 1129 Nm/rad (see Equation 6.16), a mean rotor thrust from the 9 m/s wind field of 114 N (see Figure 7.3) which results in an overturning moment of 217 Nm given the hub height of 1.9 m. Given this overturning moment, an eccentric mass moment of -7 Nm, and ignoring the tower drag, the mean base rotation of the turbine under the highest speed test case is approximately 0.19 rad. Conversely, assuming no wind loading on the turbine, the eccentric mass moment alone causes a minimum base rotation of approximately -0.006 rad. Therefore it can be said across all the proposed test cases the base rotation will range from roughly -0.01 rad to 0.25 rad, given the range of wind speed angles that must be simulated in the CFD tests. Table 7.9 lists the eight wind angles that will be tested in the CFD simulations. These wind angles are applied by rotating the main direction of the wind according to Figure 7.9 in the along wind-vertical plane.

Wind angle α_U (rad)
-0.01
0.00
0.01
0.05
0.10
0.15
0.20
0.25

Table 7.9: Eight wind angles tested in CFD simulations

Each URANS simulations was run for a total of 1 s at 0.001 s time steps with five iterations per time step. This time step length meets the Courant number limitation in Equation 3.28 for the free stream flow. The total simulation lengths were limited to 1 s due to the computational requirements. For each of the 24 wind case and rotation angle combinations, the respective simulation required 5000 iterations. On a Compute Canada 32-core 192 GB supercomputing node, these 5000 iterations take approximately 36 hours to run, for a total of 36 days of supercomputing time to perform the full set of simulations.

7.3.5 CFD results

The following CFD simulations were performed as explained in Section 7.3.4: eight sets of LES simulations were run with at base rotations of -0.01, 0.00, 0.01, 0.05, 0.10, 0.15, 0.20, and 0.25 radians about the across-wind axis. Each rotor angle was tested at the three wind conditions detailed in Table 7.1, and each of these 24 simulations were run for 0.5 s at 0.001 s time steps with 5 iterations per time step. Simulations were performed at 1:4 scale while maintaining Re- and TSR-similitude.

The time histories of the following data were recorded from these CFD simulations: the base edgewise and flapwise force on each turbine blade (i.e. the loads at the roots according to

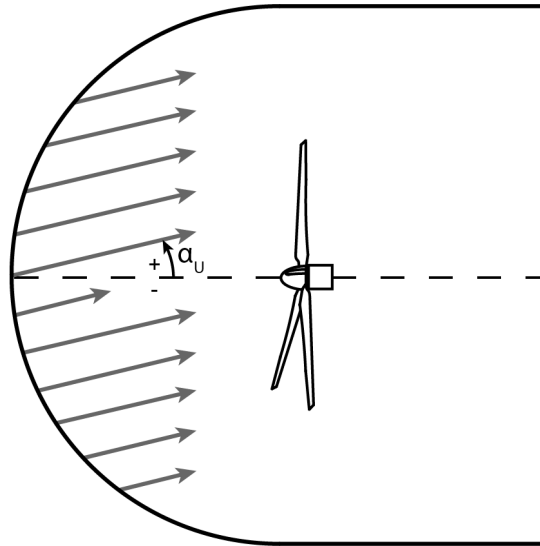


Figure 7.9: Explanation of changing angle of wind in CFD simulations, note that upwards is positive

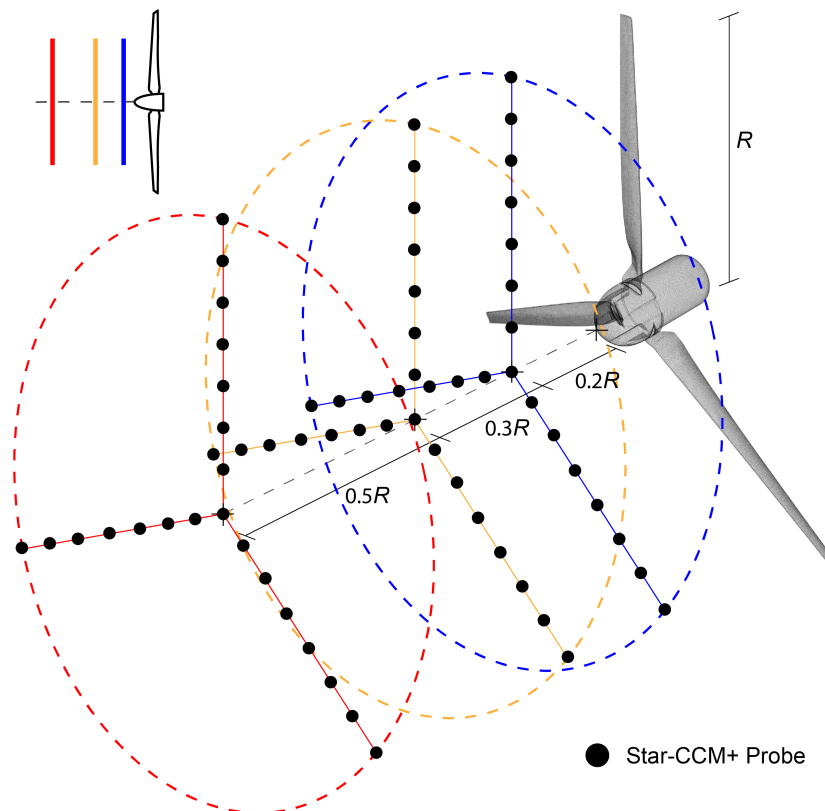


Figure 7.10: Locations of nodes used to record wind speeds in Star-CCM+. This figure shows seven nodes per blade for clarity, but the simulations had 11 per blade plus one at the hub

the blade frame of reference shown in Figure 2.3); the drag, lift, and vertical forces on the nose cone; and the overall thrust and torque on the entire rotor. Additionally, all three wind speed directions at 11 nodes along the length of the upwind projected area of turbine blades and one at the hub were recorded at distances of at $0.2R$, $0.5R$, and $1.0R$ upwind of the turbine, where R is the blade length. These locations are shown in Figure 7.10. Note that these node locations were fixed in all simulations, thus they captured the wind speed at the locations relative to the rotor plane, not the wind direction.

Figure 7.11 shows that the general behaviour of the flow around the rotor is broadly the same for all the reported cases. There is an area of reduced speed in front of the turbine as the flow is diverted around the rotor plane, followed by vortices induced by the rotating blades. The lower TSR cases show a lesser decrease in velocity in the wake, and the area of vorticity is smaller compared to the higher TSR case. Changing the angle between the rotor plane and the wind direction does not significantly change the flow behaviour. 3D plots of the immediate rotor wake are modelled using isosurfaces of the Q -criterion at 10 s^{-1} . Helical turbulence is expected to be seen in operational turbines, which is captured well in these simulations. The lower TSR sims show the helical vortices emanating from the rotor blades remain spaced apart and travel farther downstream, while the high TSR vortices are close enough that they combine into a large zone of vorticity that dissipates quickly. Interestingly, it can be seen that the direction of the wake is almost entirely governed by the rotation of the rotor rather than the wind direction; for the 3.63 m/s case there is almost no change in the 3D shape of the vortices, though the 2D capture shows the wake dissipating into the mean wind speed more quickly since it is no longer directly shielded by the rotor. For the 5.15 m/s case, despite the change in wind angle of 0.2 rad , the direction of the wake remains perpendicular to the rotor plane, though the reduced wind speed perpendicular to the rotor plane can be seen in the larger space between the helical vortices. Overall, the results of Figure 7.11 act as a sanity check for the simulations. The expected helical wake is seen, and the greater rotation speed and suction on the high-TSR rotor serves to demonstrate why the rotor thrust is strongly related to the TSR of the turbine.

The data recorded from the CFD simulations was then exported and two major adjustments were made. Firstly, since the data needs to relate the wind speed with the resulting blade loads, the wind speed data recorded at the nodes shown in Figure 7.10 are not directly related to the blade thrust and torque recorded at that same time step. Instead, those blade loads were caused by the wind that previously passed through the nodes. Thus, the recorded wind speed time histories are shifted forward to align with their approximate respective aerodynamic loads. If U is the mean wind speed of the simulation perpendicular to the rotor plane, d_n is the perpendicular distance between the given node and the rotor plane, and dt is the length of the time steps of the given simulation, then:

$$N_t = \text{round}\left(\frac{d_n}{Udt}\right) \quad (7.2)$$

Here, N_t , rounded to the nearest integer, is the number of time steps the given wind speed time history must be shifted forward relative to the recorded aerodynamic loads.

The second adjustment for the data was to change it from the 1:4 geometric CFD scale to full scale to use as training data for the NN. As defined in Table 7.5, the recorded wind speeds must be divided by $\lambda_V = 4$. The geometric and wind speed scaling factors (λ_L and λ_V) can

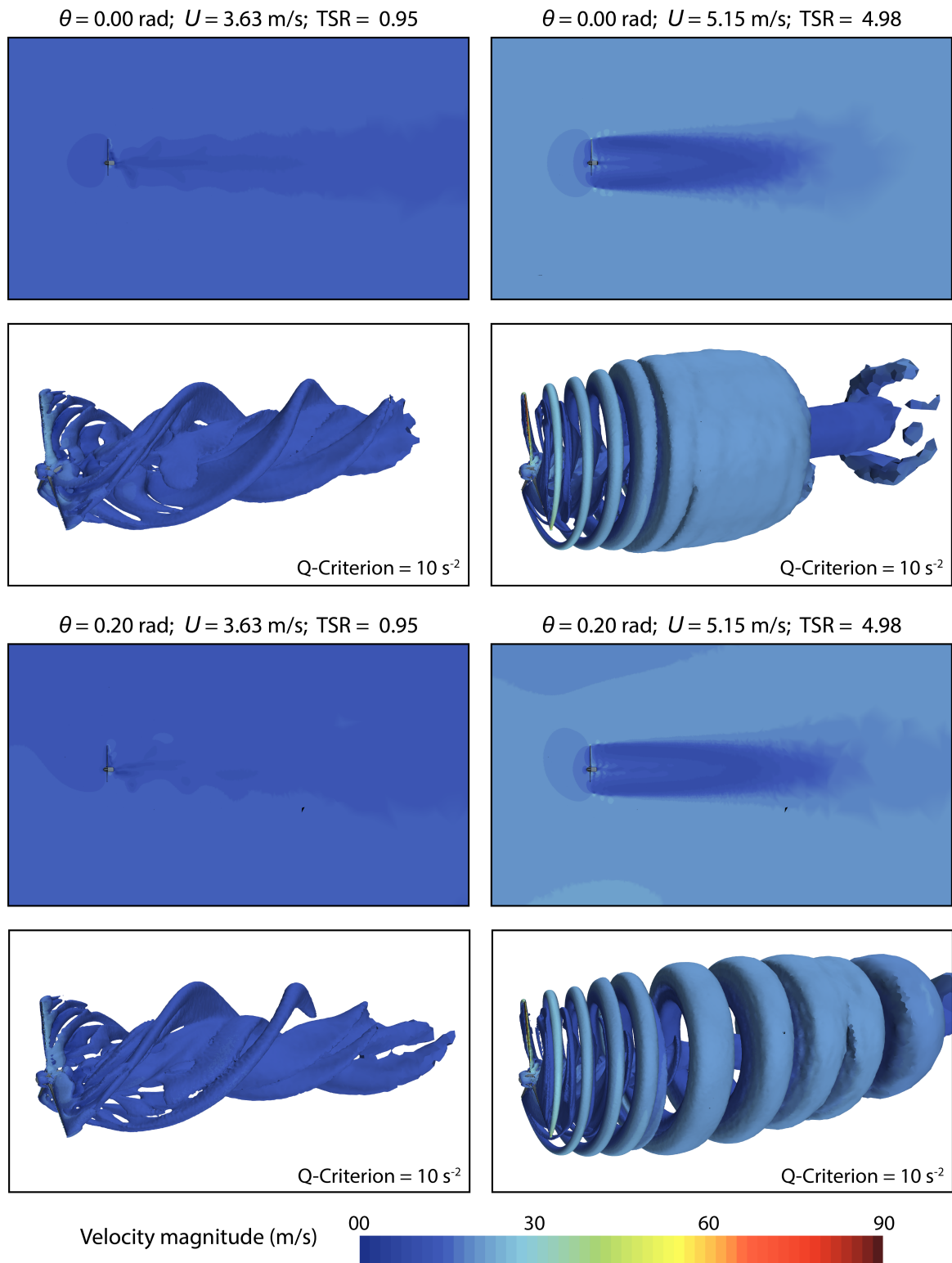


Figure 7.11: Full rotor CFD results, showing the instantaneous velocity and Q-criterion isosurfaces for the rotor at 0.0 rad and 0.2 rad base rotations and wind cases A and C

also be used to derive the thrust and torque scaling factors (λ_T and λ_Q , respectively). These are based on the thrust coefficient equation (Equation 6.21) and the similar torque coefficient equation:

$$C_Q = \frac{Q}{0.5\rho A_R R U^2} \quad (7.3)$$

Here, Q is the total torque on the rotor, roughly equal to the sum of the torque on all three blades. As detailed previously, ρ is the air density, R is the blade length, A_R is the rotor area, and U is the free stream wind speed. Since C_T and C_Q are the same for the CFD-scale (subscript m) and full-scale (subscript f) rotors provided Re and TSR are matched, their respective equations can be used to derive λ_T and λ_Q :

$$\lambda_T = \frac{T_m}{T_f} = \frac{R_f^2 U_f^2}{R_m^2 U_m^2} = \lambda_L^2 \lambda_V^2 = 1 \quad (7.4)$$

$$\lambda_Q = \frac{Q_m}{Q_f} = \frac{R_f^3 U_f^2}{R_m^3 U_m^2} = \lambda_L^3 \lambda_V^2 = \lambda_L = \frac{1}{4} \quad (7.5)$$

As can be seen, the CFD-scale thrust measurements must be divided by $\lambda_T=1$, i.e. they already match the full scale results. Conversely, the CFD-scale torque measurements must be divided by $\lambda_Q=\frac{1}{4}$ to match the equivalent full-scale results. With the exception of the first section of Table 7.10, the tabulated CFD results reported hereafter are adjusted to match full-scale using these factors.

The mean results from each simulation are collected in Table 7.10, while a sample of the data as well as a link to a digital repository of the full data set is reported in Appendix A. The results match what would be expected, including the following observations: firstly, it can be seen that the recorded thrust and torque both increase in all cases as the mean wind speed increases from 3.63 m/s to 5.15 m/s, but there is a much bigger increase when the TSR increases from 1.08 to 4.98. Secondly, it appears that the effects of across-wind and vertical wind on the thrust and torque are relatively minor, and that the main impact of the base rotation is to reduce the thrust and torque since the relative wind speed perpendicular to the rotor plane decreases as the base rotation increases. Finally, the response recorded at each blade is almost identical, as expected from the uniform wind profile.

Based on the results at a base rotation of 0.0 rad from Table 7.10, the thrust and torque coefficient of the rotor can be calculated for each wind case, and compared to previous experimental values. The thrust coefficient (C_T) can be calculated using Equation 6.21 while the torque coefficient (C_Q) can be calculated using Equation 7.3. Table 7.11 compares these coefficients to those from Shirazadeh et al. (2021) (wind case A and B) and Refan (2009) (wind case C), where available. It can be seen that the CFD and experimental results match well for thrust prediction, achieving an error of $<10\%$. Note that it was chosen to ignore the drag on the hub in this study to match the OpenFAST simulations more closely, however consideration of this additional drag would certainly reduce the thrust discrepancy seen here. Conversely, the matching between torque predictions is significantly less precise, with an error of $>30\%$. This torque discrepancy is likely due to the accuracy of predicting lift versus drag loads on the turbine airfoils. As discussed and seen in Chapter 3, numerical methods are generally more effective at predicting airfoil lift than airfoil drag at high angles of attack. Since the lift generally

SIMULATION SCALE											
Base rotation (rad)	Mean wind speed (m/s)	TSR	Mean rotor axis wind speed (m/s)			Mean thrust (N)			Mean torque (Nm)		
			Blade 1	Blade 2	Blade 3	Blade 1	Blade 2	Blade 3	Blade 1	Blade 2	Blade 3
-0.01	14.52	0.95	14.36	14.35	14.43	1.31	1.33	1.36	0.014	0.014	0.014
	20.6	1.08	20.63	20.47	20.48	2.90	2.82	2.91	0.031	0.032	0.032
	20.6	4.98	18.50	19.05	18.75	17.51	17.53	17.84	0.381	0.376	0.384
0.00	14.52	0.95	14.41	14.24	14.30	1.30	1.30	1.30	0.014	0.014	0.015
	20.6	1.08	20.33	20.32	20.02	2.84	2.90	2.87	0.031	0.031	0.033
	20.6	4.98	18.67	18.77	18.71	17.73	17.45	17.55	0.389	0.383	0.385
0.01	14.52	0.95	14.22	14.41	14.28	1.31	1.33	1.33	0.015	0.015	0.014
	20.6	1.08	20.33	20.03	20.57	2.85	2.83	2.87	0.032	0.032	0.032
	20.6	4.98	19.59	18.73	18.78	17.90	17.66	17.72	0.385	0.381	0.384
0.05	14.52	0.95	14.66	14.16	14.33	1.32	1.29	1.29	0.014	0.014	0.014
	20.6	1.08	19.96	19.65	20.20	2.87	2.89	2.86	0.032	0.031	0.032
	20.6	4.98	19.48	19.23	19.17	17.89	17.70	17.42	0.384	0.384	0.384
0.10	14.52	0.95	14.32	14.28	14.08	1.30	1.35	1.28	0.014	0.014	0.014
	20.6	1.08	20.14	20.03	20.19	2.86	2.85	2.87	0.032	0.032	0.032
	20.6	4.98	18.94	19.20	19.03	17.62	17.06	17.47	0.381	0.374	0.380
0.15	14.52	0.95	14.08	13.90	14.30	1.29	1.29	1.30	0.014	0.014	0.014
	20.6	1.08	20.00	20.06	20.38	2.82	2.85	2.78	0.031	0.031	0.031
	20.6	4.98	18.96	18.84	18.88	17.05	17.03	17.40	0.374	0.377	0.374
0.20	14.52	0.95	14.16	14.07	13.82	1.25	1.30	1.27	0.014	0.014	0.014
	20.6	1.08	20.13	19.98	19.76	2.76	2.77	2.76	0.031	0.031	0.031
	20.6	4.98	18.65	18.61	18.56	17.08	17.16	16.91	0.377	0.369	0.365
0.25	14.52	0.95	13.65	13.88	14.21	1.27	1.23	1.24	0.014	0.014	0.014
	20.6	1.08	19.76	19.67	19.58	2.68	2.70	2.66	0.029	0.030	0.030
	20.6	4.98	18.61	18.73	18.73	16.60	16.33	16.48	0.359	0.357	0.356

FULL SCALE											
Base rotation (rad)	Mean wind speed (m/s)	TSR	Mean rotor axis wind speed (m/s)			Mean thrust (N)			Mean torque (Nm)		
			Blade 1	Blade 2	Blade 3	Blade 1	Blade 2	Blade 3	Blade 1	Blade 2	Blade 3
-0.01	3.63	0.95	3.59	3.59	3.61	1.31	1.33	1.36	0.058	0.058	0.058
	5.15	1.08	5.16	5.12	5.12	2.90	2.82	2.91	0.126	0.128	0.128
	5.15	4.98	4.62	4.76	4.69	17.51	17.53	17.84	1.524	1.506	1.534
0.00	3.63	0.95	3.60	3.56	3.57	1.30	1.30	1.30	0.058	0.057	0.058
	5.15	1.08	5.08	5.08	5.00	2.84	2.90	2.87	0.124	0.125	0.130
	5.15	4.98	4.67	4.69	4.68	17.73	17.45	17.55	1.556	1.530	1.540
0.01	3.63	0.95	3.56	3.60	3.57	1.31	1.33	1.33	0.058	0.058	0.058
	5.15	1.08	5.08	5.01	5.14	2.85	2.83	2.87	0.127	0.128	0.129
	5.15	4.98	4.90	4.68	4.69	17.90	17.66	17.72	1.542	1.524	1.536
0.05	3.63	0.95	3.67	3.54	3.58	1.32	1.29	1.29	0.058	0.058	0.057
	5.15	1.08	4.99	4.91	5.05	2.87	2.89	2.86	0.128	0.126	0.129
	5.15	4.98	4.87	4.81	4.79	17.89	17.70	17.42	1.538	1.537	1.536
0.10	3.63	0.95	3.58	3.57	3.52	1.30	1.35	1.28	0.057	0.057	0.057
	5.15	1.08	5.03	5.01	5.05	2.86	2.85	2.87	0.128	0.126	0.128
	5.15	4.98	4.73	4.80	4.76	17.62	17.06	17.47	1.523	1.497	1.522
0.15	3.63	0.95	3.52	3.48	3.58	1.29	1.29	1.30	0.057	0.057	0.056
	5.15	1.08	5.00	5.01	5.10	2.82	2.85	2.78	0.124	0.125	0.125
	5.15	4.98	4.74	4.71	4.72	17.05	17.03	17.40	1.495	1.507	1.497
0.20	3.63	0.95	3.54	3.52	3.45	1.25	1.30	1.27	0.055	0.055	0.055
	5.15	1.08	5.03	4.99	4.94	2.76	2.77	2.76	0.125	0.123	0.125
	5.15	4.98	4.66	4.65	4.64	17.08	17.16	16.91	1.507	1.476	1.461
0.25	3.63	0.95	3.41	3.47	3.55	1.27	1.23	1.24	0.055	0.055	0.054
	5.15	1.08	4.94	4.92	4.89	2.68	2.70	2.66	0.118	0.119	0.119
	5.15	4.98	4.65	4.68	4.68	16.60	16.33	16.48	1.435	1.429	1.422

Table 7.10: Mean CFD simulation results. Only the mean wind speed parallel to the rotor plane is reported, recorded at $1.0R$ upwind. A sample of the full results of these simulations are reported in Appendix A

	TSR	0.95	1.08	4.98
Mean rotor-axis wind speed (m/s)		3.57	5.07	5.07
Mean rotor thrust (N)		3.90	8.61	52.73
Thrust coefficient C_T		0.139	0.150	0.917
Experimental C_T		0.21	0.21	0.96
Average OpenFAST C_T		0.084	0.092	0.620
Mean rotor torque (Nm)		0.173	0.379	4.63
Torque coefficient C_Q		0.0055	0.0060	0.0725
Experimental C_Q		–	–	0.105
Average OpenFAST C_Q		0.0102	0.0106	0.0174

Table 7.11: Comparison of the mean thrust and torque coefficients recorded by CFD simulations at a base rotation of 0.0 rad to the experimental value reported by Shirazadeh et al. (2021) and Refan (2009), as well as the average thrust coefficient predicted by the fixed OpenFAST model

governs the thrust while drag governs the torque, one would expect the accuracy of the numerical thrust predictions to exceed the accuracy of numerical torque predictions, especially given the relatively coarse mesh and the simplified high y^+ wall treatment used for these CFD simulations. Table 7.11 shows that these expectations are reflected in the results. However, since only the thrust will be applied to the structural model in the AHS performed in Section 7.4, the discrepancy in the torque prediction should have little impact on the overall simulations while the high accuracy of the predicted thrust is a positive observation.

Table 7.10 also presents the average thrust and torque coefficients predicted by the OpenFAST simulations using the fixed-base turbine model. As can be seen, there are notable differences in predicted values between the BEM-based OpenFAST simulations and the CFD simulations. In particular, in the 4.98 TSR case the CFD simulations match the experimental values much better than the BEM simulations. These results show the notable improvement to accuracy that can be achieved by replacing simplified aerodynamic models with improved, higher-accuracy techniques such as CFD.

A limitation to the collected CFD data that can be seen in Table 7.10 is the reduction of the mean recorded wind speed seen at all angles between the 1.05 and 4.98 TSR cases. This reduction is seen despite the same mean wind speed being inputted into each simulation, which is due to the effect of the flow around the turbine blades. While at the lower TSRs this effect is minimal – for example, at a base rotation of 0.0 rad and 1.08 TSR, the mean recorded wind speed of 5.07 m/s is quite close to the supplied 5.15 m/s – the effect is more significant at higher TSRs. The wind speed was recorded upwind to attempt to minimize this effect, however future simulations should include nodes ever further away from the rotor to record completely unaffected data.

These processed CFD results were supplied to the neural network as input data to train, validate, and test it. The final iteration of each time step of the second half of the CFD simulations were recorded and divided on a per blade basis. Each individual CFD simulation provided three time histories relating wind speed, rotor speed, and base angle to thrust and torque on a

blade, for a total of 72 time histories each consisting of 500 time steps of 0.001 s. In total, the neural networks were provided 43,200 time steps for training and testing.

7.3.6 Neural network specifications

Neural networks are a type of machine learning that connect input and output data. These networks can be trained on existing data to accurately predict outputs from inputs without performing the underlying calculations. A salient example would be a NN that learns to predict aerodynamic loads based on wind speed and blade position, as studied in Chapter 4. Well-trained NNs can achieve accuracy levels of >99% while remaining computationally light, allowing them to act as high-speed surrogate models.

Chapter 4 thoroughly details the development and optimization of a number of neural networks to act as surrogate aerodynamic models of a 5 MW wind turbine blade, trained using data generated in OpenFAST. Readers unfamiliar with NNs are directed to Section 4.2 in particular. The main findings of Chapter 4 are provided here; of all the compared neural networks, the convolutional neural network (CNN) was the most effective at predicting aerodynamic loads when provided with the wind speed and blade position. Accuracies exceeding 99% were seen in this model as well as operational times below 10 ms making it a promising candidate for a surrogate model.

The NN development process is detailed in Section 4.3, but as a summary: training data is organized, normalized, then randomly divided into three categories. 60% of data is supplied as training data to train the network, 20% is supplied as validation data to measure the effectiveness of training, and 20% is supplied as testing data. Training was performed in two steps: firstly, hyperparameter optimization was performed by shallowly training a number of NNs on a subsection of data using select hyperparameter values, allowing the optimal hyperparameter values to be predicted. The Bayesian optimization algorithm (Frazier, 2018) was used for this process. Secondly, the NN is trained using the full training data set using the optimized hyperparameters.

The general architecture of the CNN is shown in Figure 7.12, based on the architecture analyzed in Section 4.6. It consists of a time history window of w time steps provided as input to the network, which is convoluted, activated and pooled into a smaller data set with a wide view of the inputs, which are then connected to the outputs of a single time step. This neural network is built and trained using MATLAB's Deep Learning toolbox, using the *trainNetwork* command specifically. Appendix B includes a well-documented copy of the MATLAB code used in this study to optimize, train, and test the CNN surrogate model.

The time required for optimization increases exponentially with the number of hyperparameters, thus it is preferable to assign conservative values to hyperparameters wherever it is reasonable. In this study, the optimizable hyperparameters include the number of time steps in the input (w), the size and number of filters in the convolution layers, the size of the pooling filters, and the layer sizes for the two fully-connected layers. The remaining hyperparameters are assigned the following conservative values: 80 then 40 optimization iterations were performed respectively over the two optimization rounds; 100 epochs of training were performed for each optimization iteration while 5000 epochs were performed for the final network; 5% of the training data was used for each optimization iteration; max pooling was employed, as are a batch size of 100, a momentum value of 0.990, and an initial learning rate of 10^{-5} for all cases.

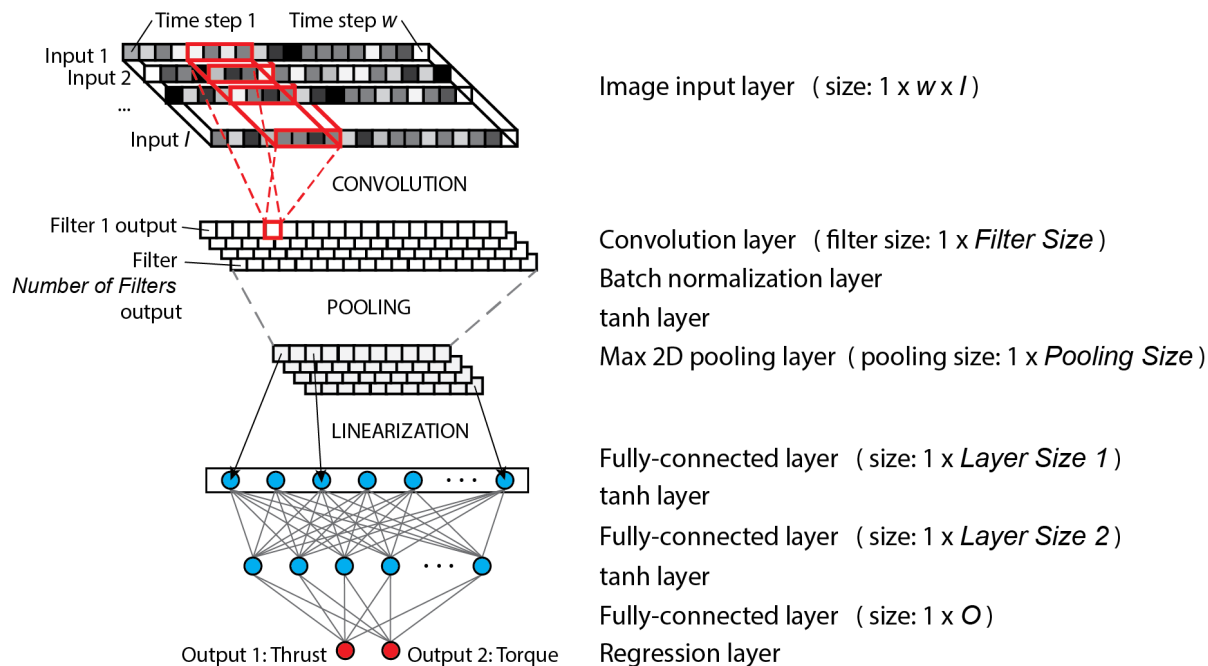


Figure 7.12: CNN architecture detailing the two outputs predicted from the input “image” of time history data. The MATLAB layer types and sizes are specified, including reference to the number of inputs (I) and outputs (O) detailed in Table 7.12 and the optimized hyperparameters (w , Filter Size , Pooling Size , Layer Sizes) detailed in Table 7.13

Two different CNNs were evaluated based on their configurations of the input CFD data as shown in Table 7.12. To predict the thrust and outputs of a single blade using the NN for a given time set, the following were provided as inputs: the rotor speed (rad/s), the base rotation (rad), and a collection of the components of the recorded wind speeds (m/s) relative to the rotor plane such that i_r is parallel to the rotor axis, j_r is the across-wind direction, and k_r is the near-vertical direction perpendicular to i_r . The first data configuration provided 60 wind speed samples per input time step consisting of 12 wind speeds in the i_r direction from each upwind projection of the blade in question at distances of $0.2R$, $0.5R$, and $1.0R$ (see Figure 7.10), as well as the j_r and k_r recordings at the $0.2R$ nodes. The second, smaller data configuration only provided the i_r wind speeds measured at $1.0R$ with the goal of reducing the influence of the rotor effects of the recorded wind speed as well as emphasizing the importance of the TSR to the CNN. The convolution process of the CNN enables supplying a large amount of input data to the network, thus w time steps are provided as an input to predict the thrust and torque outputs of a single time step, resulting in $62w$ inputs for the first data configuration and $14w$ inputs for the second configuration, to predict 2 outputs in each case.

Optimization and training was performed locally over about three hours using a 4GB NVIDIA GTX 1050 Ti GPU. A supercomputer platform could be used to accelerate this process if there was a greater amount of training data, as was done in Chapter 4.

CONFIGURATION 1: Inputs per blade per time step (62w)	Corresponding outputs (2)
12 wind speeds in i_r direction at nodes 0-10 at $0.2R$ upwind (m/s)	Base thrust (N)
12 wind speeds in j_r direction at nodes 0-10 at $0.2R$ upwind (m/s)	Base torque (Nm)
12 wind speeds in k_r direction at nodes 0-10 at $0.2R$ upwind (m/s)	
12 wind speeds in i_r direction at nodes 0-10 at $0.5R$ upwind (m/s)	
12 wind speeds in i_r direction at nodes 0-10 at $1.0R$ upwind (m/s)	
Rotor speed (rad/s)	
Base rotation angle (rad)	

CONFIGURATION 2: Inputs per blade per time step (14w)	Corresponding outputs (2)
12 wind speeds in i_r direction at nodes 0-10 at $1.0R$ upwind (m/s)	Base thrust (N)
Rotor speed (rad/s)	Base torque (Nm)
Base rotation angle (rad)	

Table 7.12: Two configurations of CNN inputs and outputs per blade per time step. i_r , j_r , k_r refer to the along, across, and vertical wind speeds relative to the rotor plane, thus i_r is not parallel to the ground plane when the base rotation $\neq 0$. Nodes 0-12 refer to the nodes upwind of the length of the blade from the hub (1) to the tip (12) which divides the blade in 11 segments of $0.091R$ length, as shown in Figure 7.10

7.3.7 Neural network optimization and training

Using the data partially reported in Appendix A and the MATLAB code detailed in Appendix B, the CNNs were optimized and trained on the two configurations of CFD data. As will be presented in Section 7.4.1, ultimately the second configuration with fewer inputs resulted in the more effective surrogate model, and as such the data presented in this section corresponds specifically to that configuration.

The first stage of training the CNN consisted of finding the optimal hyperparameters using the same two-round process as in Section 4.3.3. In the first round of optimization a wide range of allowable values for each of the hyperparameters was provided, and CNNs were assigned hyperparameter values within these ranges, quickly trained on a small subset of the data, then the accuracy of each was measured. Based on the most effective of the tested hyperparameter combinations, the second round performed another optimization search using a much smaller range centered on the optimal hyperparameter values from the first round. The optimal hyperparameters for this test case are reported in Table 7.13, where it can be seen that compared to the optimized CNN found in Section 4.6.1, the image input to this CNN takes over double the number of time steps at 19 compared to 8 in the previous network. In Chapter 4, due to memory limits in MATLAB and the large amount of training data, there were limits on the maximum number of time steps that could be supplied to the CNN, a limitation that is avoided here due to the smaller amount of data. The final fully-connected layer is also much larger here despite having a significantly smaller number of outputs to predict. Beyond this, the optimized architectures are fairly similar between here and Chapter 4.

The second stage of this process was to train the network on the full set of training and validation data using the optimal hyperparameters. The accuracy of the CNN after training

Hyperparameter:	Convolution filter size	Number of filters	Number of time steps (w)	Pooling size	First fully-connected layer size	Second fully-connected layer size
Round 1 low	1	1	11	1	10	10
Round 1 high	10	10	30	5	500	1000
Round 2 low	7	3	15	1	10	825
Round 2 high	10	10	23	5	50	925
Optimal value	8	9	19	4	11	828

Table 7.13: Hyperparameter optimization results

	Thrust prediction (N)	Torque prediction (Nm)	Overall
RMSE	0.3265	0.0307	–
NRMSE	1.97%	2.03%	2.00%

Table 7.14: Accuracy of the trained neural network for predicted the blade thrust and torque

was completed could be assessed by comparing the true outputs in the testing data set to the predicted outputs generated by the trained CNN when it was supplied the inputs from the testing data. The error between these values was quantified using the root mean square error (RMSE) and its normalized counterpart (NRMSE), as was performed in Chapter 4 – see Equations 4.2 and 4.3. The RMSE and NRMSE were calculated for both the thrust and torque predictions individually, as well as an overall average NRMSE. These are presented in Table 7.14, where it can be seen that the overall accuracy of the network was $\approx 98\%$. The network was approximately equally accurate at predicting the thrust and torque as shown by their respective NRMSEs.

This overall accuracy is quite high and was suitable for applying this trained CNN as a surrogate aerodynamic load model in the artificial HS. However, it should be emphasized that this reported 2.0% error only applies to predicting results that fall within the bounds of the training data. Since only a relatively small amount of CFD data with low variety was able to be generated for this purpose, the robustness of this surrogate model is low, as will be explored below. However, provided that testing generally remains within the bounds of the NN, the improved accuracy of the CFD results over equivalent BEM simulations should result in improved results when the surrogate model is applied in the AHS.

Compared to the CNN trained in Chapter 4, this CNN with a similar number of inputs and fewer outputs had roughly 2.5 times the error (2.0% compared to 0.8%), likely stemming from the significantly smaller amount of training data for this neural network. This network was trained on a few thousand time steps worth of data while the previous CNN had 1000 times that amount. Still, this network managed to achieve a similar level of accuracy to the baseline MLP in Chapter 4, which was also deemed a successful network in that study.

The computation time of the network, that is, the time required to generate one set of out-

puts from one set of inputs, was quantified for this trained network. This was done by predicting the output of the network ten times and averaging the recorded time requirements between them. For this network, an average computation time of 0.0034 s or 3.4 ms was found. These computation times were estimated on the same hardware used for training which had additional non-intensive background processes running. On dedicated hardware, this computation time could be further improved. Recall that this surrogate model predicts the thrust and torque on a single turbine blade and thus it must be run three times every time step during the AHS, though this could be parallelized to avoid tripling the required computation time. Overall, this computation time is slightly longer than ideal – if the network is run in series on this same hardware there is a total calculation time of 0.0102 s, which encroaches on the largest time step allowable for most HS setups (0.01 s is a rough upper limit). However, this computation time could be easily improved in a true hybrid simulation, by using more powerful hardware, parallelizing the calculation, by reducing the number of time steps input into the network (this would reduce network accuracy but increase the computation speed), or by transplanting the trained NN from MATLAB to a lighter, low-level programming environment. For the purposes of this study, this computation time is therefore deemed acceptable.

The surrogate model is not without limitations, however. As stated, due to the low variety of results within the CFD training data, particularly in the input wind speeds, the network has only been trained to accurately predict rotor thrust near wind speeds of 3.63 and 5.15 m/s. This is a direct result of the short duration of the CFD simulations, meaning that fluctuations due to turbulence were limited. Figure 7.13 plots the range of training data generated by the CFD simulations – as can be seen, data only exists at very specific values of wind speeds. As a result, a reduced turbulence intensity of 1.1% was used in the wind time histories for the artificial HS, as described in Section 7.2.1. If the turbulence intensities of near 10% were instead used to match respective wind turbine tests, the surrogate model would be provided wind speeds below 3.63 m/s and above 5.15 m/s and the accuracy of the predictions by the surrogate model would decrease well below the 98% predicted by the testing data.

The CNN trained on CFD data can thus be used a surrogate model for the artificial HS. By integrating the NN directly into the HS process, high-accuracy in-the-loop aerodynamic

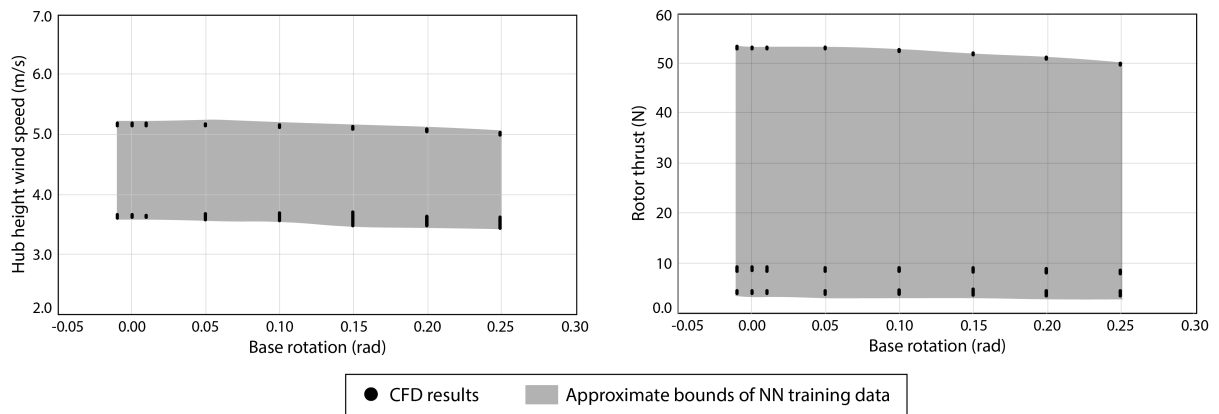


Figure 7.13: Range of data points generated by CFD testing – this likewise represents the range of data the surrogate model has been trained to predict

Case ID	Aerodynamic model	Wind speed (m/s)	Turbulence intensity	Tip speed ratio	Rotor speed (rad/s)
P1A	Precalculated	3.63	1.1%	0.95	3.14
P1B	fixed model	5.15	1.1%	1.08	5.06
P1C	(Section 7.2.3)	5.15	1.1%	4.98	23.32
P2A	Precalculated	3.63	1.1%	0.95	3.14
P2B	rotating model	5.15	1.1%	1.08	5.06
P2C	(Section 7.2.3)	5.15	1.1%	4.98	23.32
P3A	Precalculated	3.63	1.1%	0.95	3.14
P3B	rot. model w/ PTMD	5.15	1.1%	1.08	5.06
P3C	(Section 7.2.3)	5.15	1.1%	4.98	23.32
SMA	Surrogate model	3.63	1.1%	0.95	3.14
SMB	CFD-NN	5.15	1.1%	1.08	5.06
SMC	(Section 7.3)	5.15	1.1%	4.98	23.32

Table 7.15: The twelve artificial HS test cases

modelling can be performed in this novel HS framework to enable and improve the usefulness of HS for studying wind-loaded structures.

7.4 Artificial hybrid simulations

7.4.1 Description

This section reports the artificial HS of the rotating-base residential wind turbine equipped with a MR-controlled semi-active TMD. Table 7.15 summarizes the different hybrid simulations performed here; the four frameworks (see Figure 7.2) are employed for each of the three wind cases for a total of twelve AHS. The case IDs are defined as follows: the first two characters refers to the aerodynamic models (“P1”, “P2”, “P3” for the three pre-calculated models, “SM” for the surrogate model), while the third character refers to the wind case (“A”, “B”, “C”) as defined in Table 7.1.

As detailed previously, an artificial HS differs from a true HS in that the experimental substructure of the true HS is replaced by an additional numerical substructure. Thus, artificial HS avoid the experimental delays and errors that are very impactful on true HS, though it was attempted to minimize this simplification by introducing artificial numerical approximations of these errors to the information passed to and from the STMD virtual “experimental” substructure. Artificial delays were added to the voltage supplied to MR damper in the STMD as well as normally-distributed error applied to the recorded restoring force of the STMD, as described in Section 7.2.4.

The artificial hybrid simulations are run in MATLAB using the code reported in Appendix C. This code allows the user to specify which framework and wind speed time history to use, then returns the time history of the response of the rotating-base turbine. The numerical

algorithm used to solve the equation of motion of the structure is the Chen-Ricles algorithm (Chen et al., 2009), an explicit algorithm designed for time-sensitive testing such as HS. This algorithm was previously employed in Chapter 5 (see Section 5.2.4).

The dynamic simulation process involves, for each time step of the analysis:

- estimating the structural response of the turbine based on the response at the previous time step;
- estimating the response of the STMD based on the response at the previous time step;
- adjusting the future voltage supplied to the STMD if the controller calls for it;
- calculating the restoring force on the turbine based on the difference in response between the turbine and STMD;
- determining and applying the aerodynamic loads to the turbine;
- updating the predicted structural response considering the aerodynamic and restoring forces on the turbine;
- continuing to the next time step then repeating.

The only of these processes that differs greatly between the twelve test cases is the determination of the aerodynamic loads. For all nine AHS that employ pre-calculated aerodynamic loads, the rotor thrust is supplied as a time history to the code, and that load is applied at a given time step. Conversely, determining the aerodynamic loads using the surrogate model is a more demanding process.

The process to predict the aerodynamic thrust loads on the rotor via the surrogate model while using the proposed HS framework is reported in Figure 7.14. To summarize, firstly the TurbSim-generated .bts wind field time history file for the given wind case is loaded into MATLAB – this is the same wind field file used in the OpenFAST simulations to generate the pre-calculated aerodynamic loads. During the AHS, at each time step for a given turbine blade, the location of 12 to 36 blade nodes (depending on the input configuration) are identified based on the base rotation and blade azimuth. These nodes consist of three sets of 12 nodes along the upwind projected blade area at distance of $0.2R$, $0.5R$, and $1.0R$, as shown in Figure 7.10. The global wind speed components at these nodes can then be extracted, shifting the time step as described in Equation 7.2 to relate the recorded wind speed and predicted thrust as strongly as possible. These global wind components must then be transformed to the rotor-axis frame of reference using the following equations:

$$i_r = i \cos \alpha - k \sin \alpha \quad (7.6)$$

$$j_r = j \quad (7.7)$$

$$k_r = i \sin \alpha + k \cos \alpha \quad (7.8)$$

Here, i , j , and k are the along-wind, across-wind, and vertical components of the wind speed, where i is parallel to the mean wind direction; i_r , j_r , and k_r are the respective components relative to the rotor-axis; and α is the base rotation. These rotor-axis wind speeds are arranged

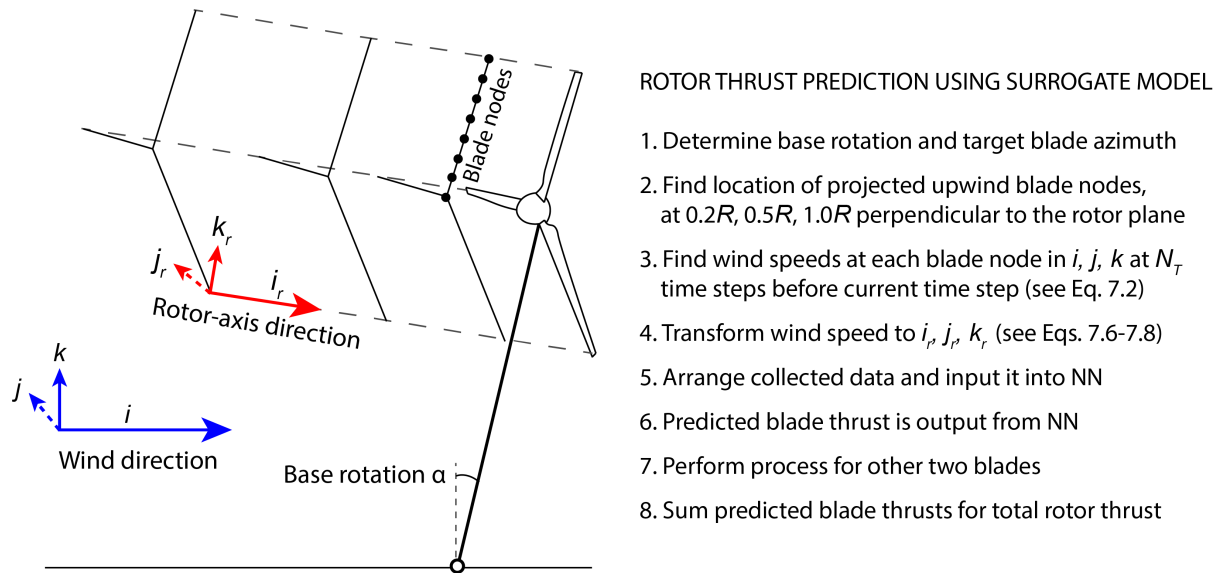


Figure 7.14: Explanation of process used to predict aerodynamic thrust using the surrogate model. Additional references for this process include Figures 2.3 and 7.10, Equations 7.2 and 7.6-7.8, and Appendix C

to match the NN input format described in Table 7.12 and shown in Appendix A, then input into the trained surrogate model which returns the predicted blade thrust. This process is performed for all three blades, and total the rotor thrust can be taken as the sum of these blade thrusts – recall that the drag on the hub and nacelle are ignored here to match the OpenFAST simulations as closely as possible. The thrust predicted by the surrogate model is then applied to the wind turbine as it is when using the pre-calculated loads.

As discussed in Section 7.3.7, 0.01 s represents an upper limit on the allowable time step durations for HS, and is also the approximate speed of the trained surrogate aerodynamic model without any further optimization (improved hardware, parallelization, etc.). As such it is logical that the AHS using the surrogate model be run at a time step of 0.01 s to approximate the time-limitations present in a true HS. Thus, 18,000 time steps are run for the 180 s simulation which takes roughly that much time to run. Conversely, the use of the pre-calculated aerodynamic loads are orders of magnitudes faster to run even compared to the surrogate model – this is partly why this technique is so widely used in existing HS. If the 180 s simulations using the pre-calculated loads are run at time steps of 0.001 s for a total of 180,000 time steps, this simulation only takes seconds to complete. The lower time step means that the numerical simulation will be more stable as well as capturing higher frequency behaviour more effectively. As a result, the nine AHS that employ pre-calculated aerodynamic loads are run at this smaller time step to benefit from this advantage the existing framework has over the proposed surrogate model-based framework.

7.4.2 Results

All twelve AHS were performed and the results were collected for analysis, allowing the accuracy of the surrogate model to be evaluated as well as the proposed HS framework that employs

it. This was performed using two different CNNs trained on different configurations of input data as described in Table 7.12. The first configuration provided 60 wind speed measurements per input time step w , while the second configuration provided only $14w$. Both models were able to achieve an average NRMSE of $\leq 2\%$ during network testing. The mean rotor thrust loads predicted by each model for each wind case are reported in Table 7.16 and compared to the mean thrust predicted by the CFD simulations. As can be seen, while the surrogate model trained on the first data configuration matched the CFD thrust slightly better than the second configuration in the 0.95 TSR case, it strongly underestimates the rotor thrust in the 4.98 TSR case. This is likely due to importance that the TSR input plays on the overall thrust; when a significant number of wind speed data points are provided to the network it waters down the relative weight of the TSR to the NN. The second model emphasizes the role of the TSR by reporting fewer wind speed measurements. As a result the surrogate model trained on the second, smaller data configuration was used for analysis and all reported results hereafter.

With the optimal surrogate model identified, Figure 7.15 plots the recorded time history of the rotor thrust generated for each surrogate model test against the pre-calculated aerodynamic loads previously reported in Figure 7.3 and the mean thrust predicted by the wind tunnel testing by Shirazadeh et al. (2021) and Refan (2009) (see Table 7.11). As can be seen, the surrogate model-generated and pre-calculated thrust time histories both underestimate the thrust compared to the experimental means, but the surrogate model predictions match much more closely, particularly as thrust increases and so does the resulting base rotations (and, in turn, the impact of aeroelastic fidelity on the results). It can also be seen that the deviation of the surrogate model-predicted thrust is less than the pre-calculated cases for case C, indicating the surrogate model may overweigh the TSR as an input due to the small range of training data. It can be concluded that the surrogate model has improved prediction of the mean thrust compared to pre-calculated loads, but whether the difference in fluctuating components is an improvement cannot be directly evaluated.

To further study the accuracy of the surrogate model, the mean and standard deviations of the thrust time histories from all twelve AHS are reported in Table 7.17. This confirms the previous observations that the mean thrusts predicted by the surrogate model (2.68, 7.80, and 49.51 N respectively for the three wind combinations) approach the values predicted by the CFD simulations compared to the pre-calculated mean thrusts using the rotating-with-PTMD model (2.39, 5.44, and 32.22 N). Recall that it was shown that the CFD results match experimental data better than the pre-calculated thrust values for the 4.98 TSR case at minimum. Overall, it can be said that the surrogate models are more accurate at predicting the mean rotor

	No. of inputs per time step	Optimal no. of time steps w	Total no. of inputs	Mean rotor thrust (N)		
				3.63 m/s 0.95 TSR	5.15 m/s 1.08 TSR	5.15 m/s 4.98 TSR
CNN using 1st config	62	24	1488	2.99	6.29	34.41
CNN using 2nd config	14	19	266	2.68	7.80	49.51
CFD simulations	–	–	–	3.97	8.64	52.86

Table 7.16: Comparison of mean rotor thrusts predicted by two surrogate models in AHS. The surrogate models differ according to input data configurations, as shown in Table 7.12

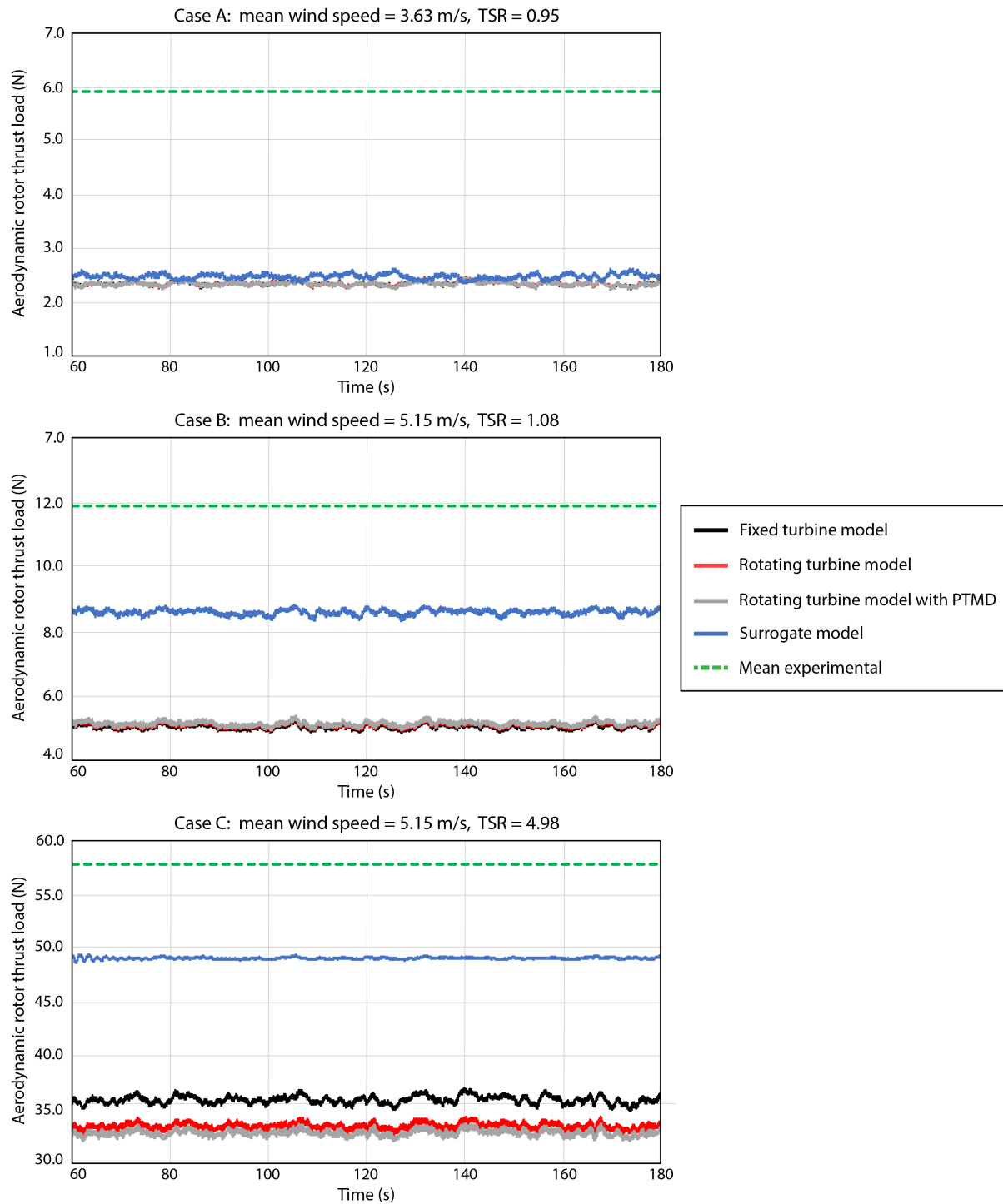


Figure 7.15: Pre-calculated and surrogate model-generated rotor thrust time histories for the three wind cases, compared to the mean experimental thrust predictions (see Table 7.11)

thrust but less effective at capturing its fluctuating component – the cause of these limitations are discussed in detail in Section 7.4.4.

The time histories of the base rotation of the turbine for all twelve AHS simulations are

WIND CASE A: 3.63 m/s 0.95 TSR				
AHS results	Fixed	Pre-calculated		Surrogate model
		Rotating	Rot. w/ STMD	
Mean hub height wind speed (m/s)	3.62	3.62	3.62	3.62
Mean rotor thrust (N)	2.36	2.35	2.35	2.48
Standard dev. of rotor thrust (N)	0.04	0.04	0.04	0.05
Mean base rotation (rad)	-1.02E-03	-1.03E-03	-1.03E-03	-8.07E-04
Standard dev. of base rotation (rad/s)	9.13E-05	9.05E-05	9.09E-05	1.56E-04

WIND CASE B: 5.15 m/s 1.08 TSR				
AHS results	Fixed	Pre-calculated		Surrogate model
		Rotating	Rot. w/ STMD	
Mean hub height wind speed (m/s)	5.14	5.14	5.14	5.14
Mean rotor thrust (N)	5.07	5.13	5.16	8.61
Standard dev. of rotor thrust (N)	0.07	0.07	0.07	0.08
Mean base rotation (rad)	4.73E-03	4.82E-03	4.87E-03	1.07E-02
Standard dev. of base rotation (rad/s)	1.89E-04	1.78E-04	1.84E-04	2.66E-04

WIND CASE C: 5.15 m/s 4.98 TSR				
AHS results	Fixed	Pre-calculated		Surrogate model
		Rotating	Rot. w/ STMD	
Mean hub height wind speed (m/s)	5.14	5.14	5.14	5.14
Mean rotor thrust (N)	35.96	33.53	32.90	49.12
Standard dev. of rotor thrust (N)	0.40	0.30	0.31	0.07
Mean base rotation (rad)	5.67E-02	5.20E-02	5.15E-02	7.89E-02
Standard dev. of base rotation (rad/s)	1.18E-03	7.29E-04	9.71E-04	2.32E-04

Table 7.17: Mean results of the twelve AHS simulations

shown in Figure 7.16, the means and standard deviations of which are also reported in Table 7.17. As can be seen, stemming from the higher mean and differences in deviation of the surrogate model-generated thrust, the resulting base rotations of the turbine are greater on average compared to the values resulting from the pre-calculated loads. As with the thrust time histories, there is greater variance in the low TSR case and lower variance in the 4.98 TSR case. Since the relationship between thrust and base rotation is so strongly correlated (with the presence of the STMD as the other major contributor to the response of the turbine), the general trends in the base response are the same as those seen in the predicted thrust time histories.

Additional observations from the reported thrust and base rotation time histories include:

- It can be seen that the all predicted base rotations lie within the chosen range of -0.01 to 0.25 rad for the CFD simulations. As a result, the surrogate model was never exposed to a base rotation that exceeded the bounds of its training data, which was the intended goal of the chosen range. However, it was seen that the chosen values for generating the testing set were overly conservative as the base rotation of the turbine equipped with the STMD was never seen to surpass 0.10 rad. Thus, the CFD simulations performed at a base rotation of 0.15 rad and beyond were not strictly necessary and that computation

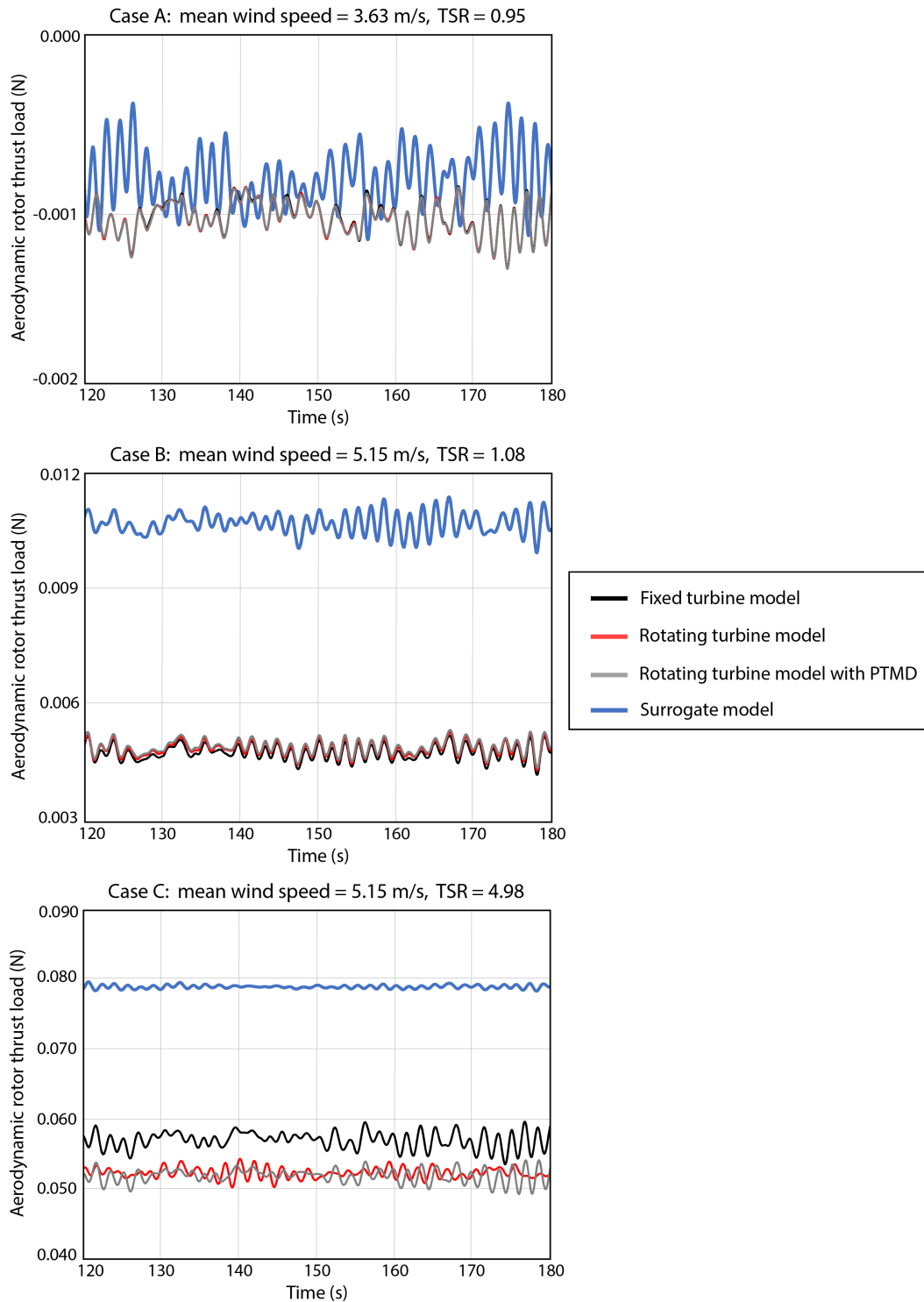


Figure 7.16: Base rotation time histories generated by the AHS using the surrogate model and pre-calculated aerodynamic loads. While there is no target time history to compare against, referencing Figure 7.15 shows the improved accuracy of the prediction using the surrogate model

time could have been directed to other wind speeds or smaller base rotations seen by the simulations for a more robust surrogate model.

- The effectiveness of the three numerical OpenFAST models of increasing fidelity used to pre-calculate the rotor thrust – the fixed, rotating, and rotating-with-PTMD models, can be contrasted by studying Figure 7.16. As can be seen, at the low TSR cases where the base rotation does not exceed ± 0.01 rad, there is very little difference in the predicted base rotation response. However, at the 4.98 TSR case where thrust loads are much higher, there is a 10% difference in the predicted mean base rotation between the fixed and rotating-with-PTMD models. This is a logical result given that whether base rotations are considered or not means very little when the turbine never rotates much away from its fixed position. The high-TSR results however support the importance of modelling aeroelasticity with as much fidelity as possible in this form of testing – if the proposed surrogate model-based framework is unavailable, using the highest-fidelity numerical model for load pre-calculation is encouraged.

To investigate the effectiveness of the STMD in these simulations, a set of nine numerical simulations were performed of the residential rotating-base wind turbine without the STMD under the pre-calculated/surrogate model thrusts using the same MATLAB code. Only nine simulations were performed as the pre-calculated loads derived from the rotating-with-PTMD model were excluded since the turbine in question is no longer equipped with a TMD. Without the presence of the STMD, these simulation no longer have the artificial “experimental” sub-structure, meaning they can not be described as AHS and are simply numerical simulations. The mean responses of the TMD-less simulations as well as the percent improvement offered by the STMD in the equivalent simulation are reported in Table 7.18. The percent improvement is calculated as:

$$\text{Percent improvement} = \frac{|\sigma_{STMD} - \sigma_{noTMD}|}{\sigma_{STMD}} \quad (7.9)$$

Here, σ_{STMD} is the standard deviation of the base response of the turbine for given simulation where it was equipped with the STMD, while σ_{noTMD} is the standard deviation of the base response for the equivalent simulation without the TMD.

The results reported in Table 7.18 show that when the turbine experiences greater deflections, the STMD reduces fluctuations effectively despite its small mass ratio of 1%, thanks to the additional control and efficiency offered by the semi-active MR damper-based control system. When the STMD was most effective, fluctuations were reduced in the order of 15% for the simulations. However, it is known that TMDs have limited effectiveness when motion is small, meaning that the vibration reduction effect was quite small for other cases. Considering the very low turbulence in the wind conditions studied here, it is understandable that the overall effective of the STMD would be lower than predicted by Chapter 5 and Section 6.4. Further, when comparing the STMD effectiveness between pre-calculated and surrogate model cases, the STMD is more effective under the more-variable thrust time history predicted by the surrogate model for wind case A, while it is more effective under the more-variable pre-calculated thrust time histories for wind case C. This emphasizes the importance of correctly capturing the dynamic behaviour of the turbine through correct aeroelastic modelling, as the effectiveness of the STMD will vary based on the predicted response.

7.4.3 Improvements offered by the proposed framework

Overall, this study presents an imperfect analysis of the proposed HS framework that incorporates a CFD-trained surrogate model for higher-accuracy, in-the-loop aerodynamic load calculation. As such, the positive observations of the effectiveness of the framework will be

WIND CASE A: 3.63 m/s 0.95 TSR – Turbine without STMD			
AHS results	Pre-calculated		Surrogate model
	Fixed	Rotating	
Mean hub height wind speed (m/s)	3.59	3.59	3.59
Mean rotor thrust (N)	2.36	2.36	2.47
Standard dev. of rotor thrust (N)	0.04	0.04	0.05
Mean base rotation (rad)	-1.01E-03	-1.02E-03	-8.25E-04
Standard dev. of base rotation (rad/s)	8.54E-05	8.46E-05	1.35E-04
% reduction in std of base rotation by STMD	6.9%	7.0%	15.7%

WIND CASE B: 5.15 m/s 1.08 TSR – Turbine without STMD			
AHS results	Pre-calculated		Surrogate model
	Fixed	Rotating	
Mean hub height wind speed (m/s)	5.14	5.14	5.14
Mean rotor thrust (N)	5.07	5.12	8.61
Standard dev. of rotor thrust (N)	0.07	0.07	0.08
Mean base rotation (rad)	4.73E-03	4.82E-03	1.07E-02
Standard dev. of base rotation (rad/s)	1.83E-04	1.73E-04	2.57E-04
% reduction in std of base rotation by STMD	2.9%	2.8%	3.2%

WIND CASE C: 5.15 m/s 4.98 TSR – Turbine without STMD			
AHS results	Pre-calculated		Surrogate model
	Fixed	Rotating	
Mean hub height wind speed (m/s)	5.14	5.14	5.14
Mean rotor thrust (N)	36.02	33.58	49.12
Standard dev. of rotor thrust (N)	0.37	0.29	0.07
Mean base rotation (rad)	5.68E-02	5.27E-02	7.89E-02
Standard dev. of base rotation (rad/s)	1.04E-03	8.36E-04	2.24E-04
% reduction in std of base rotation by STMD	13.1%	12.8%	3.5%

Table 7.18: Summary of nine simulations of the residential wind turbine without STMD, including the percent reduction in the base rotation standard deviation compared to the equivalent simulation equipped with the STMD

highlighted, while the limitations of this study will be explored in the following section.

The original goal for this study was to directly validate the AHS against the wind tunnel tests presented in Chapter 6, but limitations in the damping systems of the physical turbine model meant that the final wind tunnel tests had no artificial experimental substructure for the AHS. As a result, it can not be explicitly proven the degree to which (if any) that the new HS framework employing the CFD-based surrogate model is an improvement over the existing, pre-calculated technique. However, the following observations from this chapter strongly support that this new framework offers or has the potential to offer a notable improvement over the previous state-of-the-art:

- The blade thrust and torque coefficients measured in the single-blade CFD simulations were compared to equivalent BEM-based numerical values and wind tunnel experimental values in Figure 7.7. In these simulations, it was seen that the lift coefficients had equal or improved matching to the experimental results in the CFD simulations compared to the BEM simulations. Further, the drag coefficient, which is generally more challenging to capture using simplified numerical methods, more closely matched the CFD simulations compared to the BEM simulations. This supports the general consensus of the improved aerodynamic simulation capabilities of CFD versus simplified numerical methods.
- Table 7.11 compares the mean thrust and torque coefficients predicted by the full-rotor CFD simulations to those predicted by the BEM-based OpenFAST simulations detailed in Section 7.2.3 as well as existing experimental values. This table shows that the thrust predicted by the CFD simulations matches quite closely to the experimental value, and while the CFD torque coefficient has some amount of error compared to the experimental value, it still matches more closely than the BEM value. This serves as a demonstration of a CFD simulation with a relatively coarse mesh still surpassing the BEM-based simulations.
- The trained surrogate model had an internal accuracy of 98% and a run time of approximately 3 ms, meaning that the possibility of achieving accurate surrogate modelling of CFD-level results at a speed high enough to be run in real-time is quite feasible.
- While the fluctuating components of the time histories of the aerodynamic loads predicted by the surrogate model may have limitations, the mean values of these thrust time histories match the CFD results well, resulting in thrust predictions closer to experimental results as shown in Figure 7.15. Since the rotor thrust and base response are strongly related, the base rotation time histories share the same behaviour.
- The mean base rotations predicted by the pre-calculated loads generated using the higher-fidelity OpenFAST model differs from those predicted by the lower-fidelity models by upwards of 10% as seen in Table 7.17, indicating that capturing the aeroelastic behaviour of the structure as closely as possible can have notable impact on the results under greater aerodynamic loads.
- The difference in thrust and base rotation behaviour between the surrogate model-based AHS and the pre-calculated AHS resulted in a noticeable difference between the predicted base rotation reduction offered by the STMD. This serves as an example of the

effect that more accurately simulating the response can have on this typical type of HS damper test.

- While this application of the proposed framework employing a CFD-trained surrogate model is not without its limitations, this study clearly shows the feasibility of such a technique and identifies indicators of the advantages it offers. A future simulation with an improved set of training data for the surrogate model could confirm these initial findings.

7.4.4 Limitations in the proposed framework

The main limitations of the surrogate model used in this study, primarily stemming from the CFD simulations performed to generate the training data, are dissected here to offer suggested improvements for future applications of the proposed framework.

The primary source of any limitations in this testing stems from the quality, amount, and variety of the CFD data used to train it:

- The mesh size of these CFD simulations was severely restricted by a number of factors. To effectively capture the base rotation of the turbine changing the wind direction against the rotor plane, the entire rotor had to be simulated as opposed to the typical technique of modelling only a third of the domain and rotationally mirroring it. Further, to maintain Re- and TSR-similitude without exceeding Ma limits, the geometric scaling was restricted to no more than 1:4. As a result, the CFD domain was quite large, limiting the fineness of the mesh. Due to the number of simulations required, it was infeasible to use a domain much larger than 30M cells, resulting in the use of a relatively coarse mesh as well as a high y^+ wall treatment. As discussed in Section 3.3.2, the high y^+ wall treatment is less accurate compared to the low y^+ wall treatment, which are nearly always recommended when the mesh size permits it. As was shown in Chapter 3, airfoils are very sensitive to the CFD mesh, and a result there is likely some error in the results of the CFD simulations, though there were never instances in this study where these errors were identified as a source of issues. Future simulations could use greater computational resources to generate a finer mesh, or model a smaller turbine rotor allowing the domain size to be reduced further – it may be possible to perform single-blade simulations in a hemispherical domain to generate the required training data, though new hurdles such as accurately capturing the effect of the rotor rotation and the neighbouring blades would need to be overcome.
- Due to the large number of required simulations, the large CFD domain, and the general computational intensity of CFD, a significant amount of supercomputing resources were required to generate a set of data of even modest length. To generate the data set of 24 simulations with 500 usable time steps each required over a thousand hours of CPU time on a supercomputing platform. This limits the ability to generate more data than the relatively small amount used here, as well as redoing tests if issues are identified. Without reducing the simulation complexity or simply using more computational resources, this limitations acts as a bottleneck for the CFD-trained surrogate model. In the worst case scenario, a high-accuracy version of a simplified numerical model such

as a high-detail BEM simulation with all modern improvements could be used to generate training data, similar to what was performed in Chapter 4. This would minimize the error from using simplified aerodynamic models while still allowing for in-the-loop aeroelastic calculations.

- In an attempt to ensure the base rotation during operation did not exceed the bounds of the CFD data, only a few wind speed and TSR cases could be simulated, and due to their short duration there was not much variation in the results during these simulations. The impact of this is shown in the scatter plots presented in Figure 7.17. These show the collected CFD data and surrogate model predictions from the AHS, comparing the hub wind speeds, the base rotations, and the rotor thrust. As can be seen, the data from the CFD simulations is highly concentrated at a small number of points; there is little variation within the individual CFD simulations. Since the turbulence of the test wind fields was restricted heavily, the results from the AHS are clustered around the data point, leading to the effective prediction of CFD-level aerodynamic loads. However, given a more realistic level of turbulence (near 10%), the AHS inputs would exceed the bounds of the training data and the accuracy of the network would suffer. Therefore, the surrogate model as it is presented here is accurate but lacks robustness; this is the likely source of the difference in fluctuations of the thrust between the surrogate model and pre-calculated loads. Exasperating this issue, it can be seen that a number of the simulations performed at higher base rotations were quite separated from the entirety of the AHS simulations; the computational resources used for these simulations would have been better served running more simulations closer to the AHS conditions, though this limitation could not easily have been identified ahead of time. The robustness issue must be solved by providing a greater variety of training data to the network at a greater variety of wind speeds and TSRs.
- The locations of the nodes used to record the wind speed in the CFD simulations were set

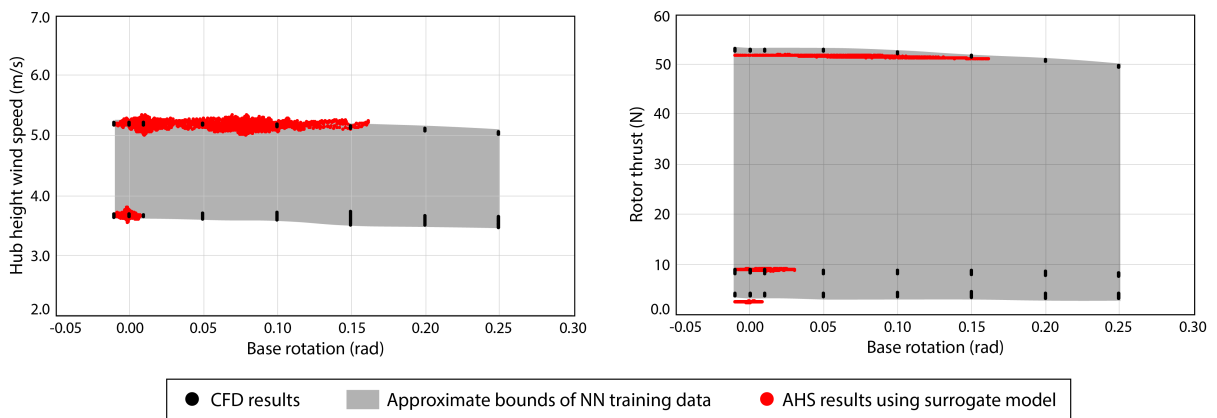


Figure 7.17: Scatter plots of CFD data and surrogate model AHS results comparing base rotation, hub height wind speed, and rotor thrust. The approximate bounds of the CFD training data are drawn, where the gradient indicates lower accuracy in the gaps between data points

back from the rotor in an attempt to reduce the influence from the rotor itself. However, even the wind speeds recorded at the furthest nodes were somewhat impacted by the rotor rotation, reducing the recorded wind speed relative to the domain input. Scaling factors were used on the inputs when provided to the surrogate model to account for this somewhat (as shown in Appendix B), but it is recommended to measure wind speeds at a greater distance from the rotor in future simulations to maximize the accuracy of the training data and resulting surrogate model.

While the overall trends of this study are supportive of the proposed HS framework despite these limitations, future applications of this framework would be well served to address these shortcomings using some of the suggested techniques. A future study investigating the proposed HS framework employing a CFD-trained surrogate model with these improvements would serve as a stronger, more explicit proof of the effectiveness of this novel technique.

7.5 Conclusions

This study presents the development of a surrogate aerodynamic wind turbine blade model based on a convolutional neural network (CNN) trained on computational fluid dynamics (CFD) data for use in an novel hybrid simulation (HS) framework, enabling previously limited HS of wind turbines and other wind-loaded structures. This surrogate model is combined with the aeroelastic rotating-base structural turbine model developed in Chapter 6 and the semi-active variable-damping tuned mass damper (STMD) optimized in Chapter 5. The framework was examined in a series of twelve artificial HS using four aerodynamic models across three wind cases of varying wind speed and tip speed ratios (TSRs). The aerodynamic models include three sets of “pre-calculated” rotor thrust time histories generated using the turbine software OpenFAST as well as the trained surrogate model. While the improvements offered by the new framework employing the CFD-trained surrogate model cannot be directly quantified, a number of observations were made indicating its effectiveness and the future potential of this technique.

This study presented a summary of the wind cases, the STMD, and the OpenFAST-generated pre-calculated thrust time histories. It then offered a detailed presentation of the development of the CFD-trained surrogate model including the numerical turbine blade approximation, a validation of the CFD parameters using 12 single-blade CFD simulations, a presentation of the 24 data-generating full-rotor CFD simulations, the architecture and optimization of the CNN based on the CFD data, and a validation of the trained network. Finally, the results of the twelve artificial HS were presented and discussed.

Overall, it was seen that while the robustness of the surrogate model was limited by the CFD-generated training data, the proposed AHS framework was shown to be computationally lightweight and resulted in improved prediction of the mean thrust and base rotations. This indicates an effective and feasible framework and supports further investigation of it using suggested improvements to advance the use of HS for studying wind-loaded structures.

Chapter 8

Conclusions and recommendations

This thesis presents six studies that collaborate towards the development of an improved hybrid simulation (HS) framework that incorporates computational fluid dynamics (CFD) for the study of wind turbines. HS is a relatively new testing technique capable of achieving experimental-level accuracy with improved scope and scale, which is particularly promising for wind turbines due to scaling issues arising in wind tunnel testing. However, due to the real-time testing speed of HS, advanced aerodynamic models are too slow computationally to be incorporated directly, limiting the technique's ability to accurately model aeroelastic responses. Thus, there exists a need for a modified HS framework that addresses these limitations, which is accomplished in the proposed HS framework by incorporating a CFD-trained surrogate model.

This thesis built toward this proposed framework as follows: firstly, the need was shown for the new framework by quantifying the limitations in the existing framework that employs low-accuracy, pre-calculated wind loads. The aerodynamic model for the HS was developed by determining the optimal CFD settings to generate the training data, as well as the optimal type and architecture of the convolutional neural network (CNN) which was trained on this CFD data to create the surrogate model. The HS test case of a turbine equipped with a semi-active tuned mass damper (TMD) was evaluated and optimized. The structural model of a residential rotating-base wind turbine was developed and validated using the results of aeroelastic wind tunnel testing. Finally, each of these components were combined into a series of artificial hybrid simulations (AHS) that compared four different HS frameworks: three versions of the existing framework employing lower-accuracy, pre-calculated aerodynamic loads, as well as the novel proposed framework that incorporated the CNN-based, CFD-trained surrogate aerodynamic model. Though these artificial HS were purely numerical, they approximated real HS by modelling realistic delays and errors into the simulation. Ultimately, while quantification of the improvement offered by the proposed framework could not be drawn, several indicators of the effectiveness of the technique were identified.

The proposed framework offers a potential avenue to enable the nascent use of HS in wind engineering. While it is particularly promising for the area of wind turbine research, the general framework could be applied for nearly any wind-loaded structure.

8.1 Summaries of the conclusions of each chapter

In addition to contributing to the overall goal of developing the surrogate model-based HS framework, each of the studies presented in this thesis stand independently. The broad conclusions and contributions for each of these studies are summarized here.

8.1.1 Chapter 2: Wind turbines and hybrid simulation: an overview of research, opportunities and limitations

In addition to collecting the general trends of existing wind turbine and HS studies, this study also includes a numerical investigation of the effects of the limitations in the existing HS framework caused by load pre-calculation and reduced aeroelastic fidelity, using a 5 MW wind turbine equipped with a tuned mass damper (TMD) as the test case. This study came to the following conclusions:

- HS is a promising testing technique for wind turbine studies due to complexity and scaling issues. Existing HS of wind turbines have primarily been focused on testing the use of complex, non-linear external damping systems such as semi-active and active TMDs.
- By studying a TMD-equipped wind turbine in an artificial hybrid simulation, it was shown that insufficient aeroelastic fidelity can result in errors exceeding 10% and that pre-calculated wind loads can result in errors exceeding 70%. As a result, there are significant improvements to be made in HS of wind turbines if a framework can be developed that minimizes or avoids these limitations.
- The following research topics are promising options for future HS of wind turbines: offshore turbines, turbines under multi-hazard loading, turbines with external damping systems, and structural failures in turbines.

The most notable contribution of this study beyond its use in this thesis was the quantification of the possible errors introduced by pre-calculated aerodynamic loads using lower-accuracy models. These values critically emphasize the limitations that currently exist in wind engineering applications of HS.

8.1.2 Chapter 3: Investigation of the drag and lift coefficient curves of the NACA 0012 airfoil using CFD and a novel synthesis method

This study characterizes the drag and lift coefficient curves of the NACA 0012 airfoil at Reynolds numbers of 10^5 to 10^6 using two methods. The first was a novel synthesis method that generated piecewise polynomial systems to characterize the average curves as a function of the Reynolds number based on data collected from existing studies. The second was a collection of 114 CFD simulations of the NACA 0012 airfoil in 2D and 3D using RANS and URANS employing the Spalart–Allmaras, k - ϵ , k - ω closure models, as well as preliminary LES simulations. This study came to the following conclusions:

- There is a notable amount of disagreement between the drag and lift coefficient curves predicted by existing studies due to the complexity of the aerodynamic behaviour of the airfoil in the specified Reynolds number range. Disagreement was greatest at the stall and post-stall angles of attack beyond 10-15°.
- Despite these disagreements, four piecewise polynomial systems of the 2D and 3D lift and drag coefficient curves as a function of the Reynolds number were able to be developed. They generally managed to agree with all data within a single standard deviation.
- The 2D CFD simulations showed the importance of vortex shedding on the predicted curves, which the existing 2D studies may have failed to model correctly. Of the tested closure models, the $k-\omega$ model was found to be most accurate.
- The 3D CFD simulations generally matched well with the curve predicted from the existing data, which represents a more realistic simulation environment. The preliminary LES results fail to accurately capture flow behaviour at the stall angle due to computational resources limiting the mesh density.
- Overall, the most effective CFD parameters, for the purpose of the improved framework proposed in Chapter 7, were deemed to be a 3D URANS simulation using the $k-\omega$ closure model. URANS captures transient behaviour more effectively than RANS simulations, and for researchers who are less-experienced with CFD, URANS is simpler to implement and requires lower computational resources compared to LES.

The most notable contribution of this study beyond its use in this thesis was the development of the piecewise polynomial system-based framework for averaging drag and lift coefficient curves from existing wind tunnel studies. This offers researchers an efficient way to compare new research to a single coefficient curve provided there are existing studies at similar – but not necessarily identical – Reynolds numbers.

8.1.3 Chapter 4: Optimizing the type and architecture of neural network-based surrogate aeroelastic wind turbine blade models

This study optimizes and compares six neural networks (NNs) for their use as surrogate aerodynamic blade models for a 5 MW wind turbine blade. The six NNs include four types of multilayer perceptron (MLP) networks, one long short-term memory (LSTM) network, and one convolutional neural network (CNN). After hyperparameter optimization and final training, the accuracy of the network was determined by calculating the average NRMSE between the predicted outputs of the network and the supplied testing data outputs. This study came to the following conclusions:

- Overall, the full-parameter MLP, which represents the baseline time-independent network of the study, was found to have a very low average NRMSE of 1.11%. Comparatively, the more-complex CNN had an extremely low average NRMSE of 0.66%, which was shown to be a result of the wide view of the input data allowed by the convolution

process. These two networks out-performed the other four, and also out-performed previous neural networks of this size applied in wind turbine studies that tended to have errors of 5% or greater.

- All the trained surrogate models were able to achieve high operation speed of <10 ms, which is sufficiently fast to be incorporated into real-time simulations.

The most notable contribution of this study beyond its use in this thesis was the identification of the utility of CNNs as surrogate models for structural and wind engineering applications. CNNs see little general application in this field due to being designed for image processing, but this study shows how they can be employed for alternative applications relatively easily and to good effect.

8.1.4 Chapter 5: A comparison of passive and semi-active tuned mass damper systems for wind turbines

This study numerically compares eight TMD systems used for vibration control of a 1.5 MW wind turbine subjected to wind and seismic loading. These TMDs include a passive TMD and three types of semi-active TMDs in single- and multi-TMD configurations; the three STMDs semi-actively controlled the stiffness, the damping, and both, respectively. This study came to the following conclusions:

- Overall, all the TMDs improved the response of the wind turbine, but the STMD that controlled both stiffness and damping was the most effective when considering all performance indices. The damping control TMDs and stiffness control TMDs also showed improved effectiveness compared to PTMD cases, with the former showing improved control of the middle of the tower and the latter showing improved control of the top of the tower. Specifically, a semi-active TMD with half the mass was found to be almost equally effective as an optimized passive TMD.
- The use of a multi-TMD system with the same collective mass as a single-TMD resulted in improved control of higher mode responses at the cost of a slight reduction in control of the first mode.
- The improvement of the STMDs versus the PTMDs was relatively small in some instances, and must be weighed against the cost of implementing the semi-active system. Overall, the absolute best damping system will depend on the desired control for each performance index, the expected loading, and a cost-benefit analysis.

The most notable contribution of this study beyond its use in this thesis was the detailing of the realistic semi-active damper controllers such as the magnetorheological damper. Most studies in this area adopt idealized controllers, but this study shows that physically-realistic controllers can be readily adopted in numerical simulations to achieve more robust and more conservative test results.

8.1.5 Chapter 6: Development of a rotating-base residential wind turbine model based on aeroelastic wind tunnel tests

This study details the development of an experimentally-validated structural wind turbine model for the purpose of acting as a testbed for subsequent hybrid simulation studies. Full-scale aeroelastic wind tunnel testing of a rotating-base turbine was performed to generate the data used to build a 1-DOF numerical structural model, including an MLP-based surrogate thrust model and a three-term damping model. Finally, the validated model was used in a simple case study of a turbine equipped with a non-linear vibration damping system. This study came to the following conclusions:

- While the results of the aeroelastic wind tunnel testing met the requirements set out at the beginning of the testing, there were a number of limitations primarily due to over-damping in the rotating base. These limitations could be avoided in future testing via a number of suggested improvements.
- The numerical model was designed and validated using existing wind tunnel research as well as the new aeroelastic tests. Matching between the designed numerical model and the experimental data was shown to be at least 80%, and very likely would be higher with an improved aerodynamic model, which was adopted in the subsequent testing in Chapter 7.
- The overly-high damping in the aeroelastic base was best modelled using a combined linear-quadratic-Coulomb damping model, which was modified for the proposed final structural model.
- The case study of the final 1-DOF model compared the effectiveness of passive and semi-active TMDs for vibration reduction. Depending on the test case, base rotation was reduced by up to 18% and the STMD was up to 8% more effective. This case study showed the structural model to be effective for hybrid simulation-like applications.

The most notable contribution of this study beyond its use in this thesis was the identification and quantification of the three-term damping model for the rotating turbine base. Aeroelastic wind tunnel models are often quite sensitive, and thus there is a real risk of additional, unintentional vibration suppression creeping into the system due to a number of sources of aerodynamic or friction damping. The ability to analyze a free-vibration curve that is not captured by a traditional linear model and identify the additional sources of damping is useful for troubleshooting and processing results.

8.1.6 Chapter 7: Development of a novel hybrid simulation framework employing a CFD-based surrogate model to study a semi-actively damped wind turbine

This study details the development of CFD-based surrogate aerodynamic wind turbine blade model, which was then employed in a novel HS framework replacing the existing technique

of pre-calculating aerodynamic loads. Development of the surrogate model consisted of validating the CFD parameters, performing 24 URANS CFD simulations, and training a CNN on the resulting data. Twelve artificial HS were then performed using both the proposed surrogate model-based framework as well as pre-calculated aerodynamic loads generated using OpenFAST; the structure for these simulations was a residential wind turbine with aeroelastic rotating base equipped with a STMD. This study came to the following conclusions:

- The single-blade validation CFD simulations as well as the main full-rotor simulations showed an improvement in thrust and torque prediction against experimental data when compared to BEM-based data. This supports the improvement offered by a HS framework incorporating a CFD-level aerodynamic model.
- By extracting the wind speed and aerodynamic loads from a series of CFD simulations, a set of training data could be generated for the NN. The trained CNN had an internal accuracy of 98% despite the relatively small amount of training data.
- As with the CFD simulations, the thrust time histories from the AHS employing the surrogate model predicted rotor thrusts that more closely matched the expected experimental values compared to the BEM-based pre-calculated aerodynamic loads.
- The different levels of aeroelastic fidelity of the pre-calculated wind loads only resulted in a noticeable difference in response at the high TSR case where displacements were greatest – understandably, when base rotations are low, ignoring them in the pre-calculation model was less impactful.
- This application of the novel HS framework using a NN-based, CFD-trained surrogate model was not without its limitations, and while the amount of improvement offered by the technique cannot be directly quantified, this study clearly showed the feasibility of such a technique, identified a number of results that support the possible advantages of this technique, and laid out how this framework can be modified in future testing to maximize its potential.

The most notable contribution of this study was plainly the presentation of the novel HS framework incorporating the CFD-based surrogate model. The methodology of the development of the surrogate model was clearly laid out for readers, and the results of the AHS showed the feasibility of this technique as well as indicate its possible ability to surpass the existing technique of pre-calculating aerodynamic loads in both accuracy of aerodynamic simulation as well as in aeroelastic fidelity. This framework is a promising tool for future HS of wind-loaded structure and warrants future study.

8.2 Observations supporting the effectiveness of the proposed framework

Given that the proposed framework is the major contribution of the overarching research presented in this thesis, its achievements should be highlighted. Observations from all chapters

indicating the effectiveness of the proposed HS framework that employs a CFD-trained surrogate aerodynamic model are collected here:

- The literature review presented in Chapter 2 reported previous findings that the simplified aerodynamic models (BEM, XFOIL, etc.) commonly employed to generate pre-calculated load time histories often introduce errors of $\geq 10\%$ compared to CFD simulations. The simulations of the NACA 0012 airfoil in Chapter 3 showed that the stall-induced drag predicted by fine-meshed, unsteady CFD simulations was significantly improved compared to XFOIL as well as coarser, simpler CFD simulations. The single-blade and full-rotor CFD simulations of the 1.1 m residential wind turbine rotor in Chapter 7 were both shown to match experimental results better than equivalent BEM simulations. The proposed framework enables CFD-level results to be introduced into a real-time HS given the development of a well-trained surrogate model.
- The numerical study detailed in Section 2.8 compared the errors introduced by limiting the aeroelastic fidelity of turbine blades in numerical simulations, finding that differences of up to 70% were seen between the low- and high-fidelity models. A similar comparison of the fidelity of the numerical models used to pre-calculated rotor thrust for the AHS in Chapter 7 identified differences of up to 10% between the low- and high-fidelity models. By performing the aerodynamic calculations during the HS, the proposed framework enables aeroelastic fidelity that can not be matched by pre-calculated loads.
- A robust CNN-based surrogate aerodynamic model of a 5 MW wind turbine blade was trained on a large body of BEM-generated data in Chapter 4. It was able to achieve an internal accuracy of 99%. A similar architecture was applied for the surrogate aerodynamic model in Chapter 7, however due to the lower size and variety of the data set, this model was comparatively less robust. The surrogate model in the proposed framework is thus shown to be capable of accurately estimating the aerodynamic results, but the training data must be carefully curated such that a sufficient variety and quantity of results are provided to the NN.
- The results of the dynamic simulations of wind turbines equipped with STMDs show that the effectiveness these systems is strongly tied to the aerodynamic loads applied on the turbine tower as a result of the tuned nature of the STMDs. This was shown by the differing reductions in vibration caused by the STMD across the different loading types in Chapter 5, as well as the different operational conditions in the case study in Section 6.4 and the AHS in Chapter 7. As a result, as the proposed framework should be capable of more accurately predicting the aerodynamic loads in such a test, its use is recommended.
- While the proposed HS framework as it was applied in Chapter 7 had limited robustness, within its area of effectiveness it was capable of improved thrust prediction compared to the existing, pre-calculated framework. The feasibility of the framework was also clearly shown: the surrogate model was capable of being integrated into an artificial HS environment while meeting the upper-limit timing requirements, and could be further quickened using different hardware, parallelization, changing programming environments, etc.

Overall, the proposed HS framework that incorporates a CNN-based, CFD-trained surrogate model for in-the-loop aerodynamic calculations was shown to be feasible to implement, thoroughly supported by theory, and had a number of positive indicators of its effectiveness as it was applied in this thesis.

8.3 Future research directions

The proposed HS framework could be applied in a host of future wind turbine tests, as well as for studying other wind-loaded structures. Due to the limitations in the existing technique of pre-calculating aerodynamic loads, HS of wind turbines has been relatively rare, but the proposed framework is designed to address those limitations to enable more and new types of testing. This includes test cases such as offshore turbines, multi-hazard loading, damping systems, and member failure.

The studies presented in this thesis are not without flaws, and future research based on these studies would be well-served to note and improve upon the identified limitations. The most noteworthy examples include:

- The quality and amount of the CFD simulations performed in Chapters 3 and 7 were often restrained due to limited computational resources, further exasperated by the number of required simulations. The 3D CFD simulations presented in Section 3.3.4, particularly the more-demanding LES simulations, were limited to shorter unsteady simulation lengths as well as lower density meshes; being forced to use a coarser mesh than the optimal found in the mesh sensitivity study in Section 3.3.2. One example of the effects of coarser meshes on turbine blade simulations is to reduce the curvature of the windward edge, resulting in airfoils with semi-sharp geometry that are at risk of changing the flow attachment behaviour. The effect of the limited number of CFD simulations performed in Chapter 7 are detailed in Section 7.4.4, resulting in a less robust surrogate model. Future simulations must account for the fine meshing requirements of full-scale and near-full-scale turbine blade simulations, either by acquiring greater computational resources or by reducing computational demand; such as by reducing the number of planned tests, testing a smaller rotor, or finding ways to perform single-blade simulations instead of modelling the full rotor. These simplifications would help ensure that each test is of maximal quality.
- The numerical approximation of the 1.1 m residential wind turbine blade used in both Chapters 6 and 7 is limited in that it is truly an approximation, not a recreation, of the real turbine blade. BEM-based simulations of the approximation approach the experimental behaviour of the true turbine blade, but not overly so. The CFD simulations using the same approximation match the experimental results somewhat more closely, but ultimately the approximation is not a perfect recreation of the test turbine blade. Future simulations of this type should either use manufacturer specifications of the blade shape, an open-source blade design, or perform a 3D scan of the turbine blade to create a perfect digital recreation.

- The proposed framework incorporates two upgrades over the existing HS method. The first is the change from lower-accuracy numerical models to CFD-level data, while the second is the change from pre-calculated aerodynamic loads to performing the calculation within the HS loop. It may be worthwhile to investigate the effect of each of these improvements separately to increase understanding and the potential usability of the proposed framework.
- The limitations in the proposed framework as it was applied in Chapter 7 and the resulting suggested improvements were well-documented in Section 7.4.4. However, ultimately its advantages or disadvantages compared to the pre-calculated framework could not be quantified due to the lack of a baseline simulation to compare against. The aeroelastic wind tunnel testing presented in Chapter 6 was intended to fill this role but ultimately was not able to be replicated in AHS form due to the lack of an external damping system. Future validation of the proposed framework would be well-served to ensure there exist baseline results to compare against for explicit confirmation of the improvements offered.
- Another notable limitation in the application of the proposed framework that warrants further study is the lack of true HS used to test it. Due to time and travel limitations partially due to the recent pandemic, performing a true HS was ultimately an infeasible option – as discussed in Chapter 2, such testing requires specific equipment that was not available at Western University. A decent approximation has been provided through the artificial HS performed in Chapter 7, however the quality of the validation as well as the novelty of this research would have been greater with a true HS.

With regards to applying the proposed framework in future wind turbine tests, in the author's opinion the two most promising avenues are:

1. **Failure testing of rotor blades in high-intensity wind events:** It has been observed that when turbines have been subjected to high-intensity winds (such as from hurricanes or typhoons) to the point of member failure, it typically manifests as either blade failure or tower failure; if the blades fail, the tower will typically remain undamaged as the drag loads on the turbine are reduced significantly without the blades. Therefore, HS could be used to design blades that intentionally fail before the tower which could be used to protect the expensive nacelle and tower at the cost of the comparatively cheaper blades, reducing the costs of typhoon- and hurricane-induced damages to wind farms. HS has been shown to be an optimal technique for member failure studies under earthquake loading and the proposed framework could enable applying this technique to wind-loaded structures.
2. **Studies of damping systems to mitigate fatigue loading under operational conditions:** The most obvious method to reduce the costs of wind turbine adoption is to increase their fatigue-limited service life. External damping systems such as TMDs could be used to reduce the magnitude of operational vibrations and, as a result, reduce the overall fatigue loading. This would represent a quantitative reliability improvement for these crucial structures. HS has been shown to be an effective technique for studying complex, non-linear dampers, which are likely the best option here given the unique loading directions and space requirements for blade- and nacelle-mounted dampers.

Additionally, these proposed research avenues only just scratch the surface of the possibilities that hybrid simulation offers for the field of wind engineering, now that a novel framework that incorporates a CNN-based, CFD-trained surrogate aerodynamic model has been designed to circumvent the limitations that have historically held back hybrid simulations of wind-loaded structures.

Bibliography

- Abdelkader, A., Aly, A., Razaee, M., Bitsuamlak, G., El Naggar, M., 2016. On the evaluation of wind loads for wind turbines' foundation design: Experimental and numerical investigations. *Structural Design of Tall and Special Buildings* 26(9). doi:<https://doi.org/10.1002/tal.1362>.
- Aboshosha, H., Bitsuamlak, G., El Damatty, A., 2015a. Turbulence characterization of down-burst using LES. *Journal of Wind Engineering and Industrial Aerodynamics* 136, 44–61. doi:<https://doi.org/10.1016/j.jweia.2014.10.020>.
- Aboshosha, H., Elshaer, A., Bitsuamlak, G., El Damatty, A., 2015b. Consistent inflow turbulence generator for LES evaluation of wind-induced responses for tall buildings. *Journal of Wind Engineering and Industrial Aerodynamics* 142, 198–216. doi:<https://doi.org/10.1016/j.jweia.2015.04.004>.
- Adeli, H., 2002. Neural networks in civil engineering: 1989-2000. *Computer-Aided Civil and Infrastructure Engineering* 16(2), 126–142. doi:<https://doi.org/10.1111/0885-9507.00219>.
- Adhikari, S., Bhattacharya, S., 2011. Vibrations of wind turbines considering soil-structure interaction. *Wind and Structures* 14(2), 85–112. doi:<https://doi.org/10.12989/was.2011.14.2.085>.
- Ahmadizadeh, M., Mosqueda, G., 2007. Combined implicit or explicit integration steps for hybrid simulation. *Earthquake Engineering and Structural Dynamics* 36(15), 2325–2343. doi:[https://doi.org/10.1061/40944\(249\)1](https://doi.org/10.1061/40944(249)1).
- Airfoil Tools, 2021. NACA 0012 AIRFOILS. <http://airfoiltools.com/airfoil/details?airfoil=n0012-il>.
- Ali, A., De Risi, R., Sextos, A., 2020. Seismic assessment of wind turbines: How crucial is rotor-nacelle-assembly numerical modeling. *Soil Dynamics and Earthquake Engineering* doi:<https://doi.org/10.1016/j.soildyn.2020.106483>.
- Alkhoury, P., Soubra, A., Rey, V., Ait-Ahmed, M., 2020. A full three-dimensional model for the estimation of the natural frequencies of an offshore wind turbine in sand. *Wind Energy* doi:<https://doi.org/10.1002/we.2598>.

- Amezquita-Sanchez, J., Valtierra-Rodriguez, M., Aldwaik, M., Adeli, H., 2016. Neurocomputing in civil infrastructure. *Scientia Iranica* 23(6), 2417–2428. doi:<https://doi.org/10.24200/SCI.2016.2301>.
- Amirinia, G., Jung, S., 2017. Along-wind buffeting responses of wind turbines subjected to hurricanes considering unsteady aerodynamics of the tower. *Engineering Structures* 138, 337–350. doi:<https://doi.org/10.1016/j.engstruct.2017.02.023>.
- Ancheta, T., Darragh, R., Stewart, J., Seyhan, E., Silva, W., Chiou, B., Wooddell, K., Graves, R., Kottke, A., Boore, D., Kishida, T., Donahue, J., 2013. PEER NGA-West2 Database. Pacific Earthquake Engineering Research Center, Berkeley, USA.
- Argyriadis, K., Hille, N., Lloyd, G., 2004. Determination of fatigue loading on a wind turbine with oil damping device. Corporate publication.
- Arrigan, J., Pakrashi, V., Basu, B., Nagarajaiah, S., 2011. Control of flapwise vibrations in wind turbine blades using semi-active tuned mass dampers. *Journal of Structural Control and Health Monitoring* 18(8), 840–851. doi:<https://doi.org/10.1002/stc.404>.
- Asareh, M., Schonberg, W., Volz, J., 2016. Fragility analysis of a 5-MW NREL wind turbine considering aero-elastic and seismic interaction using finite element method. *Finite Elements in Analysis and Design* 120, 57–67. doi:<https://doi.org/10.1016/j.finel.2016.06.006>.
- Asareh, M., Volz, J., 2013. Evaluation of aerodynamic and seismic coupling for wind turbines using finite element approach, in: International Mechanical Engineering Congress and Exposition.
- Atkins, K., 2015. Investigation into the Flow Behaviour of a NACA 0021 Airfoil with Leading Edge Undulations Simulated with RANS in STAR-CCM+. University of Manchester Master Thesis, Manchester, U.K.
- Azcona, J., Bouchotrouch, F., Gonzalez, M., Garciandia, J., Munduate, X., Kelberlau, F., Nygaard, T., 2014. Aerodynamic thrust modelling in wave tank tests of offshore floating wind turbines using a ducted fan. *Journal of Physics: Conference Series* 524. doi:<https://doi.org/10.1088/1742-6596/524/1/012089>.
- Bachynski, E., Chabaud, V., Sauder, T., 2015. Real-time hybrid model testing of floating wind turbines: sensitivity to limited actuation. *Energy Procedia* 80, 2–12. doi:<https://doi.org/10.1016/j.egypro.2015.11.400>.
- Bakre, S., Jangrid, R., 2007. Optimum parameters of tuned mass damper for damped man system. *Structural Control and Health Monitoring* 14, 448–470. doi:<https://doi.org/10.1002/stc.166>.
- Bayati, I., Belloli, M., Bernini, L., Giberti, H., Zasso, A., 2016. Scale model technology for floating offshore wind turbines. *IET Renewable Power Generation* 11(9), 1120–1126. doi:<https://doi.org/10.1049/iet-rpg.2016.0956>.

- Beale, M., Hagan, M., Demuth, H., 2019. Deep Learning Toolbox Reference. MathWorks, Natick, U.S.A.
- Benyahia, A., Berton, E., Favier, D., Maresca, C., Badcock, K., Barakos, G., 2003. Detailed evaluation of CFD predictions against LDA measurements for flow on an aerofoil, in: Integrating CFD and Experiments in Aerodynamics International Symposium.
- Berny-Brandt, E., Ruiz, S., 2016. Reliability over time of wind turbines towers subjected to fatigue. *Wind and Structures* 23(1), 75–90. doi:<https://doi.org/10.12989/was.2016.23.1.075>.
- Betz, A., 1926. *Wind-Energie und ihre ausnutzung durch Windmuhlen*. Gottingen, Vandenhoeck.
- Bieker, K., Peitz, S., Brunton, S., Kutz, J., Dellnitz, M., 2020. Deep model predictive flow control with limited sensor data and online learning. *Theoretical and Computational Fluid Dynamics* 34, 577–591. doi:<https://doi.org/10.1007/s00162-020-00520-4>.
- Bitsuamlak, G., Bedard, C., Stathopoulos, T., 2007. Modeling the effect of topography on wind flow using a combined numerical–neural network approach. *Journal of Computing in Civil Engineering* 21(6). doi:[https://doi.org/10.1061/\(ASCE\)0887-3801\(2007\)21:6\(384\)](https://doi.org/10.1061/(ASCE)0887-3801(2007)21:6(384)).
- Bre, F., Gimenez, J., Fachinotti, V., 2019. Prediction of wind pressure coefficients on building surfaces using artificial neural networks. *Energy and Buildings* 158, 1429–1441. doi:<https://doi.org/10.1016/j.enbuild.2017.11.045>.
- Brodersen, M., BJORKE, A., Hogsberg, J., 2017. Active tuned mass damper for damping of offshore wind turbine vibrations. *Wind Energy* 20(5), 783–796. doi:<https://doi.org/10.1002/we.2063>.
- Brodersen, M., Ou, G., Hogsberg, J., Dyke, S., 2016. Analysis of hybrid viscous damper by real time hybrid simulations. *Engineering Structures* 126, 675–688. doi:<https://doi.org/10.1016/j.engstruct.2016.08.020>.
- Brunton, S., Noack, B., Koumoutsakos, P., 2020. Machine learning for fluid mechanics. *arXiv Fluid Dynamics*. doi:<https://arxiv.org/abs/1905.11075>.
- Bukala, J., Damaziak, K., Karimi, H., Malachowski, J., 2016. Aero-elastic coupled numerical analysis of small wind turbine-generator modelling. *Wind and Structures* 23(6), 577–594. doi:<https://doi.org/10.12989/was.2016.23.6.577>.
- Calderer, A., Guo, X., Shen, L., Sotiropoulos, F., 2014. Coupled fluid-structure interaction simulation of floating offshore wind turbines and waves: a large eddy simulation approach. *Journal of Physics: Conference Series* 524. doi:<https://doi.org/10.1088/1742-6596/524/1/012091>.
- Campagnolo, F., Petrovic, V., Bottasso, C., Croce, A., 2016. Wind tunnel testing of wake control strategies, in: American Control Conference.

- Caterino, N., 2014. Semi-active control of a wind turbine via magnetorheological dampers. *Journal of Sound and Vibration* 345, 1–17. doi:<https://doi.org/10.1016/j.jsv.2015.01.022>.
- Caterino, N., Georgakis, C., Spizzuoco, M., Occhiuzzi, A., 2016. Design and calibration of a semi-active control logic to mitigate structural vibrations in wind turbines. *Smart Structures and Systems* 18(1), 75–92. doi:<https://doi.org/10.12989/sss.2016.18.1.075>.
- Caterino, N., Georgakis, C., Trinchillo, F., Occhiuzzi, A., 2014. A semi-active control system for wind turbines. *Wind Turbine Control and Monitoring*, 375–407. doi:https://doi.org/10.1007/978-3-319-08413-8_13.
- Caterino, N., Spizzuoco, M., Occhiuzzi, A., 2011. Understanding and modelling the physical behaviour of magnetorheological dampers for seismic structural control. *Smart Materials and Structures* 20(6). doi:<https://doi.org/10.1088/0964-1726/20/6/065013>.
- Caterino, N., Spizzuoco, M., Occhiuzzi, A., 2013. Promptness and dissipative capacity of mr dampers: Experimental investigations. *Structural Control and Health Monitoring* 20, 1424–1440. doi:<https://doi.org/10.1002/stc.1578>.
- Chabaud, V., 2016. Real-Time Hybrid Model Testing of Floating Wind Turbines. Norwegian University of Science and Technology Ph.D. Thesis, Trondheim, Norway.
- Chae, Y., Phillips, B., Ricles, J., Spencer, B., 2013. An enhanced hydraulic actuator control method for large scale real-time hybrid simulations, in: *Structures Congress*.
- Chae, Y., Ricles, J., Sause, R., 2012. Modeling of a large-scale magneto-rheological damper for seismic hazard mitigation. part 1: Passive mode. *Earthquake Engineering and Structural Dynamics* 42(5). doi:<https://doi.org/10.1002/eqe.2237>.
- Chang, S., 2012. Discussion of paper ‘Real-time hybrid testing using the unconditionally stable explicit CR integration algorithm’ by Cheng Chen, James M. Ricles, Thomas M. Marullo and Oya Mercan, *Earthquake Engineering and Structural Dynamics* 2009; 38:23–44. *Earthquake Engineering and Structural Dynamics* 41(5). doi:<https://doi.org/10.1002/eqe.1159>.
- Chapain, S., Aly, A., 2020. Vibration attenuation in wind turbines: A proposed robust pendulum pounding TMD. *Engineering Structures* 233. doi:<https://doi.org/10.1016/j.engstruct.2021.111891>.
- Chen, C., Ricles, J., 2012. Response to ‘Discussion of paper “Real-time hybrid testing using the unconditionally stable explicit CR integration algorithm” by Cheng Chen, James M. Ricles, Thomas M. Marullo and Oya Mercan’ in *Earthquake Engineering and Structural Dynamics* 2009; 38:23–44. *Earthquake Engineering and Structural Dynamics* 41(5). doi:<https://doi.org/10.1002/eqe.1160>.
- Chen, C., Ricles, J., Marullo, T., Mercan, O., 2009. Real-time hybrid testing using the unconditionally stable explicit CR integration algorithm. *Earthquake Engineering Structural Dynamics* 38(1), 23–44. doi:<https://doi.org/10.1002/eqe.838>.

- Chen, J., Liu, Y., Bai, X., 2015b. Shaking table test and numerical analysis of offshore wind turbine tower systems controlled by TLCD. *Earthquake Engineering and Engineering Vibration* 14, 55–75. doi:<https://doi.org/10.1007/s11803-015-0006-5>.
- Chen, X., Li, C., Xu, J., 2015a. Failure investigation on a coastal wind farm damaged by super typhoon: A forensic engineering study. *Journal of Wind Engineering and Industrial Aerodynamics* 147, 132–142. doi:<https://doi.org/10.1016/j.jweia.2015.10.007>.
- Chen, X., Xu, J., 2016. Structural failure analysis of wind turbines impacted by super typhoon Usagi. *Engineering Failure Analysis* 60, 391–404. doi:<https://doi.org/10.1016/j.engfailanal.2015.11.028>.
- Chey, M., Chase, J., Mander, J., Carr, A., 2009. Semi-active tuned mass damper building systems: Design. *Earthquake Engineering Structural Dynamics* 39(2), 119–139. doi:<https://doi.org/10.1002/eqe.934>.
- Chou, J., Tu, W., 2011. Failure analysis and risk management of a collapsed large wind turbine tower. *Engineering Failure Analysis* 18, 295–313. doi:<https://doi.org/10.1016/j.engfailanal.2010.09.008>.
- Chung, L., Lai, Y., Yang, C., Lien, K., Wu, L., 2013. Semi-active tuned mass dampers with phase control. *Journal of Sound and Vibration* 332, 3610–3625. doi:<https://doi.org/10.1016/j.jsv.2013.02.008>.
- Colak, I., Sagiroglu, S., Yesilbudak, M., 2012. Data mining and wind power predictions: A literature review. *Renewable Energy* 46, 241–247. doi:<https://doi.org/10.1016/j.renene.2012.02.015>.
- Colherinhas, G., Petrini, F., de Morais, M., Bontempi, F., 2020. Optimal design of passive-adaptive pendulum tuned mass damper for the global vibration control of offshore wind turbines. *Wind Energy*, 1–23doi:<https://doi.org/10.1002/we.2590>.
- Connor, J., 2002. *Introduction to Structural Motion Control* (1st edition). Pearson.
- Cornford, D., Ramage, G., Nabney, I., 2000. A scatterometer neural network sensor model with input noise. *Neurocomputing* 30, 13–21. doi:[https://doi.org/10.1016/S0925-2312\(99\)00137-X](https://doi.org/10.1016/S0925-2312(99)00137-X).
- Critzos, C., Heyson, H., Boswinkle, R., 1955. Aerodynamic characteristics of NACA 0012 airfoil section at angles of attack from 0° to 180°. National Advisory Committee for Aeronautics.
- Dagnew, A., Bitsuamlak, G., 2013. Computational evaluation of wind loads on buildings: a review. *Wind and Structures* 16(6), 629–660. doi:<https://doi.org/10.12989/was.2013.16.6.629>.
- Dai, K., Huang, H., Lu, Y., Meng, J., Mao, Z., Camara, A., 2020. Effects of soil–structure interaction on the design of tuned mass damper to control the seismic response of wind turbine towers with gravity base. *Wind Energy* doi:<https://doi.org/10.1002/we.2576>.

- Dai, K., Sheng, C., Zhao, Z., Yi, Z., Camara, A., Bitsuamlak, G., 2017b. Nonlinear response history analysis and collapse mode study of a wind turbine tower subjected to tropical cyclonic winds. *Wind and Structures* 25(1), 79–100. doi:<https://doi.org/10.12989/was.2017.25.1.079>.
- Dai, K., Wang, Y., Huang, Y., Zhu, W., Xu, Y., 2017a. Development of a modified stochastic subspace identification method for rapid structural assessment of in-service utility-scale wind turbine towers. *Wind Energy* 20, 1687–1710. doi:<https://doi.org/10.1002/we.2117>.
- Dai, K., Yichao, H., Changqing, G., Zhenhua, H., Xiaosong, R., 2015a. Rapid seismic analysis methodology for in-service wind turbine towers. *Earthquake Engineering and Engineering Vibration* 14(3), 539–548. doi:<https://doi.org/10.1007/s11803-015-0043-0>.
- Dermitzakis, S., Mahin, S., 1985. Development of substructuring techniques for on-line computer controlled seismic performance testing (UCB/EERC-85/04). University of California.
- Diaz, O., Suarez, L., 2014. Seismic analysis of wind turbines. *Earthquake Spectra* 30(2), 743–765. doi:<https://doi.org/10.1193/123011EQS316M>.
- Dinh, V., Basu, B., Nagarajaiah, S., 2016. Semi-active control of vibrations of spar type floating offshore wind turbines. *Smart Structures* 18(4), 683–705. doi:<https://doi.org/10.12989/sss.2016.18.4.683>.
- Do, T., van de Lindt, J., Mahmoud, H., 2015a. Fatigue life fragilities and performance-based design of wind turbine tower base connections. *Journal of Structural Engineering* 141(7). doi:[https://doi.org/10.1061/\(ASCE\)ST.1943-541X.0001150](https://doi.org/10.1061/(ASCE)ST.1943-541X.0001150).
- Dogo, E., Afolabi, O., Nwulu, N., Twala, B., Aigbavboa, C., 2018. A comparative analysis of gradient descent-based optimization algorithms on convolutional neural networks, in: *International Conference on Computational Techniques, Electronics and Mechanical Systems*. doi:<https://doi.org/10.1109/CTEMS.2018.8769211>.
- Douvi, E., Athanasios, T., Margaris, D., 2012. Evaluation of the turbulence models for the simulation of the flow over a national advisory committee for aeronautics (NACA) 0012 airfoil. *Journal of Mechanical Engineering Research* 4(3), 100–111. doi:<https://doi.org/10.5897/JMER11.074>.
- Drela, M., 1989. Xfoil: An analysis and design system for low Reynolds number airfoils, in: *Low Reynolds Number Aerodynamics Conference*.
- Drela, M., Youngren, H., 2000. XFOIL Subsonic Airfoil Development System. <http://web.mit.edu/drela/Public/web/xfoil/>.
- Dreyfus, G., 2005. *Neural Networks: Methodology and Applications*. Springer, Germany. pp. 629–660.

- Eason, R., Sun, C., Dick, A., Nagarajaiah, S., 2013. Attenuation of a linear oscillator using a nonlinear and a semi-active tuned mass damper in series. *Journal of Sound and Vibration* 332(1), 154–166. doi:<https://doi.org/10.1016/j.jsv.2012.07.048>.
- Ebrahimi, A., Mardani, R., 2018. Tip-vortex noise reduction of a wind turbine using a winglet. *Journal of Energy Engineering* 144(1). doi:[https://doi.org/10.1061/\(ASCE\)EM.1943-7889.0000909](https://doi.org/10.1061/(ASCE)EM.1943-7889.0000909).
- Eleni, D., Athanasios, T., Dionissios, M., 2012. Evaluation of the turbulence models for the simulation of the flow over a national advisory committee for aeronautics (NACA) 0012 airfoil. *Journal of Mechanical Engineering Research* 4(3), 100–111. doi:<https://doi.org/10.5897/JMER11.074>.
- Elshaer, A., Bitsuamlak, G., 2018. Multiobjective aerodynamic optimization of tall building openings for wind-induced load reduction. *Journal of Structural Engineering* 144.
- Elshaer, A., Bitsuamlak, G., El Damatty, A., 2017. Enhancing wind performance of tall buildings using corner aerodynamic optimization. *Engineering Structures* 136, 133–148.
- Esteki, K., Bagchi, A., Sedaghati, R., 2011. Semi-active tuned mass damper for seismic applications, in: *Smart Materials, Structures NDT in Aerospace Conference*.
- Fortuner, B., 2017. Can neural networks solve any problem. From Towards Data Science. <https://towardsdatascience.com/can-neural-networks-really-learn-any-function-65e106617fc6/>.
- Frazier, P., 2018. A tutorial on bayesian optimization. *arXiv Machine Learning*. doi:<https://arxiv.org/abs/1807.02811>.
- Friedman, A., Dyke, S., Philips, B., Ahn, R., Dong, B., Chae, Y., Castaneda, N., Jiang, Z., Zhang, J., Cha, Y., Ozdagli, A., Spencer Jr., B., Ricles, J., Christenson, R., Agrawal, A., Sause, R., 2014. Large-scale real-time hybrid simulation for evaluation of advanced damping system performance. *Journal of Structural Engineering* 141(6). doi:[https://doi.org/10.1061/\(ASCE\)ST.1943-541X.0001093](https://doi.org/10.1061/(ASCE)ST.1943-541X.0001093).
- Gairola, A., Bitsuamlak, G., 2019. Numerical tornado modeling for common interpretation of experimental simulators. *Journal of Wind Engineering and Industrial Aerodynamics* 186, 32–48. doi:<https://doi.org/10.1016/j.jweia.2018.12.013>.
- Germanischer Lloyd (GL), 2010. *Guidelines for the Certification of Wind Turbines*. Hamburg, Germany.
- Glauert, H., 1935. Airplane propellers. *Aerodynamic Theory*, 169–360.
- Global Wind Energy Council (GWEC), 2017. *Global Wind Statistics 2017*. Brussels, Belgium.
- Grinderslev, C., Sorensen, N., Horcas, S., Troldborg, N., Zahle, F., 2021. Wind turbines in atmospheric flow: fluid-structure interaction simulations with hybrid turbulence modeling. *Wind Energy Science* 6, 627–643. doi:[https://doi.org/10.1061/\(ASCE\)EY.1943-7897.0000254](https://doi.org/10.1061/(ASCE)EY.1943-7897.0000254).

- Hakuno, M., Shidawara, M., Hara, T., 1969. Dynamic destructive test of a cantilever beam, controlled by an analog computer. *Proceedings of the Japanese Society of Civil Engineers* 171, 1–9.
- Haldar, S., Basu, D., 2016. Effect of climate change on the reliability of offshore wind turbine foundations, in: *Geo Chicago*.
- Hall, M., Moreno, J., Thiagarajan, K., 2014. Performance specifications for real-time hybrid testing of 1:50-scale floating wind turbine models, in: *International Conference on Ocean, Offshore and Arctic Engineering*.
- Hashemi, M., Al-Ogaidi, Y., Al-Mahaidi, R., Kalfat, R., Tsang, H., Wilson, J., 2017. Application of hybrid simulation for collapse assessment of post-earthquake cfrp-repaired rc columns. *Journal of Structural Engineering* 143(1). doi:[https://doi.org/10.1061/\(ASCE\)ST.1943-541X.0001629](https://doi.org/10.1061/(ASCE)ST.1943-541X.0001629).
- Hau, E., 2006. *Wind Turbines; Fundamentals, Technologies, Application, Economics*. Springer.
- He, E., Hu, Y., Zhang, Y., 2017. Optimization design of tuned mass dampers for vibration suppression of a barge-type offshore floating wind turbine. *Journal of Engineering for the Maritime Environment* 231(1), 302–315. doi:<https://doi.org/10.1177/1475090216642466>.
- Hemmati, A., Oterkus, E., 2018. Semi-active structural control of offshore wind turbines considering damage development. *Journal of Marine Science and Engineering* 6(3), 102. doi:<https://doi.org/10.3390/jmse6030102>.
- House, D., 2014. *Spline Curves*. <https://people.cs.clemson.edu/~dhouse/courses/405/notes/splines.pdf>.
- Hrovat, D., Barak, P., Rabins, M., 1983. Semi-active versus passive or active tuned mass dampers for structural control. *Journal of Engineering Mechanics* 109(3), 691–705. doi:[https://doi.org/10.1061/\(ASCE\)0733-9399\(1983\)109:3\(691\)](https://doi.org/10.1061/(ASCE)0733-9399(1983)109:3(691)).
- Hu, G., Kwok, K., 2020. Predicting wind pressures around circular cylinders using machine learning techniques. *Journal of Wind Engineering and Industrial Aerodynamics* 198. doi:<https://doi.org/10.1016/j.jweia.2020.104099>.
- Hu, G., Liu, L., Tao, D., Song, J., Tse, K., Kwok, K., 2020. Deep learning-based investigation of wind pressures on tall building under interference effects. *Journal of Wind Engineering and Industrial Aerodynamics* 201. doi:<https://doi.org/10.1016/j.jweia.2020.104138>.
- Huang, C., Arrigan, J., Nagarajaiah, S., Basu, B., 2010. Semi-active algorithm for edgewise vibration control in floating wind turbine blades, in: *Earth and Space 2010 Conference*.

- Huang, L., Huang, P., LeBeau, R., Hauser, T., 2004. Numerical study of blowing and suction control mechanism on NACA0012 airfoil. *Journal of Aircraft* 41(5). doi:<https://doi.org/10.2514/1.2255>.
- Hussan, M., Rahman, M., Sharmin, F., Kim, D., Do, J., 2018. Multiple tuned mass damper for multi-mode vibration reduction of offshore wind turbine under seismic excitation. *Ocean Engineering* 160, 449–460. doi:<https://doi.org/10.1016/j.oceaneng.2018.04.041>.
- Hussan, M., Sharmin, F., Kim, D., 2017. Multiple tuned mass damper based vibration mitigation of offshore wind turbine considering soil-structure interaction. *Mechanical Systems and Signal Processing* 105, 338–360. doi:<https://doi.org/10.1016/j.ymssp.2017.12.011>.
- Imraan, M., Sharma, R., Flay, R., 2013. Wind tunnel testing of a wind turbine with telescopic blades: The influence of blade extension. *Energy* 53, 22–32. doi:<https://doi.org/10.1080/j.energy.2013.03.008>.
- International Electrotechnical Commission (IEC), 2005. IEC 61400-1 International Standard: Wind turbines 3rd ed. Geneva, Switzerland.
- Jacobs, E., Sherman, A., 1939. Airfoil section characteristics as affected by variations of the Reynolds number. National Advisory Committee for Aeronautics. doi:[https://doi.org/10.1016/s0016-0032\(37\)90818-4](https://doi.org/10.1016/s0016-0032(37)90818-4).
- Jacobs, E., Ward, K., Pinkerton, R., 1932. The characteristics of 78 related airfoil sections from tests in the variable-density wind tunnel. NACA, Washington, D.C., U.S.A.
- Jimenez, A., Munoz, C., Marquez, F., 2017. Machine learning for wind turbine blades maintenance management. *Energies* 11(13). doi:<https://doi.org/10.3390/en11010013>.
- Jin, X., Cheng, P., Chen, W., Li, H., 2018. Prediction model of velocity field around circular cylinder over various Reynolds numbers by fusion convolutional neural networks based on pressure on the cylinder. *Physics of Fluids* 30. doi:<https://doi.org/10.1063/1.5024595>.
- Johari, H., Durgin, W., 1998. Direct measurement of circulation using ultrasound. *Experiments in Fluids* 25, 445–454. doi:<https://doi.org/10.1007/s003480050250>.
- Jonkman, J., B.M., Hayman, G., Jonkman, B., Mudafort, R., Platt, A., Sprague, M., 2021. OpenFAST. <https://github.com/OpenFAST/openfast>.
- Jonkman, J., Butterfield, S., Musial, W., Scott, G., 2009. NREL/TP-500-38060: Definition of a 5-MW Reference Wind Turbine for Offshore System Development. Golden, U.S.A.
- Kao, C., Loh, C., 2011. Monitoring of long-term static deformation data of Fei-Tsui arch dam using artificial neural network-based approaches. *Structural Control and Health Monitoring* 20(3), 282–303. doi:<https://doi.org/10.1002/stc.492>.

- Kareem, A., 2020. Emerging frontiers in wind engineering: Computing, stochastics, machine learning and beyond. *Journal of Wind Engineering and Industrial Aerodynamics* 206. doi:<https://doi.org/10.1016/j.jweia.2020.104320>.
- Karimirad, M., Bachynski, E., 2017. Sensitivity analysis of limited actuation for real-time hybrid model testing of 5MW bottom-fixed offshore wind turbines. *Energy Procedia* 137, 14–25. doi:<https://doi.org/10.1016/j.egypro.2017.10.331>.
- Katsanos, E., Thöns, S., Georgakis, C., 2016. Wind turbines and seismic hazard: a state-of-the-art review. *Wind Energy* 19(11), 2113–2133. doi:<https://doi.org/10.1002/we.1968>.
- Kaveh, A., Pirgholizadeh, S., Khademhosseini, O., 2015. Semi-active tuned mass damper performance with optimized fuzzy controller using CSS algorithm. *Asian Journal of Civil Engineering* 16(5), 587–606.
- Kelley, N., Jonkman, B., 2012. TurbSim. National Renewable Energy Laboratory, Washington DC, USA.
- Kelly, N., Jonkman, B., 2012. TurbSim. National Renewable Energy Laboratory, Washington D.C., USA.
- Khandui, A., Bedard, T., Stathopoulos, T., 1997. Modelling wind-induced interference effects using backpropagation neural networks. *Journal of Wind Engineering and Industrial Aerodynamics* 72, 71–79. doi:[https://doi.org/10.1016/S0167-6105\(97\)00259-6](https://doi.org/10.1016/S0167-6105(97)00259-6).
- Kimball, R., Goupee, A., Fowler, M., de Ridder, E., Helder, J., 2014. Wind/wave basin verification of a performance-matched scale-model wind turbine on a floating offshore wind turbine platform, in: *International Conference on Ocean, Offshore and Arctic Engineering*.
- Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., Inman, D., 2019. 1D convolution neural networks and applications: A survey. *arXiv Signal Processing*. doi:<https://arxiv.org/abs/1905.03554>.
- Kolay, C., Ricles, J., 2017. Improved explicit integration algorithms for structural dynamic analysis with unconditional stability and controllable numerical dissipation. *Journal of Earthquake Engineering* 5, 771–792. doi:<https://doi.org/10.1080/13632469.2017.1326423>.
- Kolay, C., Ricles, J., Marullo, T., Mahvashmohammadi, A., Sauce, R., 2015. Implementation and application of the unconditionally stable explicit parametrically dissipative KR- α method for real-time hybrid simulation. *Earthquake Engineering and Structural Dynamics* 44(5). doi:<https://doi.org/10.1002/eqe.2484>.
- Koukina, E., Kanner, S., Yeung, R., 2015. Actuation of wind-loading torque on vertical axis turbines at model scale, in: *OCEANS*.
- Kühn, M., 2001. Dynamics and design optimisation of offshore wind energy conversion systems. Delft University of Technology Ph.D. Thesis, Delft, Netherlands.

- Lackner, M., Rotea, M., 2010. Passive structural control offshore wind turbines. *Journal of Wind Energy* 14(3), 373–388. doi:<https://doi.org/10.1002/we.426>.
- Lackner, M., Rotea, M., 2011. Structural control of floating wind turbines. *Journal of Mechatronics* 21(4), 704–719. doi:<https://doi.org/10.1016/j.mechatronics.2010.11.007>.
- Ladson, C., 1988. Effect of Independent Variation of Mach and Reynolds Numbers on the Low-Speed Aerodynamic Characteristics of the NACA 0012 Airfoil Section. NASA Technical Memorandum 4074.
- Laitone, E., 1997. Wind tunnel tests of wings at Reynolds numbers below 70 000. *Experiments in Fluids* 23, 405–409. doi:<https://doi.org/10.1007/s003480050128>.
- Lalonde, E., Bitsuamlak, G., Dai, K., 2020a. Effect of aeroelastic blade deflections on the overall response of a 5 MW wind turbine tower, in: Canadian Society of Civil Engineering Conference.
- Lalonde, E., Dai, K., Bitsuamlak, G., Zhao, Z., 2020b. Comparison of semi-active and passive tuned mass damper systems for vibration control of a wind turbine. *Wind and Structures* 30(6), 663–678. doi:<https://doi.org/10.12989/was.2020.30.6.663>.
- Lalonde, E., Dai, K., Lu, W., Bitsuamlak, G., 2019. Wind turbine testing methods and application of hybrid testing: A review. *Wind and Structures* 29(3), 195–207. doi:<https://doi.org/10.12989/was.2019.29.3.195>.
- Lalonde, E., Vischschraper, B., Bitsuamlak, G., Dai, K., 2021. Comparison of neural network types and architectures for generating a surrogate aerodynamic wind turbine blade model. *Journal of Wind Engineering and Industrial Aerodynamics* 216. doi:[10.1016/j.jweia.2021.104696](https://doi.org/10.1016/j.jweia.2021.104696).
- Langer-Moller, A., Lowe, J., Kessler, R., 2017. Investigation of the NREL phase VI experiment with the incompressible CFD solver THETA. *Wind Energy* 20(9), 1529–1549. doi:<https://doi.org/10.1002/we.2107>.
- Lee, J., Zhao, F., 2021. Global Wind Report 2021. Global Wind Energy Council, Brussels, Belgium. <https://gwec.net/global-wind-report-2021/>.
- Lee, K., Huque, Z., Kommalapati, R., Han, S., 2017. Fluid-structure interaction analysis of NREL phase VI wind turbine: Aerodynamic force evaluation and structural analysis using FSI analysis. *Renewable Energy* 113, 512–531. doi:<https://doi.org/10.1016/j.renene.2017.02.071>.
- Leishmann, J., 2002. Challenges in modelling the unsteady aerodynamics of wind turbines. *Wind Energy* 5, 85–132. doi:<https://doi.org/10.1002/we.62>.
- Li, S., Laima, S., Li, H., 2018. Data-driven modeling of vortex-induced vibration of a long-span suspension bridge using decision tree learning and support vector regression. *Journal of Wind Engineering and Industrial Aerodynamics* 172. doi:<https://doi.org/10.1016/j.jweia.2017.10.022>.

- Li, X., Ozdagli, A., Dyke, S., Liu, X., Christenson, R., 2017. Development and verification of distributed real-time hybrid simulation methods. *Journal of Computing in Civil Engineering* 31(4). doi:[https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000654](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000654).
- Li, Y., Paik, K., Xing, T., Carrica, P., 2012. Dynamic overset CFD simulations of wind turbine aerodynamics. *Renewable Energy* 37(1), 285–298. doi:<https://doi.org/10.1016/j.renene.2011.06.029>.
- Lim, C., Neild, S., Stoten, D., Drury, D., Taylor, C., 2007. Adaptive control strategy for dynamic substructuring tests. *Journal of Engineering Mechanics* 133(8), 864–873. doi:[https://doi.org/10.1061/\(ASCE\)0733-9399\(2007\)133:8\(864\)](https://doi.org/10.1061/(ASCE)0733-9399(2007)133:8(864)).
- Lin, Z., Liu, X., 2020. Wind power forecasting of an offshore wind turbine based on high-frequency SCADA data and deep learning neural network. *Energy* 201. doi:<https://doi.org/10.1016/j.energy.2020.117693>.
- Liu, S., Janajreh, I., 2012. Development and application of an improved blade element momentum method model on horizontal axis wind turbines. *International Journal of Energy and Environmental Engineering* 3(30). doi:<https://doi.org/10.1186/2251-6832-3-30>.
- Liu, Y., Xiao, Q., Incecik, A., Peyrard, C., 2018. Aeroelastic analysis of a floating offshore wind turbine in platform-induced surge motion using a fully coupled CFD-MBD method. *Wind Energy* 22(1), 1–20. doi:<https://doi.org/10.1002/we.2265>.
- Liu, Z., Lu, S., Ishihara, T., 2020b. Large eddy simulations of wind turbine wakes in typical complex topographies. *Wind Energy* doi:<https://doi.org/10.1002/we.2606>.
- Liu, H., Z.Z.J.H.L.Q.L.Y., Leng, J., 2020. A novel method to predict the stiffness evolution of in-service wind turbine blades based on deep learning models. *Composite Structures* 252. doi:<https://doi.org/10.1016/j.compstruct.2020.112702>.
- Luo, H., H.M., Xiong, W., 2019. Application of a recurrent neural network and simplified semianalytical method for continuous strain histories estimation. *Shock and Vibration* doi:<https://doi.org/10.1155/2019/7289314>.
- Ma, H., Zhang, D., Ma, Q., 2015. Scale model experimental of a prestressed concrete wind turbine tower. *Wind and Structures* 21(3), 353–367. doi:<https://doi.org/10.12989/was.2015.21.3.353>.
- Macquart, T., Maheri, A., Busawon, K., 2012. Improvement of the accuracy of the blade element momentum theory method in wind turbine aerodynamics analysis, in: *International Symposium on Environment-Friendly Energies and Applications*.
- Madhiarasan, M., Deepa, S., 2017. Comparative analysis on hidden neurons estimation in multi-layer perceptron neural networks for wind speed forecasting. *Artificial Intelligence Review* 48, 449–471. doi:<https://doi.org/10.1007/s10462-016-9506-6>.

- Madsen, H., Mikkelsen, R., Oye, S., Bak, C., Johansen, J., 2007. A detailed investigation of the blade element momentum (BEM) model based on analytical and numerical results and proposal for modifications of the BEM model. *Journal of Physics: Conference Series* 75. doi:<https://doi.org/10.1088/1742-6596/75/1/012016>.
- Maheri, A., Noroozi, S., Toomer, C., Vinney, J., 2006. Damping the fluctuating behaviour and improving the convergence rate of the axial induction factor in the BEMT based rotor aerodynamic codes, in: *European Wind Energy Conference*.
- Mancini, S., Boorsma, K., Caboni, M., Cormier, M., Lutz, T., Schito, P., Zasso, A., 2020. Characterization of the unsteady aerodynamic response of a floating offshore wind turbine to surge motion. *Wind Energy Science* 5, 1713–1730. doi:<https://doi.org/10.5194/wes-5-1713-2020>.
- Mao, Z., Dai, K., Zhao, Z., Wang, Y., Meng, J., Zhao, C., 2018. Shaking table test study on seismic responses of a wind turbine under ground motions with different spectral characteristics. *International Journal of Advanced Engineering Science and Technological Research* 50(3), 125–133. doi:<https://doi.org/10.15961/j.jsuese.201800369>.
- Mardfekri, M., Gardoni, P., 2015. Multi-hazard reliability assessment of offshore wind turbines. *Wind Energy* 18(8), 1433–1450. doi:<https://doi.org/10.1002/we.1768>.
- Martinat, G., Braza, M., Hoarau, Y., Harran, G., 2008. Turbulence modelling of the flow past a pitching NACA0012 airfoil at 10^5 and 10^6 Reynolds numbers. *Journal of Fluids and Structures* 24(8), 1294–1303. doi:<https://doi.org/10.1016/j.jfluidstructs.2008.08.002>.
- Martinez-Aranda, S., Gracia-Gonzalez, A., Parras, L., Velazquez-Navarro, J., Del Pino, C., 2016. Comparison of the aerodynamic characteristics of the NACA0012 airfoil at low-to-moderate Reynolds numbers for any aspect ratio. *International Journal of Aerospace Sciences* 4(1), 1–8. doi:<https://doi.org/10.5923/j.aerospace.20160401.01>.
- Martinez-Tossas, L., Leonardi, S., 2012. *Wind Turbine Modelling for Computational Fluid Dynamics*, (NREL/SR-5000-55054). National Renewable Energy Laboratory.
- Martynowicz, P., 2015. Vibration control of wind turbine tower-nacelle model with magnetorheological tuned vibration absorber. *Journal of Vibration and Control* 1-22. doi:<https://doi.org/10.1177/1077546315591445>.
- Martynowicz, P., 2016. Study of vibration control using laboratory test rig of wind turbine tower-nacelle system with MR damper based tuned vibration absorber. *Bulletin of the Polish Academy of Sciences* 64(2). doi:<https://doi.org/10.1515/bpasts-2016-0040>.
- Martynowicz, P., 2017. Control of a magnetorheological tuned vibration absorber for wind turbine application using the refined force tracking algorithm. *Journal of Low Frequency Noise, Vibration and Active Control* 36(4), 339–353. doi:<https://doi.org/10.1177/1461348417744299>.

- Matyushenko, A., Kotov, E., Garbaruk, A., 2017. Calculations of flow around airfoils using two-dimensional RANS: an analysis of the reduction in accuracy. *St. Petersburg Polytechnical University Journal: Physics and Mathematics* 3(1), 15–21. doi:<https://doi.org/10.1016/j.spjpm.2017.03.004>.
- McAlister, K., Carr, L., McCroskey, W., 1978. Dynamic Stall Experiments of the NACA 0012 Airfoil. NASA Technical Paper 1100.
- McCroskey, W., 1987. A Critical Assessment of Wind Tunnel Results for the NACA 0012 Airfoil. NASA Technical Report 87-A-5.
- McCrum, D., Broderick, B., 2013. Evaluation of a substructured soft-real time hybrid test for performing seismic analysis of complex structural systems. *Computers and Structures* 129, 111–119. doi:<https://doi.org/10.1016/j.compstruc.2013.02.009>.
- McCrum, D., Williams, M., 2016. An overview of seismic hybrid testing of engineering structures. *Engineering Structures* 118, 240–261. doi:<https://doi.org/10.1016/j.engstruct.2016.03.039>.
- McTavish, S., Feszty, D., Nitzsche, F., 2013. Evaluating Reynolds number effects in small-scale wind turbine experiments. *Journal of Wind Engineering and Industrial Aerodynamics* 120, 81–90. doi:<https://doi.org/10.1016/j.jweia.2013.07.006>.
- Ministry of Housing and Urban-Rural Development of the People's Republic of China (MHURD), 2010. Code for seismic design of buildings. Beijing, China.
- Mo, R., Kang, H., Li, M., Zhao, X., 2017. Seismic fragility analysis of monopile offshore wind turbines under different operational conditions. *Energies* 10. doi:<https://doi.org/10.3390/en10071037>.
- Moni, M., Hwang, Y., Kwon, O., Kim, H., Jeong, U., 2020. Real-time aeroelastic hybrid simulation of a base-pivoting building model in a wind tunnel. *Frontiers in Built Environment* 6. doi:<https://doi.org/10.3389/fbuil.2020.560672>.
- Mosqueda, G., Stojadinovic, B., Mahin, S., 2007. Real time error monitoring for hybrid simulation. part i: Methodology and experimental verification. *Journal of Structural Engineering* 133(8), 1100–1108. doi:[https://doi.org/10.1061/\(ASCE\)0733-9445\(2007\)133:8\(1100\)](https://doi.org/10.1061/(ASCE)0733-9445(2007)133:8(1100)).
- Murtagh, P., Ghosh, A., Basu, B., Broderick, B., 2008. Passive control of wind turbine vibrations including blade/tower interaction and rotationally sampled turbulence. *Wind Energy* 11, 305–317. doi:<https://doi.org/10.1002/we.249>.
- Nagarajaiah, S., 2009. Adaptive passive, semiactive, smart tuned mass dampers: identification and control using empirical mode decomposition, Hilbert transform, and short-term Fourier transform. *Journal of Structural Control and Health Monitoring* 16(7), 800–841. doi:<https://doi.org/10.1002/stc.349>.

- Nagarajaiah, S., Sonmez, E., 2007. Structures with semiactive variable stiffness single/multiple tuned mass dampers. *Journal of Structural Engineering* 133(1), 67–77. doi:[https://doi.org/10.1061/\(ASCE\)0733-9445\(2007\)133:1\(67\)](https://doi.org/10.1061/(ASCE)0733-9445(2007)133:1(67)).
- Nakashima, N., Kato, H., Takaoka, E., 1992. Development of real-time pseudo dynamic testing. *Earthquake Engineering and Structural Dynamics* 21, 79–92. doi:<https://doi.org/10.1002/eqe.4290210106>.
- Navalkar, S., van Solingen, E., van Wingerden, J., 2015. Wind tunnel testing of subspace predictive repetitive control for variable pitch wind turbines. *Transactions on Control Systems Technology* 23(6), 2101–2116. doi:<https://doi.org/10.1109/TCST.2015.2399452>.
- Newmark, N., 1959. A method of computation for structural dynamics. *Journal of Engineering Mechanics Division* 85, 67–94.
- Newth, G., 2004. A general solution for velocity-squared damping. *The Mathematical Gazette* 88, 556–561.
- Nielsen, M., 2019. A visual proof that neural nets can compute any function. *Neural Networks and Deep Learning*, Chapter 4. <http://neuralnetworksanddeeplearning.com/chap4.html>.
- Nordanger, K., Holdahl, R., Kvarving, A., Rasheed, A., Kvamsdal, T., 2015. Implementation and comparison of three isogeometric Navier–Stokes solvers applied to simulation of flow past a fixed 2D NACA0012 airfoil at high Reynolds number. *Computer Methods in Applied Mechanics and Engineering* 284, 664–668. doi:<https://doi.org/10.1016/j.cma.2014.10.033>.
- Nunez-Casado, C., Lopez-Garcia, O., de las Heras, E., Cuerva Tejero, A., Gallego-Castillo, C., 2017. Assembly strategies of wind turbine towers for minimum fatigue damage. *Wind and Structures* 25(6), 569–588. doi:<https://doi.org/10.12989/was.2017.25.6.569>.
- Ojaghi, M., Williams, M., Dietz, M., Blakeborough, A., Martinez, I., 2014. Real-time distributed hybrid testing: coupling geographically distributed scientific equipment across the internet to extend seismic testing capabilities. *Earthquake Engineering and Structural Dynamics* 43, 1023–1043. doi:<https://doi.org/10.1002/eqe.2385>.
- Owji, H., Shirazi, A., Sarvestani, H., 2011. A comparison between a new semi-active tuned mass damper and an active tuned mass damper. *Procedia Engineering* 14, 2779–2787. doi:<https://doi.org/10.1016/j.proeng.2011.07.350>.
- Park, S., Lackner, M., Cross-Whiter, J., Tsouroukdissian, A., La Cava, W., 2016. An investigation of passive and semi-active tuned mass dampers for a tension leg platform floating offshore wind turbine in ULS conditions, in: *OMAE 2016 Proceedings*.
- Park, S., Lackner, M., Pourazarm, P., Tsouroukdissian, A., Cross-Whiter, J., 2019. An investigation on the impacts of passive and semiactive structural control on a fixed bottom and a floating offshore wind turbine. *Journal of Wind Energy* 22, 1451–1471. doi:<https://doi.org/10.1002/we.2381>.

- Petrillu, J., Paul, R., Gopalarathnam, A., Frink, N., 2013. A CFD database for airfoils and wings at post-stall angles of attack, in: AIAA Applied Aerodynamics Conference.
- Phi, M., 2018. Illustrated Guide to LSTM's and GRU's: A step by step explanation. From Towards Data Science. <https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21>.
- Phillips, B., Spencer, B., 2012a. Model-based feedforward-feedback actuator control for real-time hybrid simulation. *Journal of Structural Engineering* 139(7). doi:[https://doi.org/10.1061/\(ASCE\)ST.1943-541X.0000606](https://doi.org/10.1061/(ASCE)ST.1943-541X.0000606).
- Phillips, B., Spencer, B., 2012b. Model-based multiactuator control for real-time hybrid simulation. *Journal of Engineering Mechanics* 139(2). doi:[https://doi.org/10.1061/\(ASCE\)EM.1943-7889.0000493](https://doi.org/10.1061/(ASCE)EM.1943-7889.0000493).
- Pinkaew, T., Fujino, Y., 2001. Effectiveness of semi-active tuned mass dampers under harmonic excitation. *Engineering Structure* 23(7), 850–856. doi:[https://doi.org/10.1016/S0141-0296\(00\)00091-2](https://doi.org/10.1016/S0141-0296(00)00091-2).
- Plummer, A., 2006. Model-in-the-loop testing. *Institution of Mechanical Engineers* 220, 183–199. doi:<https://doi.org/10.1243/09596518JSCE207>.
- Prowell, I., Veletzos, M., Elgamal, A., Restrepo, J., 2009. Experimental and numerical seismic response of a 65kW wind turbine. *Journal of Earthquake Engineering* 13, 1172–1190. doi:<https://doi.org/10.1080/13632460902898324>.
- Rahman, M., Ong, Z., Chong, W., Julai, S., Khoo, S., 2015. Performance enhancement of wind turbine systems with vibration control: A review. *Journal of Renewable and Sustainable Energy Reviews* 51, 43–54. doi:<https://doi.org/10.1016/j.rser.2015.05.078>.
- Ramos, M., Mosqueda, G., Hashemi, M., 2016. Large scale hybrid simulation of a steel moment frame building structure through collapse. *Journal of Structural Engineering* 142(1). doi:[https://doi.org/10.1061/\(ASCE\)ST.1943-541X.0001328](https://doi.org/10.1061/(ASCE)ST.1943-541X.0001328).
- Rasmussen, F., Hansen, M., Thomsen, K., Larsen, T., Bertagnolio, F., Johansen, J., Madsen, H., Bak, C., Hansen, A., 2003. Present status of aeroelasticity of wind turbines. *Wind Energy* 6, 213–228. doi:<https://doi.org/10.1002/we.98>.
- Reddy, J., 1993. *An introduction to the finite element method* (2nd edition). McGraw-Hill, Inc.
- Refan, M., 2009. *Aerodynamic Performance of a Small Horizontal Axis Wind Turbine*. University of Western Ontario Master Thesis, London, Canada.
- Refan, M., Hangan, H., 2012. Aerodynamic performance of a small horizontal axis wind turbine. *Journal of Solar Energy Engineering* 134(2). doi:<https://doi.org/10.1115/1.4005751>.
- Ren, K., Hu, J., Xiong, X., Zhang, L., Wei, J., 2009. Validation of turbulence models in STAR-CCM+ by N.A.C.A. 23012 airfoil characteristics, in: ASEE Northeast Section Conference.

- Rezaee, M., Aly, A., 2016. Vibration control in wind turbines for performance enhancement: A comparative study. *Wind and Structures* 22(1), 107–131. doi:<https://doi.org/10.12989/was.2016.22.1.107>.
- Ribeiro, A., Awruch, A., Gomes, H., 2012. An airfoil optimization technique for wind turbines. *Applied Mathematical Modelling* 36(10), 4898–4907. doi:<https://doi.org/10.1016/j.apm.2011.12.026>.
- Ricciardelli, F., Occhiuzzi, A., Clemente, P., 2000. Semi-active tuned mass damper control strategy for wind excited structures. *Journal of Wind Engineering and Industrial Aerodynamics* 88(1), 57–74. doi:[https://doi.org/10.1016/S0167-6105\(00\)00024-6](https://doi.org/10.1016/S0167-6105(00)00024-6).
- Richmond, M., Sobey, A., Pandit, R., Kolios, A., 2020. Stochastic assessment of aerodynamics within offshore wind farms based on machine-learning. *Renewable Energy* 161, 650–661. doi:<https://doi.org/10.1016/j.renene.2020.07.083>.
- Riso National Laboratory (Riso), 2002. *Guidelines for Design of Wind Turbines*, 2nd edition. Copenhagen, Denmark.
- Rizcallah, J., 2019. Revisiting the coulomb-damped harmonic oscillator. *European Journal of Physics* 40(5). doi:<https://doi.org/10.1088/1361-6404/ab33d1>.
- Rong, X., Xu, R., Wang, H., Feng, S., 2017. Analytical solution for natural frequency of monopile supported wind turbine towers. *Wind and Structures* 25(5), 459–474. doi:<https://doi.org/10.12989/was.2017.25.5.459>.
- Sadowski, A., Camara, A., Malaga-Chuquitaype, C., K., D., 2017. Seismic analysis of a tall metal wind turbine support tower with realistic geometric imperfections. *Wind and Structures* 46, 201–219. doi:<https://doi.org/10.1002/eqe.2785>.
- Salehi, H., Burgueño, R., 2018. Emerging artificial intelligence methods in structural engineering. *Engineering Structures* 171, 170–189. doi:<https://doi.org/10.1016/j.engstruct.2018.05.084>.
- Selig, M., McGranahan, B., 2004. *Wind tunnel aerodynamic tests of six airfoils for use on small wind turbines (NREL/SR-500-34515)*. National Renewable Energy Laboratory.
- Sessarego, M., Feng, J., Ramos-Garcia, N., Horcas, S., 2020. Design optimization of a curved wind turbine blade using neural networks and an aero-elastic vortex method under turbulent inflow. *Renewable Energy* 146. doi:<https://doi.org/10.1016/j.renene.2019.07.046>.
- Shao, X., Reinhorn, A., Sivaselvan, M., 2011. Real time hybrid simulation using shake tables and dynamic actuators. *Journal of Structural Engineering* 137(7), 748–760.
- Sheldahl, R., Klimas, P., 1981. *Aerodynamic characteristics of seven symmetrical airfoil sections through 180-degree angle of attack for use in aerodynamic analysis of vertical axis wind turbines*. Sandia National Labs, Albuquerque, USA.

- Shida, Y., Kuwahara, K., Ono, K., Takami, H., 1987. Computation of dynamic stall of a NACA-0012 airfoil. *American Institute of Aeronautics and Astronautics Journal* 25(3), 408–413. doi:<https://doi.org/10.2514/3.9638>.
- Shirazadeh, K., Hangan, H., Crawford, C., Tari, P., 2021. Investigating the loads and performance of a model horizontal axis wind turbine under reproducible IEC extreme operational conditions. *Wind Energy Science* 6, 477–489. doi:<https://doi.org/10.5194/wes-6-477-2021>.
- Sim, H., Prowell, I., Elgamal, A., Uang, C., 2014. Flexural tests and associated study of a full-scale 65-kW wind turbine tower. *Journal of Structural Engineering* 140(5). doi:[https://doi.org/10.1061/\(ASCE\)ST.1943-541X.0000924](https://doi.org/10.1061/(ASCE)ST.1943-541X.0000924).
- Simiu, E., Scanlan, R., 1986. *Wind Effect on Structures*. John Wiley & Sons.
- Smith, V., Mahmoud, H., 2016. Multihazard assessment of wind turbine towers under simultaneous application of wind, operation, and seismic loads. *Journal of Performance of Constructed Facilities* 30(6). doi:[https://doi.org/10.1061/\(ASCE\)CF.1943-5509.0000898](https://doi.org/10.1061/(ASCE)CF.1943-5509.0000898).
- Song, B., Yi, Y., Wu, J., 2013. Study on seismic dynamic response of offshore wind turbine tower with monopile foundation based on m method. *Advanced Materials Research* 663, 686–691. doi:<https://doi.org/10.4028/www.scientific.net/AMR.663.686>.
- Song, W., Sun, C., Zuo, Y., Jahagiri, V., Lu, Y., Han, Q., 2020. Conceptual study of a real-time hybrid simulation framework for monopile offshore wind turbines under wind and wave loads 6. doi:<https://doi.org/10.3389/fbuil.2020.00129>.
- Sony, S., Dunphy, D., Sadhu, A., Capretz, M., 2021. A systematic review of convolutional neural network-based structural condition assessment techniques. *Engineering Structures* 226.
- Spencer, B., Finholt, T., Foster, I., Kesselman, C., Beldica, C., Futrelle, J., Gullapalli, S., Hubbard, P., Liming, L., Marcusiu, D., Pearlman, L., Severance, C., Yang, G., 2004. NEESgrid: A Distributed Collaborator for Advanced Earthquake Engineering Experiment and Simulation, in: *13th World Conference on Earthquake Engineering*.
- Stamatopoulos, G., 2013. Response of a wind turbine subjected to near-fault excitation and comparison with the Greek Seismic Code provisions. *Soil Dynamics and Earthquake Engineering* 46, 77–84. doi:<https://doi.org/10.1016/j.soildyn.2012.12.014>.
- Stein, V., Kaltenbach, H., 2016. Wind-tunnel modelling of the tip-speed ratio influence on the wake evolution. *Journal of Physics: Conference Series* 753. doi:<https://doi.org/10.1088/1742-6596/753/3/032061>.
- Stetco, A., Dinmohammadi, F., Zhao, Y., Robu, V., Flynn, D., Barnes, M., Keane, J., Nenadic, G., 2020. Machine learning methods for wind turbine condition monitoring: A review. *Renewable Energy* 133, 620–635. doi:<https://doi.org/10.1016/j.renene.2018.10.047>.

- Stewart, G., 2012. Load reduction of floating wind turbines using tuned mass dampers. University of Massachusetts Amherst Master Thesis, Amherst, USA.
- Sun, C., 2017. Mitigation of offshore wind turbine responses under wind and wave loading: Considering soil effects and damage. *Structural Control and Health Monitoring* 25(3). doi:<https://doi.org/10.1002/stc.2117>.
- Sun, C., Jahangiri, V., 2018. Bi-directional vibration control of offshore wind turbines using a 3D pendulum tuned mass damper. *Mechanical Systems and Signal Processing* 105, 338–360. doi:<https://doi.org/10.1016/j.ymssp.2017.12.011>.
- Sun, C., Nagarajaiah, S., 2013. Study on semi-active tuned mass damper with variable damping and stiffness under seismic excitations. *Journal of Structural Control and Health Monitoring* 21(6), 890–906. doi:<https://doi.org/10.1002/stc.1620>.
- Sun, C., Nagarajaiah, S., Dick, A., 2014. Experimental investigation of vibration attenuation using nonlinear tuned mass damper and pendulum tuned mass damper in parallel. *Journal of Nonlinear Dynamics* 78(4), 2699–2715. doi:<https://doi.org/10.1007/s11071-014-1619-3>.
- Tait, M., Isyumov, N., El Damatty, A., 2008. Performance of tuned liquid dampers. *Journal of Engineering Mechanics* 134(5), 417–427. doi:[https://doi.org/10.1061/\(ASCE\)0733-9399\(2008\)134:5\(417\)](https://doi.org/10.1061/(ASCE)0733-9399(2008)134:5(417)).
- Takanashi, K., 1974. Seismic failure analysis of structures by computer-pulsator on-line system. *Journal of the Institute of Industrial Science* 26(11), 13–25.
- Takanashi, K., Nakashima, M., 1987. Japanese activities on on-line testing. *Journal of Engineering Mechanics* 113(7), 1014–1032.
- Talatahari, S., Kaveh, A., Mohajer Rahbari, N., 2012. Parameter identification of Bouc-Wen model for MR fluid dampers using adaptive charged system search optimization. *Journal of Mechanical Science and Technology* 26(8), 2523–2534. doi:<https://doi.org/10.1007/s12206-012-0625-y>.
- Tang, Y., Lou, M., 2017. New unconditionally stable explicit integration algorithm for real-time hybrid testing. *Journal of Engineering Mechanics* 143(7). doi:[https://doi.org/10.1061/\(ASCE\)EM.1943-7889.0001235](https://doi.org/10.1061/(ASCE)EM.1943-7889.0001235).
- Ti, Z., Deng, X., Yang, H., 2020. Wake modeling of wind turbines using machine learning. *Applied Energy* 257. doi:<https://doi.org/10.1016/j.apenergy.2019.114025>.
- Tian, J., Gurley, K., Diaz, M., Fernandez-Caban, P., Masters, F., Fang, R., 2020. Low-rise gable roof buildings pressure prediction using deep neural networks. *Journal of Wind Engineering and Industrial Aerodynamics* 196. doi:<https://doi.org/10.1016/j.jweia.2019.104026>.

- Tran, T., Kim, D., 2017. A CFD study of coupled aerodynamic-hydrodynamic loads on a semisubmersible floating offshore wind turbine. *Wind Energy* 21(1). doi:<https://doi.org/10.1002/we.2145>.
- Tsouroukdissian, A., Park, S., Pourazarm, P., La Cava, W., Lackner, M., Lee, S., Cross-Whiter, J., 2016. Smart novel semi-active tuned mass damper for fixed-bottom and floating offshore wind, in: *Offshore Technology Conference*.
- Ueland, E., Skjetne, R., Vilsen, S., 2018. Force actuated real-time hybrid model testing of a moored vessel: A case study investigating force errors. *IFAC-PapersOnLine* 51(29), 74–79. doi:<https://doi.org/10.1016/j.ifacol.2018.09.472>.
- Valamanesh, V., Myers, A., 2014. Aerodynamic damping and seismic response of horizontal axis wind turbine towers. *Journal of Structural Engineering* 140. doi:[https://doi.org/10.1061/\(ASCE\)ST.1943-541X.0001018](https://doi.org/10.1061/(ASCE)ST.1943-541X.0001018).
- Van der Woude, C., Narasimhan, S., 2014. A study on vibration isolation for wind turbine structures. *Engineering Structures* 60, 223–234. doi:<https://doi.org/10.1016/j.engstruct.2013.12.028>.
- Veers, P., 1988. *Three-dimensional wind simulation*. Sandia National Laboratories, Albuquerque, USA.
- Versteeg, H., Malalasekera, W., 2007. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*, 2nd edition. Pearson.
- Vilsen, S., Sauder, T., Sorensen, A., Fore, M., 2019. Method for real-time hybrid model testing of ocean structures: Case study on horizontal mooring systems. *Ocean Engineering* 172, 46–58. doi:<https://doi.org/10.1016/j.oceaneng.2018.10.042>.
- Wang, L., Liu, X., Kolios, A., 2016. State of the art in the aeroelasticity of wind turbine blades: Aeroelastic modelling. *Renewable and Sustainable Energy Reviews* 64, 195–210. doi:<https://doi.org/10.1016/j.rser.2016.06.007>.
- Wang, L., Liu, X., Renevier, N., Stables, M., Hall, G., 2014. Nonlinear aeroelastic modelling for wind turbine blades based on blade element momentum theory and geometrically exact beam theory. *Energy* 76, 487–501. doi:<https://doi.org/10.1016/j.energy.2014.08.046>.
- Wang, Q., Sprague, M., Jonkman, J., Johnson, N., , Jonkman, B., 2017. Beamdyn: a high-fidelity wind turbine blade solver in the fast modular framework. *Wind Energy* 20(8), 1439–1462. doi:<https://doi.org/10.1002/we.2101>.
- Wang, T., McCormick, J., Nakashima, M., 2008. Verification test of a hybrid test system with distributed column base tests, in: *Analysis and Computation Specialty Conference*.
- Wang, X., Gao, W., Gao, T., Li, Q., Wang, J., Li, X., 2018. Robust model reference adaptive control design for wind turbine speed regulation simulated by using fast. *Journal of Energy Engineering* 144(2). doi:[https://doi.org/10.1061/\(ASCE\)EY.1943-7897.0000520](https://doi.org/10.1061/(ASCE)EY.1943-7897.0000520).

- Wang, Z., Zhao, Y., Li, F., Jiang, J., 2013. Extreme dynamic responses of MW-level wind turbine tower in the strong typhoon considering wind-rain loads. *Mathematical Problems in Engineering* doi:<https://doi.org/10.1155/2013/512530>.
- Watanabe, E., Kitada, T., Sugiura, K., Nagata, K., 2001. Parallel pseudo-dynamic seismic loading test on elevated bridge system through the internet, in: *East Asia-Pacific Conference on Structural Engineering and Construction*.
- Wu, J., Lu, X., Denny, A., Fan, M., Wu, J., 1998. Post-stall flow control on an airfoil by local unsteady forcing. *Journal of Fluid Mechanics* 371, 21–58. doi:<https://doi.org/10.1017/S0022112098002055>.
- Wu, T., Kareem, A., 2011. Modeling hysteretic nonlinear behavior of bridge aerodynamics via cellular automata nested neural network. *Journal of Wind Engineering and Industrial Aerodynamics* 99(4), 379–388. doi:<https://doi.org/10.1016/j.jweia.2010.12.011>.
- Yalla, S., Kareem, A., 2007. Dynamic load simulator: Actuation strategies and applications. *Journal of Engineering Mechanics* 133(8), 855–863. doi:[https://doi.org/10.1061/\(ASCE\)0733-9399\(2007\)133:8\(855\)](https://doi.org/10.1061/(ASCE)0733-9399(2007)133:8(855)).
- Yang, G., Spencer, B., Carlson, J., Sain, M., 2002. Large-scale MR fluid dampers: modeling and dynamic performance considerations. *Engineering Structures* 24, 309–323. doi:[https://doi.org/10.1016/S0141-0296\(01\)00097-9](https://doi.org/10.1016/S0141-0296(01)00097-9).
- Yang, X., Zhang, Y., Lv, W., Wang, D., 2021. Image recognition of wind turbine blade damage based on a deep learning model with transfer learning and an ensemble learning classifier. *Renewable Energy* 163, 386–397. doi:<https://doi.org/10.1016/j.renene.2020.08.125>.
- Yelmule, M., Anjuri, E., 2013. CFD predictions of NREL Phase VI Experiments in NASA/AMES Wind tunnel. *International Journal of Renewable Energy* 3(2).
- Yin, X., Zhao, X., 2019. Big data driven multi-objective predictions for offshore wind farm based on machine learning algorithms. *Energy* 186. doi:<https://doi.org/10.1016/j.energy.2019.07.034>.
- Zaman, K., Mckinzie, D., Rumsey, C., 1989. A natural low-frequency oscillation of the flow over an airfoil near stalling conditions. *Journal of Fluid Mechanics* 202, 403–442. doi:<https://doi.org/10.1017/S0022112089001230>.
- Zhang, R., Zhao, Z., Dai, K., 2019. Seismic response mitigation of a wind turbine tower using a tuned parallel inerter mass system. *Engineering Structures* 180, 29–39. doi:<https://doi.org/10.1016/j.engstruct.2018.11.020>.
- Zhang, Z., Basu, B., Nielsen, S., 2017. Real-time hybrid aeroelastic simulation of wind turbines with various types of full-scale tuned liquid dampers. *Wind Energy* 22(2), 239–256. doi:<https://doi.org/10.1002/we.2281>.

- Zhang, Z., Fitzgerald, B., 2020. Tuned mass-damper-inerter (TMDI) for suppressing edgewise vibrations of wind turbine blades. *Engineering Structures* 221. doi:<https://doi.org/10.1016/j.engstruct.2020.110928>.
- Zhang, Z., Li, J., Zhuge, P., 2014a. Failure analysis of large scale wind power structure under simulated typhoon. *Mathematical Problems in Engineering* doi:<https://doi.org/10.1155/2014/486524>.
- Zhang, Z., Nielsen, S., Basu, B., Li, J., 2015. Nonlinear modeling of tuned liquid dampers (TLDs) in rotating wind turbines blades from damping edgewise vibrations. *Journal of Fluids and Structures* 59, 252–269. doi:<https://doi.org/10.1016/j.jfluidstructs.2015.09.006>.
- Zhang, Z., Nielsen, S., Blaabjerg, F., Zhao, D., 2014b. Dynamics and control of lateral tower vibrations in offshore wind turbines by means of active generator torque. *Energies* 7. doi:<https://doi.org/10.3390/en7117746>.
- Zhang, Z., Staino, A., Basu, B., Nielsen, S., 2016. Performance evaluation of full-scale tuned liquid dampers (TLDs) for vibration control of large wind turbines using real time hybrid testing. *Engineering Structures* 126, 417–431. doi:<https://doi.org/10.1016/j.engstruct.2016.07.008>.
- Zhao, Z., Dai, K., Camara, A., Bitsuamlak, G., Sheng, C., 2019. Wind turbine tower failure modes under seismic and wind loads. *Journal of Performance of Constructed Facilities* 33(2). doi:[https://doi.org/10.1061/\(ASCE\)CF.1943-5509.0001279](https://doi.org/10.1061/(ASCE)CF.1943-5509.0001279).
- Zhao, Z., Dai, K., Camara, A., Bitsuamlak, G., Sheng, C., 2019a. Wind turbine tower failure modes under seismic and wind loads. *Journal of Performance of Constructed Facilities* 33(2). doi:[https://doi.org/10.1061/\(ASCE\)CF.1943-5509.0001279](https://doi.org/10.1061/(ASCE)CF.1943-5509.0001279).
- Zhao, Z., Dai, K., Lalonde, E., Meng, J., Li, B., Ding, Z., Bitsuamlak, G., 2019b. Studies on application of scissor-jack braced viscous damper system in wind turbines under seismic and wind loads. *Engineering Structures* 196. doi:<https://doi.org/10.1016/j.engstruct.2019.109294>.
- Zuo, H., Bi, K., Hao, H., 2017. Using multiple tuned mass dampers to control offshore wind turbine vibrations under multiple hazards. *Engineering Structures* 141(15), 303–315. doi:<https://doi.org/10.1016/j.engstruct.2017.03.006>.

Appendix A

CFD data set sample for NN development

The following tables present a sample of the recorded CFD data that is provided to the CNN for training and testing. The CFD simulations that generated this data, including modifications from the raw data to the processed data reported here, were presented in Section 7.3.5. This data was used to train the surrogate aerodynamic model, as discussed in Section 7.3.6. The full table was supplied in the first data configuration listed in Table 7.12, while the smaller, more effective configuration only supplied columns 49-64 as the inputs and output data. The full 64 columns are provided to the MATLAB code in Appendix B regardless of data configuration, the removal of columns is performed within the code.

After processing, one set of data was assembled for each blade at each base rotation and wind condition. With three blades, eight base rotations, and three wind conditions, 72 data sets were assembled consisting of 500 time steps each. The six tables presented here list a sample of the first 50 rows of the Blade 1 data from the 0.00 rad base rotation, 3.63 m/s wind speed, and 0.95 TSR time history. Blade 1, 2 and 3 are identified in Figure A.1. Each set of data is made of 64 columns of input/outputs, including 60 wind speed measurements. The nodes where these speeds were recorded are identified by their upwind distance (x) from the rotor plane and rotational distance (r) from the rotor axis, both of which are reported as a function of the blade length (R).

The six tables only report 0.14% of the total supplied to the NN. A .csv collection of this entire data set can be found at <https://github.com/elalond3/Lalonde-thesis-CNN-AHS.git>.

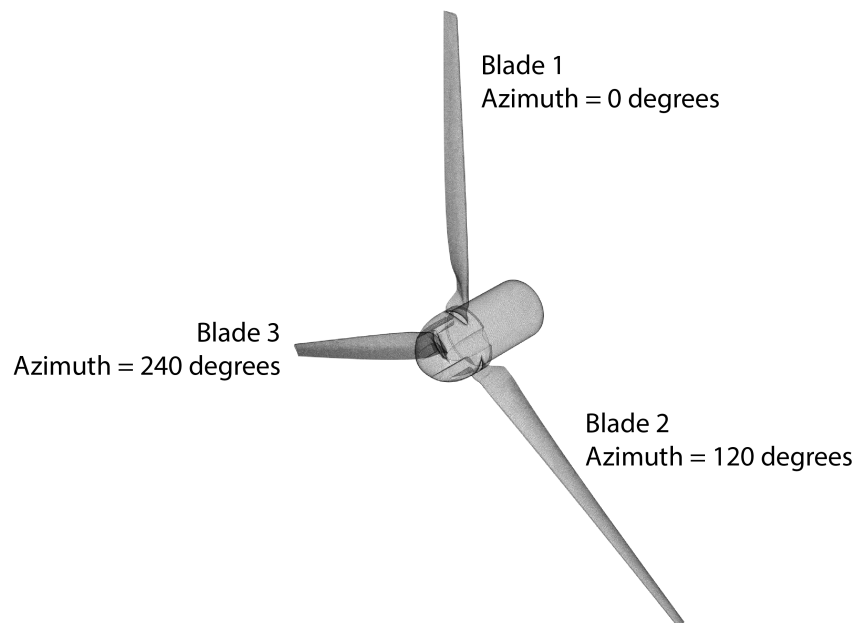


Figure A.1: Blade identification

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
U_i (m/s)	U_i (m/s)	U_i (m/s)	U_i (m/s)	U_i (m/s)	U_i (m/s)	U_i (m/s)	U_i (m/s)	U_i (m/s)	U_i (m/s)	U_i (m/s)	U_i (m/s)
$x=-0.2R$	$x=-0.2R$	$x=-0.2R$	$x=-0.2R$	$x=-0.2R$	$x=-0.2R$	$x=-0.2R$	$x=-0.2R$	$x=-0.2R$	$x=-0.2R$	$x=-0.2R$	$x=-0.2R$
$r=0.00$	$r=0.09R$	$r=0.18R$	$r=0.27R$	$r=0.36R$	$r=0.46R$	$r=0.55R$	$r=0.64R$	$r=0.73R$	$r=0.82R$	$r=0.91R$	$r=1.0R$
2.341	3.007	3.188	3.252	3.250	3.252	3.260	3.263	3.292	3.325	3.380	3.462
2.345	3.013	3.196	3.260	3.255	3.254	3.261	3.262	3.287	3.318	3.372	3.456
2.342	3.010	3.193	3.257	3.252	3.251	3.261	3.266	3.292	3.321	3.375	3.457
2.344	3.010	3.192	3.256	3.249	3.248	3.260	3.266	3.291	3.319	3.373	3.456
2.343	3.008	3.189	3.252	3.246	3.246	3.261	3.268	3.292	3.319	3.374	3.456
2.341	3.003	3.183	3.245	3.240	3.244	3.261	3.271	3.294	3.323	3.378	3.460
2.342	3.002	3.181	3.243	3.239	3.245	3.263	3.270	3.290	3.318	3.373	3.457
2.341	3.001	3.179	3.243	3.243	3.252	3.271	3.270	3.285	3.309	3.362	3.447
2.342	2.999	3.174	3.237	3.239	3.251	3.271	3.270	3.285	3.308	3.362	3.446
2.344	3.000	3.173	3.237	3.240	3.254	3.272	3.268	3.281	3.305	3.358	3.444
2.347	3.003	3.175	3.238	3.241	3.256	3.273	3.266	3.280	3.304	3.358	3.442
2.347	3.003	3.175	3.239	3.244	3.259	3.274	3.267	3.281	3.306	3.359	3.443
2.347	3.006	3.177	3.240	3.246	3.260	3.274	3.266	3.281	3.307	3.359	3.443
2.343	3.002	3.176	3.240	3.248	3.261	3.275	3.270	3.287	3.314	3.365	3.446
2.345	3.006	3.181	3.245	3.251	3.261	3.272	3.266	3.286	3.314	3.366	3.448
2.345	3.008	3.186	3.250	3.255	3.262	3.271	3.267	3.288	3.317	3.368	3.450
2.347	3.013	3.193	3.257	3.259	3.263	3.269	3.264	3.286	3.315	3.367	3.449
2.346	3.013	3.196	3.261	3.260	3.262	3.267	3.265	3.288	3.317	3.368	3.451
2.349	3.017	3.200	3.264	3.260	3.258	3.263	3.261	3.284	3.314	3.366	3.450
2.344	3.012	3.196	3.260	3.256	3.255	3.263	3.265	3.290	3.319	3.371	3.453
2.344	3.011	3.194	3.257	3.252	3.251	3.262	3.266	3.290	3.319	3.372	3.454
2.343	3.008	3.190	3.253	3.247	3.249	3.263	3.269	3.292	3.320	3.373	3.455
2.343	3.007	3.188	3.251	3.245	3.249	3.265	3.270	3.293	3.321	3.374	3.455
2.342	3.003	3.182	3.245	3.240	3.247	3.266	3.272	3.293	3.321	3.374	3.456
2.344	3.004	3.181	3.243	3.240	3.249	3.268	3.270	3.289	3.315	3.368	3.453
2.344	3.002	3.177	3.239	3.238	3.250	3.270	3.271	3.288	3.314	3.369	3.453
2.344	3.001	3.174	3.237	3.238	3.251	3.272	3.272	3.288	3.315	3.369	3.454
2.348	3.005	3.177	3.240	3.242	3.256	3.274	3.270	3.285	3.310	3.364	3.449
2.347	3.004	3.176	3.241	3.244	3.259	3.276	3.270	3.286	3.312	3.366	3.450
2.349	3.007	3.179	3.244	3.248	3.261	3.276	3.269	3.285	3.313	3.366	3.450
2.349	3.010	3.183	3.248	3.252	3.263	3.275	3.268	3.285	3.314	3.367	3.450
2.347	3.008	3.183	3.249	3.253	3.263	3.273	3.269	3.290	3.320	3.374	3.455
2.348	3.012	3.189	3.254	3.257	3.262	3.270	3.266	3.289	3.321	3.375	3.457
2.348	3.014	3.195	3.259	3.260	3.263	3.268	3.266	3.291	3.322	3.376	3.457
2.351	3.020	3.202	3.266	3.264	3.263	3.267	3.264	3.289	3.320	3.373	3.457
2.348	3.018	3.201	3.265	3.263	3.261	3.266	3.267	3.292	3.322	3.375	3.458
2.347	3.017	3.200	3.263	3.259	3.257	3.264	3.267	3.292	3.321	3.375	3.458
2.343	3.011	3.193	3.257	3.253	3.252	3.263	3.269	3.295	3.324	3.378	3.460
2.343	3.008	3.189	3.252	3.248	3.248	3.262	3.270	3.295	3.324	3.379	3.461
2.337	2.999	3.179	3.242	3.241	3.245	3.263	3.274	3.298	3.328	3.383	3.464
2.339	2.999	3.176	3.238	3.236	3.243	3.264	3.273	3.296	3.326	3.381	3.463
2.337	2.994	3.169	3.230	3.231	3.243	3.266	3.275	3.298	3.327	3.383	3.464
2.336	2.990	3.162	3.223	3.227	3.241	3.266	3.275	3.297	3.327	3.383	3.464
2.331	2.983	3.153	3.216	3.222	3.241	3.268	3.278	3.300	3.331	3.386	3.466
2.331	2.982	3.151	3.213	3.222	3.242	3.268	3.276	3.298	3.330	3.386	3.467
2.331	2.981	3.150	3.213	3.223	3.245	3.270	3.276	3.299	3.331	3.386	3.468
2.335	2.987	3.156	3.219	3.229	3.249	3.272	3.273	3.295	3.327	3.382	3.465
2.332	2.985	3.156	3.220	3.231	3.251	3.272	3.274	3.299	3.331	3.385	3.467
2.333	2.989	3.161	3.225	3.234	3.251	3.269	3.272	3.297	3.331	3.385	3.467
2.333	2.990	3.164	3.228	3.237	3.251	3.267	3.270	3.297	3.331	3.386	3.469

Table A.1: Columns 1-12 of processed CFD data sample extracted from Blade 1 in the 0.00 rad base rotation, 3.63 m/s mean wind speed, and 0.95 TSR conditions

C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23	C24
U_j (m/s)	U_j (m/s)	U_j (m/s)	U_j (m/s)	U_j (m/s)	U_j (m/s)	U_j (m/s)	U_j (m/s)	U_j (m/s)	U_j (m/s)	U_j (m/s)	U_j (m/s)
$x=-0.2R$	$x=-0.2R$	$x=-0.2R$	$x=-0.2R$	$x=-0.2R$	$x=-0.2R$	$x=-0.2R$	$x=-0.2R$	$x=-0.2R$	$x=-0.2R$	$x=-0.2R$	$x=-0.2R$
$r=0.00$	$r=0.09R$	$r=0.18R$	$r=0.27R$	$r=0.36R$	$r=0.46R$	$r=0.55R$	$r=0.64R$	$r=0.73R$	$r=0.82R$	$r=0.91R$	$r=1.0R$
0.008	0.298	0.229	0.138	0.107	0.103	0.108	0.125	0.140	0.159	0.182	0.191
0.009	0.303	0.235	0.146	0.117	0.115	0.123	0.138	0.149	0.165	0.187	0.195
0.006	0.300	0.231	0.138	0.108	0.106	0.116	0.131	0.143	0.161	0.185	0.193
0.002	0.299	0.231	0.139	0.110	0.109	0.119	0.133	0.143	0.161	0.185	0.194
-0.004	0.294	0.226	0.134	0.106	0.107	0.117	0.129	0.140	0.160	0.184	0.194
-0.009	0.289	0.220	0.128	0.099	0.100	0.108	0.119	0.131	0.155	0.181	0.193
-0.010	0.290	0.225	0.137	0.112	0.115	0.119	0.127	0.137	0.159	0.185	0.195
-0.009	0.295	0.233	0.152	0.135	0.140	0.144	0.146	0.154	0.171	0.194	0.201
-0.016	0.286	0.223	0.141	0.124	0.128	0.131	0.134	0.143	0.164	0.189	0.198
-0.017	0.286	0.225	0.147	0.131	0.133	0.133	0.134	0.143	0.164	0.188	0.196
-0.019	0.283	0.222	0.145	0.130	0.130	0.128	0.131	0.142	0.163	0.187	0.196
-0.016	0.283	0.221	0.143	0.126	0.123	0.120	0.124	0.137	0.159	0.184	0.193
-0.013	0.284	0.222	0.142	0.124	0.119	0.115	0.121	0.135	0.158	0.183	0.193
-0.008	0.285	0.219	0.135	0.112	0.104	0.100	0.108	0.125	0.150	0.177	0.190
-0.004	0.288	0.221	0.135	0.109	0.101	0.098	0.109	0.126	0.150	0.177	0.189
0.002	0.292	0.223	0.132	0.103	0.094	0.093	0.107	0.125	0.149	0.175	0.187
0.005	0.295	0.225	0.133	0.102	0.093	0.095	0.110	0.127	0.149	0.175	0.187
0.009	0.297	0.226	0.131	0.097	0.089	0.093	0.109	0.126	0.148	0.174	0.186
0.006	0.298	0.228	0.133	0.100	0.094	0.100	0.116	0.130	0.151	0.176	0.186
0.006	0.297	0.226	0.129	0.095	0.090	0.098	0.114	0.128	0.150	0.175	0.186
0.002	0.296	0.225	0.130	0.097	0.095	0.103	0.117	0.130	0.152	0.178	0.188
-0.003	0.293	0.223	0.128	0.098	0.097	0.106	0.118	0.130	0.152	0.178	0.189
-0.007	0.291	0.222	0.129	0.101	0.101	0.108	0.119	0.130	0.153	0.179	0.190
-0.014	0.286	0.217	0.127	0.102	0.103	0.109	0.117	0.129	0.152	0.179	0.190
-0.016	0.285	0.219	0.133	0.113	0.116	0.121	0.126	0.136	0.157	0.183	0.192
-0.020	0.281	0.216	0.132	0.114	0.116	0.119	0.122	0.133	0.156	0.182	0.192
-0.022	0.279	0.215	0.132	0.114	0.116	0.115	0.119	0.131	0.156	0.182	0.192
-0.021	0.279	0.217	0.138	0.123	0.125	0.124	0.127	0.140	0.162	0.188	0.196
-0.019	0.279	0.216	0.136	0.119	0.117	0.115	0.119	0.134	0.158	0.184	0.193
-0.015	0.281	0.218	0.137	0.118	0.115	0.111	0.117	0.133	0.157	0.183	0.193
-0.010	0.285	0.220	0.138	0.117	0.111	0.107	0.115	0.132	0.156	0.182	0.192
-0.004	0.288	0.220	0.133	0.106	0.096	0.092	0.102	0.121	0.148	0.175	0.187
0.001	0.291	0.222	0.133	0.104	0.095	0.093	0.106	0.125	0.149	0.176	0.188
0.007	0.296	0.225	0.132	0.100	0.090	0.091	0.107	0.125	0.148	0.175	0.187
0.008	0.298	0.228	0.135	0.103	0.095	0.099	0.114	0.130	0.152	0.178	0.188
0.009	0.299	0.228	0.132	0.098	0.091	0.097	0.113	0.128	0.150	0.176	0.188
0.007	0.299	0.228	0.132	0.098	0.093	0.101	0.116	0.130	0.151	0.178	0.189
0.005	0.298	0.226	0.129	0.095	0.091	0.099	0.114	0.127	0.149	0.177	0.188
-0.001	0.295	0.225	0.130	0.097	0.095	0.104	0.116	0.128	0.151	0.178	0.189
-0.004	0.291	0.221	0.124	0.092	0.091	0.099	0.110	0.123	0.147	0.175	0.187
-0.010	0.288	0.220	0.127	0.099	0.100	0.106	0.115	0.127	0.150	0.176	0.187
-0.014	0.284	0.216	0.126	0.101	0.101	0.106	0.113	0.125	0.149	0.175	0.186
-0.019	0.280	0.213	0.127	0.104	0.105	0.107	0.113	0.126	0.150	0.176	0.187
-0.020	0.276	0.210	0.124	0.101	0.100	0.101	0.107	0.121	0.146	0.173	0.185
-0.021	0.275	0.210	0.127	0.106	0.104	0.102	0.107	0.122	0.148	0.174	0.185
-0.020	0.275	0.210	0.127	0.106	0.103	0.100	0.106	0.122	0.147	0.173	0.184
-0.018	0.278	0.214	0.134	0.116	0.113	0.109	0.114	0.129	0.152	0.177	0.185
-0.013	0.279	0.214	0.132	0.111	0.105	0.101	0.109	0.126	0.150	0.175	0.184
-0.009	0.284	0.219	0.136	0.113	0.105	0.102	0.112	0.129	0.152	0.177	0.186
-0.006	0.286	0.220	0.135	0.110	0.102	0.102	0.112	0.129	0.152	0.176	0.184

Table A.2: Columns 13-24 of processed CFD data sample extracted from Blade 1 in the 0.00 rad base rotation, 3.63 m/s mean wind speed, and 0.95 TSR conditions

C25	C26	C27	C28	C29	C30	C31	C32	C33	C34	C35	C36
U_k (m/s)	U_k (m/s)	U_k (m/s)	U_k (m/s)	U_k (m/s)	U_k (m/s)	U_k (m/s)	U_k (m/s)	U_k (m/s)	U_k (m/s)	U_k (m/s)	U_k (m/s)
$x=-0.2R$	$x=-0.2R$	$x=-0.2R$	$x=-0.2R$	$x=-0.2R$	$x=-0.2R$	$x=-0.2R$	$x=-0.2R$	$x=-0.2R$	$x=-0.2R$	$x=-0.2R$	$x=-0.2R$
$r=0.00$	$r=0.09R$	$r=0.18R$	$r=0.27R$	$r=0.36R$	$r=0.46R$	$r=0.55R$	$r=0.64R$	$r=0.73R$	$r=0.82R$	$r=0.91R$	$r=1.0R$
0.000	0.046	0.011	0.026	0.029	0.040	0.041	0.055	0.044	0.044	0.032	0.025
-0.006	0.040	0.006	0.023	0.029	0.042	0.045	0.060	0.051	0.052	0.038	0.032
-0.008	0.040	0.007	0.025	0.031	0.043	0.043	0.055	0.044	0.047	0.035	0.029
-0.011	0.039	0.008	0.028	0.035	0.047	0.045	0.055	0.045	0.048	0.036	0.029
-0.010	0.042	0.012	0.033	0.039	0.049	0.044	0.053	0.043	0.047	0.036	0.028
-0.007	0.048	0.020	0.041	0.046	0.053	0.043	0.049	0.038	0.042	0.033	0.024
-0.013	0.045	0.019	0.043	0.050	0.056	0.046	0.053	0.043	0.047	0.036	0.028
-0.022	0.037	0.012	0.038	0.048	0.055	0.048	0.059	0.054	0.059	0.047	0.041
-0.009	0.049	0.024	0.046	0.051	0.054	0.043	0.053	0.047	0.053	0.041	0.035
-0.008	0.050	0.024	0.046	0.050	0.053	0.043	0.054	0.049	0.055	0.043	0.036
0.001	0.056	0.029	0.049	0.050	0.051	0.042	0.055	0.051	0.056	0.044	0.037
0.004	0.058	0.029	0.047	0.046	0.046	0.038	0.052	0.048	0.053	0.043	0.035
0.007	0.059	0.028	0.045	0.043	0.043	0.037	0.052	0.048	0.053	0.043	0.035
0.008	0.058	0.026	0.041	0.037	0.037	0.032	0.047	0.041	0.046	0.037	0.030
0.007	0.055	0.022	0.036	0.033	0.036	0.034	0.050	0.044	0.048	0.038	0.030
0.004	0.051	0.017	0.030	0.029	0.034	0.034	0.051	0.044	0.047	0.037	0.030
0.001	0.047	0.012	0.027	0.027	0.035	0.036	0.053	0.045	0.048	0.038	0.030
-0.003	0.043	0.008	0.024	0.025	0.036	0.037	0.053	0.044	0.047	0.037	0.030
-0.006	0.042	0.008	0.025	0.030	0.043	0.045	0.059	0.050	0.051	0.040	0.031
-0.008	0.041	0.008	0.027	0.032	0.045	0.044	0.056	0.045	0.047	0.037	0.028
-0.010	0.041	0.010	0.030	0.037	0.049	0.047	0.057	0.045	0.047	0.037	0.029
-0.011	0.042	0.013	0.035	0.042	0.053	0.048	0.056	0.045	0.047	0.036	0.029
-0.010	0.045	0.018	0.040	0.048	0.056	0.049	0.055	0.043	0.045	0.035	0.028
-0.008	0.049	0.023	0.046	0.053	0.059	0.049	0.054	0.043	0.045	0.034	0.027
-0.003	0.054	0.028	0.051	0.057	0.061	0.051	0.058	0.049	0.051	0.040	0.032
0.002	0.060	0.034	0.056	0.059	0.061	0.049	0.056	0.048	0.051	0.040	0.031
0.007	0.064	0.038	0.059	0.060	0.060	0.047	0.055	0.048	0.051	0.040	0.031
0.013	0.069	0.041	0.060	0.059	0.058	0.048	0.059	0.053	0.057	0.045	0.038
0.016	0.070	0.041	0.057	0.054	0.052	0.042	0.055	0.049	0.053	0.042	0.035
0.018	0.069	0.038	0.054	0.049	0.048	0.040	0.055	0.049	0.053	0.041	0.034
0.018	0.067	0.035	0.049	0.044	0.044	0.039	0.055	0.049	0.052	0.041	0.033
0.016	0.064	0.031	0.044	0.039	0.040	0.035	0.050	0.041	0.045	0.034	0.025
0.012	0.058	0.024	0.037	0.034	0.039	0.037	0.053	0.045	0.046	0.035	0.026
0.008	0.053	0.018	0.032	0.031	0.038	0.038	0.054	0.045	0.046	0.035	0.026
0.003	0.049	0.014	0.029	0.031	0.041	0.043	0.058	0.049	0.050	0.038	0.029
-0.001	0.045	0.011	0.027	0.030	0.041	0.042	0.056	0.045	0.047	0.037	0.028
-0.005	0.042	0.009	0.027	0.032	0.045	0.044	0.056	0.045	0.047	0.037	0.028
-0.008	0.042	0.010	0.030	0.036	0.048	0.045	0.055	0.044	0.046	0.036	0.027
-0.010	0.042	0.013	0.034	0.041	0.052	0.047	0.055	0.044	0.047	0.036	0.027
-0.009	0.044	0.016	0.038	0.045	0.052	0.044	0.051	0.040	0.043	0.033	0.024
-0.008	0.048	0.021	0.045	0.052	0.058	0.048	0.054	0.044	0.047	0.035	0.026
-0.004	0.052	0.027	0.051	0.056	0.059	0.047	0.052	0.042	0.045	0.034	0.025
0.000	0.058	0.033	0.056	0.060	0.061	0.047	0.052	0.043	0.046	0.034	0.025
0.005	0.061	0.037	0.058	0.060	0.058	0.043	0.049	0.039	0.042	0.031	0.023
0.009	0.065	0.039	0.059	0.060	0.057	0.043	0.051	0.042	0.044	0.032	0.024
0.013	0.067	0.040	0.058	0.057	0.054	0.041	0.050	0.042	0.044	0.032	0.024
0.014	0.066	0.037	0.055	0.054	0.053	0.044	0.055	0.048	0.050	0.037	0.028
0.014	0.064	0.034	0.049	0.046	0.046	0.039	0.052	0.044	0.046	0.035	0.027
0.012	0.061	0.029	0.043	0.041	0.043	0.039	0.053	0.045	0.047	0.035	0.026
0.010	0.057	0.024	0.038	0.036	0.041	0.039	0.054	0.046	0.048	0.036	0.026

Table A.3: Columns 25-36 of processed CFD data sample extracted from Blade 1 in the 0.00 rad base rotation, 3.63 m/s mean wind speed, and 0.95 TSR conditions

C37	C38	C39	C40	C41	C42	C43	C44	C45	C46	C47	C48
U_i (m/s)	U_i (m/s)	U_i (m/s)	U_i (m/s)	U_i (m/s)	U_i (m/s)	U_i (m/s)	U_i (m/s)	U_i (m/s)	U_i (m/s)	U_i (m/s)	U_i (m/s)
$x=-0.5R$	$x=-0.5R$	$x=-0.5R$	$x=-0.5R$	$x=-0.5R$	$x=-0.5R$	$x=-0.5R$	$x=-0.5R$	$x=-0.5R$	$x=-0.5R$	$x=-0.5R$	$x=-0.5R$
$r=0.00$	$r=0.09R$	$r=0.18R$	$r=0.27R$	$r=0.36R$	$r=0.46R$	$r=0.55R$	$r=0.64R$	$r=0.73R$	$r=0.82R$	$r=0.91R$	$r=1.0R$
3.433	3.435	3.450	3.455	3.472	3.482	3.493	3.503	3.514	3.528	3.543	3.559
3.433	3.435	3.451	3.457	3.474	3.483	3.494	3.503	3.514	3.528	3.543	3.559
3.436	3.437	3.451	3.456	3.472	3.481	3.493	3.504	3.518	3.532	3.548	3.562
3.437	3.440	3.457	3.462	3.478	3.486	3.495	3.503	3.513	3.527	3.543	3.558
3.441	3.445	3.461	3.467	3.481	3.489	3.497	3.503	3.513	3.526	3.540	3.556
3.439	3.443	3.460	3.466	3.482	3.490	3.498	3.504	3.514	3.526	3.541	3.556
3.449	3.453	3.471	3.477	3.489	3.494	3.500	3.504	3.510	3.522	3.536	3.552
3.441	3.445	3.462	3.468	3.485	3.492	3.500	3.505	3.514	3.525	3.538	3.554
3.445	3.448	3.465	3.471	3.485	3.493	3.500	3.506	3.514	3.525	3.539	3.555
3.442	3.446	3.462	3.468	3.484	3.492	3.499	3.506	3.514	3.525	3.539	3.554
3.437	3.439	3.455	3.461	3.479	3.489	3.498	3.507	3.517	3.528	3.542	3.557
3.439	3.441	3.458	3.463	3.480	3.489	3.498	3.505	3.514	3.526	3.540	3.555
3.438	3.441	3.459	3.466	3.484	3.494	3.501	3.504	3.509	3.519	3.532	3.547
3.440	3.441	3.458	3.464	3.482	3.491	3.499	3.504	3.509	3.519	3.532	3.547
3.444	3.445	3.462	3.467	3.483	3.492	3.499	3.503	3.507	3.518	3.530	3.546
3.451	3.452	3.468	3.473	3.486	3.493	3.499	3.503	3.508	3.518	3.530	3.546
3.447	3.449	3.464	3.469	3.484	3.492	3.499	3.503	3.508	3.517	3.530	3.546
3.448	3.451	3.465	3.470	3.484	3.492	3.498	3.503	3.508	3.518	3.531	3.546
3.433	3.435	3.450	3.456	3.477	3.488	3.498	3.505	3.513	3.522	3.535	3.549
3.437	3.439	3.455	3.461	3.479	3.489	3.499	3.506	3.513	3.524	3.536	3.551
3.433	3.436	3.452	3.460	3.479	3.490	3.500	3.507	3.514	3.524	3.537	3.551
3.438	3.442	3.459	3.466	3.482	3.491	3.499	3.505	3.512	3.522	3.534	3.549
3.435	3.438	3.456	3.464	3.482	3.491	3.500	3.506	3.513	3.523	3.535	3.550
3.448	3.452	3.469	3.476	3.488	3.494	3.500	3.504	3.510	3.522	3.534	3.549
3.438	3.442	3.459	3.466	3.483	3.491	3.499	3.506	3.514	3.525	3.537	3.552
3.442	3.445	3.462	3.468	3.483	3.491	3.500	3.506	3.514	3.526	3.538	3.553
3.440	3.443	3.459	3.466	3.482	3.491	3.500	3.507	3.515	3.526	3.539	3.554
3.441	3.444	3.461	3.467	3.482	3.490	3.499	3.506	3.514	3.526	3.539	3.553
3.439	3.442	3.458	3.465	3.481	3.489	3.499	3.506	3.515	3.526	3.539	3.554
3.447	3.449	3.466	3.472	3.486	3.492	3.499	3.505	3.512	3.524	3.537	3.552
3.445	3.447	3.463	3.469	3.484	3.491	3.499	3.505	3.513	3.524	3.537	3.552
3.443	3.445	3.461	3.468	3.484	3.491	3.499	3.505	3.513	3.525	3.538	3.553
3.452	3.455	3.471	3.476	3.489	3.495	3.500	3.505	3.511	3.522	3.535	3.550
3.446	3.449	3.464	3.470	3.486	3.493	3.499	3.504	3.511	3.521	3.534	3.549
3.447	3.450	3.465	3.471	3.487	3.493	3.500	3.504	3.511	3.522	3.535	3.550
3.448	3.451	3.466	3.472	3.487	3.493	3.500	3.504	3.511	3.522	3.535	3.550
3.436	3.439	3.455	3.461	3.480	3.489	3.498	3.506	3.515	3.526	3.540	3.555
3.439	3.442	3.458	3.465	3.482	3.491	3.500	3.507	3.516	3.528	3.542	3.557
3.437	3.440	3.457	3.464	3.483	3.492	3.501	3.508	3.517	3.528	3.542	3.557
3.448	3.452	3.469	3.476	3.489	3.495	3.502	3.507	3.514	3.527	3.540	3.556
3.442	3.446	3.463	3.470	3.486	3.493	3.501	3.507	3.515	3.526	3.540	3.555
3.444	3.447	3.463	3.469	3.485	3.492	3.500	3.506	3.514	3.526	3.540	3.555
3.436	3.439	3.455	3.461	3.480	3.490	3.500	3.508	3.518	3.530	3.543	3.559
3.438	3.440	3.455	3.461	3.478	3.488	3.499	3.508	3.517	3.530	3.543	3.558
3.422	3.424	3.439	3.446	3.470	3.484	3.497	3.509	3.521	3.533	3.547	3.561
3.430	3.432	3.447	3.453	3.473	3.485	3.497	3.507	3.518	3.531	3.545	3.560
3.425	3.427	3.442	3.448	3.469	3.482	3.495	3.506	3.518	3.530	3.544	3.559
3.426	3.427	3.441	3.447	3.467	3.481	3.494	3.505	3.517	3.530	3.544	3.559
3.413	3.413	3.428	3.435	3.460	3.477	3.493	3.506	3.520	3.532	3.546	3.561
3.416	3.417	3.432	3.439	3.462	3.478	3.493	3.507	3.520	3.533	3.547	3.562

Table A.4: Columns 37-48 of processed CFD data sample extracted from Blade 1 in the 0.00 rad base rotation, 3.63 m/s mean wind speed, and 0.95 TSR conditions

C49	C50	C51	C52	C53	C54	C55	C56	C57	C58	C59	C60
U_i (m/s)	U_i (m/s)	U_i (m/s)	U_i (m/s)	U_i (m/s)	U_i (m/s)	U_i (m/s)	U_i (m/s)	U_i (m/s)	U_i (m/s)	U_i (m/s)	U_i (m/s)
$x=-1.0R$	$x=-1.0R$	$x=-1.0R$	$x=-1.0R$	$x=-1.0R$	$x=-1.0R$	$x=-1.0R$	$x=-1.0R$	$x=-1.0R$	$x=-1.0R$	$x=-1.0R$	$x=-1.0R$
$r=0.00$	$r=0.09R$	$r=0.18R$	$r=0.27R$	$r=0.36R$	$r=0.46R$	$r=0.55R$	$r=0.64R$	$r=0.73R$	$r=0.82R$	$r=0.91R$	$r=1.0R$
3.555	3.555	3.557	3.560	3.563	3.568	3.572	3.576	3.580	3.585	3.590	3.593
3.559	3.559	3.561	3.563	3.565	3.568	3.572	3.577	3.580	3.585	3.589	3.593
3.551	3.551	3.554	3.556	3.559	3.565	3.570	3.576	3.581	3.586	3.590	3.594
3.558	3.558	3.560	3.562	3.564	3.568	3.572	3.576	3.580	3.584	3.589	3.593
3.553	3.553	3.555	3.557	3.560	3.565	3.570	3.575	3.578	3.583	3.588	3.592
3.554	3.554	3.556	3.559	3.561	3.565	3.570	3.575	3.578	3.583	3.588	3.592
3.557	3.557	3.558	3.561	3.563	3.567	3.571	3.575	3.579	3.584	3.588	3.592
3.555	3.555	3.557	3.560	3.562	3.567	3.571	3.575	3.579	3.584	3.588	3.592
3.561	3.562	3.563	3.565	3.567	3.570	3.572	3.575	3.578	3.583	3.587	3.591
3.559	3.559	3.561	3.563	3.565	3.568	3.571	3.575	3.578	3.583	3.587	3.591
3.560	3.560	3.562	3.564	3.566	3.569	3.572	3.576	3.578	3.583	3.587	3.590
3.559	3.560	3.562	3.564	3.566	3.570	3.573	3.576	3.579	3.583	3.587	3.591
3.561	3.562	3.563	3.565	3.567	3.570	3.573	3.577	3.580	3.584	3.587	3.591
3.561	3.562	3.564	3.566	3.568	3.571	3.573	3.576	3.579	3.583	3.587	3.591
3.564	3.565	3.566	3.568	3.570	3.572	3.574	3.577	3.579	3.582	3.587	3.590
3.561	3.562	3.564	3.567	3.569	3.572	3.574	3.577	3.579	3.583	3.587	3.590
3.569	3.570	3.571	3.573	3.575	3.576	3.576	3.577	3.578	3.582	3.585	3.588
3.561	3.562	3.564	3.566	3.568	3.572	3.574	3.577	3.579	3.583	3.586	3.590
3.566	3.566	3.568	3.570	3.572	3.574	3.577	3.579	3.581	3.584	3.588	3.591
3.563	3.563	3.565	3.567	3.569	3.572	3.575	3.578	3.580	3.583	3.587	3.590
3.559	3.559	3.561	3.564	3.566	3.571	3.574	3.578	3.581	3.584	3.588	3.591
3.561	3.561	3.563	3.565	3.567	3.571	3.574	3.577	3.580	3.583	3.587	3.590
3.560	3.560	3.562	3.565	3.566	3.570	3.573	3.576	3.578	3.581	3.585	3.588
3.562	3.562	3.564	3.566	3.567	3.571	3.573	3.576	3.578	3.581	3.585	3.588
3.564	3.565	3.566	3.568	3.569	3.571	3.573	3.576	3.578	3.581	3.585	3.588
3.570	3.571	3.572	3.573	3.574	3.575	3.575	3.577	3.578	3.581	3.584	3.587
3.566	3.565	3.567	3.568	3.569	3.572	3.573	3.575	3.577	3.580	3.584	3.587
3.567	3.566	3.568	3.569	3.570	3.572	3.574	3.576	3.577	3.581	3.584	3.587
3.554	3.553	3.555	3.558	3.560	3.565	3.570	3.575	3.578	3.582	3.586	3.589
3.560	3.560	3.561	3.564	3.565	3.569	3.573	3.577	3.580	3.583	3.587	3.590
3.556	3.556	3.558	3.561	3.564	3.568	3.572	3.576	3.579	3.583	3.587	3.590
3.558	3.558	3.561	3.563	3.565	3.569	3.572	3.575	3.578	3.582	3.586	3.589
3.555	3.557	3.559	3.562	3.564	3.568	3.572	3.575	3.578	3.582	3.586	3.589
3.568	3.569	3.571	3.573	3.574	3.575	3.575	3.576	3.578	3.581	3.584	3.588
3.559	3.559	3.562	3.564	3.566	3.570	3.573	3.576	3.578	3.582	3.586	3.590
3.563	3.564	3.566	3.568	3.569	3.572	3.575	3.577	3.580	3.583	3.587	3.590
3.561	3.562	3.564	3.566	3.568	3.571	3.574	3.577	3.580	3.583	3.587	3.590
3.561	3.561	3.563	3.565	3.567	3.570	3.572	3.576	3.578	3.582	3.586	3.590
3.559	3.560	3.562	3.564	3.566	3.570	3.573	3.576	3.578	3.582	3.586	3.590
3.565	3.566	3.568	3.570	3.571	3.573	3.575	3.576	3.578	3.582	3.586	3.589
3.563	3.563	3.565	3.567	3.569	3.572	3.574	3.576	3.578	3.582	3.586	3.589
3.561	3.562	3.564	3.566	3.568	3.572	3.574	3.577	3.579	3.582	3.586	3.589
3.568	3.569	3.570	3.573	3.574	3.575	3.576	3.577	3.578	3.581	3.585	3.588
3.562	3.562	3.564	3.566	3.568	3.571	3.573	3.575	3.577	3.580	3.584	3.587
3.563	3.563	3.565	3.567	3.569	3.572	3.574	3.576	3.577	3.581	3.584	3.588
3.564	3.563	3.565	3.567	3.569	3.571	3.574	3.576	3.578	3.581	3.584	3.588
3.555	3.555	3.557	3.560	3.562	3.567	3.572	3.576	3.579	3.583	3.587	3.591
3.559	3.560	3.562	3.564	3.566	3.571	3.574	3.578	3.581	3.585	3.588	3.592
3.556	3.558	3.560	3.563	3.565	3.570	3.574	3.578	3.581	3.585	3.588	3.592
3.565	3.566	3.568	3.570	3.572	3.574	3.576	3.578	3.580	3.583	3.587	3.590

Table A.5: Columns 49-60 of processed CFD data sample extracted from Blade 1 in the 0.00 rad base rotation, 3.63 m/s mean wind speed, and 0.95 TSR conditions

C61	C62	C63	C64
Rotor speed ω (rad/s)	Base rotation θ (rad)	Blade thrust T (N)	Blade torque Q (Nm)
3.14	0.00	1.332	0.0694
3.14	0.00	1.350	0.0709
3.14	0.00	1.354	0.0700
3.14	0.00	1.369	0.0697
3.14	0.00	1.384	0.0659
3.14	0.00	1.399	0.0618
3.14	0.00	1.388	0.0572
3.14	0.00	1.379	0.0526
3.14	0.00	1.361	0.0510
3.14	0.00	1.349	0.0469
3.14	0.00	1.331	0.0469
3.14	0.00	1.314	0.0472
3.14	0.00	1.318	0.0513
3.14	0.00	1.316	0.0532
3.14	0.00	1.327	0.0596
3.14	0.00	1.337	0.0639
3.14	0.00	1.344	0.0672
3.14	0.00	1.331	0.0672
3.14	0.00	1.355	0.0706
3.14	0.00	1.355	0.0702
3.14	0.00	1.377	0.0699
3.14	0.00	1.387	0.0671
3.14	0.00	1.407	0.0660
3.14	0.00	1.402	0.0594
3.14	0.00	1.393	0.0538
3.14	0.00	1.377	0.0497
3.14	0.00	1.349	0.0478
3.14	0.00	1.343	0.0459
3.14	0.00	1.327	0.0476
3.14	0.00	1.317	0.0488
3.14	0.00	1.326	0.0550
3.14	0.00	1.333	0.0596
3.14	0.00	1.324	0.0617
3.14	0.00	1.351	0.0664
3.14	0.00	1.334	0.0676
3.14	0.00	1.361	0.0701
3.14	0.00	1.368	0.0716
3.14	0.00	1.382	0.0711
3.14	0.00	1.387	0.0705
3.14	0.00	1.414	0.0650
3.14	0.00	1.413	0.0597
3.14	0.00	1.414	0.0573
3.14	0.00	1.379	0.0495
3.14	0.00	1.355	0.0454
3.14	0.00	1.333	0.0450
3.14	0.00	1.318	0.0463
3.14	0.00	1.311	0.0490
3.14	0.00	1.298	0.0517
3.14	0.00	1.318	0.0581
3.14	0.00	1.328	0.0625

Table A.6: Columns 61-64 of processed CFD data sample extracted from Blade 1 in the 0.00 rad base rotation, 3.63 m/s mean wind speed, and 0.95 TSR conditions

Appendix B

MATLAB code for CNN optimization and training

The following MATLAB code was used to train a surrogate aerodynamic model on CFD data as detailed in Section 7.3.6. The surrogate model is a convolutional neural network (CNN) that is optimized then full trained on the supplied data. This code requires MATLAB's Deep Learning Toolbox and was written using version R2019b. A copy of this .m file can be found at <https://github.com/elalond3/Lalonde-thesis-CNN-AHS.git>. This code was originally written to provide 62 inputs per time step to the NN, however was updated to take less using the *col_remove* value – this aspect of the code could be made more efficient to reduce computation time.

```
1 function surrogate_NN_cfd
2
3 %% %% SUMMARY =====
4 % This code optimizes and trains a CNN on CFD data. This trained network relates the incoming wind
5 % speed and rotor speed to predict the thrust and torque on a 1.1 m residential turbine blade. This
6 % surrogate model will be applied in artificial hybrid tests (run once per blade per time step)
7 % for quickly calculating aerodynamic blade loads based on the provided wind field during testing.
8 %
9 % This code is separated into the following sections:
10 % 1. Parameter input, which allows the users to specify all required parameters
11 % 2. Data importing, which imports the CFD data in tabular form
12 % 3. Data arrangement, which arranges the import data into the specific form required by
13 % trainNetwork and splits it into training, validation and testing sets
14 % 4. Data pre-processing, which normalizes and shuffles data, divides into inputs and outputs
15 % 5. Hyperparameter optimization, which performs two rounds of Bayesian optimization to
16 % optimize the network's hyperparameters, the bounds of which are defined in Section 1
17 % 6. Final training, which trains the optimized network on the full set of data
18 % 7. Network testing, which checks both the accuracy of the network compared to the testing data
19 % as well as quantifying the run time
20 % 8. Save results
21 %
22 % The CNN will relate 58 inputs per input time step to two outputs. These inputs are 33 along-wind
23 % wind speeds at 0.2R, 0.5R, and 1.0R upwind of the blade where R = 1.1 m = blade length; 11
```

```

24 % across-wind and 11 vertical wind speeds at 0.2R; the rotor speed, the blade number, and the base
25 % rotation. These 58 data points will be input per time step input into the CNN (the exact number
26 % of which are a hyperparameter to optimize). The two outputs are the root thrust (along-wind force
27 % (N)) and torque (moment about rotor axis (Nm)) for the given time step.
28 % Data will be input as [t, Ux_0.2R, Uy_0.2R, Uz_0.2R, Ux_0.5R, Ux_1.0R, w, B, theta, T, Q] where t
29 % is time, w is the rotor speed, T is the thrust, Q is the torque. This data represents the time
30 % history of a single blade during a single simulation, thus each CFD simulation yielded three such
31 % time histories per wind speed/rotor speed configuration. Wind speeds are in m/s, rotor speed in
32 % rad/s, blade number is 1, 2 or 3, theta is in rad, thrust in N and torque in N/m.
33
34 clear
35 clc
36 %#ok< *NOPRT>
37
38 % Save output of command window to a log file
39 diary 'filepath'
40
41 %% 1. PARAMETER INPUT =====
42
43 % Filepaths
44 filepath_data='filepath';
45 filepath_save='filepath';
46
47 % DATA CONFIGURATION: use 62 inputs or 14 inputs? Original code was designed to run with 62 inputs,
48 % but has been updated to allow for fewer. For 64 inputs col_remove=0; for 14 inputs col_remove=48
49 col_remove=48; % number of leftmost columns to remove from input
50
51 % Static hyperparameter specification (conservative estimates made)
52 n_bayopt_evals1=80; % Number of Bayesian optimization evaluations in first round
53 n_bayopt_evals2=40; % Number of Bayesian optimization evaluations in second round
54 n_epochs_bayopt=50; % Number of epochs during optimization
55 n_epochs_final=5000; % Number of epochs during final training
56 init_learn_rate=0.001; % Initial learning rate
57 momentum=0.98; % Momentum
58
59 % Optimizable hyperparameter specifications
60 n_hyppar=6; % Number of hyperparameters to investigate (must equal number of rows in
61 % hyppar_table). hyppar = hyperparameters
62 hyppar_names={'ConvolutionFilterSize'; 'NumberOfFilters'; ...
63 'NumberOfInputTimeSteps'; 'PoolingSize'; 'FirstFCLayerSize'; 'SecondFCLayerSize'};
64 % FC = fully-connected
65 hyppar_range_low= [ 1; 1; 11; 1; 10; 10];
66 hyppar_range_high=[ 10; 10; 30; 6; 1000; 1000]; % Initial high and low values for the respective
67 % hyperparameter, optimal value will be found between these
68 hyppar_countstyle={'integer'; 'integer'; 'integer'; 'integer'; 'integer'; 'integer'};
69 % Hyperparameters can count uniformly, by integer, and logarithmically. Correct choice depends
70 % on type of hyperparameter. All here are integers, but for example learning rate would be
71 % counted logarithmically while momentum would use the default
72 hyppar_secondround=[2; 4; 4; 2; 50; 50]; % Added/subtracted from optimal hyperparameter from round 1
73 % to determine ranges in round 2
74
75 hyppar_table=table(hyppar_names, hyppar_range_low, hyppar_range_high, ...
76 hyppar_countstyle, hyppar_secondround);
77 % Table collecting hyperparameter names and ranges. Each row corresponds to one hyperparameter

```

```

78 %% 2. DATA IMPORTING =====
79
80 % Data files
81 n_blades=3; % Number of turbine blades
82 n_wind=3; % Number of wind speed/TSR configurations [3.63/0.95, 5.15/1.08, 5.15/4.98]
83     wind_names={'363_095'; '515_108'; '515_498'}; % Wind names for file calling
84 n_angles=8; % Number of base rotation angles [-0.01, 0.00, 0.01, 0.05, 0.10, 0.15, 0.20, 0.25]
85     angle_names={'N001'; 'P000'; 'P001'; 'P005'; 'P010'; 'P015'; 'P020'; 'P025'}; % Angle names
86 n_data_files_total=n_blades*n_wind*n_angles;
87 data_file_names=cell(n_data_files_total,1); % Preallocate cell size to assign data file names
88
89 % Create file names
90 for i=1:n_blades
91     for j=1:n_wind
92         for k=1:n_angles
93             % The name of each data file
94             data_file_names{(n_angles*n_wind)*(i-1)+n_angles*(j-1)+k,1}=...
95                 ['cfdresult_',angle_names{k,1},'_',wind_names{j,1},'_blade',num2str(i),'.csv'];
96         end
97     end
98 end
99
100
101
102 % Import first data file
103 temp_data=readtable([filepath_data,data_file_names{1,1}]);
104 temp_data=table2array(temp_data); % Convert data imported as table into a matrix
105 [~,n_col]=size(temp_data); % Count the number of columns
106 n_col=n_col-1; % Reduce by 1 as the time column will be removed from the saved data
107
108 % Extract time information
109 t=temp_data(:,1); % Time history (s)
110 dt=t(2); % Time step (2)
111 n_time=length(t); % Number of time steps
112 T=t(end,1); % Largest time step (s)
113
114 % File for data storage (3D matrix: rows=time, columns=input, depth=test case)
115 raw_data=zeros(n_time,n_col,n_data_files_total);
116
117 % Assign first data set to raw data matrix
118 raw_data(:,:,1)=temp_data(:,2:end);
119
120 % Assign remaining data
121 for i=2:n_data_files_total
122     temp_data=readtable([filepath_data,data_file_names{i,1}]);
123     temp_data=table2array(temp_data);
124     raw_data(:,:,i)=temp_data(:,2:end);
125 end
126 clear temp_data
127
128 % Remove unwanted data based on col_remove
129 n_col=n_col-col_remove;
130 raw_data=raw_data(:,(col_remove+1):end,:);
131

```

```

132 %% 3. DATA ARRANGEMENT =====
133
134 % Normalize data to a possible range of [-1,1] and record the normalization factor (important). Data
135 % is normalized according to the training data but is applied to all data
136 norm_factors=zeros(n_col,1);
137 for i=n_col
138     maximum=max(raw_data(:,i,:),[],'all');
139     minimum=min(raw_data(:,i,:),[],'all');
140     raw_data(:,i,:)=raw_data(:,i,:)/(maximum-minimum);
141     norm_factors(i)=1/(maximum-minimum);
142 end
143
144
145 % Split data into Training (60%), Validation (20%), and Testing (20%) segments
146 % Since time step adjacency must be maintain, each TH will be split twice into 3 THs. This splitting
147 % is randomly done in one of six ways. Let Tr be training, V be validation, and Te be testing. For
148 % each TH, a random integer between 1 and 6 is chosen, and the TH is cut appropriately:
149 % If rand = 1, the TH is cut into Tr, V, Te
150 % If rand = 2, the TH is cut into Tr, Te, V
151 % If rand = 3, the TH is cut into V, Tr, Te
152 % If rand = 4, the TH is cut into V, Te, Tr
153 % If rand = 5, the TH is cut into Te, Tr, V
154 % If rand = 6, the TH is cut into Te, V, Tr
155 % The data histories (the stacking of the sheets in the 3D matrix) are then randomly shuffled
156
157 % Determine sizes of each split time history
158 n_test=floor(0.2*n_time); % Number of time steps in the testing segments
159 n_val=n_test; % Number of time steps in the validation segments
160 n_train=n_time-2*n_test; % Number of time steps in the training segments
161
162 % Preallocate matrix sizes
163 data_train=zeros(n_train,n_col,n_data_files_total);
164 data_val=zeros(n_val,n_col,n_data_files_total);
165 data_test=zeros(n_test,n_col,n_data_files_total);
166
167 % Split up full data samples into training, validation, testing
168 for i=1:n_data_files_total
169     rand_num=randi(6); % Generate random integer to decide how data is split
170
171     if rand_num==1
172         data_train(:, :, i)=raw_data(1:n_train, :, i); % Split the time histories into three sections
173         data_val(:, :, i)=raw_data(n_train+(1:n_val), :, i);
174         data_test(:, :, i)=raw_data(n_train+n_val+(1:n_test), :, i);
175
176     elseif rand_num==2
177         data_train(:, :, i)=raw_data(1:n_train, :, i);
178         data_test(:, :, i)=raw_data(n_train+(1:n_test), :, i);
179         data_val(:, :, i)=raw_data(n_train+n_test+(1:n_val), :, i);
180
181     elseif rand_num==3
182         data_val(:, :, i)=raw_data(1:n_val, :, i);
183         data_train(:, :, i)=raw_data(n_val+(1:n_train), :, i);
184         data_test(:, :, i)=raw_data(n_val+n_train+(1:n_test), :, i);
185

```



```

186     elseif rand_num==4
187         data_val(:,i)=raw_data(1:n_val, :,i);
188         data_test(:,i)=raw_data(n_val+(1:n_test), :,i);
189         data_train(:,i)=raw_data(n_val+n_test+(1:n_train), :,i);
190
191     elseif rand_num==5
192         data_test(:,i)=raw_data(1:n_test, :,i);
193         data_train(:,i)=raw_data(n_test+(1:n_train), :,i);
194         data_val(:,i)=raw_data(n_test+n_train+(1:n_val), :,i);
195
196     elseif rand_num==6
197         data_test(:,i)=raw_data(1:n_test, :,i);
198         data_val(:,i)=raw_data(n_test+(1:n_val), :,i);
199         data_train(:,i)=raw_data(n_test+n_val+(1:n_train), :,i);
200     end
201 end
202 clear raw_data
203
204
205 % Shuffle 3D matrices: 3rd dimension is randomized, done independently for each data group
206 rand_order=randperm(n_data_files_total); % Generates random arrangement of integers from 1 to
207                                     % n_data_files_total
208 data_train=data_train(:, :, rand_order); % Randomize the stacking of the matrices in the training data
209 rand_order=randperm(n_data_files_total);
210 data_val=data_val(:, :, rand_order);
211 rand_order=randperm(n_data_files_total);
212 data_test=data_test(:, :, rand_order);
213
214
215 % Prepare to split into input and output data. Last two values are outputs (T and Q), rest are
216 % inputs
217 n_inputs=n_col-2;
218 n_outputs=2;
219
220
221 % Define function to rearrange data into the form required by the image input layer in trainNetwork
222 % Can't be performed ahead of time due to window size being an optimizable hyperparameter, therefore
223 % will be run inside of optimization/training loops
224 function [input_train, output_train, input_val, output_val, input_test, output_test]=...
225         data_rearranger(data_train, data_val, data_test, window_size)
226
227 % Initialize matrices
228 % Input into CNN as a number of 3D images (x,y,channels). Fourth dimension is the number of
229 % images, which is equal to n_time-window_size (as can't test until enough timesteps have passed
230 % that a full window can be filled) per data sample
231 n_train_images=(n_train-window_size+1)*n_data_files_total;
232 n_val_images=(n_val-window_size+1)*n_data_files_total;
233 n_test_images=(n_test-window_size+1)*n_data_files_total;
234 input_train=zeros(1, n_inputs, window_size, n_train_images);
235 input_val=zeros(1, n_inputs, window_size, n_val_images);
236 input_test=zeros(1, n_inputs, window_size, n_test_images);
237
238
239

```

```

240 % Outputs are 2D matrix equal to number of images x number of outputs
241 output_train=zeros(n_train_images, n_outputs);
242 output_val=zeros(n_val_images, n_outputs);
243 output_test=zeros(n_test_images, n_outputs);
244
245
246 % Assign training data
247 for ii=1:n_data_files_total
248     for jj=window_size:n_train
249         % Rearrangement is complex, but in broad strokes use reshape command to change
250         % time X input matrix into 1 X input X time matrix
251         input_train(:,:,:(n_train-window_size+1)*(ii-1)+(jj+1-window_size))=...
252         reshape((data_train((jj-window_size+1):jj,1:n_inputs,ii')),1,n_inputs,[]);
253
254         output_train((n_train-window_size+1)*(ii-1)+(jj+1-window_size),:)=...
255         data_train(jj,(n_inputs+1):end,ii);
256     end
257 end
258
259 % Shuffle training data so that adjacent time steps aren't next to each other in fourth
260 % dimension, while making sure that inputs and outputs remain aligned
261 rand_order=randperm(n_train_images); % Generate random order
262 input_train=input_train(:,:,:,rand_order); % Shuffle along fourth dimension
263 output_train=output_train(rand_order,:); % Shuffle along first dimension
264
265
266 % Assign validation data
267 for ii=1:n_data_files_total
268     for jj=window_size:n_val
269         % Rearrangement is complex, but in broad strokes use reshape command to change
270         % time X input matrix into 1 X input X time matrix
271         input_val(:,:,:(n_val-window_size+1)*(ii-1)+(jj+1-window_size))=...
272         reshape((data_val((jj-window_size+1):jj,1:n_inputs,ii')),1,n_inputs,[]);
273
274         output_val((n_val-window_size+1)*(ii-1)+(jj+1-window_size),:)=...
275         data_val(jj,(n_inputs+1):end,ii);
276     end
277 end
278
279 % Shuffle validation data so that adjacent time steps aren't next to each other in fourth
280 % dimension, while making sure that inputs and outputs remain aligned
281 rand_order=randperm(n_val_images); % Generate random order
282 input_val=input_val(:,:,:,rand_order); % Shuffle along fourth dimension
283 output_val=output_val(rand_order,:); % Shuffle along first dimension
284
285
286
287
288
289
290
291
292
293

```

```

294 % Assign testing data
295 for ii=1:n_data_files_total
296     for jj=window_size:n_test
297         % Rearrangement is complex, but in broad strokes use reshape command to change
298         % time X input matrix into 1 X input X time matrix
299         input_test(:,:,:(n_test-window_size+1)*(ii-1)+(jj+1-window_size))=...
300             reshape((data_test((jj-window_size+1):jj,1:n_inputs,ii)),1,n_inputs,[]);
301
302         output_test((n_test-window_size+1)*(ii-1)+(jj+1-window_size),:)=...
303             data_test(jj,(n_inputs+1):end,ii);
304     end
305 end
306
307 % Shuffle testing data so that adjacent time steps aren't next to each other in fourth
308 % dimension, while making sure that inputs and outputs remain aligned
309 rand_order=randperm(n_test_images); % Generate random order
310 input_test=input_test(:,:,:,rand_order); % Shuffle along fourth dimension
311 output_test=output_test(rand_order,:); % Shuffle along first dimension
312 end
313
314
315
316 %% 4. HYPERPARAMETER OPTIMIZATION =====
317
318 % Hyperparameter optimization is performed using Bayesian optimization. Firstly a function that
319 % trains a reduced network using a given set of hyperparameter values and evaluates the training must
320 % be developed. Then the Bayesian optimization program is run to estimate optimal parameter.
321 % Finally, the parameter ranges are reduced by the predicted optimal hyperparameters and a second
322 % round of Bayesian optimization is performed, where the predicted optimal hyperparameters are
323 % carried forward into the final simulation.
324
325 % Initialize neural network outside of a function
326 net = feedforwardnet(10);
327 % Initialize tracker, which helps track the progress of the program
328 tracker=0;
329
330 % Function for small set of training during hyperparameter optimization
331 function hyppar_opt_RMSE = hyppar_opt(parameters)
332
333     % bayesopt function imports parameters as a table, convert it to a matrix
334     parameters=table2array(parameters);
335     cov_filt_size=parameters(1);
336     n_filt=parameters(2);
337     window_size=parameters(3);
338     pool_size=parameters(4);
339     fully_con_size1=parameters(5);
340     fully_con_size2=parameters(6);
341
342     % The following value is the maximum possible size of the pooling process (it pools the entire
343     % data set into a linear layer). If requested pool size exceeds this, must be reduced
344     max_pool_size=n_inputs+1-cov_filt_size;
345     if max_pool_size<pool_size
346         pool_size=max_pool_size;
347     end

```

```

348 % Arrange data based on window size using pre-defined program, testing data only needed for
349 % after the final training
350 [input_train, output_train, input_val, output_val, ~, ~]=...
351     data_rearranger(data_train, data_val, data_test, window_size);
352
353 % Increment tracker and print progress
354 tracker=tracker+1;
355 ['Hyperparameter optimization: iteration ', num2str(tracker), ...
356     ' of ', num2str(n_bayopt_evals1+n_bayopt_evals2), ' (', num2str(n_bayopt_evals1), ...
357     ' in first round)']
358
359 % Define the layers of the CNN
360 layers = [ ...
361     imageInputLayer([1, n_inputs, window_size])
362     convolution2dLayer([1, cov_filt_size], n_filt, 'NumChannels', window_size)
363     batchNormalizationLayer % Recommended by documentation
364     tanhLayer
365     averagePooling2dLayer([1, pool_size], 'Stride', [1, 1])
366     fullyConnectedLayer(fully_con_size1)
367     tanhLayer
368     fullyConnectedLayer(fully_con_size2)
369     tanhLayer
370     fullyConnectedLayer(n_outputs)
371     regressionLayer];
372
373 % analyzeNetwork(layers) % Uncommented command will generate figure of neural network, useful
374 % for trouble shooting
375
376
377 % Since optimization is not occurring on full data set, will only supply a limited section of the
378 % (randomized) training and validation data. One sixth of training and one third of testing data
379 % will be supplied
380 n_train_images_hypopt=floor((n_train-window_size+1)*n_data_files_total/6);
381 n_val_images_hypopt=floor((n_val-window_size+1)*n_data_files_total/3);
382
383 % Define NN options
384 options = trainingOptions('sgdm', ...
385     'MaxEpochs', n_epochs_bayopt, ...
386     'InitialLearnRate', init_learn_rate, ...
387     'Verbose', true, ...
388     'VerboseFrequency', 100, ...
389     'Momentum', momentum, ...
390     'Shuffle', 'never', ...
391     'LearnRateSchedule', 'piecewise', ...
392     'MiniBatchSize', 10, ...
393     'Plots', 'none', ...
394     'ValidationData', ...
395     {input_val(:, :, :, 1:n_val_images_hypopt), output_val(1:n_val_images_hypopt,:)}, ...
396     'ValidationPatience', 5);
397
398 % Train small net
399 net = trainNetwork(input_train(:, :, :, 1:n_train_images_hypopt), ...
400     output_train(1:n_train_images_hypopt,:), layers, options);
401

```

```

402 % Predict output of net
403 output_val_predicted = predict(net,input_val(:,:,1:n_val_images_hypopt));
404 output_val_true=output_val(1:n_val_images_hypopt,:);
405
406 % Calculate RMSE of net based on predicted versus true validation data outputs
407 hyppar_opt_RMSE=(output_val_true-output_val_predicted).^2;
408 hyppar_opt_RMSE=sqrt(sum(hyppar_opt_RMSE,'all')/numel(hyppar_opt_RMSE));
409 end
410
411
412 % Assign previously-declared variables to optimize in first round of hyperparameter
413 optimVars = [
414     optimizableVariable(char(hyppar_table{1,1}),[hyppar_table{1,2},hyppar_table{1,3}],...
415     'Type',char(hyppar_table{1,4}))
416     optimizableVariable(char(hyppar_table{2,1}),[hyppar_table{2,2},hyppar_table{2,3}],...
417     'Type',char(hyppar_table{2,4}))
418     optimizableVariable(char(hyppar_table{3,1}),[hyppar_table{3,2},hyppar_table{3,3}],...
419     'Type',char(hyppar_table{3,4}))
420     optimizableVariable(char(hyppar_table{4,1}),[hyppar_table{4,2},hyppar_table{4,3}],...
421     'Type',char(hyppar_table{4,4}))
422     optimizableVariable(char(hyppar_table{5,1}),[hyppar_table{5,2},hyppar_table{5,3}],...
423     'Type',char(hyppar_table{5,4}))
424     optimizableVariable(char(hyppar_table{6,1}),[hyppar_table{6,2},hyppar_table{6,3}],...
425     'Type',char(hyppar_table{6,4}));
426
427
428 % Run first round of hyperparameter optimization and track required time
429 tic
430 optimal_hyperparameters = bayesopt(@hyppar_opt,optimVars,'Verbose',1000,...
431     'MaxObjectiveEvaluations',n_bayopt_evals1,'UseParallel',false,'PlotFcn',[]);
432 toc
433
434 % Print out predicted hyperparameters
435 optimal_hyperparameters.XAtMinObjective
436
437
438 % Update hyperparameter ranges for second round of testing, the ranges are selected as the minimum/
439 % maximum of the base starting values and the optimal value from round 1 plus/minus the second
440 % round range
441 optimVars = [
442     optimizableVariable(char(hyppar_table{1,1}),...
443     [max([hyppar_table{1,2},optimal_hyperparameters.XAtMinObjective{1,1}-hyppar_table{1,5}]),...
444     min([hyppar_table{1,3},optimal_hyperparameters.XAtMinObjective{1,1}+hyppar_table{1,5}])],...
445     'Type',char(hyppar_table{1,4}))
446     optimizableVariable(char(hyppar_table{2,1}),...
447     [max([hyppar_table{2,2},optimal_hyperparameters.XAtMinObjective{1,2}-hyppar_table{2,5}]),...
448     min([hyppar_table{2,3},optimal_hyperparameters.XAtMinObjective{1,2}+hyppar_table{2,5}])],...
449     'Type',char(hyppar_table{2,4}))
450     optimizableVariable(char(hyppar_table{3,1}),...
451     [max([hyppar_table{3,2},optimal_hyperparameters.XAtMinObjective{1,3}-hyppar_table{3,5}]),...
452     min([hyppar_table{3,3},optimal_hyperparameters.XAtMinObjective{1,3}+hyppar_table{3,5}])],...
453     'Type',char(hyppar_table{3,4}))
454
455

```

```

456     optimizableVariable(char(hyppar_table{4,1}),...
457         [max([hyppar_table{4,2},optimal_hyperparameters.XAtMinObjective{1,4}-hyppar_table{4,5}],...
458             min([hyppar_table{4,3},optimal_hyperparameters.XAtMinObjective{1,4}+hyppar_table{4,5}]),...
459             'Type',char(hyppar_table{4,4}))
460     optimizableVariable(char(hyppar_table{5,1}),...
461         [max([hyppar_table{5,2},optimal_hyperparameters.XAtMinObjective{1,5}-hyppar_table{5,5}],...
462             min([hyppar_table{5,3},optimal_hyperparameters.XAtMinObjective{1,5}+hyppar_table{5,5}]),...
463             'Type',char(hyppar_table{5,4}))
464     optimizableVariable(char(hyppar_table{6,1}),...
465         [max([hyppar_table{6,2},optimal_hyperparameters.XAtMinObjective{1,6}-hyppar_table{6,5}],...
466             min([hyppar_table{6,3},optimal_hyperparameters.XAtMinObjective{1,6}+hyppar_table{6,5}]),...
467             'Type',char(hyppar_table{6,4}))];
468
469     % Run second round of hyperparameter optimization and track required time
470     tic
471     optimal_hyperparameters = bayesopt(@hyppar_opt,optimVars,'Verbose',1000,...
472         'MaxObjectiveEvaluations',n_bayopt_evals2,'UseParallel',false,'PlotFcn',[]);
473     toc
474
475     % Print out optimal hyperparameters
476     optimal_hyperparameters.XAtMinObjective
477
478
479
480     %% 5. FINAL TRAINING =====
481
482     % Assign optimized hyperparameters
483     parameters=table2array(optimal_hyperparameters.XAtMinObjective);
484     cov_filt_size=parameters(1);
485     n_filt=parameters(2);
486     window_size=parameters(3);
487     pool_size=parameters(4);
488     fully_con_size1=parameters(5);
489     fully_con_size2=parameters(6);
490
491     % Reduce pool size if required
492     max_pool_size=n_inputs+1-cov_filt_size;
493     if max_pool_size<pool_size
494         pool_size=max_pool_size;
495     end
496
497     % Define the layers of the CNN
498     layers = [ imageInputLayer([1,n_inputs>window_size])
499         convolution2dLayer([1,cov_filt_size],n_filt,'NumChannels',window_size)
500         batchNormalizationLayer % Recommended by documentation
501         tanhLayer
502         averagePooling2dLayer([1,pool_size],'Stride',[1,1])
503         fullyConnectedLayer(fully_con_size1)
504         tanhLayer
505         fullyConnectedLayer(fully_con_size2)
506         tanhLayer
507         fullyConnectedLayer(n_outputs)
508         regressionLayer];
509

```

```

510 % Arrange data based on window size using pre-defined program
511 [input_train, output_train, input_val, output_val, input_test, output_test]=...
512     data_rearranger(data_train, data_val, data_test, window_size);
513
514 % Define NN options
515 options = trainingOptions('sgdm', ...
516     'MaxEpochs', n_epochs_final, ...
517     'InitialLearnRate', init_learn_rate, ...
518     'Verbose', true, ...
519     'VerboseFrequency', 100, ...
520     'Momentum', momentum, ...
521     'Shuffle', 'never', ...
522     'LearnRateSchedule', 'piecewise', ...
523     'MiniBatchSize', 10, ...
524     'Plots', 'none', ...
525     'ValidationData', {input_val, output_val}, ...
526     'ValidationPatience', 5);
527
528 % Train the final network
529 net = trainNetwork(input_train, output_train, layers, options);
530
531
532
533
534 %% 6. NETWORK TESTING =====
535
536 % With network trained, the testing data must be fed into it and the predicted output compared to
537 % the true testing output data. The RMSE and NRMSE is calculated for both thrust and torque
538 % separately and as a whole
539
540 % Predict the output of the trained network when testing data offered as input:
541 output_test_predicted = predict(net, input_test);
542
543 % Preallocate RMSE matrix. Row 1 is RMSE, row 2 is NRMSE, columns are equal to n_outputs. Row 3 is
544 % the mean NRMSE across all outputs, so there is only one value in this row
545 RMSE_results=zeros(3,n_outputs);
546
547 % Calculate RMSE for each column of output data between predicted and true
548 for i=1:n_outputs
549     % Get RMSE
550     RMSE_calc_temp=(output_test(:,i)-output_test_predicted(:,i)).^2;
551     RMSE_results(1,i)=sqrt(sum(RMSE_calc_temp,'all')/numel(RMSE_calc_temp));
552
553     % Get NRMSE
554     RMSE_results(2,i)=RMSE_results(1,i)/...
555         (max(output_test(:,i),[],'all')-min(output_test(:,i),[],'all'));
556
557     % Un-normalize RMSE
558     RMSE_results(1,i)=RMSE_results(1,i)/norm_factors(n_col-n_outputs+i);
559 end
560
561 % Find mean NRMSE
562 RMSE_results(3,1)=mean(RMSE_results(2,:));
563

```

```
564 % Check testing speed. Program is run six times on single image inputs, first is discarded and other
565 % five are used to determine the average speed of a single step
566 for i=1:6
567     tic
568     temp = predict(net,input_test(:,:,i));
569     toc
570 end
571
572
573
574 %% 7. SAVE RESULTS =====
575
576 % Save the trained neural network with the normalization factors
577 save([filepath_save,'trained NN.mat'],'net','norm_factors');
578
579 % Optimal hyperparameter output is unnecessarily detailed, only need some specific information
580 optimal_hyperparameters_reduced={optimal_hyperparameters.MinObjective; ...
581     optimal_hyperparameters.XAtMinObjective; ...
582     optimal_hyperparameters.MinEstimatedObjective; ...
583     optimal_hyperparameters.XAtMinEstimatedObjective; ...
584     optimal_hyperparameters.NumObjectiveEvaluations; ...
585     optimal_hyperparameters.TotalElapsedTime; ...
586     optimal_hyperparameters.ObjectiveMinimumTrace; ...
587     optimal_hyperparameters.EstimatedObjectiveMinimumTrace};
588
589 % Save most of the results (input and output data will not be saved)
590 save([filepath_save,'full results.mat'],'net','optimal_hyperparameters_reduced','RMSE_results',...
591     'norm_factors');
592
593 'PROGRAM COMPLETE'
594
595 % End command window logging
596 diary off
597
598 end
```


Appendix C

MATLAB code for artificial hybrid simulations

The following MATLAB code was used to perform the artificial hybrid simulations. It requires the three recalculated aerodynamic load time histories generated using OpenFAST as described in Section 7.2.3 as well as the trained neural network developed in Section 7.3.6 using the code from Appendix B. The specific combination of aerodynamic model and wind speed/TSR is set manually, this code does not have the functionality to automatically perform all 12 simulations in a single run. Note that the original code was written to provide 62 inputs to network, but was updated to account for less using the *col_remove* value – this aspect of the code could be made more efficient to reduce computation time. These inputs cannot be easily transcribed here, but they are supplied at <https://github.com/elalond3/Lalonde-thesis-CNN-AHS.git>. The artificial hybrid simulation code requires no additional MATLAB toolboxes and was written using version R2019b. A copy of this .m file can likewise be found at the above link.

```
1 function Artificial_turbine_hybrid_sim
2
3 %% % Description
4 % This program will test the rotational residential wind turbine model derived from aeroelastic wind
5 % tunnel testing equipped with a semi-active variable-damping TMD to three sets of wind loads using
6 % four different aerodynamic models: 1. FAST TH of a fixed turbine; 2. FAST TH of a rotational
7 % turbine; 3. FAST TH of a rotational turbine with PTMD; 4. CFD-based neural network. The results
8 % from these four sims will be compared
9
10 % The three wind speed cases are from my wind tunnel testing (3.63 m/s mean, 6.7% turbulence, 0.95
11 % TSR, 3.14 rad/s rotor speed); Kamran's testing (5.15 m/s mean, 10% turbulence, 1.08 TSR, 5.06
12 % rad/s rotor speed); and the recommended operational conditions of the wind turbine (9.00 m/s mean,
13 % 10% turbulence, 5.20 TSR, 42.54 rad/s rotor speed)
14
15 % Turbine model is approximated as a 1 DOF inverted pendulum. The weight of the turbine is simulated
16 % as a lumped mass at the end of the tower and the hub height wind speed loads both the lumped mass
17 % rotor and the tower. The lumped mass is also shifted forward by a value of d, this eccentricity is
18 % due to the fact that the COM of the turbine isn't over the tower center.
```

```

19 % mr, Er, kr are the rotational mass, damping ratio, and stiffness of the system
20 % theta is base rotation
21 % h is tower height and d is effective mass overhang
22 %      m d
23 %      O---|
24 %          |
25 %          | h
26 %      <---|--->
27 %      (k,E) | theta
28 % -----O-----
29
30
31 clear
32 clc
33 close all
34
35 %% Initial specifications
36 windspeedcase=3; % (= 1,2,3) selects the wind speed and TSR
37 meanwindspeeds=[3.63, 5.15, 5.15];
38 tsrcases=[0.95, 1.08, 4.98];
39
40 aeromodelcase=4; % (= 1,2,3,4) selects how aerodynamic loads are determined. 1–3 are FAST THs while
41     % 4 is calculated within the time history analysis using the CFD-based NN
42
43 no_TMD_yn=0; % Trigger to run test with or without the additional STMD on the turbine. If no_TMD_yn
44     % is equal to 1, the test will be run WITHOUT the TMD. This is achieved easily (and slightly
45     % inefficiently) by reducing the damper restoring force to 0 every time step
46
47 % DATA CONFIGURATION: use 62 inputs or 14 inputs? Original code was designed to run with 62 inputs,
48 % but has been updated to allow for fewer. For 64 inputs col_remove=0; for 14 inputs col_remove=48
49 % This value must be the same as the one used in
50 col_remove=48; % number of leftmost columns to remove from input
51
52 dt=0.01; % (s) length of analysis time steps
53 T=180; % (s) total simulation time
54 t=0:dt:T; % (s) time step array
55 n=length(t); % number of time steps in analysis
56
57
58 %% Structural properties
59 % Base rotation properties
60 mrot=86.32; % (kgm2) rotational inertia of the system, derived from experimental free vibration tests
61 krot=1129; % (Nm/rad) rotational stiffness of system, derived from experimental free vibration tests
62 nfrot=sqrt(krot/mrot); % (rad/s) natural frequency of base rotation
63 Erot=0.020; % structural damping ratio, from free vibration tests
64 crot=2*mfrot*mrot*(Erot); % (Nms/rad) rotational damping coefficient
65 CRturb=4*mrot/(4*mrot+2*crot*dt+krot*dt^2); % Chen–Ricles factor
66
67 % Mass and lengths
68 d=0.0296; % (m) effective mass overhang
69 h=1.9; % (m) turbine tower height
70 bladelen=1.1; % (m) blade length
71 m=23.9; % (kg) lumped mass
72 mg=234.6; % (N) lumped weight

```

```

73 % Other turbine properties
74 avchord=0.5*(0.128+0.050); % (m) average chord length of airfoil defined by Maryam
75 towerarea=h*0.08; % (m2) area of tower cross-section to on coming wind with a 8cm diameter, used
76 % with tower coefficient to calculate tower loads
77 Ctower=1.2; % drag coefficient of the tower, based on basic cylinder
78 fthrust=0.872; % thrust factor, a reduction in CT compared to FAST predictions, derived from
79 % comparisons to Maryam's Cl and Cd curves for the entire blades
80 ftower1=0.953; % tower factor 1, a reduction in the wind speed on the tower used to calculate the
81 % tower drag loads, derived from a power law curve with exponent of 0.05, which in turn was
82 % derived from profiles acquired from the wind tunnel
83 ftower2=0.517; % tower factor 2; normalized height at which the center of the tower loads occur. In
84 % a uniform flow, this would be 0.5 but because wind speed is low close to ground, gets raised
85 % slightly. ftower2*h gives the moment arm of the tower drag loads
86 theta_static=d/(1.9-krot/mg); % (rad) the static deflection of the tower from mass eccentricity
87 rotorspeed=tsrcases(windspeedcase)*meanwindspeeds(windspeedcase)/bladelen; % (rad/s) rotor speed
88
89 % Wind properties
90 airden=1.225; % (kg/m3) density of air
91 airkinvic=1.48e-5; % (m2/s) kinematic viscosity of air
92 meanwind=meanwindspeeds(windspeedcase);
93 reynolds=meanwind*avchord/airkinvic; % Reynolds number of simulation. Should be around 10^4-10^5
94
95 % Passive moments
96 massmoment=-mg*d; % (Nm) passive moment stemming from mass eccentricity
97 towermoment=0.5*airden*towerarea*Ctower*(ftower1*meanwind)^2*h*ftower2; % (Nm) passive mean
    moment
98 % stemming from tower drag. We're ignoring the fluctuating wind loads on the tower in this case
99
100
101 %% Blade node specs (for aeromodelcase==4)
102 if aeromodelcase==4
103     n_blade_nodes=11; % There will always be a node at the hub, so this is nodes in addition to
104     % that, resulting in the spacing between nodes equally bladelen/n_blade_nodes
105     blade_node_positions=zeros(n_blade_nodes+1,6); % Pre-allocate; columns 1 and 2 are for Y, Z of
106     % blade 1, columns 3 and 4 are Y, Z of blade 2, etc. Extra row if for all zeros at top
107 end
108
109
110 %% TMD properties
111 % Optimal values determined using equations from Connor (2202)
112 massratioTMD=0.01; % mass ratio between turbine and TMD
113 mTMD=m*massratioTMD; % (kg) TMD mass
114 ETMD=sqrt((massratioTMD*(3-sqrt(0.5*massratioTMD)))/(8*(1+massratioTMD)*(1-0.5*massratioTMD)));
115 % Damping ratio
116 QTMD=1-1.2*massratioTMD; % frequency ratio
117 nftMD=nfrot*QTMD; % (rad/s) TMD natural frequency
118 ctMD=2*mTMD*nftMD*ETMD; % (Ns/m) TMD damping coefficient
119 ktMD=mTMD*nftMD^2; % (N/m) TMD stiffness
120 CRtMD=4*mTMD/(4*mTMD+2*ctMD*dt+ktMD*dt^2); % Chen-Ricles ratios for the TMD EoM
121
122
123
124
125

```

```

126 % Bouc–Wen parameters: **** SCALED DOWN 10000 TIMES (SEE TEXT) ****
127 x0BW=0; % (m) initial displacement
128 gammaBW=3.62/100000000; % (1/m2)
129 betaBW=3.62/100000000; % (1/m2)
130 ABW=154.60;
131 nBW=2;
132 alphasBW=35.84/10000; % (N/m)
133 alphabBW=83.04/10000; % (N/mV)
134 c0aBW=1.539/10000; % (Ns/m)
135 c0bBW=32.82/10000; % (Ns/mV)
136 c1aBW=99.47/10000; % (Ns/m)
137 c1bBW=403.86/10000; % (Ns/mV)
138 k0BW=5.86/10000; % (N/mV)
139 k1BW=0.016/10000; % (N/m)
140 etaBW=60; % (Hz)
141
142
143
144 %% Initialize matrices
145 % Main structure
146 theta=zeros(n,3); % collect theta time history. The first column is rotation in rad, the
147     % second is rotational velocity in rad/s and the third is rotational acceleration in rad/s2
148 theta(:,1)=theta_static; % assign theta_static as initial theta value
149
150 % TMD
151 disptmd=zeros(n,3); % collect theta time history. The first column is TMD displacement in m,
152     % the second is TMD velocity in m/s and the third is TMD acceleration in m/s2
153     % Note, need to convert theta into displacement then shift by theta_static to find displacement
154     % of TMD
155 restoretmd=zeros(n,1); % (N) the resotring force from the TMD, which is equal to the relative
156     % displacement between turb and TMD times the TMD stiffness, plus the relative velocity times
157     % the TMD damping
158 dampingforce=zeros(n,1); % (N) the force from the MR damper, roughly equal to ctmd*tmdvelocity
159 stiffnessforce=zeros(n,1); % (N)
160 vBWhigh=2.32; % (V) Using ground hook controller, the high power
161 vBWlow=2.32;%0.79; % (V)
162 BWdelay=10; % number of time steps between wanting to change vBW and the change arriving to the MR
    damper
163 restoredelay=20; % number of time steps between applying voltage and getting resulting restoring force
164
165 % Bouc–Wen parameters
166 yBW=zeros(n,2); % (m) internal displacement in MBW model, true overall displacement is disptmd
167 vBW=ones(n,1)*vBWhigh; % (V) applied voltage
168 uBW=zeros(n,2); % (V) internal voltage, second column is first derivative of uBW
169 zBW=zeros(n,2); % (m) evolutionary variable, second column is firs derivative of zBW
170
171 % Measuerment errors – random normally–distributed error that are added to measured displacements and
172     % force in STMD, simulates experimental error of a true hybrid simulation
173 measureerror_disp=(randn(n,1)*0.0005+1);
174 measureerror_force=(randn(n,1)*0.0005+1);
175
176
177
178

```

```

179 %% Pre-calculated aerodynamic load time history
180 % Provided aeromodelcase=1-3, import the related TH file. Loads are the final column in the file
181 filepath_aero='<filepath>';
182 filepath_BTS='<filepath>';
183 aeromodelstrings=['_fixed'; '_rotnt'; '_rotwt']; % naming convention for three pre-calc models;
184                                     % rotnt = rotation no TMD, rotwt = rotation with TMD
185
186 if aeromodelcase~=4 % When using a pre-calc model
187     windspeedstrings=['3_63'; '5_15'; '5_50']; % 5_50 refers 5.15 m/s case with high TSR
188
189     % Simply import pre-calculated rotor thrust
190     aeroloads=importdata([filepath_aero,'FAST_',...
191         windspeedstrings(windspeedcase,:),aeromodelstrings(aeromodelcase,:),'.mat']);
192
193 else % When using the trained surrogate model
194     windspeedstrings=['3_63'; '5_15'; '5_15'];
195
196     % Load .bts time history generated using OpenFAST, using MATLAB code provided from NREL, found
197     % at https://github.com/OpenFAST/matlab-toolbox/blob/main/Utilities/readfile\_BTS.m
198     % A following function is used to call wind speeds from the BTS file during running
199
200     [BTS_velocity, ~, BTS_y, BTS_z, ~, ~, ~, ~, ~, BTS_dt, ~, ~]=...
201         readfile_BTS([filepath_BTS,'windTH_',windspeedstrings(windspeedcase,:),'.bts']);
202     % BTS_velocity = 4D time history of wind speeds where 1st dimension is time steps, second
203     % dimension refers to i, j, k components, third dimension refers to lateral position y in field,
204     % and fourth dimension refers to vertical position z in field. BTS_y and BTS_z are arrays
205     % listing the y and z values of these dimensions
206
207     BTS_z=BTS_z-h; % Adjust recorded z nodes so that z=0 corresponds to hub height. The BTS file
208     % records ground height as z=0
209
210     dt_ratio=dt/BTS_dt; % Ratio of MATLAB time step to the BTS file, if the MATLAB time steps don't
211     % have an equivalent value in the BTS file, will throw an error. Since BTS_dt=0.001s here,
212     % this is easy to achieve
213
214     if (floor(dt_ratio)==dt_ratio) || (dt_ratio < 0) % Check to make sure test is compatible w/ .bts
215         error(['Time step incompatible with BTS file; adjust so that it is a multiple of ',...
216             BTS_dt, ' s'])
217     end
218
219
220     % Import pre-trained NN ('net') to use as surrogate aerodynamic model
221     filepath_save="<filepath>"; % net filepath
222
223     NNload=load([filepath_save,'trained NN.mat'],...
224         'net','norm_factors','optimal_hyperparameters_reduced'); % Load net saved from other code
225
226     % Extract values from import data set
227     net=NNload.net; % Trained neural network
228     norm_factors=NNload.norm_factors; % Normalization factors used on inputs before going into NN
229     optimal_hyperparameters_reduced=NNload.optimal_hyperparameters_reduced;
230     % Normalization factors used on inputs before going into NN
231
232

```

```

233     window_size=optimal_hyperparameters_reduced{2,1}{1,3}; % Find optimal hyperpar. of num inputs
234     ninputs=62; % col_remove adjustment happens further down
235     n_of_outputs=2; % Number of outputs is always 2
236
237     NN_inputs=zeros(1,window_size,ninputs); % Pre-allocate NN input matrix. Will rewrite every dt
238     blade_thrust=zeros(n,3); % Pre-allocate time histories of predicted thrust for each blade
239     blade_torque=zeros(n,3); % Same, for torque. TORque isn't applied in EoM so not as important
240
241     % Define wind speed reduction factors. These are rough adjustments made to try and account for
242     % the rotor influence on the recorded wind speeds of the CFD sims, derived from the mean
243     % recorded wind speed divided by the max mean of the given wind speed
244     if windspeedcase==1
245         adjusts=[0.891, 0.978, 1.0];
246     elseif windspeedcase==2
247         adjusts=[0.892, 0.976, 1.0];
248     else
249         adjusts=[0.751, 0.866, 0.918];
250     end
251 end
252
253
254 function [u_i,u_j,u_k]=BTS_wind_speed_extract(time,y_node,z_node,base_rotation)
255     % For the previously loaded BTS wind field time history, the three components of the wind will
256     % be extracted at time t at the position (X,Y). Note that x isn't relevant since when training
257     % the NN that this data will be fed into, time steps were adjusted such that wind is adjusted by
258     % the average to when it reaches the rotor plane
259
260     % This program works by finding the wind speed components at four coordinates that box in the
261     % desired point in the X,Y plane and averages them
262
263     % Note that this code assumes that the turbine blades will never escape the bounds of the wind
264     % field (reasonable since there is 0.4 m extra on each side), and thus hasn't been programmed
265     % with any way to account for if they do leave these bounds
266
267
268     current_time_step=round(time/BTS_dt); % Found out time step based on provided time in s
269
270     % Find where y_node is bounded within BTS_y as y_index_plus and y_index_minus.
271     % y_ratio_plus and minus are the proportional weights towards of each bound, i.e. if y_node is
272     % 75% of the way from minus to plus, then y_ratio_plus=0.75 while y_ratio_minus=0.25
273     [y_index]=min(abs(BTS_y-y_node));
274     if (BTS_y(y_index)-y_node)>=0
275         y_index_minus=y_index;
276         y_index_plus=y_index+1;
277     else
278         y_index_minus=y_index-1;
279         y_index_plus=y_index;
280     end
281     y_ratio_plus=y_index-y_index_minus; % Opposite since a small gap corresponds with a big ratio
282     y_ratio_minus=y_index_plus-y_index;
283
284
285
286

```

```

287 % Same for Z
288 [~,z_index]=min(abs(BTS_z-z_node));
289 if (BTS_z(z_index)-z_node)>=0
290     z_index_minus=z_index;
291     z_index_plus=z_index+1;
292 else
293     z_index_minus=z_index-1;
294     z_index_plus=z_index;
295 end
296 z_ratio_plus=z_index-z_index_minus;
297 z_ratio_minus=z_index_plus-z_index;
298
299
300 % Find average wind speed components. First find horizontal then vertical averages
301 u_collected=zeros(1,3);
302 for ii=1:3
303     u_plusplus=BTS_velocity(current_time_step,ii,y_index_plus,z_index_plus);
304     u_plusminus=BTS_velocity(current_time_step,ii,y_index_plus,z_index_minus);
305     u_minusplus=BTS_velocity(current_time_step,ii,y_index_minus,z_index_plus);
306     u_minusminus=BTS_velocity(current_time_step,ii,y_index_minus,z_index_minus);
307
308     u_collected(ii)=z_ratio_plus*(y_ratio_plus*u_plusplus+y_ratio_minus*u_minusplus)+...
309         z_ratio_minus*(y_ratio_plus*u_plusminus+y_ratio_minus*u_minusminus);
310 end
311
312 u_i=u_collected(1); % Split values to be output by function
313 u_j=u_collected(2);
314 u_k=u_collected(3);
315
316 % Adjust i and k to account for base rotation. BTS has i parallel to rotor axis only when base
317 % rotation = 0, otherwise, needs to be updated slightly. Since rotation is in the XZ plane (in
318 % BTS coordinates) j is unaffected
319 u_j=u_collected(2);
320 u_i=u_collected(1)*cos(base_rotation) - u_collected(3)*sin(base_rotation);
321 u_k=u_collected(1)*sin(base_rotation) + u_collected(3)*cos(base_rotation);
322 end
323
324
325
326
327 %% % Run TH simulation
328 % This section of the code runs the time history in a loop using the Chen-Ricles algorithm
329 % The Alogirthm in at the end of the Chapter 4 methodology explains Chen-Ricles clearly, and the
330 % end of Chapter 6 has a good summary of the structural model
331 for i=2:n
332
333     % Track sim progress, outputs to command line
334     if rem(i,1000)==0
335         ['Simulation step ',num2str(i),' out of ',num2str(n)] %#ok<NOPRT>
336     end
337
338
339
340

```

```

341 % Predict rotation and rotational velocity of time step using Chen–Ricles algorithm
342 theta(i,2)=theta(i-1,2)+CRturb*dt*theta(i-1,3);
343 theta(i,1)=theta(i-1,1)+dt*theta(i-1,2)+CRturb*dt^2*theta(i-1,3);
344
345 f_ang=cos(theta(i,1)); % Factor used to adjust horizontal loads based on base rotation
346 % (=1 in small angle assumption)
347
348
349 if aeromodelcase~=4
350     % If using pre-calculated loads, just import the value plus multiply by hub height to get
351     % the overturning moment
352     thrustmoment=aeroloads(i,1)*h; %Nm
353
354
355 else
356     % Use the trained surrogate model
357
358     % Note that since drag loading on the hub and nacelle is neglected by OpenFAST, it is chosen
359     % to do the same here
360
361     % Determine rotational position of blades based on time and rotor speed. Recall blade 1
362     % begins at 0 rad, blade 2 at 2pi/3 rad and blade 3 at 4pi/3 rad
363     rotation_amount=rotorspeed*i*dt; % Rotation in rads
364     blade_1_angle=rem(rotation_amount,2*pi);
365     blade_2_angle=rem(rotation_amount+(2*pi/3),2*pi);
366     blade_3_angle=rem(rotation_amount+(4*pi/3),2*pi);
367
368     % Determine the YZ coordinates of the blade nodes given the current rotation. Starts at the
369     % first node beyond the hub and continues up to point 12 (there's one hub node then
370     % n_blade_nodes blade nodes
371     for j=2:n_blade_nodes+1
372         local_radius=j*(bladelen/n_blade_nodes);
373         blade_node_positions(j,:)=[...
374             local_radius*cos(blade_1_angle), local_radius*sin(blade_1_angle),...
375             local_radius*cos(blade_2_angle), local_radius*sin(blade_2_angle),...
376             local_radius*cos(blade_3_angle), local_radius*sin(blade_3_angle)];
377     end
378
379     % Need to find thrust loads on each blade
380     for blade_num=1:3
381         % Generate input to be placed in NN, which is a 1 x window_size x 64 matrix. The third
382         % dimension refers to the i,j,k blade components, the i components of which are repeated
383         % for 34–44 and 45–55. 56 refers to the rotor speed and 57 to the base rotation. The
384         % second component has time steps ranging from t-9 to t. Based on col_remove the left
385         % most columns are removed. IF col_remove=48, there's just the one set of i values
386
387         if i<window_size % Supply uniform values prior to enough back data being available
388
389             for k=1:n_blade_nodes+1
390                 % Use previously defined program, to get u_i, u_j, u_k for the node
391                 y_coordinate=blade_node_positions(k,2*blade_num-1);
392                 z_coordinate=blade_node_positions(k,2*blade_num);
393                 [u_BTS1,u_BTS2,u_BTS3]=...
394                 BTS_wind_speed_extract(i*dt,y_coordinate,z_coordinate,theta(i,1));

```



```

395         % Place values in input matrix
396         NN_inputs(:,1,k)=u_BTS1*adjusts(1); % Adjust i component by first ring scale
397         NN_inputs(:,1,k+(n_blade_nodes+1))=u_BTS2;
398         NN_inputs(:,1,k+2*(n_blade_nodes+1))=u_BTS3;
399     end
400     NN_inputs(:,1,(3*(n_blade_nodes+1)+1):(4*(n_blade_nodes+1)))=...
401         NN_inputs(:,1,1:(n_blade_nodes+1))*adjusts(2)/adjusts(1);
402         % Adjust u_i data to scale to second ring
403     NN_inputs(:,1,(4*(n_blade_nodes+1)+1):(5*(n_blade_nodes+1)))=...
404         NN_inputs(:,1,1:(n_blade_nodes+1))*adjusts(3)/adjusts(2);
405         % Adjust u_i data to scale to third ring
406     NN_inputs(:,1,5*(n_blade_nodes+1)+1)=rotorspeed; % Provide rotor speed
407     NN_inputs(:,1,5*(n_blade_nodes+1)+2)=theta(i,1); % Provide base rotation
408
409     for j=2:window_size
410         NN_inputs(:,j,:)=NN_inputs(:,1,:); % For this specific case, take current time
411             % step and fill out as uniform for the whole analysis since we don't have
412             % enough previous time steps yet
413     end
414
415     else % used for vast majority of simulation
416         for j=1:window_size
417             for k=1:(n_blade_nodes+1)
418                 % Use predefined program, to get u_i, u_j, u_k for the node
419                 y_coordinate=blade_node_positions(k,2*blade_num-1);
420                 z_coordinate=blade_node_positions(k,2*blade_num);
421                 [u_BTS1,u_BTS2,u_BTS3]=BTS_wind_speed_extract(dt*(i-window_size+j),...
422                     y_coordinate,z_coordinate,theta(i,1));
423
424                 % Place values in input matrix
425                 NN_inputs(:,j,k)=u_BTS1*adjusts(1); % Adjust i component by first ring scale
426                 NN_inputs(:,j,k+(n_blade_nodes+1))=u_BTS2;
427                 NN_inputs(:,j,k+2*(n_blade_nodes+1))=u_BTS3;
428             end
429             NN_inputs(:,j,(3*(n_blade_nodes+1)+1):(4*(n_blade_nodes+1)))=...
430                 NN_inputs(:,j,1:(n_blade_nodes+1))*adjusts(2)/adjusts(1);
431                 % Adjust u_i data to scale to second ring
432             NN_inputs(:,j,(4*(n_blade_nodes+1)+1):(5*(n_blade_nodes+1)))=...
433                 NN_inputs(:,j,1:(n_blade_nodes+1))*adjusts(3)/adjusts(2);
434                 % Adjust u_i data to scale to third ring
435             NN_inputs(:,j,5*(n_blade_nodes+1)+1)=rotorspeed; % Provide rotor speed
436             NN_inputs(:,j,5*(n_blade_nodes+1)+2)=theta(i,1); % Provide base rotation
437         end
438     end
439
440
441     % Remove col_remove excess columns from left hadn side
442     NN_inputs=NN_inputs(:,:(col_remove+1):end);
443
444     % Normalize NN input data
445     for j=1:(64-col_remove)
446         NN_inputs(:,j)=NN_inputs(:,j)*norm_factors(j);
447     end
448

```

```

449     % Run neural network
450     NN_output=predict(net,NN_inputs);
451
452     % Unnormalize NN output data
453     for j=1:2
454         NN_output(j)=NN_output(j)/norm_factors(j+(64-col_remove));
455     end
456
457     % Record output thrust and torque for the given turbine blade
458     blade_thrust(i,blade_num)=NN_output(1);
459     blade_torque(i,blade_num)=NN_output(2);
460
461 end
462
463     % Combine predicted thrust on each blade to get the total blade load
464     thrustmoment=(sum(blade_thrust(i,1:3)))*h;
465 end
466 % END OF SURROGATE MODEL CODE, REST IS STMD TH SIM
467
468
469
470 % Predict TMD displacement and velocity using Chen Ricles algorithm
471 disptmd(i,2)=disptmd(i-1,2)+CRtmd*dt*disptmd(i-1,3);
472 disptmd(i,1)=disptmd(i-1,1)+dt*disptmd(i-1,2)+CRtmd*dt^2*disptmd(i-1,3);
473
474
475 % Restoring force = sum of stiffness and damping forces
476 % Calucate stiffness force from relative displacement between rotating tower and the
477 % TMD, assume sin(theta) is equal to horzional nacelle displacement
478 stiffnessforce(i,1)=ktmd*(h*sin(theta(i,1))-theta_static)-disptmd(i,1)*measureerror_disp(i,1);
479
480     % Check to see what voltage value was set 10 ms previously, delay models true experimental
481     % delay between voltage sending and being received by MR damper
482     if (theta(i,1)-theta_static)*theta(i,2)<0 && (n-i)>BWdelay
483         vBW(i+BWdelay,1)=vBWlow;
484     end
485
486     % Perform the complicated Modified Bouc–Wen calculations. Firstly do the voltage calcs,
487     % get rates using backwards difference method, perform calcs
488     xBW=h*sin(theta(i,1))-theta_static)-disptmd(i,1)*measureerror_disp(i,1);
489     xpBW=h*sin(theta(i,2))-disptmd(i,2)*measureerror_disp(i,1);
490
491     uBW(i,2)=-etaBW*(uBW(i-1,1)-vBW(i,1));
492     uBW(i,1)=uBW(i-1,1)+dt*uBW(i,2);
493
494     alphaBW=alphaaBW+alphabBW*uBW(i,1);
495     c0BW=c0aBW+c0bBW*uBW(i,1);
496     c1BW=c1aBW+c1bBW*uBW(i,1);
497
498     yBW(i,2)=(1/(c0BW+c1BW))*(alphaBW*zBW(i-1,1)+c0BW*xpBW+k0BW*(xBW-yBW(i-1,1)));
499     yBW(i,1)=yBW(i-1,1)+yBW(i,2)*dt;
500
501
502

```

```

503     zBW(i,2)=-gammaBW*abs(xpBW-yBW(i,2))*zBW(i-1,1)*(abs(zBW(i-1,1)))^(nBW-1)...
504         -betaBW*(xpBW-yBW(i,2))*(abs(zBW(i-1,1)))^nBW...
505         +ABW*(xpBW-yBW(i,2));
506     zBW(i,1)=zBW(i-1,1)+dt*zBW(i,2);
507
508     % MBW model ends up with a prediction of the damping force
509     dampingforce(i,1)=alphaBW*zBW(i,1)+c0BW*(xpBW-yBW(i,2))+k0BW*(xBW-yBW(i,1))+k1BW*(
        xBW-x0BW);
510
511     % Get total restoring force of the TMD
512     restoretmd(i,1)=dampingforce(i,1)+stiffnessforce(i,1);
513
514     % If value was set to not consider TMD, zero the rpredicted value, TMD will never have effect on
515     % the main turbine
516     if no_TMD_yn==1
517         restoretmd(i,1)=0; % Eliminate influence of the PTMD when no_TMD_yn==1
518     end
519
520
521     % Calculate resulting TMD accelration
522     disptmd(i,3)=(restoretmd(i,1))/mtmd;
523
524     % Calculate rotational acceleration using full EoM, including the delay in calculating TMD
525     % restoring force of 20 ms. Replicates experimental limitation. If time step of analysis is
526     % before 20 ms has occurred, the delay is ignored
527     if i>restoredelay
528         theta(i,3)=(massmoment*f_ang+towermoment*f_ang^2+thrustmoment-krot*theta(i,1)...
529             -crot*theta(i,2)-restoretmd(i-restoredelay,1)*h*measureerror_force(i,1)*f_ang)/mrot;
530     else
531         theta(i,3)=(massmoment*f_ang+towermoment*f_ang^2+thrustmoment-krot*theta(i,1)...
532             -crot*theta(i,2)-restoretmd(1,1)*h*measureerror_force(i,1)*f_ang)/mrot;
533     end
534 end
535
536
537 %% % Save results
538
539 filepath_save="filepath";
540 aeromodelstrings=['fixed';'rotn';'rotwt';'CFDNN']; % Identify if using the pre-calculated fixed,
541     % rotation without TMD, rotation with TMD; or using in-the-loop CFD-NN surrogate model
542 windspeedstrings=['3_63'; '5_15'; '5_50'];
543 tsrstrings=['0_95';'1_08';'4_98'];
544
545 if no_TMD_yn~1
546     save_file_name=[filepath_save,aeromodelstrings(aeromodelcase,:),',',...
547         windspeedstrings(windspeedcase,:),',',tsrstrings(windspeedcase,)];
548 else
549     save_file_name=[filepath_save,aeromodelstrings(aeromodelcase,:),',',...
550         windspeedstrings(windspeedcase,:),',',tsrstrings(windspeedcase,),'undamped'];
551 end
552
553 save(save_file_name);
554
555 end

```

Curriculum Vitae

Name: Eric Rowland Lalonde

Post-Secondary Education and Degrees: University of Western Ontario
London, Canada
2019 - 2021

Tongji University
Shanghai, China
2017 - 2019
Ph.D. in wind engineering

University of Western Ontario
London, Canada
2016 - 2017
M.Eng. in structural engineering

University of Western Ontario
London, Canada
2012 - 2016
B.E.Sc. in civil engineering

Honours and Awards: Ontario Graduate Scholarship (2020 and 2021)
Alan G. Davenport Memorial Scholarship (2020)
Tongji University Scholarship for International PHD Students (2017 - 2019)
Canadian Institute of Steel Construction Scholarship (2016)

Related Work Experience: Research Engineer
Boundary Layer Wind Tunnel Laboratory
London, Canada
2021 - Present

Teaching Assistant
University of Western Ontario
London, Canada
2019 - 2021

Journal Publications:

Lalonde, E., Vischschraper, B., Bitsuamlak, G., Dai, K., (2021). Comparison of neural network types and architectures for generating a surrogate aerodynamic wind turbine blade model. *Journal of Wind Engineering and Industrial Aerodynamics*, 216. doi:10.1016/j.jweia.2021.104696.

Lalonde, E., Dai, K., Bitsuamlak, G., Zhao, Z., (2020). Comparison of semi-active and passive tuned mass damper systems for vibration control of a wind turbine. *Wind and Structures*, 30(6), 663–678. doi:https://doi.org/10.12989/was.2020.30.6.663.

Lalonde, E., Dai, K., Lu, W., Bitsuamlak, G., (2019). Wind turbine testing methods and application of hybrid testing: A review. *Wind and Structures*, 29(3), 195–207. doi:https://doi.org/10.12989/was.2019.29.3.195.

Zhao, Z., Dai, K., Lalonde, E., Meng, J., Li, B., Ding, Z., Bitsuamlak, G., (2019). Studies on application of scissor-jack braced viscous damper system in wind turbines under seismic and wind loads. *Engineering Structures*, 196. doi:https://doi.org/10.1016/j.engstruct.2019.109294.

Additional research can be found at <https://www.researchgate.net/profile/Eric-Lalonde/>