



Genetic Algorithm Optimisation of Actively Deforming Vertical Axis Wind Turbine Blade Profiles

Juan-Philip Marx

A thesis submitted in partial fulfilment of the requirements for the degree of
Doctor of Philosophy

The University of Sheffield

Faculty of Engineering

Department of Mechanical Engineering

April 2025

This page is left intentionally blank.

Declaration

I, Juan-Philip Marx, confirm that the Thesis is my own work. I am aware of the University's Guidance on the Use of Unfair Means (www.sheffield.ac.uk/ssid/unfair-means). This work has not been previously been presented for an award at this, or any other, university.

Acknowledgements

I am indebted to my supervisors, Professor Lin Ma, Professor Derek Ingham and Professor Mohamed Pourkashanian for their invaluable guidance, wisdom, and encouragement. Your insightful feedback and dedication to my development as a researcher have been instrumental in shaping this work. I am truly fortunate to have had the opportunity to learn from you.

A heartfelt thanks to my colleagues in the Energy2050 group. Your camaraderie and thought-provoking discussions made the challenges of research more manageable and rewarding. I have learned a great deal from each of you, and I am grateful for the opportunity to be part of such an inspiring group of individuals.

I would like to express my deepest gratitude to my wife, Priya, for her unwavering love, patience, and support throughout this journey. Your encouragement and belief in me has been my constant source of strength, and I could not have completed this thesis without you by my side.

Finally, I would like to extend my appreciation to my parents Rudi and Raelene, for their sacrifices and love, providing me the opportunity to be where I am today.

Publications

Transient Genetic Algorithm Optimisation of a VAWT blade profile. - UK Fluids Conference 2022

Abstract

Some studies to optimise the turbine blade profile to mitigate or overcome the dynamic stall issues that plague Vertical Axis Wind Turbines (VAWTs) have shown promise, however, they do not consider how the new optimised dynamic turbine blade profile could be replicated in an actual turbine, and typically utilise prescribed uninformed deformation profiles for the turbine blade. This thesis uses a novel approach by creating a dynamically changing turbine blade profile that alters the blade profile's camber based on azimuthal position, but the profile is derived from using a genetic algorithm optimisation process. The core body of work is in developing a genetic algorithm to optimise the blade camber magnitude at specific positions within the turbine's rotation, significantly improving overall and instantaneous power generation. Three techniques were tested for predicting the torque of each candidate camber profile in the GA optimisation process: using XFOIL to quickly predict torque at the desired azimuthal positions, using a transient rotating VAWT CFD simulation to model real-time blade deformation, and using an Artificial Neural Network Surrogate to predict torque based on turbine tip speed ratio, windspeed and azimuthal angle. These three methods were integrated into a bespoke genetic algorithm optimiser, to find the optimal turbine blade profiles through its rotation. The thesis also explores variations in the genetic algorithm's optimisation parameters, such as the number of optimisation positions, bounds of camber adjustment, and variations in the aerofoil configurations. All three techniques integrated into the GA showed significantly strong improvements in turbine power generation and reduced blade loading. The XFOIL GA showed the lowest improvement in mean moment at 59.1%, the CFD-In-The-Loop provided an uplift of 150% and the ML optimiser had an impressive 164% improvement!

Contents

Declaration.....	ii
Acknowledgements.....	iii
Publications.....	iv
Abstract.....	v
Contents.....	vi
List of Tables	x
List of Figures	xi
Nomenclature	xviii
1 Introduction.....	1
1.1 Background of Renewable Energy	1
1.2 Types of Vertical Axis Wind Turbines	2
1.3 Aerodynamic Concepts	4
1.3.1 Conventional Aerofoils.....	4
1.3.2 Lift Generation	6
1.3.3 VAWT Aerodynamic Theory.....	9
1.4 Aims and Objectives.....	12
2 Literature Review	14
2.1 High Lift Devices.....	14
2.1.1 Leading Edge Devices.....	15
2.1.2 Trailing Edge Devices	18
2.2 Turbine Blade Modification	24
2.2.1 Flow Manipulation Techniques.....	24
2.2.2 Aerofoil Profile	27
2.3 Optimisation Techniques	32
2.3.1 Parametric Design Studies	32
2.3.2 Adjoint Optimisation.....	33
2.3.3 Evolutionary Algorithms	33
2.3.4 Machine Learning (ML)	34
2.4 Summary and Thesis Outline	35
3 Methodology	39

3.1	Computational Fluid Dynamics	39
3.1.1	Navier Stokes Equations	39
3.1.2	Boundary Conditions.....	43
3.1.3	Turbulence Modelling.....	44
3.2	CFD Setup and Validation	47
3.2.1	Geometry	47
3.2.2	Mesh Independence Study	51
3.2.3	Final Mesh Characteristics	53
3.2.4	Timestep Independence Study	56
3.3	Deforming VAWT Methodology	57
3.4	Genetic Algorithm Optimiser	62
3.4.1	Flow chart and logic flow	63
3.4.2	Tuning the Genetic Algorithm.....	66
4	NACA Aerofoil Camber Optimisation of a VAWT Blade Deformation Profile at Predefined Angles of Attack using XFOIL.....	67
4.1	Introduction	67
4.1.1	Xfoil Specific Genetic Algorithm Implementation	68
4.1.2	XFOIL Genetic Algorithm Workflow	68
4.1.3	Genetic Algorithm Modifications in XFOIL.....	71
4.2	XFOIL-Based Genetic Algorithm Results	78
4.2.1	Optimisation Limits Investigation (Setting Angle = 6°)	78
4.2.2	Setting Angle Comparisons	84
4.2.3	0° Blade Setting Angle.....	85
4.2.4	Convergence Conditions	88
4.2.5	Sampling Count Analysis	91
4.2.6	Reynolds Number Sensitivity	92
4.2.7	Tip Speed Ratio Sensitivity	95
4.2.8	Alternative Angle of Attack Formulations	97
4.3	CFD Analysis of the Genetic Algorithm Optimised Deformation Profiles.....	106
4.3.1	Genetic Algorithm Evaluation in CFD.....	106
4.4	Summary of Findings.....	113
5	Transient Genetic Algorithm Optimisation of a VAWT Blade Profile	115

5.1	Introduction	115
5.2	CFD-In-the-Loop Transient Optimisation Setup	116
5.2.1	Genetic Algorithm Process Flowchart.....	117
5.2.2	Fitness Function	118
5.2.3	Convergence Criteria	119
5.3	PyFluent Functions.....	122
5.3.2	Implementation in cluster-based computer resources	124
5.4	Results & Discussion	125
5.4.1	Genetic Algorithm Parameters Investigation	126
5.4.2	Aerofoil Variations	135
5.4.3	Half Blade Deflection Model	141
5.4.4	Optimisation of VAWT blade camber profile at TSR = 1.5 & TSR = 3	146
5.5	Summary of Findings.....	151
6	Artificial Neural Network Surrogate for the Optimisation of the Aerofoil Camber Deformation Profile	152
6.1	Introduction	152
6.2	Machine Learning Model	154
6.2.1	Dataset generation method.....	155
6.2.2	Two-dimensional static airfoil simulation for Machine Learning Model Generation.....	158
6.2.3	CFD Validation and Verification	161
6.3	Machine Learning Model for the Blade Torque Prediction	169
6.3.1	Existing Regression Models	169
6.3.2	MLVAWT Artificial Neural Network	170
6.4	MLVawt ANN Specification and Validation.....	176
6.4.1	Sample Distribution Comparison	182
6.5	Genetic Algorithm Optimisation of the Camber Profile for VAWT Blade.....	185
6.5.1	Testing with Alternative Turbine Conditions	199
6.6	Summary of Findings.....	203
7	Thesis Conclusions and Future Work	205
7.1	Conclusion.....	205
7.2	Future Work.....	207

8	Appendix A.....	208
9	Appendix B – AI Regression Based Optimised VAWT Camber Profile Velocity Field Images 213	
10	Appendix C – Parametric Deforming VAWT Blade UDF.....	217
11	References	226

List of Tables

Table 3-1: Domain sizing comparisons	48
Table 3-2: O-Grid Sizing Comparison for Mesh Independence Study	51
Table 4-1: Flow Conditions for different XFOIL setup test scenarios	73
Table 4-2: Lift Coefficient comparison between XFOIL and published literature [102]	76
Table 4-3: Turbine Parameters	77
Table 4-4: Tangential Force Coefficient Comparison for different Genetic Algorithm Bounds with a 6° Blade Setting Angle	83
Table 4-5: Mean Tangential Force Coefficient for 0 degree setting angle comparison for ±5%c Optimisation Bounds and ±10%c Optimisation Bounds profiles	87
<i>Table 4-6: Moment Force Comparison for multiple Genetic Algorithm setups.....</i>	<i>108</i>
Table 4-7: Statistical Analysis of the Turbine's Moment Comparisons	111
Table 5-1: Example of the Camber Values for the First Generation of Genetic Algorithm Optimisation	118
Table 6-1: Bounds of the Input Variables for the Training Dataset.....	173
Table 6-2: TensorFlow Machine Learning Model Configurations	175
Table 6-3: VAWT Operating Conditions and VAWT Blade Parameters Tested.	186
Table 6-4: Polynomial Curve Fitting Statistics	194

List of Figures

Figure 1-1: (a) Schematic of the Savonius Rotor Patent [5] (b) Schematic of the Darrieus Rotor Patent [7].....	3
Figure 1-2: Cambered NACA 4 Series Aerofoil Schematic [12]	5
Figure 1-3: Typical Streamlines for Aerofoils at $\alpha = 0$ [14]. (a) Streamlines of Flow around Symmetrical Aerofoil. (b) Streamlines of Flow around Cambered Aerofoil. .	8
Figure 1-4: Effects of Angle of Attack on Lift Coefficient for a Cambered Aerofoil [15].	9
Figure 1-5: Relative flow velocity triangles for a rotating aerofoil and constant inflow velocity	10
Figure 1-6: Angle of attack comparison for multiple tip speed ratios	11
Figure 1-7: Reynolds Number Comparison for multiple tip speed ratios with a wind speed of 7 m/s and a turbine radius of 0.85m	12
Figure 2-1: Chougule et al. Verification of the DMSTM model [3].	17
Figure 2-2: Scungio et al. Auxiliary Blade Layout Compared with Standard Darrieus Turbine [4].	17
Figure 2-3: Angle of Attack as a function of the Azimuth Angle, Yang et al. [5].	21
Figure 2-4 Pressure Coefficient as a function of TSR for 2% c Flap [44]	23
Figure 2-5: Pressure Coefficient as a function of TSR for 3% c Flap [44]	23
Figure 2-6: - Suction Slots Locations [46]	25
Figure 2-7: Suction Slots Performance [46]	25
Figure 2-8: Depiction of flow jet domain for a) NACA0015, b) generic CFJ-NACA0015 and c) CFJ-NACA0015 for VAWT application [47].	25
Figure 2-9: Leading Edge Radius Power Coefficient [49].	28
Figure 2-10: CAD Model of an Aerofoil Assembly [57]	31

Figure 2-11: CAD Model of an Aerofoil Assembly [57]	32
Figure 3-1: CFD Geometry	49
Figure 3-2: Instantaneous Moment Comparison at Multiple Mesh Sizes	52
Figure 3-3: Mean Moment averaged over period of rotation	52
Figure 3-4: Entire Flow Domain Mesh	53
Figure 3-5: Near Blade Mesh.....	54
Figure 3-6: Near Blade Mesh at Sharp Trailing Edge	54
Figure 3-7: Boundary Conditions Highlighted on Flow Domain Edges	55
Figure 3-8: Single Blade Torque Coefficient Comparison with Li et al.[77] and El Sakka et al.[76] works	55
Figure 3-9: Comparison of Blade Torque for 1°, 0.5° and 0.25° of Rotation per Timestep.....	57
Figure 3-10: Visual representation of Deflection Model	59
Figure 3-11: Example Logic Flowchart for the Genetic Algorithm	64
Figure 3-12: Example of offspring genotype generation	65
Figure 4-1: Comparison of NACA5315 Aerodynamic Coefficients for Different XFOIL Setup Methods	75
Figure 4-2: Tangential Force Coefficient for 6 degree setting angle with NACA0015, ±5%c Optimisation Bounds and ±10%c Optimisation Bounds profiles	80
Figure 4-3: Aerofoil maximum camber value for 6 degree setting angle with ±5%c Optimisation Bounds and ±10%c Optimisation Bounds profiles	81
Figure 4-4: Angle of attack Comparison for a 0° and 6° setting angle of the turbine blade.	84
Figure 4-5: Tangential Force Coefficient Comparison for NACA0015, ±5%c Optimisation Bounds and ±10%c Optimisation Bounds profiles	86

Figure 4-6: Camber magnitude at the optimised positions for different optimisation bounds of $\pm 5\%c$ and $\pm 10\%c$	87
Figure 4-7: Variation in optimised tangential force coefficient for different convergence conditionsp	89
Figure 4-8: Reduced scale view of variation in optimised Tangential Force Coefficient for different convergence conditions.....	89
Figure 4-9: Optimisation Convergence Plot for 1, 10 and 25 Convergence Generations.	90
Figure 4-10: Comparison for number of Sampling points between every 36° and every 18° of azimuthal rotation	92
Figure 4-11: Impact of Reynolds Number on Tangential Force Coefficient	95
Figure 4-12: Angle of Attack and Reynolds Number Variation with respect to Azimuthal Angle for TSRs of 1.5, 2.29 and 3.....	95
Figure 4-13: Comparison of Tip Speed Ratio on the Performance of the Optimised VAWT Camber profile. attack formulations	96
Figure 4-14: Comparison of Camber Profile of the VAWT Optimisation process at multiple tip speed ratios	97
Figure 4-15: Velocity Contour of NACA0015 Blade Profile at 150° Azimuthal Angle. 98	
Figure 4-16: Velocity Contour of NACA0015 Blade Profile at 160° Azimuthal Angle. 99	
Figure 4-17: Velocity Contour of NACA0015 Blade Profile at 170° Azimuthal Angle. 99	
Figure 4-18: Velocity Contour of NACA0015 Blade Profile at 180° Azimuthal Angle 100	
Figure 4-19: Turbine Velocity Plot at the Centre Line ($\phi = 0^\circ$ to 180°) extracted from the Baseline NACA0015 VAWT CFD Model in ANSYS Fluent.....	101
Figure 4-20: Angle of Attack ($^\circ$) Comparison for different quantification methods. 103	
Figure 4-21: Optimised camber profile for the three variations of the angle of attack.	104

Figure 4-22: Tangential force coefficient comparison between the optimised blade and the NACA0015 blade profile for the Induced angle of attack definition	105
Figure 4-23: Tangential force coefficient comparison between the optimised blade and the NACA0015 blade profile for the Elsakka et al. angle of attack definition. ..	105
Figure 4-24: Moment comparison for the different Angle of Attack models from a transient turbine CFD simulation	109
Figure 4-25: Velocity Contours from 0->50m/s at $\phi = 0^\circ$ for (a) NACA0015 Non-Deforming & (b) XFOIL Normalised GA Deformation Profile Turbine.	110
Figure 4-26: Violin Plots representing the Distribution in Torque for the different Variations of the XFOIL Based Genetic Algorithm	113
Figure 5-1: Transferred variables between Genetic Algorithm Simulation Control steps	122
Figure 5-2: Impact of a randomised crossover magnitude compared to a constant crossover magnitude	127
Figure 5-3: Camber profile comparison for randomised and constant crossover techniques.	128
Figure 5-4: Moment contribution comparison for eleven phenotype method and five phenotype method.	131
Figure 5-5: Genotype count comparison impact on camber profile	132
Figure 5-6: Effect of Sampling Count on Optimised Turbine Profile	134
Figure 5-7: Impact of Sampling Count on Optimised Turbine Camber Profile	135
Figure 5-8: Deflection Centre Influence on Torque Profile	137
Figure 5-9: Position of maximum camber effect on the turbine camber profile	138
Figure 5-10: Velocity contour for aerofoil optimised at 72 degrees with a 20% maximum camber	139
Figure 5-11: Velocity contour for aerofoil optimised at 72 degrees with a 50% max camber	140

Figure 5-12: Velocity contour for aerofoil optimised at 108 degrees with a 20%c max camber	141
Figure 5-13: Velocity contour for aerofoil optimised at 108 degrees with a 50%c max camber	141
Figure 5-14: Moment Distribution for a Standard NACA Aerofoil Shape, Leading-edge only Deformation, Trailing-edge only Deformation and combined NACA _x 515 deformation model.....	144
Figure 5-15: Camber deflection magnitude profile for the standard NACA _x 515 Profile, Leading Edge only Deformation and Trailing Edge only Deformation.	145
Figure 5-16: Average Turbine Moment over one period of rotation	146
Figure 5-17: Comparison between the Transient Genetic Algorithm Optimised Camber Profile and Symmetric NACA0015 Turbine Blade Profile at TSR=1.5	148
Figure 5-18: Comparison between the Transient Genetic Algorithm Optimised Camber Profile and Symmetric NACA0015 Turbine Blade Profile at TSR=3.....	149
Figure 6-1: Preliminary 2D CFD Domain Block Decomposition.....	160
Figure 6-2: The 2D CFD domain with the blocking lines visible	163
Figure 6-3: The fully meshed CFD domain for the NACA0015 aerofoil	164
Figure 6-4: Domain mesh in the vicinity of the rotating aerofoil fluid zone	165
Figure 6-5: The zoomed mesh with blocking edges of the aerofoil region	165
Figure 6-6: Zoomed in mesh around the blade trailing edge with blocking edges ..	165
Figure 6-7: Cp Plot Comparison from the present setup in this study compared to that of El Sakka et al.[76].....	167
Figure 6-8: Mesh independency study for the 2D aerofoil simulation used as datum for generating the ML Training dataset.	168
Figure 6-9: Scikit-Learn Multivariate Regression Fitting Comparison between Model Predicted Values and True Values from the training dataset.....	170

Figure 6-10: Graphical representation of the layers of a simple neural network [137]	172
Figure 6-11: Neural Network Activation Functions for the Input vs Output	174
Figure 6-12: Training history for the 5 layered Tanh Activation Model	177
Figure 6-13: Regression fitting accuracy of the CFD based Machine Learning Model	178
Figure 6-14: NeuralFoil prediction of the torque generation of the NACA0015 VAWT blade across the turbine rotation	179
Figure 6-15: Prediction of NACA0015 Aerofoil for the design turbine (TSR = 2.29, Windspeed = 7m/s)	180
Figure 6-16: Pair plot of the dataset distribution of the intended dataset in blue, and simulated dataset in red.	182
Figure 6-17: Predicted NACA0015 Tangential Force for a 10000 sample dataset	183
Figure 6-18: Torque generation of the NeuralFoil optimised camber deformation profile	187
Figure 6-19: Torque generation of the MLVawt optimised camber deformation profile	188
Figure 6-20: Optimised camber profile using the NeuralFoil predictor	189
Figure 6-21: Camber profile optimised using the Tensor Flow Machine Learning Prediction	189
Figure 6-22: Raw Torque values of the NeuralFoil optimised GA Camber Profile from ANSYS Fluent Report Definition	191
Figure 6-23 : Polynomial Curve Fitted to NeuralFoil based GA Optimised Camber Profile Torque	192
Figure 6-24: TensorFlow based GA Optimised Camber Profile Torque	192
Figure 6-25: Residuals of Polynomial Fitting between the fitted and raw Torque for the NeuralFoil Optimised Camber CFD Results	193

Figure 6-26: Residuals of Polynomial Fitting between the fitted and raw Torque for the TensorFlow Optimised Camber CFD Results.....	194
Figure 6-27: NeuralFoil Optimised Torque compared to NACA0015 turbine blade torque.....	195
Figure 6-28: Custom TensorFlow Optimised Torque compared to NACA0015 turbine blade	196
Figure 6-29: NACA0015 vs NeuralFoil Based GA vs TensorFlow Based GA Torque Profile	196
Figure 6-30: Velocity Contour at 15° Azimuth. a) NACA0015, b) NeuralFoil Based Optimisation Profile, c) TensorFlow Based Optimisation Profile.....	197
Figure 6-31: Velocity Contour at 150° Azimuth. a) NACA0015, b) NeuralFoil Based Optimisation Profile, c) TensorFlow Based Optimisation Profile.....	198
Figure 6-32: Velocity Contour at 225° Azimuth. a) NACA0015, b) NeuralFoil Based Optimisation Profile, c) TensorFlow Based Optimisation Profile.....	198
Figure 6-33: Torque of NeuralFoil GA Optimised Blade Deformation profile TSR = 1.5	200
Figure 6-34: Torque of MLVAWT ANN GA Optimised Blade Deformation profile for TSR = 1.5	200
Figure 6-35 : NeuralFoil Based GA compared with NACA0015 at TSR=3	201
Figure 6-36 : TensorFlow Based GA compared with NACA0015 at TSR=3	202

Nomenclature

Symbol	Description	Units
A	Area / Second order polynomial coefficient	m^2 /---
B	First order polynomial coefficient	---
$C_{l,mean}$	Mean Coefficient of Lift	---
C	Contraction Ratio	---
$\bar{c}$	Chord Length	m
C_d	Drag Coefficient	---
C_l	Lift Coefficient	---
C_p	Power Coefficient	---
C_t	Coefficient of Torque	---
f_x	Body Forces	N
F_x	Force acting along the X-axis	N
F_y	Force acting along the Y-axis	N
h	Enthalpy of fluid	$J\ kg^{-1}$
H	Turbine Blade Span	m
k	kinetic energy per unit mass of the turbulent fluctuations	$J\ kg^{-1}$
m	Maximum Aerofoil Camber	---
$\dot{m}$	Mass Flow Rate	$kg\ s^{-1}$
m_ϕ	Instantaneous Aerofoil Maximum Camber	---
p	Position of Maximum Camber	---
P	Pressure	Nm^{-2}
r	Turbine radius	m
R	Fluid Gas Constant	$J\ kg^{-1}\ K$
T	Torque or Temperature	Nm / K
t	Elapsed flow time	s
$t(x)$	Thickness Distribution Equation	---
U	Velocity of Fluid Flow	m/s
u	Velocity in the first principal axis	ms^{-1}
U_∞	Freestream Velocity	m/s
V	Velocity	ms^{-1}
v	Velocity in the second principal axis	ms^{-1}
V_t	Tangential Velocity	ms^{-1}

V_r	Relative Velocity	ms^{-1}
V_∞	Freestream Velocity / Wind Speed	ms^{-1}
w	Velocity in the third principal axis	ms^{-1}
$\bar{x}_u$	Upper Aerofoil Surface Element X co- ordinate	m
$\bar{x}_l$	Lower Aerofoil Surface Element X co- ordinate	m
y_+	Dimensionless wall distance	---
$y_c(x)$	Camber Line Equation	---
$\bar{y}_u$	Upper Aerofoil Surface Element Y co- ordinate	m
$\bar{y}_l$	Upper Aerofoil Surface Element Y co- ordinate	m
y_t	Thickness Distribution	---
α	Angle of Attack	<i>degrees</i> °
β	Slat Angle	<i>degrees</i> °
γ	Adiabatic Index (Ratio of Specific Heats)	---
ϵ	Rate of dissipation of turbulent kinetic energy	---
θ	Derivative of camber line equation	---
μ	Absolute Viscosity	
ρ	Density of Fluid	$kg\ m^{-3}$
τ	Fluid Shear Stress	$N\ m^{-2}$
ϕ	Azimuthal Angle	<i>degrees</i> °
ω	Rotational Velocity / Turbulent Eddy Dissipation Rate	$rad\ s^{-1}$ / ---
2D	Two Dimensional	---
3D	Three Dimensional	---
AI	Artificial Intelligence	---
AOA	Angle of Attack	<i>degrees</i> °
ANN	Artificial Neural Network	---
CAD	Computer Aided Design	---
CFD	Computational Fluid Dynamics	---
D	Diameter of Turbine	m
DES	Detached Eddy Simulation	---
EU	European Union	---
FSI	Fluid Structure Interaction	---

<i>HAWT</i>	Horizontal Axis Wind Turbine	---
<i>LE</i>	Leading Edge	---
<i>LES</i>	Large Eddy Simulation	---
<i>ML</i>	Machine Learning	---
<i>MLVAWT</i>	Machine Learning Vertical Axis Wind Turbine	---
<i>NACA</i>	National Advisory Committee for Aeronautics	---
<i>PINN</i>	Physics Informed Neural Network	---
<i>PMM</i>	Porous Media Model	---
<i>RANS</i>	Reynolds Averaged Navier Stokes	---
<i>RNG</i>	Re-Normalisation Group	---
<i>RSM</i>	Reynolds Stress Model	---
<i>RSO</i>	Response Surface Optimisation	---
<i>SIMPLE</i>	Semi-Implicit Method for Pressure Linked Equations	---
<i>SST</i>	Shear Stress Transport	---
<i>TE</i>	Trailing Edge	---
<i>TSR</i>	Tip Speed Ratio	---
<i>UDF</i>	User Defined Function	---
<i>URANS</i>	Unsteady Reynolds Averaged Navier Stokes	---
<i>VAT</i>	Value Added Tax	---
<i>VATT</i>	Vertical Axis Tidal Turbine	---
<i>VAWT</i>	Vertical Axis Wind Turbine	---
<i>UN</i>	United Nations	---

1 Introduction

1.1 Background of Renewable Energy

With further increasingly ambitious renewable energy targets being set and a rapidly approaching hard deadline for climate change reversal, it is more important than ever before to improve renewable energy technology. One of the most powerful driving factors in renewable energy research is the European Union's (EU) proactive stance on climate change, encouraging environmental concerns to be included in other policy areas such as agriculture and transport. Additionally at the 7th EAP, a call for a minimum of 20% of the EU budget for 2014-2020 to be devoted to climate change mitigation was made [1]. Alongside the EU, the United Nations (UN) have also been actively involved in promoting green energy. In late 2015 governing bodies of various nations met to discuss goals to combat climate change, discussing the steps required by all member nations. These terms were defined in The Paris Agreement of early 2016. The Paris Agreement has a collection of important and ambitious goals and targets, but also outlines the methods that should be taken to reach those targets. Articles 9, 10 and 11 of the 2016 Paris Agreement outlines the importance of helping developed countries to establish renewable energy systems. It also states the importance of developed countries to continue to utilise a wealth of financial resources, enabling innovation to help achieve their long-term goals such as "limiting the temperature increase to 1.5°C above pre-industrial levels" [2].

One of the fastest growing areas of renewable energy is in wind energy, with investments accounting for 63% of all renewable energy within Europe in 2018 [3]. It is therefore crucial that these investments are properly utilised, further lowering our dependence on fossil fuels. Wind energy is converted to mechanical and often electrical energy by devices known as wind turbines. These devices convert the kinetic energy of the wind flow into kinetic energy of the turbine rotor. This rotates a shaft which is utilised as is required. Historically, windmills could be considered the first wind turbines, using the wind energy to turn a mill for grinding wheat into flour as early as the 7th century. Whilst these devices are commonly used still today,

because of electrical advances, it is more common to attach the shaft to an electrical generator.

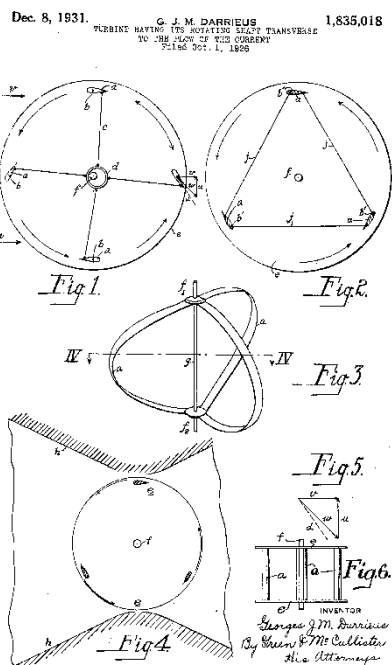
One emergent area of research is in the use of Vertical Axis Wind Turbines (VAWTs) due to their adaptability to the unsteady and erratic winds of urban environments. VAWTs are also able to utilise wind from any direction, unlike the more common Horizontal Axis Wind Turbines (HAWTs) which require a yaw adjustment to face into the oncoming wind [4]. Regardless of the extensive advantages of VAWTs over HAWTs, they are much less well established in industry, likely due to the low power production due to their inherent design. Therefore, it is crucial that they are thoroughly investigated to bring them to a comparably viable energy generation method.

1.2 Types of Vertical Axis Wind Turbines

There are two primary types of VAWTs; lift and drag based. The drag-based wind turbine is known as the Savonius, invented and patented by the Finnish engineer Sigurd Johannes Savonius in 1925. It utilised 2 overlapping vanes curved to create “an S-like passage of substantially constant area” [5], as seen in Figure 1-1a. These turbines have exceptional self-starting ability but have a drastic performance drop at higher tip speed ratios [6]. The primary limiting factor of drag based wind turbines, however, is their inability to achieve tip speeds higher than unity, as their tangential speed cannot be faster than the windspeed.

Lift-based VAWTs are often referred to as the Darrieus wind turbine, named after its inventor; Darrieus Georges Jean Marie, who filed the patent in late 1925 [7]. Rather than using the ‘scoop’ type rotor blades of the Savonius, a Darrieus wind turbine utilises aerofoils, which create a lift force and, as such, a moment coefficient. This moment then produces a torque which is characteristically used to drive a generator. A pictorial description of the Darrieus Turbine can be seen in Figure 1-1b. Albeit the Darrieus wind turbine was patented in the same year (1931) that the first HAWT we are familiar with was built, they are moderately unresearched in comparison. A newer adaptation of the Darrieus Turbine is more common in modern usage, which uses straight blades similar to an aircraft wing, known as the H-Type VAWT. The H-

The current literature on VAWT aerodynamic improvements varies from aerofoil shapes and optimisation to the employment of additional devices, providing static and active flow control options. This project aims to build upon the latter, active flow control through a combination of previously researched devices. By analysing the current literature, an understanding of the weaknesses and strengths of current technologies can be explored. This knowledge can then be employed in the combination and adaption of current technologies, and/or used to develop new technologies.



1.3 Aerodynamic Concepts

One reason for the extensive research on VAWTs is their complex, unsteady aerodynamics. This subsection outlines some of the core concepts that define their ability to operate, providing the reader with a fundamental understanding of the factors influencing their performance.

1.3.1 Conventional Aerofoils

A variety of aerofoils have been developed for different use cases. They are normally defined by the maximum thickness and position thereof, maximum camber and position thereof, alongside a series of other properties such as design C_l . One common feature between all NACA series aerofoils is that their last 2 digits are their maximum thickness in percentage chord.

The four primary types of NACA aerofoil are 4-Series, 5-Series, 1-Series and 6-Series. They vary based on different premises, for example the NACA 5-series defines the design C_l , NACA 1-series the position of minimum pressure etc. Within lift-based VAWTs, the 4-series is the most employed aerofoil profile due to its symmetric profile [11]. The general formulae for defining the NACA aerofoil with a sharp trailing edge include the thickness distribution $t(x)$, the mean camber line $y_c(x)$, the camber line gradient at $x \frac{dy_c}{dx}$, mean camber line angle at x ; θ , and finally the x and y coordinates for the upper surface $(\bar{x}_U, \bar{y}_U)$ and the lower surface $(\bar{x}_L, \bar{y}_L)$. The value for x is given for $0 \rightarrow 1$ (as in percentage along the chord for a unit chord length). The visualisation of these lines is seen in Figure 1-2.

$$y_t(x) = 5t_c \left[0.2969\sqrt{x} - 0.1260x - 0.3516x^2 + 0.2843x^3 - 0.1036x^4 \right] \quad 1.1$$

$$y_c(x) = \frac{m}{(1-p)^2} \left((1-2p) + 2px - x^2 \right) \left. \vphantom{\frac{m}{(1-p)^2}} \right\} \begin{matrix} \frac{m}{p^2}(2px - x^2) \\ \text{if } \begin{matrix} 0 \leq x \leq p \\ p < x \leq 1 \end{matrix} \end{matrix} \quad 1.2$$

$$\frac{dy_c}{dx} = \left. \begin{array}{l} \frac{2m}{p^2} (p - \bar{x}) \\ \frac{2m}{(1-p)^2} (p - \bar{x}) \end{array} \right\} \text{if } \begin{array}{l} 0 \leq x \leq p \\ p < x \leq 1 \end{array} \quad 1.3$$

$$\theta = \arctan\left(\frac{dy_c}{dx}\right) \quad 1.4$$

$$\bar{x}_U = \bar{x} - y_t \sin \theta \quad 1.5$$

$$\bar{y}_U = y_c + y_t \cos \theta$$

$$\bar{x}_L = \bar{x} + y_t \sin \theta \quad 1.6$$

$$\bar{y}_L = y_c - y_t \cos \theta$$

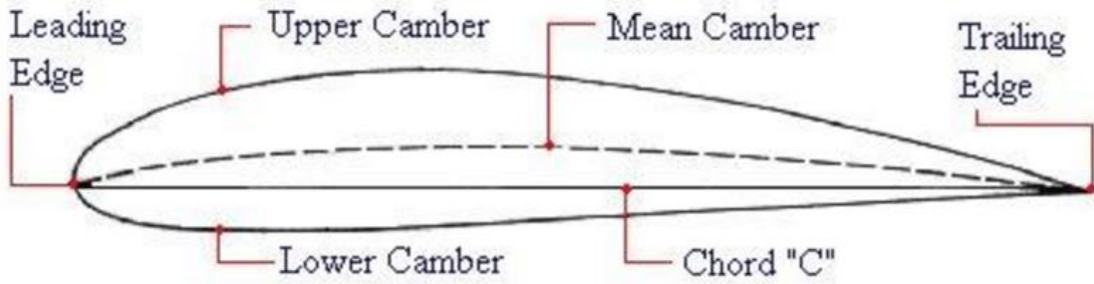


Figure 1-2: Cambered NACA 4 Series Aerofoil Schematic [12]

For an uncambered aerofoil, the mean camber line contributions in Equations 1.5 and 1.6 would be 0, the resulting equations are therefore seen in Equations 1.7 to 1.11.

$$y_{c(x)} = 0 \quad 1.7$$

$$\frac{dy_c}{dx} = 0 \quad 1.8$$

$$\theta = \arctan\left(\frac{0}{dx}\right) = \arctan(0) = 0 \quad 1.9$$

$$\bar{x}_U = \bar{x} - y_t \sin \theta \quad 1.10$$

$$\bar{y}_U = 0 + y_t \cos 0 = \frac{y_t}{c}$$

$$\bar{x}_L = \bar{x} + y_t \sin \theta \quad 1.11$$

$$\overline{y_L} = 0 - y_t \cos(0) = y_t \quad 1.12$$

1.3.2 Lift Generation

There are a few different ways of explaining how lift is generated. Still, they effectively all boil down to the same concept of turning the fluid direction, which creates a pressure difference and in turn creates a net force perpendicular to the direction of the incoming fluid flow. For the remainder of this explanation, the lifting body will be referring specifically to aerofoils, as they are the profiles that are extruded to form the blades of a H-Type Darrieus VAWT. Other examples of lifting bodies could be as complex as the wing of a fighter jet, as intriguing as a spinning golf ball or as simple as a flat plate.

There are two common groups when it comes to describing the lift generation, those that base the description on Bernoulli's principle and those that base the description on Newton's third law of motion. But there are three common incorrect theories about how these principles generate lift.

First incorrect theory is that a change in velocity of the fluid is often mistakenly assumed to be a result of "Equal Transit Theory" [13]. This theory claims that if two fluid particles travel parallel towards the lifting body, once they reach the lifting body, they will split, where one particle goes over the upper surface and one particle travels across the lower surface of the aerofoil. They are then assumed to reach the end of the aerofoil at the same time, which means for a cambered aerofoil, the particle that travelled across the upper surface has a longer distance to travel, and as such must travel faster. This is then fed into Bernoulli's principle that pressure must remain constant, so there is a lower pressure on the top surface than the bottom surface, generating a net upwards force. This is clearly incorrect, because this theory would suggest that uncambered aerofoils would not be able to produce lift, as the upper surface has an identical path length as the lower surfaces.

$$P + \frac{1}{2}\rho V^2 = \text{constant} \quad 1.13$$

The second common incorrect method of describing lift generation is a fallacy called “Skipping Stone Theory” [14]. This theory suggests that the incoming flow deflects off the suction surface and thus imparts a momentum change on the lifting body. Whilst the principle of a change in velocity is true, its origin is misunderstood.

The third incorrect method is the “Venturi Theory” [15]. This theory suggests that the flow velocity accelerates above a cambered aerofoil or symmetric aerofoil at a positive angle of attack, as a result of constricting the flow, in the same way that a Venturi nozzle does due to conservation of momentum. This relationship is shown mathematically in Equation 1.14

$$\dot{m} = \rho * V * A = \text{constant} \quad 1.14$$

This is fed into Bernoulli's principle, which would suggest that the dynamic pressure would be higher and so the static pressure is lower above the aerofoil, creating a net pressure difference in the upward direction and creating a lifting force. The reason why this is inaccurate however is that there is no arbitrary wall above the aerofoil that will create the nozzle and accelerate the flow, instead the velocity will slowly tend to the free stream velocity as you move away from the aerofoil [15].

The accurate description of how lift is a combination of aerodynamic concepts which describe how an aerofoil or any other lifting body turns fluid flow. In its simplest form, the turning effect of the aerofoil can be described using the Joukowski analysis method, with the Kutta condition. This theorem, aptly named the Kutta-Joukowski Theorem, was derived by the early 20th century aerodynamicists Martin Kutta and Nikolai Joukowski. Its premise is that the fluid streamlines are a superposition of rotating and translational streamlines, with the camber and α of the aerofoil providing the rotating component. The Kutta condition is critical to defining the flow as meeting smoothly at the trailing edge rather than recirculating on the upper surface, or the absence of a trailing edge separation bubble. In practice this results in streamlines that are more densely packed on one surface than the other, resulting in different static pressures and as such, a pressure difference creating a net force. Figure 1-3 depicts this effect, and it can be clearly seen that downstream of the leading edge, there is a larger density of streamlines. The limitations with this theory

however are that it is designed for inviscid, irrotational, attached flow, but has been seen to be applicable in most operating conditions of aerofoils [16].

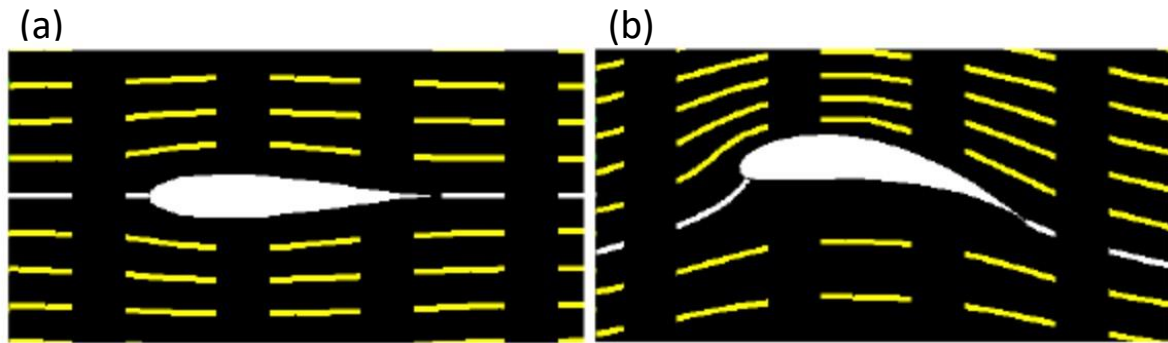


Figure 1-3: Typical Streamlines for Aerofoils at $\alpha = 0$ [17]. (a) Streamlines of Flow around Symmetrical Aerofoil. (b) Streamlines of Flow around Cambered Aerofoil.

To understand how this turning effect imposes a lift force, consider the aerofoil in Figure 1-3(b) with chord c and infinite span (effectively equivalent to 2D) moving through a fluid of density ρ at a speed of V . The velocity above the aerofoil can be defined as $V+c$ and the velocity on the lower side is V

Combining Bernoulli's conservation of energy approach and Newton's conservation of mass results in a complex series of equations known as the Euler Equations. Later, the Navier-Stokes equations included the concept of viscosity, to aid in modelling the boundary layer growth effect on the "shape" of the aerofoil. These equations only have analytical solutions in very limited applications and are therefore normally "solved" using numerical iterative processes employed in commercial codes.

Arguably the most important concept in the lift generation is the angle of attack (α), defined as "The angle between the free stream and chord line of the aerofoil" [16]. There is a well-defined relationship between the lift coefficient (C_l), drag coefficient (C_d) and angle of attack (AOA). Typically, the C_l will increase until the stall angle, after which

the lift will rapidly decrease until no lift is produced. This can be seen in Figure 1-4.

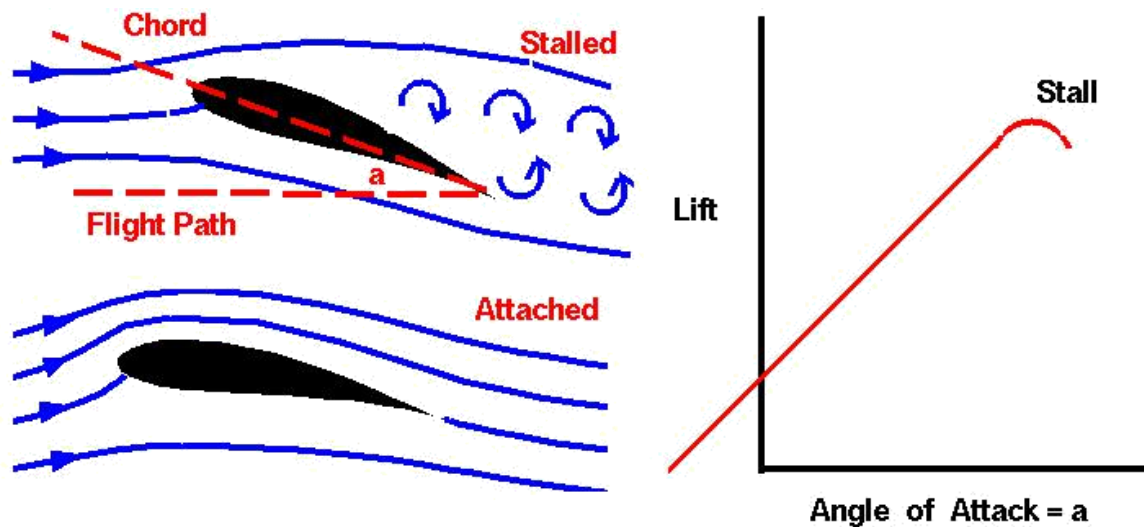


Figure 1-4: Effects of Angle of Attack on Lift Coefficient for a Cambered Aerofoil [18].

It should be noted that the $C_l - \alpha$ curve shown is for a cambered aerofoil, as a symmetrical aerofoil does not produce lift at $\alpha = 0^\circ$.

The stall phenomenon occurs due to a large separation of the flow on the upper (pressure) surface. The separation is not present at low angles of attack, as the adverse pressure gradient of the flow near the trailing edge is not sufficient, but as α increases, a separation bubble starts creeping towards the leading edge [18].

Typically, the geometrical factors which have the largest influence on the aerofoil performance are the thickness and camber. Thickness does not directly affect the lift production of an aerofoil and mainly affects the transition and separation properties [19]. Camber however is one of the largest contributors to the lift for several theories.

1.3.3 VAWT Aerodynamic Theory

The relative velocity triangles seen in Figure 1-5. are used to derive the proceeding equations (1.15 $\rightarrow$ 1.21), which can be used to calculate the theoretical AoA (α). It is important to clarify the difference between the theoretical angle of attack, and the true angle of attack. In situ, some of the energy of the flow is extracted in the upstream wake, $0 \leq \phi \leq 180$, thus reducing the flow's velocity magnitude and therefore reducing the downstream α based on the wake of the upstream stroke. In this diagram, V_t represents the tangential blade velocity, as calculated using Equation 1.15

and Equation 1.16, V_∞ represents the freestream velocity, ω is the radial velocity of the turbine, r is the turbine radius and ϕ represents the azimuthal angle, taken from 0° at the top of the turbine rotation. Using simple trigonometry, it is possible to calculate the relative blade velocity (V_r) as in Equation 1.17 and the theoretical flow angle of attack (α), Equation 1.18 as a function of tip speed ratio and azimuthal position.

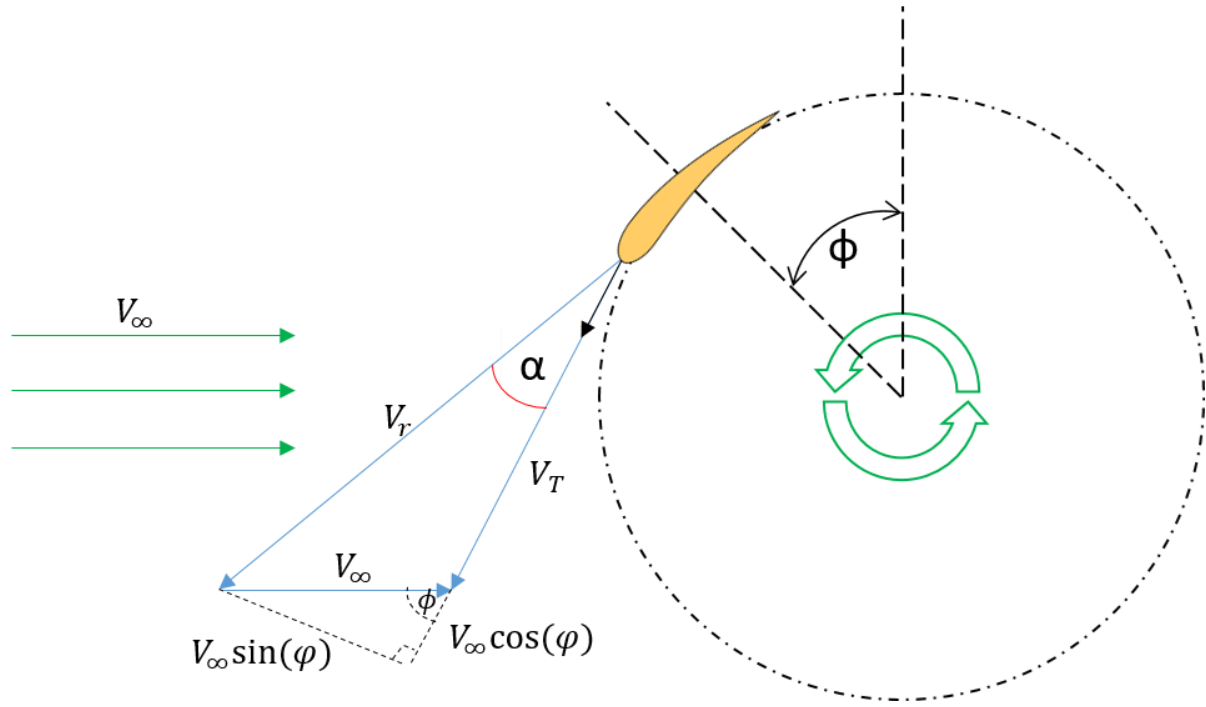


Figure 1-5: Relative flow velocity triangles for a rotating aerofoil and constant inflow velocity

$$\omega = \frac{TSR * V_\infty}{r} \quad 1.15$$

$$V_t = \omega * r \quad 1.16$$

$$V_r = \sqrt{(V_T + V_\infty \cos(\phi))^2 + (V_\infty \sin(\phi))^2} \quad 1.17$$

$$\alpha = \frac{\sin(\phi)}{TSR + \cos(\phi)} \quad 1.18$$

By analysing the lift and drag polars, one can calculate the effective local tangential force coefficient, C_t . The tangential force coefficient calculated is generated using Equation 1.19, where C_l represents the blade's coefficient of lift and C_d is the blades coefficient of drag.

$$C_t = C_l \sin(\alpha) - C_d \cos(\alpha) \quad 1.19$$

$$M = \frac{\sqrt{((\omega \cdot R) + V_\infty \cos(\phi))^2 + (V_\infty \sin(\phi))^2}}{\sqrt{\gamma R T}} \quad 1.20$$

$$Re = \frac{\rho \cdot \sqrt{((\omega \cdot R) + V_\infty \cos(\phi))^2 + (V_\infty \sin(\phi))^2} \cdot \bar{c}}{\mu} \quad 1.21$$

The Reynolds number and blade angle of attack are proportional to the tip speed ratio of the turbine, with lower tip speed ratios having significant portions where the angle of attack is significantly greater than the stall angle for most symmetric NACA aerofoils. Looking at Figure 1-6, the aerofoil will be clearly stalled in most of the turbine rotation for tip speed ratios of 2.29 and 3, and a significant portion at a tip speed ratio of 1.5.

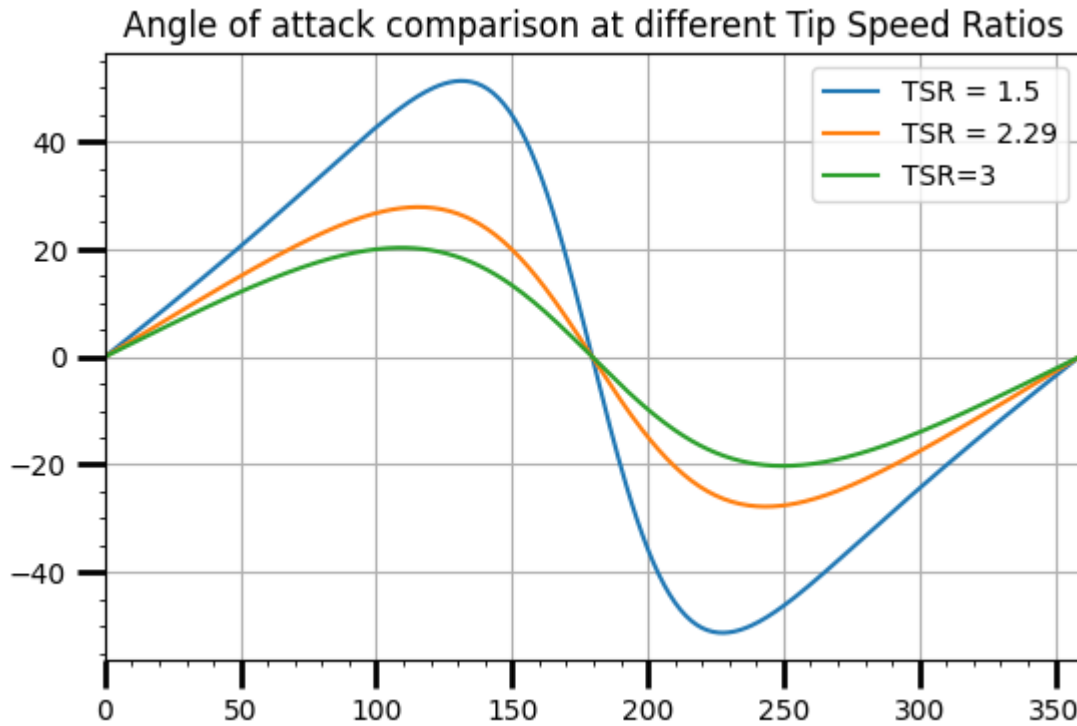


Figure 1-6: Angle of attack comparison for multiple tip speed ratios

As an increase of Reynold's number also helps delay boundary layer separation

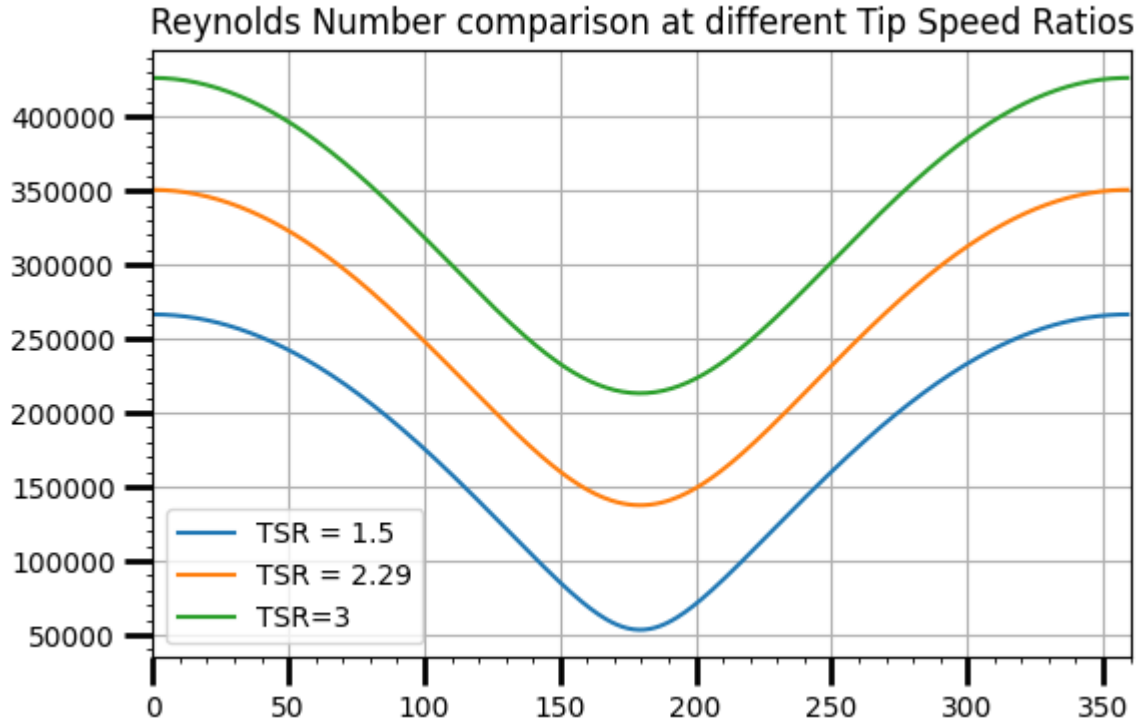


Figure 1-7: Reynolds Number Comparison for multiple tip speed ratios with a wind speed of 7 m/s and a turbine radius of 0.85m

1.4 Aims and Objectives

As a result of the current requirements and environmental demands, straight bladed VAWTs will be the scope of this thesis as their adaptability makes them suitable in the ever growing urban sprawl throughout the world.

The primary aims of this thesis is to improve the aerodynamic performance and efficiency of the VAWT, to offset their typically asymmetrical power generation, through design optimisations of the turbine blade profile. As a result, various relevant techniques and methods will be researched in the following sections to achieve this. State-of-the-art computational fluid dynamics (CFD) simulations will be utilised for the modelling and investigation approach, due to their adaptability and substantial cost savings over physical model testing in wind tunnels.

These aims will be achieved through three different approaches to modelling the turbine profile's performance characteristics; using a high speed panel method solver, a complex and thorough transient CFD simulation and finally a CFD surrogate AI model to estimate turbine power based on blade profile characteristics and operating conditions. These tools will all be fed into a genetic algorithm (GA)

optimiser to find the most suitable variant of the blade profile's camber magnitude at pre-specified azimuthal positions in the turbine's rotation. The primary research objectives will therefore be;

1. Creation and validation of a verified 2D CFD simulation to provide accurate and reliable understanding of the turbine flow physics and performance.
2. Generation of a Genetic Algorithm Optimiser procedure that can be easily ported between different turbine simulation tools.
3. Investigation and implementation of the GA in a publicly available panel method solver and understanding of its limitations.
4. Investigation and implementation of the GA in a complex and robust CFD-In-The-Loop optimisation procedure to consider the transient effects of the turbine rotation on the blade profile.
5. Design, build and training of an artificial neural network (ANN) to act as a CFD surrogate including dataset generation, pre and post processing of the data, ANN architecture investigation and implementation in the GA as a replacement for the panel methods.

2 Literature Review

There are numerous technologies currently being employed in trying to improve the performance of the Vertical Axis Wind Turbine. They vary substantially from implementing ancillary technology to entirely altering the overall turbine design or blade aerofoils. This chapter aims to outline some of these performance improving techniques and to critique their strengths and weaknesses.

Due to the aerodynamics of VAWTs being comparable to that of the wings on aircraft or helicopter rotors in certain azimuthal angles, the technology used in the latter is often applied to the former. The most crucial comparable aspects are dynamic stall [31], [32] as it has been shown to have significant impacts on the performance [33], [34], [35] of both VAWTs and helicopter rotors. It is therefore crucial to investigate if the techniques implemented to reduce dynamic stall in aircraft wings and helicopter rotors are appropriate within the scope of VAWTs. The primary areas of this research are aerofoil profile modifications, application of optimisation techniques for VAWT blades and machine learning utilisation, which will therefore be the target of the literature review.

2.1 High Lift Devices

High lift devices have been widely used within the aviation industry for almost a century [36], but only marginally in wind turbines, and less so specifically in VAWTs. These devices work by creating an effective increase in the camber of the aerofoil, in turn altering the circulation of the flow, and as such the strength of the vortex generated [37]. Typically, these devices are situated at the trailing or leading edge in the form of flaps or slats, respectively. It is worth mentioning that static slats, in that their angle of attack cannot be altered and they are more commonly referred to as slots.

Flow controlling devices have an extensive amount of well-established research due to the attraction of retrofitting current VAWTs with this technology. This provides the opportunity for the performance of existing wind turbines to be increased, without complete overhaul of design. They are typically split into 2 subsections: active and passive. Active devices are often implemented within a feedback loop to provide stability to an aircraft. This allows constant parameter to change without the intervention of the user, thus optimising the efficiency. Passive devices however are set to a constant value, i.e. deflection angle, and then forgotten. These passive devices are sometimes automated, in the sense that their flow conditions actively change their workings, but are not controlled by the user. These 'automated' passive devices subtract somewhat from the appeal of using passive flow control devices, due to the necessity of maintenance required to maintain their functionality.

2.1.1 Leading Edge Devices

Research on slats was first published by F. Handley Page on December 22nd, 1921 [38]. They found that creating a slotted wing allows for air acceleration in the passage between the auxiliary and main aerofoil. This in turn appeared to drastically improve the lift scalar, thus reducing the onset of stall whilst simultaneously increasing the lift coefficient [38]. Their primary use is during take-off and landing of aircraft but have occasionally been considered for VAWTs too.

Chougule et al. [39] tested the influence of slat angle on VAWTs. They tested a series of angles of attack for the slat aerofoil and compared the lift and drag polars with the standard design. It was found that the 20° slat angle (β) had not yet initiated stall at the maximum testing angle of 24° , and aerodynamic performance quickly started deteriorating at $\beta = 25^\circ$ and greater. To compare the power coefficients of their new blade with a commercially available wind turbine, they used the *Double Multiple Stream Tube Method* (DMSTM) numerical analysis method. They maintained the same design parameters such as rotor height, diameter etc. but compared the numerical simulation of their turbine with experimental results of the commercial turbine. It should also be noted that the flow is assumed to be inviscid for the DMSTM model, which could have a substantial influence on the power coefficients. The numerical simulation suggests a largely different power coefficient, but similar total power

output as seen in Figure 2.1, which appears to be contradictory from the expected performance due to the different chord lengths used between the DMSTM and experimental data.

Whilst convention is to place the slat such that the flow stream is guided to the pressure surface, Scungio et al. [4] examined the influence of placing them on the suction side, this can be seen in Figure 2.2. An experimental comparison between a standard three bladed wind turbine and their specialised turbine was undertaken. The wind tunnel used is a closed-circuit open chamber, with pressure sensors to ensure the constant flow velocity accounting for blockage effects. They discovered that the use of an optimal slat improved the self-starting ability of the wind turbine and more dynamic torque at low wind speeds [4].

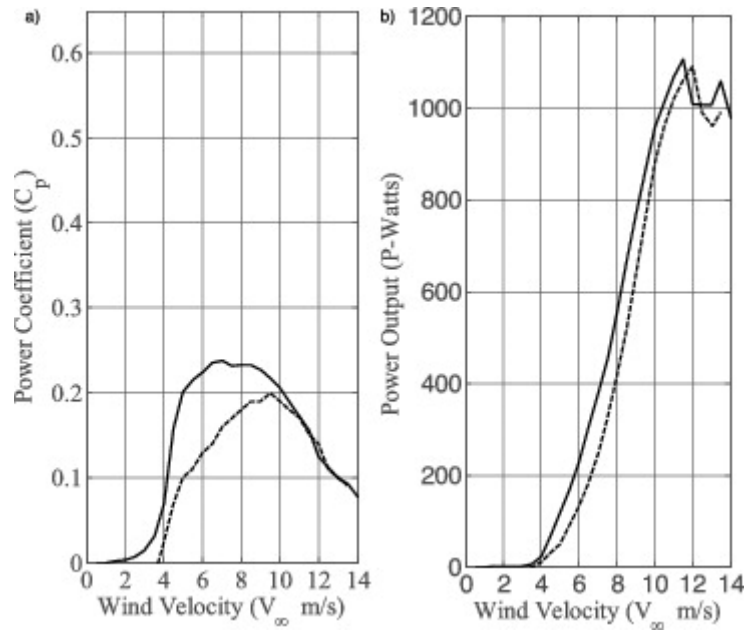


Figure 2-1: Chougule et al. Verification of the DMSTM model [39].

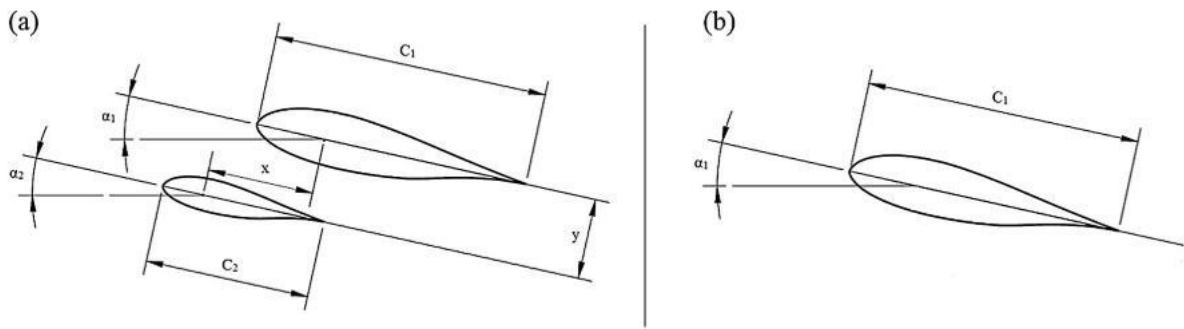


Figure 2-2: Scungio et al. Auxiliary Blade Layout Compared with Standard Darrieus Turbine [40].

2.1.2 Trailing Edge Devices

Flaps also have a long history in aviation, first being produced on the Fairey Hamble Baby aircraft in 1916 [41]! This flap is now known as a plain flap, in which the flap is formed as part of the aerofoil. These are typically not seen as much nowadays due to improvements on the design but are still the most fundamentally simple.

The plain flap has been developed and adapted since its conception, with the most notable innovations being the Fowler flap, Junkers flap and slotted flap. These advanced flaps all have varying characteristics with the aim of altering the performance of the core aerofoil. The Fowler flap uses a very complex system of hydraulics to extend the flap out of the original aerofoil, thus assisting not only with camber and AOA alterations, but also for increasing wing surface area. The Fowler flap does not modify the existing aerofoil but rather has an ancillary aerofoil acting as the flap. However, these progressive flaps do not mean plain flaps are not suitable for use in VAWTs, as their simplicity assists in minimising maintenance costs. Additionally, their simplicity allows for cost-effective and easily managed implementations.

The Fowler flap has widespread use in modern aircraft but has not been investigated in the use of VAWTs and this is likely due to the structural complexities that would be associated. This may be an effective solution to the balance between start-up ability and higher power coefficients as a blade optimised for higher tip speed ratios (TSR) could still show some self-starting properties.

Minor research has been performed on the use of Junkers flaps in the form of auxiliary blades on the wind turbine. Li et al. [42] studied the effect of various fixing angles of a secondary blade for a series of symmetrical NACA aerofoils. A commercially available CFD code (ANSYS FLUENT) was used with the Reynolds Averaged Navier Stokes (RANS) equations, employing the RNG $k-\epsilon$ turbulence model. It was found that using an auxiliary blade, with an appropriate thickness and angle of attack (AoA), significantly altered the overall static torque and power production of the turbine [42]. They also discovered that by setting the flap at a small angle to the main blade it further improved the performance.

Albeit that the slotted flap is slightly dated in modern aviation, its use in VAWTs has previously been explored. Wei and Li [43] briefly investigated the effect of having a two-element aerofoil. Unfortunately, their experimentation was not particularly thorough, with most of the publications comparing different simulation methods such as experimental, XFOIL and CFD. The brief experimentation of the two-element model was tested at a 0° and 10° deflection, and their comparison was based solely on the lift-coefficient. It was noted that the use of the flap altered the angle of attack of the blade, resulting in a lower C_l - α curve, and that a proper deflection profile could improve the performance. It is important to note that their computational models are not optimal either. When using the Spalart-Allmaras model, it is necessary to keep the y^+ value either less than one or greater than 30, due to its wall functions, whereas theirs varies from 0.5 to 9 [44]. Building on from this work, Yang et al. [45] used the same theory of utilising a slotted flap, but with an active control system. A two-dimensional unsteady CFD study was undertaken using dynamic meshes. The datum aerofoil was a NACA0012 with a trailing edge slotted flap, which creates room for some speculation of the results. Comparing the flapping aerofoil with a flapped aerofoil with no deflection on the flap, does not provide the authors the opportunity to compare it to the common “single seam” aerofoils. By implementing active control on the deflection angle of the flap, the dynamic stall was able to be delayed [45]. The active control was a sinusoidal profile, changing the flap angle with respect to azimuthal angle. This is represented pictorially in Figure 2.3. Additionally, it was found that there was a positive effect on the trailing edge wake and on the amplitude of oscillations [45]. In terms of turbine performance, the peak power coefficient was increased by 10%, and probably this is due to the minimised flow separation from the lower angle of attack.

Some alternative approaches to the flap have also been undertaken. Liu et al. [46] created a movable flap close to the trailing edge, and studied its influence on noise and aerodynamic performance. They considered the position, length and angle of the flap's effect on performance and noise generation. The numerical method employed was the Detached-Eddy Simulation (DES) which is based on the Spalart-Allmaras single equation turbulence model and can be treated like the Large Eddy Simulation

(LES) in refined enough sections of mesh and Reynolds Averaged Navier Stokes (RANS) elsewhere [46]. They had a series of tests with this flap, starting with a series of static inflow tests to calculate the aerodynamic polars. This showed that at lower angles of attack the flap hinders performance, causing large separation. When testing in the scenario of a vertical axis wind turbine however, they found that the flap helped to suppress the vortices when an active control was implemented. The flap would open between azimuth angles (ϕ) of 90 and 135 and then proceed to close at $135^\circ < \phi < 180^\circ$. This resulted in higher torque coefficients at each tested TSR, albeit only from 1.2 to 1.5.

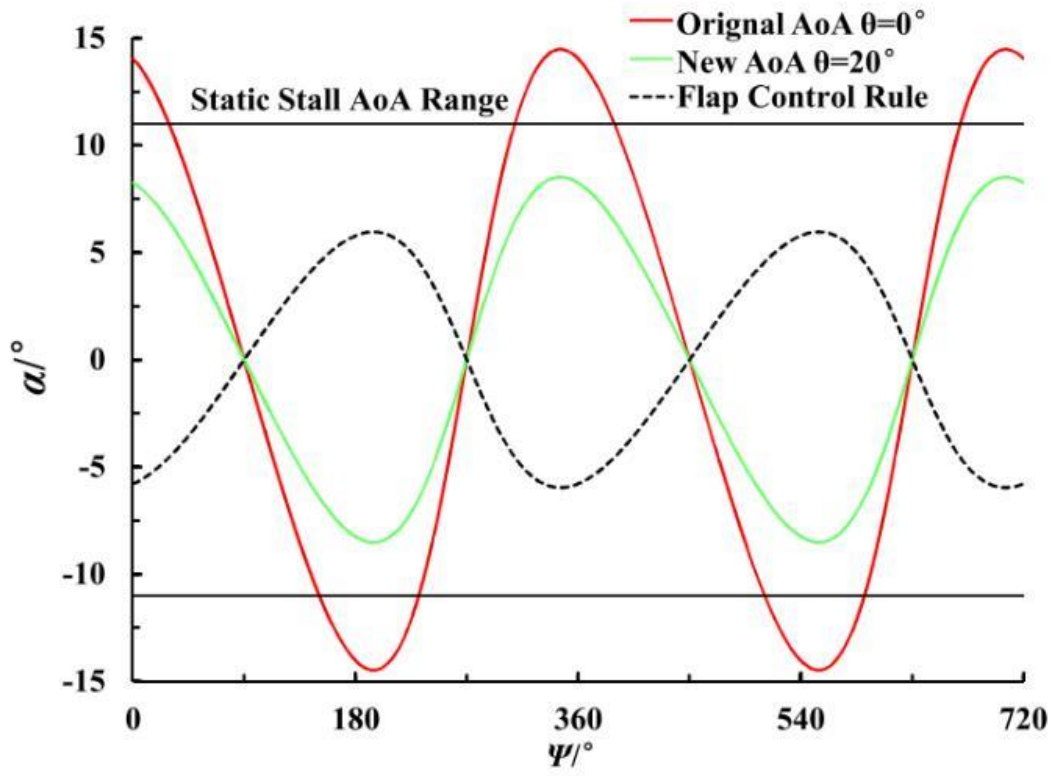


Figure 2-3: Angle of Attack as a function of the Azimuth Angle, Yang et al. [5].

Gurney Flaps are most commonly known from their presence in motorsports such as Formula 1, to assist in increasing their grip [47]. Recently, they have also been shown to be adaptable to VAWTs. Bianchini et al. [48] tested this theory using 2D unsteady CFD simulations in both static and rotating domains. Three initial static test cases were made for aerodynamic polar calculation; a flap on the pressure side, suction side and on both sides and compare against a baseline aerofoil. A sensitivity analysis on the flap height was also undertaken at 3% chord length ($\bar{c}$) and $4\%\bar{c}$ against the baseline $2\%\bar{c}$. Finally, the optimised flap was implemented on a three bladed vertical axis wind turbine through a rotational domain. The SST $k-\omega$ turbulence model with enhanced wall treatment was used alongside an unsteady RANS method, with the Coupled pressure-density algorithm. The $2\%c$ on the inner side of the aerofoil created a 23.1% increase in the power coefficient at a TSR=3.3 [48]. This comparison can be seen in Figure 2.4. When the $3\%c$ analysis was undertaken, it is seen that the use of an inner flap improves the pressure coefficient at $TSR < 2.7$ before the baseline aerofoil starts to perform better. This can be seen in Figure 2.5. Zhu et al. [49] also researched Gurney Flaps on wind turbines, but considered the effect of solidity, rather than one base turbine, alongside both conventional and dimple Gurney flaps. It was noted that increasing the solidity improved the power coefficient (C_p) at lower TSR but also reduced the peak TSR alongside the performance at higher TSR. This was seen in both the turbine with and without dimples, but only an outboard Gurney flap. The Transition SST model was used due to its accuracy in relation to the experimental data, but requires a very fine boundary mesh. This was well implemented and as such, the results can be considered reliable. Increasing solidity improves C_p at low TSR but reduces peak TSR and performance at higher TSR.

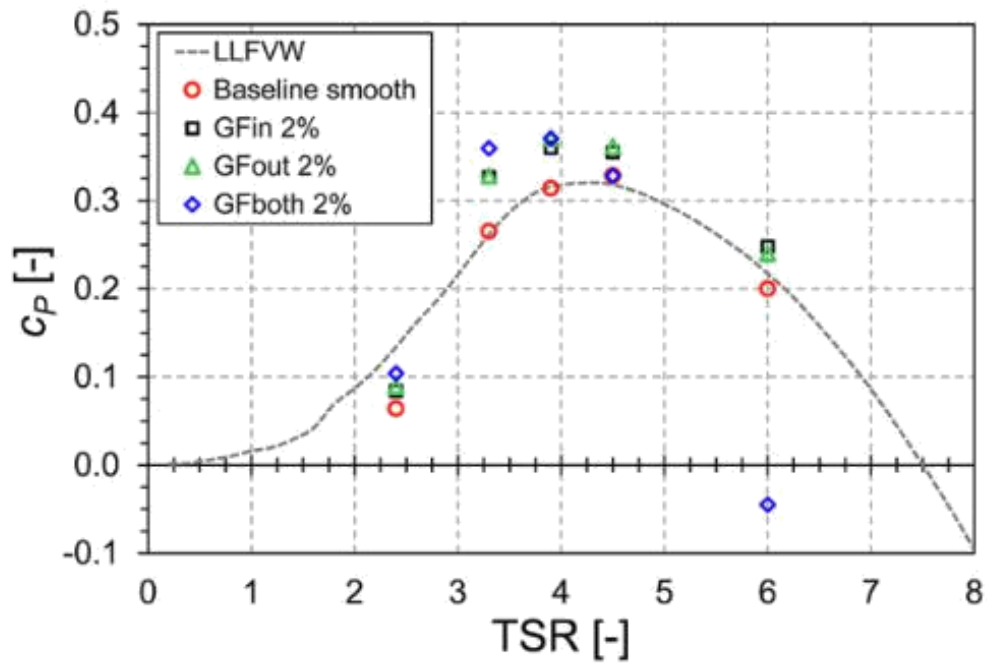


Figure 2-4 Pressure Coefficient as a function of TSR for 2%c Flap [48]

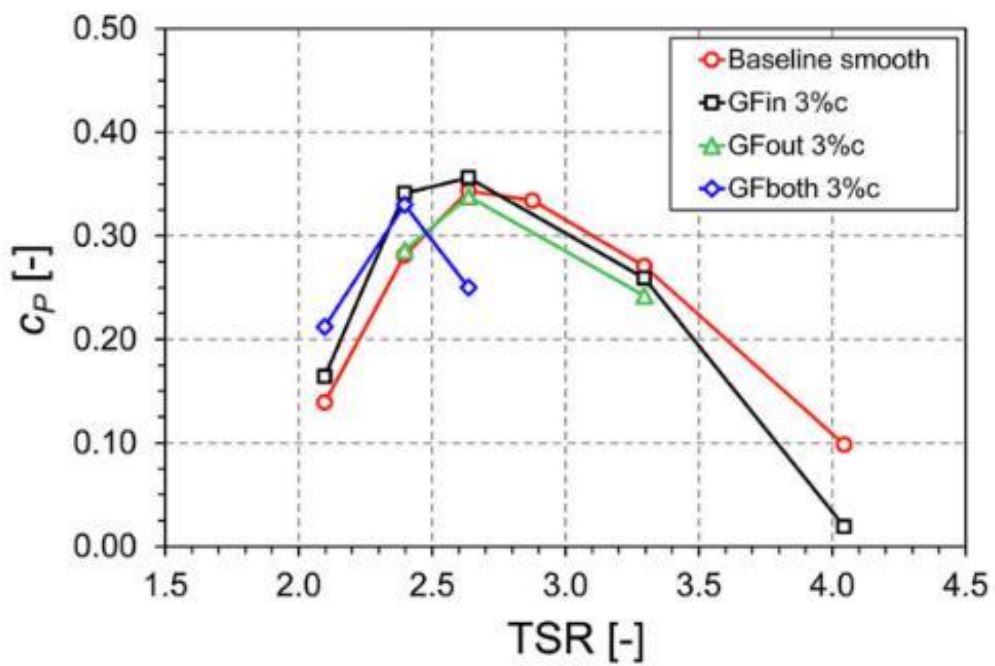


Figure 2-5: Pressure Coefficient as a function of TSR for 3%c Flap [48]

2.2 Turbine Blade Modification

Auxiliary devices are not the only techniques that have been explored in an attempt of improving turbine performance. There has been a vast array of alternative aerofoils, some of which are drastic and some minor changes.

2.2.1 Flow Manipulation Techniques

A promising research avenue is in the use of jets to help delay flow separation at lower tip speed ratios, but it must be remembered that there is a certain amount of power draw that will be taken from the pumping devices. Rezaeiha et al. [50] recently investigated this by implementing a suction slot on the inner surface of a NACA0018 wind turbine with a solidity of 0.06. Their study was numerically simulated using ANSYS Fluent with a well defined mesh ($y_{+,max} < 1$) using the Transition SST turbulence model within an incompressible URANS approach. The slot had a velocity inlet boundary condition of -6.23m/s. The suction scenario greatly increased the overall power coefficient, due to the improved lift to drag coefficient ratio. Sensitivity analysis was also undertaken to determine if the slot's inlet velocity, position and Reynolds number of the flow. Whilst the velocity magnitude did not seem to affect the moment coefficient much from 0.5% freestream velocity (U_∞) to 10% of U_∞ , the different positions and Reynolds number drastically altered the performance of the turbine. The effect of the position is more prominent at lower TSR and tends to converge as TSR increases. Figure 2-6 shows the positions to help give perspective to Figure 2-7 which displays the effect of changing the position at a TSR of 2.5. For the given wind turbine a sizeable improvement is seen, particularly at low TSR. If the power required for the pump is taken into consideration and assumed to have a constant flow rate, that is to ignore any pressure differences on the inner surface, a net power gain for the TSR of 2.5, 3 and 3.5 was 219.6%, 74.3% and 19.6% respectively

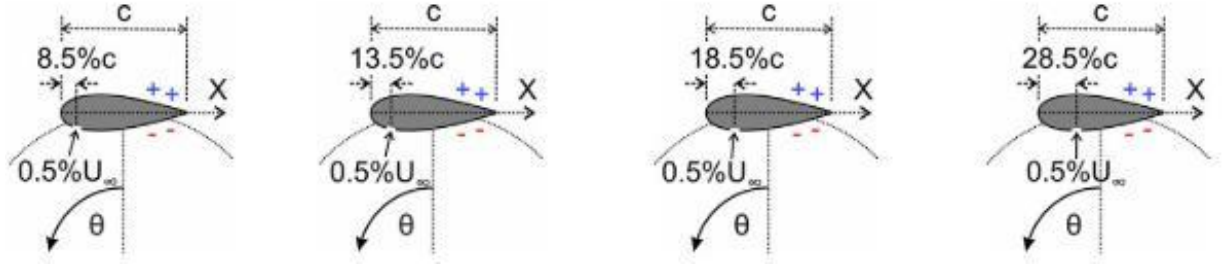


Figure 2-6: - Suction Slots Locations [50]

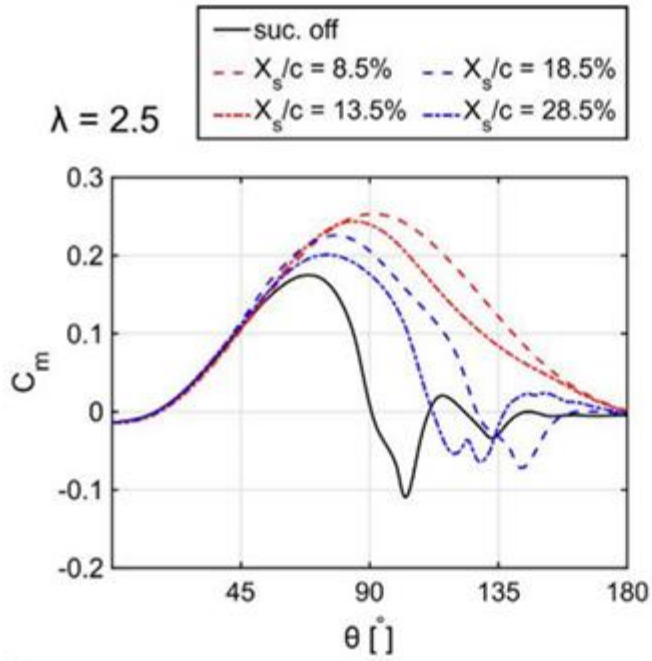


Figure 2-7: Suction Slots Performance [50]

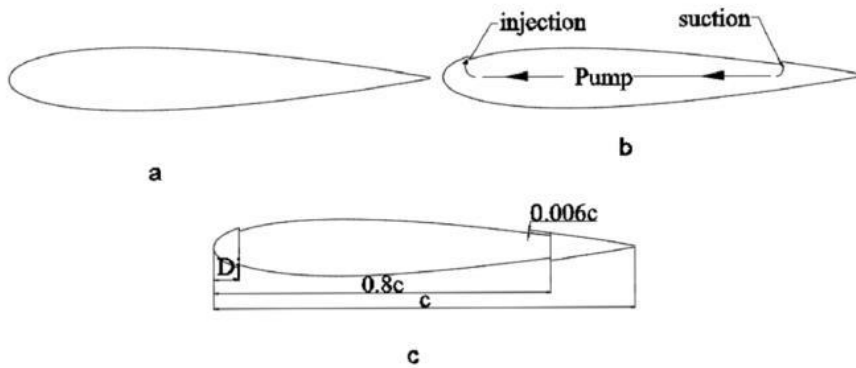


Figure 2-8: Depiction of flow jet domain for a) NACA0015, b) generic CFJ-NACA0015 and c) CFJ-NACA0015 for VAWT application [51].

A similar concept was researched by Sun et al. [51] known as a co-flow jet. Rather than sucking from the suction side of the aerofoil, the flow is siphoned from near the trailing edge of the pressure side and injected near the leading edge of the pressure side as seen in Figure 2.8. They performed a comparison between the different turbulence models to decide on the most appropriate, whilst ensuring appropriate y^+ values were created in the mesh for each. The SST $k - \omega$ model was selected as it best matched the experimental data. To ensure mass continuity the two jet sections were set as velocity inlets with the same magnitude but different sign. It greatly improved the low TSR performance of the turbine by 170%, 120% and 17% at TSR of 0.8, 1 and 1.25 respectively. This was particularly prevalent when using their intermittent method. The intermittent method simply alternates the co-flow jet between the inner and outer blade surface dependant on azimuthal position.

2.2.2 Aerofoil Profile

Arguably the most important areas of research within the scope of VAWTs is that of the aerofoil profile itself. This can include using asymmetrical aerofoils, increasing camber, modifying specific sections etc.

Song et al. [52] investigated the influence of altering the leading-edge radius of a standard aerofoil within the industry, a NACA0015. An unsteady, two dimensional CFD approach was undertaken using the Realisable $k - \epsilon$ with the SIMPLE pressure scheme. Whilst the y_+ value is 6.2 for the selected mesh, it might still be able to resolve the viscous sub layer due to the higher Reynolds number [44]. It seems that by increasing the leading-edge radius, it tends to slightly increase the power coefficient up until 6%c, whereafter, it quickly reduces in performance. For 3%c to 7%c the performance is marginally better than the conventional NACA0015, but 1%, 1.5%, 2%, 8% and 9% all appeared to have worsened performance. This can be seen in Figure 2.9.

Wang and Zhuang [53] tested if dynamic separation could be minimised or limited by adding sinusoidal serrations on the leading edge. Star-CCM+, a commercial CFD software, was used for modelling the flow and numerical simulation. A thorough mesh dependence study was undertaken to validate the reliability of the numerical simulation and to build confidence in the results it output. This found that the Realisable $k - \epsilon$ model used under predicted the vortex shedding of the shafts and turbine blades at some azimuth angle, but was otherwise satisfactorily [53]. The discretisation process resulted in y_+ values along the blade profile of mostly 30-50, so the *all* y_+ wall treatment was used. First a series of fixed wavelength, increasing amplitude serrations were compared with the base profile. The reduced amplitude designs improved the C_p , with the smallest amplitude (0.025c) having a consistently better C_p for all TSR. The 0.025c design was then tested with different wavelengths and the smallest (0.33c) wavelength had the most improved overall performance, with a 18.7% improvement on CP at a TSR of 2.

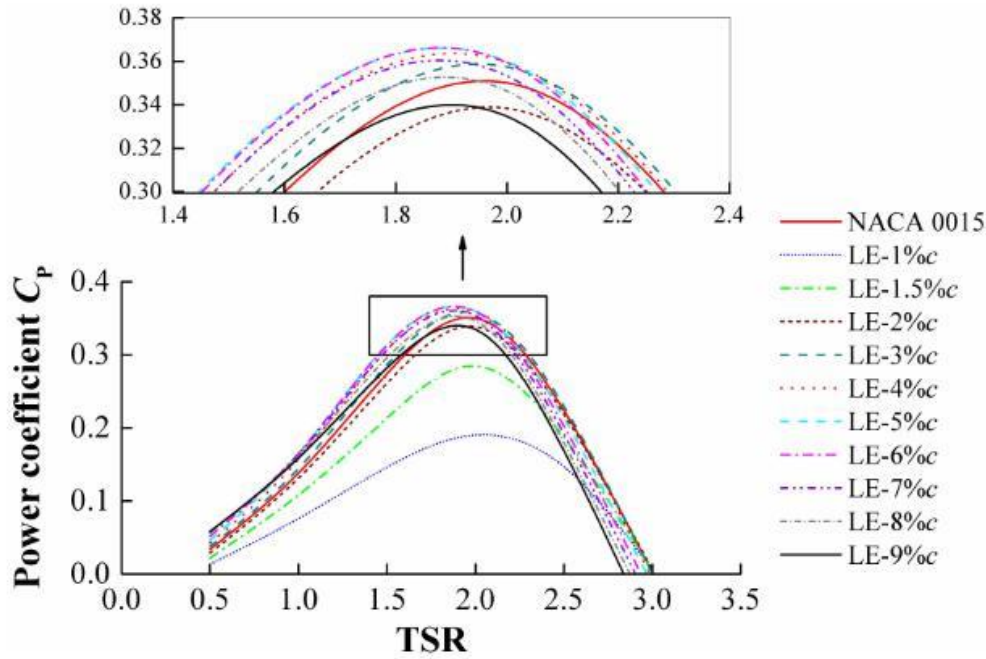


Figure 2-9: Leading Edge Radius Power Coefficient [53].

Whilst altering existing commonly used aerofoils develops an understanding of what improving the established norm does, investigating alternative aerofoils is another promising approach. Wang et al. [54] investigated the feasibility of this concept by investigating the aerodynamic performance of a series of different aerofoils in a VAWT scenario. A thorough array of profiles were selected and tested numerically through the use of CFD with an O type grid and sliding mesh methods. The $k\omega$ – SST model was used for turbulence modelling. The tested aerofoils are a series of NACA 00XX, NACA63-0XX, NACA4X15 and NACA X015. For the symmetrical aerofoils, the positions of maximum thickness were altered to understand their influence on performance. It was found that moving the maximum thickness from LE to TE resulted in increasing and then decreasing C_p after an optimal value. It was also found that the asymmetrical NACA 4 and 6 series aerofoils performed better at lower TSRs, with performance decreasing as TSR increased. It should be noted that the surface pressure curves indicated that a smaller maximum camber allowed for larger work ability [54]. Liu and Xiao [55] investigated the impact that structural stresses may have on the power extraction performance of VAWTs. A focus on spanwise deformation with a priority on bending and twist deflection was used. They utilised Fluid-Structure-Interaction (FSI) within a commercially available software to simulate the coupled behaviour that aerodynamic and structural loads exhibited in-situ. It was found that the stiffness of the blade drastically effects the performance of

the turbine, with less rigid turbines showing poorer performance, likely due to the poor aerodynamic shape because of larger deflections. This provides plenty of insight into design constraints that should be considered in the manufacturing of these machines.

Zeiner-Gundersen [56], [57] looked at the use of a Vertical Axis Tidal Turbine (VATT) with flexible aerofoils and spring-loaded pitching. It was found that the flexible aerofoils allowed the turbine to remain at an almost constant TSR variation, only increasing by a factor of 1.3 with a doubled inflow velocity! Additionally, this turbine resulted in typically higher power coefficients at low TSR when compared with static, fixed blade turbines with lower solidity. This is impressive, as typically the lower solidity results in a larger C_p and narrower operating range of TSRs. Additionally, it appeared to improve the startup ability of the turbine. Miao et al. [58] researched the effect of a flexing aerofoil on its propulsive efficiency. Whilst the propulsive aspects of a flapping are not of particular relevance in the scope of VAWTs, some of the aerodynamic concepts discussed, such as lift and drag, are useful. They utilised an unsteady, laminar flow model in Fluent 6.1 with a deforming mesh. The mesh is a structured, conforming C-type grid close to the aerofoil before having the far field as a triangular coarser mesh. Noticeably, one of the most significant results is the fact that the larger the frequency of flapping, the larger the required input power to power the flapping is. Additionally, if the flexure amplitude is increased too much, there is substantially more drag than thrust produced. It should also be noted that as the Reynolds number of the flow increases, the peak propulsive efficiency occurs with lower flap frequency. Based on these conclusions, this type of technology is likely better suited to large offshore turbines, due to their typically lower tip speed ratios, alongside their usually larger initial investment costs.

Zhuang et al. [59] investigated what impact having a morphable trailing edge would have on aerodynamic load control in the design of HAWTs. Whilst the aerodynamics are fairly different to that of a VAWT, some of the techniques employed and conclusions are equally valid. The use of a deforming mesh boundary layer allows for

simulation of a smooth and consistent aerofoil deformation according to the transformation matrix. This uniformity is dependent on the node count on the aerofoil surface, with little improvement in smoothing for coarse discretization [59]. The diffusion coefficient method is used for remeshing, allowing for good mesh properties in the outer flow regions. A diffusion coefficient of 1.2 is used in this paper, which is rather conservative. The larger amplitude of deflection turbine aerofoil showed a substantially larger lift to drag ratio at low (-5° to 7°) and approximately equal to the rigid at higher angles (7° to 25°). An analysis of deflection length, amplitude and phase was also undertaken. Increasing the deflection length enables better load control, a key issue with VAWTs. The slight phase lag and lead also helped to limit the fluctuation in lift coefficient, with a phase-lag of $\frac{\pi}{8}$ reducing $C_{l,mean}$ by 50% [59]. Liu et al. [60] took inspiration from nature in investigating alternative energy extraction methods. Using a rainbow trout and a hawkmoth for the formation of their turbine blade, they tested the flapping of a rigid, LE control mode, TE control mode and integrated mode. The integrated mode is of most interest in this review, as it seems to drastically reduce performance in comparison to the rigid blade. This seems to be primarily due to the LE control method, as the TE control method appears to increase the peak power considerably. This is likely due to the LE and TE having the same direction of deflection and as such are counteracting each other and causing large leading-edge vortices resulting in flow separation and stall.

Bouzaher et al [26] investigated the influence of using the work of Miao et al. [58], and Liu et al. [60], and applying it to Vertical Axis Tidal Turbines (VATTs) through the application of CFD. They used the spring-based remeshing method for their deforming mesh, with entirely triangular elements, but a y_+ of approximately 1. They confirmed that the use of a flexible aerofoil can increase the power output compared to a rigid aerofoil [26], but only if the flexibility does not become excessive. Additionally, only the trailing edge is deformed and the effect of leading-edge deflection is not considered, which could change the performance drastically. To come to a more robust conclusion of the flexible aerofoils power increase, it was

compared on multiple aerofoils, it was found that the NACA0024 aerofoil had the greatest increase with a average power coefficient increase of 0.13!

Whilst most of the research has been utilising numerical simulations, Wang et al. [61] decided to physically implement the system and perform wind tunnel testing. It was found that after a certain level of deflection, the lift enhancement gets outweighed by the increased frictional resistance [61]. Using a single shaft transmission model, as seen in Figure 2.10, the NACA0012 aerofoil was able to deflect such that “the profile after rotation should be close to the NACA4412 as possible”. It was interesting to note that there was no true comparison between the NACA4412 deflected aerofoil and other experimental NACA4412 data, or numerical data. The only comparison is between their version of the NACA0012 (deflection parameter, $d^* = 0$) and their deflected models. This data however shows that at lower AoA, the lift to drag ratio is comparable between the deformed and undeformed models, until about 18° . All these conclusions should be considered apprehensively however, as the test platform used does not consider the blockage or wall effects, as seen in Figure 2.11.

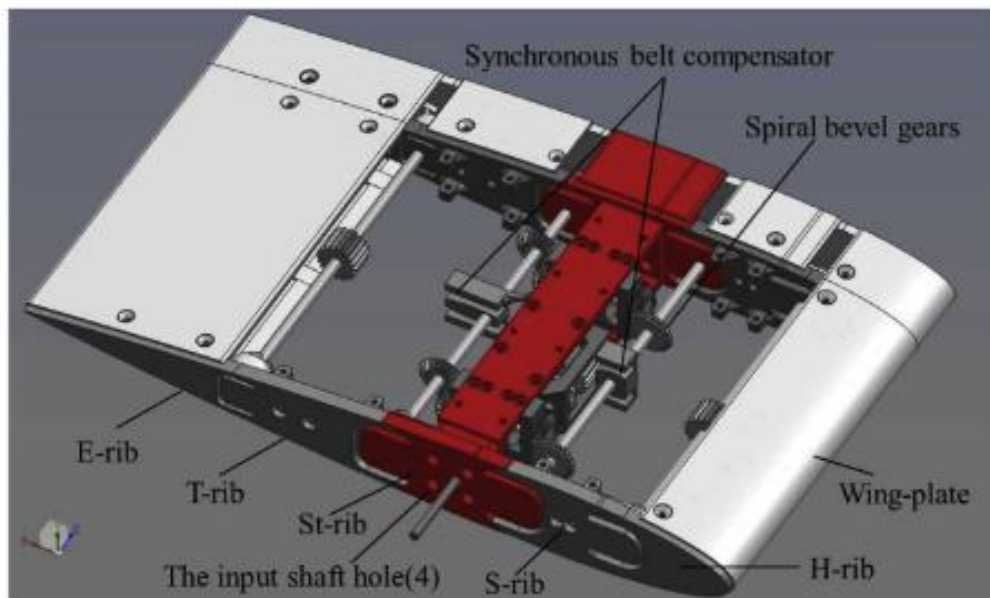


Figure 2-10: CAD Model of an Aerofoil Assembly [61]

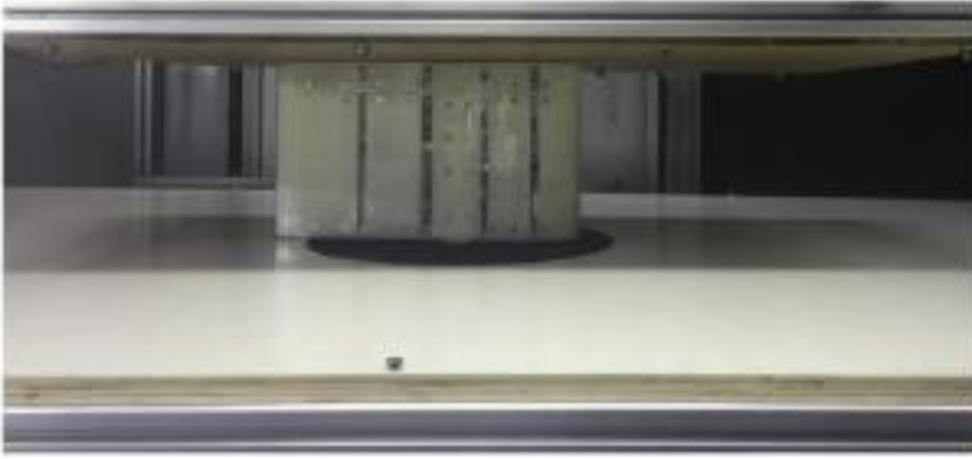


Figure 2-11: CAD Model of an Aerofoil Assembly [61]

2.3 Optimisation Techniques

There are many ways of optimising a design that is employed in engineering, with varying levels of mathematical and computational complexity, with the best suited optimisation method being dictated by the problem at hand. Some techniques that have been employed in the scope of VAWTs include using parametric design studies, adjoint optimisation for turbine blade optimisation, and evolutionary algorithms. A summary of some of these studies is discussed in this subchapter.

2.3.1 Parametric Design Studies

The application of parametric design studies allows for the ability to explore the impact of many design variables in a systematic manner, such as blade characteristics including twist [62] or aerofoil characteristics such as thickness [63], [64] and camber [63]. This also opens the door to the opportunity to create reduced order models of the VAWT profile, such as Mifsud et al. [65] using a mixed fidelity model which is created from a range of high fidelity aerofoil geometry simulations, but allows for simpler models to be implemented to predict between the datapoints, similarly to what was done by LeGresley and Alonso [66]. This technique has also been applied to turbomachinery by Epureanu et al. [67]. Whilst using these reduced order models can assist in speeding up predictions, they do not appear to be broad

enough or refined enough to be applicable to the range of scenarios a single VAWT would experience.

2.3.2 Adjoint Optimisation

Adjoint Optimisation utilises a gradient decent approach to optimise design problems and has been used for the optimisation of VAWT blade designs, as it is very considerate of the complex flow fields that a VAWT blade experiences, resulting in designs unique to a specific turbine's features and operating conditions, typically showing promising results. Day et al. [23] showed that adjoint optimisation can be used at a single snapshot in the turbine's rotation to improve overall turbine power generation, and also at multiple snapshots [68].

2.3.3 Evolutionary Algorithms

Evolutionary algorithms are a set of optimisation algorithms designed to mimic natural selection. They work by mimicking how nature utilises selective reproduction, mutation and recombination to improve a species as a whole, to gradually improve a design until there is no longer any improvements between parents and their offspring. Whilst there are a few variations of evolutionary algorithms, this work focuses on the use of genetic algorithms (GAs) as they allow for a wide breadth of search but strong ability to find the true optimal design for specific operating conditions.

Ferreira and Geurts [69] showed the strength of using genetic algorithm optimisation with a panel methods code, with significant improvements to the lift to drag ratio of the VAWT blade. The work was performed at high tip speed ratios, so the angle of attack is lower, and not reflective of the complex dynamic stall that a VAWT blade experiences at lower angles of attack. Tejeda-del-Cueto et al. [70] optimised the lift coefficient of some aerofoils using an XFOIL based Genetic Algorithm from 0-10° angle of attack and found at least a 10.7% increase in lift coefficient. They noted that XFOIL had an approximately 5% difference in results compared to CFD. Wu et al.

[71] demonstrated the strength of joining the evolutionary algorithm with using a deep neural network surrogate, showing four orders of magnitude improvement of speedup for aerofoil lift-drag prediction.

2.3.4 Machine Learning (ML)

Whilst evolutionary algorithms could be considered a form of AI due to their self regulating methods, they do not train a general model that can be applied to other operating conditions in the same way that machine learning is able to infer trends from massively complex and sparse datasets. This is where machine learning has an opportunity to aid engineers in developing appropriate turbine designs, particularly for deforming VAWT blades. If the machine learning model is trained sufficiently accurately and on a wide enough scale of data, it is theoretically possible to simply place in some operating conditions and estimate exactly what the turbine's performance will be! This could be as complex as considering wind patterns, turbine locations and accessibility and maintenance availability, to something as standard as a user providing specific wind speed, TSR and rotation position. This vast adaptability provides the opportunity to have in depth trends within the coupled effects of design choices too be analysed and exploited without extensive and complex manual statistical analysis. As a result, some work has been done on utilising machine learning and artificial intelligence to optimise and search aerofoils and VAWT blades, with varying levels of applicability.

Ahmed et al. [72] created a CFD surrogate using a machine learning model, but did not provide validation, only statistical analysis of their results. Du et al. [73] created a CFD surrogate for aerofoils which shows very strong agreement to the training data (which consisted of 3000 profiles, with the pressure, temperature and velocity in x and y directions for the grid), comparing the flow fields, pressure coefficients and they also implemented their model into an optimisation method to optimise the lift to drag coefficient of input aerofoils. They utilised the latin hypercube sampling method to generate a design space for mach numbers between 0.3 and 0.6, AoA from $0 \rightarrow 6^\circ$ and Reynold Number from 6×10^6 to 1.2×10^7 . This is a relatively narrow field

for so many training points which would explain the good predictions of the model, but seems like an excessive number of simulations for what is arguably a limited applicability. Sun et al. [74] also created a physics informed neural network (PINN) to act as a surrogate for CFD simulations of aerofoils, and implemented it into a stochastic gradient descent optimiser but found that they often converged on local minima in the optimisation, as the PINN had differing convergence many times. They did utilise a smart usage of adaptive sampling, allowing the PINN to be trained on more points near the desired optimal point improving it's accuracy, but it could be argued that at that point it is not much of a neural network and is effectively just linearly interpolating around points it believes to be optimal. Karbasian and van Rees [75] provided an interesting model that aligns well with the intentions of the thesis; using a surrogate model to optimise morphing aerofoils. Their dataset is calculated using XFOIL to predict the C_l , C_d and C_m characteristics based on a hinge position, and found that often XFOIL fails to converge for large angles of attack, so they discard these datapoints. They tested two datasets; one containing 200 samples and one containing 400 samples, so a relatively small sample size, but as it is a supplement for an FSI approach, it was found to be several orders of magnitude more efficient.

2.4 Summary and Thesis Outline

In summary, this thesis will look at the different methods of generating optimised transiently deforming camber lines of standard VAWT blade profiles.

One of the primary issues with the Vertical Axis Wind Turbines is the impact of the blade-wake interactions and sensitivity to the operational tip speed ratio of the turbine. Some previous investigations have shown that using a cambered aerofoil can improve the overall turbine performance [20], [21], [22], but this is often at the cost of increasing the variance in the turbine moment generation. Furthermore, optimisation techniques, such as the adjoint optimisation, have shown promise in optimising the turbine performance, in single and multiple snapshot techniques [23]. Most of these optimisation techniques however are not necessarily an improvement throughout the turbine's rotation, often resulting in an increase in the performance

at the optimised snapshot position, but a reduction in performance at other positions as a side-effect. The inherently transient nature of the flow angle experienced by a turbine blade profile results in large separation in certain regions of the rotation and the generation of dynamic stall issues [24], [25]. Other investigators have shown that a prescribed deforming blade profile, bending the camber line of the aerofoil with respect to time, has shown an increase in the turbine performance [26]. However, these deformation profiles tend to use arbitrary selections of frequency and amplitude of sinusoidal profiles such as $Camber = A\sin(\omega t)$ and then conclusions are drawn upon the use of these variables, such as the deflection magnitude (A) or the deflection frequency (ω). This leads to the question as to whether these variable combinations are sufficiently broad in their investigative bounds to truly make generalised conclusions on the impact of the deforming turbine profile, or are they simply making broad generalised conclusions? Baghdadi et al. [27] used a freeform deformation algorithm to optimise the aerofoil profile based on the tip speed ratio and azimuthal angle, but had a predefined optimisation constraint of only deforming the final 30%c of the aerofoil, but the actual deformation was completely free, which could alter the thickness of the blade in some regions. They optimised the blade shape every 30°, 60° and 120°, and tested it at 4 tip speed ratios: 1, 1.5, 2 and 2.5, using a polynomial fit to create a smooth curve for the resulting moment plots. They showed a very strong increase in the power coefficient for all the optimisation scenarios. This work builds on the work of Baghdadi et al. [27] through characterising the deforming blade profile, utilising a genetic algorithm to optimise the camber for each optimisation position, and testing the influence of different optimisation constraints for the genetic algorithm and the aerofoil. This will allow for a much better understanding of which aerofoil parameters are providing the largest influence on the deforming blade profile, thus potentially opening further avenues of research.

If optimising a single position in the turbine's rotation can improve the turbine performance using techniques such as the adjoint method, then one can confidently apply this optimisation at multiple positions in the turbine rotation to further

improve the turbine performance. This has been discussed in the literature [27] with the conclusion that applying the multi-snapshot technique to a dynamically deforming blade profile will have further performance improvements. However, using the adjoint or free-form deformation optimisation technique limits the realistic applicability of this dynamic optimisation. It is unrealistic for a turbine to deform its entire shape and thickness dynamically, but some parameters could be relatively easily be controlled such as the thickness or camber magnitude. Whilst the thickness has been shown to change the performance of a VAWT [28], such as increasing the power coefficient at low TSRs with thicker aerofoils, this would be an incredibly complex system to replicate in an actual turbine. This thickness investigation would be very interesting and novel, but does not align with the ethos of this thesis, which is to optimise a dynamic aerofoil profile that could be easily implemented into real world scenarios. By bending the aerofoil, one can increase the curvature of the flow and the lift generation of the aerofoil [29][30]. If the characterisation of the bending shape is that of the generalised NACA 4-series shape equations, then it is relatively straightforward and self-explanatory to convert a typical VAWT blade profile from something such as the NACA0015 by prescribing the deformation centre and magnitude of the deformation to create a highly cambered aerofoil such as a NACA9315. As a result, alternative optimisation techniques would be more appropriate.

The concluded Thesis plan is therefore to initially evaluate the ability of using a genetic algorithm to create an optimised deforming turbine blade camber profile, by optimising the camber at multiple positions in the turbine's rotation and utilising linear interpolation between each azimuthal position to create a dynamic profile based on the turbine blade's azimuthal position. In Chapter 4 this theory is initially evaluated in XFOIL to develop the genetic algorithm workflow for the actual optimisation process and test the impact of varying different GA parameters such as; the number of optimisation points to use in the rotation, the optimisation camber bounds and convergence conditions. These findings are carried forward to Chapter 5 to investigate a transient optimisation procedure, where the turbine is rotated and the blade profile is continuously deformed up to the desired optimisation position,

which incorporates the transient effects of the deformation into the optimisation procedure. This work is evaluated at multiple operating conditions such as sampling count, aerofoil constraints and tip speed ratios to develop an understanding of how turbine and aerofoil parameters influence the camber profile, and the moment generation increase. Finally, to alleviate the computational cost of the transient simulation, but for more fidelity than using XFOIL, utilising an Artificial Neural Network (ANN) as a torque prediction surrogate is tested and applied to the optimal GA conditions discovered in Chapter 5, including using a publicly available ANN model trained on XFOIL data and developing a custom ANN utilising hundreds of static aerofoil CFD simulations to predict tangential force coefficient similarly to that which was established in Chapter 4.

3 Methodology

Many methods are employed throughout the thesis including Computational Fluid Dynamics (CFD) software (primarily ANSYS Fluent), genetic algorithms for camber optimisation and dynamic mesh motion within the CFD process. These techniques will be described in this section, allowing for understanding and appreciation of the various procedures and tools before detailing the different studies into the camber profile optimisation.

3.1 Computational Fluid Dynamics

Throughout this thesis, the primary tool for CFD analysis is ANSYS Fluent, a commercially available package that has been in continuous use and development since the late 20th century, which is a thoroughly verified and trustworthy tool for conducting a wide range of highly complex CFD studies. ANSYS Fluent uses the finite volume method to iteratively “solve” the Navier-Stokes equations, which describe how any fluid moves and interacts with its environment.

3.1.1 Navier Stokes Equations

To study the aerodynamics of any problem, it is fundamentally crucial to understand and appreciate the underlying physics that dictate fluid dynamics. These vary widely depending on the specific area you are studying, ranging all the way from atomic level physics to generalised macroscopic fluid flow.

There are however a few concepts that are uniformly applicable to any type of fluid flow, namely Newton’s Laws and conservation laws. These can be compiled into a set of equations that govern fluid flows. In the scope of inviscid fluids, these equations are referred to as the Euler equations, but when applied to viscous fluids the Navier-Stokes equations are used, named after the French physicist Claude-Louis Navier who first defined them, and the Irish physicist George Gabriel Stokes who first explained their significance. The Navier-Stokes equations are based on Newton’s second law of motion but applied to fluid motion with viscous effects included, with the

conservation of mass and momentum included. Often, the Navier-Stokes equations are used in conjunction with other supplementary formulae such as energy equations, but this can be negated in systems where no work is being done to or taken from the fluid volume being studied.

The mass continuity equation of the Navier-Stokes equations is given as;

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0 \quad 3.1$$

The conservation of momentum is given as;

$$\rho \left[\frac{\partial \vec{V}}{\partial t} + \vec{V} \cdot \nabla \vec{V} \right] = -\nabla p + \nabla \cdot \boldsymbol{\tau}_{ij} + \vec{F}_b \quad 3.2$$

$$\mathbf{F} + \rho \mathbf{g} - \nabla p + \nabla \cdot \boldsymbol{\tau}_{ij} = \rho \frac{d\mathbf{V}}{dt} \quad 3.3$$

$$\boldsymbol{\tau}_{ij} = \begin{bmatrix} \tau_{xx} & \tau_{yx} & \tau_{zx} \\ \tau_{xy} & \tau_{yy} & \tau_{zy} \\ \tau_{xz} & \tau_{yz} & \tau_{zz} \end{bmatrix} \quad 3.4$$

$$\mathbf{V} = \frac{\partial \mathbf{V}}{\partial t} + u \frac{\partial \mathbf{V}}{\partial x} + v \frac{\partial \mathbf{V}}{\partial y} + w \frac{\partial \mathbf{V}}{\partial z} \quad 3.5$$

This essentially means the External body forces + gravity force per unit volume + viscous force per unit volume = density * acceleration of the fluid volume [16].

We can break the Navier-Stokes equation down into it's expanded parts, which allows us to understand how each term affects physical flow properties and phenomena. If we expand in the x direction, our equation would become:

$$\rho \frac{\partial u}{\partial t} + \rho \left(u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} \right) = -\frac{\partial P}{\partial x} + \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} + \rho f_x \quad 3.6$$

[1]
[2]
[3]
[4]
[5]

1. Refers to transient acceleration of the fluid flow
2. The inertia term, describing the inertia of the fluid, or the acceleration due to movement from one location to another

3. The source/sink term, the driving force behind the flow, providing the energy for movement.
4. Diffusion Terms. These terms refer to the viscous stresses within the fluid element and are the effect of viscosity, shearing and changing the volume of the element.
5. f_x refers to the body forces per unit volume that can act on a fluid element. These could include gravitational effects, magnetics etc.

It is important to clarify what the viscous stress tensor physically means, however.

Implementing Newton's law of viscosity (Equation to our stress tensor and adding the following assumptions made by Stokes;

$$\tau = \mu \frac{\partial u}{\partial y} \quad 3.7$$

1. Newton's law of viscosity is followed meaning stress is linearly proportional to the velocity gradient and viscous stress is linearly proportional to the strain rate
2. The fluid is isotropic (has the same fluid property in all directions)
3. The viscous force only acts tangentially to the fluid element, it's normal is null.
4. Rigid body rotation of the fluid element doesn't provide any source of viscous stress.

The resulting stress tensor is therefore given as

$$\boldsymbol{\tau}_{ij} = \begin{bmatrix} \tau_{xx} & \tau_{yx} & \tau_{zx} \\ \tau_{xy} & \tau_{yy} & \tau_{zy} \\ \tau_{xz} & \tau_{yz} & \tau_{zz} \end{bmatrix} = \begin{bmatrix} 2\mu \frac{\partial u}{\partial x} & \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) & \mu \left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right) \\ \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) & 2\mu \frac{\partial v}{\partial y} & \mu \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) \\ \mu \left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right) & \mu \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) & 2\mu \frac{\partial w}{\partial z} \end{bmatrix} \quad 3.8$$

Or in a simplified manner;

$$\tau_{ij} = 2\mu \left[\varepsilon_{ij} - \frac{1}{3} \nabla \cdot \vec{V} \delta_{ij} \right] \quad 3.9$$

$$\varepsilon_{ij} = \frac{1}{2} \left[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right] \quad 3.10$$

Where μ is the fluid's dynamic viscosity and δ_{ij} is the Kronecker delta. Implementing this definition of the stress tensor, the convective or non-conservative form of the Navier-Stokes is given as;

$$\rho \left[\frac{\partial \vec{V}}{\partial t} + \vec{V} \cdot \nabla \vec{V} \right] = -\nabla p + \mu \nabla^2 \vec{V} + \frac{1}{3} \mu \nabla (\nabla \cdot \vec{V}) + \rho \vec{g} + \vec{F}_b \quad 3.11$$

The non-conservative and conservative versions of the Navier-Stokes momentum equation are only equivalent for “smooth” flows, but if discontinuities are present such as shocks, the non-conservative version is invalid, and in such scenarios, the conservative version must be used [76].

When applied to slow moving external aerodynamics (such as that experienced by a VAWT) we can simplify the Navier-Stokes Equations. Due to the low (subsonic, $M < 0.3$) speed and fluid properties of air (Newtonian Fluid), the flow field is assumed to have a constant density and viscosity and allows the negation of all external body forces including gravity.

Sometimes it is necessary to also include the conservation of energy and species when analysing fluid flows. Scenarios where this might be the case include supersonic flows, heat transfer and combustion scenarios. Heat transfer doesn't always require species conservation due to a lack of chemical reactions, but very high-speed flows (hypersonic) can cause chemical breakdown of fluids such as air, and as such it is necessary to maintain species continuity. These can both be negated for the consideration of flow fields impacted by VAWTs, unless complex devices are being investigated to influence the flow field such as icing on blades.

$$\rho \left[\frac{\partial h}{\partial t} + \nabla \cdot (h \vec{V}) \right] = -\frac{Dp}{Dt} + \nabla \cdot (k \nabla T) + \phi \quad 3.12$$

[1] [2] [3] [4] [5]

Again, each section of the equation explains a different source of energy in the fluid.

1. Enthalpy change with respect to time
2. Convection
3. Pressure Work
4. Diffusion of energy
5. Dissipation of energy

3.1.2 Boundary Conditions

The Navier-Stokes equations can only be solved for a finite number of situations, such as Stokes or creep flow, which is incredibly low Reynolds number flow, or Couette flow, inviscid flow through a pipe. But these cases, and any other case, require the implementation of boundary conditions, on the edge of our flow volume. These vary significantly but in the scope of VAWTs, the most relevant include:

1. Inlets
 - a. Velocity Inlets
 - b. Pressure Far fields
 - c. Mass flow inlets
2. Outlets
 - a. Pressure Outlet
 - b. Mass flow outlets
3. Symmetry
4. Walls
 - a. Free shear
 - b. No-slip
5. Interfaces

Most commonly it is found to have a velocity inlet as the inlet boundary condition for a VAWT CFD flow domain, as it allows for the wind speed that the turbine will experience to be defined. Pressure outlets are typically used to “vent” the flow

domain to atmospheric conditions, which can prevent creating a pressure differential at the outlet which would cause a flow acceleration in the domain. Symmetry is often used at the outer faces of the flow domain, or on symmetry planes of 3D simulations (such as at a blade midspan) as it minimises the blockage effects of the wind turbine in the domain, allowing for smaller domains to be used than if walls were applied. No-slip walls are assigned to the components of the VAWT such as the blades and the shaft to model the boundary layer generation on these surfaces as they would in real life. Finally, an interface pair is created between the rotating mesh region containing the turbine geometry and the static flow domain. There are two primary methods implemented; conformal and non-conformal interfaces. Conformal interfaces have a matching mesh either side of the interface, so the vertices on the edge of each respective fluid domain's interface are connected and share topology. Non-conformal interfaces are when the two fluid domains have different mesh resolutions on their respective interfaces, or they do not share topology, for example the interface of one domain could have 100 nodes, but the other domain could have 200 nodes, or if one of the domains moves w.r.t the other domain. All interfaces share one key feature; they conserve the flux of mass, momentum and energy across the boundaries. In the VAWT CFD, a non-conformal interface is applied to the common edge between the flow domain and the rotating turbine domain. To maintain the flux conservation, Fluent computes the flux across the interface and interpolates the values for each overlapping cell on the domain, by calculating the area-weighted average from one side, it can then interpolate what the appropriate flux will be on the opposite domain's cells each time-step.

3.1.3 Turbulence Modelling

When studying these fluid flows there are a finite count of situations where they can be analytically solved [77], such as Stokes flow which is incredibly low Reynolds number flows or inviscid flow through a pipe. Other more complex flow scenarios that are more common in engineering research and development require the use of CFD codes to investigate and study flow fields. Due to the nature of the Navier-Stokes

equations, it is necessary to use some form of discretisation with carefully selected boundary conditions to model complex flow phenomena such as turbulence. There are three primary methods of modelling the Navier-Stokes equations in order of increasing flow resolution;

1. Reynolds Averaged Navier-Stokes
2. Large Eddy Simulation
3. Direct Numerical Simulation

Whilst the resolution of the flow decreases, the computational requirements also rapidly decrease. Modern advances in computing ability allow for simulation of the flow fields around complex bodies such as aircraft from a device as simple as a laptop! Of course, this comes with the caveat that it is necessary to utilise lower resolution models but is regardless an impressive feat considering the complex computing power that was necessary a few decades ago. The computationally cheapest method, Reynolds Averaged Navier-Stokes, uses Reynolds averaging to split the flow parameters into two terms, their absolute value and their fluctuations due to turbulence. These fluctuations are defined as the deviation of a variable, in this example u , from their average value [16];

$$\bar{u} = \frac{1}{T} \int_0^T u \, dt \quad 3.13$$

Therefore

$$u = \bar{u} + u' \quad 3.14$$

It is mathematically clear that the mean of u' has zero value, but this is not true of the mean square of the fluctuation, u'^2 , is not zero, and can be used as a measure of turbulent intensity [16]. Oscar Reynolds used this idea to split all the variables in the Navier-Stokes equations, to produced time-averaged versions, replacing velocity and pressure components with their averaged counter parts i.e.

$$u = \bar{u} + u' \quad 3.15$$

$$v = \bar{v} + v' \quad 3.16$$

$$w = \bar{w} + w' \quad 3.17$$

$$p = \bar{p} + p' \quad 3.18$$

As a result, however, there are some terms left that cannot be resolved and require an empirical formula or alternative method of calculation.

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_i u_j) = -\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j} \left[\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \delta_{ij} \frac{u_k}{x_k} \right) \right] + \frac{\partial}{\partial x_j} (-\rho \overline{u'_i u'_j}) \quad 3.19$$

These turbulent based stresses $(-\rho \overline{u'_i u'_j})$ are referred to as Reynolds Stresses and create a closure problem within the Reynolds Averaged Navier-Stokes equations. The most common method of calculating the Reynolds stresses is through the application of a turbulence model. Just like the fact that there are differing methods of solving the Navier-Stokes equations with varying levels of computational complexity, there are differing levels of resolution for formulating the Reynolds stresses. Each turbulence model has particular use cases that they were designed for, and as a result have different formulations.

The most common approach to closing the Reynolds Averaged Navier-Stokes equations is to use the Boussinesq Approximation. This method links the mean velocity gradient to create a new viscosity for eddies, the eddy viscosity. One of the primary drawbacks of this method is that the eddies are considered isotropic, this is not the case for the Reynolds Stress Model (RSM) that is included in some commercial solvers such as ANSYS Fluent, which solves the transport equation for the eddies directly, with an equation for dissipation of the eddies [78]. The formulation of the Reynolds Stresses when using the Boussinesq Approximation is

$$-\rho \overline{u'_i u'_j} = \mu_t \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \left(\rho k + \mu_t \frac{\partial u_k}{\partial x_k} \right) \delta_{ij} \quad 3.20$$

Where the different turbulent models will have different methods of calculating k and μ_t . The most commonly employed in VAWTs are the $k - \omega$ SST and $k - \epsilon$ models due to their ability to model the freestream turbulence well, but k-w is preferred for its ability to capture complex flow phenomena such as boundary layer separation and vortex shedding [79]. The full derivation of the mathematical definition of these turbulence models can be found in common literature, and is left as an exercise to the reader.

3.2 CFD Setup and Validation

To validate the results produced by the various optimisation techniques within this thesis, the deformation profile will be placed within a CFD simulation and compared to a symmetric NACA0015 aerofoil. The turbine used for comparison is the same as that used by El Sakka et al.[80] and Li et al.[81], with the latter study testing the turbine experimentally at three TSRs. The turbine parameters are selected to match the published data, corresponding to a TSR of 2.29, freestream velocity of 7 m/s, turbine radius of 0.85m, blade chord length of 0.225m with the leading edge shifted forward of the shaft center by 25% chord. For the validation studies, the blade has a setting angle of 6° as is found in the work of Li et al. [81].

3.2.1 Geometry

A two-dimensional simulation is setup using the domain arrangement seen in Figure 3-1. The C-Grid far field technique is used to optimise and control the mesh structure as it requires less elements than a rectangular domain would for a given mesh density. A sufficiently large domain is crucial to minimise wall effects affecting the flow solution and capturing the flow field adequately. Due to the large domain, a structured mesh is created to optimise the allocation of elements, ensuring they are in regions of flow where high fidelity is desired, including the boundary layers, the turbine wake and the turbine core.

A series of domain sizes were tested and validated against the literature (the work of Elsakka et al. [80]) by comparing the average tangential force coefficient. The

different comparisons for the domain sizing are listed in Table 3-1 below. Domain D was the final design consideration, as it provided the most stable simulation results (consistent max residuals target of $1e-5$ achieved each timestep) and also when running the full turbine simulation, it was found to match the literature results most accurately.

Property	A	B	C	D
Downstream Length (m)	5.325	11.7	8.5	12.55
Perpendicular domain Thickness (m)	5.1	8.5	8.5	8.5
Aerofoil O-Grid Size (m)	0.3375	0.255	0.3	0.3
Rotating Domain Size (m)	2.1	2.1	1.9	2.5

Table 3-1: Domain sizing comparisons

The final geometrical sizes include 10 turbine radii downstream (8.5m), and an inlet radius of 5 radii (4.25m). The wake region is sized to be 0.75 diameters (1.19m) ahead of and above the turbine domain and 1 diameter aft of the turbine (1.7m).

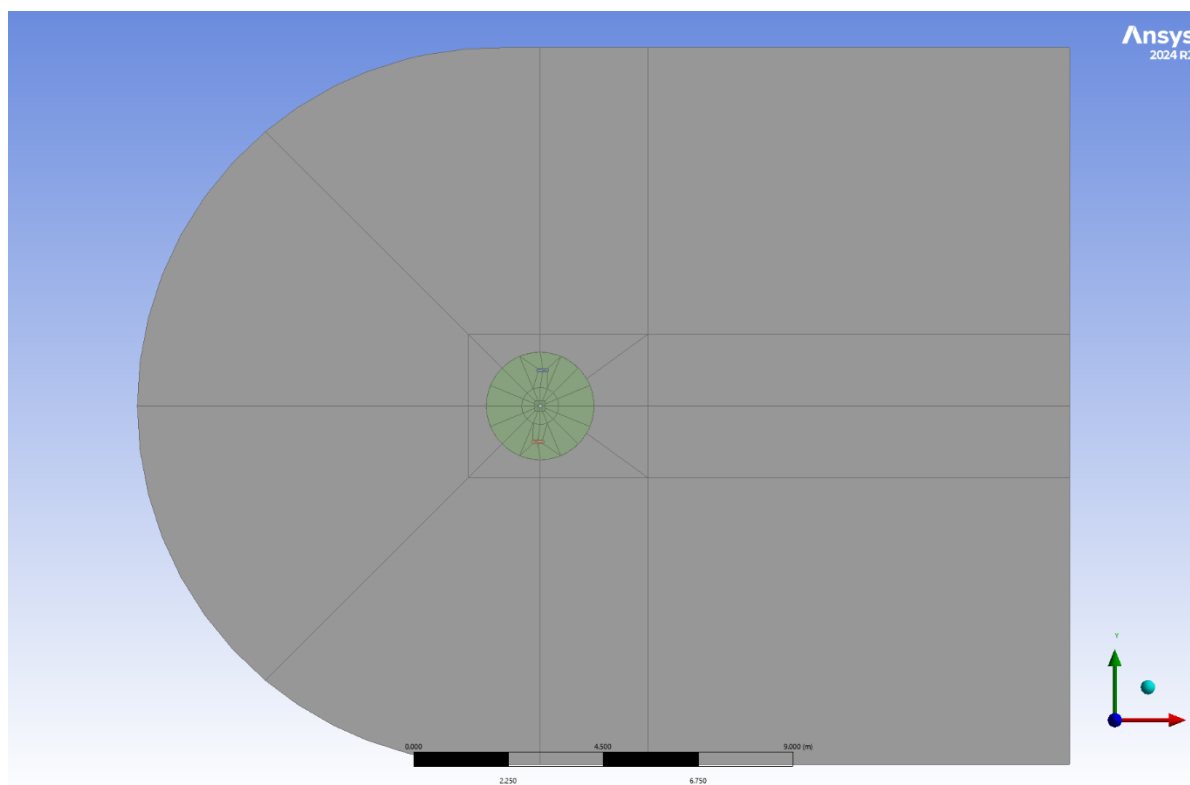


Figure 3-1: Image of entire flow domain geometry

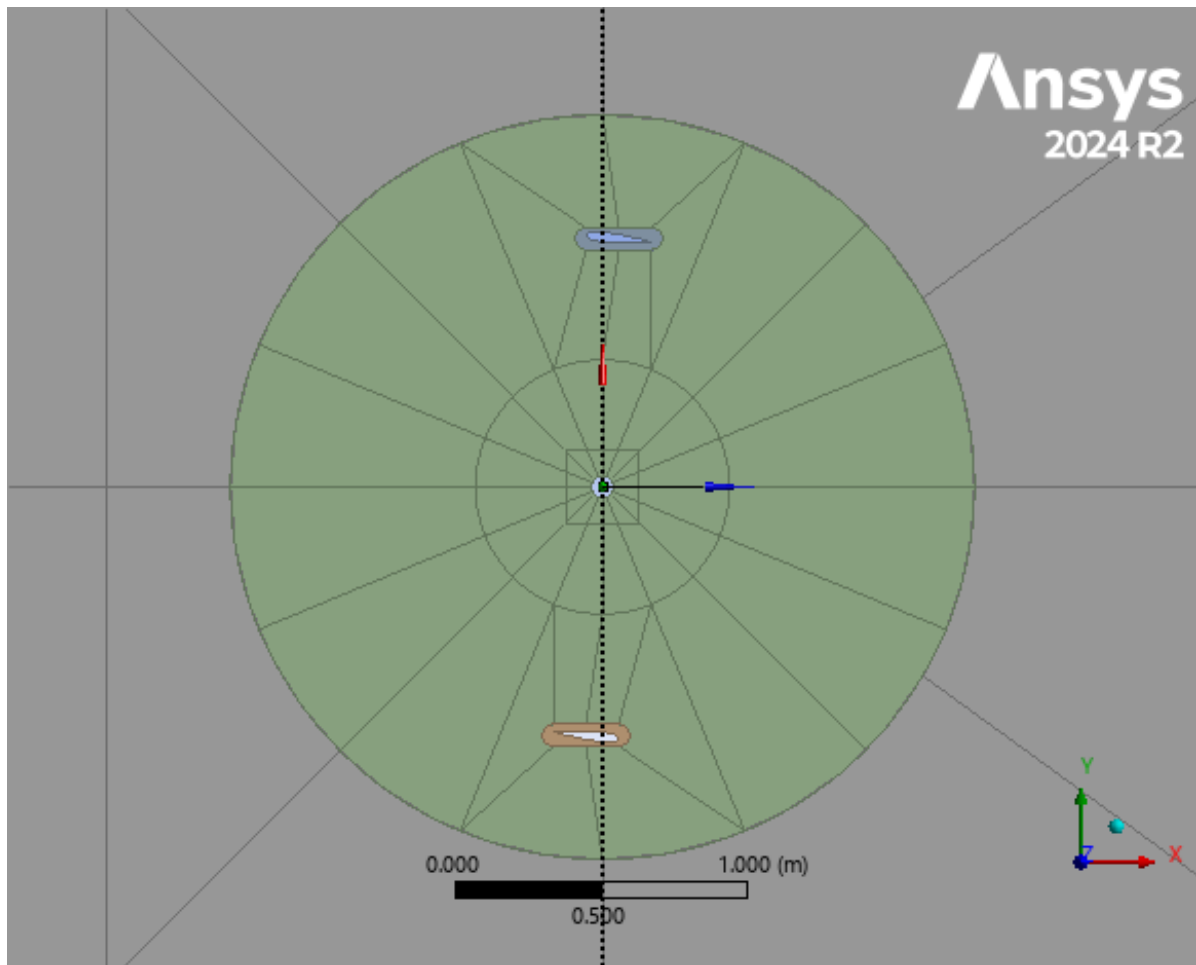


Figure 3-2: Zoomed image of geometry around fluid turbine fluid zone

The solver is the ANSYS Fluent software with the double precision solver. The COUPLED method is employed to ensure good convergence behaviour and help with any divergence that may be experienced. Furthermore, all solver methods are set to Second Order Upwind to ensure solution stability. The Unsteady Reynolds-Averaged Navier Stokes approach is undertaken to reduce the computational costs, but with good acceptance of their applicability in the application of VAWT simulations. The $k-\omega$ SST turbulence model is employed due to its design for capturing separated flow more accurately than most other turbulence models and resistance to adverse pressure gradients [82], as is typically present in stalled vertical axis wind turbine blades.

3.2.2 Mesh Independence Study

To validate the mesh chosen for the design, and ensure the mesh accurately captures the flow physics, whilst minimising element count for computational efficiency, a mesh independence study is undertaken.

The mesh design was altered by changing the element dimensional sizing in the near blade flow region, as this alters the number of nodes along the aerofoil surface and reduces/increases the number of elements for capturing the blade wake. The inflation layer settings are kept constant to ensure an appropriate y^+ is maintained in the 5 studies, ensuring confidence that the difference in results is purely due to blade-wake resolution. Furthermore, if the inflation layer settings were adjusted (number of layers or first cell height), then the turbulence model employed, k-omega SST, might underpredict the boundary layer effects for a coarse inflation due to the lack of wall functions [83] or too fine and it might employ the k-epsilon model too close to the wall rather than the k-omega model as the boundary layer isn't fully resolved by the inflation. The change between each mesh was simply to alter the body sizing of the mesh size in the o-grid region around the turbine blade profile. The different sizing are listed in Table 3-1.

Element Count	425	625	774	1100	1900
O-Grid Element Size (m)	6e-4	5e-4	4e-4	3e-4	2e-4

Table 3-2: O-Grid Sizing Comparison for Mesh Independence Study

Figure 3-3 shows that the overall capture of the moment for the VAWT is very similar between all mesh resolutions, with the main differences in the peak value and the variance seen in the downstream period of rotation ($\phi > 180^\circ$). Taking a closer look at the averaged moment value however shows that there is significant variation between the different meshes. A mesh with 1.1 million elements is less than 1%

different to 1.9 million elements, but with substantially faster computation time (wall-clock time ~17 hours versus ~29 hours respectively with 96 CPU Cores). The mesh with 1.1 million elements is therefore selected as the optimal mesh for further CFD cases, including validation against previous literature and genetic algorithm evaluation.

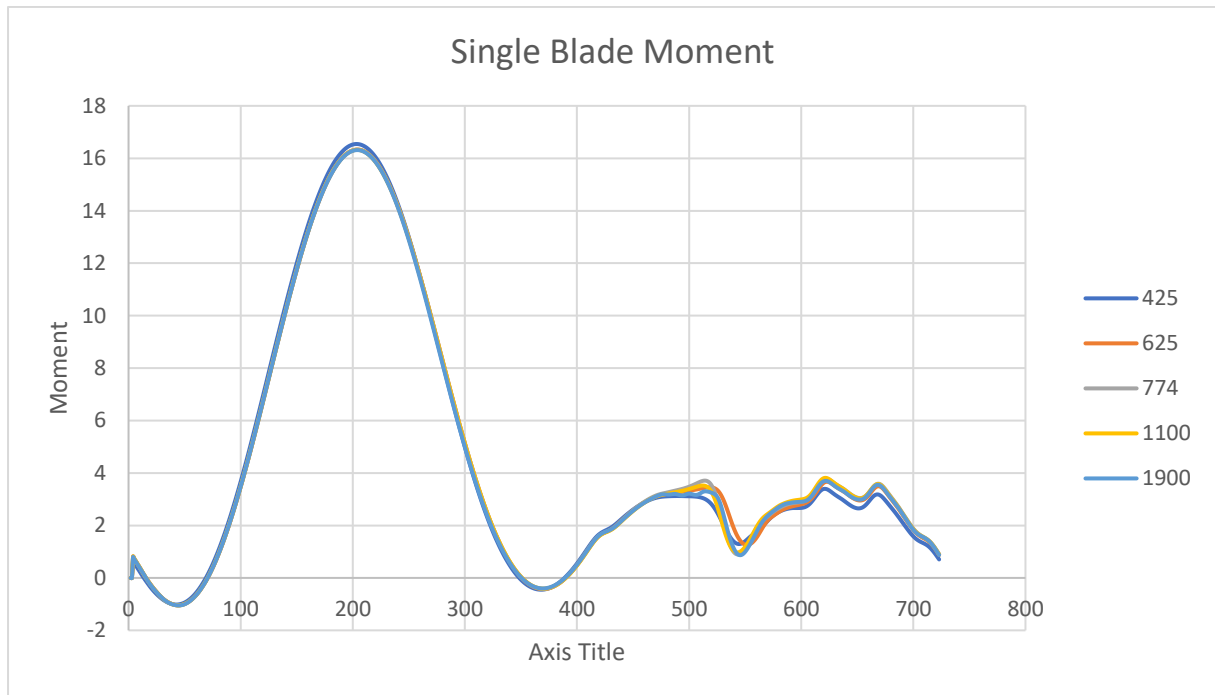


Figure 3-3: Instantaneous Moment Comparison at Multiple Mesh Sizes

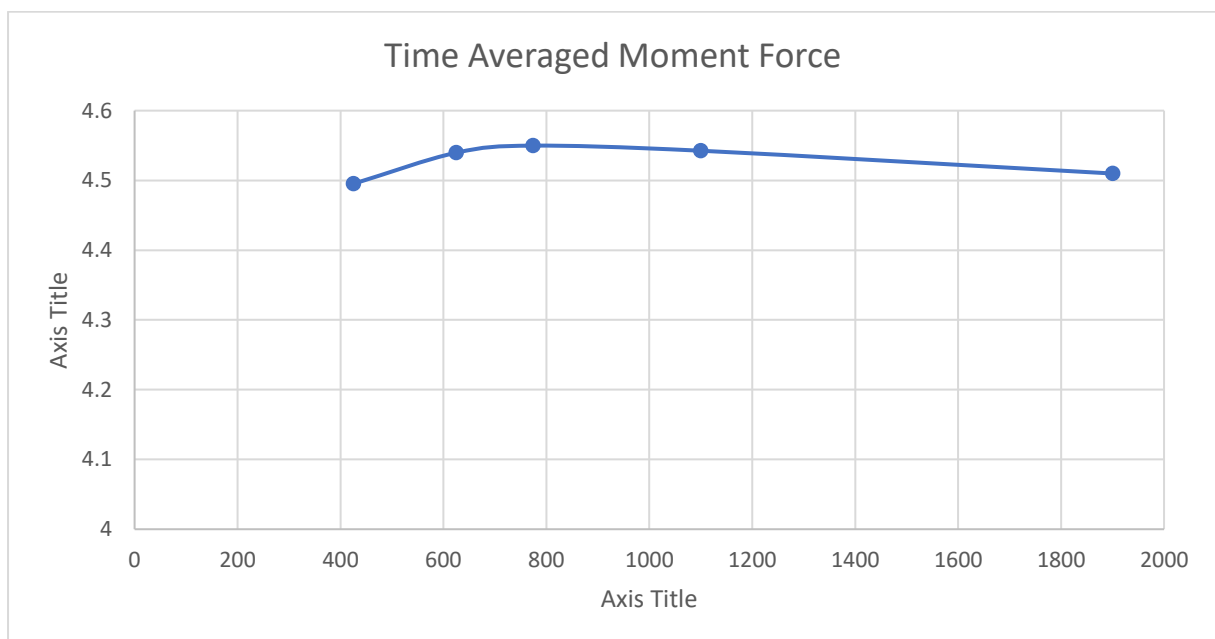


Figure 3-4: Mean Moment averaged over period of rotation

3.2.3 Final Mesh Characteristics

A $y^+ \approx 1$ is used in the mesh as is required by the $k-\omega$ SST model, with no wall functions. This y^+ value is achieved by the application of inflation layers at the aerofoil and shaft surface, with a first layer height of $1.94e-5m$, 30 layers and a growth rate of 1.1. The final mesh count is 360k for each of the O-Grid regions encapsulating the aerofoils, 264.8k for the remainder of the rotating turbine area, and 125k for the rest of the stationary fluid far-field region, creating a total mesh size of 1.1098M elements.

To enable the application of the deforming mesh motion, it is better to utilise an unstructured mesh in the near aerofoil flow region, particularly in the vicinity of the trailing edge due to the large deformation and step in geometry experienced in these regions by the application of a flapping mesh motion. This mesh sizing is controlled by applying the inflation layer mesh control and specifying a face element sizing within ANSYS' meshing tool. Additionally, the number of elements along the span of the aerofoil wall is controlled to ensure a smooth transition between the quadrilateral inflationary boundary layer and the triangular elements. The resultant overall flow discretisation and near-blade regions can be seen in Figure 3-5, Figure 3-6 and Figure

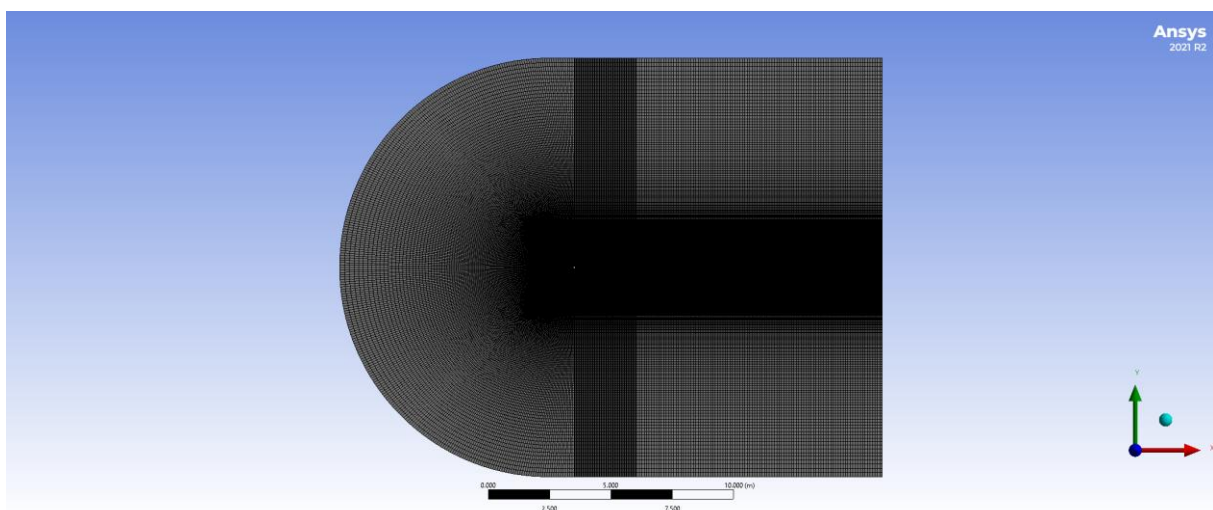


Figure 3-5: Entire Flow Domain Mesh

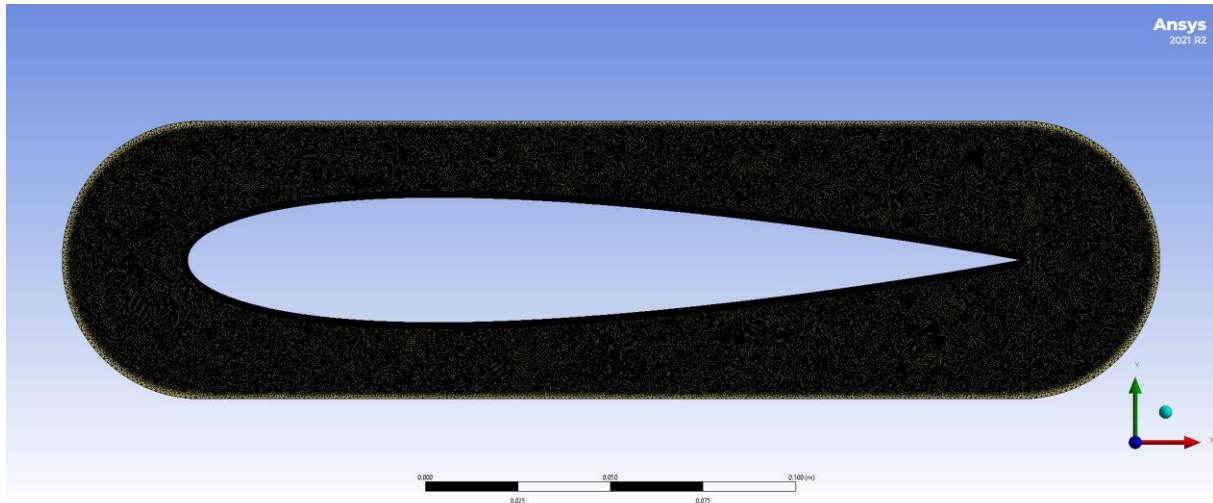


Figure 3-6: Near Blade Mesh

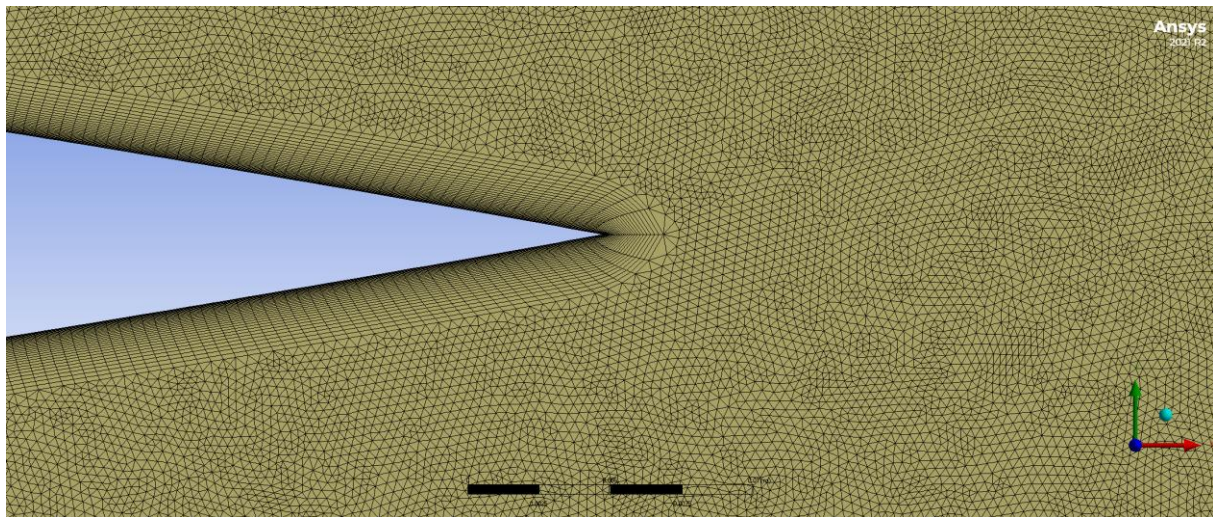


Figure 3-7: Near Blade Mesh at Sharp Trailing Edge

The flow domain boundary conditions are highlighted in Figure 3-8. A velocity inlet is setup, with symmetry walls at the outer edges (yellow lines) and a pressure outlet as seen in Figure 3-8. The inlet velocity is set to 7m/s in the X direction, 0m/s in the Y direction and 0m/s in the Z direction (seen in blue) to match the work of Li et al.[81]. Furthermore, a no-slip wall condition is employed on the turbine blade edges and on the central shaft (seen in grey), with a defined rotational speed of 18.86 rad/s applied to the turbine fluid domain by application of mesh motion (corresponding to a TSR of 2.29, as seen in the work of Li et al.[81] and El Sakka et al.[80]) and a corresponding timestep size of 0.0004238656755s (1° per timestep). In addition, there is a mesh interface created between the turbine flow domain and the far field domain (shown

as the yellow circle in the domain centre) to ensure accurate flow physics by using the sliding mesh model. A symmetry wall boundary condition is applied to the top and bottom domain edges (yellow) to encapsulate the flow and reduce the need for a much larger domain required for the no-slip wall conditions. A pressure outlet is used with atmospheric conditions, seen in red at the rear of the flow domain.

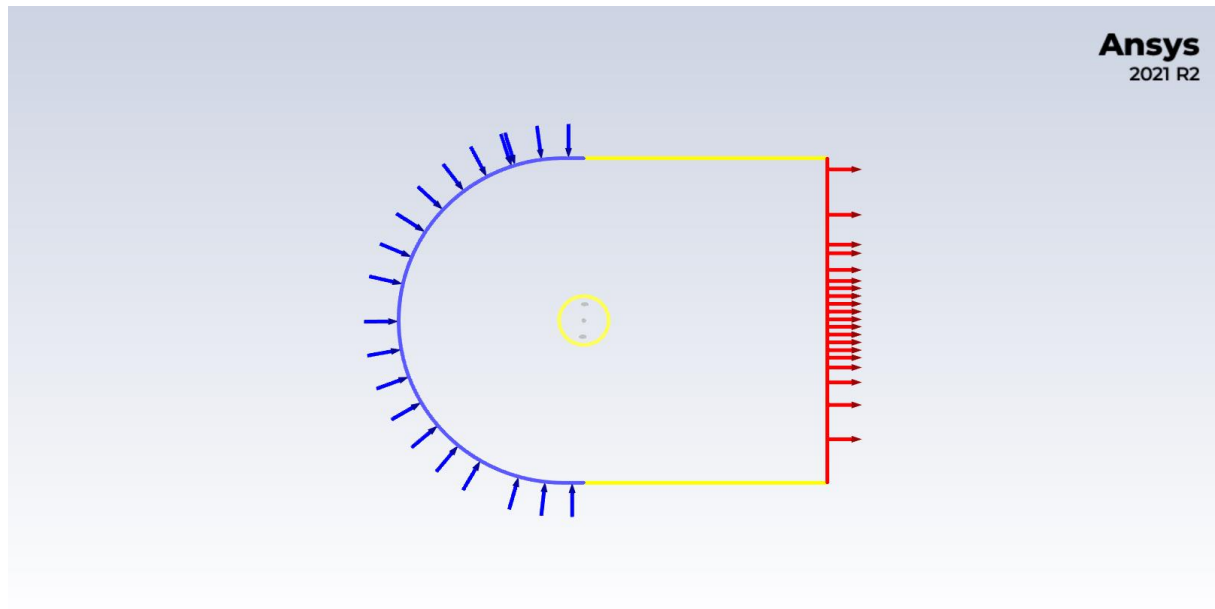


Figure 3-8: Boundary Conditions Highlighted on Flow Domain Edges

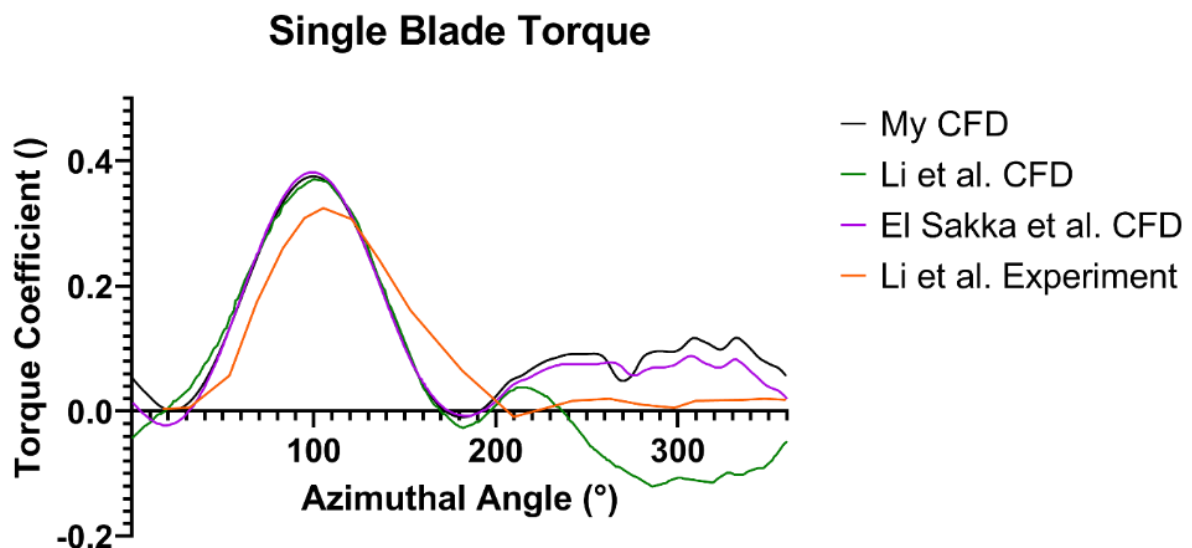


Figure 3-9: Single Blade Torque Coefficient Comparison with Li et al.[81] and El Sakka et al.[80] works

It is clear from Figure 3-9 that there is strong agreement in the CFD setup results with the upstream torque performance compared with the experimental work of Li et

al[81]. The difference in the downstream between this work's CFD study and the experimental data can be due to a variety of factors, including the lack of tip vortices affecting the performance that are not present in the 2D simulations (this work and El Sakka et al. [80] but are in the experiment and 3D simulations (Li et al.[81]).

This CFD setup shows good agreement in the upstream region ($0^\circ < \phi < 180^\circ$) with the CFD work of Li et al.[81] and El Sakka et al.[80], and also good similarity with the downstream region of El Sakka et al.[80]. It is therefore considered suitable for investigating the turbine performance improvement by application of GA optimisation.

3.2.4 Timestep Independence Study

To understand the impact of temporal resolution on the accuracy of the CFD model, the simulation is run with three different timestep sizes; 1° of rotation per timestep, 0.5° of rotation per timestep and 0.25° of rotation per timestep. The resultant torque vs azimuthal angle is presented in Figure 3-10 for a 0 degree setting angle of the turbine parameters described in Section 3.2, with the green line representing the 1° case, the blue line is the 0.5° case and the orange line is the 0.25° case. The overall torque behaviour is captured well for all three setups, with negligible differences in the initial period of rotation from 0° to 180° . The main differences start to appear around the peak, where the 0.5° degree has the highest single blade torque, but the downstream period of rotation is very similar between all 3 cases. The variance between the temporal resolutions clearly starts at the first large peak around 90° azimuth, I suspect this is due to the difference in resolutions predict a different angle at which the leading edge vortex sheds and the blade stalls, this would also then have significant impact at around 144° again as the blade rotates, cutting through the previously shed vortex that was carried downwind, again with a minor difference in the position in rotation causing a small butterfly effect and exacerbating the difference in turbine power generations due to coupling with the blade's lift and drag generation at the different angles of attack. For example, if we take the large peak position, the angle of attack is the highest in magnitude that the aerofoil experiences

and the aerofoil performance would be very sensitive to any changes in the flow conditions. By shedding the vortex a half degree earlier in rotation, the small change in angle of attack could result in the flow remaining attached that tiny bit more resulting in a slightly higher peak.

As this case will be later utilised for transient GA optimisation, the computational cost will be directly associated with the time step size, and as such the minor difference in turbine performance differences are acceptable for the large improvement in simulation wall clock time.

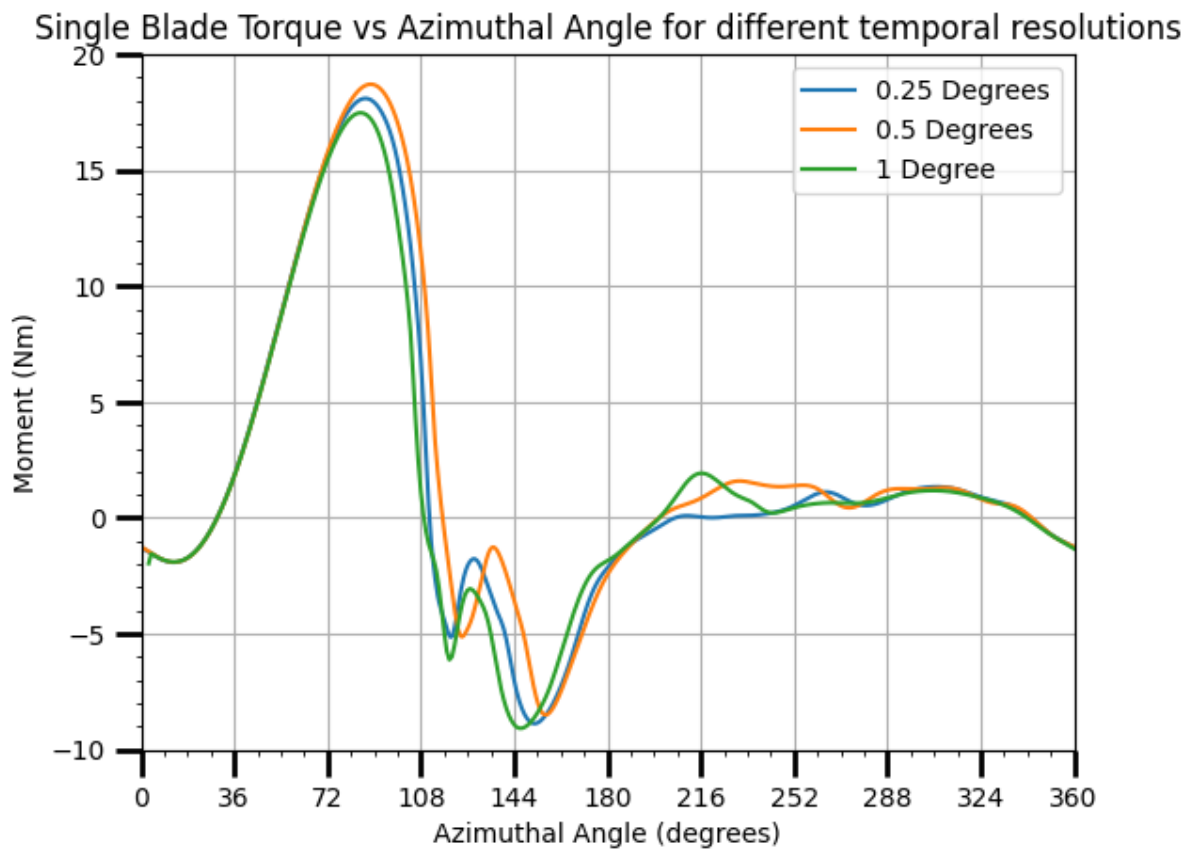


Figure 3-10: Comparison of Blade Torque for 1°, 0.5° and 0.25° of Rotation per Timestep.

3.3 Deforming VAWT Methodology

It is crucial to evaluate the performance of how the optimised VAWT blade camber profiles perform in the scope of a full rotating VAWT to understand the impacts on dynamic flow effects such as dynamic stall, leading edge separation bubbles and

blade wake interactions. This was achieved by implementing the camber profile into a complex UDF in ANSYS Fluent that will deform the turbine blade aerofoil profile based on its azimuthal position and operating characteristics. The UDF and an example setup of the parametrised CFD simulation that is detailed below can be found on GitHub at [84].

As stated in Section 4.1.3.1, the optimisation algorithm is applied at a 0° setting angle to reduce the modelling complexity of the flapping profile on a 6° setting angle. Additionally, as this work intends to solely explore the impact of the GA on the flapping motion, it is crucial to attempt to eliminate any coupled effects that may be presented by the application of a flapping turbine with a 6° setting angle. By having the 6° setting angle included, there could be unforeseen impacts on the flow physics, resulting in our conclusions potentially being misplaced due to exacerbation or even damping of the flapping profile's impact.

A user defined function (UDF) is implemented to deform the aerofoil profile to the maximum camber profile determined in the CFD analysis. A series of linear interpolation functions were created between each optimisation point and the mathematical representation of this line created an equation for the relation of the camber against azimuthal position. Linear interpolation was utilised instead of complex polynomial curve fitting as curve fitting would be very computationally complex and inefficient within the CFD process, significantly increasing the time taken per timestep. Additionally, a curve would need to be recalculated for every potential genotype, which would require an incredibly robust polynomial to ensure that all possible phenotypes would not be overfitted, nor would they be under fitted. This is an unnecessary complexity in the scheme of this thesis, but could be an interesting point of research in future works. These lines can be used to interpolate the camber at any specified azimuthal position, allowing for easy adaptation to other turbine optimisation scenarios.

For example, if at $\theta = 36^\circ$ the camber value is given as $5\%c$, the UDF will calculate the required deflection between the current timestep's shape (based on the camber) and the next timestep's shape (based on the camber value at the next step). NACA

equations for a 4-digit cambered aerofoil with a maximum camber position of (NACA $XX15$) and calculate the local y and x values for the nodes as if the aerofoil leading edge x value is 0, then translate them back into the global reference frame. This is a necessary requirement to use the NACA 4-digit modelling method, as the leading edge is defined as $x=0$. This is repeated for every node along the aerofoil blade surface. A graphical representation of the parameters required to calculate the node positions is seen in Figure 3-11.

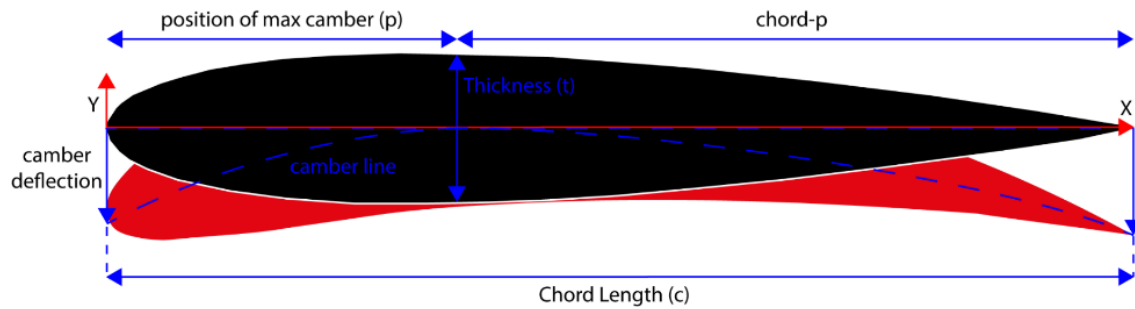


Figure 3-11: Visual representation of Deflection Model

The black aerofoil is the standard NACA0015 aerofoil profile, with the X and Y coordinates at any point on the boundary described by the NACA 4-Series set of equations described in Section 1.3.1.

To calculate the deflected aerofoil (the red aerofoil in Figure 3-11) the UDF will take the X and Y coordinates of the current aerofoil, transpose them onto the local XY frame for the NACA 4 Series (where the XY value at the leading edge would be $x=0$, $y=0$), calculate the required translation of every node on the aerofoil surface to the next timestep position, add that vertical translation (calculated with Equations 3.21 and 3.22) to each node and transpose the aerofoil back to the correct azimuthal position.

$$m = m_{\phi} - m_{\phi-1} \quad 3.21$$

$$dy_c = \begin{cases} -m\bar{c} \left(\frac{p\bar{c} - x}{p\bar{c}} \right)^2 & x < p\bar{c} \\ -m\bar{c} \left(\frac{x - p\bar{c}}{c - p\bar{c}} \right)^2 & x \geq p\bar{c} \end{cases} \quad 3.22$$

$$y_c(\phi) = y_{c(x)} - dy_c \quad 3.23$$

Where $y_c(\phi)$ is the updated co-ordinate position of the node, dy_c is the vertical displacement of the node, m is the camber value, $\bar{c}$ is the chord length, m_ϕ is a function of the azimuthal angle.

There is no change in the x-coordinate of each node, because whilst it is shown to change in the original NACA 4-Series equations, the difference is negligible as we are not calculating the deformation by shifting every node along the blade surface for the curvature. For example, if you have a symmetrical aerofoil, the leading edge node would be at exactly $x=0$, but when you take a positively cambered aerofoil, that exact leading edge shifts slightly positively, however, the next node along the aerofoil upper surface would be in approximately the same x position as the previously leading edge. This very minor difference is negligible in the shape, but implementing it into the UDF would increase complexity significantly and would affect skewness and even generate negative cell volumes of the first wall elements on the aerofoil surface.

Each optimised position i.e., $\phi = 36^\circ$, has a corresponding camber value associated, as such, linear interpolation is used to calculate the instantaneous camber value between each optimised position. For example, if at $\phi = 0$, $m = 36$ and at $\phi = 72^\circ$, $m = 5$, then to calculate the camber at 54° :

$$m_\phi = \frac{5-36}{72-36} * (\phi - 36) + 36, \text{ for } \phi = 54, m_\phi = 2.5 \quad 3.24$$

When this deflection model is applied within the UDF, some assumptions must be taken. For example, as the linear interpolation points are only defined between 0° and 360° , the instantaneous azimuthal angle must be translated into this region. In order to implement this, a series of simple if statements are used i.e., if $ts < 360$, completed revolutions count (rev) = 0, if $360 < ts < 720$, $rev = 1$ etc. which is implemented into Equation 3.25.

$$\phi = (ts - ts_r * rev) * \frac{360}{ts_r} \quad 3.25$$

Where ts is the current timestep, ts_r is the number of timesteps per rotation, rev is the current turbine rotation number. ts_r is calculated by simply defining how many timesteps per degree of rotation are desired, i.e., 4 ts per degree of rotation or 0.25° per timestep results in $ts_r = 1440$.

The full UDF is provided in the thesis Appendix or available on the provided GitHub repo [84], but a summary will be provided.

Firstly, in the actual ANSYS Fluent simulation, the user provides a series of input parameters named $camber_n$, where n is any number up to the maximum optimisation positions, with each value set as the percentage value of chord that the profile camber should be at position n i.e. for NACA5315 at 36° $camber_1$ would be 5 for 10 optimisation points. For this thesis, $n=24$, as there was no benefit found to doing more, although most applications of the GA only used $n=11$, but the justification and study on this is explained in the subsequent chapters. The opt_count parameter and p parameter are also set in Fluent to define the number of optimisation positions (for calculating the angle between optimisation points) and the position of maximum camber respectively. Then some parameters relevant to the turbine are defined in the UDF, including the blade chord length, turbine radius, tip speed ratio, windspeed and %c LE offset. These are specific to the actual CFD turbine setup and co-ordinate frame so care should be taken to ensure these are accurate. During the simulation, the angular rotation per timestep is calculated based on the simulation timestep size and the turbine parameters. This is then used to calculate the deflection magnitude (using equations) by calculating the current azimuthal angle, it's respective camber value, and the next timestep camber value and then applying the difference to the blade shape. To apply the deformation, each timestep the aerofoil is translated to the global co-ordinate center such that the LE is at (0,0) using the matrix in Equation 3.26

$$\begin{bmatrix} x_{NACA} \\ y_{NACA} \end{bmatrix} = \begin{bmatrix} \cos(\phi) & \sin(\phi) \\ -\sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} x_{Global} \\ y_{Global} \end{bmatrix} \quad 3.26$$

then the aerofoil is deformed, using equations 3.21→3.25, before being translated/rotated back to the correct azimuthal position using the inverse of equation 3.26, shown using equation 3.27.

$$\begin{bmatrix} x_{Global} \\ y_{Global} \end{bmatrix} = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} x_{NACA} \\ y_{NACA} \end{bmatrix} \quad 3.27$$

3.4 Genetic Algorithm Optimiser

A genetic algorithm, also often referred to as an evolutionary algorithm, is a type of sequential optimisation algorithm designed to evaluate and improve a specific criterion. These are often found in computer science journals as sorting algorithms [85], for example creating an optimal route, such as the travelling salesman problem [86] or finding optimal designs. It is based on the Darwinian Evolution Theory [87], primarily, the concept of natural selection[88, p. 444]. This theory states that more organisms are created than can survive due to limited resources, therefore in a bid to survive, variations in certain characteristics are developed in order to maximise the chances of survival. These organisms breed with other surviving variations and cause further variations, and sometimes random variations are seen, independent of parents, and this is known as mutations. The best adapted individuals continue to survive and thrive, whilst less-fit members of the population slowly get extinct, thus the term coined “Survival of the fittest” [88, p. 444].

This concept is yet to be applied to dynamically controlled vertical axis wind turbines (VAWTs) but would allow the prior research in dynamically controlled VAWTs to be expanded on. It is achieved by using the Genetic Algorithm to find the optimal aerofoil camber based on a selected maximum thickness of the aerofoil, within the user defined bounds at pre-specified positions. The magnitude of these positions is known as the azimuthal angle, calculated as the angle taken between the position where the blade chord is parallel to the freestream velocity and its current position, this value is signified by ϕ in Figure 1-5. These azimuthal positions are selected as an array of equally spaced positions in the turbine’s rotation i.e., at 18°, 36° ... 306°, 324° to create a wholistic view of the deformation profile. The equally spaced positions are

used to improve the overall turbine performance by feeding the optimised cambers into the 2D VAWT Transient CFD simulation described in Chapter 3.3, to model the dynamic deflection of the blade as the turbine rotates. It is possible for the user to use as many optimisation positions as desired, at the cost of computational time. The chosen azimuthal angles are then used to calculate the theoretical angle of attack (α) on the turbine blade profile, based on the tip speed ratio (TSR) of the turbine and the wind freestream velocity. The steps to optimise each azimuthal position are outlined in the following sections.

3.4.1 Flow chart and logic flow

The general logic flow for the genetic algorithm crossover can be seen in Figure 3-12. It is relatively straight-forward in its methodology, with a multitude of logical arguments to ensure a reliable, accurate, and repeatable optimisation procedure. It starts with defining the algorithm constraints such as number of optimisation positions, crossover functions and the constraints of the turbine such as the TSR, blade chord and turbine radius. These parameters are then utilised to initialise the first generation of genotypes. Each of these genotypes are evaluated in the chosen tool (i.e. XFOIL, CFD or ANN) to calculate their torque, which is the fitness function of choice for the genetic algorithm in this work. Other fitness functions could also be easily implemented such as delaying the flow separation point on the aerofoil as much as possible, or maybe reducing the wake vorticity as much as possible, but these are outside the scope of this thesis.

The genotypes are sorted by best to worst to assign the parent genotypes for the next generation before creating the offspring and mutated genotypes. A simple check is undertaken to ensure that the genotypes are within the defined optimisation bounds, and if not, some normalization is applied to set the value to the closest optimisation bounds. Finally, after a number of generations, a convergence check is undertaken, to see if there has been any change in torque coefficient. If there has been an increase in torque over the convergence period, the simulation continues until there is no longer any changes in tangential force coefficient.

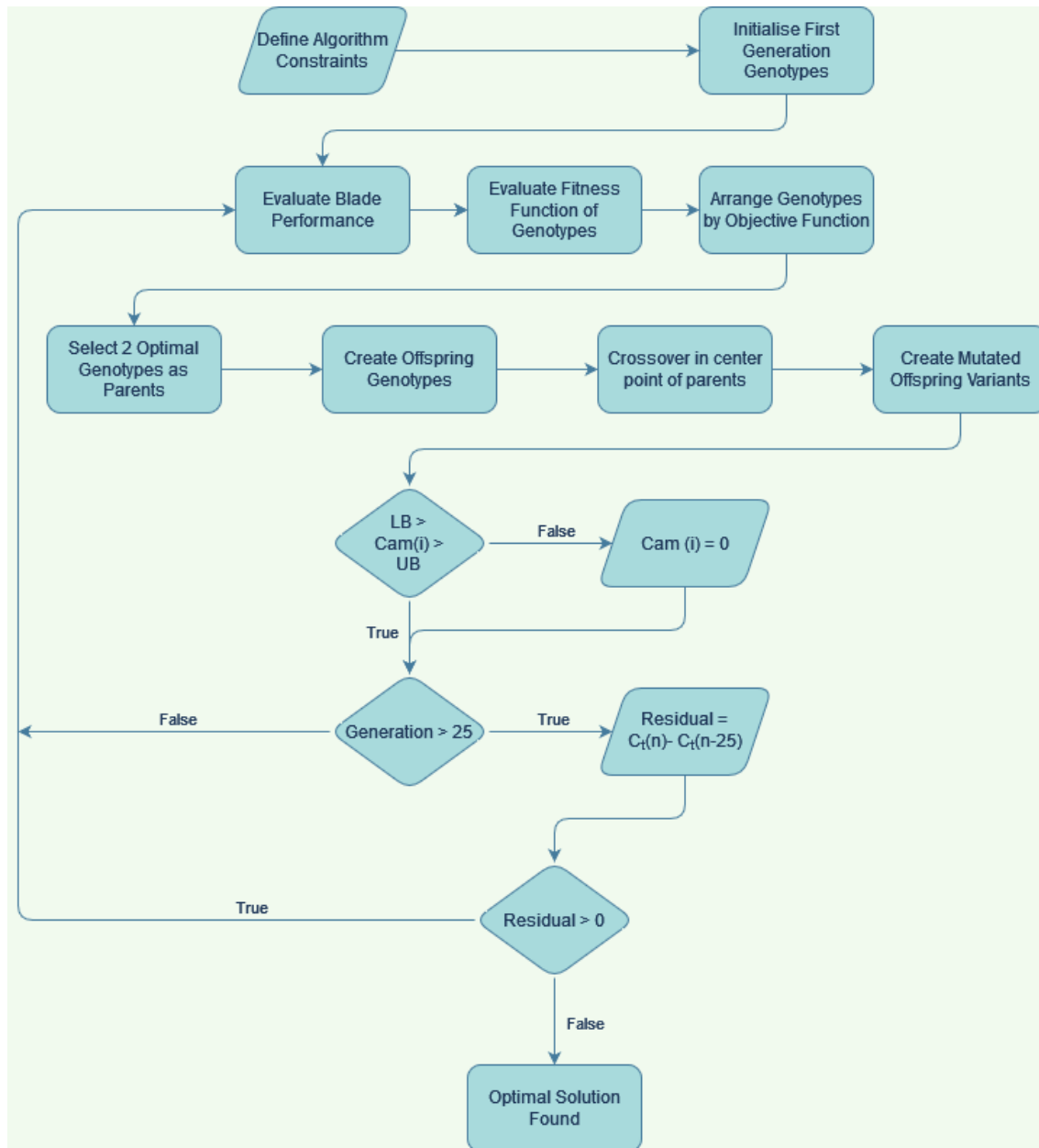


Figure 3-12: Example Logic Flowchart for the Genetic Algorithm

An example of the Genetic Algorithm's logic flow in action is presented in Figure 3-13. The initial generation is seen in the first row, with the optimal genotypes (camber values) highlighted. The offspring aerofoils are presented in the second row. These offspring include the parent genotypes, an offspring at the centre point of the parents, $\pm 5/10\%$ variations in camber magnitude, and a randomised genotype.

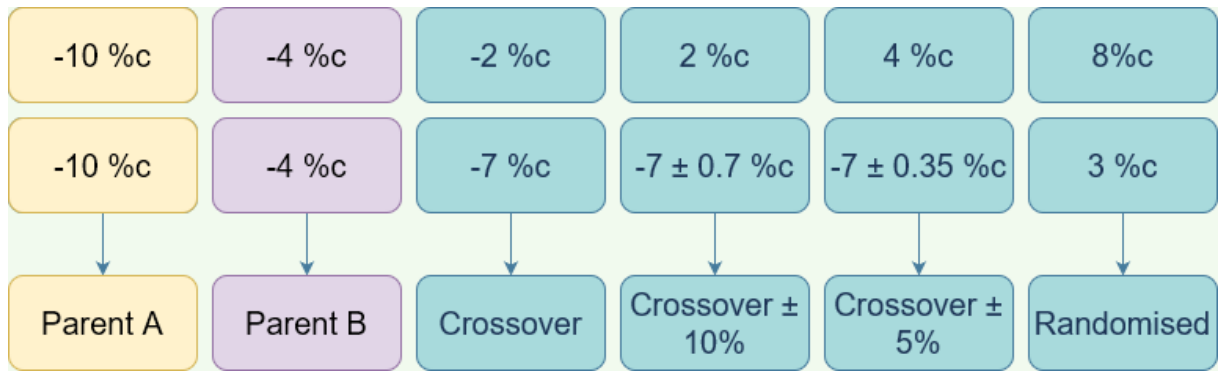


Figure 3-13: Example of offspring genotype generation

The crossover variation is to enable searching around local optima to increase the algorithm's searching efficiency. The randomisation is crucial to enabling a large breadth of search in the algorithm and prevents premature convergence on a local optimum aerofoil, rather than the overall optimal aerofoil. The crossover functions allow the algorithm to converge onto the optimal solution simultaneously, until the convergence criteria are met. In this scenario the convergence criteria is defined as either no change in the tangential force coefficient, as that is the value being optimised, for the last twenty-five generations, or until an upper limit to the number of generations is reached. Once this limit is reached, the algorithm is terminated, and the latest Parent A phenotype is taken as the optimised value. The reasoning for utilising twenty-five generations for the residual calculation is to ensure that there is sufficient time for any sudden changes in the searching for optimal aerofoils, based on the assumption that after 25 generations of no changes, the optimal would have been found. The 25 generations value was chosen as it was found that any less resulted in large variance in outputs, when the GA was repeated at certain azimuthal positions i.e., optimal camber at 36° varied across three runs (7.8%c, 9.8%c, 6.3%c). Any larger values resulted in excessive algorithm runtime without any improvements, the optimisation would plateau, and be considered as an optimal result, wasting computational resources and time for no further benefit.

3.4.2 Tuning the Genetic Algorithm

It should be noted, that based on Equation 1.19, the lift would not contribute to the tangential force generation at azimuthal angles of 0° , 180° and 360° , based on the reference frame. Therefore, the only way to maximise the tangential force at these angles, is to minimise the drag contribution. As the induced drag is increased from higher lifting profiles, such as a cambered profile, the conclusion is that a simple symmetrical aerofoil would be the optimal aerofoil at $\theta = 0^\circ$, 180° and 360° . Other parameters are tuned for the GA, including the chance of having mutations occur, the bounds of the algorithm and the mutation magnitudes. There are arrays of studies on the decision of these parameters within this thesis, and the complex factors that can affect these decisions.

4 NACA Aerofoil Camber Optimisation of a VAWT Blade Deformation Profile at Predefined Angles of Attack using XFOIL

4.1 Introduction

This chapter will outline the investigation into the benefits, drawbacks and limitations of utilising a genetic algorithm to optimise a transiently deforming blade profile for the VAWT design described in Section 3.2. The study's primary objective is to optimise the performance of the VAWT blade aerofoil profile through computationally efficient and inexpensive methods. XFOIL [89] is utilised as a rapid analysis tool to calculate and evaluate the performance of various genotypes, XFOIL is used to predict each potential aerofoil's tangential force coefficient within the genetic algorithm optimisation process with minimal computational cost. Whilst XFOIL is well established as an aerofoil analysis tool below a Mach number of 0.4 and in the linear angle of attack range [90], it has difficulties capturing viscous effects such as flow separation at high angles of attack and transitional flow. A commercial Computational Fluid Dynamics analysis software is therefore utilised to facilitate a more accurate evaluation of the optimised turbine shape profile in situ of a rotating, transient VAWT simulation. This chapter details the analysis between these methods, considering the reliability of XFOIL's predictions and provides recommendations on the utilization of XFOIL in future work.

4.1.1 Xfoil Specific Genetic Algorithm Implementation

To evaluate this theory, it is crucial to understand the detailed workings of a GA and its typical components. All files created as a result of employing this methodology can be found on GitHub at [91]. Firstly, the implementation of the GA in XFOIL is discussed, followed by some of the modifications made to the GA workflow which are unique to its implementation in XFOIL including geometry generation, turbine conditions and limitations.

4.1.2 XFOIL Genetic Algorithm Workflow

XFOil was used as the aerofoil performance calculation tool for the GA. It was chosen due to its robust accuracy and flow analysis performance as shown by Morgado et al. [92] and it has a relatively inexpensive overhead, allowing for the GA to be run with a fast turnaround. Additionally, due to its command line-based process, it is possible to implement within batch scripts to allow for an automated and input free optimisation process.

To create the GA, a series of MATLAB functions are required, and these include:

- i. A MATLAB function for creating a .dat file containing the XY coordinates of vertices, depicting the aerofoil profile to be assessed, defined using the NACA 4-Series profile equations (discussed earlier in Section 1.3.1).
- ii. A MATLAB function for inputting the filename for the respective aerofoil into a script that is readable by Xfoil, along with defining the simulation parameters such as the Mach and Reynolds number.
- iii. A MATLAB function that parses the output data from Xfoil and imports it into a MATLAB session for data analysis and recombination.
- iv. A MATLAB function for the mutation/crossover of parent genotypes to generate offspring.

Function (i) would use the equations detailed in the thesis methodology for a NACA 4-series aerofoil to calculate the co-ordinates of the aerofoil surface. XFOIL limits the

number of points per curve file to 494, so that was selected as the number of points to span the aerofoil chord, such that if the chord is a unit length (1m), each point would have an x-value in the NACA frame; with $x=0$ at the leading edge and $x=1$ at the trailing edge, increasing in value by $1/494$ for each successive position i.e. If one was to calculate the co-ordinate of the 5th point, the value for x that is submitted into the respective NACA equations would be $5/494$.

Function (ii) acts as a setup script for XFOIL, it sets the angle of attack, Mach number, Reynolds number using Equations 1.18, 1.20 and 1.21 respectively for each optimisation position. The function then proceeds to import the previously generated aerofoil co-ordinates file, and instructs XFOIL to run the solution.

Function (iii) uses regular expressions to read the output file from XFOIL (containing the performance polars i.e. lift and drag coefficient) and the simulation's operating conditions as specified in Function (ii). The complex regular expressions allow for the imported text file to be turned into a tabulated data set and then manipulated as needed to be feed into the optimisation algorithm or utilised for figure creation, the full script outlines the expressions and tabulation procedure in the code repository [91].

Function (iv) is the most crucial function for the genetic algorithm setup. This function takes the working dataset (containing the tangential force coefficients calculated using Equation 1.19 and camber of each generation's phenotypes) and sorts them by highest to lowest tangential force coefficient.

This sorted array is used to create the offspring candidate of the next generation from the two current most optimal candidate aerofoils, and mutations and variations of the offspring are utilised to create a new array of optimal candidates for testing. This function also evaluates convergence of the algorithm to establish the point at which the optimal candidate is found and the optimisation process can be terminated.

These functions are combined into one script to ensure consistency, and a “for” loop is iterated from the first non-zero angle of attack to the $(n - 1)^{\text{th}}$ term. In this study, the total number of optimised positions (n) is varied, but if the selected number of

points was 10, then the profile would be optimised every 36° of azimuthal rotation. Azimuthal angles multiples of 180° are not optimised due to the assumptions discussed in Section 0, where at 0° , 180° and 360° the camber would be negatively dominated by drag, so it is not necessary to run the algorithm at these angles of attack.

It was found that MATLAB would develop what is assumed to be a memory leak and would often stop the loop after an undetermined number of generations at an arbitrary number of angles of attack. This issue was attempted to be resolved in a multitude of ways, including running on multiple computers and running for less generations before restarting the computer. It was ultimately found that the optimal method was to create a function within the Genetic Algorithm script to run the optimisation process with ease and successfully. This allowed for the memory to be cleared and reset between each optimisation generation, thus preventing data conflicts and the resultant corruption.

For the sake of efficiency, it is crucial to automate the process of running this function in parallel. Due to the development environment being based on Windows 10, it was found that creating Windows Batch Files would enable the user to easily run separate instances of the function at the same time in the command terminal, and these scripts could be easily implemented on alternative operating systems such as Linux. These batch files required a folder structure as follows:

Main Source Folder -> *Batch_gen.m* + Output Data Folder

When one runs the *batch_gen.m* function with the required input arguments, it will create a series of batch files that run the GA function, alongside a single batch file called *Parallel.bat*. This parallel file sets the affinity of the optimisation process to ensure each process has sufficient computational power, rather than all the processes potentially sharing the compute power, reducing the chances of memory and file corruption.

The *Parallel.bat* GA script will therefore run at various AoA simultaneously, and once the desired convergence criteria (in this study the peak moment must not have

changed by more than 1% in the previous 25 generations) is met, the data and variables will be saved in an appropriately named MATLAB file, and images of the GA process is saved.

4.1.3 Genetic Algorithm Modifications in XFOIL

A handful of features of the Genetic Algorithm workflow are discussed in Section 3.4 have been modified specifically for suitability with XFOIL, due to the specific configuration and requirements of using XFOIL. One of the most important variations is the number of generations used for the convergence checking, which required extensive generations when using XFOIL. The algorithm was tested with one generation, 10 generations and 25 generations as the sampling count. The turbine operating conditions used to evaluate the different convergence checks matched the turbine design discussed in Section 3.2, namely a two-bladed turbine with a radius of 0.85m, utilising a NACA0015 blade profile with a chord length of 0.225m, wind speed of 7 m/s, and a tip speed ratio of 2.29. These values are chosen as they are the characteristics of the turbine design and operating conditions used by many established and impactful studies [23], [80], [93], [94], and is, therefore, a well-understood turbine that will allow for deep insights into the applicability, performance and reliability of the Genetic Algorithm optimised turbine blade profile technique. These tests were run three times for each configuration, namely Test A, Test B and Test C as an example to ensure consistency and reliability in the genetic algorithm optimisation workflow.

4.1.3.1 XFOIL Normalisation Effect

To select the most appropriate and efficient method of calculating aerofoil characteristics such as lift, drag, and, as a result, the tangential force coefficient, it is important to understand how XFOIL interprets the geometrical inputs created in the genotype candidate creation due to the non-unit-length chord of the turbine blade profile.

According to the XFOIL User Guide, the lift coefficient is not normalised with the chord length [89]. There is a weak Reynolds number effect on the lift due to the chord length of the turbine, but the chord length significantly impacts the drag due to its coupling with the chord Reynolds number [89]. Extensive studies have shown that increasing Reynolds number should reduce drag coefficient and increase lift coefficient for an aerofoil [95], [96]. As a result, when instructing XFOIL to import an aerofoil geometry file (.dat file), if the chord length is less than a unit metre, XFOIL will largely underpredict the lift coefficient and over-predict the drag [97]. The user guide states that a solution to this issue is to increase the Reynolds number proportionally with a reduction in the chord length. According to the user guide theory, this should weakly increase the lift coefficient and decrease the drag coefficient in line with what would be expected of a unit-length aerofoil. The guide also mentions that enabling normalisation in the XFOIL menu will scale the imported aerofoil up to unit chord length, and therefore, the imported aerofoil lift and drag properties of the aerofoil should match that of a unit chord aerofoil if it were generated within XFOIL's geometry generation.

To ensure the XFOIL user guide's accuracy, a series of solution settings combinations were tested for a NACA5315 aerofoil at the equivalent flow conditions of the VAWT described in Section 3.2 at an azimuthal angle of 36° . This aerofoil was selected due to XFOIL's geometry generation methods; namely that when creating a NACA aerofoil using the built-in tools, an integer value is required to be provided for all 4 sections of the NACA 4-Series code i.e. you cannot request a camber of 4.335%c, it must be 4% or 5%. The 15%c thickness was chosen as that is the same thickness as the original turbine and is used for all other sections within the thesis (ensuring consistent solidity, as that can alter the optimal tip-speed ratio otherwise), and the 30%c position of maximum camber was selected as it is simply a commonly selected option for most examples. The 36° azimuthal position was implemented as it is in the linear region of the aerofoil's angle of attack – C_l range still, ensuring that the aerofoil would not be stalled, ensuring that any conclusions drawn are solely based on XFOIL setup parameters, rather than differences in XFOIL's simulation abilities.

The configurations included an increased chord length aerofoil (1m rather than 0.225m), normalisation enabled in the XFOIL menus, and scaling the Reynolds number appropriately according to the .dat chord length. These are all recommended techniques for normalising non-chord length aerofoils, either in common knowledge or provided in the XFOIL user manual. The five test scenarios were as follows:

1. Original 0.225m chord length aerofoil geometry file
2. Original 0.225m chord length aerofoil geometry file with a Reynold number set using a chord length of 1/0.225
3. Original 0.225m chord length aerofoil geometry file with normalisation enabled in the XFOIL menus
4. Original 0.225m chord length aerofoil geometry file with normalisation enabled in the XFOIL menus and the increased Reynolds number
5. Generating the aerofoil geometry within XFOIL

The MATLAB geometry and XFOIL profile were both NACA5315 aerofoils, as this relatively high camber design will accentuate any potential method irregularities, with the base flow conditions listed in Table 4-1.

Case Type:	1	2	3	4	5
Angle of Attack (°)	10.746				
Mach Number	0.0474				
Reynolds Number in XFOIL	2.34 e+5	1.040 e+6	2.34 e+5	1.040 e+6	2.34 e+5
Effective Chord Length in Geometry (m)	0.225	4.44	0.225	4.44	1
XFOIL Normalisation?	No	No	Yes	Yes	No

Table 4-1: Flow Conditions for different XFOIL setup test scenarios

The resultant lift and drag coefficients for the NACA5315 are shown in Figure 4-1. The original workflow (black bar) disagrees significantly with the aerofoil generated by XFOIL's input functions. This signifies that the imported .dat file with a chord length of 0.225m significantly underpredicts the lift coefficient and drastically over-predicts the drag coefficient, this is suspected to be due to the change in profile drag from the larger chord lengths, additionally, the local Reynolds number would vary altering the transition length on the aerofoil surface. If the Reynolds number is scaled accordingly by a factor of c^{-1} , then, according to the XFOIL documentation [98], this should accurately report the lift and drag polar, but it is clear this is not the case. Utilising XFOIL's built-in normalisation command provides significantly improved results. The lift coefficient for the normalised .dat geometry is slightly below the NACA5315 generated in XFOIL, as is the case for the drag coefficient. Again, increasing the Reynolds number proportionally shows better agreement with the XFOIL lift coefficient but severely underpredicts the drag coefficient. It is therefore concluded that importing the true-to-scale ($c = 0.225\text{m}$) file generated from MATLAB and then applying normalisation within XFOIL is the most suitable method, as the lift-to-drag ratio (which dictates the tangential force coefficient) is closest to that of the XFOIL generated profile. Using the XFOIL generated profile as the baseline is a reliable and safe choice, as it is a well-established and robust program that has been extensively utilised in the literature for aerofoil lift and drag evaluation [99], [100], [101] and is well validated [92].

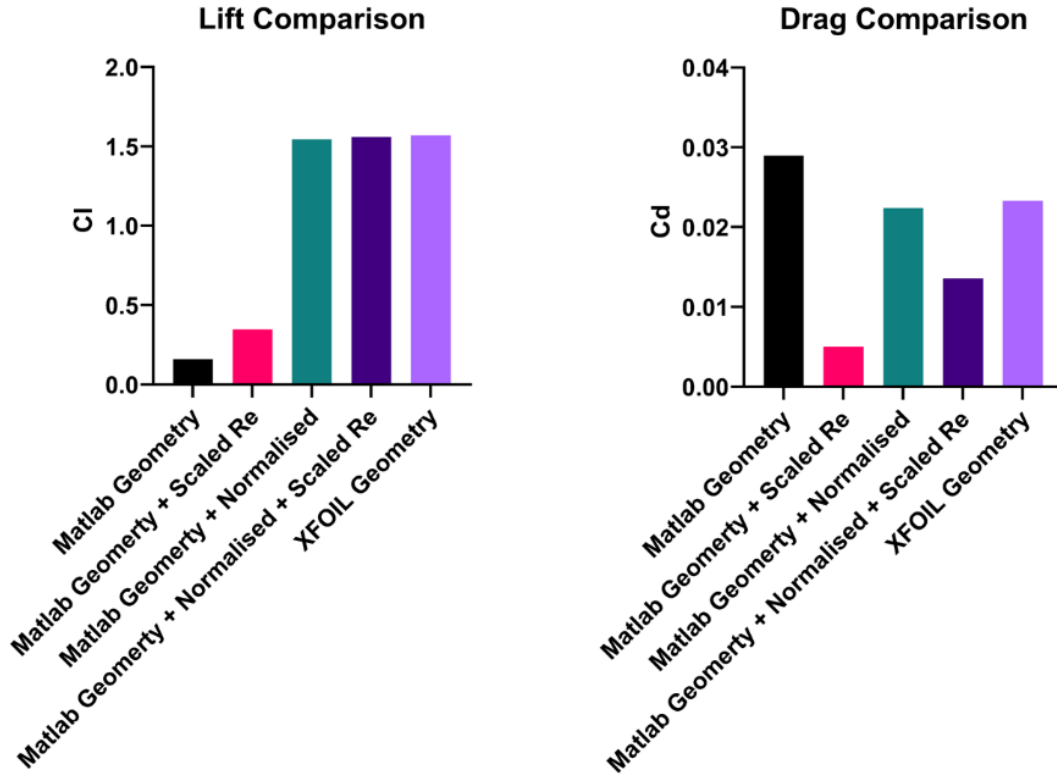


Figure 4-1: Comparison of NACA5315 Aerodynamic Coefficients for Different XFOIL Setup Methods

To further ensure the validity of the XFOIL methodology for implementing the GA optimisation, the most robust geometry method from the XFOIL setup studies (MATLAB Geometry generation with normalisation enabled in XFOIL) is compared to a CFD simulation of a NACA2412 blade as completed by Heteyi et al. [102]. The study was using the conditions of 0° and 10° angle of attack for a free stream velocity of 42.89 m/s, for air with a density of 1.2041 kg/m^3 and a kinematic viscosity of $1.511083 \times 10^{-5} \text{ m}^2/\text{s}$. The Reynolds number provided was 2.85×10^6 , meaning the chord length is 1m. This was implemented into the MATLAB geometry generator and then the MATLAB XFOIL interface detailed in Section 4.1.1. The resultant values from XFOIL and the literature are presented in Table 4-2.

	Lift Coefficient at 0° AoA	Lift Coefficient at 10° AoA
Literature [102](CFD)	0.23217	1.253836
XFOIL Method (This work)	0.2479	1.3195
Percentage Difference	6.78%	5.23%

Table 4-2: Lift Coefficient comparison between XFOIL and published literature [102]

Whilst the difference between XFOIL and the CFD method is not negligible, it is somewhat expected due to XFOIL's design of a panel method, which simplifies complex flow phenomena such as turbulence, laminar to turbulent transition and separation. These simplified models will introduce some assumptions which can result in overpredicting lift coefficients, but the results are considered acceptable for the significant increase in computational speed, which is ideal for this preliminary GA camber profile optimisation study.

4.1.3.2 Limitations with XFOIL based optimisation

One of the disadvantages of utilising XFOIL as the aerofoil performance analysis tool is the difficulty in the convergence when separated flows are present. This is seen in lower TSR turbines ($TSR < 3$), where the theoretical AoA is often within the deep stall region [80]. The solution would be to apply this methodology to higher TSR turbines, where the effective AoA is often lower, but that severely limits the overall applicability of XFOIL. Some specialised techniques to aid the method's reliability include 'initialising' the solution by solving at an angle of attack of 1° before calculating the lift and drag coefficients at the target angle of attack and setting the iteration count to 1000. 1000 iterations is chosen as it allows for sufficient opportunity for the solution to converge, but effectively limits the timeout to ~15 seconds. Finally, if XFOIL can still not get the aerofoil performance at the desired flow conditions, the corresponding torque value is set to -9999 in MATLAB. This effectively creates NULL values for the problematic phenotypes, removing it from the pool of potential candidates. This is implemented as the phenotype is clearly unsuitable for use in the

turbine, and should therefore not be considered, as XFOIL is unable to converge as a result of significant flow separation.

Due to the lack of transient effects in the instantaneous XFOIL optimisation technique, it is crucial to verify and validate the XFOIL-based optimised camber profile by using a CFD solver, such as ANSYS Fluent. CFD is used instead of experimental data due to the increased time and monetary requirements of creating a physical flapping turbine model compared to CFD, and it allows for substantial flexibility in the optimisation, such as testing alternative positions of maximum camber.

4.1.3.3 Turbine Conditions

The GA is applied to the turbine investigated by El Sakka et al. [80] and Li et al.[81], which have been experimentally evaluated at a series of TSRs. The turbine parameters are presented in Table 4-3.

Parameter	Aerofoil Profile	Aerofoil Thickness (m)	Chord Length (m)	Turbine Radius (m)	Setting Angle (°)
Value	NACA0015	0.035	0.225	0.85	6

Table 4-3: Turbine Parameters

The setting angle is defined as the position that the blade is at in relation to the 0 azimuthal angle. For reference, a 0° setting angle would correspond to the chord being parallel to the freestream flow at ϕ (Azimuthal Angle) = 0°, so a 6° setting angle would result in rotating the blade leading edge outwards (away from the turbine centre). For a graphical representation, please refer back to **Section 0**. These turbine parameters are input to the XFOIL input and aerofoil profile generation functions to define the angle of attack, Reynolds Number, Mach number, and aerofoil shape, based on the formulae previously discussed.

4.2 XFOIL-Based Genetic Algorithm Results

With the turbine parameters described in Section 4.1.3.3, the GA is run at a TSR of 2.29 to replicate the flow physics best and compare it with the validated CFD case found in Section 3.2. An array of different variations of configurations for the GA are tested, including using a turbine setting angle of 6° as the validation CFD case uses and tested at a 0° setting angle. Furthermore, these setting angle cases are tested for both an original geometry coordinates definition with the turbine chord length (0.225m) and with aerofoil normalisation enabled within XFOIL to understand the impact of this setting and deciding which is most appropriate, by comparing with generating the aerofoil profile within XFOIL's internal design scheme. Additionally, alternative angle of attack formulations including an induced velocity technique and a more complex CFD based profile by Elsakka et al. are detailed and used to approximate and apply the genetic algorithm to a more realistic angle of attack. The standard angle of attack formulation is based on the theoretical angle of attack calculated using the formulae described in Section 0, this is not the case in the turbine scenario, as the energy extraction in the upstream rotation results in a lower freestream velocity impacting the downstream section, combined with the wake impact on flow vorticity and direction. Accurately implementing the changes in the flow vorticity, turbulence, pressure, etc. from the upstream optimised blade's wake is outside the scope of this work but would be an interesting avenue for exploration in future works.

4.2.1 Optimisation Limits Investigation (Setting Angle = 6°)

The turbine parameters state a setting angle of 6° , which was incorporated in the GA angle of attack definition by simply adding the setting angle to the calculated theoretical angle of attack (see Section 0 for details on how this angle is calculated). This was used to establish some of the GA bounds for later investigations, such as whether a 0° setting angle would increase the overall turbine performance and what would be appropriate bounds for the optimisation search.

The initial test of the Genetic Algorithm was run for both a 0° and 6° setting angle to investigate if a $\pm 5\%$ camber magnitude bound was appropriate. This bound was tested to limit the range of motion of the blade flapping, which can assist in reducing the power consumption of the motors driving said motion and improve the overall turbine performance and efficiency in situ. After the initial testing, it was found that for the 6° setting angle, the GA would hit the predefined camber constraints when set to a $\pm 5\%$ limit and, as a result, would reduce the overall performance of the GA. Therefore, it was also evaluated with a camber limit of $\pm 10\%$, this was found to validate the hypothesis of the GA trying to exceed the predefined bounds. If we take the optimisation at $\phi = 108^\circ$ as an example, the $\pm 5\%$ limit is chosen as the optimised camber, but when allowed to explore values of $\pm 10\%$, the found optimal is $\approx 10\%$. The optimisation positions with a camber value greater than $\pm 5\%$ also resulted in a larger tangential force coefficient.

These points selected are placed every 36° of rotation and are chosen as they are evenly dispersed through the entire blade's rotation, whilst maintaining enough points to reflect the primary points of interest in the rotation such as at 108° which was approximately the ximum torque generation, but also covers other points of interest such as around the 180° position. If this number were to be increased, computationally cost will increase proportionally, but it might not provide further overall benefit to the turbine's power generation ability. Linear interpolation is applied between the optimised positions to provide a gradual deformation between each camber magnitude. The linear interpolation was found to be appropriate as it provides the most flexibility in implementing the deformation profile in CFD, but also allows for the camber profile at the optimised positions to be exactly as intended; if some form of polynomial curve fitting were applied (instead of linear interpolation) to have a continuous equation describing the deformation, the positions of large camber increase or decrease rate would overshoot, resulting in large camber magnitudes than the GA found, causing a reduction in relative performance.

Figure 4-2 shows the tangential force coefficient for the NACA0015, $\pm 5\%$ optimisation bounds and $\pm 10\%$ optimisation bound at the optimised azimuthal

positions at a 6° blade setting angle. The overall average tangential force coefficient for the GA Optimised deformation profile is significantly better than that of the symmetric NACA0015 aerofoil at all optimised positions. This conclusion is valid for both the $\pm 5\%c$ and $\pm 10\%c$ limits. This is a result of not only the increased lift to drag ratio of the cambered aerofoils in comparison to the NACA0015 aerofoil, but further emphasised by the fact that only the optimised profiles were able to report the torque values. By not reporting torque values (especially emphasised at 108° azimuth), it implies that XFOIL is unable to converge the simulation, which is a result of significant separation of the aerofoil. This in turn implies that the cambered aerofoil has significantly less separation than the NACA0015 at some of the optimisation positions such as 72° , 108° , 144° and 252° . The positive tangential force generation at 0° , 180° and 360° is a result of the 6° setting angle, but for the 0° blade setting angle it is expected to have no tangential force generation (as a symmetric aerofoil produces no lift at 0° angle of attack).

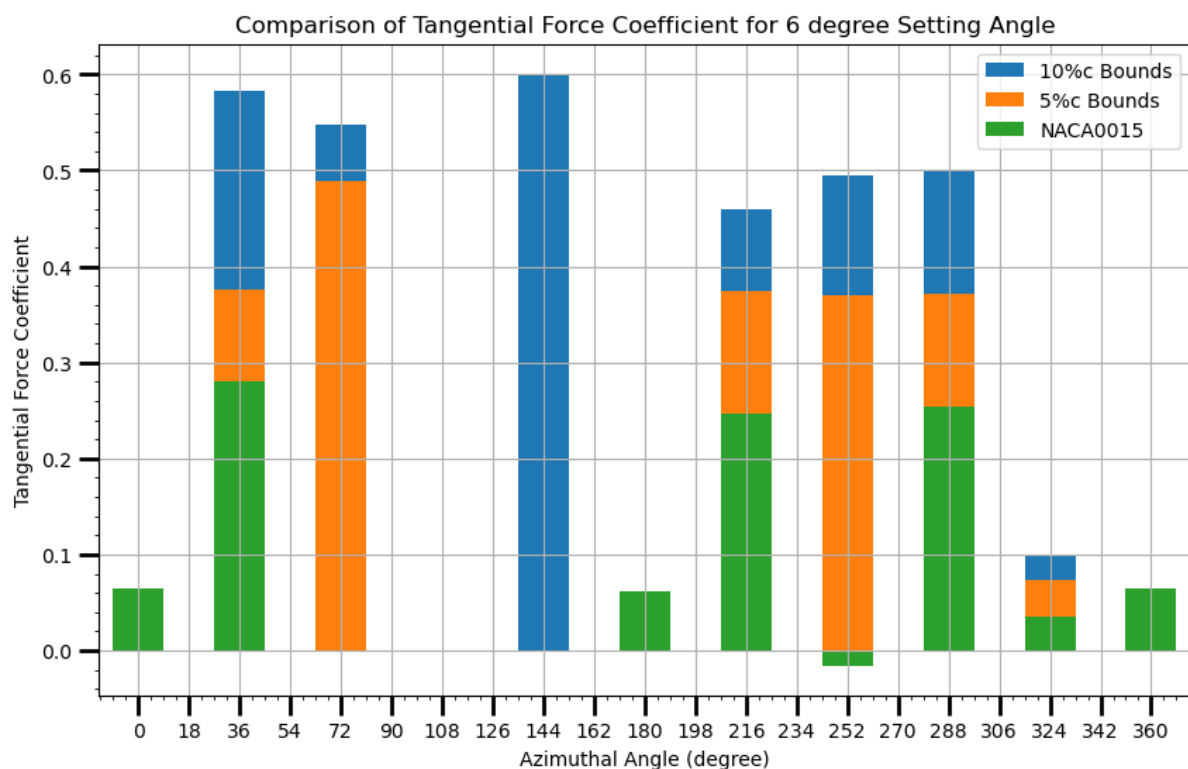


Figure 4-2: Tangential Force Coefficient for 6 degree setting angle with NACA0015, $\pm 5\%c$ Optimisation Bounds and $\pm 10\%c$ Optimisation Bounds profiles

Figure 4-3 shows the aerofoil maximum camber magnitude at each optimised position for the $\pm 5\%c$ and $\pm 10\%c$ optimisation bounds. Looking at the 72° and 252° optimised camber positions shows that both optimised camber positions have less separation than the symmetric NACA0015 aerofoil. Interestingly, the 144° position only has a converged aerofoil for the $\pm 10\%c$ bounds, which suggests that one cause of reduced separation could be the dipping effect of the leading edge to the incoming flow, as a $5\%c$ bound was insufficiently cambered to improve the lift enough to offset the increase in profile drag. It is expected that the optimisation positions that correlate with a lower angle of attack would have the highest camber magnitude, as these positions are less susceptible to the aerofoil stalling. This also explains why the $5\%c$ bound was hit for all optimisations in the downstream rotation and in the upstream region, where the angle of attack was significantly reduced enough due to the setting angle, as the aerofoil could produce a better lift to drag ratio.

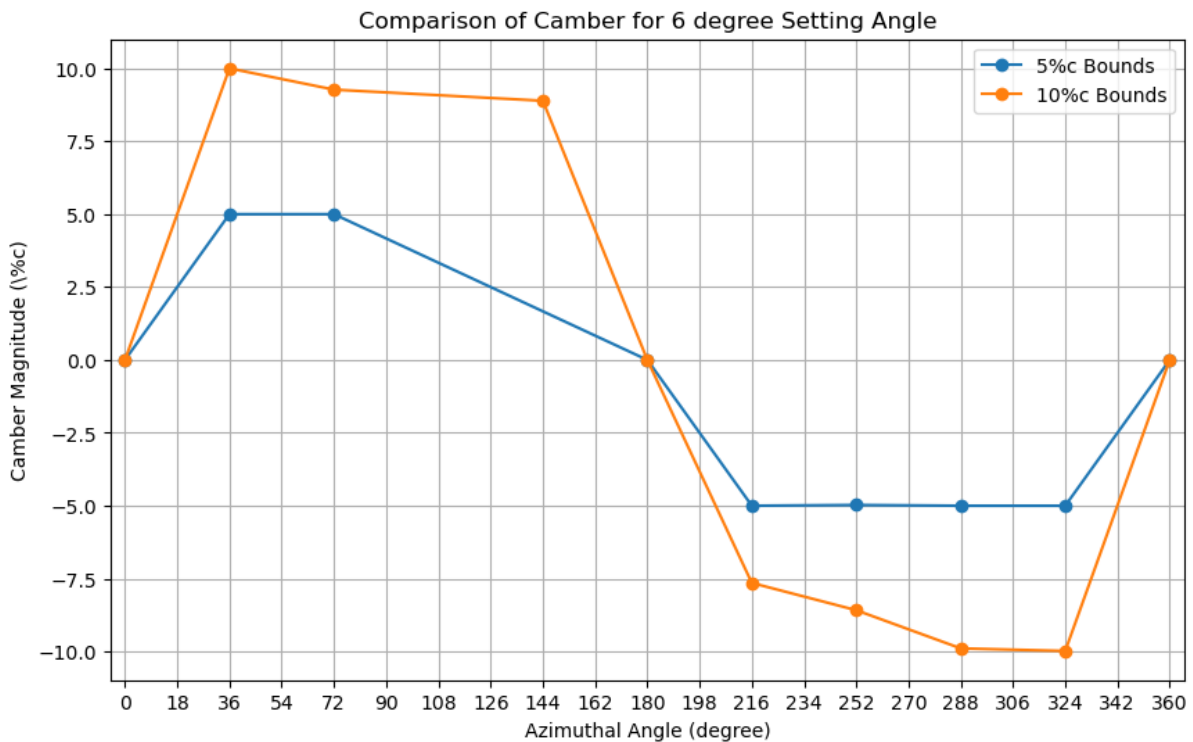


Figure 4-3: Aerofoil maximum camber value for 6 degree setting angle with $\pm 5\%c$ Optimisation Bounds and $\pm 10\%c$ Optimisation Bounds profiles

When tested at a setting angle of 6° as seen in Li et al. [81], the tangential force coefficient is lower at the downstream azimuthal positions ($\phi > 180^\circ$), as a result of the lower angle of attack experienced by the blade, due to the positive setting angle

reducing the blade angle of attack magnitude and the effects of the blade-wake interaction. This will result in uneven turbine loading, which is one of the issues plaguing VAWTs, and this work reduces, by significantly improving the downstream tangential force generation and reducing overall tangential force variance in the turbine rotation. Therefore, it is decided that the GA is best suited to a zero-degree setting angle to alleviate this uneven weighting of the tangential force contribution and for the simplicity of application when incorporated into the rotating turbine CFD model in ANSYS Fluent. In addition, it should be noted that due to XFOIL's solution methods, it struggles in the stall scenarios, as mentioned in Section 4.1.3.1. This resulted in some optimisation positions not completing as the Genetic Algorithm process would crash due to XFOIL's inability to complete lift and drag evaluations. An example of such is when the blade had a 6° setting angle, with the peak angle of attack = 31.4° at $\phi = 108^\circ$, evidenced by the Tangential Force Coefficient being zero at these positions. This is an inherent limitation of XFOIL for the high angle of attack scenarios, the only method that would circumvent the issue would be to test at significantly higher tip speed ratios, as the angle of attack would be sufficiently reduced to ensure XFOIL is able to converge. Alternatively, these positions, wherein the GA is unable to converge, could be ignored for the camber profile creation and instead the camber value could be interpolated between the last and next optimised position. Taking the 108° azimuthal position example again, that would suggest that the camber value would be $\sim 9\%c$, by interpolating between the $\sim 9\%c$ camber optimum found at 72° and the same at 144° . This technique was employed for calculating the mean tangential force coefficients to alleviate the NACA0015's difficulty to converge, which would heavily skew the statistics.

Table 4-4 contains the period averaged tangential force coefficient for the standard NACA0015 aerofoil compared to the GA optimised camber profile for the $\pm 5\%c$ and $\pm 10\%c$ camber bounds.

To calculate the mean tangential force coefficient value for each dataset, the Equation 4.1 is used to calculate and to calculate the relative time averaged performance, Equation 4.2 is used. $C_{t,i}$ is the instantaneous torque coefficient, n is

the number of optimised / tested positions and $\overline{C_{t,rel}}$ is the relative tangential force coefficient and $C_{t,NACA0015}$ is the average tangential force coefficient of the NACA0015.

$$\overline{C_t} = \frac{\sum_{i=0}^n C_{t,i}}{n} \quad 4.1$$

$$\overline{C_{t,rel}} = \frac{\overline{C_t}}{C_{t,NACA0015}} \quad 4.2$$

The relative difference to the standard aerofoil is provided to quantify the significant improvement in the turbine performance. Both optimised profiles have significantly higher average tangential force coefficients than the NACA0015 profile, with the $\pm 10\%c$ case performing considerably better than the $\pm 5\%c$ case. This is a result of two factors; a “missing” optimisation point at 144° and secondly, the camber consistently being limited to the maximum bounds at all the optimised positions. The $\pm 5\%c$ case not converging at an azimuthal angle of 144° is hypothesised to be the result of the leading edge not being drooped enough to accommodate the high angle of attack (21.6475°) at that optimisation position. The impact of regularly reaching the optimisation bounds is that the tangential force coefficient is also stunted, shown by the larger magnitude of the tangential force coefficient seen in Figure 4-2.

Aerofoil	Average Tangential force Coefficient	Relative Time Averaged Performance (%)
NACA0015	0.1239625	100
GA ($\pm 5\%c$)	0.24954444444444446	201.3
GA ($\pm 10\%c$)	0.34784000000000004	280.6

Table 4-4: Tangential Force Coefficient Comparison for different Genetic Algorithm Bounds with a 6° Blade Setting Angle

4.2.2 Setting Angle Comparisons

The original turbine utilised a setting angle of 6° . This is selected as it “corresponds to the maximum power coefficient” [81] for the turbine being optimised, assuming that it increases the average power coefficient, but without any details for this selection reasoning, it is difficult to infer whether this premise is accurate. Figure 4-4 shows the angle of attack against the azimuthal angle for a turbine operating at a tip speed ratio of 2.29. The 0° azimuthal position is given as the point at the top of the turbine’s rotation, where the wind travels parallel to the blade’s chord, with the leading edge upstream of the trailing edge. The 0° setting angle shows a perfectly symmetrical distribution about the azimuthal angle of 180° , where the angle of attack flips due to the turbine blade rotating in the downstream period of rotation. The angle of attack of 6° setting angle is offset vertically by 6° , resulting in a higher peak angle of attack, but also a higher minimum angle of attack. In theory, this practice can help to improve the power generation in the downstream period of rotation due to less time spent in the regions of significant static stall of the blade [103].

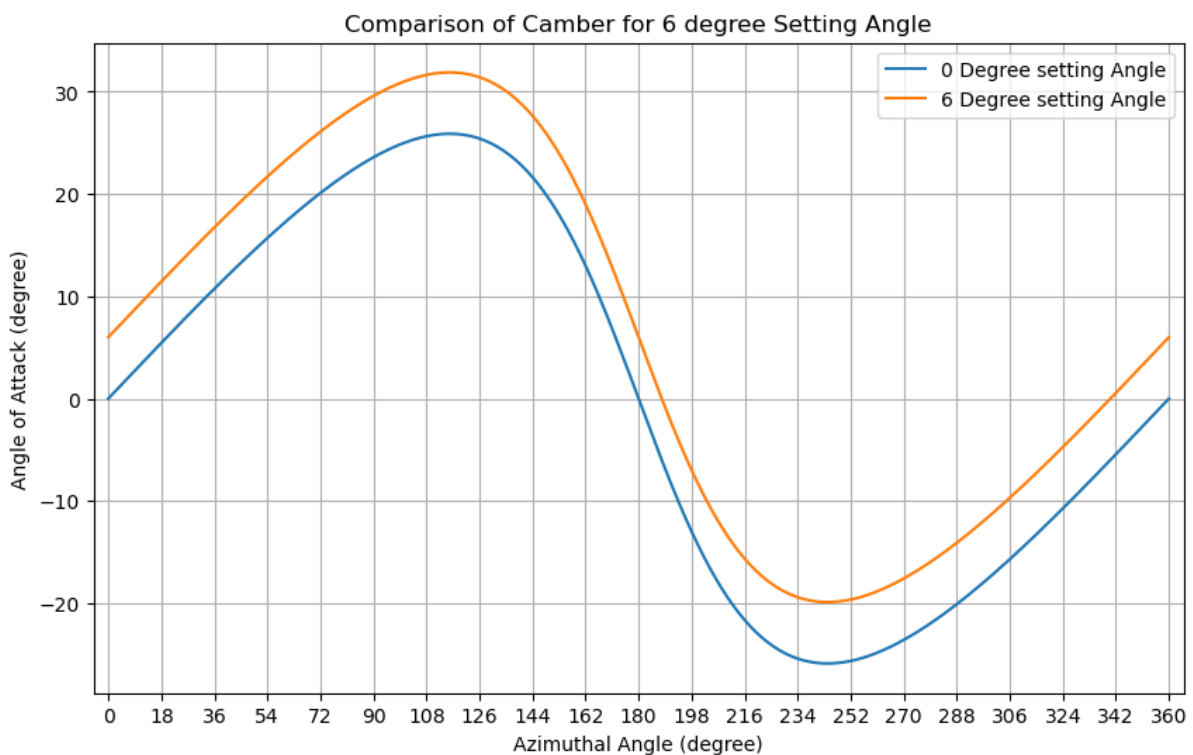


Figure 4-4: Angle of attack Comparison for a 0° and 6° setting angle of the turbine blade.

4.2.3 0° Blade Setting Angle

Running the GA with a 0° setting angle gives a better understanding of the GA methodology and applicability. Figure 4-5 shows the comparison of the tangential force coefficient for the NACA0015 and $\pm 5\%c$ and $\pm 10\%c$ optimisation bounds, with a setting angle of 0°

Figure 4-5 shows the results of the tangential force coefficient from the optimisation process compared to the standard NACA0015 aerofoil, with a 0° setting angle. One of the key takeaways is that the NACA0015 is completely stalled at $\phi = 108^\circ, 144^\circ, 216^\circ$ and 252° , shown by the 0 tangential force coefficient as XFOIL is unable to converge the solution. In contrast, the optimised shapes are not only able to have their performance evaluated but also show significant improvements overall. Every optimisation position is higher than the symmetric NACA0015, particularly for the $\pm 10\%c$ bounds. The symmetric profile is expected, as the use of the theoretical angle of attack means that the angle magnitude is symmetric at 180° , as seen in Figure 1-6 in **Section 1**.

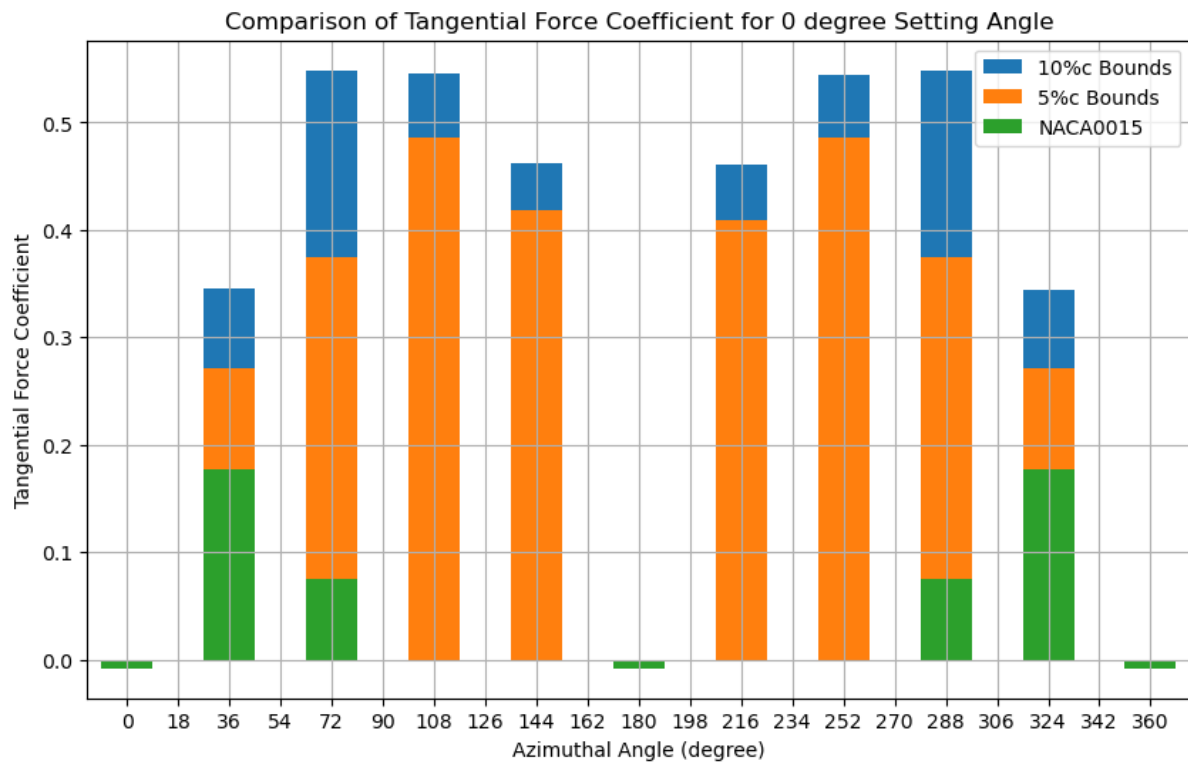


Figure 4-5: Tangential Force Coefficient Comparison for NACA0015, $\pm 5\%c$ Optimisation Bounds and $\pm 10\%c$ Optimisation Bounds profiles

The camber plot in Figure 4-6 for the $\pm 5\%c$ and $\pm 10\%c$ optimisation also suggests that the optimisation typically finds the highest possible camber that is not fully stalled at each angle of attack, and the magnitude of this camber reduces after the peak angle of attack is reached. It is also clear that the $\pm 5\%c$ bounds do not provide a large enough searching space for the optimisation algorithm, with all the optimisation positions found to be greater when $\pm 10\%c$ bounds are tested.

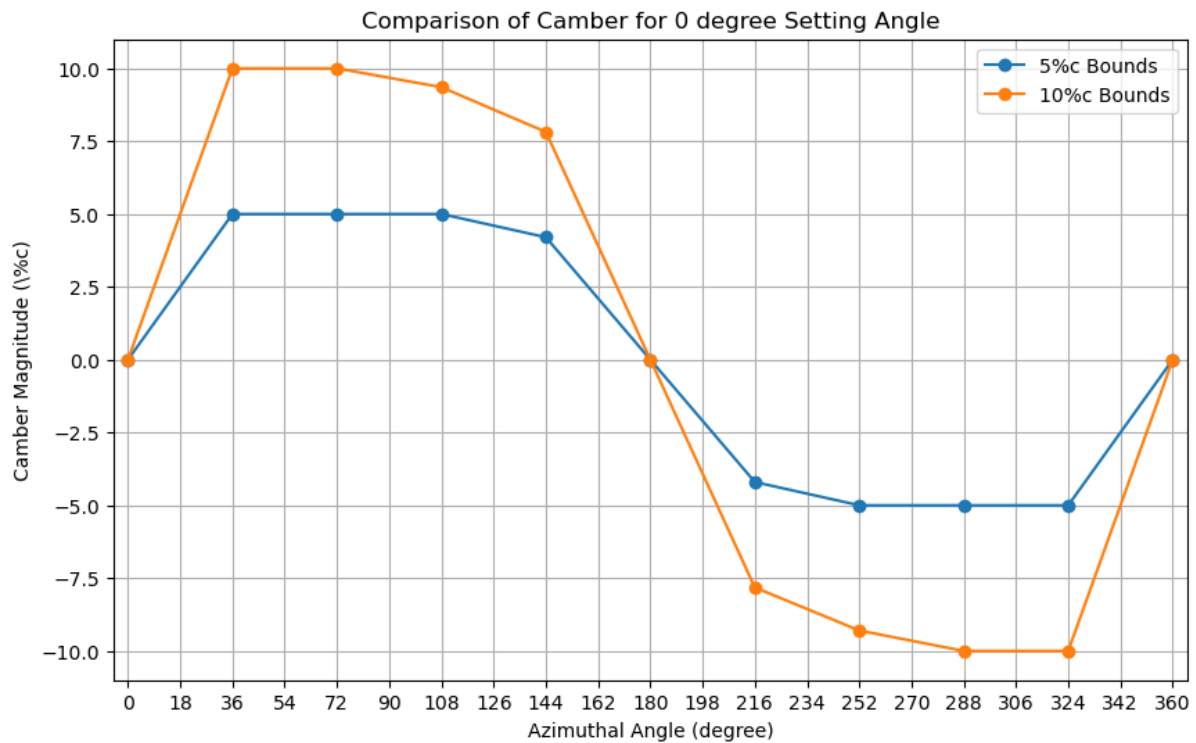


Figure 4-6: Camber magnitude at the optimised positions for different optimisation bounds of $\pm 5\%c$ and $\pm 10\%c$

Table 4-5 further demonstrates the strength of the GA optimisation procedure, with the GA process showing significant improvements over the NACA0015 based on the averaging of the instantaneous XFOIL results.

Aerofoil	Average Tangential force Coefficient	Relative Time Averaged Performance (%)
NACA0015	0.0747	100.00
GA ($\pm 5\%c$)	0.2788	372.23
GA ($\pm 10\%c$)	0.3431	459.3

Table 4-5: Mean Tangential Force Coefficient for 0 degree setting angle comparison for $\pm 5\%c$ Optimisation Bounds and $\pm 10\%c$ Optimisation Bounds profiles

4.2.4 Convergence Conditions

Figure 4-7 shows the variation in instantaneous tangential force coefficient of the basic GA configuration, taken from XFOIL after the GA optimisation procedure at the 6th optimisation position, which demonstrates the variation within each convergence condition across the three runs. The 6th position was selected as it was often found to be the most problematic optimisation position to optimise, and would provide the most hardened conclusions as a result. This position was at an azimuthal angle of 144° or in the case of this turbine the angle of attack was 21.65° with a Reynolds number of 169845 and a Mach number of 0.032508. The conditions were altered by changing the number of generations that the convergence was checked over, with 3 values tested; $n-1$, $n-10$ and $n-25$, such that n is the current generation count. For example, $n-1$ would imply that if the optimal camber of the second generation is less than 1% larger than the first generation's optimal camber profile, the optimisation would be considered converged and would terminate. $n-25$ would comparatively start comparing between the 1st generation and 26th generation.

Figure 4-8 shows the same variations but on a reduced scale, allowing for a more in depth comparison. For the single generation of convergence checking method, it was found that the final camber value was different for one of the tests with one generation of convergence checking. However, the resultant torque value is much lower than the 10 and 25-generation cases, validating the hypothesis that insufficient generations to converge results in a non-global optimised value for camber. The ten generations case had a different conclusion, with each test providing a different torque, showing that the optimisation proves still insufficient generations for optimisation. The 25 generations case had the smallest spread in the torque generation, and the highest overall torque produced, so is clearly the most suitable convergence method for further tests. It is not necessary to run more convergence periods than 25, as the 25 generation case consistently converged, without the need for further generations.

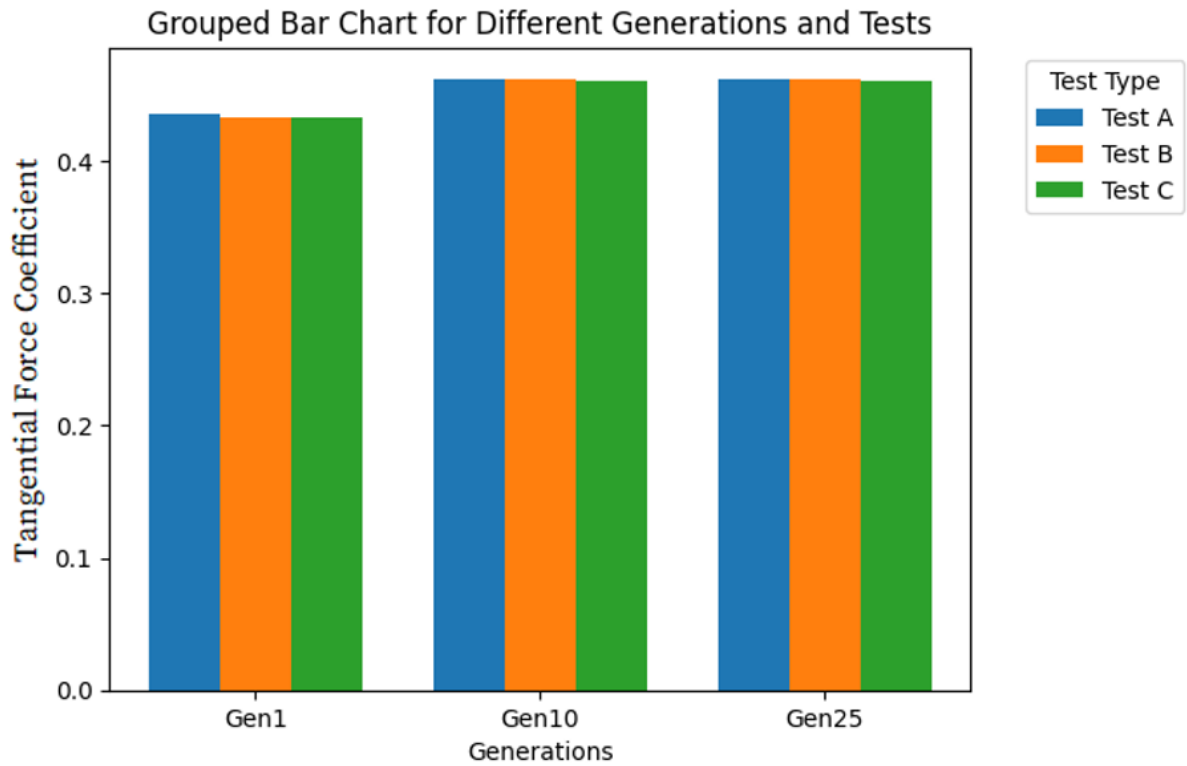


Figure 4-7: Variation in optimised tangential force coefficient for different convergence conditions

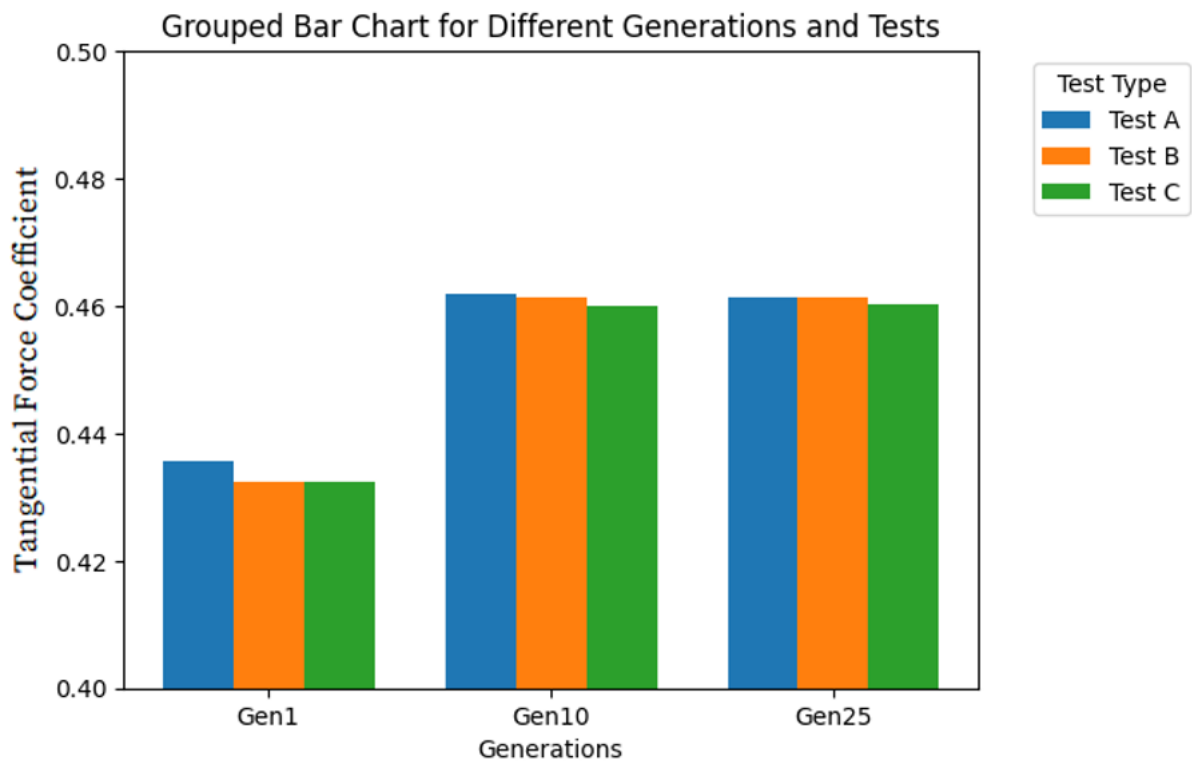


Figure 4-8: Reduced scale view of variation in optimised Tangential Force Coefficient for different convergence conditions.

Figure 4-9 is a plot showing the progression and gradual increase in torque as the optimisation generation count increases. During the period of generation from 15-25, the torque is constant but then increases slightly further thereafter, emphasising that a convergence range of 10 generations is insufficient and would lead to the premature conclusion of the optimisation process.

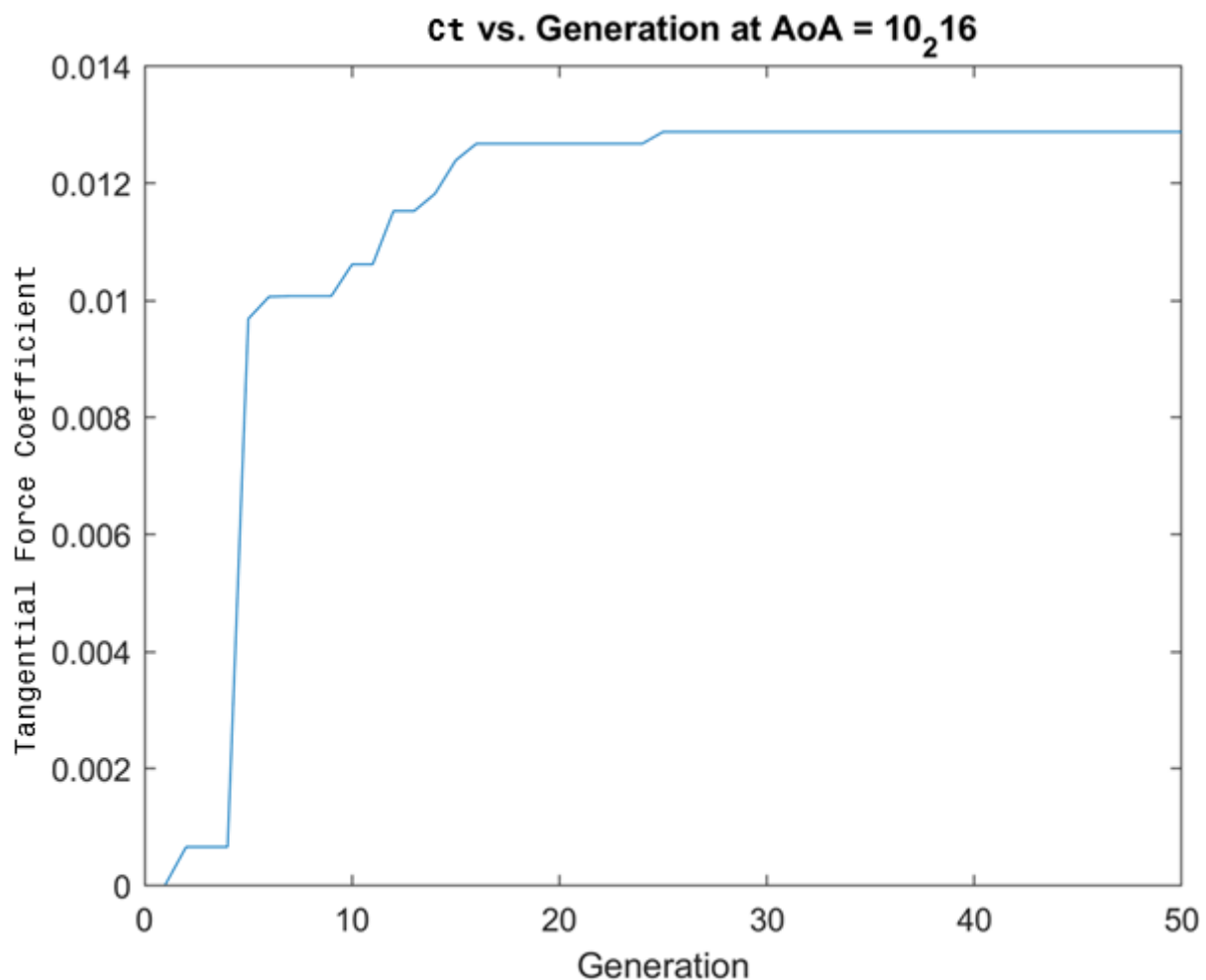


Figure 4-9: Optimisation Convergence Plot for 1, 10 and 25 Convergence Generations.

It is crucial to understand that the setup of the genetic algorithm is not the cause for the variation in optimal camber for each test. This is purely a result of the XFOIL and MATLAB interface creating memory leaks and memory corruption within the testing environment, resulting in very minor variations in the optimal camber. For example, when running the genetic algorithm optimisation process, one of the optimal camber values was found to be 7.8489 %c. Repeating the XFOIL run for this case by simply

specifying the input camber and allowing the scripts to generate an identical set of input files for XFOIL meant that the reported torque in the GA (0.4613) was not the same as that in the standalone XFOIL run (0.4598). As this minor error is consistent throughout the optimisation process, every value would be comparably offset and, can therefore still produce reliable insights into the validity of applying the genetic algorithm optimisation of a deforming VAWT blade camber profile.

The other modification tested in the XFOIL-based genetic algorithm was in the actual convergence condition itself. The difference across the sample range was considered converged when the difference in tangential force coefficient was less than 0.01. This value was selected as it is small enough to ensure the global optima is found, but coarse enough to avoid diminishing returns and excessive generations. If the value for convergence is too small, it could also cause the algorithm to run continuously, as the convergence criteria value (tangential force coefficient difference) could simply be smaller than the variance in the calculated tangential force coefficient. This could also be the case if using a percentage-based convergence criteria, as the criteria could be reached prematurely unless it is set to be very small. For example, the calculated tangential force coefficient is calculated to 6 decimal places (due to the rounding for the lift and drag coefficients reported from XFOIL), so if one were to set the difference to 0.000001, then the likelihood of convergence is significantly reduced.

4.2.5 Sampling Count Analysis

To understand if the GA is properly optimising the overall camber profile, it is important to test at an increased number of sampling points, as it will allow us to infer if the reduced sampling count significantly hampers the potential performance gains. Figure 4-10 shows the camber of the optimisation procedure for 10 optimisation points in the turbine rotation (every 36°) or 20 optimisation points (every 18°). Running additional sampling points shows that the optimisation count makes a negligible impact on the overall camber profile, with the only significant difference being how quickly the optimal camber profile reaches 10% c . As a result, it is decided

that 10 optimisation points is sufficient for generating the camber deformation profile, due to it's very similar performance to the 20 point optimisation, but with half of the computational resource requirements.

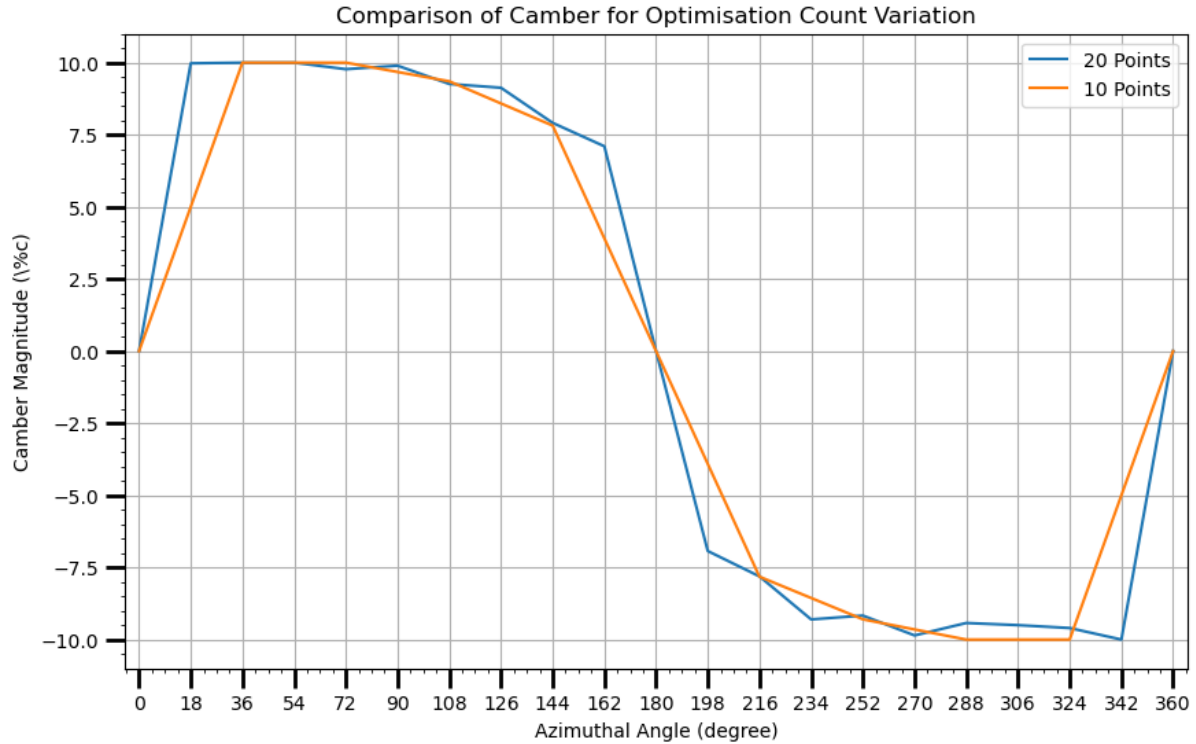


Figure 4-10: Comparison for number of Sampling points between every 36° and every 18° of azimuthal rotation

4.2.6 Reynolds Number Sensitivity

It is crucial to understand how the GA Optimised VAWT blade profile might work in different Reynolds number ranges, as it allows for the method to be applied to the same turbine, but placed in various regions of differing wind conditions. Considering Equation 4.3) below, it is evident that the Reynold number is directly proportional to fluid density (ρ), speed (U) and characteristic length (l), and inversely proportional to the fluid's viscosity (μ).

As the aim is to demonstrate the optimisation's applicability at different Reynold's numbers, any of these values can be changed to alter the testing Reynolds number, but changing density and viscosity is not realistic in situ as it would imply either changing the air temperature/pressure significantly or changing the fluid altogether, which would result in an unfair comparison to the baseline VAWT as the intention is. This leads to changing the characteristic length or fluid velocity, and as the characteristic length is the turbine blade's chord length, altering this would change the solidity of the turbine, making it an unfair comparison. The turbine solidity can be calculated using Equation 4.4); where N is the number of turbine blades, c is the

$$Re = \frac{\rho * U * l}{\mu} \quad 4.3$$

blade chord length and R is the turbine radius.

$$\sigma = N * c * R \quad 4.4$$

As a result, the fluid velocity is the most sensible parameter to change, but it is crucial to maintain the same TSR to have the same blade angle of attack distribution, so the windspeed and turbine rotational velocity are both increased to maintain TSR, but change the tangential velocity that the turbine blade would experience. The rotational velocity and wind speed are intrinsically coupled because of a fixed TSR, so for the sake of simplicity, the input that is changed between the tests is the wind speed.

To evaluate how the Reynolds number impacts the performance gains by the GA, the method was tested at two alternative inlet velocities of 5 and 9 ms⁻¹, with the tip speed ratio kept constant at 2.29. These values were selected, as they allow for an even comparison of the method applied in a high and low windspeed region, but by using a $\pm 2\text{ms}^{-1}$ windspeed difference, there is a sufficiently large difference in Reynolds number between each test to allow for testing more extreme wind speed scenarios. Considering the average wind speed for different regions of the Europe vary significantly, with averages of around 5ms⁻¹ in the south of France in the Bordeaux region to highs of around 9ms⁻¹ in the Peak District [104], these alternative test velocities demonstrate the ability to optimise the VAWT across Europe. The peak

turbine tangential velocity tested was, therefore, 11.45 ms⁻¹, 16.03 ms⁻¹ and 20.61 ms⁻¹ for freestream velocities of 5, 7 and 9 ms⁻¹, meaning the airspeed the blade experiences is 16.45m/s, 23.03m/s and 29.61m/s respectively. As the chord length is 0.225m, using Equation 4.3 with a characteristic length equal to the chord length (0.225m), the velocity is the relative airflow velocity experienced by the blade or aerofoil and the density/viscosity of air at atmospheric conditions, the Reynolds number is found to be 240164, 350699 and 450898 for 5, 7 and 9ms⁻¹ windspeeds.

The expected outcome is that at the azimuthal positions that correlate to higher angles of attack (around 108° ± 36° and 252 ± 36°), there will be a better tangential force coefficient at the high Reynolds test, due to the significant reduction in drag and increase in lift coefficient [105], [106]. Figure 4-11 shows the resultant tangential force coefficient through the turbine's rotation for the NACA0015 and the GA optimised blade profile, run at the lower and higher tip speed ratios. This figure corroborates the hypothesis, as the 9 m/s tests have a higher tangential force coefficient at all points within the GA Optimised Blade profiles and show a significantly higher tangential force coefficient at 72° azimuth due to the reduced separation and higher lift at high Reynolds numbers.

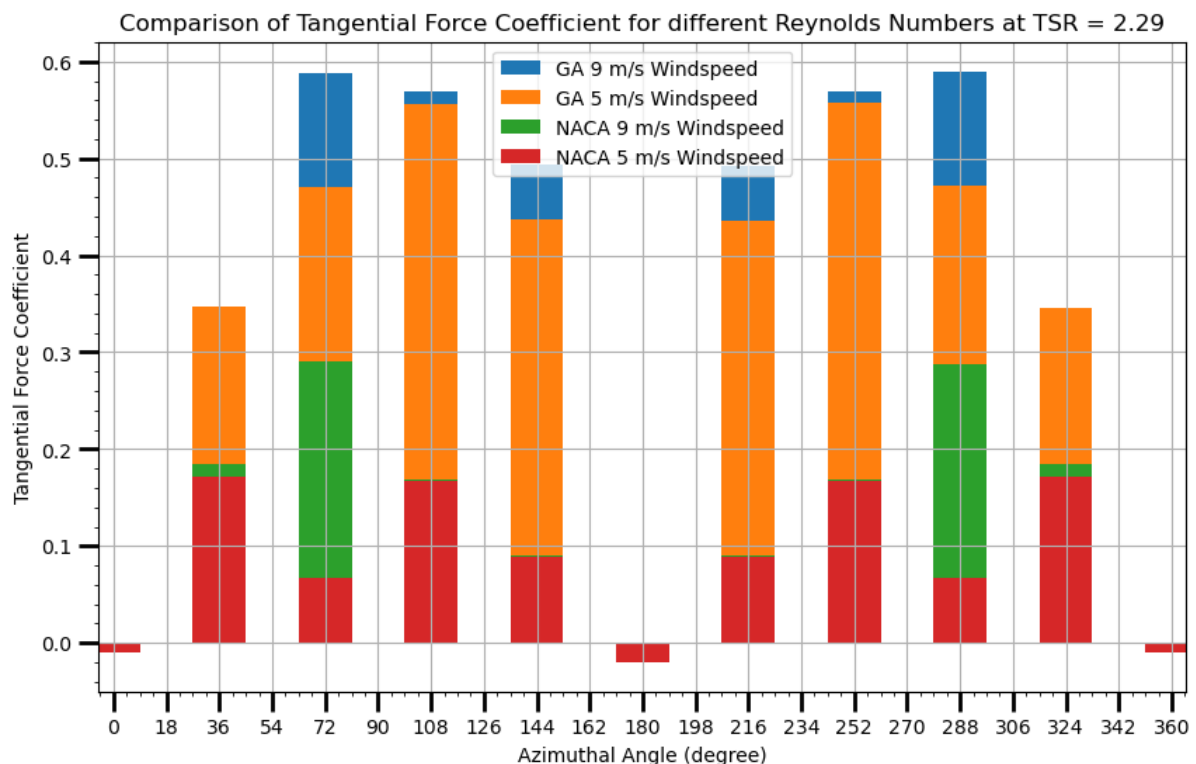


Figure 4-11: Impact of Reynolds Number on Tangential Force Coefficient

4.2.7 Tip Speed Ratio Sensitivity

It is also crucial to understand the limits of the application by testing at multiple tip speed ratios. As mentioned in Section

VAWT Aerodynamic Theory, in low tip speed ratio cases, the angle of attack is likely to be larger than the stall angle of a NACA0015 symmetric aerofoil for most of the turbine's rotation. By testing the methodology at two additional tip speed ratios, 1.5 and 3.5, we can infer if there is an inherent tip speed ratio or Reynolds number-based impact on the overall GA performance increase. Figure 4-12 displays the variations in angle of attack and compares them for all 3 TSRs and their respective Reynolds number variations through the turbine's rotations.

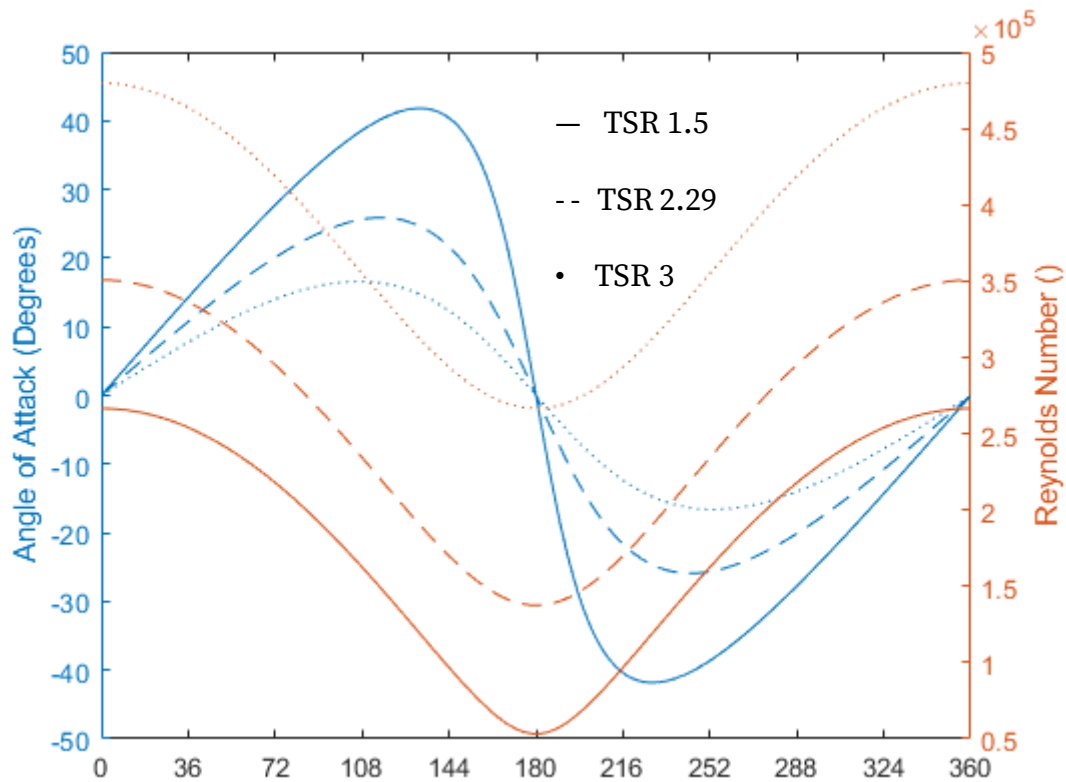


Figure 4-12: Angle of Attack and Reynolds Number Variation with respect to Azimuthal Angle for TSRs of 1.5, 2.29 and 3

Looking at the results from Figure 4-13, some interesting conclusions can be inferred. For example, it was expected that the lowest TSR case (TSR = 1.5) would likely have

the worst performance around the peak angle of attack (at $\phi \cong 100^\circ$) as a result of the aerofoil being significantly stalled and also the reduced Reynolds number which would suggest the stall angle is reduced [107]. It should be noted however, that the higher angle of attack at low azimuthal position explains why the first optimisation point has a higher tangential force coefficient than the higher TSRs before quickly stalling and performing worse. Figure 4-14 displays the camber profiles generated for the three tip speed ratios that were tested; 1.5, 2.29 and 3. The sharp drop in camber magnitude for the TSR 1.5 scenario is a result of the very high angle of attack (38.61°) that the blade experiences at that azimuthal position (108°) for a TSR of 1.5, causing significant separation and complete stall of the blade. At the higher TSRs due to the lower angle of attack and higher Reynolds number the blade experiences, there is less separation, and as such the blade to be less sensitive to the adverse pressure gradient of the high camber.

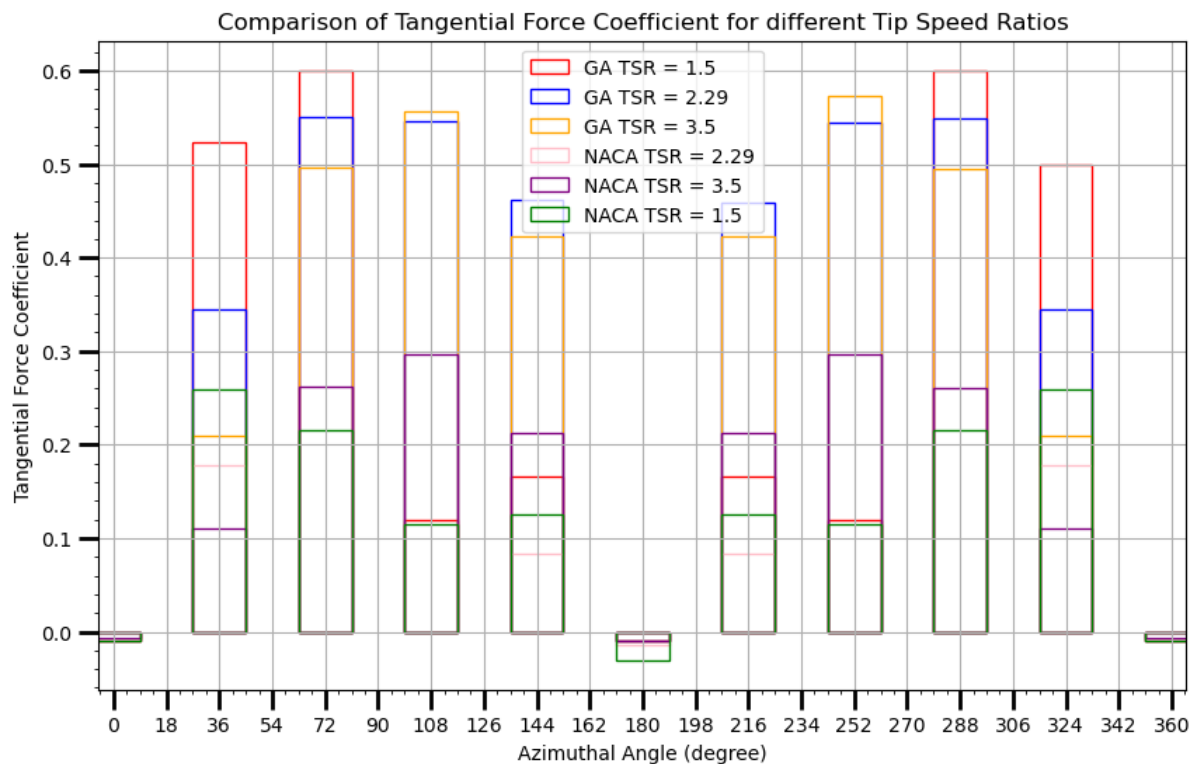


Figure 4-13: Comparison of Tip Speed Ratio on the Performance of the Optimised VAWT Camber profile. attack formulations

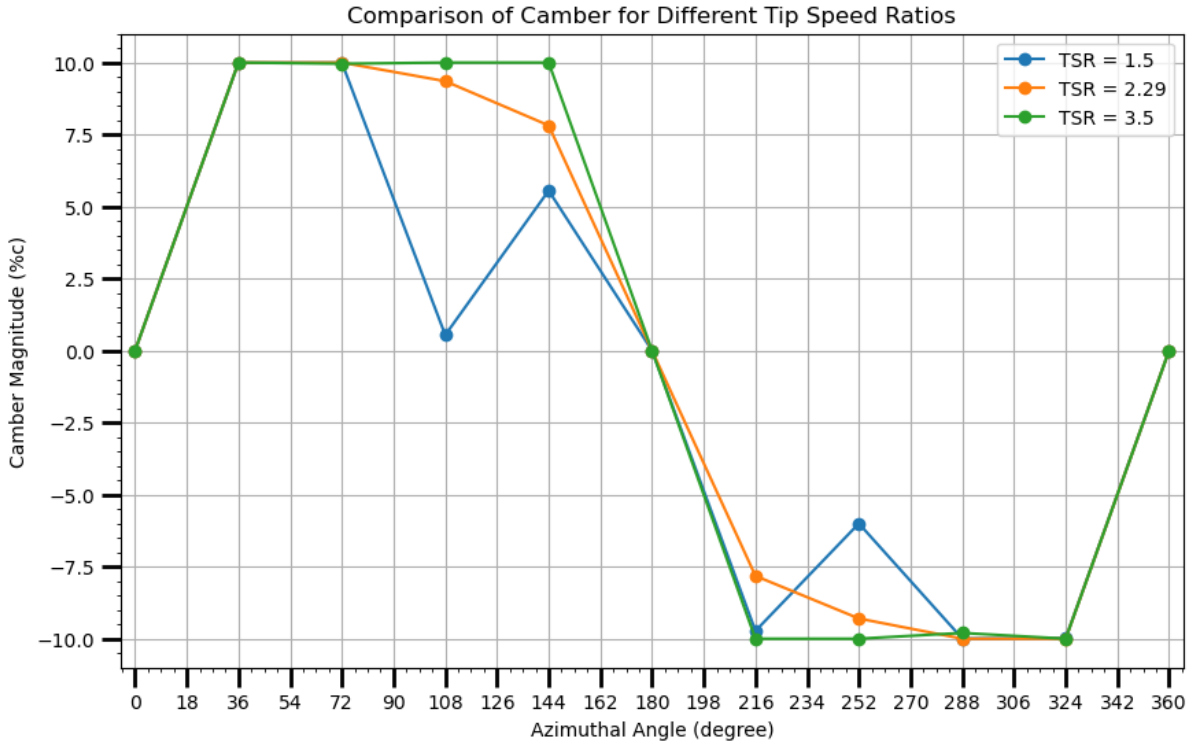


Figure 4-14: Comparison of Camber Profile of the VAWT Optimisation process at multiple tip speed ratios

4.2.8 Alternative Angle of Attack Formulations

According to the formulae derived in Section 0, the turbine angle of attack will have a periodic angle of attack magnitude (i.e. at $\phi = 20^\circ$ the AoA is the same as at 200° , with opposite sign). This is not realistically true, due to the wake of the turbine blade, thus causing fluctuations in the velocity magnitude and swirl. Considering the Starting Vortex theory, there will be an induced vorticity from the lift generation of the turbine blade, imparting the swirl component on the blade wake and increasing turbulence intensity in this region of high circulation. The reduction in velocity is because of the turbine extracting energy from the freestream flow, without sufficient time and opportunity to diffuse with the freestream velocity and increase before the downstream turbine region.

To ensure the GA method's breadth of applicability, alternative angle of attack profiles are also tested to bridge the gap between the theoretical angle of attack and the true angle of attack experienced by the rotating VAWT blade due to blade wake

interactions. These include the application of an angle of attack profile derived from work in the current literature by El Sakka et al. [80] and using a mean velocity from a non-deforming turbine simulation.

The work of El Sakka et al. [80] uses sampling points just ahead of the turbine blade profile to estimate the incoming flow velocity magnitude on the turbine blade. These relative flow velocities are then utilised in calculating the angle of attack that is experienced by the turbine blade. This is completed for both a 0° setting angle and a 6° setting angle.

Figure 4-15 → Figure 4-18 display the velocity contours around the NACA0015 turbine blade profile from 150° to 180° at a TSR of 2.29. A large vortex is seen to be carried downstream, which causes a large reduction in tangential force generation at 150° - 180° azimuthal angle. This separation bubble is a combined result of the large angle of attack and effective heaving of the aerofoil at the start of this range, and due to the relatively low TSR, the blade is effectively unable to escape the large recirculation, severely reducing the circulation generated by the blade and in turn reducing the lift coefficient and increasing pressure drag significantly.

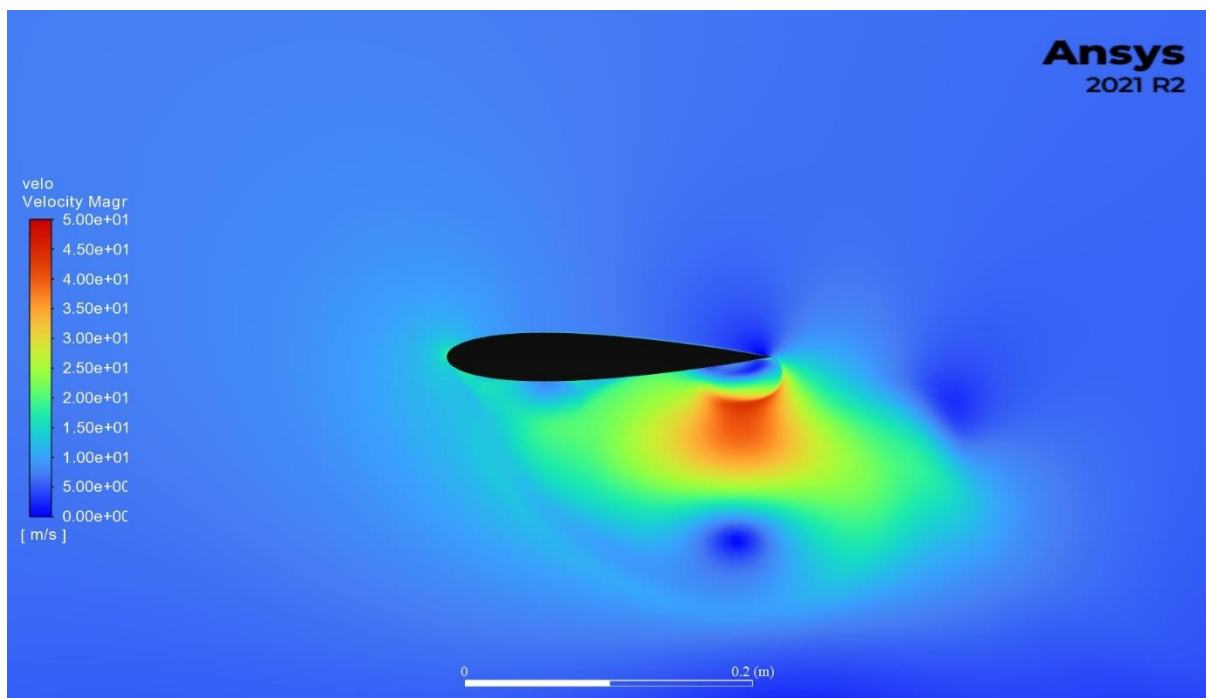


Figure 4-15: Velocity Contour of NACA0015 Blade Profile at 150° Azimuthal Angle

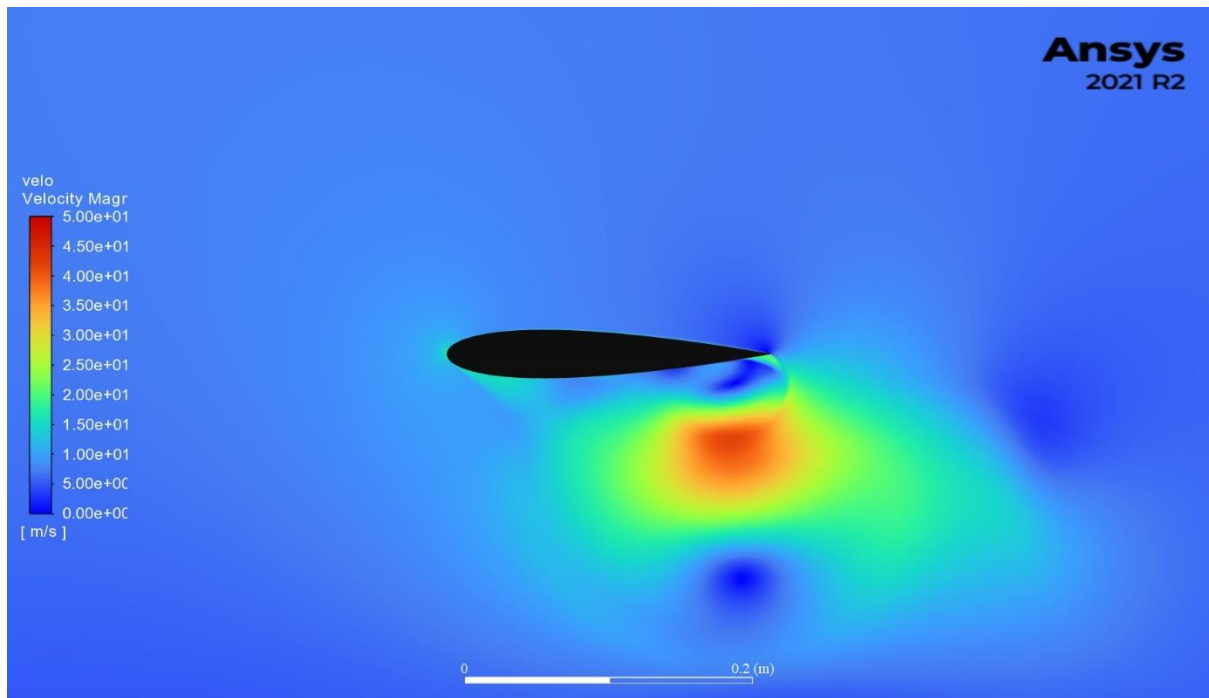


Figure 4-16: Velocity Contour of NACA0015 Blade Profile at 160° Azimuthal Angle

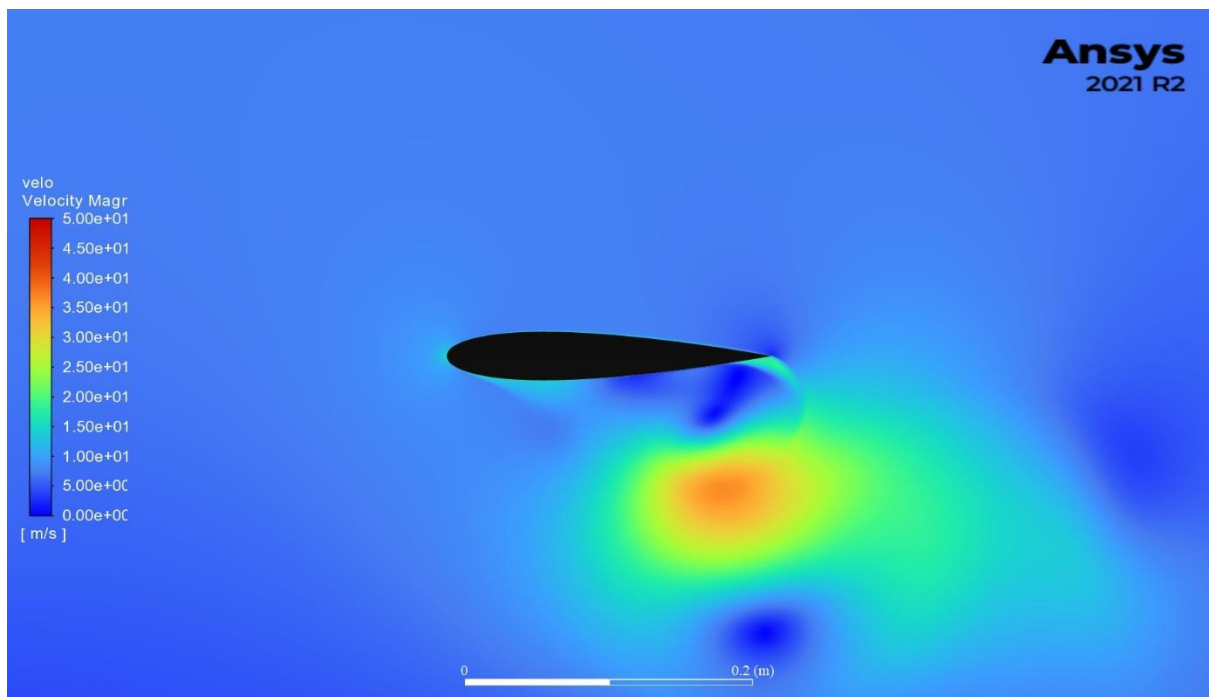


Figure 4-17: Velocity Contour of NACA0015 Blade Profile at 170° Azimuthal Angle

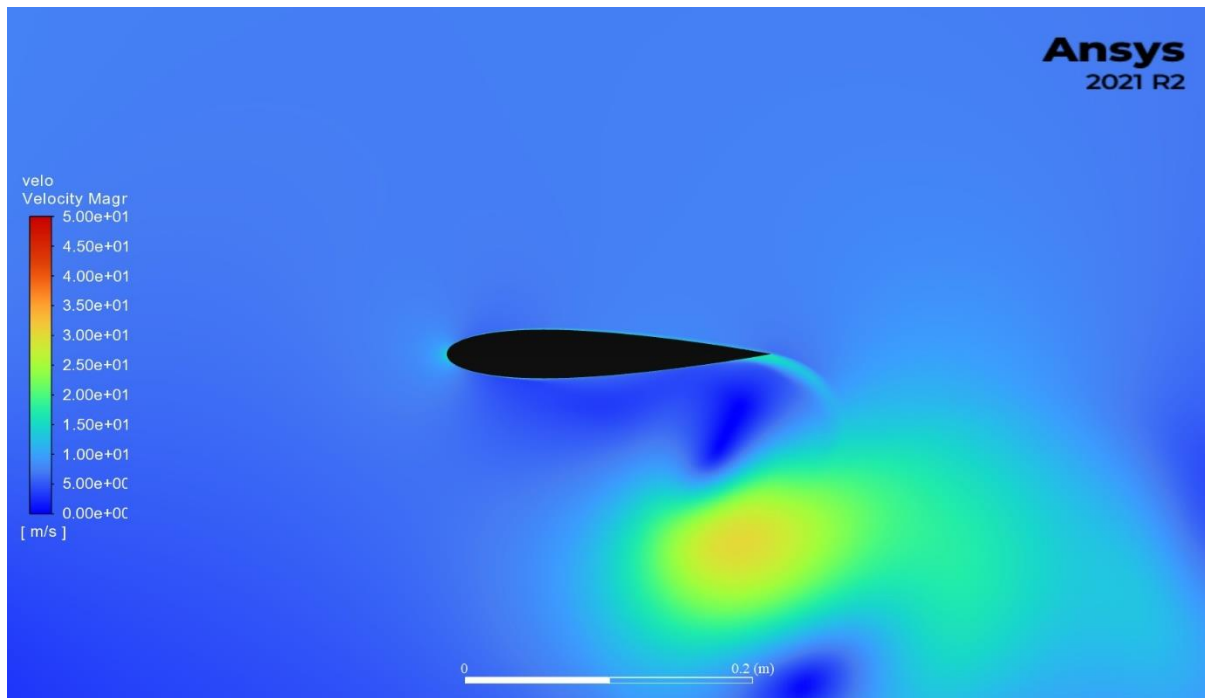


Figure 4-18: Velocity Contour of NACA0015 Blade Profile at 180° Azimuthal Angle

This in turn causes the flow velocity magnitude to be higher in these regions, hence the sudden increase in velocity at the radial distance from -0.25m to -0.75m seen in the velocity profile in Figure 4-19.

Figure 4-19 shows the flow velocity along a sampling line spanning from 0° azimuth vertically down to 180° azimuth, taken from the datum NACA0015 VAWT simulation. By plotting the flow velocity magnitude against radial distance (top to bottom of the turbine) from the non-deforming turbine simulation with a 0-degree setting angle, it is possible to see the effect of the flow velocity reducing due to the extraction of energy by the turbine. The small gap in the centre of the line is due to the presence of a rotating circular wall in the CFD setup that would replicate the shaft of the turbine.

Velocity Plot at Turbine Centre Line

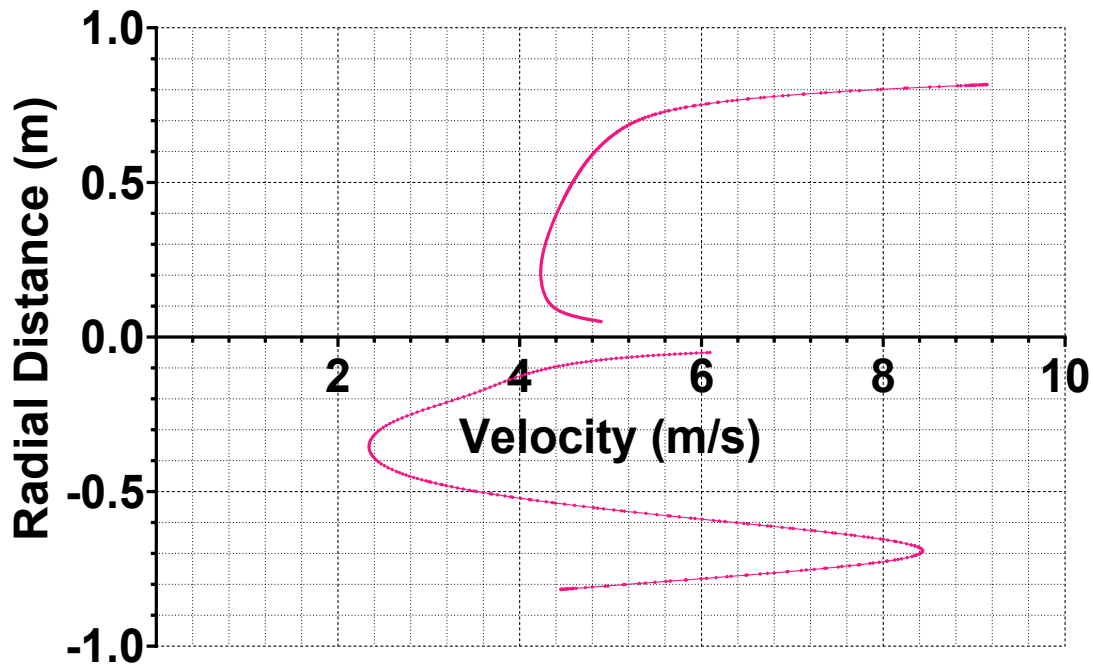


Figure 4-19: Turbine Velocity Plot at the Centre Line ($\phi = 0^\circ$ to 180°) extracted from the Baseline NACA0015 VAWT CFD Model in ANSYS Fluent

By taking the mean velocity at this centre line, it is found that the freestream velocity incident on the blade from $\phi = 180^\circ$ to 360° is 5.22 m/s, or 74.6% of the initial freestream velocity. This will effectively increase the tip speed ratio in the downstream section, as the turbine maintains a constant rotational speed. As a result, the angle of attack will be reduced.

It should be noted that both the alternative angle of attack profiles (Induced Velocity and Elsakka et al. findings) tested are based on the symmetric NACA0015 aerofoil at a TSR of 2.29. If one was to optimise the camber profile based on the true angle of attack in the downstream portion of rotation, it would be a very involved and computationally expensive setup. This could be achieved by optimising the upwind stroke of the turbine, and feeding that optimised camber profile into the CFD study detailed in Chapter 3.3, with the angle of attack approximation technique employed by Elsakka et al. [80] to calculate the angle of attack experienced by the blade for each optimisation position in the downstream period of rotation. This would be iterated for every optimisation position in the downstream, to ensure that the blade-wake

interaction's influence on the angle of attack is properly modelled. As a result, this would result in taking the fast turnaround of the XFOIL optimised methodology and significantly increasing the total computational time to create the full optimised GA profile. This is outside the scope of this chapter, but is a promising path for future research, to potentially further amplify the benefits of using the GA to optimise the turbine blade's camber profile.

The resulting Induced Velocity angle of attack method is calculated using the new mean freestream velocity in the downwind rotation (180° to 360°), to calculate the new effective TSR in this period and maintaining the same rotational velocity of the turbine. As the mean velocity was found to be 5.22m/s; and the tangential blade velocity is 16.03m/s, the new TSR is 3.0708. This new TSR is used to calculate the theoretical angle of attack and is seen from $180^\circ \rightarrow 360^\circ$ azimuthal position in Figure 4-20. As the TSR is higher in this region, the angle of attack the blade experiences can be seen to be lower.

A comparison of the three proposed angles of attack formulations is presented in Figure 4-20. El Sakka shows a dip in the AoA curve at 252° , and this is likely due to the increased flow velocity impacting on the blade from the upstream vortex shedding interaction mentioned earlier. Another point of interest in Elsakka et al.'s findings is that the angle of attack is not 0° at 0° , 180° and 360° azimuth; this could be a result of the inductance effect of the turbine that has been found in other works [108].

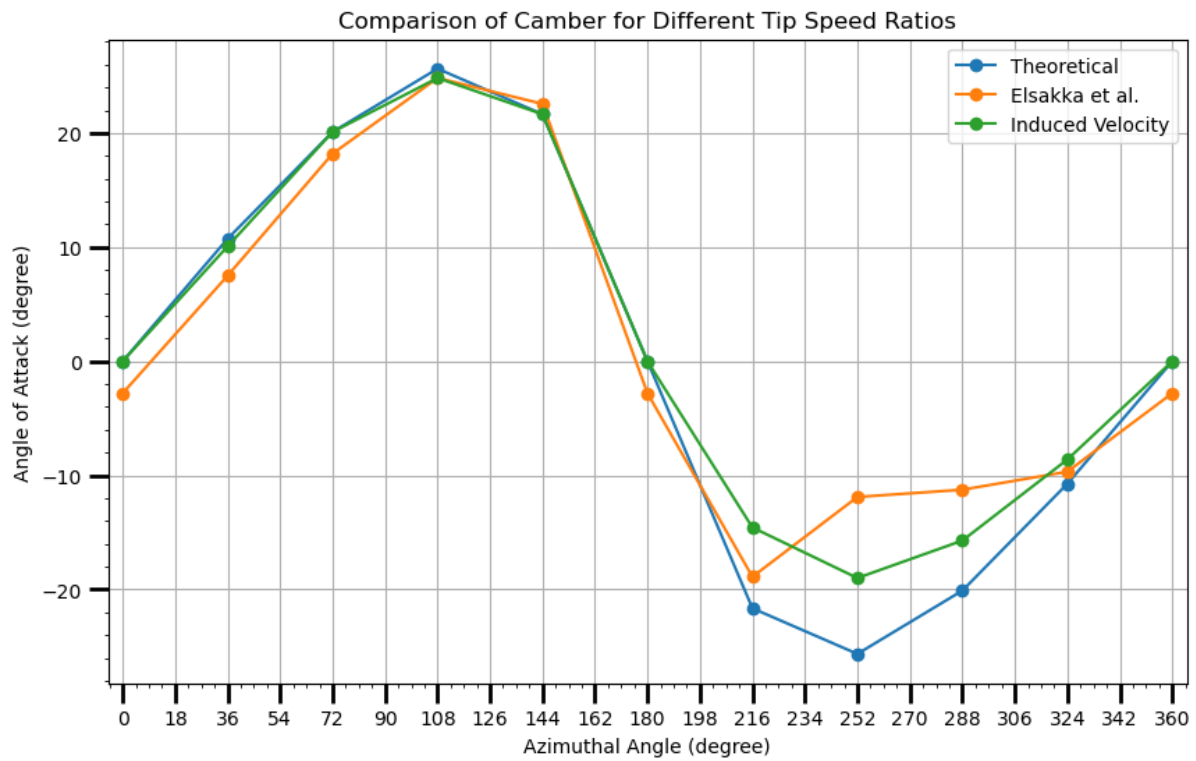


Figure 4-20: Angle of Attack ($^{\circ}$) Comparison for different quantification methods.

Interestingly, Figure 4-21 (the camber profile comparison for the three angle of attack definitions) shows that there is very little variation in the optimal camber selection among the three different angle of attack methods. One common trait however appears to be that the lower the angle of attack experienced by the blade, the larger the optimal camber is. This is due to the 0-lift angle being higher for the higher camber aerofoils, thus producing more lift at the lower angles of attack.

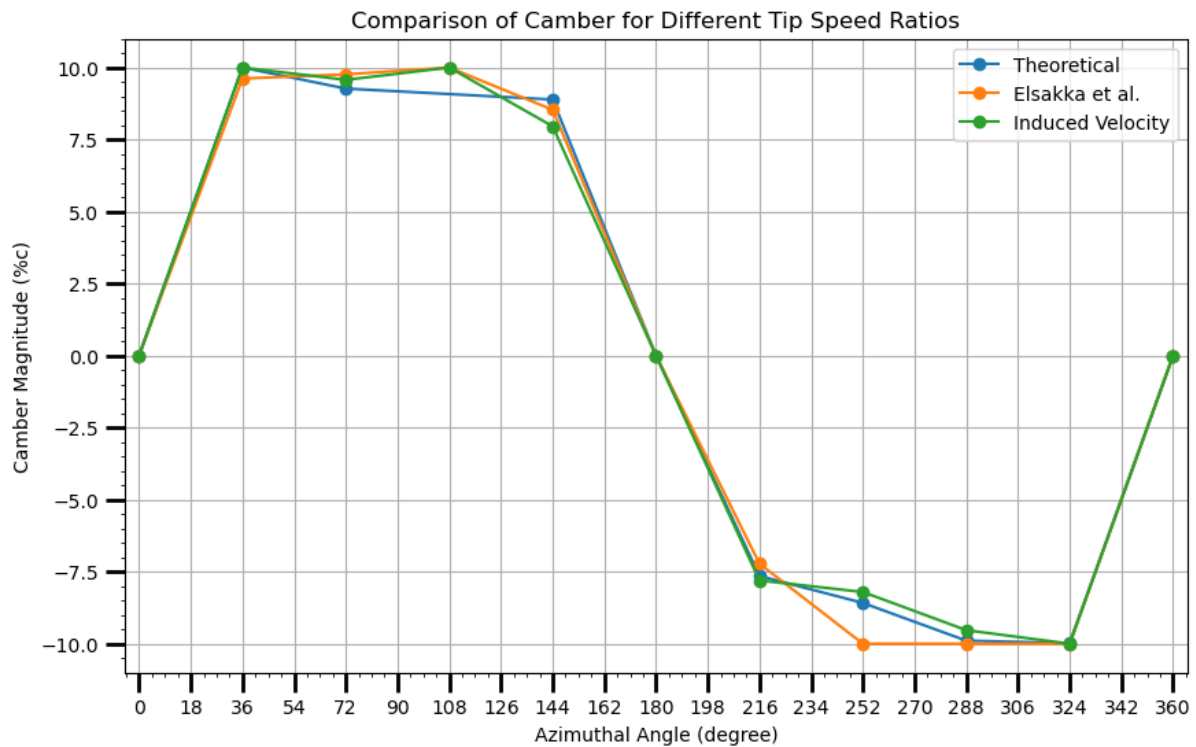


Figure 4-21: Optimised camber profile for the three variations of the angle of attack.

This increase in lift directly translates to an increase of the tangential force coefficient, and in turn the power coefficient.

Figure 4-22 and Figure 4-23 show the resultant tangential force coefficient for the Elsakka et al. [80] angle of attack definition, as shown Figure 4-22 and the theoretical induced angle of attack definition, in Figure 4-23 respectively. The induced method saw an increase in average tangential force coefficient from 0.06883 to 0.3337, signifying a 484.82% increase. A comparable result is seen for the angle of attack of El Sakka et al. [80] with an increase from 0.09961 to 0.3032, corresponding to a 304.39% increase. This is fundamentally important, as the tangential force coefficient is the force that drives the turbine's rotation, and results in the vertical axis wind turbine's ability to generate power from rotating the rotors in the attached generator.

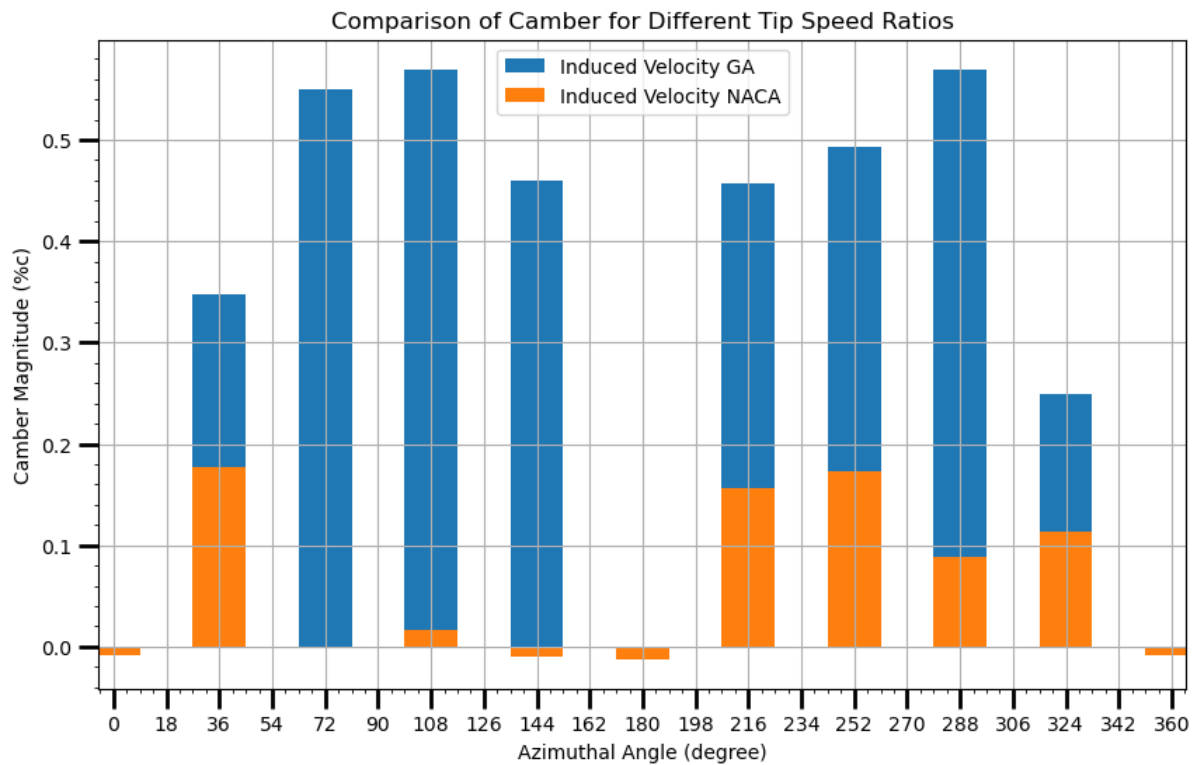


Figure 4-22: Tangential force coefficient comparison between the optimised blade and the NACA0015 blade profile for the Induced angle of attack definition

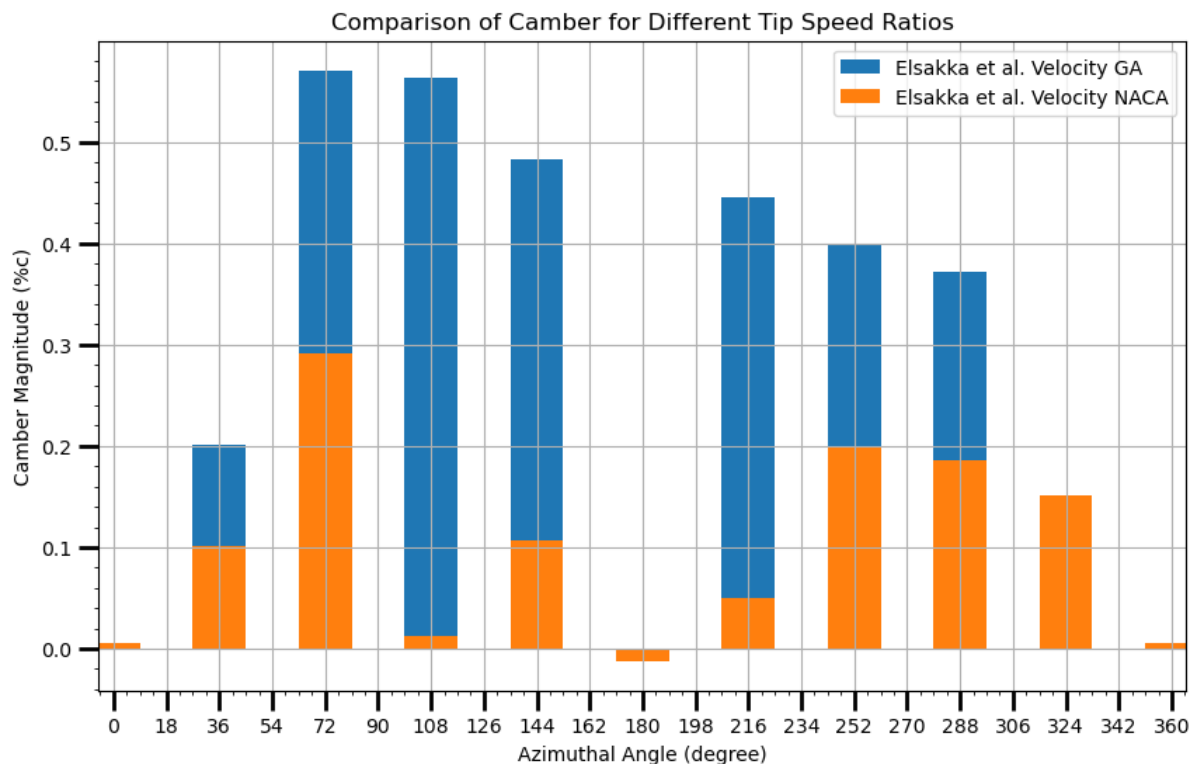


Figure 4-23: Tangential force coefficient comparison between the optimised blade and the NACA0015 blade profile for the Elsakka et al. angle of attack definition.

In the more advanced angle of attack models (induced angle of attack and Elsakka et al.'s model) there is a clear trend of a reduction in the tangential force coefficient in the downstream portion of the rotation. This effect was not evident for the theoretical angle of attack, and this is a result of a combination of factors. These factors include the reduced kinetic energy in the freestream airflow in the downstream section, resulting in less power being available for the turbine to extract, compounded with the reduced angle of attack in the regions because of the increased effective tip speed ratio. The lower angle of attack implies lower lift coefficient from the turbine optimisation results, even using the same aerofoil profile, as the tangential force is directly proportional to the lift force.

4.3 CFD Analysis of the Genetic Algorithm Optimised Deformation Profiles

It is crucial to understand how the Genetic Algorithm performs in the scope of the dynamic VAWT model, to consider how the blade wake interactions, and transient flow effects the validity of the XFOIL based GA optimisation when applied in situ of a VAWT. This is achieved by applying the camber profile that the GA optimisation procedure creates within the turbine that was detailed in Section 3.3, and the results of that CFD procedure are discussed in the following sections. The profiles were implemented using a custom made User-Defined Function subroutine in ANSYS Fluent, enabling the dynamic deformation of the turbine blade profile as the blade rotates.

4.3.1 Genetic Algorithm Evaluation in CFD

Utilising the force report feature in ANSYS Fluent, it was straightforward to extract the single blade moment generation at any point in the turbine rotation. For the initial comparison, the different angle of attack model optimised camber profiles are compared at the azimuthal angles that the optimisation procedure was evaluated at (every 36° from $0^\circ \rightarrow 360^\circ$).

The blade moments are presented in *Table 4-6*, the cell highlighting indicates an improvement or decrease in performance in comparison to the NACA0015 symmetrical aerofoil. A red highlight implies a worse performance and green indicates a better performance than the non-deforming NACA0015 aerofoil. There is a significant difference in the torque and therefore power generation between the symmetrical NACA0015 and GA optimised aerofoil profiles for both normalised and standard XFOIL solution methods. Both methods create a significant increase in the mean torque, with less variance in torque generation throughout the turbine rotation. It is interesting to note that the genetic algorithm has a worse performance than the symmetric aerofoil in the upstream part of the rotation ($0^\circ < \phi < 180^\circ$), but it has significant improvements in the downstream section. There could be a few possibilities for this upstream reduction; firstly, due to the angle of attack differences between the real turbine and the values provided in XFOIL, causing an early onset of stall. Secondly, it could be the result of an effectively increased blockage of the turbine due to the improved downstream performance which could be slowing the upstream flow velocity or changing the flow angle the blade experiences.

Azimuthal Angle (°)	Moment (Nm)			
	NACA0015	Normalised XFOIL GA	Induced AoA GA	El Sakka et al. AoA GA
0	-1.3351	-1.4026	-1.4392	-1.3405
36	2.3226	-1.3741	-1.172	-1.2348
72	15.6634	15.465	15.5546	15.5002
108	6.8144	6.7249	-2.7976	4.0255
144	-6.6655	1.2623	2.8307	1.532
180	-2.1934	-1.1929	-0.2343	-0.2511
216	0.0776	4.9497	6.6008	8.5072
252	0.5467	2.8585	0.1028	1.3361
288	0.7657	1.0837	0.8153	1.0556
324	0.8911	1.7279	1.5963	1.8321

Table 4-6: Moment Force Comparison for multiple Genetic Algorithm setups

Figure 4-24 shows the continuous moment generation data for a single blade in the turbine CFD simulation, for the results seen in *Table 4-6*. The distinct second peak present in all four genetic algorithm methods reinforces these findings, and a negligible moment for the symmetrical non-deforming NACA0015 aerofoil.

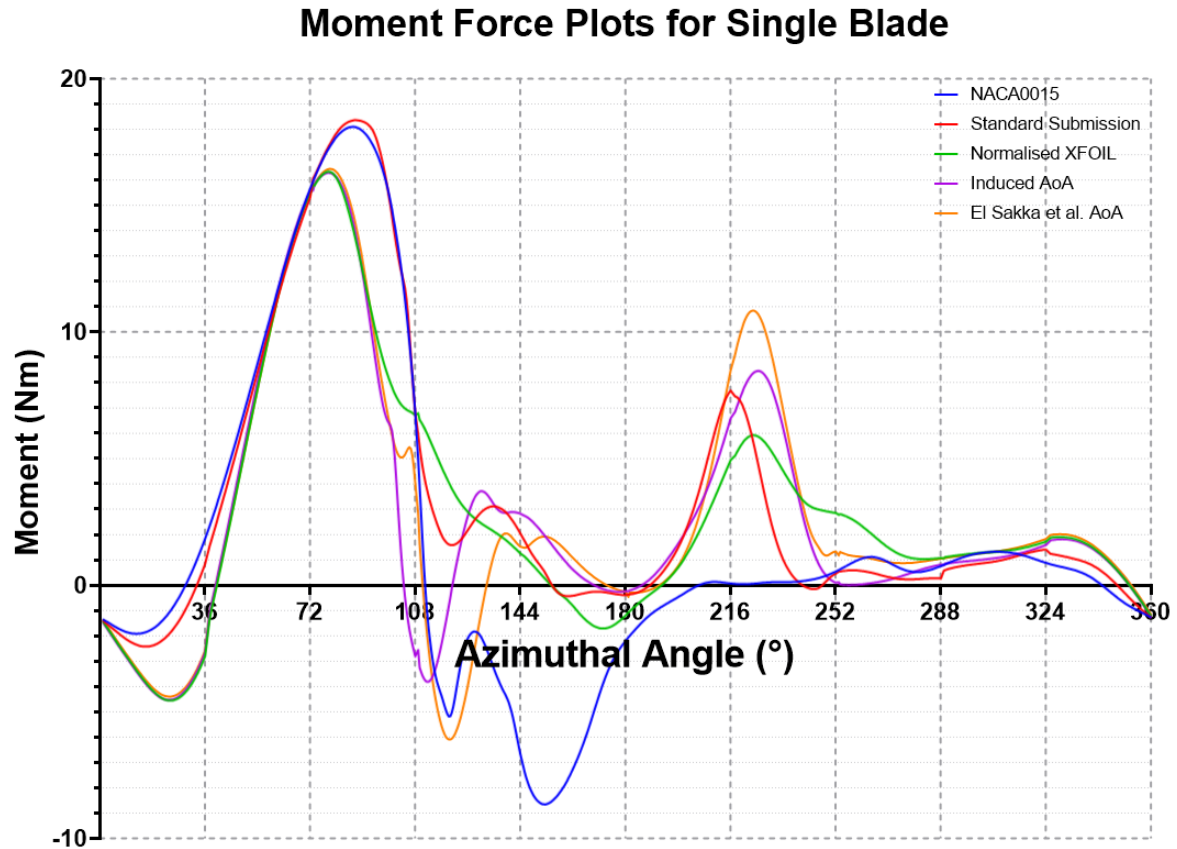


Figure 4-24: Moment comparison for the different Angle of Attack models from a transient turbine CFD simulation

The lower peak is likely attributed to the reduction in freestream velocity experienced by the incoming blade as seen in the velocity contours of the optimised turbine profile simulation compared to the non-deforming NACA0015 profile, in Figure 4-25 (a) and (b). It is shown by the increase in velocity upstream of the blade at 0° azimuthal angle in the NACA0015 datum simulation compared to the Optimised GA deformation profiles, which would provide more energy to the turbine blade, allowing for greater torque production. This arises due to the increased energy extraction from the preceding cambered blade.

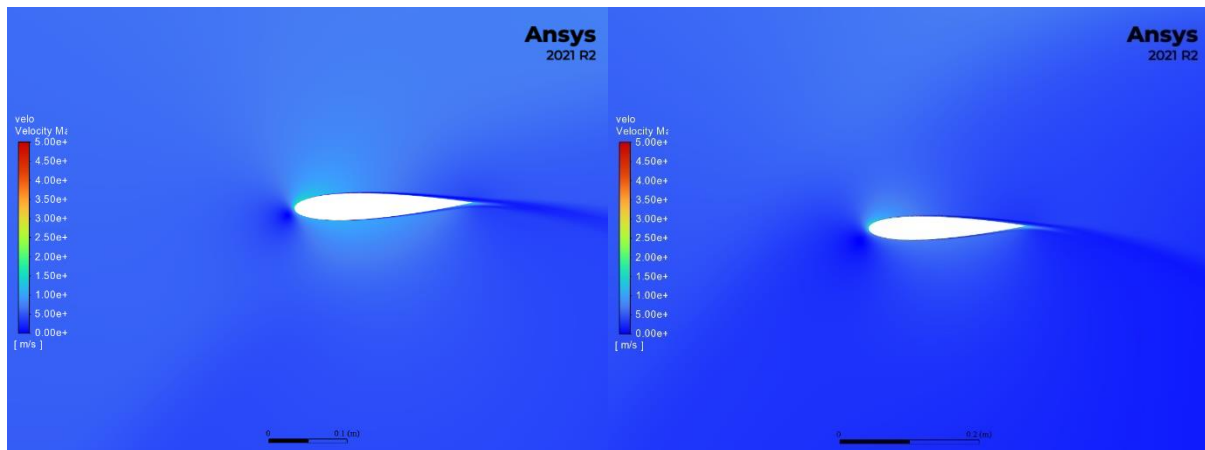


Figure 4-25: Velocity Contours from 0->50m/s at $\phi = 0^\circ$ for (a) NACA0015 Non-Deforming & (b) XFOIL Normalised GA Deformation Profile Turbine.

The alternative angle of attack characterisations (Induced and El Sakka definitions) and the Standard XFOIL method have a similar shape to the original NACA0015 around the region of $100^\circ - 144^\circ$, with the drop in the moment force before a rapid increase again, followed by a gradual dip again. This trough is much shallower for the Standard XFOIL method with the standard angle of attack, but the shape is still similar. This does not appear to be the case for the Normalised XFOIL case, as it has a much more gradual reductions in the moment force, the Normalised profile only briefly reached a negative moment force at an azimuthal angle of 160° .

Comparing the variation of the moment plot allows for a much better understanding of the fluctuations experienced by the blade and provides a wholistic rather than localised viewpoint.

From Table 4-7

	NACA0015	Normalised XFOIL GA	Induced AoA GA	El Sakka et al. AoA GA
Median Moment Value (Nm)	0.1888	1.622	1.136	1.294
Mean Moment Value (Nm)	1.615	2.842	2.577	2.654

Standard Deviation of Moment Value (Nm)	6.293	4.702	4.824	5.105
Mean: Standard Deviation Ratio	0.2566	0.6044	0.5342	0.5200

Table 4-7: , which shows some of the statistical properties of the turbine's moment distribution, it is evident that all the genetic algorithm method variations tested allow for much lower standard deviations in the moment force, combined with higher median and mean moments than that of the non-deforming NACA0015. The significance of this finding translates to a reduction in the overall blade fatigue and an increase in the life of the blade due to the more constant blade loading. Additionally, the wear that is commonly found in the shaft's load bearing would be reduced as there would not be a large resistance in certain regions of the bearings.

	NACA0015	Normalised XFOIL GA	Induced AoA GA	El Sakka et al. AoA GA
Median Moment Value (Nm)	0.1888	1.622	1.136	1.294
Mean Moment Value (Nm)	1.615	2.842	2.577	2.654
Standard Deviation of Moment Value (Nm)	6.293	4.702	4.824	5.105
Mean: Standard Deviation Ratio	0.2566	0.6044	0.5342	0.5200

Table 4-7: Statistical Analysis of the Turbine's Moment Comparisons

The mean moment values are significantly higher, with Normalised GA presenting the largest value, followed by the Elsakka et al. AoA definition, and finally the Induced AoA GA. The trend is different if we compare the standard deviation, with a

lower standard deviation signifying that more of the turbine's rotation produces a moment close to the mean value. The lowest variance is experienced by the Normalised GA, then the Induced AoA GA and finally then the El Sakka et al. AoA GA.

By comparing the ratio of the mean to the standard deviation, the conclusion can be made of the wholistically best method tested. The larger the ratio, the larger is the mean or the smaller is the standard deviation. The best overall method tested is therefore the Normalised Method using the theoretical angle of attack. This value did not have the highest mean value, but the median value was the highest and the standard deviation the lowest, resulting in the conclusion that it is the best method overall to employ in a real life VAWT.

Figure 4-26 depicts a violin plot of the moment distributions for the compared optimisation characterisations, with the thickness at any one torque value representing the relative commonality of said torque value. One can therefore conclude that the thickest section of the graph indicates the mode of the data, which is easily read as the Y-Value of the white dot seen in the plot. The tips of the chart show the minimum and maximum values of the distribution, in this work that translates to the minimum and maximum torque experienced by the VAWT in it's rotation. The smaller the difference between the peaks, the more constant the torque experienced by the blade, so the perfect scenario would be a perfectly flat line at a single torque value as that would eliminate any vibrational fatigue in the structure, improving the turbine blade's lifespan. Finally, the rectangular box with the two thin strips is known as a box and whisker plot. The white point represents the median, the black rectangle is the interquartile range of the distribution and the whiskers represent 1.5x the interquartile range.

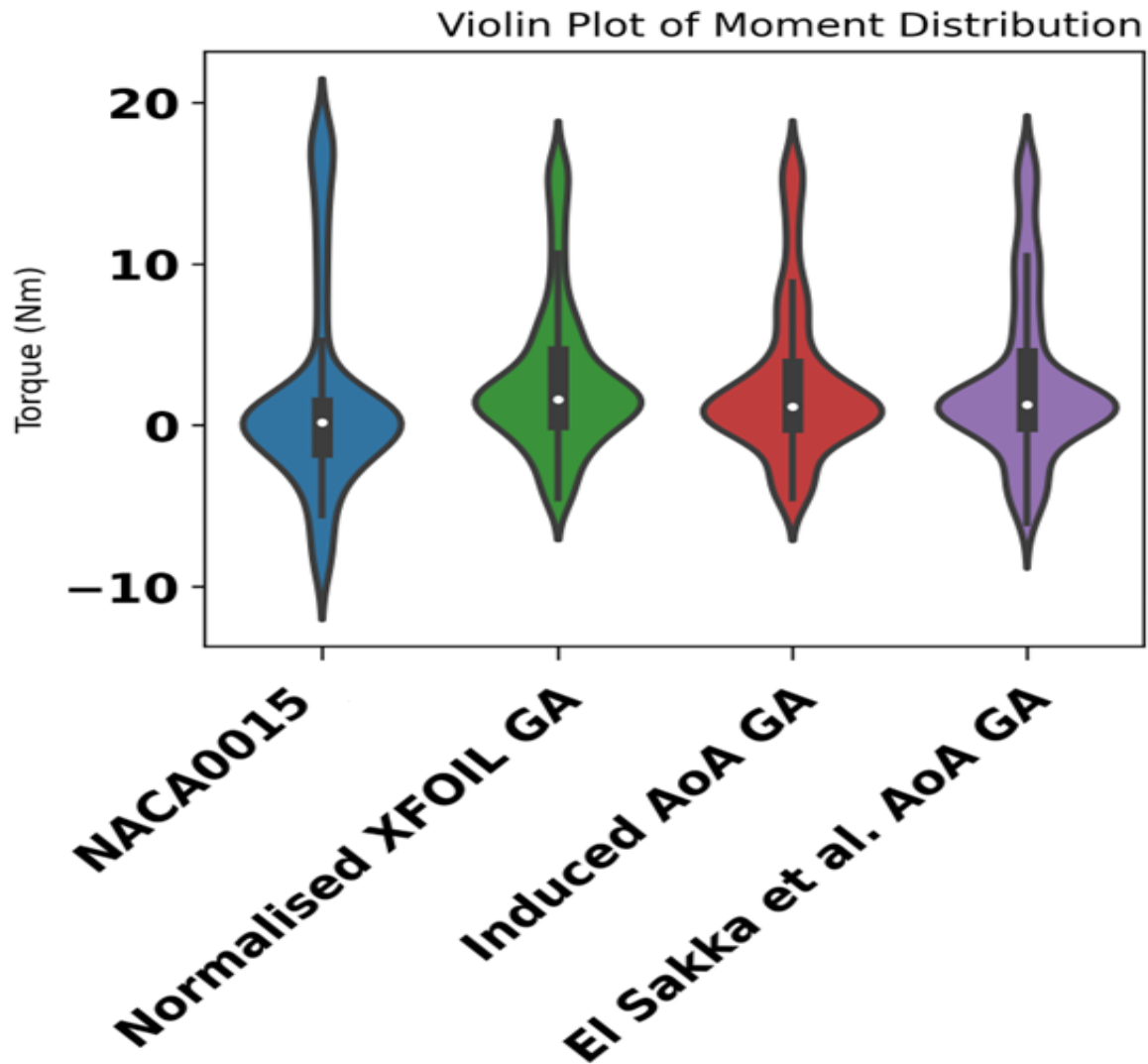


Figure 4-26: Violin Plots representing the Distribution in Torque for the different Variations of the XFOIL Based Genetic Algorithm

4.4 Summary of Findings

This investigation shows the viability and power of using a Genetic Algorithm optimisation technique to characterise the deformation profile of an actively deforming vertical axis wind turbine blade. The use of a GA with both a 5%c and 10%c limit on the GA search bounds showed substantial improvements in the turbine's tangential force coefficient. When evaluated in XFOIL, the optimisation technique showed large increases in the time-averaged torque coefficient, enabling an increase in the overall power generation of the turbine. The optimised aerofoil had a higher torque coefficient value at all the sampling positions when compared to the

symmetric NACA0015. These conclusions are confirmed when applied in CFD by creating a user defined function to deform the aerofoil profile to the optimised camber magnitudes as it rotates in the VAWT scenario. The mean and standard deviation of the moment force is shown to be significantly higher when the genetic algorithm profile is applied. The lowest improvement in standard deviation of the moment is 19.9% and the lowest improvement in mean moment is 59.1%!

The optimal Genetic Algorithm settings was found to be for the turbine consisting of a 10%c bound, with the Normalised XFOIL geometry setup and the theoretical angle of attack. The corresponding tangential force coefficient and camber profile is seen in Figure 4-5 and Figure 4-6 respectively.

The downfalls of utilising XFOIL as the flow solver method were multifaceted. These include its inability to capture the flow properties when there is large flow separation, it also does not capture the upstream flow induction effects of the turbine in the downstream sections. Fixing these issues are topics of improvement that will assist to further demonstrate the strengths of the GA methodology and even further improve the turbine performance. As such, the recommended further work is to implement the GA technique in a CFD solver, i.e., the ANSYS Fluent software for a rotating turbine situation to better understand and incorporate the effects of the VAWT's dynamic flow effects.

5 Transient Genetic Algorithm

Optimisation of a VAWT Blade Profile

5.1 Introduction

The previous chapter highlighted some of the weaknesses of using XFOIL in the scope of VAWT blade aerodynamic performance calculation, due to many failures in calculating lift and drag for high angles of attack. Furthermore, the limitations of the theoretical VAWT aerodynamics based on first principles showed that even more complex angle of attack models failed in capturing the dynamic flow conditions that the blade experiences. This chapter overcomes these issues, and builds on the successes, through the implementation of several additions to the methodology, including utilising transient rotating VAWT CFD simulations instead of XFOIL to evaluate the performance of each genotype, having a coupled time marching of the turbine profile such that the previous optimisation camber is taken as the starting point for the following optimisation position, and also utilising a transient approach to evaluate the aerofoil performance and the modelling of time-dependent aerodynamic events such as the leading-edge stall on the aerofoil. Implementing these complex deformation profiles into a CFD-in-the-loop based Genetic Algorithm is a new avenue of research for VAWT optimisation, which will provide valuable insights into the effect of the transient flow phenomena. Variations in the GA's parameters were investigated including crossover function variations, phenotype counts and sampling count tests to optimise the computational efficiency of the optimiser, without sacrificing performance and ability to increase VAWT power generation. Additionally, the position of maximum camber's influence on deforming VAWT blade performance was found to have significant impacts on the regions of largest gain in the VAWT's rotation, and the contribution of the leading edge and trailing edge deformations was studied; finding that the leading edge improves the upstream turbine power generation and trailing edge deformations improve downstream power generation.

Computational Fluid Dynamics is an appropriate method for evaluating the phenotype's viability and fitness function calculation. Whilst other methods are available to evaluate the aerofoil performance, such as the panel method solvers including XFOIL or JAVAFOIL, these methods limit the initial flow conditions because of the limitations discussed in Section 4.1.3.2, the primary issue being the inability to measure the lift and drag when the aerofoil boundary layer is completely separated. In XFOIL, the user can define the Reynolds number, Mach Number, angle of attack and freestream turbulence level (N_{crit}), but when attempting to optimise the camber at multiple azimuthal positions, it does not consider the wake or inductance effects on the flow at the subsequent optimisation positions. Furthermore, the dynamic flow characteristics are not carried over between the optimisation positions in XFOIL, as the flow uses instantaneous characteristics. Transient CFD simulations, however, allows for the wake effects to be carried between each optimisation position due to the time-marching progression of the simulation, allowing the optimisation to capture the previous flow conditions as the starting point (the new optimisation position's initial conditions), and optimise the following position (the targeted angle of attack for camber optimisation) accounting for the true local flow field at the desired optimisation position.

5.2 CFD-In-the-Loop Transient Optimisation Setup

The computational domain and setup used for the VAWT blade camber optimisation and the turbine torque performance is the same as that described in Section 4.3 of the thesis. As this investigation uses the same two-dimensional computational domain sizing and the same turbine characteristics as the previously validated turbine studies, in Section 3.2, and current literature [23], [68], [93], [94], it is considered satisfactory for the basis of the current study. This CFD domain allows for the validation and investigation of all the required aerofoil genotypes that the optimisation process undertakes, regardless of the combinations used.

5.2.1 Genetic Algorithm Process Flowchart

For a general overview of the workflow of the optimisation process, the following simplified overview outlines the steps:

- i. Initialise the GA
- ii. Create the initial array of phenotypes (camber values)
- iii. Evaluate each phenotype individually
 - a. Open the case file
 - b. Append the associated camber input parameter
 - c. Run the GA for a user specified number of time steps to reach the desired optimisation position
 - d. Export the Moment Force from the CFD to the desired dataset
- iv. Capture the flow image
- v. Sort all moment values from the highest to the smallest
- vi. Create the next generation of phenotypes based on the current generation's moment values
- vii. Repeat steps iii to vi until the convergence criteria is met
- viii. Output the final optimised camber
- ix. Run the case file with the optimised camber for the defined time steps
- x. Save the final case/data file for the next optimisation position

In order to complete the first step--initialising the genetic algorithm, it is necessary to understand how the algorithm works and which defining characteristics are required for the process. These include the aerofoil thickness, rotational centre, turbine size and speed, and the general flow conditions such as the freestream velocity.

The initial array of camber values is simply an array of the potential values evenly distributed over the user-specified range. For this investigation, $\pm 10\%c$ is used as the bounds for the camber magnitude. Whilst the NACA profiles could theoretically model substantially larger camber values, the convention is that the camber typically only has a range of $\pm 9.5\%c$ for most NACA aerofoils in use due to the naming

convention (single integer in name i.e. NACA5315). If one were to decide on using 11 genotypes per generation to evaluate every 36° , the initial generation can be seen in Table 5-1.

Genotype ID	0	1	2	3	4	5	6	7	8	9	10
Camber Value	-10	-8	-6	-4	-2	0	2	4	6	8	10

Table 5-1: Example of the Camber Values for the First Generation of Genetic Algorithm Optimisation

The details of the later stages in the optimisation are now described in detail. To evaluate each phenotype, the potential aerofoil must be solved in the CFD solver (ANSYS Fluent) to capture the flow field around the aerofoil and derive the force coefficients appropriately. This allows for consistent, methodical and precise comparison between each genotype. The sorting is applied in Python, extracting the data from ANSYS, and then organising and mutating them as necessary. For details on the full process, see Section 3.4 which breaks down the function of each step in a genetic algorithm optimisation.

The most important step in evolutionary algorithms is the child creation. This is referred to as crossover, where two parent genotypes combine to form an offspring, as is seen in nature. The typical crossover function takes the optimal values from the previous generation and treats them as Parents, the child is then taken as the midpoint value (camber in this investigation) between the two parents, and the predefined mutations on the child genotype is added to the generation to maintain the genetic diversity, ensuring that the global optima are found rather than the local optima in the searching space [109].

5.2.2 Fitness Function

Many different parameters can be used to define the fitness function of a genetic algorithm, and they are one of the factors of the problem that one is attempting to

optimise. A basic example that is commonly used for teaching the fundamentals of the genetic algorithm is the travelling sales person--where the individual has to visit a list of cities exactly once before returning home, but the optimisation function is to minimise the total distance travelled by the sales person.

In the case of VAWTs, the fitness function could be related to the structure, aerodynamics or even the physical mechanisms of transferring the generated power back to the grid. One could investigate the optimal winding pattern of the magnets in the generator, the minimum thickness of carbon fibre layup to provide sufficient structural rigidity, or in the case of this investigation, the aerodynamic properties that directly correlate to the power generation--namely Lift, Drag, Tangential Force and Moment of the turbine blades. It should be noted that the moment force is likely to be the most suitable parameter to optimise, as this is the most holistic characteristic used to define the turbine performance. The power generation and power coefficient of the VAWT are directly proportional to the moment on the turbine blades, which in turn is a result of the tangential force and the lift and drag forces on the blade.

5.2.3 Convergence Criteria

To terminate the genetic algorithm, it is necessary to define some criteria for when the optimisation has been completed. Much like the iterative algorithms used when solving the traditional Navier-Stokes-based finite volume or finite element methods, some sort of convergence criteria must be set. Otherwise, the process will continue indefinitely or until there is user intervention. There are multiple ways that this can be achieved due to the number of variables within the process. One option is to check the difference in the optimal camber between each generation and limit what is acceptable. This difference could be a percentage difference or an absolute difference, e.g. 1% or 0.01 in value. The issue with utilising absolute values is that the optimisation process could terminate early, potentially focusing on a local optimum and negating the impacts of the randomisation factor in the offspring crossover functions. Furthermore, utilising the camber as the value to consider for

convergence, can also contribute to this early termination concept, as after a few generations there can be large changes in the moment coefficient for very small changes in the aerofoil camber.

Considering these effects, the most appropriate convergence criteria is therefore selected as a percentage difference in the moment force between the strongest genotype of each generation. The convergence criteria are only evaluated after 10 generations of optimisation to ensure sufficient randomisation before the convergence checks are implemented. Starting the convergence testing prematurely or setting the convergence criteria too large can potentially cause local optima to be selected, rather than searching for the global optima. In this investigation, the criteria for convergence is a 0.01 absolute difference in optimum moment, as shown in Equation 5.1. 0.01 was found to be a good balance in the previous chapter's investigations and was therefore utilised in this chapter.

$$Moment_n - Moment_{n-1} < 0.01, \quad n > 10 \quad 5.1$$

There are many factors that influence what would be considered a sufficient convergence condition; too small a convergence condition and the optimisation process could end up running for a very long time without much increase in performance of the optimised design. It is especially crucial to minimise this when using CFD simulations as the source of chromosome evaluation, as it results in unnecessary consumption of energy and restricts access to the high-performance clusters typically used for intensive CFD. Too large a convergence condition and it leads to uncertainty in the validity of the optimised result, and whether it is truly the optimal design in the search space. It could result in premature optimisation, which defeats the purpose of utilizing the evolutionary algorithm in the first place! Case file / UDF setup

The mesh method defined previously in Chapter 4.3 is utilised in the cases for the genetic algorithm evaluation stage. It has input parameters defined within ANSYS Fluent which can be easily altered as appropriate, using ANSYS's pythonic fluent interface library. These are set as the camber magnitude at each optimisation

position, which is used within the User-Defined Function (UDF), thus allowing for the dynamic changing of the variables within the UDF. This pythonic interface method does not need to recompile the UDF between each case, thus improving the evaluation speed for each phenotype. Furthermore, using the Pythonic interface means that it is not necessary to programmatically change the variable values in the UDF file, which would require the use of complex regular expressions to parse the UDF file and change a value without altering the formatting of the UDF file. As the UDF is written in the C programming language, it is susceptible to indenting, formatting and inclusion of punctuation for proper line-by-line evaluation. Editing any of these factors in the UDF would result in a failed compilation of the UDF and, therefore, the failure to properly implement the flapping motion in ANSYS Fluent. The UDF, therefore, uses the same method of pulling the Input parameter values from within ANSYS Fluent to define the values of each variable in the camber profile of the turbine.

The general data flow for the transient simulation-based genetic algorithm process is as follows:

- i. Compile the UDF, ensuring the appropriate camber variables are defined to match what is used in the case file.
- ii. Create matching input parameters in ANSYS Fluent.
- iii. Save the Baseline Simulation Case File.
- iv. Change the Input Parameter in ANSYS Fluent through Pythonic Interface, which corresponds to the relevant optimisation position.
- v. Run the ANSYS simulation for the necessary amount timesteps to achieve rotation and deformation to the required optimisation position.
- vi. Export Moment values from ANSYS Fluent to the python session
- vii. Sort genotypes (camber) by corresponding values of their phenotypes (moment).
- viii. Repeat steps iv to vii for each genotype in a generation.
- ix. Create the next generation of offspring, including mutations and crossover between parents.

- x. Repeat steps viii and ix for each genotype per generation until convergence is achieved.

This is summarised visually in Figure 5-1, which shows how the variables are carried between the different coding environments.

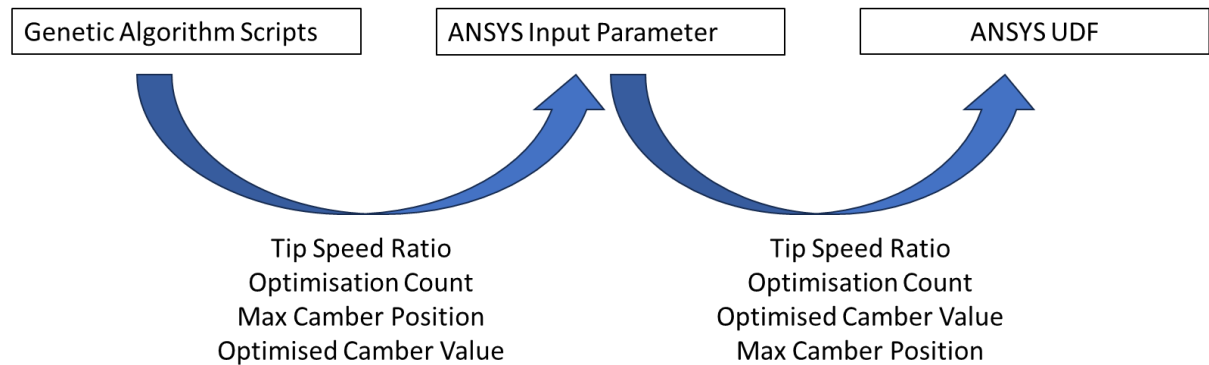


Figure 5-1: Transferred variables between Genetic Algorithm Simulation Control steps

5.3 PyFluent Functions

In order to implement the genetic algorithm within ANSYS Fluent dynamically, it is necessary to utilise some sort of interface between the dataset and the ANSYS Fluent simulation settings. This could be implemented by using regular expressions to modify text files such as the UDF script or journal files to be read by ANSYS Fluent, as was implemented in Chapter 4 for the automation of XFOIL. This could work, but leads to very complex text based scripts and would require separate scripts each time a setup variation is changed.

In the past few years, ANSYS has started developing a pythonic connection library that allows for users to control the ANSYS software through specially made python libraries. One of their first Pythonic libraries was PyFluent, a Pythonic interface to ANSYS Fluent which allows the control of all the settings in ANSYS Fluent, either through commands that follow the same format as journal files or with a specialised Settings API. This allows for all setup variables to be dynamically changed and fed into the Fluent client directly without a need for recompiling UDFs for each genotype, rapidly increasing simulation turnaround for each optimisation candidate. The settings API is a newer addition, so it does not have full coverage of all the settings in

ANSYS Fluent. For this investigation, the TUI-based method is used. The details of these scripts are found in the following section.

5.3.1.1 Solver

The workflow for implementing the genetic algorithm in ANSYS Fluent has a similar principle to that used in the previous chapter based on XFOIL. The XFOIL workflow utilised a script to generate a text script containing the coordinates of each phenotype's profile, which is imported into the XFOIL solver instead of utilising the built-in NACA aerofoil generator. Even though all the genotypes use the NACA shape profiles, the typical cambered NACA aerofoil has a maximum possible camber of 9.5%c.

This investigation changes the camber of the aerofoil by using a UDF, which actively deforms the aerofoil as it rotates to the optimisation position. The camber is linearly interpolated between the optimisation positions, which allows for gradual changes in the aerofoil shape as the turbine rotates. By transforming the camber as the turbine rotates, one can capture the transient effects of the camber deformation process. More importantly, the data utilised for optimisation considers the extensive differences in the flow characteristics.

Turbine flow characteristics are largely dominated by their tip speed ratio, radius, and free stream velocity [110]. By optimising the camber profile in the context of the turbine's rotation, this algorithm allows for these flow characteristics and their impact on the torque and power generation to be incorporated into the optimisation procedure. Theoretically, the angle of attack and flow speed are identical between the upwind ($0^\circ < \phi \leq 180^\circ$) and downwind ($180^\circ < \phi \leq 360^\circ$) portions of rotation, as is the windspeed and tip speed ratio. Actually, due to the turbine extracting energy from the flow in the upwind region, the windspeed that the turbine blade experiences is actually lower than the windspeed. The result is that the TSR is therefore, higher in the downwind region and subsequently, the angle of attack experienced by the blade and the energy in the flow is much lower [108]. The full script that was implemented

to control the solver functions and define the aerofoil characteristics is found in the Appendix and online on my GitHub repository.

5.3.1.2 Image Capture

A script was also developed that allows for a snapshot of the flow to be taken after each optimised aerofoil. This is useful when analysing the optimised camber profile, as it allows for easier comparison of the aerofoil near flow field whilst reducing the post-processing time. The script takes the current azimuthal position, rotates the camera by that same angle in the opposite direction and zooms in on the flow field around the blade. This can be easily implemented within the optimisation loop to take an image of the flow field local to the blade for every single genotype if so desired, but to reduce the computational overhead and reduce the total disk usage of the optimisation workflow.

5.3.2 Implementation in cluster-based computer resources

There are several ways of running the optimisation methods discussed so far. The options available range from being as simple as running the Python script on a personal desktop computer to creating complex scripts for asynchronous job submissions on high-performance clusters. Using asynchronous submission allows for each genotype to be evaluated simultaneously every generation, with the system waiting for all genotypes to be evaluated before creating the next series of genotypes and submitting the offspring evaluation simulations.

The alternative method is to have a serial case, where a standard single job runs, and sequentially evaluates the genotypes using ANSYS' new PyFluent pythonic interface library. This allows the user to create and control ANSYS Fluent sessions through a Python API, passing through any variables within the Python session directly into ANSYS. This avoids the complex and tedious method of recreating new submission scripts for every genotype to be tested, using complex regular expressions, and potentially introducing issues into the solver due to syntactic errors in the scripts. For

example, between versions of ANSYS, the text user interface (TUI) occasionally changes, which can result in a script used in ANSYS 22.2 not being compatible with ANSYS 23.1. To correct this, it is necessary to rewrite the regular expressions to be compatible with each specific version of ANSYS Fluent. Using the PyFluent API however does not have this same issue, simply change the version of ANSYS that one intends to load in the script's initialisations and then run the script as normal, and no alterations are needed!

5.4 Results & Discussion

It is crucial to understand the impacts of the various features of the genetic algorithm, and as such, it is necessary to evaluate and experiment with variations of the different aspects of the algorithm setup. Studying the effect of these factors allows for the opportunity to minimise the genetic algorithm's computational cost whilst simultaneously exploring its ability to optimise different aerofoil characteristics and operating conditions.

Firstly, the effects of variations to the genetic algorithm are studied to ensure the computational efficiency and the potential impacts of altering parameters on the GA's ability to fully optimise the blade camber. This is followed by a study of the aerofoil shape constraints to better understand the contribution of each section of the blade's influence on the performance. Finally the most optimal genetic algorithm combination found is implemented on the same turbine at a lower tip speed ratio of 1.5 and a higher tip speed ratio of 3 to provide insight into the effects of blade angle of attack, Reynolds number and flow conditions on the genetic algorithm in order to form more well-rounded conclusions of its applications.

The following characteristics of the genetic algorithm were investigated in this investigation:

- i. Crossover Variation
- ii. Phenotype Count

iii. Sampling Count

The aerofoil constraints include;

iv. Position of maximum camber

v. Leading edge only optimisation and trailing edge only optimisation.

5.4.1 Genetic Algorithm Parameters Investigation

The characteristics mentioned above are studied and compared in this sub section. It focuses on the parameters of the GA from a computational perspective, but also ensures that the actual turbine performance is not impacted by altering the different components.

5.4.1.1 Crossover Variation

The crossover function has an array of methods of creating the offspring genotypes, with variations on randomisation, overlapping between parents and mutation magnitudes. The standard crossover follows the following trend:

- i. Parent A
- ii. Parent B
- iii. $(\text{Parent A} + \text{Parent B}) / 2 = \text{Offspring}$
- iv. Offspring $\pm$ random(0-10%)
- v. Offspring $\pm$ random(0-5%)
- vi. 3 x Random(lower bound, upper bound)

Incorporating the randomisation is a two-sided method. The benefit includes the potential to converge to an appropriate genotype more quickly, ensuring the breadth of search to prevent local minima convergence to tend towards the global maximum. On the other hand, it forces the process to potentially run unnecessary simulations that don't improve the overall optimisation process, thus increasing the overall computational time with no meaningful reward.

An alternative method is to replace genotypes 4 and 5 by using a variable offspring mutation instead of using a predefined variation (± 0.1 or ± 0.05). This method is

likely to be much faster as the minimum change in the camber would be 0.025 (from the offspring in the next generation), whereas with the randomisation method the change in camber can be as low as 0.005 between generations, resulting in another generation of the simulation being required, but for potentially a negligible increase in the actual blade moment.

These two crossover methodologies were compared to gauge which technique captures a more optimised camber profile, and to ensure the most reliable method is used for the later aerofoil characteristic investigations. Figure 5-2 shows the resultant moment plot of the optimised profile for the two techniques and Figure 5-3 shows the camber magnitudes of the profile based on their azimuthal position.

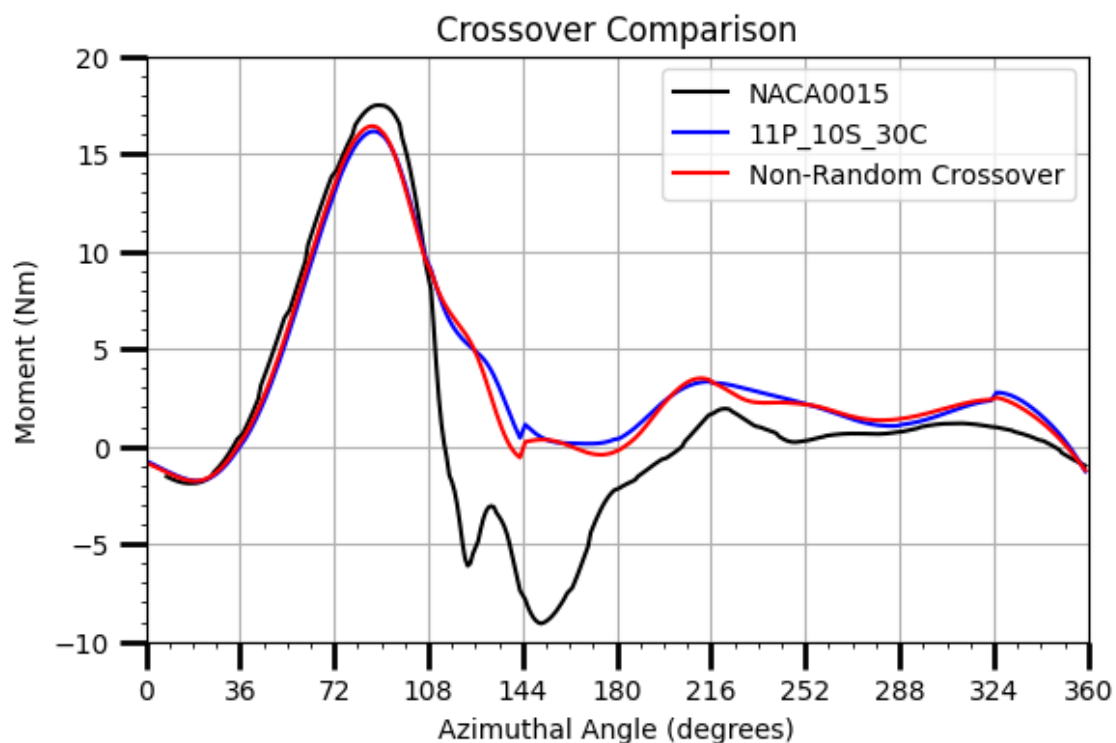


Figure 5-2: Impact of a randomised crossover magnitude compared to a constant crossover magnitude

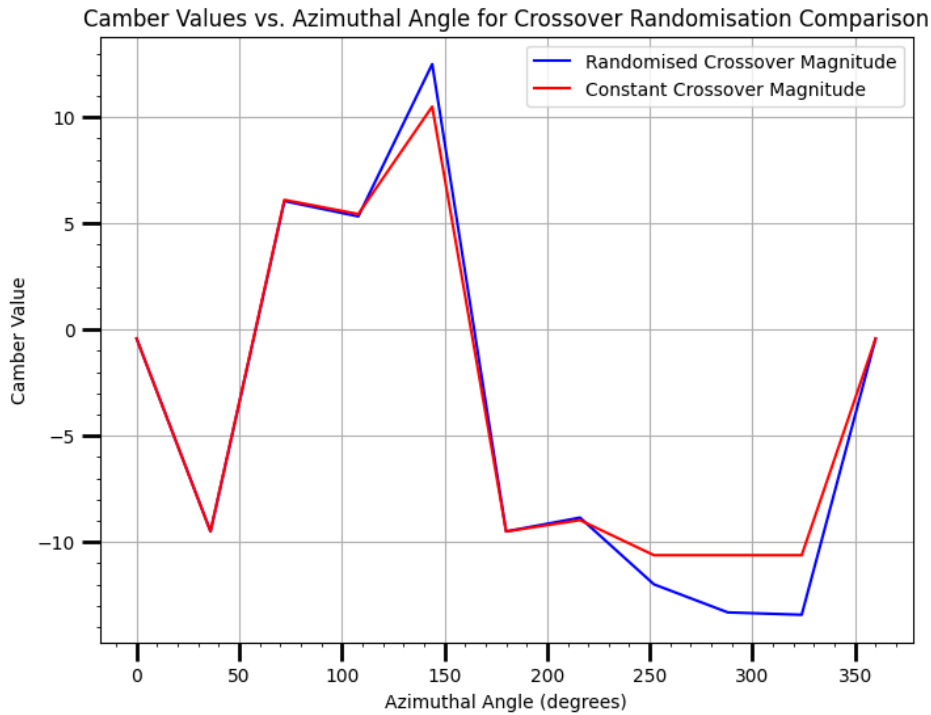


Figure 5-3: Camber profile comparison for randomised and constant crossover techniques.

The moment comparisons are almost identical for both profiles, with the only noticeable differences at 144°, ~175° and 324°. These regions of difference directly correspond to the positions where the optimiser has converged on a different camber value between the two methods. The 144° position only has a moment difference of around 1Nm, but the randomised crossover had a 3% $\bar{c}$ larger camber magnitude. This shows how important it is to have a robust and consistent optimiser, as a significantly different camber profile might not have a significant impact on the performance. This is emphasised at the other two azimuthal positions mentioned, 175° and 324°, where the camber difference is ~3.5% $\bar{c}$ bar larger in magnitude for the randomised crossover, but with only ~0.5Nm and 0.1Nm difference, respectively.

The computational time was almost identical for the randomised and constant crossover in the offspring generation. This means that the only deciding factor for which method to implement in further testing is that of the overall performance and reliability.

As a result, it is decided to utilise the randomised crossover for future studies. The primary reason is the robustness of the technique—by introducing randomisation,

the chance of missing a global optimum profile at the desired azimuthal angle is reduced. An example of this scenario would be if, during the optimisation process, an aerofoil camber genotype of 10%*c* is found to produce the greatest moment. If we have a set crossover value of, say, 0.2%*c*, then the child genotypes would be 9.8% and 10.2%. If, however, there was a randomised value up to 0.2%*c*, then there is an infinite number of potentially better candidates to be found in the range of 9.8-10%*c* or 10-10.2%*c*. If the true global optimum is 10.1%*c*, then a specific and constant crossover would never find the global optimum unless a very small variation value (i.e. $\pm 0.01\%c$) is used, which would make the optimisation process a very slow and steady climb and is still limited to the precision of the selected crossover magnitude.

5.4.1.2 Phenotype Count Comparison

Another method for improving the turnaround times of the genetic algorithms is by optimising the phenotype array between each generation. The main factor impact the computational cost of the genetic algorithm is how long it takes for each genotype within each generation to be simulated in the CFD solver, and it is reasonably consistent that each simulation takes around 5 minutes to solve on a 32 CPU Core machine on the University of Sheffield High Performance Cluster (HPC). Therefore in order to reduce the total CPU time and in turn the amount of wall clock time for the simulation, it is necessary to reduce the number genotypes per generation without sacrificing the overall effectiveness of the genetic algorithm.

As a refresher, the array described in Section 5.4.1.1 forms eleven genotypes each generation as follows;

- i. Parent A
- ii. Parent B
- iii. $(\text{Parent A} + \text{Parent B}) / 2 = \text{Offspring}$
- iv. $\text{Offspring} + \text{random}(0-10\%)$
- v. $\text{Offspring} - \text{random}(0-10\%)$
- vi. $\text{Offspring} + \text{random}(0-5\%)$
- vii. $\text{Offspring} - \text{random}(0-5\%)$

- viii. Random(lower bound, upper bound)
- ix. Random(lower bound, upper bound)
- x. Random(lower bound, upper bound)
- xi. Random(lower bound, upper bound)

This results in 11 simulations being required for each generation. If the number of genotypes is reduced to 5, the computational time will also significantly reduce. One of the primary issues with this method is that it may take more generations to reach convergence. However, the number of generations to convergence would need to more than double to take the same computational time as the original eleven phenotypes method.

An alternative phenotype formulation is the following:

- i. Parent A
- i. $(\text{Parent A} + \text{Parent B}) / 2 = \text{Offspring}$
- ii. $\text{Offspring} + \text{random}(0-5\%)$
- iii. $\text{Offspring} - \text{random}(0-5\%)$
- iv. Random(lower bound, upper bound)

The reduced phenotype formulation would still carry the same optimisation philosophy as the previous generation--the optimal phenotype from the previous generation would be carried forward, and the offspring and offspring mutations would still be implemented similarly, but with only one randomised entry each generation, instead of four.

To check that the five-phenotype method doesn't have an impact on the overall turbine performance, it is compared with the moment of the eleven-phenotype method that was previously implemented in Section 4.2, as that is shown to be a reliable setup. Figure 5-4 shows the moment distribution for both the eleven-genotype model and the 5-genotype model, displaying that the difference in performance is negligible and almost identical between the two techniques overall. The moment curve is identical from 0° to around 120° degrees azimuthal angle, with a minor increase in the eleven-phenotype setup at 144°. It is then almost identical

between the two techniques again until the first noticeable difference at around 252°, where the eleven-phenotype method is lower than the five-phenotype method. The average moment for the eleven and five phenotype methods is 6.90426 and 6.918025, respectively, resulting in a difference of only 0.2%. This demonstrates that utilising a smaller generation size has a negligible impact on the overall performance of the camber deflection profile but radically improves the optimisation turnaround time. Comparing the camber profile of the two genotype implementations in Figure 5-5 shows that there is a small difference in camber profile at the optimisation positions of 288° and 324° but the performance difference is so negligible that this would not warrant a cause for concern in the method's reliability to find an optimal camber profile.

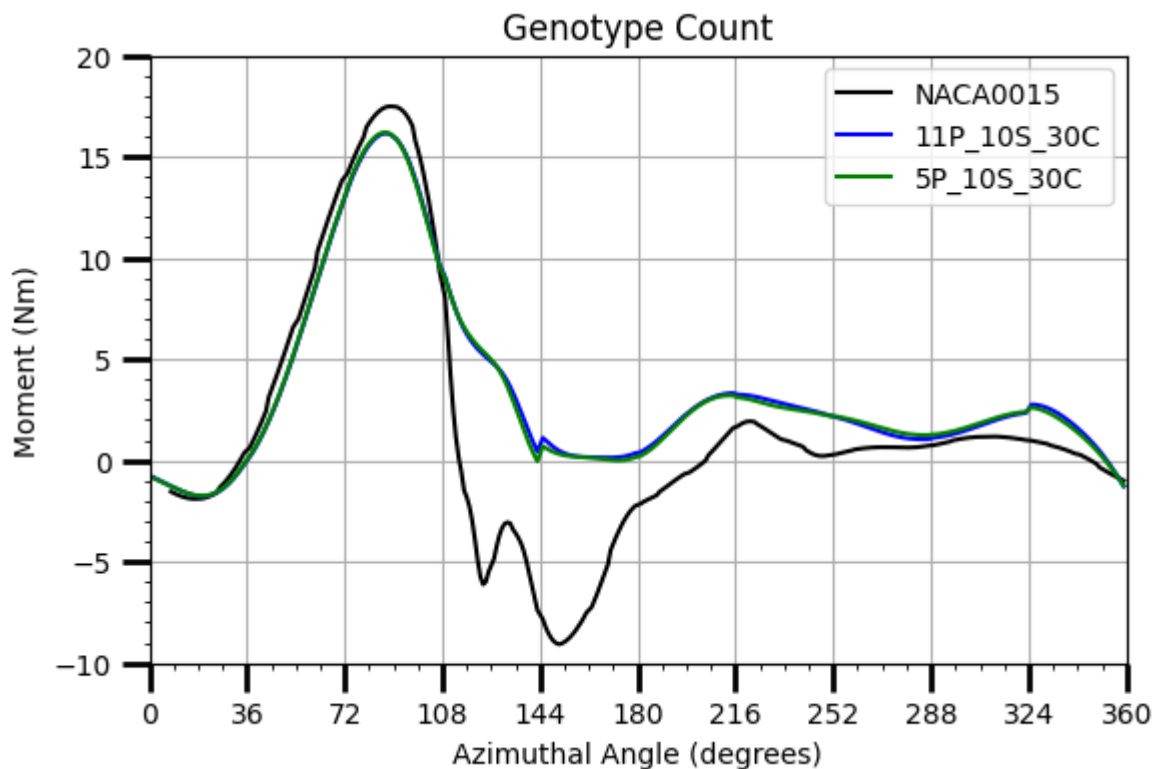


Figure 5-4: Moment contribution comparison for eleven phenotype method and five phenotype method.

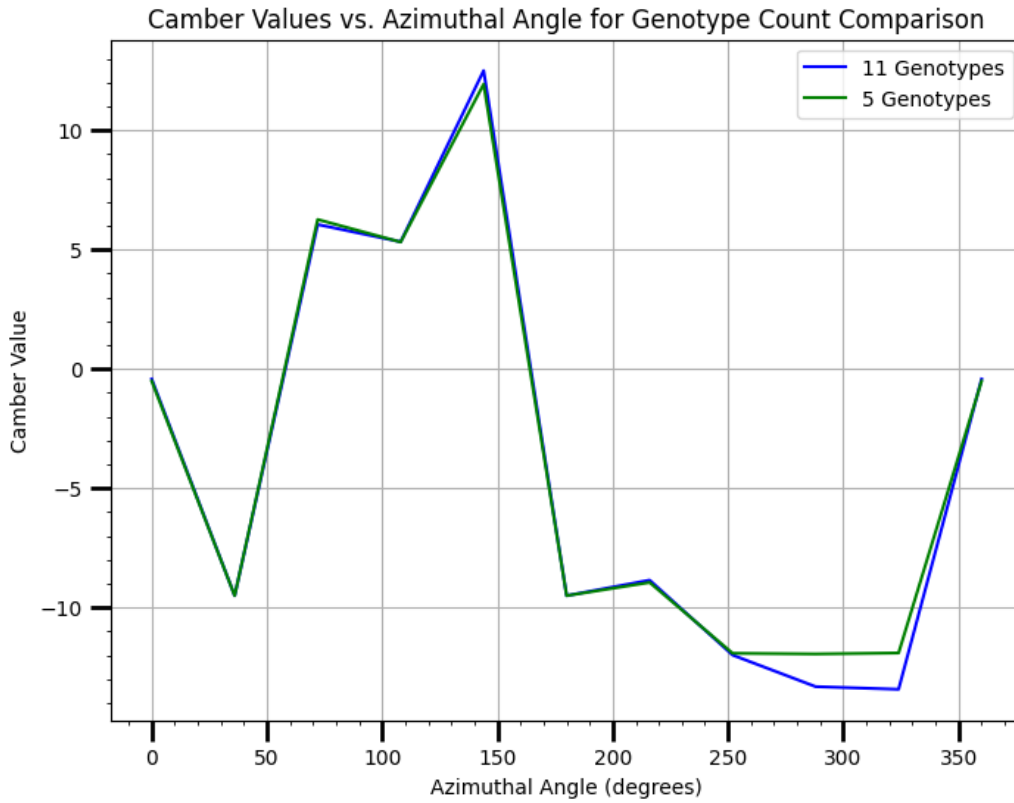


Figure 5-5: Genotype count comparison impact on camber profile

As was explained in Section 5.4.1.2, the use of a reduction in the phenotype count has the potential to reduce the overall computational requirements of the genetic algorithm optimisation procedure. Both optimisation procedures are carried out on identical hardware; a single node on the Stanage HPC Cluster at the University of Sheffield. These nodes consist of 4GB of memory per CPU core and two 32 core processing units per node, resulting in a total capacity of 64 CPU cores and 256GB of physical memory. Both cases only required 10 generations for sufficient optimisation to be achieved before the process was terminated. As there were less than half the number of phenotypes evaluated each generation, this directly equates to an almost double increase in efficiency of the system, taking half the time to complete!

Based on these conclusions, it is decided that for any subsequent tests and variations in the genetic algorithm setup, the five-phenotype model will be used. The effect of this decision increases the computational efficiency, allowing for better turnaround of the method and in the future, allows the method to be more accessible by other users of the method, due to the reduced computational requirements.

5.4.1.3 Sampling Count Comparison – TSR 2.29, $p = 30\%c$, $t=15\%c$

Another method of potentially optimising the genetic algorithm's computational efficiency is to test if changing the sampling points will be beneficial to the overall performance of the algorithm. Two scenarios are compared, labelled as 10S and 24S. The 10S case implies ten sampling positions, or every 36° of azimuthal rotation the camber value is defined, whereas the 24S case samples/optimises the camber every 15° of rotation.

This computational efficiency gain is based on the same premise as earlier subsections of this chapter: reducing the number of simulations will reduce the computational time as each simulation is approximately the same duration.

Figure 5-6 and Figure 5-8 show the difference in the single blade moment and optimised camber value respectively. Note that there are some gaps in the curve for for optimised blade moment, this is due to spikes in the output file generated in ANSYS Fluent, which would cause the resultant curves to be misleading. These spikes coincide perfectly with the location of the optimisation points in the UDF, but there is no significant change in the camber across these points, suggesting some sort of unrealistic mathematical error at the exact instance the rate of camber deformation changes. An additional figure is seen for the sake of transparency in Figure 5-7, which contains the original raw data from ANSYS Fluent before it was pre-processed. This is emphasised by the lack of spikes seen in the moment of the 24 sample count method's downstream period of rotation, wherein the blade deformation is more gradual and not as sudden.

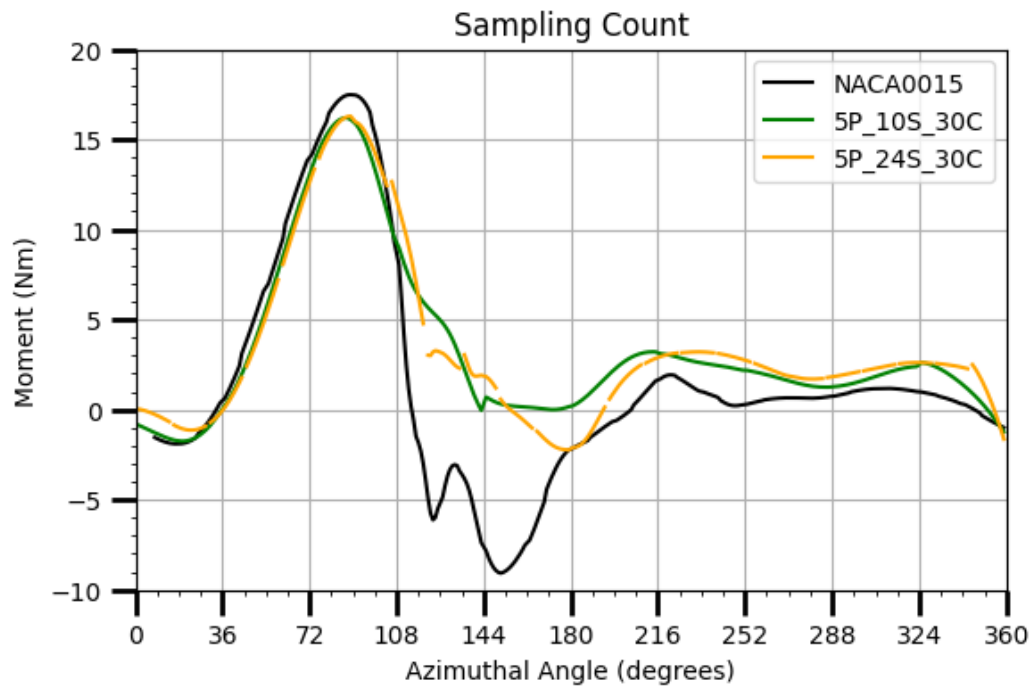


Figure 5-6: Effect of Sampling Count on Optimised Turbine Profile.

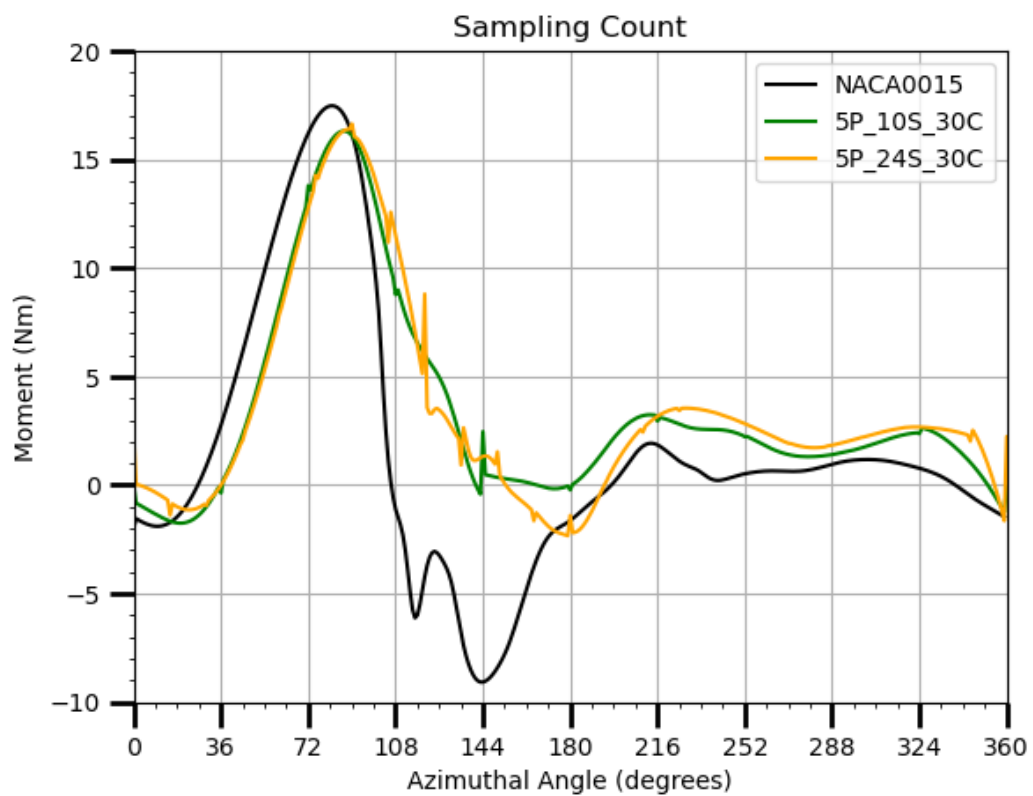


Figure 5-7: Raw CFD data for Sampling Count effect.

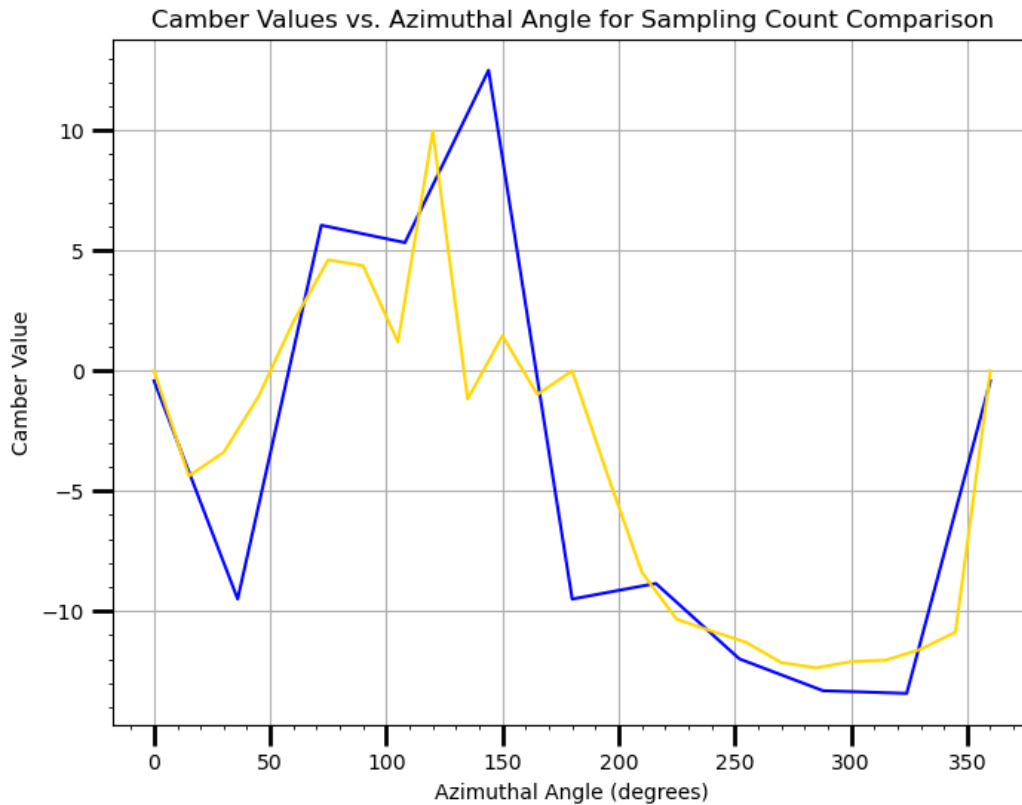


Figure 5-8: Impact of Sampling Count on Optimised Turbine Camber Profile

Alternatively, one could use predetermined optimisation positions. Rather than optimising the camber every n degrees of azimuthal rotation, the user could consider the impact of simply optimising the camber at the desired optimisation position. For example, the potential single position optimisation candidates could be the middle of the downstream stroke (260°), or at the peak moment position (100°).

As a result of the GA configuration investigations, the proposed GA techniques that will be carried forward for the remaining studies will be to use 10 sampling points (every 36°), with 5 phenotypes within each optimisation generation and randomised crossover.

5.4.2 Aerofoil Variations

Moving from optimising the efficiency and testing the limitations of the genetic algorithm, the next logical step is to study the effects of altering the aerofoil constraints of the genetic algorithm. As was described in the introduction of the thesis, it is mechanically complex to design a blade mechanism that can alter the

blade thickness, and extending the chord length could have significant structural implications for the turbine. Therefore the aerofoil constraints investigated are all components of the camber—namely the position of max camber and whether it is better suited to only have the leading edge deforming, the full blade length deforming, or only the aft edge deforming (similar to what was described in the thesis introduction).

5.4.2.1 Position of Maximum Camber

The first consideration for aerofoil variations is the position of the maximum camber, defined as a distance along the chord line, such as 20% from the leading edge to the trailing edge, 30%, etc. Altering the aerofoil's position of maximum camber has significant effects on the aerofoil's aerodynamic performance.

It is theorised that by shifting the position of the maximum camber towards the leading edge, the aerofoil could be pointing into the direction of the incoming flow better, reducing the incident angle of the flow on the blade, especially at high angles of attack. This is a result of the camber line (which is centered in the aerofoil along the blade profile from leading to the trailing edge) potentially being parallel to the incoming flow, allowing for smoother airflow over the blades [111].

A potential downside, however, from the dropped leading edge of the turbine blade is that the overall curvature of the aerofoil is very high. This will create a sizeable curvature of the flow, which has the potential to increase the aerofoil's lift generation, but also comes at the cost of an increased pressure gradient on the aerofoil's boundary layer. The position of maximum camber will also be the location of maximum aerofoil curvature, and can therefore be very disruptive to the turbine performance—if this position is too close to the leading edge, it could result in larger regions of the blade being separated and lead to stall at lower angles of attack than if the position of maximum camber was shifted towards the trailing edge.

To investigate these hypotheses, the genetic algorithm aerofoil constraints for the position of maximum camber was altered and tested at a series of different positions of maximum camber (denoted by the letter p in common notation when discussing

NACA aerofoils). Figure 5-9 shows the results of the moment across the turbine's rotational period for the tested positions and provides interesting insights into the effects of altering the position of maximum camber. The first clear conclusion is that the 20%*c* case performed the worst among all of the tested maximum camber positions. This ties in with the theory that if the position of maximum camber is shifted too far forward, a more significant amount of the aerofoil will stall—resulting in a drop in the lift and consequently a reduction in torque. The significant stall can be seen from the peak @ 90° azimuthal angle, through to around 126° where the aerofoil recovered slightly. The stalled region is still better performing than the NACA0015, but the overall trend in the moment generation is very similar, suggesting that the position of maximum camber is too close to the leading edge. The 50%*c* maximum camber position seems to support this conclusion, as shifting the position further back might allow for a gentler curvature of the blade, resulting in less stall of the aerofoil. In fact, from 0 to 180 degrees, the trend of the moment generation plot almost mirrors the theoretical angle of attack plot for a VAWT blade (described in Figure 1-6), reinforcing the conclusion that the blade has not stalled.

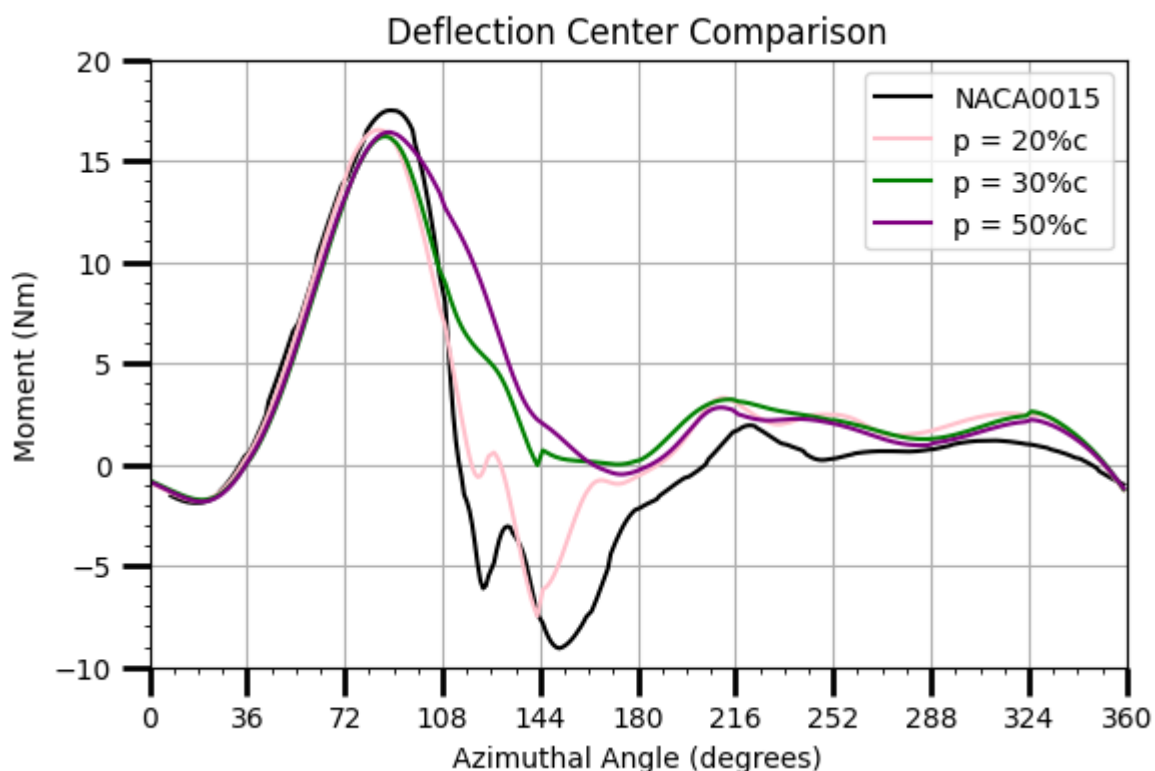


Figure 5-9: Deflection Centre Influence on Torque Profile

Figure 5-10 shows the camber value of the optimised aerofoil throughout the turbine rotation. Each vertex in the figure represents the actual camber value that was optimised by the genetic algorithm, with linear interpolation between the points. Interestingly, the optimised azimuthal position is identical for the 30%c and 50%c aerofoil variations before the 50%c position continues to increase in camber at 108° whilst the 30%c position camber decreases slightly, as a result of the improved separation in the 50%c case. 144° has the opposite result, but interestingly this is not reflected in the blade moment, with the 30%c blade having a larger deflection, but a moment of almost half that of the 50%c blade! This is a result of the optimisation case not considering the potential consequences of it's optimisation i.e. the 108° optimisation of the 30%c blade is the best option, but this could result in a less energetic flow impacting the blade at the next optimisation position (144°) resulting in a poorer blade torque, but potential for a larger blade camber.

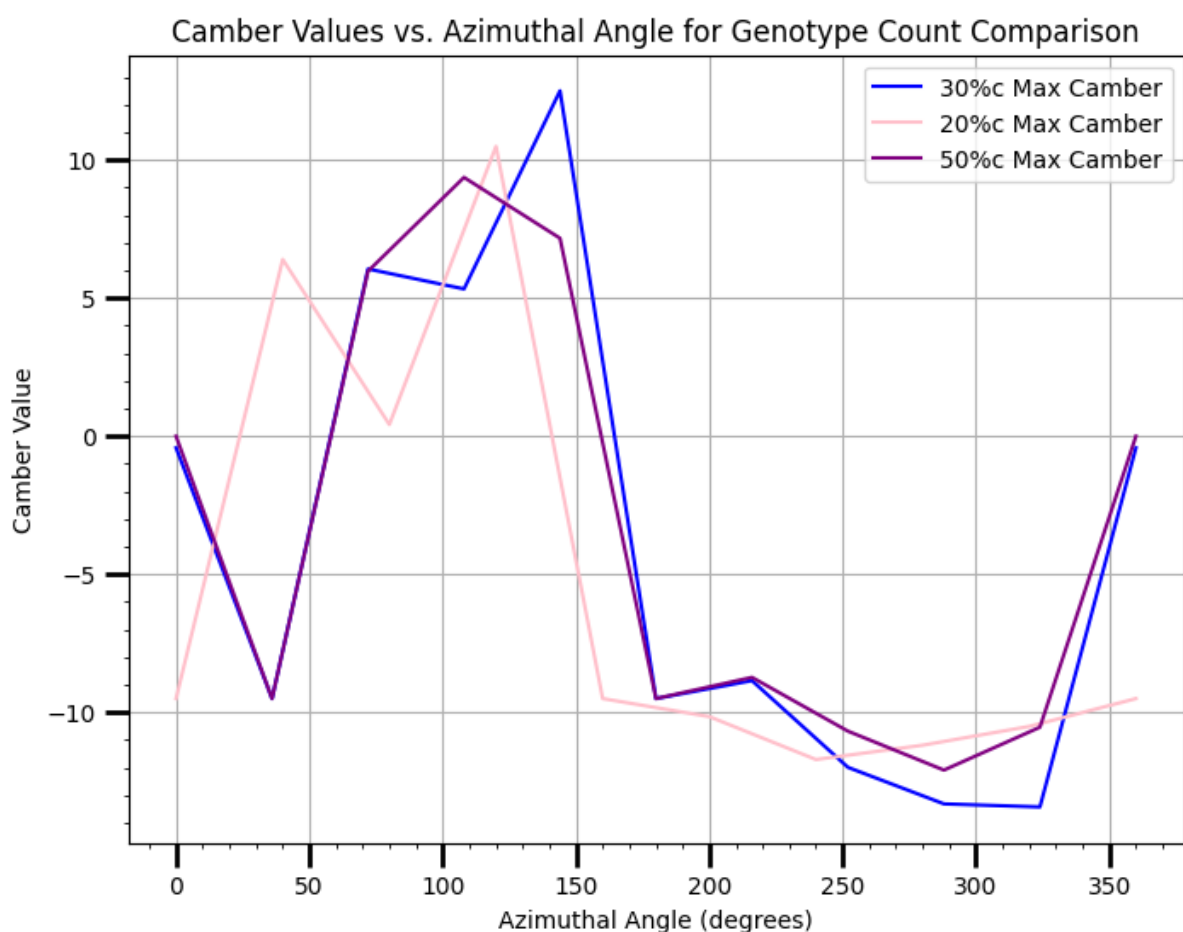


Figure 5-10: Position of maximum camber effect on the turbine camber profile

Looking at the flow snapshot at these two optimisation points (72° and 108°), there is clear differences in the velocity of the airflow, particularly in the wake and the magnitude of separation that the blade experiences. Comparing the velocity contours at 72° first, seen in Figure 5-11 and Figure 5-13, demonstrates that the optimised camber profile is very similar in performance at this point, regardless of the aerofoil's position of maximum camber, seen by the almost indistinguishable differences in the aerofoil's wake. The separation point appears to be approximately around the position of the maximum camber, which is understandable, as that is the position where the rate of the local blade normal angle is changing the most, implying the potential for an adverse pressure gradient is increased at that position.

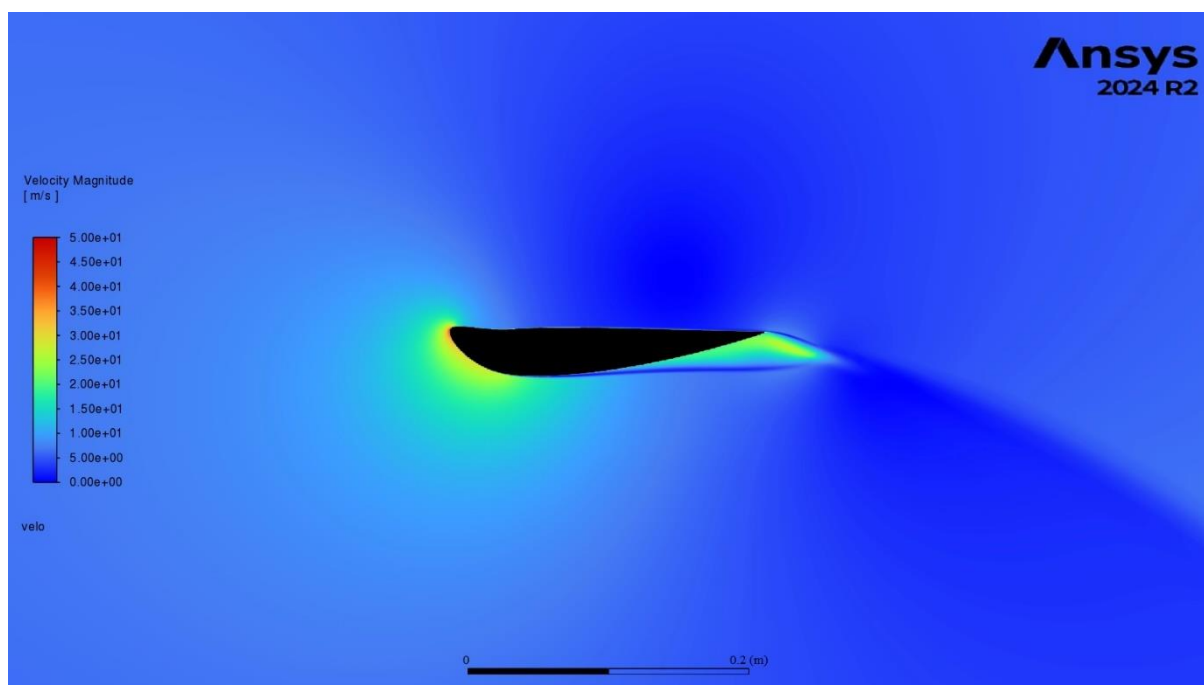


Figure 5-11: Velocity contour for aerofoil optimised at 72 degrees with a 20% maximum camber

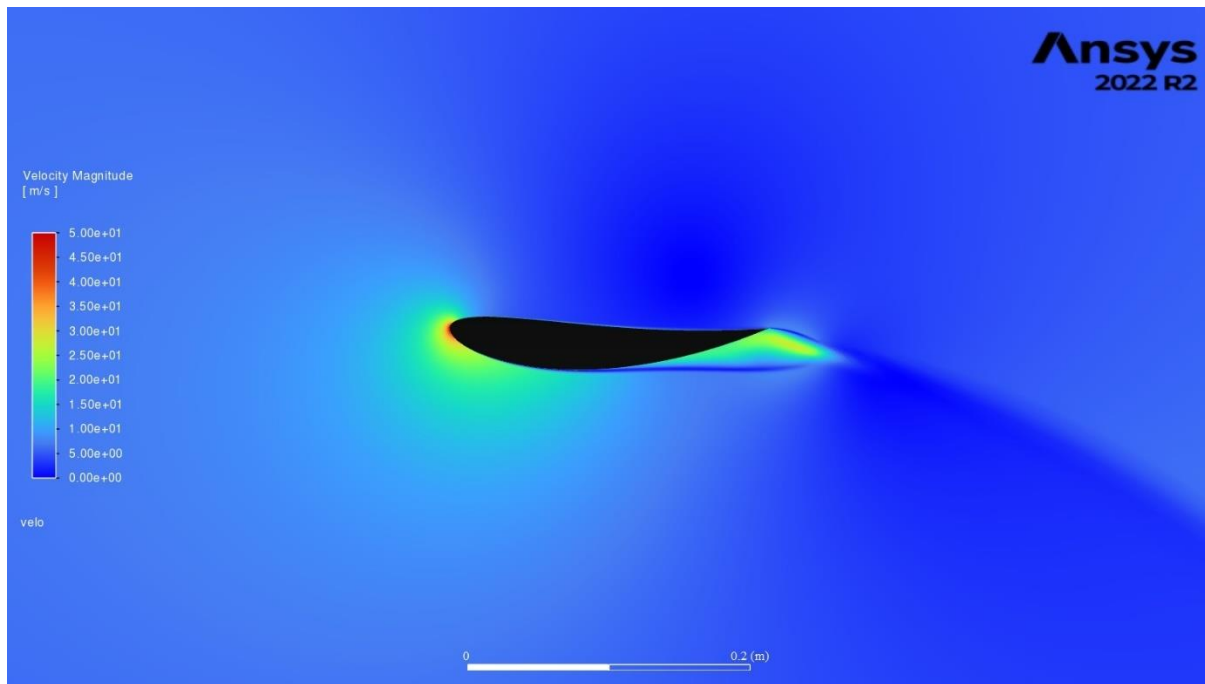


Figure 5-12: Velocity contour for aerofoil optimised at 72 degrees with a 50% max camber

Figure 5-13 and Figure 5-14 show the velocity contours for the 20% and 50% camber blades, respectively, at 108° azimuth. They tell a significantly different story than the 72° optimisation position, as the wakes and separation points are different! For starters, the 20% camber case had an optimised camber of only slightly above 0, whereas the 50% camber case had a camber magnitude of 6%! The 20% case is lower as all other camber values resulted in worse tangential force generation, due to significantly increased flow separation and increased drag due to the larger region of low-pressure wake. Conversely, the 50% camber position had more of the blade attached (albeit not much more), resulting in a later separation and reduced separation bubble size and magnitude. The magnitude is lower compared to the 20% camber case, displayed by the lack of red velocity region, whereas the high-velocity region in the 20% case signifies a much stronger vortex in the wake of the adverse pressure.

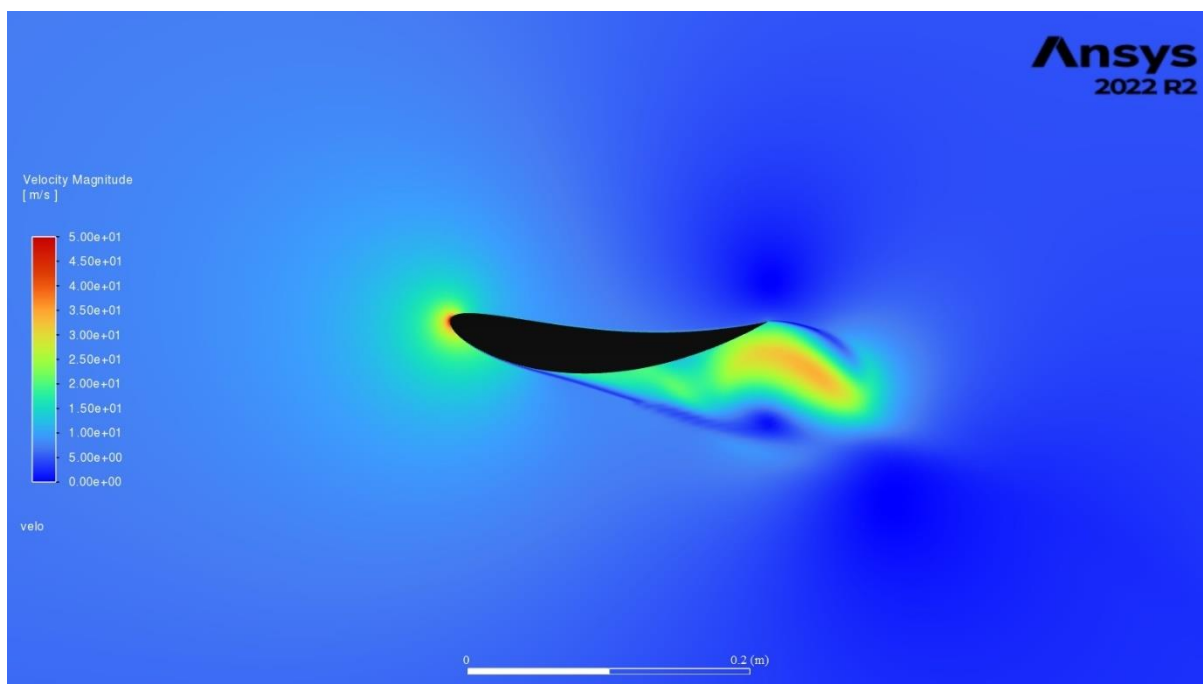
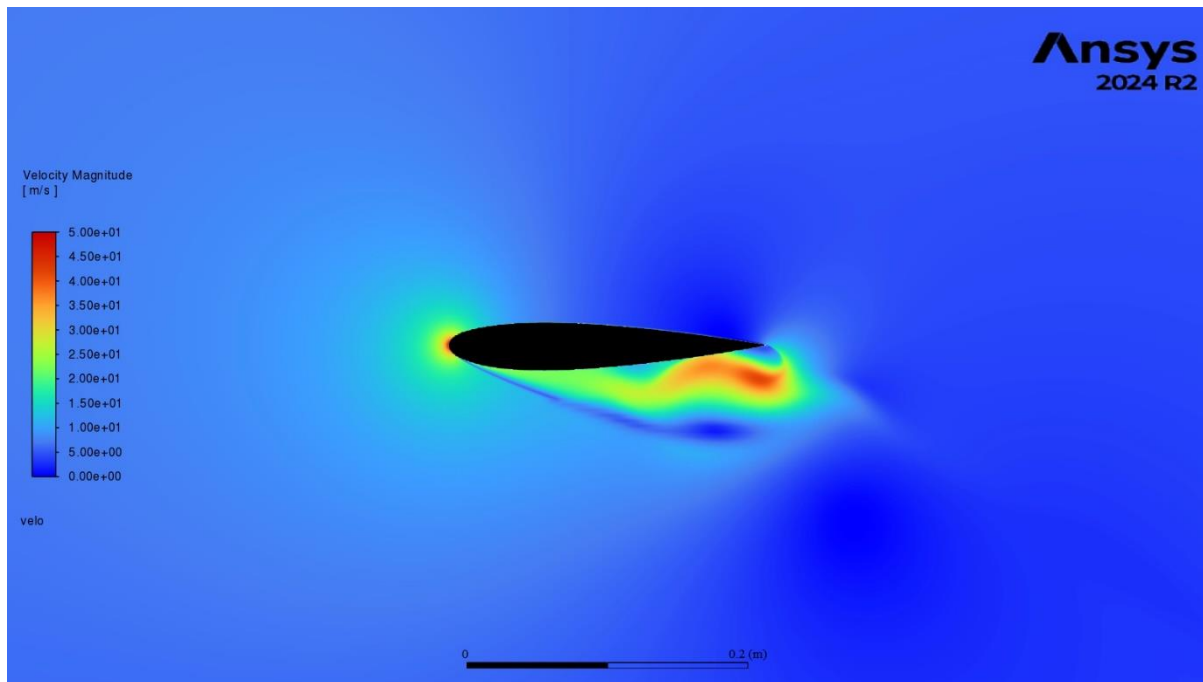


Figure 5-14: Velocity contour for aerofoil optimised at 108 degrees with a 50%c max camber

5.4.3 Half Blade Deflection Model

The two regions of camber deflection have two different reasons for improving the turbine blade performance. When the leading edge is pointing into the flow direction, it reduces the adverse pressure and assists to reduce the separation on the

suction side of the turbine blade [112]. By increasing the blade curvature in the trailing edge region, it effectively increases the camber of the aerofoil and, as such, increases the circulation that the blade will impart on the fluid flow.

The impact of each respective region of deformation is investigated to observe the impact of each region separately, and to understand how they impact the flow structures. This is particularly useful when considering the validity of this method in the context of vertical axis wind turbines, as they are very sensitive to the blade's incoming flow conditions such as swirl, angle of attack and energy [113]. This was achieved by creating a new version of the UDF, which simply specified that $dy_c = 0$ in the section of the aerofoil that was intended to remain stationary. The adapted formulae for the deflection magnitude are seen in Equations 5.2 and 5.3, where the symbols are the same as described in Sections 0 and 03.3

$$\text{For Leading Edge Only Deformation: } \left. \begin{aligned} dy_c &= -\frac{m}{p^2}(2px - x^2) \\ dy_c &= 0 \end{aligned} \right\} \begin{aligned} x &\leq p * c \\ x &\geq p * c \end{aligned} \quad 5.2$$

$$\text{For Trailing Edge Only Deformation: } \left. \begin{aligned} dy_c &= 0 \\ dy_c &= -m\bar{c} \left(\frac{x - p\bar{c}}{c - p\bar{c}} \right)^2 \end{aligned} \right\} \begin{aligned} x &\leq p * c \\ x &\geq p * c \end{aligned} \quad 5.3$$

This UDF version was utilised to run the genetic algorithm optimisation process much the same as the other methods implemented so far.

Figure 5-15 shows the instantaneous single-blade moment through the turbine rotation. There are three lines presented: a solid line for the standard NACA515 aerofoil with continuous deflection from the leading to trailing edge through the turbine rotation, a dashed line for the result of only deforming the forward half of the turbine blade up to the position of maximum camber, and a dotted line showing the results of deflecting only the rear half of the turbine blade profile starting from the position of maximum camber. All three lines use the position of maximum camber as half chord, therefore for the 50c_LE and 50c_TE curves, the static part of the blade will remain to have the same shape as a standard NACA0015 aerofoil. Interestingly, the LE deformation model is almost identical to the NACA0015 up until 90° azimuthal,

at which point the blade chord is perpendicular to the wind direction, before being significantly improved for most of the rotation, before lining up with the NACA profile again at $\sim 330^\circ$. Conversely, the TE only model starts slightly higher ($\sim 2\text{Nm}$ more) than the NACA model, but then is lower from $\sim 30^\circ$ until 100° azimuthal, and maintains that trend through the remainder of the rotation, with a particular increase in the downstream half of the rotation. Both these findings suggest that the LE model assists with reducing stall at the high angles of attack experienced in the upstream rotation ($0^\circ \rightarrow 180^\circ$), but then the trailing edge model can extract more energy in the downstream rotation, either due to the reduced energy extraction in the upstream rotation, or due to simply improved aerodynamic efficiency in the downstream rotation. Overall, the LE model has the highest average moment generation, followed by a $\sim 0.5\text{Nm}$ reduction for the NACA515 model, then another $\sim 0.5\text{Nm}$ reduction for the TE model, and then finally a comparatively large reduction for the standard NACA0015 aerofoil of $\sim 2\text{Nm}$.

As a result, although the LE model has marginally better average moment than the NACA515 model, the larger variance in the moment generation would lead to a larger structural fatigue on the blade. The NACA515 model appears to strike a balance between the LE and TE models, namely; the reduced upstream performance of the TE model, followed by the impressive performance of the LE model from 90° to 180° , before the improved downstream performance of the TE model.

NACA0015 vs GA Optimised Transient Camber Profile for Deflection Sections

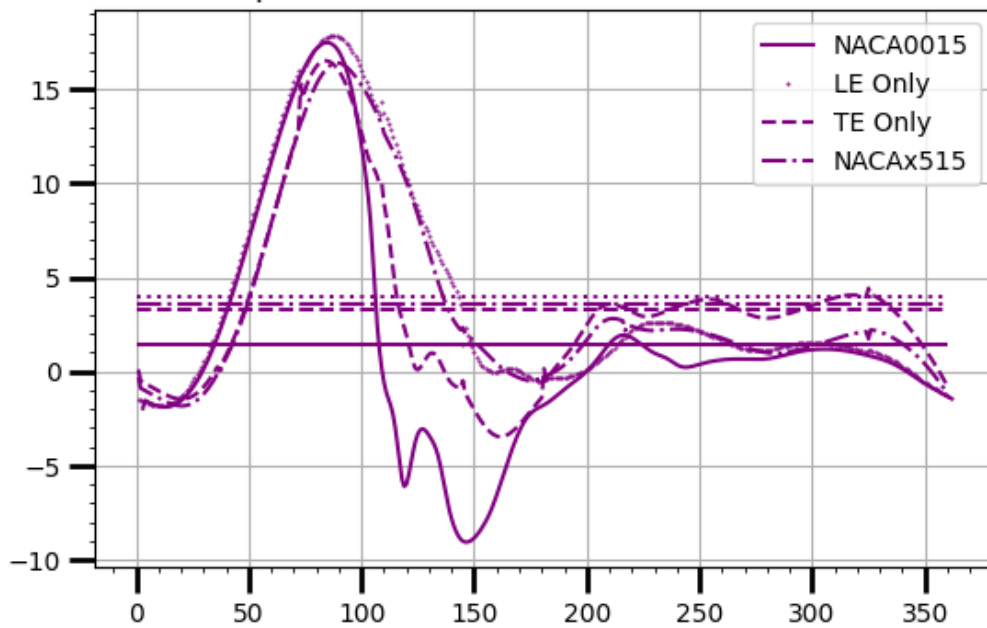


Figure 5-15: Moment Distribution for a Standard NACA Aerofoil Shape, Leading-edge only Deformation, Trailing-edge only Deformation and combined NACAx515 deformation model.

Figure 5-16 shows the camber magnitude profile of the LE and TE optimisations. The vertices signify the optimised azimuthal positions, and linear interpolation is used between each point, as has been shown (i.e. Figure 5-3, Figure 5-5 and Figure 5-10) in the chapter. The dotted line represents the TE only deformation, the dashed line is the LE only deformation, and the solid line represents the deformation based on the traditional NACA 4 Series formulation. Interestingly, at 36° , the NACA and LE model have the same camber magnitude, before the LE model increases to $10\%c$, and the NACA model is slightly reduced (suggesting better stall resistance of the LE model) but then the opposite is seen at the following optimised position, with the LE model slightly lower than the NACA. The lower LE than NACA could be a result of the LE model having a rapid loss in lift from the previous position, but then the trend returns at 108° azimuthal.

NACA0015 vs GA Optimised Transient Camber Profile for Deflection Sections

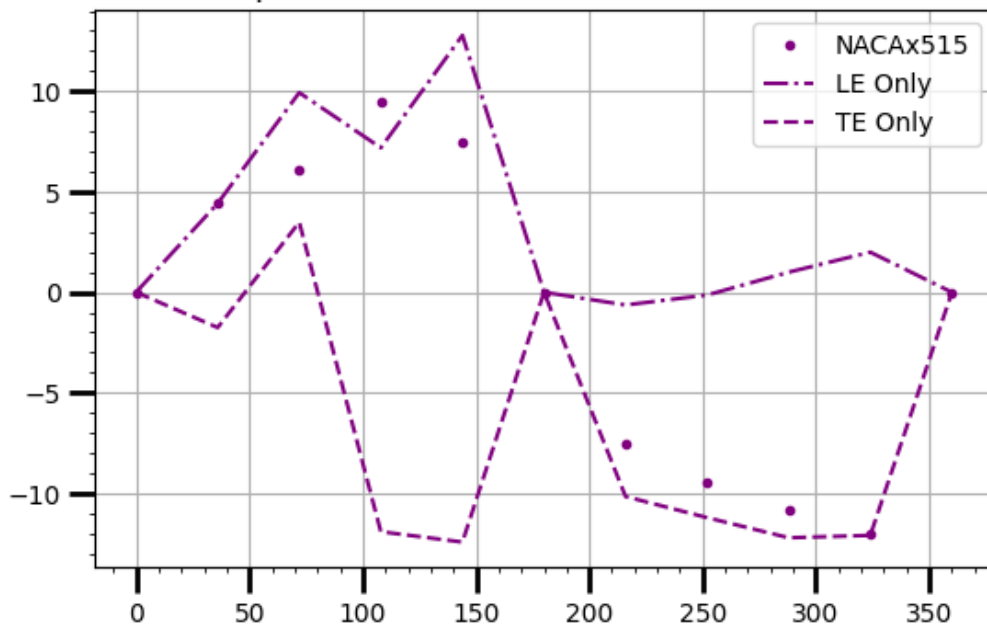


Figure 5-16: Camber deflection magnitude profile for the standard NACA0015 Profile, Leading Edge only Deformation and Trailing Edge only Deformation.

5.4.3.1 Summary of Findings

Based on the study of the genetic algorithm variations, the best camber profile was found using the 5 Phenotype, 10 Samples with Randomised Crossover techniques with a deflection center of 50%c. This camber profile is seen in Figure 5-10, and allowed for a significant increase in the average blade moment from 1.42 Nm to 3.56 Nm! Figure 5-17 shows the incredible improvements in average turbine blade moment, with an increase of 250.7% for the best configuration, and even 178.9% for the worst GA configuration investigated. These increases were all primarily due to the reduced stall magnitude of the optimised turbine blade profile, resulting in significant improvements in turbine performance in azimuth range from $100^{\circ} \rightarrow 180^{\circ}$. The optimised profiles also provided some improvements over the standard NACA0015 aerofoil in the downstream turbine rotation from $180^{\circ} \rightarrow 360^{\circ}$.

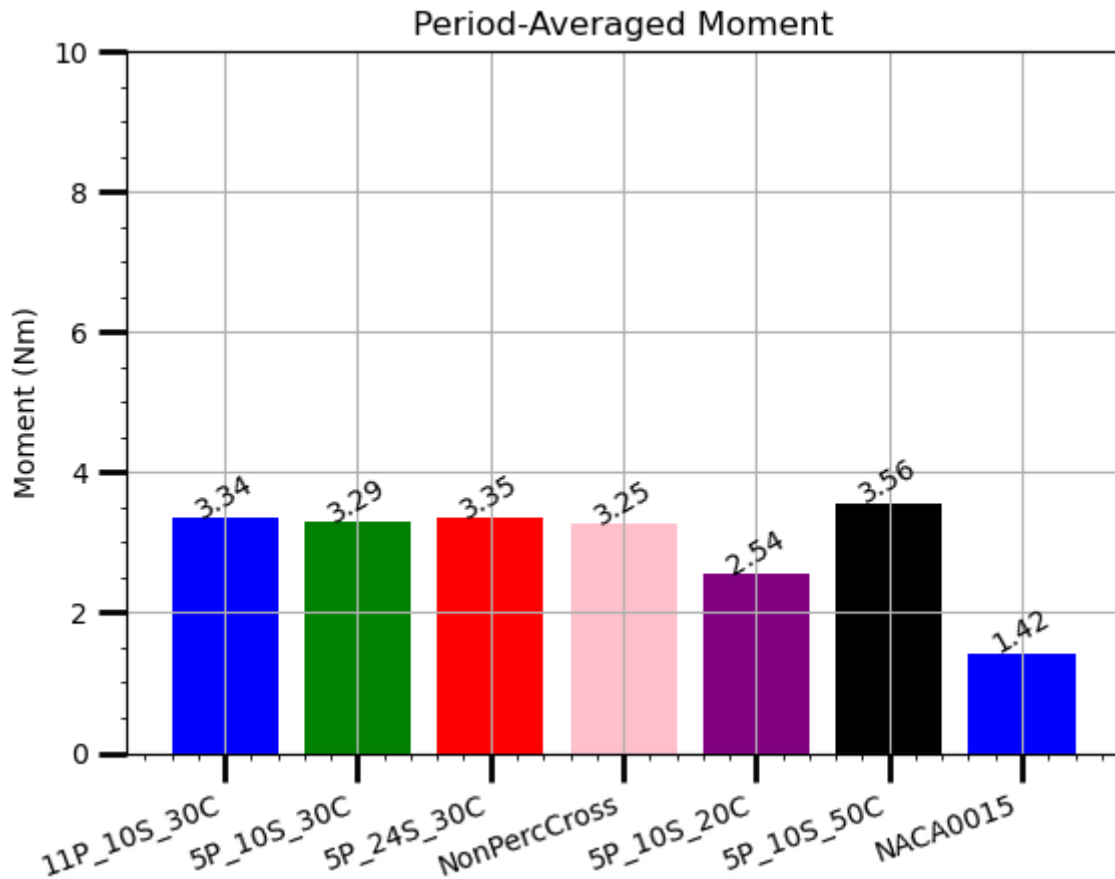


Figure 5-17: Average Turbine Moment over one period of rotation

5.4.4 Optimisation of VAWT blade camber profile at TSR = 1.5 &

TSR = 3

Due to the extensive number of genetic algorithm parameter comparisons made in this aerofoil parameter study, including the genotype count, sampling count and maximum camber position, only the optimal configuration is selected for comparison at alternative TSRs to the NACA0015. This was shown to be the 5-genotype model, with 11 optimisation positions and the position of maximum camber at 50%c.¹

To ensure the validity and the understanding of the limitations of the transient simulation genetic algorithm technique, it was tested at a lower tip speed ratio of 1.5 and at a higher tip speed ratio of 3. The primary consideration in comparison to a tip speed ratio of 2.29 is the much higher angle of attack that the turbine blade profile

would experience at low tip speed ratios and the lower angle of attack at the higher tip speed ratio of 3. The secondary comparison is the difference in Reynolds Number (as a result of the change in airspeed that the blade experiences) at different tip speed ratios, as it is well known that the Reynolds Number directly influences the lift and drag experienced by an aerofoil section [114].

Examining the 1.5 tip speed ratio case first, the torque against azimuthal angle plot can be seen in Figure 5-18. The clearest improvement is in the mean torque that the blade experiences, increasing from 0.05 $\rightarrow$ 2.8 for the optimised camber profile compared to the non-deforming NACA0015. The primary region of the moment contribution is from 225° to 360° where the turbine blade experiences high angles of attack, but also experiences a lower Reynolds number due to the reduced flow velocity that is experienced in the downstream half of the rotation. The optimised profile has a lower torque contribution in the 180° to 225° rotational region, which is a result of the optimised deformation profile not experiencing the same high intensity vortex shedded from the blade that is carried downstream which is exhibited in the symmetric NACA0015 case. The lack of vortex is a result of the change in the upstream rotation of the optimised profile reducing separation, and therefore resulting in a more uniform trailing flow field experienced by the deformed blade. The reduction in moment generation could also be a result of a lower mean flow velocity due to a greater power extraction from the flow in the upstream region.

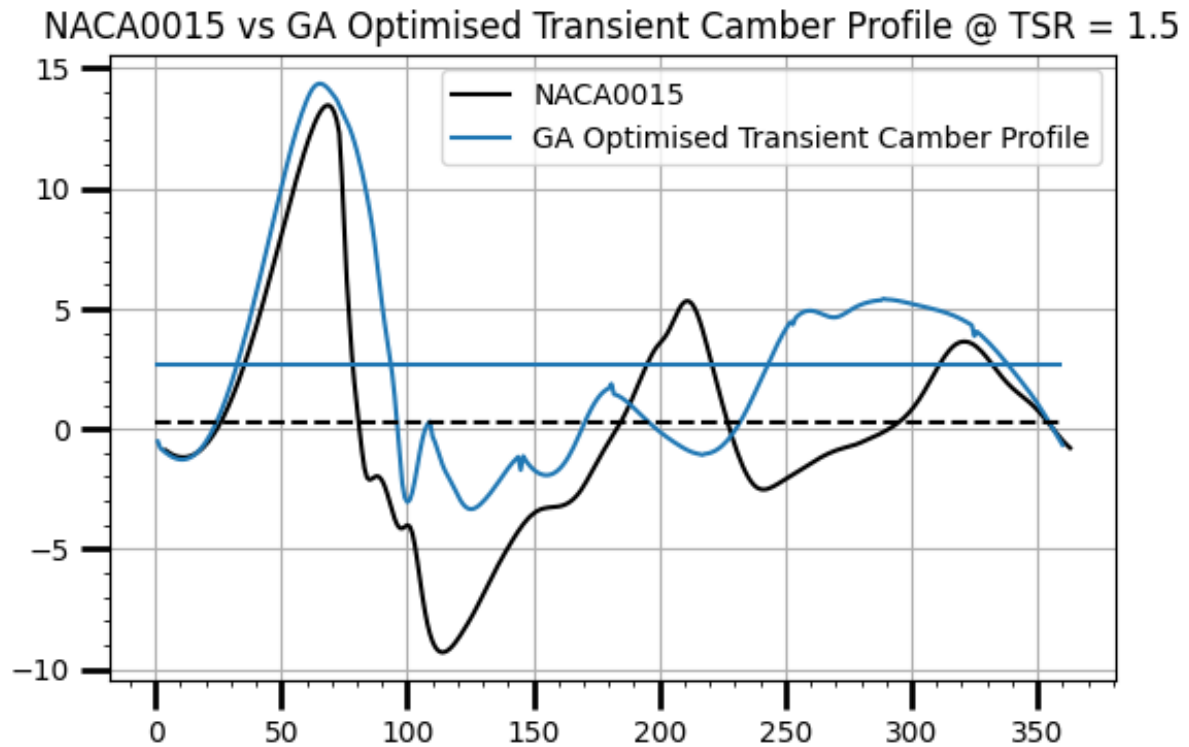


Figure 5-18: Comparison between the Transient Genetic Algorithm Optimised Camber Profile and Symmetric NACA0015 Turbine Blade Profile at $TSR=1.5$

Figure 5-19Figure 5-18 shows the deformation profile's moment generation in comparison to the non-deforming NACA0015 aerofoil at a higher tip speed ratio scenario ($TSR = 3$). The results show an interesting result wherein the upstream section is reduced in the power region, but the downstream region is entirely more performant in the optimised profile. This could be a consequence of the optimisation process only being carried out on a single period of rotation, and then running the optimised profile for 3 full turbine rotations before sampling the output. Therefore, the increased power generation in the downstream region could result in a coupling to the flow that wouldn't be seen in the first rotation (where the optimisation is undertaken). As a result, the overall turbine performance of the optimised aerofoil is slightly lower than the NACA0015, but the reduced variance in power generation between the upstream and downstream portions of the rotation could assist in the mechanical fatigue that the blade experiences.

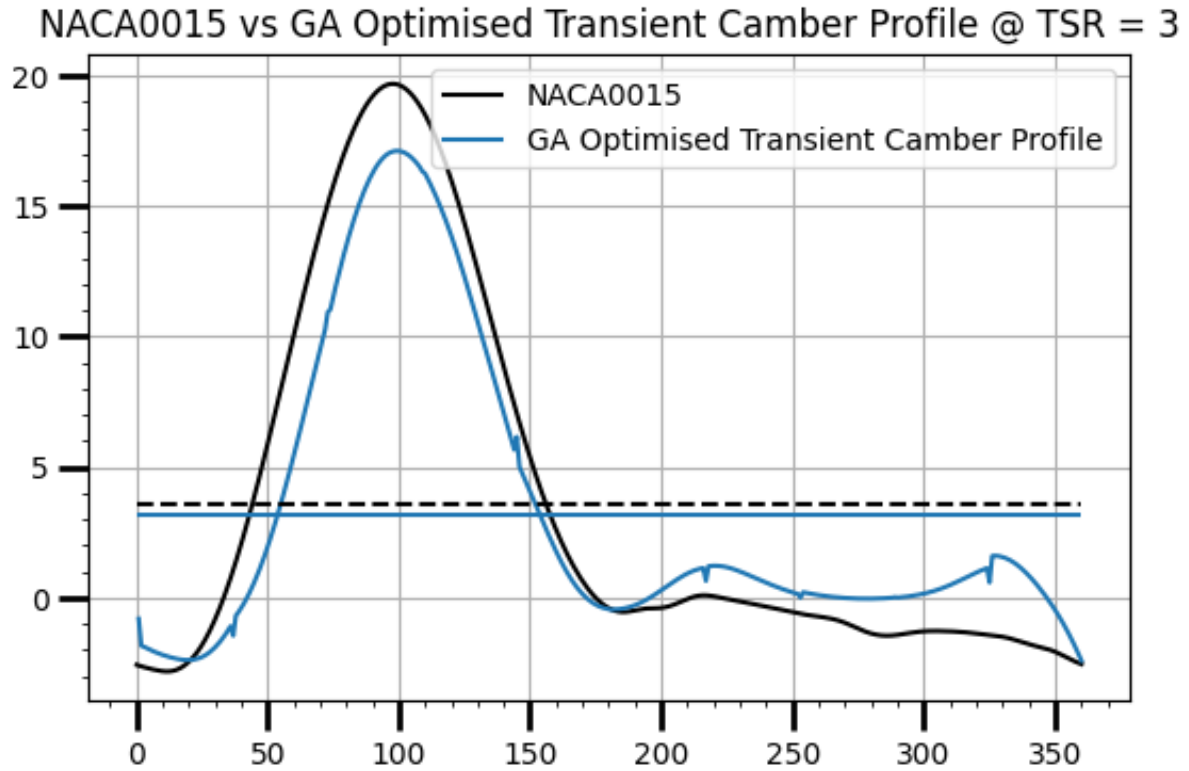


Figure 5-19: Comparison between the Transient Genetic Algorithm Optimised Camber Profile and Symmetric NACA0015 Turbine Blade Profile at TSR=3

Figure 5-20 shows the camber profile for both tip speed ratios, which provides insight into the potential causes of the moment difference with the TSR=2.29 case. Firstly, for TSR=1.5 there is a rapid change in camber between 108° and 144° , which interestingly keeps the moment above the NACA0015. This location also coincides with the rotational position where the blade starts to build up significant separation, which could be the reason that a highly negative camber is converged upon; in an attempt for the blade to reduce the separation.

The TSR=3 case interestingly has a lower camber at 72° than the TSR=1.5. Which is surprising, as one would expect the lower angle of attack and higher Reynolds number for TSR=3 to allow the blade to increase camber before separation, but this does not seem to be the case. This lower camber could also lead to the reduced moment generation from 0° - 180° than the NACA0015, as the dynamic profile change could increase the profile drag faster than the increase in lift generation.

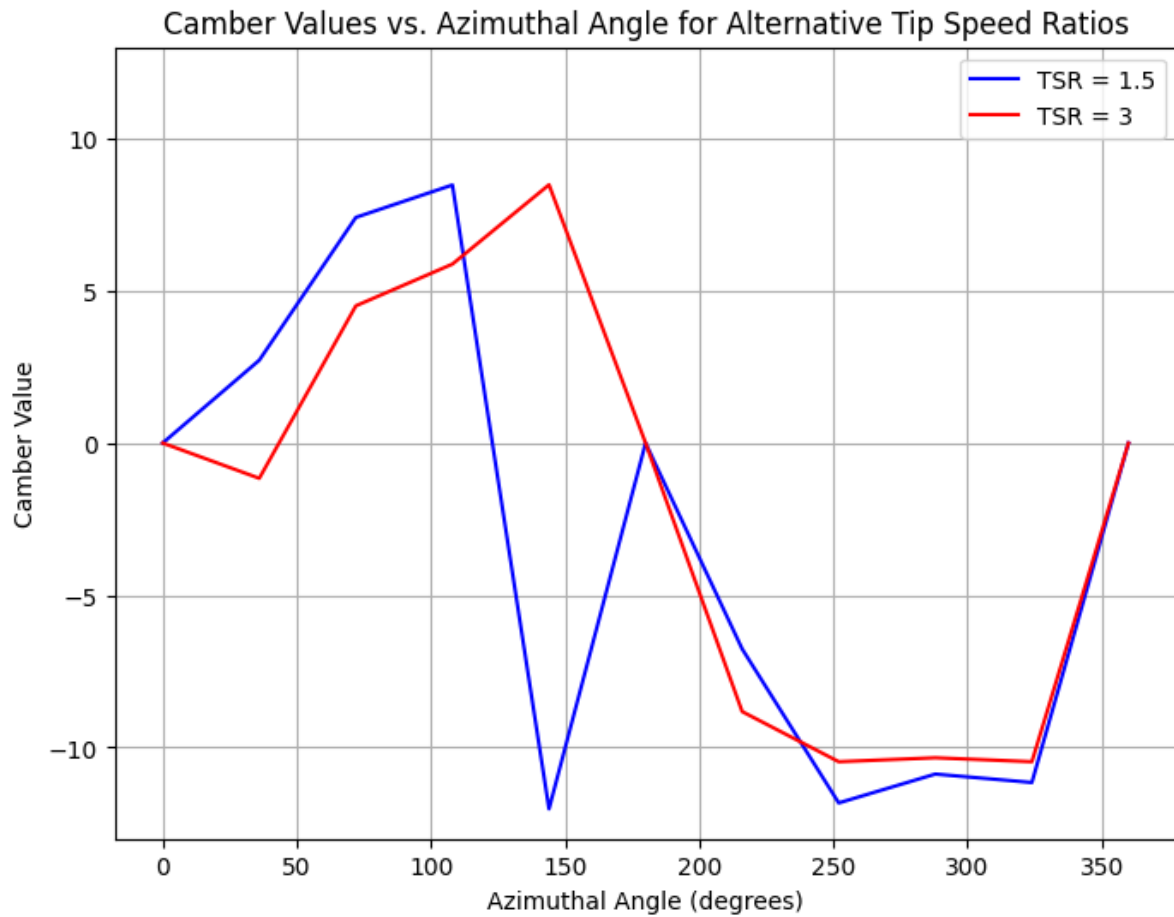


Figure 5-20: Camber profile for optimisation at $TSR = 1.5$ and $TSR = 3$

5.4.4.1 Summary of Findings

The transient GA method improved the average tangential force coefficient significantly for the lower tip speed ratio scenario ($TSR = 1.5$), but had a minor reduction in the higher tip speed ratio scenario ($TSR = 3$). This is a result of the GA being run over a single turbine rotation, meaning that the changes in upstream windspeed (as a result of blockage effects from the optimised turbine's increased downstream energy extraction). The only method to circumvent this

5.5 Summary of Findings

This chapter has shown the strength of utilising the transient CFD simulation of a VAWT for evaluating the optimal camber at user-specified positions in the turbine's rotation. Whilst other work has considered using optimisation techniques to change the aerofoil shape to improve the overall torque generation, the blade shape that they generate typically will improve either the upstream or downstream region whilst performing worse in the other region. This work introduces a new performance-inspired optimisation procedure, as each position's optimisation is used as a starting point for the next optimisation, thus allowing for continuous feedback considering the instantaneous flow fields at each optimisation position. As a result, almost all the optimised positions perform better than their symmetric NACA0015 counterparts within the GA optimisation.

A variety of configurations for the genetic algorithm setup was evaluated, including randomisation of the crossover mutations, number of samples per rotation and reducing the number of genotypes per optimisation generation. It was found that a reduction in genotype count from 11 to 5 reduced computational costs significantly, but had negligible impact on the optimised profile's power generation capabilities. Aerofoil variations were also considered including the position of maximum camber, and having a leading or trailing edge only deflection. As a result it was seen that the leading edge deflection was the largest contributor to power generation for the bending profile as a result of the reduced separation that it allowed at the tested operating conditions.

All optimised scenarios significantly improved the turbine's performance in the power generation, and reduced the variance in the power generation throughout the rotation by improving the downstream power generation significantly. This will have a knock-on effect of reducing the wear experienced by the turbine due to reduced fluctuations in the aerodynamic loading. Also, there is the potential that the reduction in separation of the turbine could also contribute to a reduction in the noise generation.

6 Artificial Neural Network Surrogate for the Optimisation of the Aerofoil Camber Deformation Profile

6.1 Introduction

This chapter leverages the unique capabilities of an artificial neural network for the optimisation of turbine blade profile aerofoil camber prediction. The artificial neural network can swiftly provide an estimated instantaneous torque value by inputting the operating parameters of a desired turbine design, including azimuthal position, tip speed ratio and wind speed. This torque and camber is fed into a genetic algorithm to generate the optimal aerofoil design at the desired azimuthal position, which is then implemented into an azimuthal position-dependent camber profile.

The training data provided is the aerodynamic efficiency (lift-to-drag ratio) of a two-dimensional airfoil section in a steady-state airflow. Each data point has a different Reynolds number and angle of attack, which corresponds precisely with the tip speed ratios, windspeeds, and azimuthal positions.

The model's fitting accuracy is evaluated using mean square error and is compared against the theoretical torque that XFOIL would predict, using the workflow described in Chapter 1. The artificial neural network-based genetic algorithm camber optimiser has shown strong consistency and high degrees of performance improvement for the turbine blade. The comparison provides valuable insights into the efficiency and accuracy of the ANN in predicting optimal airfoil designs.

Finally, the ANN profile is compared to the employment of both the static GA and transient GA to assess the vertical axis wind turbine performance improvement and the total computational efficiency, highlighting the promising potential of ANNs in enhancing the wind turbine performance, as well as considerations that should be

made when designing and utilising artificial neural networks for aerofoil performance prediction.

The genetic algorithm showed promise in achieving the optimal camber of the turbine blade profile for specific testing parameters, i.e. tip speed ratio, free stream velocity and azimuthal angle, but transient optimisation such as that employed in Chapter 2 required significant amounts of simulations to achieve the optimal camber at each localised azimuthal position, and the method in Chapter 1 struggled with optimising the camber position at high angles of attack.

This limits the accessibility of using the transient genetic algorithm for optimising VAWTs, as the user would require significant computational power or time for every specific optimisation position. For example, if the designer wants to optimise the camber magnitude at an azimuthal angle of 36° for a freestream of 7m/s and a tip speed ratio of 2.29 (one of the scenarios that was optimised in the previous chapter), they will need to run the transient genetic algorithm optimisation workflow for that specific condition. Then, if one wants to see the difference between a higher and lower tip speed ratio, one will have to rerun the optimisation algorithm with different optimisation conditions. There is no opportunity to use the previously tested conditions to help inform the current optimisation. If we take one of the previous chapter's scenarios, 36 generations of optimisation generations, for 11 genotypes, each generation will have a simulation count of 396! For this to be repeated at another tip speed ratio would double the number of simulations and for a total of 792 simulations to only optimise two positions!

This is where the potential for employing another form of optimisation, such as Kriging and Response Surface Optimisation, provide a potential advantage. These methods are used to create predictions based on trends in their optimisation dataset to allow a user to input some variables (such as tip speed ratio, wind speed and azimuthal position) and have an output provided (in the scope of this thesis, that would be a tangential force value).

These optimisers have their limitations, of course; Kriging is typically poor at optimising multiple variables, and Response Surface Optimisation (RSO) is very

susceptible to being trapped in local maxima or minima due to the methodology [115]. Kriging and RSO are typically limited to the number of variables that can be used to create the optimised design. On the other hand, artificial neural networks (ANNs) can have any number of potential inputs, hence their common usage in analysing massive datasets [116].

ANNs can be split into subcategories, with two main themes referring to how the model learns. The method of learning is referred to as supervised or unsupervised training, which has an ironic implication in that for it to be supervised or unsupervised suggests it needs to be carefully monitored as if teaching a young child, when neither method entails any supervision in the actual training process, but instead in the way that the data is used to train the setup. Supervised learning refers to using labelled training data, but unsupervised learning does not use labelled training data. The three main tasks of unsupervised learning are clustering, association, and dimensional reduction [117], whereas supervised learning has two primary categories; classification and regression [118].

6.2 Machine Learning Model

The approach used in this chapter to optimise the turbine camber profile has the same overall process as in the previous chapters; pre-process the geometry/simulation, generate the torque coefficient, and feed the camber and equivalent torque into the genetic algorithm optimiser. The novelty of this approach is to utilise an artificial neural network in place of XFOIL or a CFD simulation for generating the torque value. Many of the tasks needed to generate the artificial neural network are, therefore, a result of the automation required to easily generate, model and analyse the dataset.

First, a baseline domain was created and meshed manually, and it was modified by all data points as required within ANSYS Fluent. To generate the dataset, the main steps in the flow are to submit a substantial amount of 2D aerofoil CFD simulations with variations in the flow conditions, modify the dataset to normalise all data points (for efficiency of the machine learning), define the machine learning model, fit the

data set to the model (forward-propagation), evaluate the model (backpropagation) against the test dataset to quantify the weights of all the nodes per layer and repeat the training until the error between the predicted and tested values is sufficiently small based on the evaluation criteria decided by the engineer. The trained model can then be utilised to predict the torque for any set of operating conditions that it was trained on. In this case, the training data is the tip speed ratio, camber, windspeed and azimuthal position, which is modelled to estimate the torque of the aerofoil profile. Once trained, predicting the torque coefficient is almost instantaneous, resulting in the potential to create a turbine blade camber deformation profile in almost real-time, potentially significantly reducing the overall energy costs in computing when optimising multiple positions.

6.2.1 Dataset generation method

It is crucial to establish a dataset for the training model that is sufficiently broad to cover a variety of optimisation scenarios, but it is equally important to ensure that the difference between the data points is not excessively large. If the datapoints are too sparsely distributed in the operating conditions range, it could result in poor generalisation of the model, and if they are too dense it will increase the computational cost significantly, and risk overfitting the model.

Space optimal filling [119], a technique used for design of experiments to maximise breadth with the least experiments possible, is used to generate a series of randomised two-dimensional CFD runs of a static two-dimensional aerofoil with different operating conditions equivalent to the theoretical flow conditions that a rotating turbine would experience. Whilst it is understood that static aerofoils do not experience the dynamic flow physics of a VAWT, the computational cost of producing a VAWT ML prediction model would be exorbitant as every possible datapoint would need a transient simulation with very careful consideration for the turbine operating parameters, particularly the azimuthal angle of the datapoint. Each CFD simulation is solved to a satisfactory convergence level of $1e-4$, and the aerodynamic efficiency is appended to a dataset compiling operating conditions and resultant torque value,

which is used for training the artificial neural network. In this thesis, the dataset's variables would be the tip speed ratio, windspeed, azimuthal angle and camber as inputs and the blade torque as the output. Some example techniques for generating the parameters of the dataset are the Full Factorial, Fractional Factorial design and latin hypercube sampling.

6.2.1.1 Full Factorial

The full factorial method [120] employs changing only one variable between each data point. This is the most robust and extensive method for designing an experiment plan, but also the costliest. To appropriately quantify the data set, it requires a factorial number of experiments, hence the name. In an example distribution, if one would like to test two variables, say the tip speed ratio and camber, but one wanted, say, ten samples for each variable, then this would require 100 experiments to cover all of the possible combinations of the two variables. The mathematical representation of the number of possible combinations is straightforward and is seen below in Equation 3.1, where N is the number of samples and Y is the number of variables.

$$\text{Combinations} = N^Y \quad .$$

6.2.1.2 Fractional Factorial Design

This is a derivative of the full factorial model but uses a reduced subset of the sample range. Instead of sampling say 10 different camber values from -10 to 10, we could just do 3 samples that span the domain, but with larger spacing between each value, i.e. the 3 values for camber could be -10, 0 and 10. This would still result in several combinations that are directly proportional to the testing range but would allow for a significant reduction in the computational expense for dataset generation due to the reduced number of experiments. Of course, one of the major downsides of having a uniform distribution of samples is that the full factorial will always be the most robust sampling method as it covers all potential combinations of the variables within

the experimental domain, thus providing a more thorough and robust dataset to train the machine learning model on.

6.2.1.3 Latin Hypercube Sampling

Latin hypercube sampling [121] is a method for generating a semi-randomised sample of parameters. The randomisation allows for the reduction in the bias of the dataset, but the breadth of the design is optimised such that it spans the full breadth of the search domain. The true advantage of the Latin hypercube sampling, however, is its ability to utilise any number of design parameters, making it particularly suited and often used in the generation of datasets used for the training of machine learning models, due to their expansive training variables that are used.

As the turbine's characteristics in this case are known already (the turbine, blade profile chord length and azimuthal angle) it is possible to derive the angle of attack and local windspeed on the turbine blade when provided with the tip speed ratio, free stream velocity and azimuthal angle, utilising the equations outlined in **Section 2.4**. The aerofoil characteristics used to characterise the dataset in this study is simply the camber magnitude. As was found in previous work (namely **Chapter 4**) the optimal position of the maximum camber is at the half-chord of the aerofoil, and therefore as such, the position of the maximum camber is set as constant for all the testing conditions. Furthermore, this study only considers the effects of camber, rather than including the thickness distributions of the aerofoil (maximum thickness as percentage of chord), to provide findings that could be relatively easily implemented into future VAWT designs that utilise a NACA0015 blade and can be built and controlled in a relatively straightforward manner. Future work should include increasing the breadth of the testing dataset to train the model on more potential turbine designs and improve the breadth of application of this technique.

This generated dataset is used to train the ANN model, as these input variables allow for the widest range of application for this turbine. If one wished to optimise a different profile, this model should still be valid even at different operating conditions such as wind speed, tip speed ratio and chord length.

6.2.2 Two-dimensional static airfoil simulation for Machine Learning Model Generation

To generate the training data for the machine learning model, a Python script was created to automate the geometric creation, make the mesh, set up the simulation, and calculate the solution data.

Whilst it is possible to automate this process using XFOIL, as was done in earlier work (Section 4) there would be limitations when reaching the edge cases, such as the high stall angles with low Reynolds number and large camber. It was, therefore, decided that ANSYS Fluent would be used to evaluate the performance characteristics of each data sample. This also provides the additional option for further work to be implemented wherein other characteristics can also be implemented within a machine learning workflow, potentially utilising unsupervised learning to find correlations between characteristics such as separation, vortex shedding frequency and overall turbine performance. Furthermore, using ANSYS Fluent enables the additional option to implement a qualitative approach to the candidate comparison, as the user can infer how the different turbine operating conditions and design choices can impact on the overall flow field, not only the quantitative performance improvements.

The general workflow is as follows:

- i. Initialise the array of candidate simulations for the dataset generation
- ii. Use the dataset parameters to generate the appropriate geometries
- iii. Mesh the geometries created in step ii
- iv. Setup the simulation using operating conditions assigned in step i
- v. Solve the flow field using the mesh from step iii
- vi. Take the relevant data from the simulation results and append to the original dataset
- vii. Generate the AI model for the predictive simulation

- viii. Utilise the AI model as the torque prediction tool within a genetic algorithm to find the optimal camber for the desired operating parameters.

The design plan is to create a highly structured flow domain around the aerofoil, controlling the stability of the simulation well and the placement of the elements to maintain the appropriate level of mesh resolution required in key areas of the flow field. The angle of attack would then be controlled by changing the components of the incoming airflow and the lift and drag forces used to calculate the blade's torque generation.

The entire geometry, mesh and solution steps is controlled through the PyAnsys libraries, allowing for the easy parameterised automation of the potentially thousands of datapoints that will be generated for the machine learning training dataset.

The geometry generation script takes the provided NACA aerofoil parameters (camber, thickness, maximum camber position and chord length), allowing for any NACA 4-Series aerofoil to be generated as required when creating the training dataset simulations. This script creates a list of tuples containing the XY coordinates of the aerofoil surface points. Initially, this script made very blocky curves with sharp edges, so it was altered to create segment lines, merged into a single spline of best fit in Spaceclaim [122]. If a flat trailing edge were to be used, it would be necessary to implement some form of flag in the geometrical construction to label the coordinates of the trailing edge node for both the pressure and suction surface, and create a line between them, and a line to split the decomposition into an additional surface.

There have been some investigations that look into the shape of the trailing edge and its influence on the aerofoil performance prediction accuracy [123][124][125], of course, an infinitely sharp trailing edge is not manufacturable [126], but the flat trailing edge can lead to significant inaccuracies in the aerofoil's wake from the increased separation compared to a rounded or sharp trailing edge, especially in the stall region. The domain decomposition includes a vertical split at the aerofoil trailing edge and 10 percent chord length downstream of the leading edge to help reduce

element skewness; a split also spans from the O-Grid's curve downstream of the boundary's midspan to the aerofoil leading edge, and from the trailing edge to the midspan of the furthest downstream boundary of the domain. Figure 6-1 shows this domain region for clarity. This is decomposed into quadrilateral regions to allow for strong mesh control to create a fully structured conformal mesh with a maximum y^+ value of 1, as is required by the $k - \omega$ turbulence model [127]. Figure 6-1 shows the resulting domain decomposition after implementing the philosophies discussed, with the shared edges visible as the blue lines.

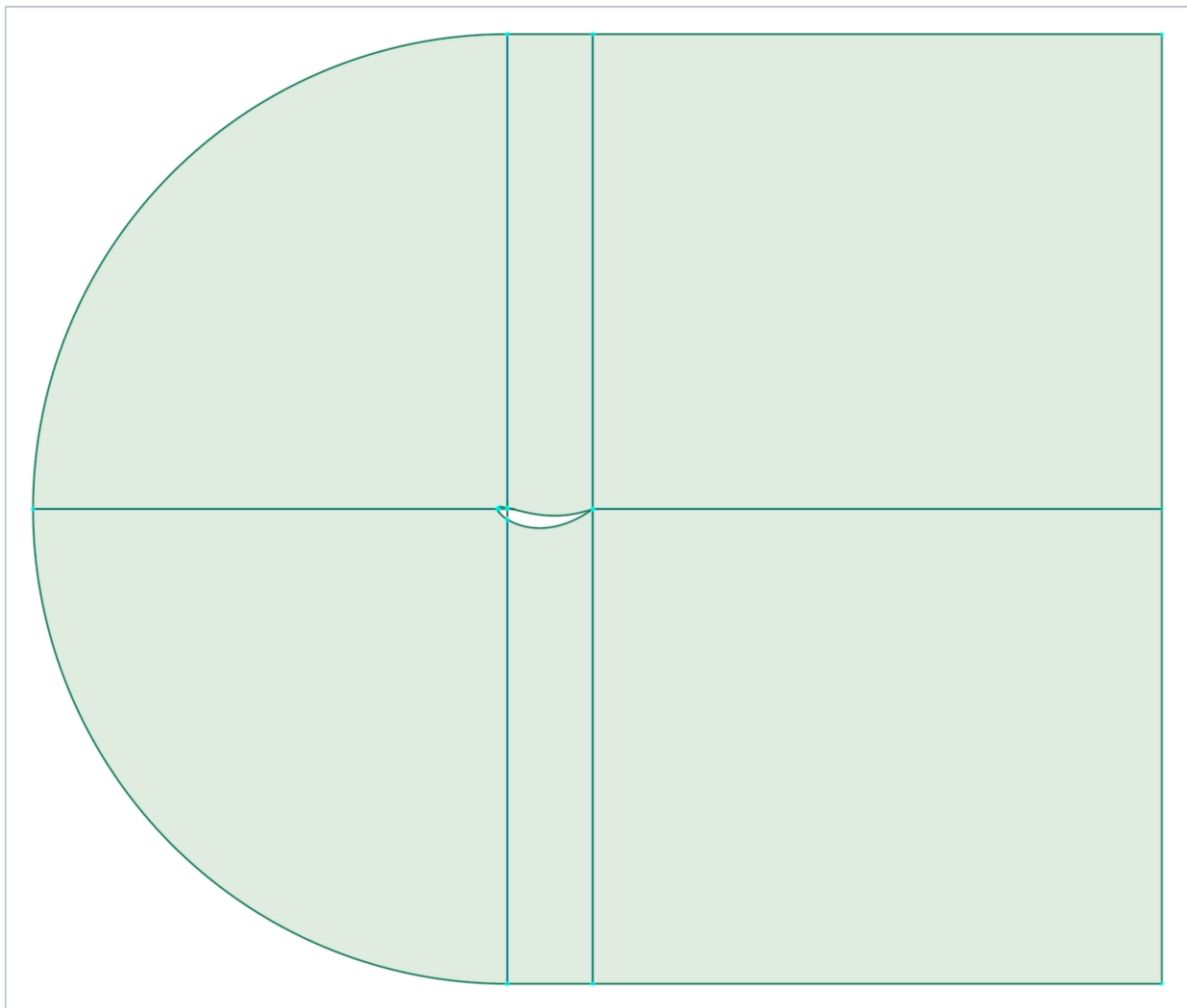


Figure 6-1: Preliminary 2D CFD Domain Block Decomposition

The sizing on all the topologically shared edges was used for implementing a gradual inflation from the aerofoil surface, alongside controlling the number of divisions along the aerofoil edge, thus allowing for a more straightforward study of mesh independence and control of the overall mesh resolution. This also makes it easier to

control the aspect ratio of the cells and y -plus in the boundary layer regions. One of the most significant benefits due to the structured mesh being fully defined, however is actually in the speed of the meshing and the simulation [128]. A structured quadrilateral mesh also provides the benefit of reducing the numerical diffusion [129], particularly in the flow regions that are typically linear, such as within the attached boundary layer [129].

Upon saving the generated flow domain geometry, it meshes within SpaceClaim for the controlled meshing along the named selections generated in the geometry file using the edge sizes described before, including a 0.2mm element sizing on the aerofoil surface and all corresponding edges of the domain which are topologically parallel to the aerofoil. This element sizing reduces skewness when allocating a first cell height of 1.94e-5m on the aerofoil surface to ensure a $y^+ < 1$.

As the named selections are pre-defined in the geometry generation stage, they are used to specify the edge sizes that are unique to each region more easily. The final mesh is then imported into an ANSYS Fluent case with the boundary conditions set up for the simulation. The conditions include symmetry on the top and bottom edges of the solution domain to reduce the wind tunnel effect of having a domain that is too small, a velocity inlet and a velocity outlet upstream and downstream of the aerofoil, respectively, and no-slip walls on the aerofoil edges, which is necessary to enable the viscous flow effects for proper aerodynamic evaluation, such as the circulation needed for generating lift. If the aerofoil walls were slippery (no-shear stress on the fluid-wall surface), then the lift and drag of the aerofoil would not be well captured.

6.2.3 CFD Validation and Verification

In order to ensure that the simulation setup is effective and accurate, it is compared against the trustworthy source of Elsakka et al.'s study on the 'Angle of attack estimation for a VAWT turbine' [80], which is in turn validated against the work of Li et al. [130] on the 3D aerodynamics of VAWTs. There are two main methods of simulation accuracy comparison that were used; Lift and Drag polar comparison, and Pressure Coefficient along the aerofoil surface.

Using the pressure coefficient on the aerofoil surface is a more robust method than using lift and drag, as it allows for more nuanced evaluation such as if more refinement is required in certain regions around the blade. Using the pressure coefficient distribution provides a more nuanced understanding of the flow characteristics, as the lift and drag are calculated directly from the aerofoil surface pressure distribution, but don't provide the details about how the flow is behaving, especially at higher angles of attack [131].

6.2.3.1 Preliminary Design

The initial design was to have a common C-Grid for the domain shaping which connects with a rectangular section, seen in Figure 6-1. This proved to be a poor design choice as the shaping resulted in large skewness and poor orthogonal quality as the elements were blocked between the arc inlet and the aerofoil edge. Some alternative domains and setups were studied including:

- i. A flat trailing edge – 5c upstream arc, 7c downstream arc
- ii. Flat trailing edge + Blocked Domain – Worse convergence than FF, same domain
- iii. Flat trailing edge with blocked domain – 10C upstream, 30C downstream
- iv. Sharp trailing edge with blocked domain – 10C upstream, 30C downstream

However, after much refinement, it was found that it was not possible to create a valid comparison to the published results, and as such, the structured meshing approach needed a rethink. This is mentioned merely for the reader to avoid making the same design decisions if undertaking a verification study of their own, so the pressure coefficient plots for the poor designs will not be included.

6.2.3.2 Final Design

The best course of action would be to replicate the 2D design provided by El Sakka et al. [94], as the domain methods mentioned briefly in Section 6.2.3.1 had very different pressure coefficient plots to the published data. This consisted of an aerofoil within a small circular domain with a structured mesh, connected by a non-conformal

interface to a structured far field mesh which is rectangular in shape. The geometrical design with dimensions is shown in Figure 6-2. This allowed for the aerofoil domain to be rotated as required to the desired angle of attack without any need to alter the inflow boundary conditions, leaving the inlet flow perpendicular to the inlet face.

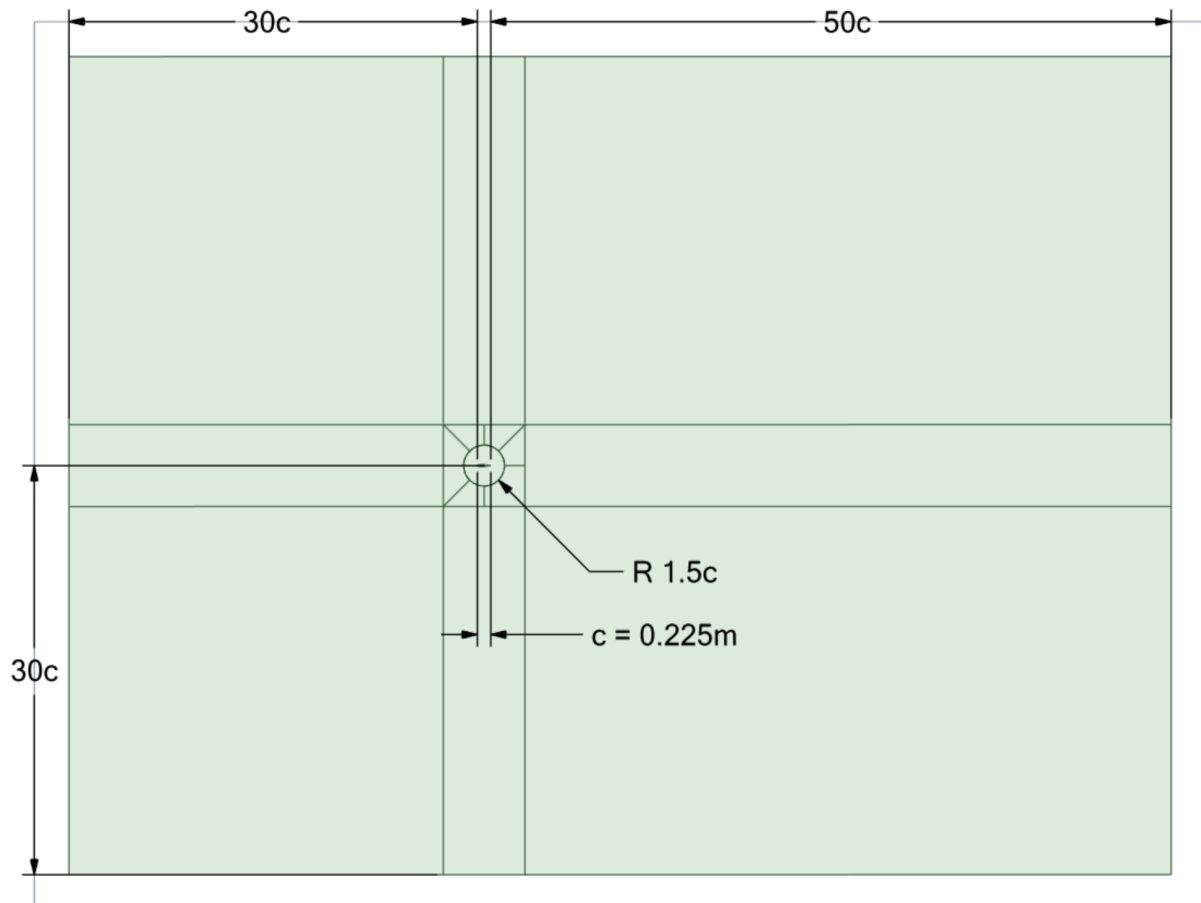


Figure 6-2: The 2D CFD domain with the blocking lines visible

The mesh is generated such that the first cell is sufficiently accurate to capture the viscous sub-layer in the boundary layer, but it is also designed such that the elements are concentrated in regions of large gradients, particularly near the blade and in its wake. An O-Grid was also blocked around the aerofoil to assist in the transition of the mesh, reducing the skewness and allowing for better control of the region near the aerofoil, whilst maintaining a coarser mesh in the far field where flow gradients are not as likely, and therefore a coarser mesh will suffice [132]. The resultant mesh sizes include $\sim 0.2\text{mm}$ element edges along the aerofoil blade and $\sim 0.05\text{mm}$ edge sizing at the curved trailing edge region to reduce the skewness in the highly curved region.

Biased edge sizes were also placed throughout the far-field of the domain, with small elements close to the aerofoil domain progressively increasing in size to the boundaries. 110 divisions were allocated for all edges normal to the aerofoil surface with a bias factor of 421 to set the first cell height to $5.4\text{e-}7\text{m}$, followed by 50 divisions in the rotating section housing the aerofoil at a bias factor of 4. This resulted in a high-quality mesh, with the minimum orthogonal quality at 0.7 and maximum at 1. There were 100 elements with a bias factor of 5 along the 30c length edges (vertical and horizontal upstream of the blade), and 300 elements with a bias factor of 5 along the 50c wake edges (horizontal, downstream of the aerofoil) from the aerofoil to the outlet boundary. An image of the full meshed domain is seen in Figure 6-3**Error! Reference source not found.**, a zoomed image nearer to the aerofoil in Figure 6-4, and a zoomed image of the region near the aerofoil is seen in Figure 6-5 and Figure 6-6.

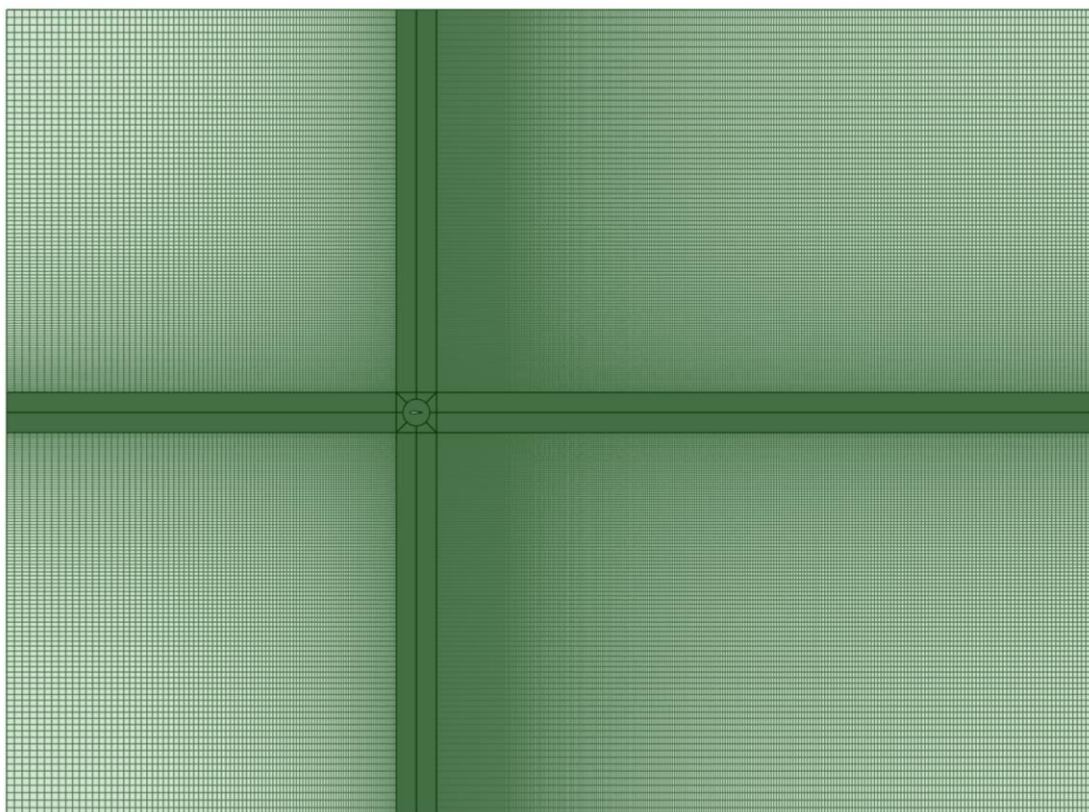


Figure 6-3: The fully meshed CFD domain for the NACA0015 aerofoil

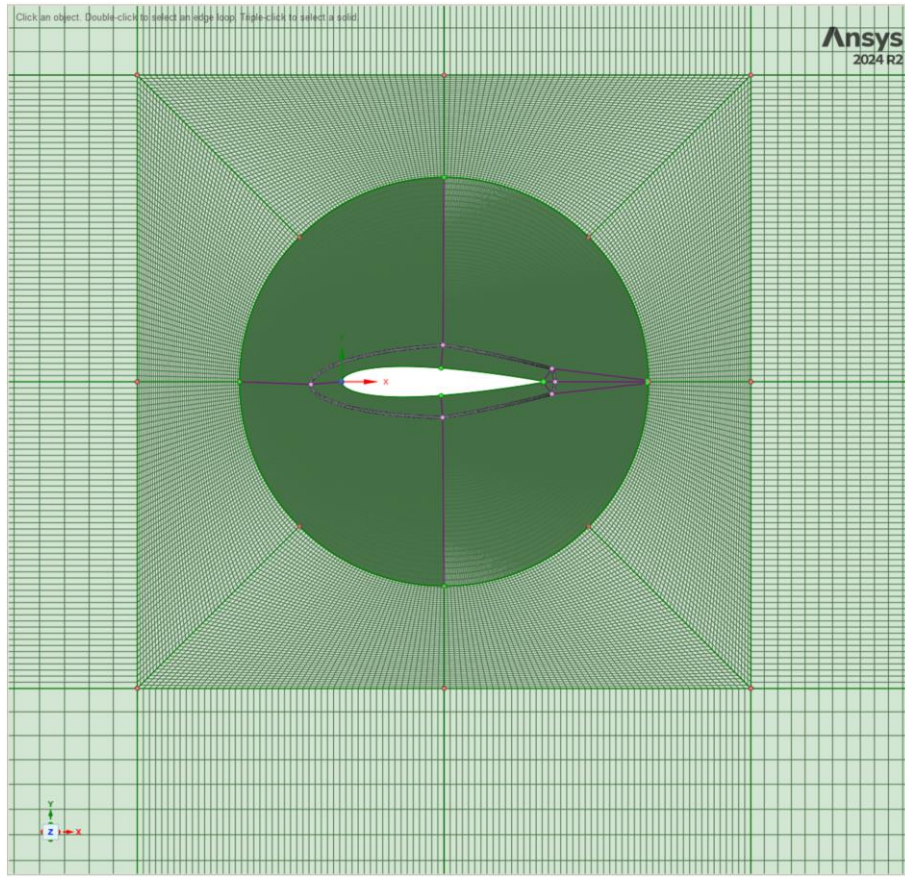


Figure 6-4: Domain mesh in the vicinity of the rotating aerofoil fluid zone

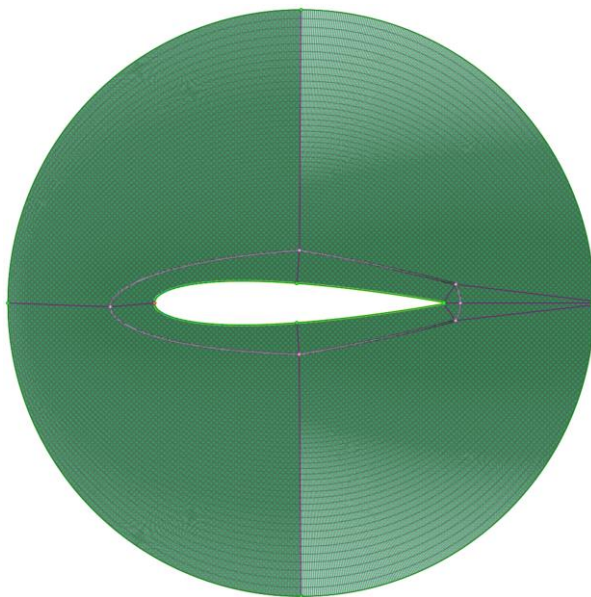


Figure 6-5: The zoomed mesh with blocking edges of the aerofoil region

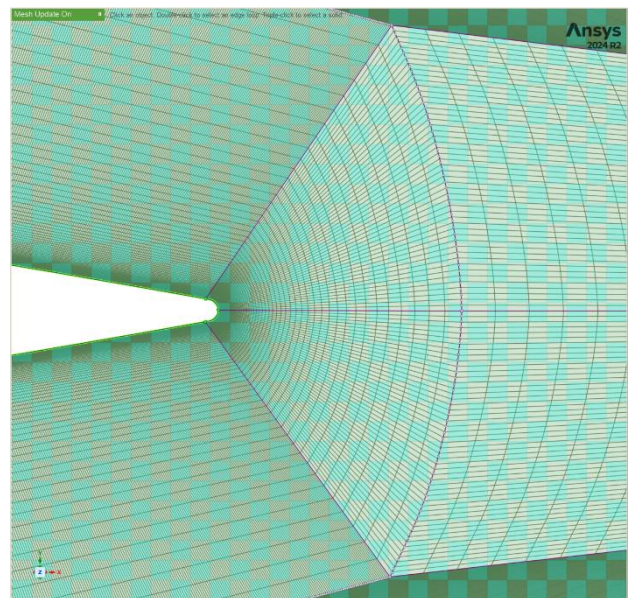


Figure 6-6: Zoomed in mesh around the blade trailing edge with blocking edges

This final mesh was implemented into an ANSYS Fluent CFD simulation, with the following boundary conditions:

1. A velocity inlet at the X-Min boundary, with air at standard atmospheric conditions, travelling at 10 ms^{-1} along the vector of $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$.
2. Pressure outlet at the X-Max boundary.
3. No-slip walls on the aerofoil surface.
4. A non-conformal interface between the circular fluid-zone containing the aerofoil (Figure 6-5) and the corresponding edge of the larger outer flow domain.
5. Symmetry on the Y-Min and Y-Max boundaries to minimise any wind-tunnel effects and prevent boundary-layer growth on outer domain edges.

The aerofoil fluid-zone is rotated by 10° clockwise to create an angle of attack of the turbine blade section of 10° to replicate the flow conditions of the static aerofoil simulation evaluated by Elsakka et al. [80]. Namely a 10 m/s wind speed and 10° angle of attack. The turbulence model used is the $k - \omega$ SST model, which is appropriate as the first cell height has a y^+ of < 1 with the mesh sizing described, and helps to predict the separation of the aerofoil better than the $k - \epsilon$ or SA models that are commonly used in aerospace applications .

Comparing the pressure coefficient along the aerofoil surface with that obtained by the work of Elsakka et al. [80] is seen in Figure 6-7. The plot shows a very strong agreement in pressure coefficient along the entire span of the aerofoil, and as such this simulation's domain sizing, mesh and simulation settings is deemed to be properly setup and can be used for generating the machine learning dataset with confidence.

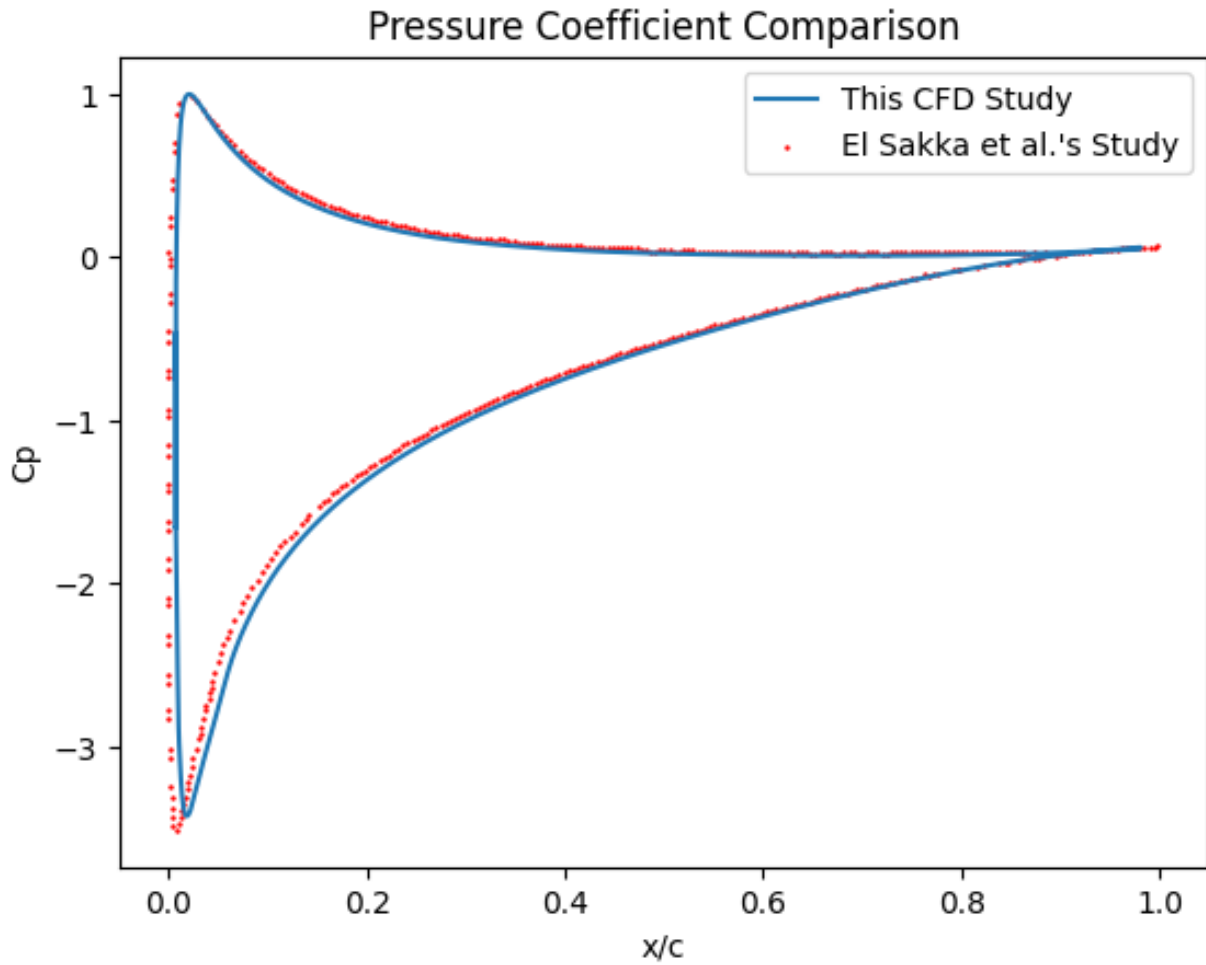


Figure 6-7: C_p Plot Comparison from the present setup in this study compared to that of El Sakka et al.[80]

To ensure this is the most computationally efficient mesh, that also captures the required flow physics, a mesh independence study was performed to compare the lift coefficients for four meshes – 260k, 350k, 585k 756k elements. The sizing variations were implemented by changing the number of divisions on the aerofoil edge, and the associated edges in the circular domain section as seen in Figure 6-5. Figure 6-8 shows that there is almost no difference between the use of the two finest meshes so the second finest mesh tested is used for all the remaining studies and the ML dataset generation to improve computational efficiency. The 756k mesh could show a reduction in lift by more accurately marking the separation point on the aerofoil surface, but it is almost indistinguishable from the 585k mesh.

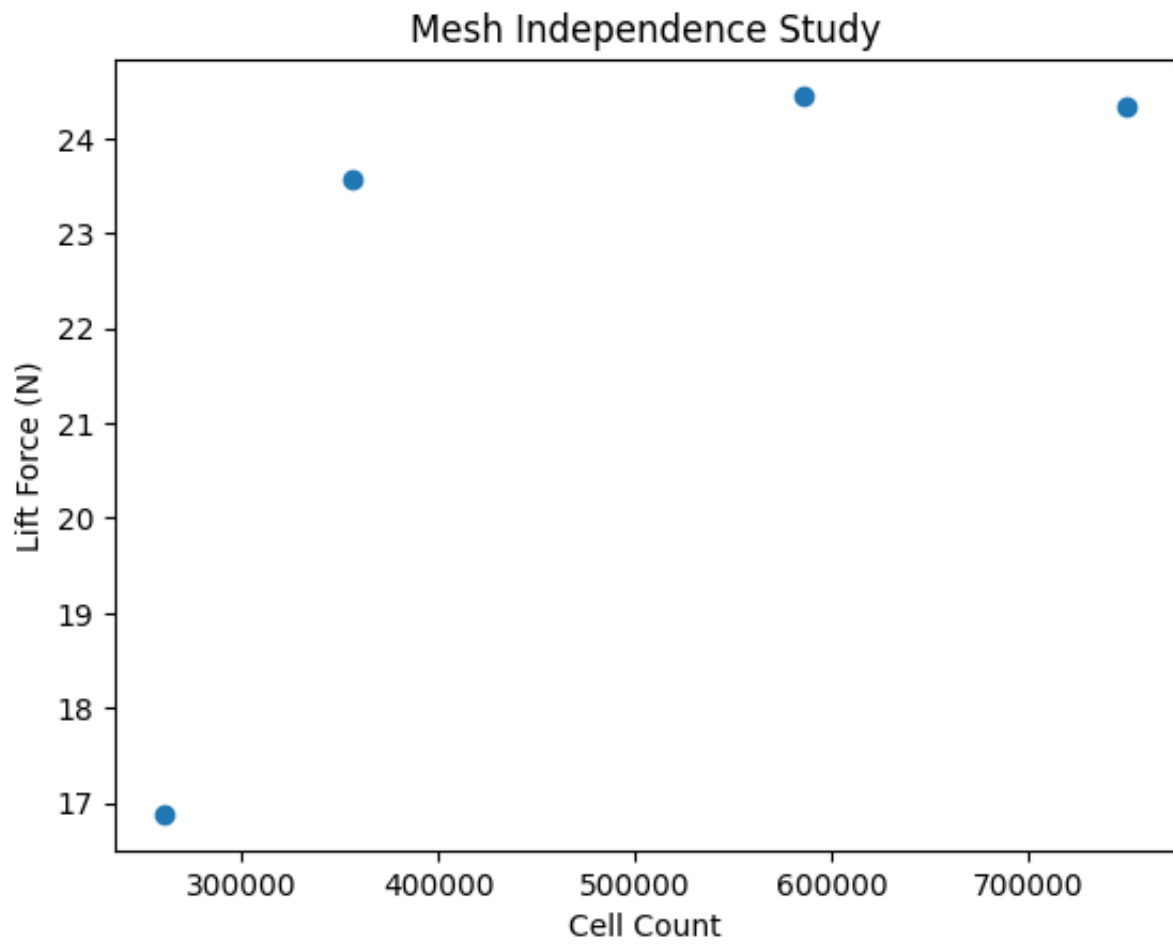


Figure 6-8: Mesh independency study for the 2D aerofoil simulation used as datum for generating the ML Training dataset.

6.3 Machine Learning Model for the Blade Torque Prediction

This section outlines the techniques for creating the ML model, testing other existing regression models, and evaluating the performance and validity of the ML Models.

6.3.1 Existing Regression Models

There are some available models and studies available [133], [134], [135] for predicting aerofoil performance using a machine learning model, but there is currently not a model that is suitable for the range of conditions that the designed VAWT operates in.

The model is therefore trained using the sci-kit-learn library [136], which is much more straightforward and has a high-level library, thus allowing for data scientists to more readily focus on the actual connections in the data and outcomes instead of the time-consuming task of fine-tuning neural networks in TensorFlow [137] and PyTorch [138]. Furthermore, it has been seen that if the dataset generated is insufficiently complex or sparse for the neural network to be fully utilised, thus likely to result in strong overfitting of the data, leading to poor regression performance and poor performance on the un-seen data. Scikit-learn [139] is used in much the same way as TensorFlow [137] but has the entire process of hidden layers and hyperparameter tuning that is hidden away from the user.

The scikit-learn based methodology was deemed to be insufficient, as the comparison between the predicted and true values was significantly poor. See Figure 6-9 for the comparison between the predicted and true values, **which clearly shows that the there is very poor fitting between the predicted and true values. A perfect fit would match the dashed red line visible. This line is simply $y=x$, or in this case the predicted value = true value, which is clearly a best case scenario in which every predicted value is the same as the trained/tested value, but is near impossible to achieve in practice.**

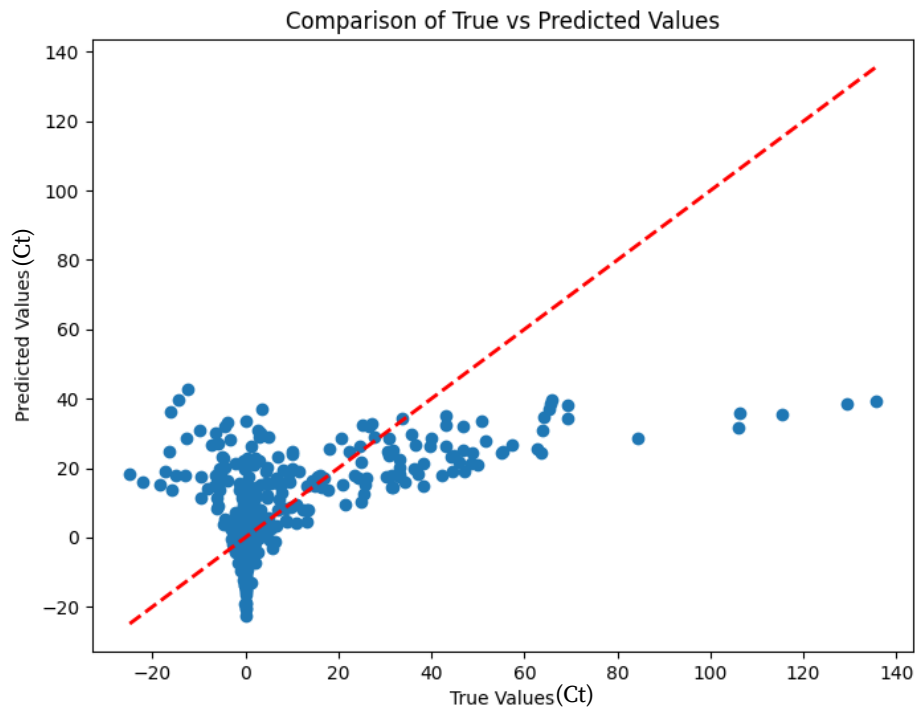


Figure 6-9: Scikit-Learn Multivariate Regression Fitting Comparison between Model Predicted Values and True Values from the training dataset.

An already established, the trained machine learning model to predict aerofoil performance was found on GitHub as an opensource project. This project is called NeuralFoil and is a neural network predictor that was trained on millions of XFOIL runs with varying angles of attack, aerofoil parameters and Reynolds numbers. This clearly displays the significant breadth of the dataset required for generating a neural network for a complex interweb of variables as is the aerofoil design and optimisation. NeuralFoil runs ~10x faster than XFOIL, making the optimisation process incredibly fast.

6.3.2 MLVAWT Artificial Neural Network

Neural networks typically have a few layers split into four categories: input, "hidden", sorting, and output. Figure 6-10 shows an example of the structure of a simplified artificial neural network.

- The input layer acts as the first step between the dataset preparation and model training, wherein the user defines the characteristics that they are interested in. This stage also applies min-max scaling to manipulate the input data into a form that is more efficient for the matrix operations when employing the training model [140], and a common step is to normalize the data to a unit range.
- The 'hidden' layers in the neural network are where the coefficients or factors of the specific machine learning model implemented are utilized to weight the different inputs as it was trained. These layers play a crucial role in the learning process, processing the input data and applying the learned factors to make predictions.
- The sorting layer plays a crucial role in the neural network. It categorises the factored/scaled data and combines all hidden layers. This process is essential for organizing and processing the data effectively, thus leading to accurate predictions.
- The output layer is the final stage of the neural network. Depending on the specific design parameters and the user's requirements, this could be a yes or no answer, an estimated value, or a specific category. In the context of wind turbine design, the output layer would be regression based, so this layer would provide the optimal camber profile for a given set of turbine operating conditions, as the user desires. As a result of only having one optimised variable, this layer “squashes” the nodes of the previous layers into a single node, but it could be expanded to as many output variables as needed depending on the scenario.

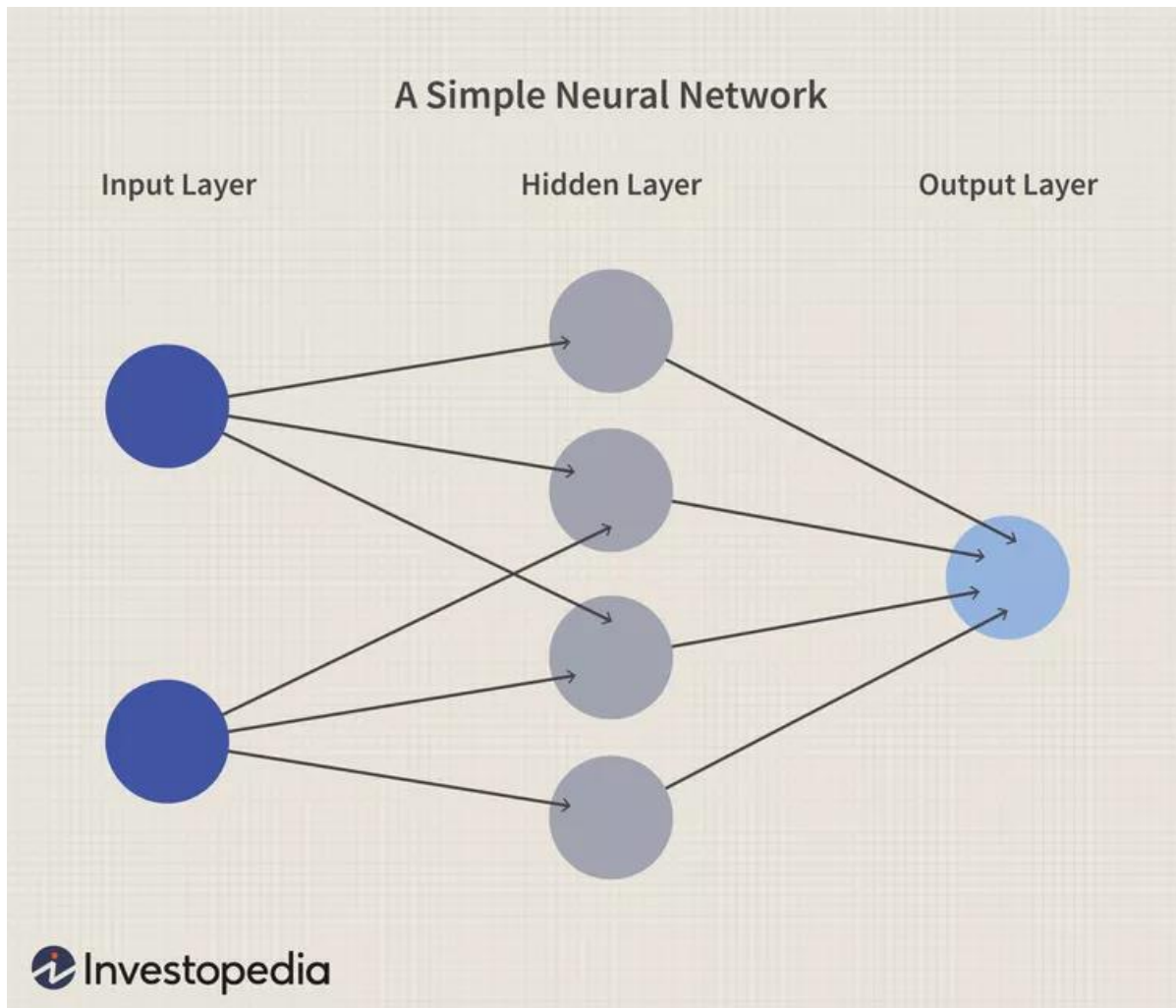


Figure 6-10: Graphical representation of the layers of a simple neural network [141]

The dataset that is generated in this application uses input parameters that are all operating conditions of the turbine that would be pre-defined before selecting the optimal aerofoil. The parameters used in this study are providing the azimuthal angle, camber, wind speed and tip-speed ratio, with the output of the model being the predicted tangential force coefficient. The range of the parameters are seen in Table 6-1 below.

Parameter	Minimum Value	Maximum Value
TSR	1	5
Phi	0	360

Windspeed	0.1	14
Camber	-9.5	9.5

Table 6-1: Bounds of the Input Variables for the Training Dataset

These parameters are used to create a set of CFD simulations to generate the tangential force coefficients for specific operating conditions, which is fed into the machine learning model to create the torque predictor, as that is the driving force that allows a turbine to generate power. It is possible to directly predict the power coefficient, but that is a factor in the operating conditions, and the simplicity of taking a value that is directly proportional to the turbine blade's lift and drag allows for less variance in the prediction. The model is used to replace the CFD solution stage employed in the previous chapter's genetic algorithm. By using the ANN to predict the torque for a blade at specified operating conditions, it is possible to optimise the turbine profile in under a minute instead of over the course of several hours.

The hidden layers are how the weights are calculated for each data point, with three primary functions: tanh, sigmoid and Rectified Linear Unit (ReLU). There are many other activation types, but they are not as commonly used. The ReLU function is the simplest function, it simply takes a number and turns negative values into 0 and lets the output equal the input if the value is positive. It is very efficient and allows the network to learn quickly, but sometimes it can lead to "dead neurons" resulting in many nodes not aiding in the training of the model if many of the inputs are negative [142]. The sigmoid function takes a value and "squashes" it to a value between 0 and 1. This is very useful in binary classifications such as A/B Testing, but not much use in regression as it is limited by vanishing gradients, where if the value is very small or very large, the learning is very small, thus slowing the training down. The Tanh function is similar to the sigmoid function but scales to -1 -> 1 instead of 0 -> 1. As it is based around 0, it learns faster than the sigmoid function, but it is also susceptible to the vanishing gradient problem. Leaky ReLU works similar to ReLU, but allows a small gradient when the value of x is small, eliminating the dead neuron issue, allowing the network to continuously learn. The mathematical representation of

these functions is shown in Equations 6.1), 6.2), 6.3) and 6.4). A graphical representation is also shown in Figure 6-11, where the X axis represents the input to the function, and the Y-Axis is the function output.

$$ReLU(x) = \max(0, x) \quad 6.1$$

$$Leaky\ ReLU = \begin{cases} x & \text{if } x > 0 \\ 0.01x & \text{if } x \leq 0 \end{cases} \quad 6.2$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad 6.3$$

$$Tanh = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad 6.4$$

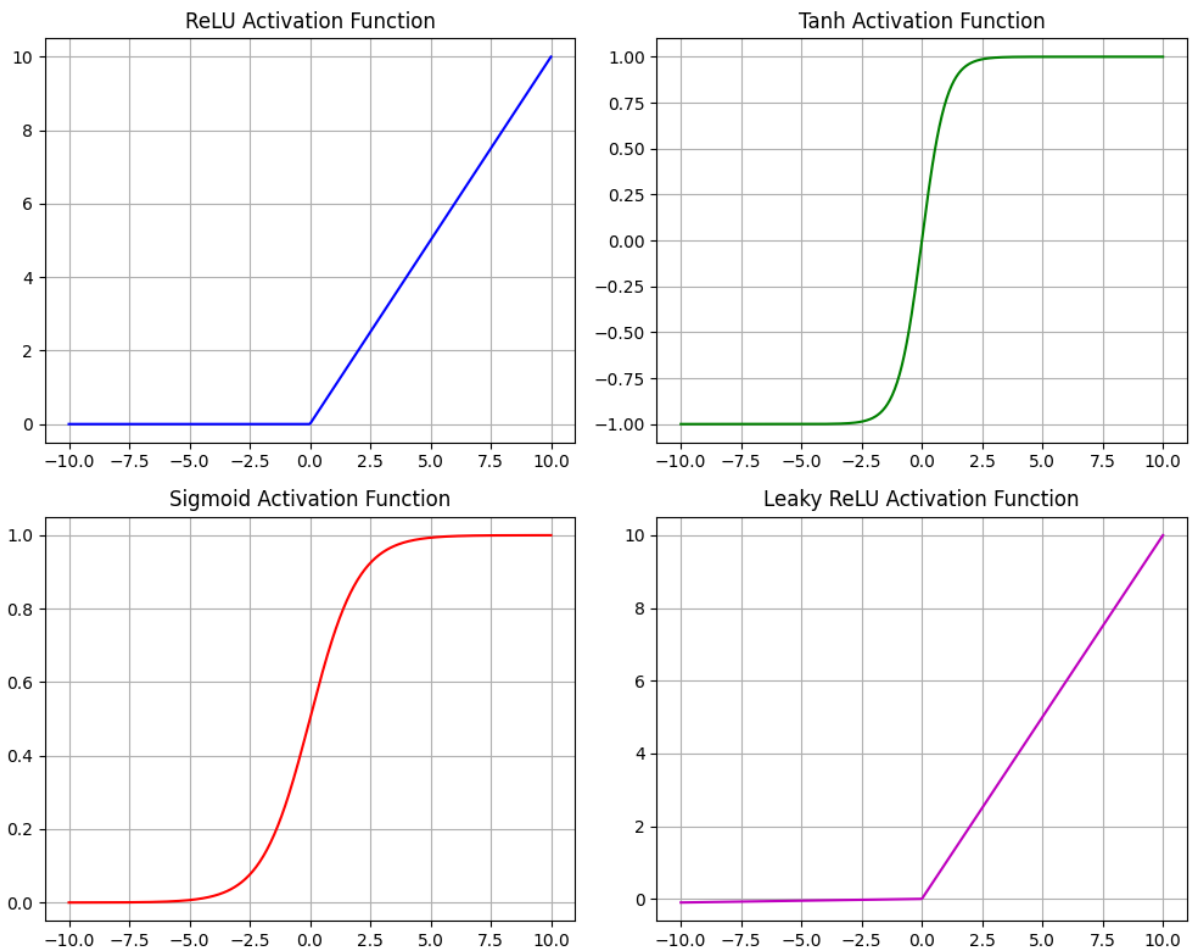


Figure 6-11: Neural Network Activation Functions for the Input vs Output

Four variations of the model were created and these are seen in Table 6-2. The machine learning model's configurations are used to investigate the effects of the

different parameters on the efficiency of the model training and the accuracy of the model. Due to significant advances in graphical processing units (GPU) in the last decade [143], it is now possible to rapidly accelerate the model training process using the GPU instead of CPU, and this is due to their thousands of parallel threads [144]. Due to the embarrassingly parallel nature and incredible matrix mathematical efficiency of GPU computing, there are also some model optimisers built into the most common machine learning libraries such as TensorFlow and PyTorch [145]. These optimisers are outside the scope of this chapter, but would be a worthy investigation in future work.

Number of Layers	Layers Density	Layer Activation Functions
3	128 x 64 x 32	Tanh, ReLU, ReLU
3	256 x 128 x 64	Tanh, ReLU, ReLU
3	128 x 64 x 32	Tanh, Tanh, Tanh
5	512 x 256 x 128 x 64 x 32	Tanh, Tanh, Tanh, Tanh, Tanh

Table 6-2: TensorFlow Machine Learning Model Configurations

The tested model structures vary by changing the density of the TensorFlow model layers including 128x64x32x1 and 256x128x64x1 nodes per layer, respectively. Both cases also use the tanh activation in the first layer, followed by ReLU in the two following layers. These model architectures were used to ensure that their depth would allow for strong reflection of the underlying trends in the dataset, but also are not too thick so that training is relatively fast due to the reduced number of variables that the model is trained on.

As this model is a regression model, the performance and accuracy is gauged using the measure Mean Square Error. This measures the average of the squared difference between the predicted and actual values. The formula for MSE is shown in Equation 6.5, where y_i is the target value (taken as the value in the test dataset), $\hat{y}_i$ is the predicted value and n is the number of samples. The mean squared error is used because it amplifies any large errors in the model and penalises them significantly

more, thus allowing the model to minimise large mistakes in the prediction [146]. The primary drawback to this approach is the sensitivity to outliers, which can end up affecting the model performance negatively. This was avoided by filtering the data to only include points that were reported as converged in ANSYS Fluent, based on all residuals being less than 0. If it is not possible to easily filter outliers from the dataset, then it is better to use the Mean Absolute Error instead, namely.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad 6.5$$

6.4 MLVawt ANN Specification and Validation

The model was trained on 2000 simulations using the Latin hypercube sampling with the parameters described before. An 80:20 split between the training and testing data was created, to ensure there is sufficient data to train on, but also enough test samples to ensure good adaptability of the model. This is the industry standard ratio split between training and testing data. The most performant model was found to be the thick tanh model (5 x 512 x 256 x 128 x 64 x 32 x 1) with minor dropout (5%) implemented within the model to prevent overfitting.

Good accuracy was achieved with the training model, with a very low mean square error with a value of approximately 5e-5. The training history is plotted in Figure 6-12, the smaller the loss the better the model is at modelling the trends in the provided dataset. The spikes in the training history seen in Figure 6-12 are a result of implementing dropout within the layers, which means that in each training iteration a certain number of nodes will be deactivated, assisting in reducing the likelihood of overfitting, but causing temporary spikes as there is a single model .

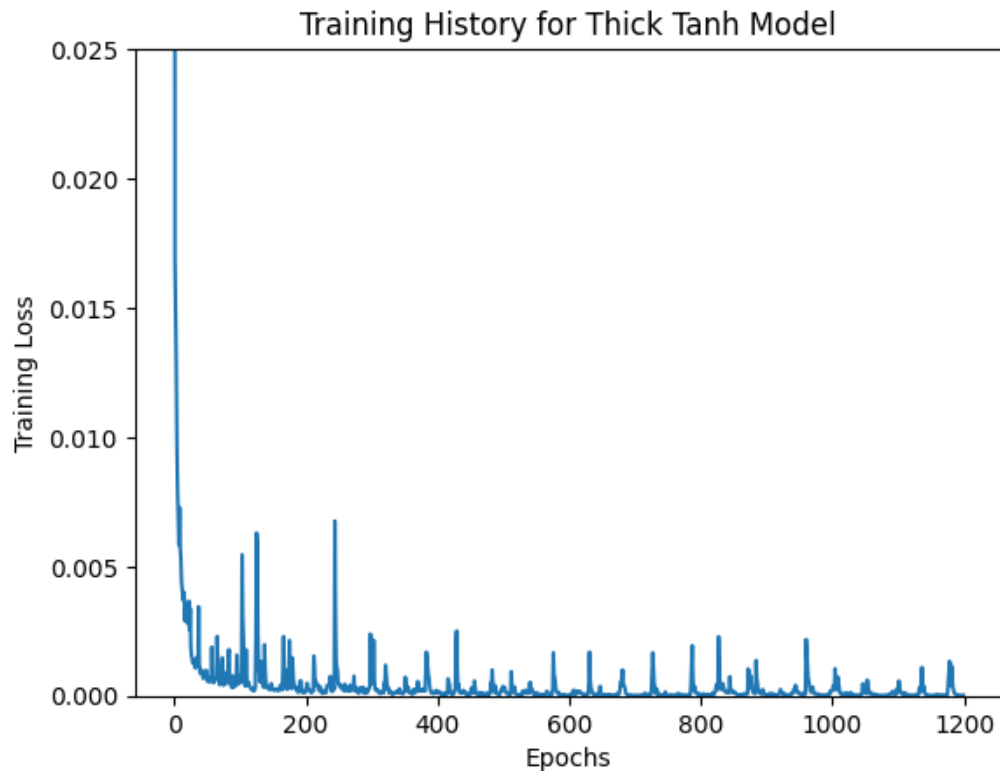


Figure 6-12: Training history for the 5 layered Tanh Activation Model

This provides much confidence that the predictions that the model makes are reliable and can be trusted with little reservation. A comparison between the predicted and actual values of the torque for the same inputs allows us to determine the reliability with a great deal of certainty. This is evaluated by entering the input parameters for the validation dataset into the ML model, and then comparing the predicted value to the actual corresponding tangential force for those flow conditions. In an ideal case, the predicted values from the machine learning model would match perfectly with the tested values, that is for a given input, the prediction is identical to the corresponding testing data. This relationship is visualised as a line with a gradient of 1 on a plot of the predicted vs true values as is seen in Figure 6-13. The closer the grouping to the line of $y = x$, the closer to a perfect fit our machine learning model is. A perfect fit would imply that for each given datapoint in the testing set, the predicted value would be exactly the same as the actual value for that datapoint.

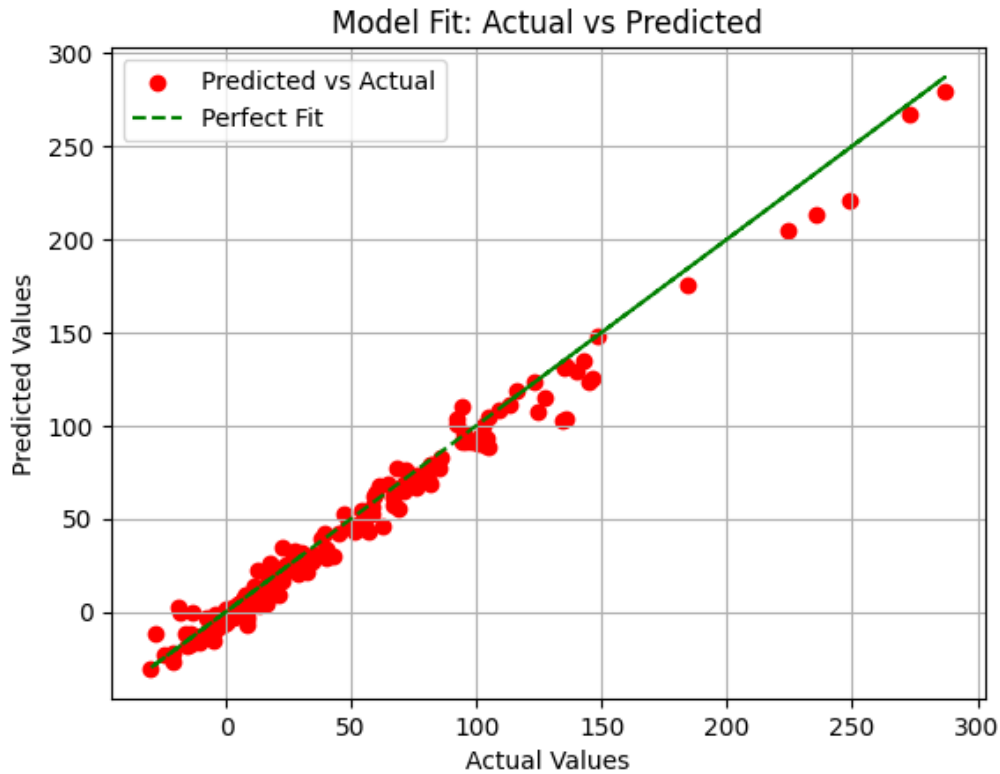


Figure 6-13: Regression fitting accuracy of the CFD based Machine Learning Model

To verify the validity of this model in the scope of VAWT blade torque prediction, it is important to firstly quantify that the machine learning models can predict the aerodynamic trends such as azimuthal angle and tip speed ratio, that can be observed within Vertical Axis Wind Turbines, before utilising the models for optimisation. The models are therefore tested for a standard NACA0015 VAWT that has been used throughout the previous chapters, operating at a 7m/s wind speed, $TSR = 2.29$, turbine radius = 0.85m and chord length= 0.225m

Figure 6-10 shows the torque distribution plot of the VAWT when predicted using NeuralFoil. The first point of interest is that NeuralFoil appears to have similar limitations with the lift and drag at higher angles of attack as was found in XFOIL. This is reasonable however, just as a child learning from a parent, it will learn the same traits and trends seen in the tool that it was trained on, and mimic them. In this case, XFOIL struggles to predict the lift and drag coefficients when the aerofoil is in deep stall due to the limitations with XFOIL, which is evident in the plot of the azimuthal angle vs torque for a NACA0015 as shown in Figure 6-14. This figure is generated by using the flow conditions for the theoretical angle of attack (described

in Section 1.3.3 and using NeuralFoil to predict the torque of the blade profile at every azimuthal angle.

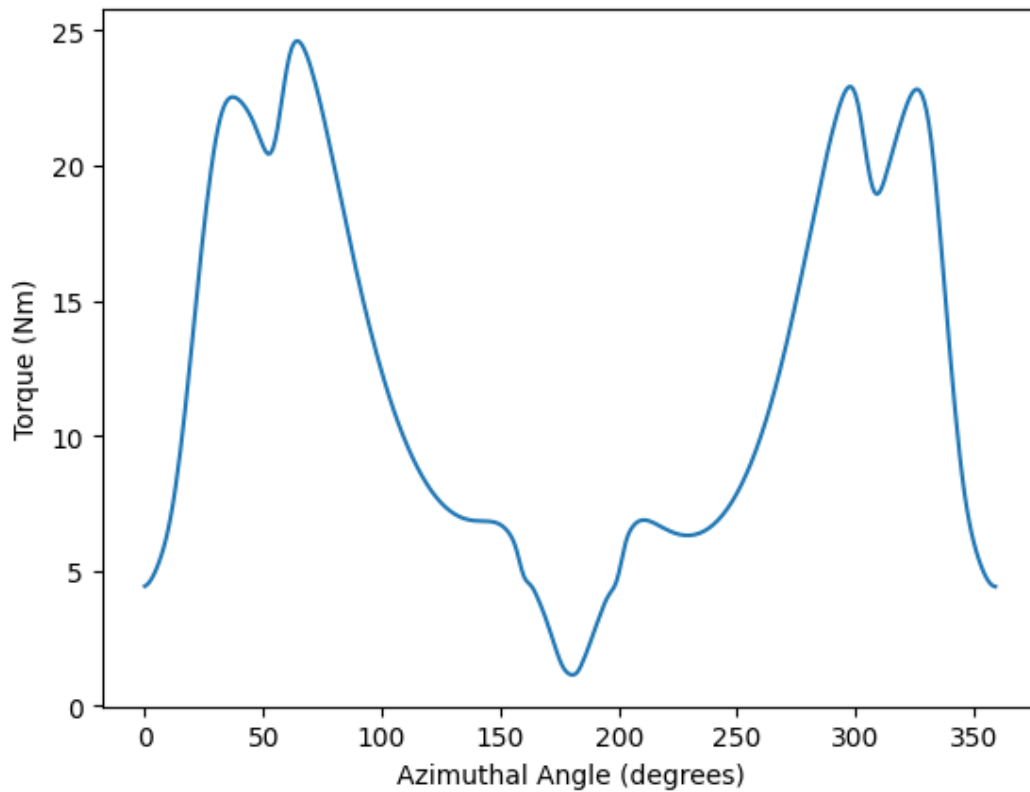


Figure 6-14: NeuralFoil prediction of the torque generation of the NACA0015 VAWT blade across the turbine rotation

Figure 6-15 shows the distribution of torque for the VAWT when using the different ANN models configurations that were tested. The ML models developed from the CFD dataset have a slightly different issue, where they model the overall trend better such as the sudden drop in the torque at the higher angles of attack which was seen in the rotating turbine CFD. However, the azimuthal angle that correlates with the peak torque appears to be earlier than that which is seen in the rotating turbine study in Chapter 4.

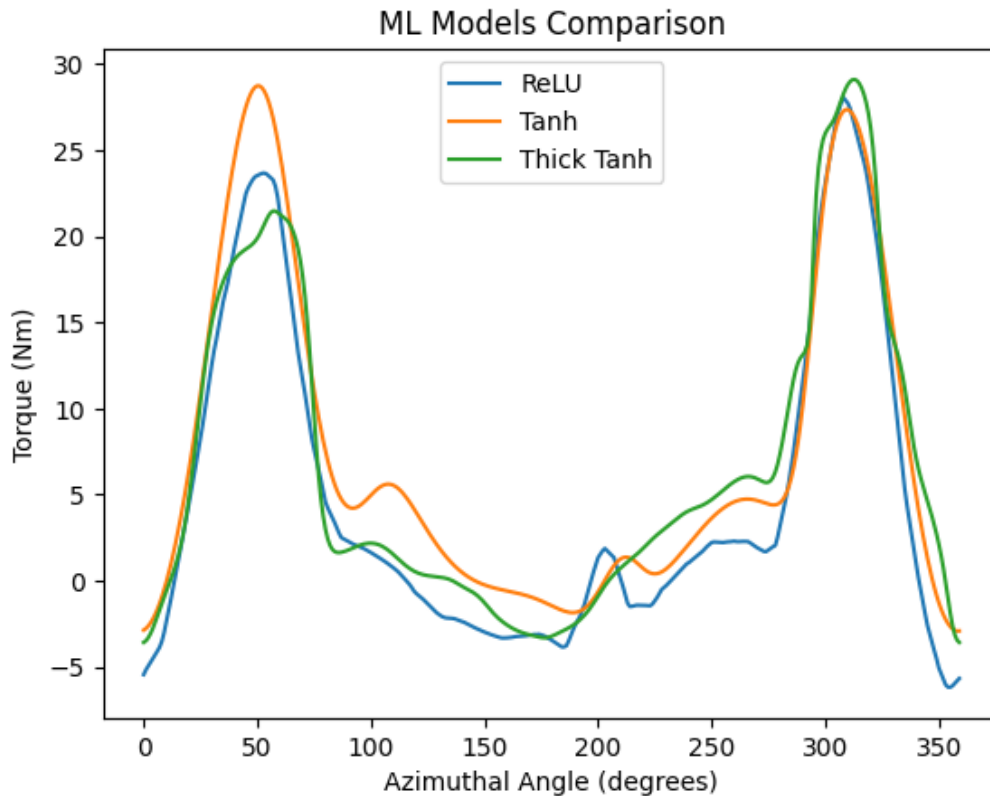


Figure 6-15: Prediction of NACA0015 Aerofoil for the design turbine ($TSR = 2.29$, Windspeed = 7m/s)

This is likely a result of the static training data not capturing the transient effects that are apparent in a rotating VAWT such as the leading-edge separation bubble that slowly grows before suddenly bursting and this reduces the lift. The rotating turbine is likely to have some inertial effects that “grabs” this separation bubble, resulting in the bursting at a later point in the rotation. This early peak could also be a result of insufficient data that causes poor fitting. This emphasises the importance of a large training dataset for complex problems including predicting the aerodynamic efficiency of a vertical axis wind turbine blade. One of the largest issues with a small dataset is the risk of overfitting the data or capturing noise in the dataset, which in turn would result in poor generalization of the model when being exposed to unseen data (data which the model hasn’t been trained or tested on) [147]. Upon reflection of the work, it was discovered that the generated dataset did not utilise the full LHS based sample domain, rather only the first $2/5^{\text{th}}$ of the dataset were simulated and compiled for training the MLVawt ANN, and is likely the root cause of the asymmetry that is seen Figure 6-15 between the $0 \rightarrow 180^\circ$ and $180 \rightarrow 360^\circ$ sections. Figure 6-16 shows

the distribution of the intended dataset and the true dataset, with the intended dataset shown in blue and the generated dataset in red. These plots were generated by plotting each variable against one of the other input variables. The flat histograms of the intended dataset demonstrate that they would have been an even distribution of every variable, seen by the moderately flat histogram distribution, or the almost solid blue square appearance of the scatter plots. The training dataset used (in red) is clearly not as even or well distributed, with a clear bias in camber for high camber aerofoils, and more focus on lower azimuthal angles or TSR values. This is reflected in the scatter plots, which show gaps in the distribution, which indicate a missing comparison point. These biases would be reflected in the trained model, and as a result it is expected that the model would be unable to model the negative camber or low windspeed combinations as effectively, which is seen by the uneven torque distributions of Figure 6-15. A well designed ANN which is trained on a broad and evenly distributed dataset would almost perfectly reflect the theoretical turbine torque profile (described in Section 1.3.3), as it would not have any gaps in its underlying knowledge base (the training data) and due to it being trained on static aerofoil data, transient effects would not be present.

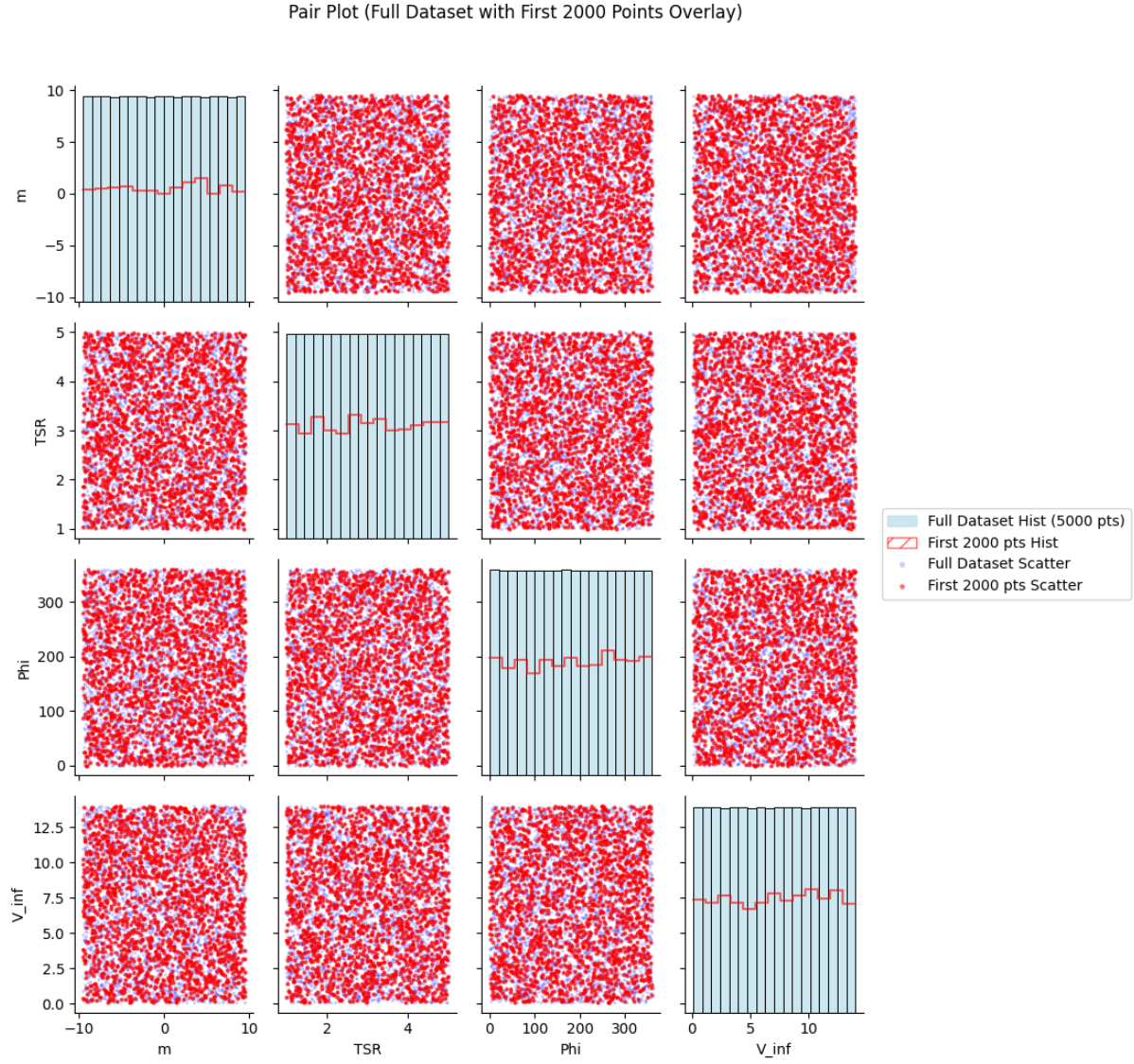


Figure 6-16: Pair plot of the dataset distribution of the intended dataset in blue, and simulated dataset in red.

6.4.1 Sample Distribution Comparison

A series of ANNs were created using the same model formulation, but were trained on different size datasets. The underlying domain of the dataset is identical; $1 \leq \text{TSR} \leq 5$, $-10 \leq m \leq 10$ %c, $1 \leq V \leq 14$ m/s and $0 \leq \phi \leq 360$, but the number of points used to create the dataset are different. The tested dataset size was 10^4 samples, to understand if utilising the larger training samples is beneficial to the overall model performance, and where the balance may lie between computational expense and model accuracy. A dataset containing 10000 points was used as an edge case, with the logic that if it were to

significantly improve model performance, then it would be worth investigating the point of diminishing return by balancing the model accuracy and the time taken to generate the model including the dataset. If there is no improvement over the 2000 point model, then clearly 2000 is already sufficient for our use case.

The dataset had models created using the keras tuner python module, which is a module which automates tuning of the hyperparameters of the ANN model automatically, to see if it is a valid technique for creating an ANN without the typical expertise and in-depth understanding of machine learning architecture. The model was tested with 3 variations; the thick tanh model architecture and two automated keras tuner models; one with the full dataset (uncleaned), and one with the data set cleaned by removing any outliers that were outside 1.5 times the interquartile range of the dataset. The resultant NACA0015 tangential force estimation is seen in Figure 6-17.

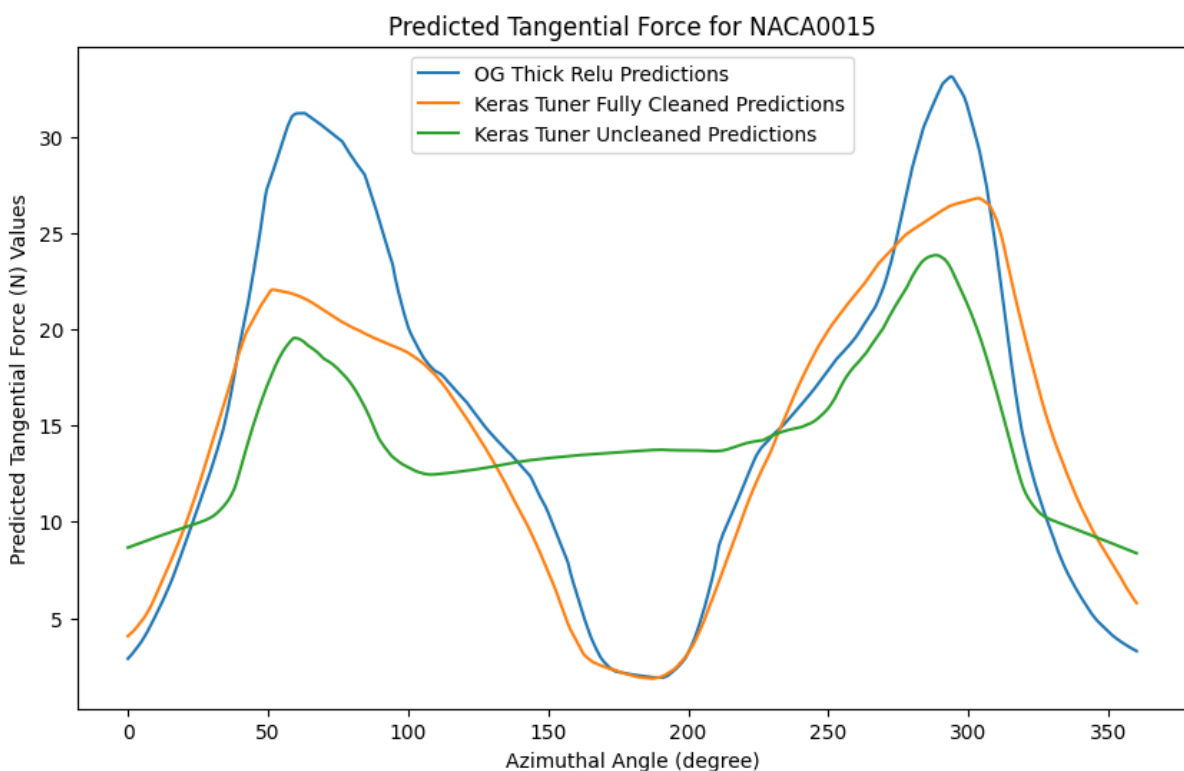


Figure 6-17: Predicted NACA0015 Tangential Force for a 10000 sample dataset

Clearly cleaning the dataset was fundamentally crucial in capturing the overall trend in the tangential force (two symmetrical distinct curves are expected), but the results were not much better than the initial partial dataset model, and due to the

reduced computational cost of the partial dataset, that model is taken forwards for comparison and utilisation in the GA optimisation process.

6.5 Genetic Algorithm Optimisation of the Camber Profile for VAWT Blade

The genetic algorithm logic flow is in the same format as that established in Chapter 1 and 2, but due to the reduced limitations on the torque estimation for a specific genotype, it is possible to have more extensive searching parameters. Taking the crossover as an example, previously it was found that only having 5 genotypes per generation was a good trade-off between accuracy and speed for the optimisation procedure, but this is unnecessary when it takes only a few seconds to calculate the torque value, and it has an unnoticeable difference in the performance if each generation consists of 20 genotypes.

With the new trained model, we can replace the previous data generation methods such as complex UDFs, or automation scripts for XFOIL and ANSYS Fluent, with a simple one-line tool to predict a value of torque from the new model. The resulting workflow is as follows:

- i. Generate the initial camber population
- ii. For each genotype, evaluate the desired property, in this case it is the blade driving the moment or torque by inputting the parameters into the ANN model.
- iii. Sort the genotype by best to worst from their respective C_t values.
- iv. Take the two best genotypes from the generation and pass to the next generation.
- v. Apply crossover functions and mutations to generate the next generation of cambers.
- vi. Repeat steps 2 -> 5 until sufficient convergence in the C_t value, or until a desired number of generations has been completed.

Taking the NeuralFoil tool as the lift and drag polar generator, one can make an extensively complex and intricate optimised camber deformation profile. One edge case example tested was to optimise 360 positions (every degree of azimuthal angle) with 21 genotypes per generation, which only took 22 minutes on a standard desktop

computer! Previously the XFOIL studies would take around 30 minutes for 11 genotypes and 10 optimisation positions, every 36 degrees of rotation. At first this was assumed to be a parallelism speed up, as XFOIL is a single threaded application, but NeuralFoil is also single threaded, so the speedup is simply due to the computational efficiency of the ANN. Similar speed up is seen in this new machine learning model, with the full 21 genotype, 360 optimised positions case taking 36 minutes.

Using the genetic algorithm workflow, the optimal camber profile is generated for the standard VAWT operating conditions in the same way as in the previous chapter. The constraints are labelled in Table 6-3.

Windspeed (m/s)	Tip Speed Ratio	Turbine Radius (m)	Chord Length (m)	Position of Max Camber (%c)	Blade Thickness (%c)	Setting Angle (degrees)
7	2.29	0.85	0.225	50	15	0
7	1.5	0.85	0.225	50	15	0
7	3	0.85	0.225	50	15	0

Table 6-3: VAWT Operating Conditions and VAWT Blade Parameters Tested.

The optimised camber profiles for both ML predictors have been compared between their estimation of the VAWT torque profile for a symmetric NACA0015 and the optimised camber profile design to ensure fair baseline comparisons. These comparisons are shown in Figure 6-18 for NeuralFoil and Figure 6-13 for the new machine learning model. Of course, neither of these are perfectly accurate at matching to the CFD studies of a rotating VAWT but this is expected due to the fact that they are a prediction which acts as a best guess. Although the training loss tends to a very small number ($\sim 4e-5$), this does not necessarily mean that the prediction will be accurate, as it could also indicate that the model is overfitting the training data and capturing noise as source values.

Both the NeuralFoil and the TensorFlow models can substantially improve the torque generation of the turbine at every optimised point by significant margins with a peak increase of 10 Nm for the Neural Foil results (seen in Figure 6-18) and as high as 20 Nm for the ANN Optimised Profile (seen in Figure 6-19). These figures show the results from the prediction models, where the TSR is constrained to 2.29, the windspeed is 7m/s and the deformation is based on the NACA0015 profile (NACA0015 with max camber at 50%c and the optimiser selecting the camber magnitude). It is impractical to run the GA for every azimuthal angle as it would be complex to set up in the ANSYS Fluent simulation for transient turbine evaluation, so the torque is only calculated and the profile is optimised every 15 degrees of rotation.

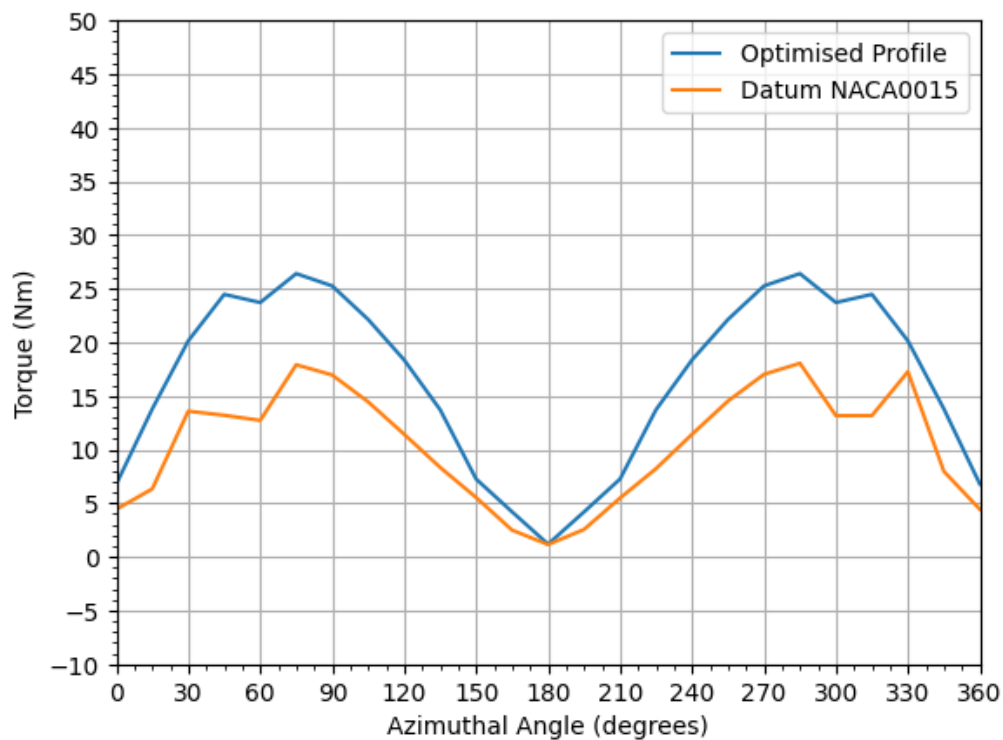


Figure 6-18: Torque generation of the NeuralFoil optimised camber deformation profile

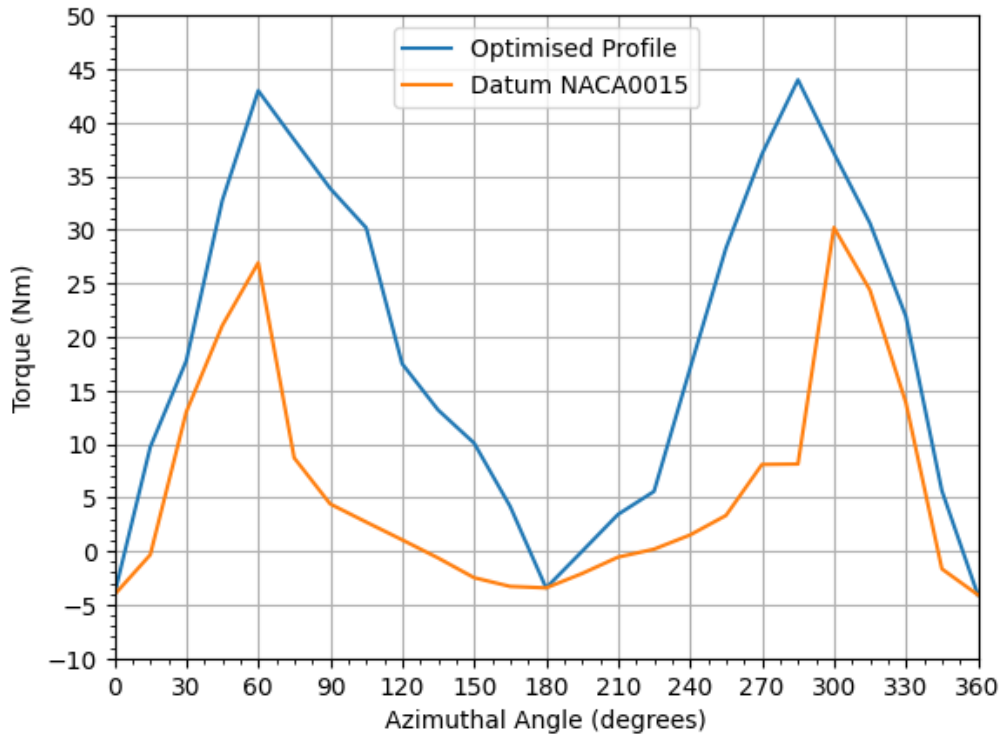


Figure 6-19: Torque generation of the MLVawt optimised camber deformation profile

The camber profiles (seen in Figure 6-20 for the NeuralFoil model and Figure 6-21 for this ANN) suggest that it may be best to increase the breadth of the camber bounds to be greater than the current limits of $\pm 9.5\%c$, as they regularly hit the bounds and are being capped, similarly to what was seen with the XFOIL optimiser in Chapter 1. This was tested at an excessive camber ($\pm 100\%c$) to determine what the result would be, but the optimised profiles were completely non-sensible. Specifically, the exact same behaviour was seen in that the camber quickly converges to the maximum value, and this resulted in being almost constantly at $100\%c$ camber, which is impossible to recreate physically and leads to untrustworthy results. This is a result of the limitations of utilising a machine learning model, and this is because it is essentially just mathematical regression at its core, and this provides the opportunity to extrapolate the data, potentially to extreme extents. It is therefore strongly recommended to stay within the trained bounds when using the model, so in the present case within $TSR = 1 \rightarrow 5$, Camber $-9.5 \rightarrow 9.5$ and Windspeed = 1-10 m/s.

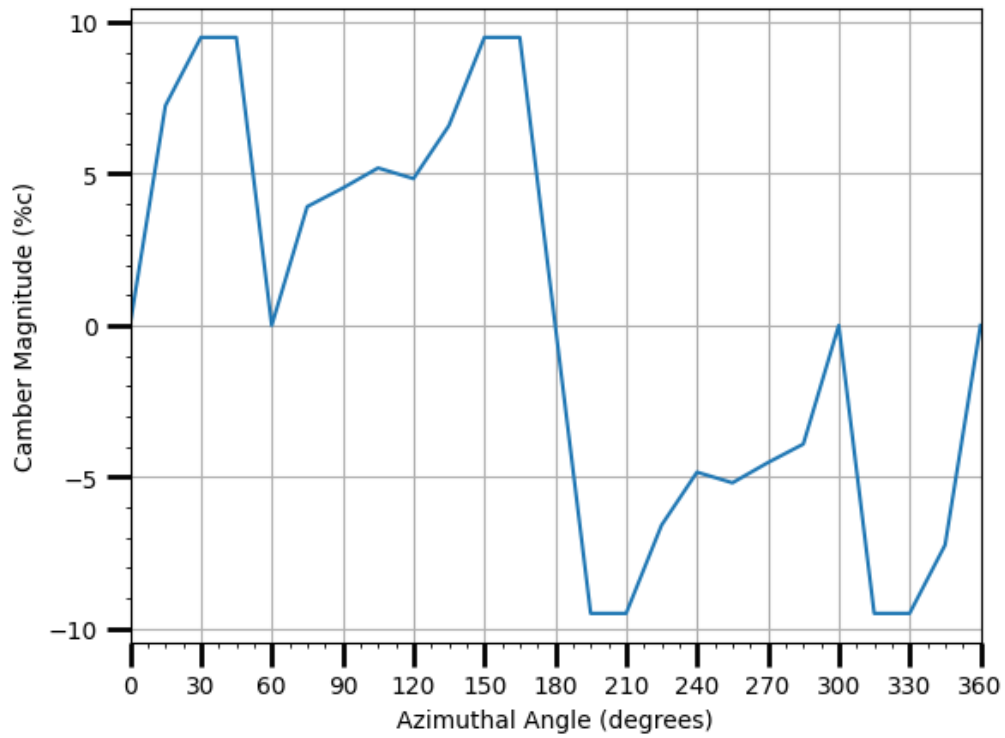


Figure 6-20: Optimised camber profile using the NeuralFoil predictor

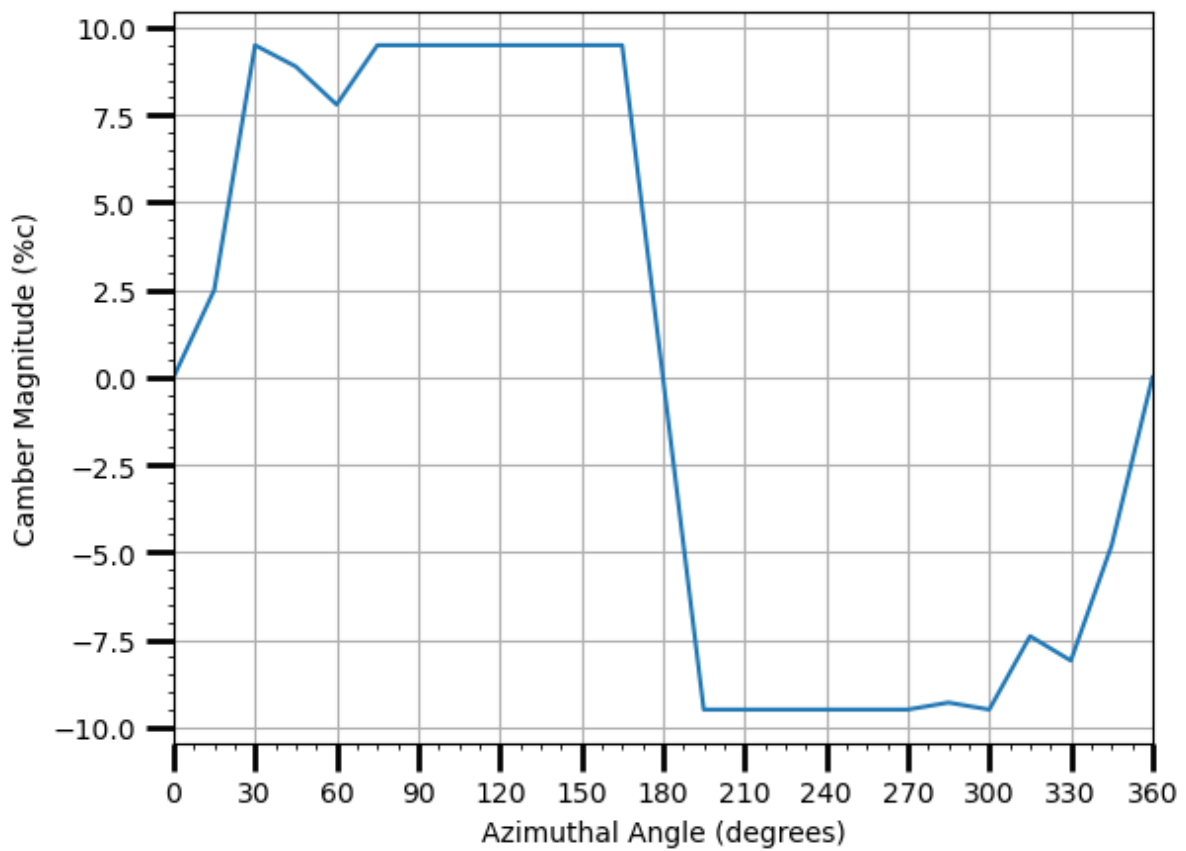


Figure 6-21: Camber profile optimised using the Tensor Flow Machine Learning Prediction

The reason for the asymmetry in the camber profile is unknown, but due to the fact that it is found in both NeuralFoil and in the present machine learning model suggests that it is likely to be a side-effect of an over-fitted model or due to the bias in the training dataset, i.e. for the testing of the training split of the dataset, maybe more of the training was on the positive angles of attack, thus allowing for a better prediction in that region, which is reflected in Figure 6-13, with the shift in the bias to the bottom left suggesting a lower camber, TSR or phi data points dominated in the overall dataset.

The optimised camber profiles were fed into the same CFD setup and the UDF was used as in **Chapter 2**, to quantify the improvement in comparison to the original symmetric NACA0015 turbine blade profile in situ. The optimised positions were added to each input parameter, which are read into the UDF when the simulation is running, allowing the camber of the aerofoil to be continuously changed each timestep. An interesting point to note is that the results from ANSYS Fluent tends to have massive fluctuations in the aerodynamic forces on the aerofoil around the optimisation points. It is unclear what causes this, as the velocity contours around the aerofoil do not show any significant differences between the problematic timesteps, so is considered a reporting error within the CFD software.

Figure 6-22 shows an example of how the moment force reports on one of the aerofoil blades changes with rotation, which is smooth and continual for the symmetric NACA0015, but choppy and sharp every $\sim 15^\circ$ of azimuthal rotation for the genetic

algorithm optimised bending profile in ANSYS Fluent.

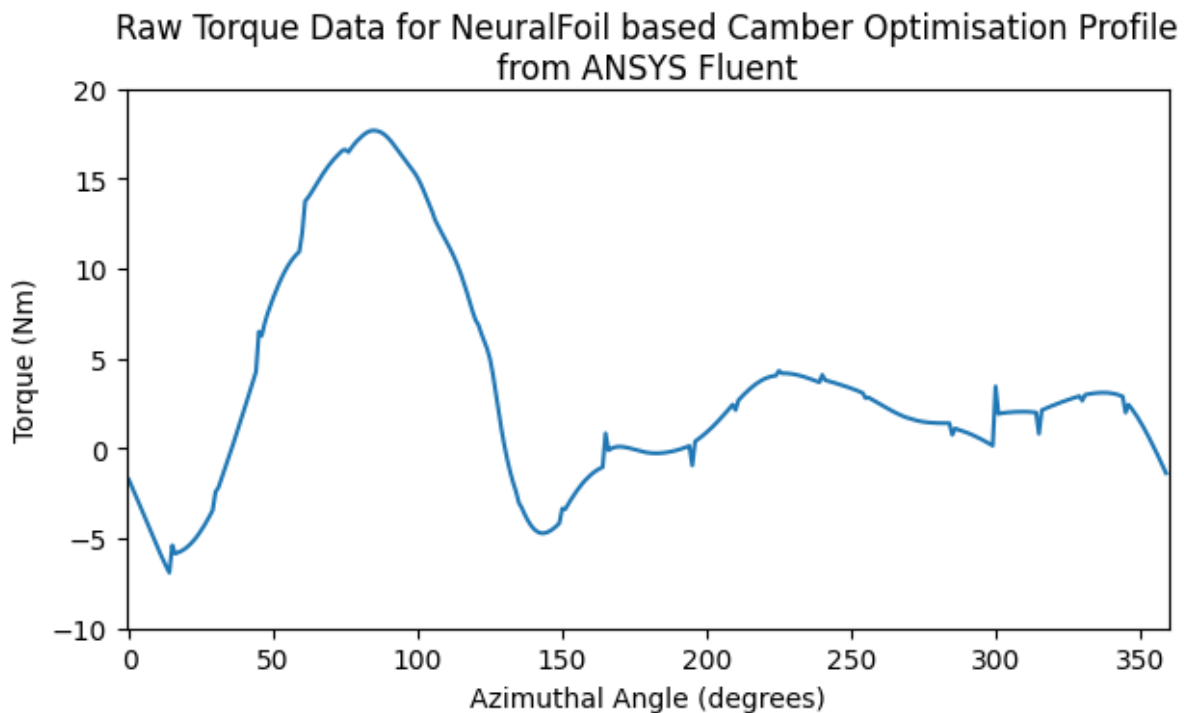


Figure 6-22: Raw Torque values of the NeuralFoil optimised GA Camber Profile from ANSYS Fluent Report Definition

It was tested if this was a result of the mesh displacement being too large per timestep, by reducing the timestep size significantly to around 10 timesteps/degree of rotation from the azimuthal position before to the azimuthal position after the optimised position. This proved to make no difference, but rapidly increased the computation time by introducing an additional 192 timesteps per turbine rotational period. As a result, it was concluded that this is likely a reporting error in ANSYS Fluent regarding the points associated with the dynamic mesh when the simulation is reloaded and restarted and can be ignored, potentially due to the way the UDF is setup but this can be investigated in future works. To circumvent this sampling issue, a 20-degree polynomial was fitted to the raw data to provide a smoother and more realistic flow profile by eliminating the outliers and large spikes that are non-sensible.

The fitted curve is mapped over the original torque data reported in ANSYS in Figure 6-23 and Figure 6-24, for the NeuralFoil based camber profile CFD results and

TensorFlow based camber profile CFD results respectively, and it is clear that there is a strong fit to the raw data.

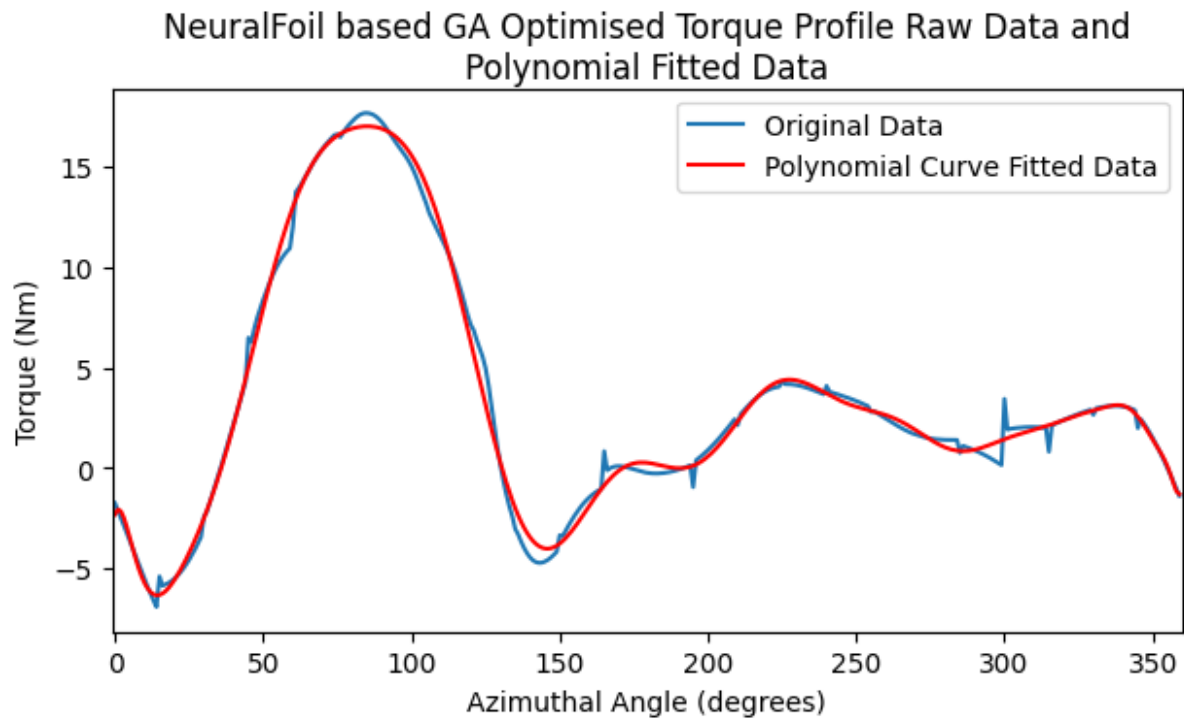


Figure 6-23 : Polynomial Curve Fitted to NeuralFoil based GA Optimised Camber Profile Torque

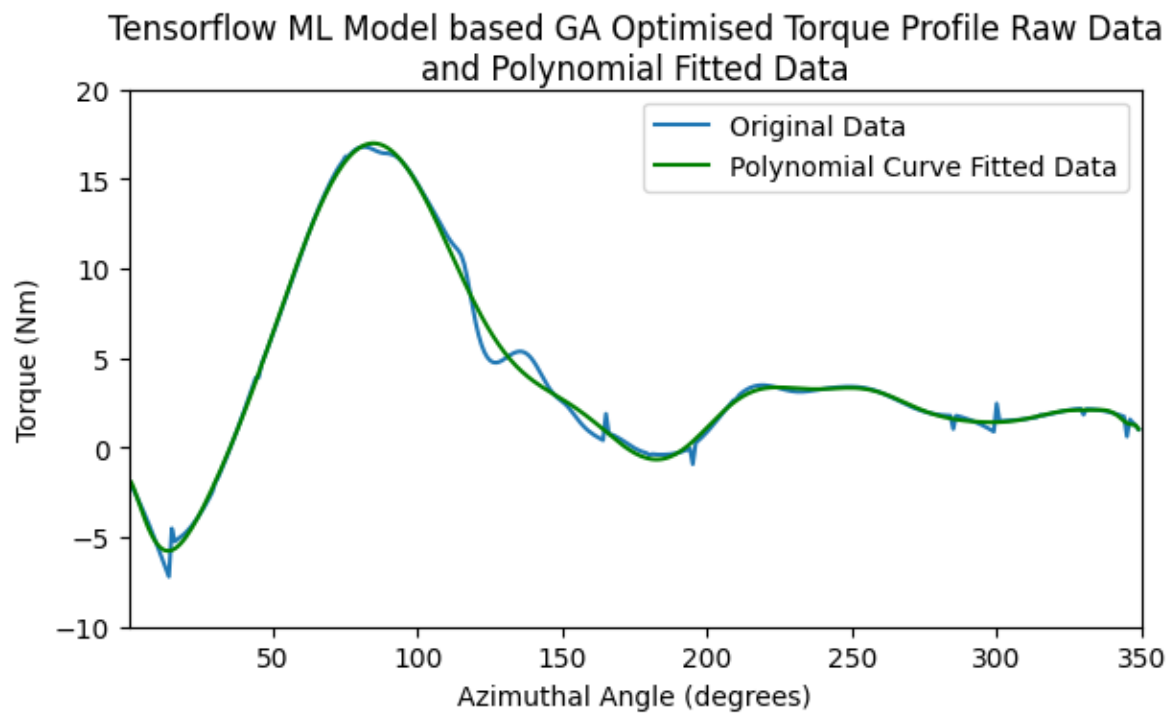


Figure 6-24: TensorFlow based GA Optimised Camber Profile Torque

The residual of the polynomial fitting is plotted in Figure 6-25 and Figure 6-26, which is calculated as the difference between the raw and predicted value at each azimuthal position. With a very small residual range of around $1e-4$, that equates to a negligible difference, but to better understand how well it matches the mean squared error the R^2 values are calculated, which are measures of the mean error and variance, respectively. A lower MSE means a closer average value, and an R^2 value closer to 1 means there is less variance, which is arguably the most important measure.

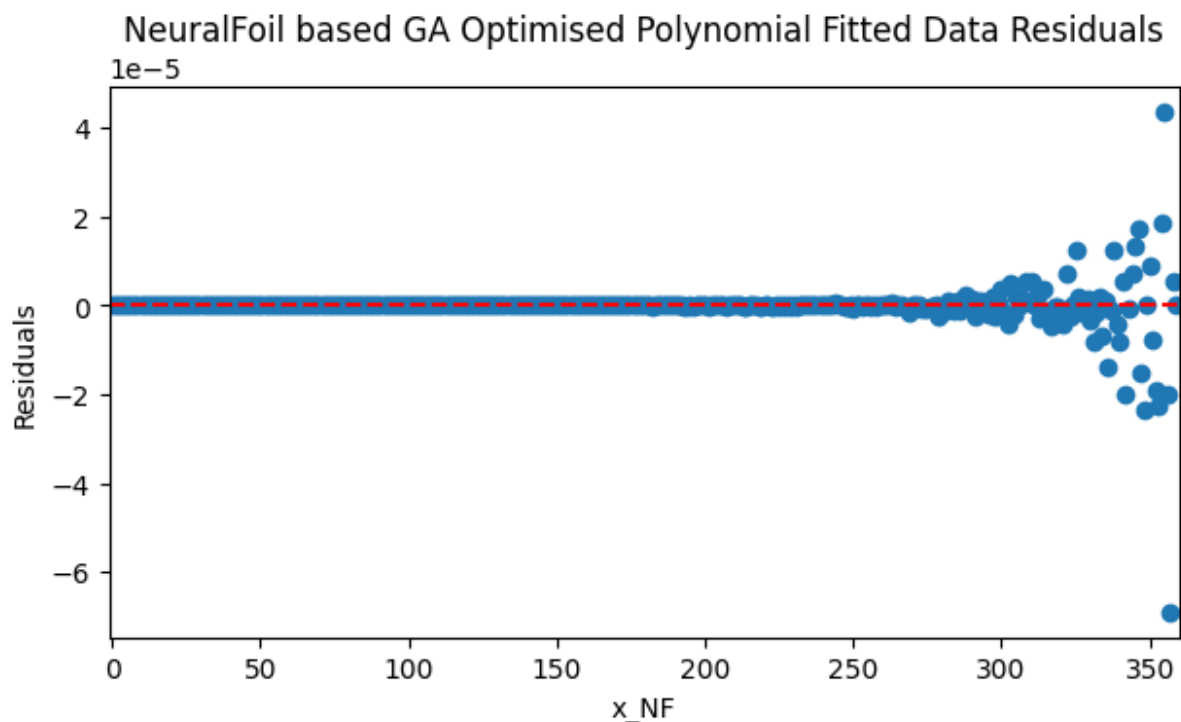


Figure 6-25: Residuals of Polynomial Fitting between the fitted and raw Torque for the NeuralFoil Optimised Camber CFD Results

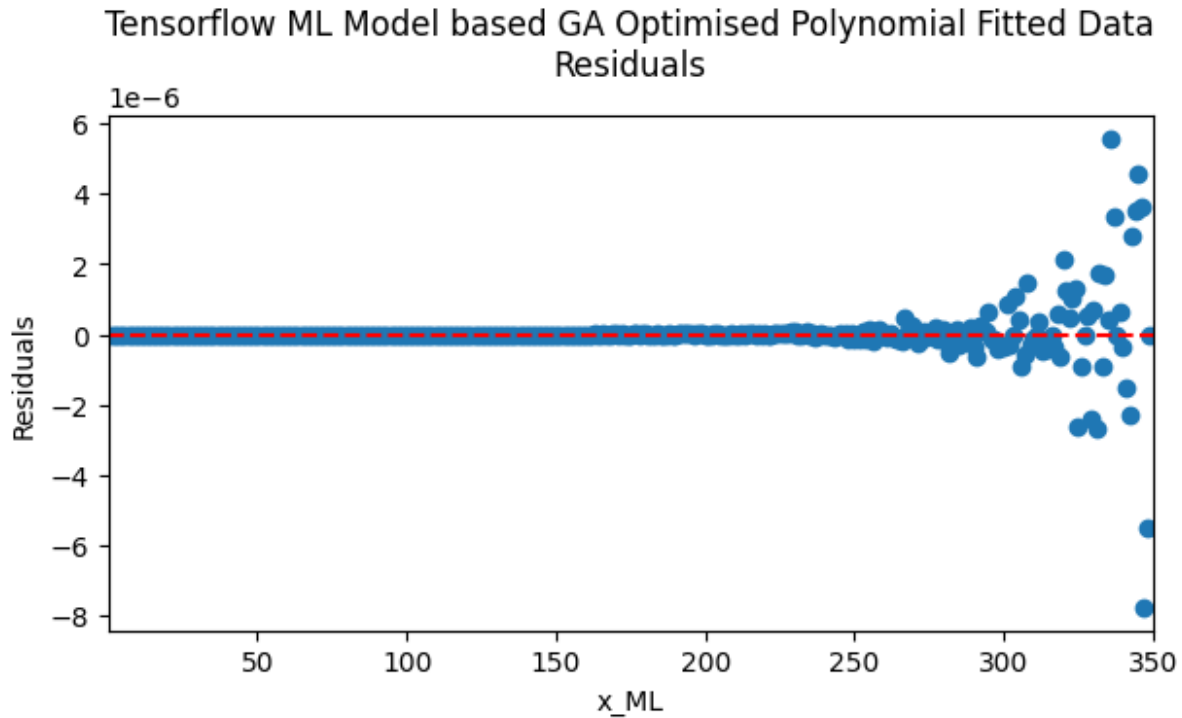


Figure 6-26: Residuals of Polynomial Fitting between the fitted and raw Torque for the TensorFlow Optimised Camber CFD Results

The Mean Squared Error and R^2 values are remarkably small (seen in Table 6-4) demonstrating the very good match, with the majority of the error present in the points near the end of rotation (see Figure 6-25 and Figure 6-26). The polynomial fitted data technique is therefore used for further comparisons to the NACA0015 blade profile performance, due to its ability to negate noise in the data, whilst capturing the data's underlying trends.

Model	Mean Squared Error	R^2 value
NeuralFoil	3.123e-11	0.999
TensorFlow	3.519e-11	0.999

Table 6-4: Polynomial Curve Fitting Statistics

Figure 6-27, Figure 6-28 show the torque distribution through the turbine's rotation for the NeuralFoil based optimised blade deformation profile and the Tensorflow optimised blade deformation profile compared to the NACA0015 static profile. Figure 6-29 shows all three datasets overlaid for easier comparison. All three figures also

have dashed horizontal lines, which display the mean torque for the respective blade profile methods.

Comparing the results between the optimised GA Profiles and the NACA0015 shows the good increase in the torque coefficient at nearly all points, but is lower in the initial period of rotation from $0 \rightarrow \sim 75^\circ$. This period of lower torque generation is a result of the blade-wake of the preceding blade of the turbine and the increase in torque generation in the downstream section causing a deceleration in the flow in the upstream half of rotation.

The rapid dip that is seen in the NACA0015 is not present in the ML Optimised and the NF Optimised blade profiles, indicating that the blade stall is significantly mitigated by optimising the camber profile, especially when using the TensorFlow ML predictor. The same trend is seen throughout the turbine rotation, from 90° through to the end of the turbine rotation, with the optimised profiles both having a better torque generation than the NACA0015.

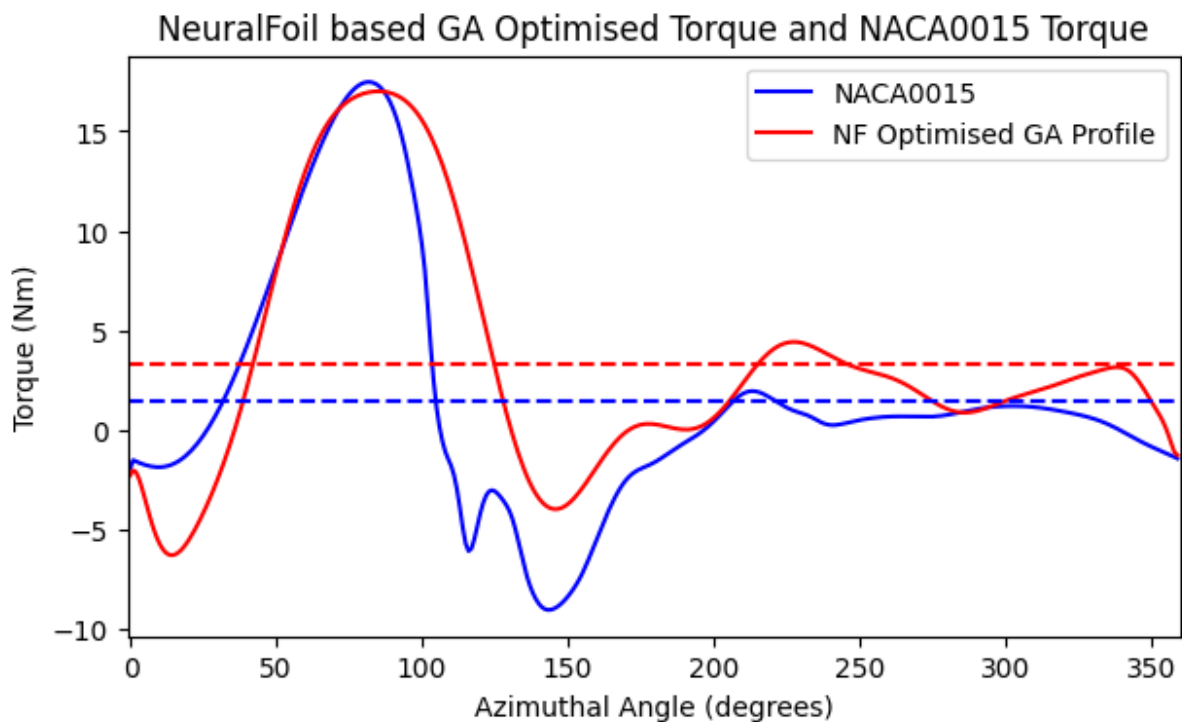


Figure 6-27: NeuralFoil Optimised Torque compared to NACA0015 turbine blade torque

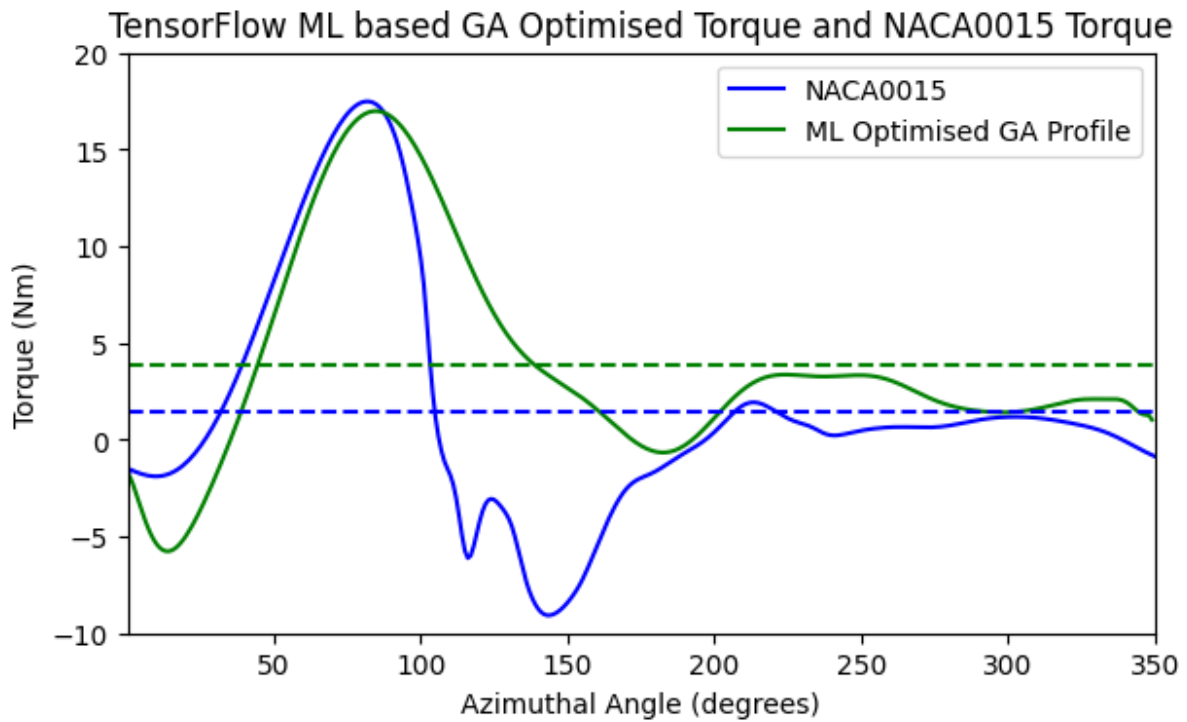


Figure 6-28: Custom TensorFlow Optimised Torque compared to NACA0015 turbine blade

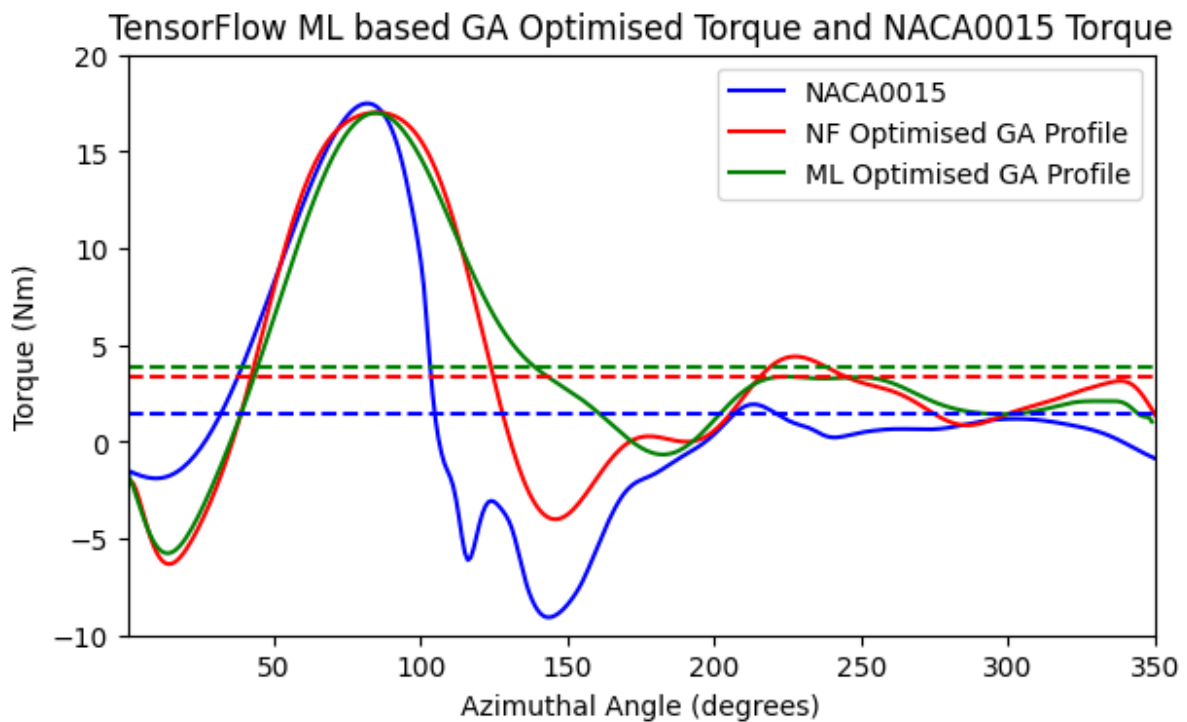


Figure 6-29: NACA0015 vs NeuralFoil Based GA vs TensorFlow Based GA Torque Profile

The CFD Contour Plots provide some context to the reasoning behind the significant difference between the optimised and NACA profile at some locations. For both

optimised camber models, the key positions of difference are at the azimuthal angles 15°, 150° and 225°, where the difference between the optimised and NACA0015 profiles are greatest.

At 15°, the optimised designs perform significantly worse than the symmetrical NACA0015, which is surprising as that was not seen in the Genetic Algorithm workflow, and was also not seen in the transient genetic algorithm of **Chapter 2**. The likely cause of this difference is due to the significant rate of change in the aerofoil shape around this optimisation position, going from -8.43%c camber to 7.473%c camber in the space of 30 degrees of rotation (from 345° to 15°). This fast deflection could result in an onset of dynamic stall, which would not be considered by the genetic algorithm, as the solution used to initialise the optimiser is based on the converged results of a standard NACA0015 turbine blade. The only way to consider this method would be to implement an iterative optimisation procedure, where the turbine profile would be optimised through one rotation, optimised again in the next rotation at the same points, and repeated until there is no change in the optimisation process. This would require development of a complex feedback loop between the CFD, machine learning and genetic algorithm and is outside the scope of this work.

Looking at the contours of the velocity taken at 15 degrees azimuth in Figure 6-30, the cambered aerofoils of the optimised designs have the stagnation point shifted slightly onto the aerofoil's pressure side, causing a loss in the lift due to the change in circulation that the aerofoil shape can generate. Furthermore, after careful inspection, it is apparent that the flow velocity magnitude upstream of the blade for the optimised profiles is lower than that of the NACA0015, seen by the darker flow near the leading edge of the aerofoil.

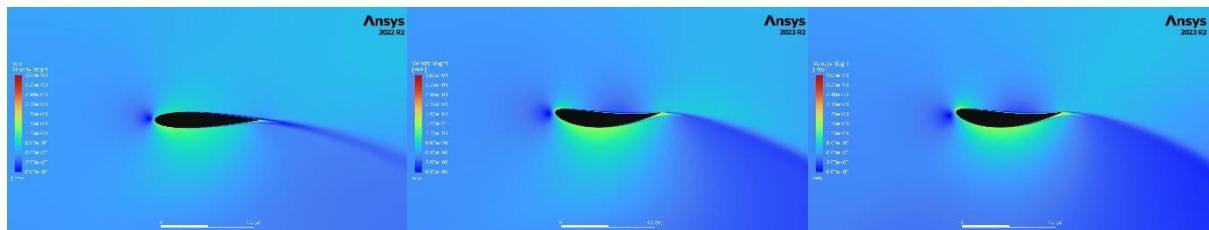


Figure 6-30: Velocity Contour at 15° Azimuth. a) NACA0015, b) NeuralFoil Based Optimisation Profile, c) TensorFlow Based Optimisation Profile

Figure 6-31 shows the velocity contours taken at 150° azimuth for the NACA0015, NeuralFoil optimisation and the TensorFlow optimisation results. The contours show a much clearer example of the flow field differences and the impact the camber profile has on the VAWTs instantaneous power generation. The large region of accelerated flow in the NACA0015 and NeuralFoil Optimised Camber is a highly rotational vortex that was shedded from the turbine blade as the Reynolds number starts to drop and the angle of attack has just passed the peak value. The reason that it is seen in the NeuralFoil and not the TensorFlow Optimised Camber

profiles is due to the differences in camber magnitude through the rotation, particularly the fact that the NeuralFoil profile snaps back to a symmetric aerofoil at 60° azimuthal angle, and then bends back to 9.5%c camber in the next optimisation stage. The local flow acceleration due to the camber increasing in this region will create an adverse pressure gradient that will cause the separation to occur earlier in the turbine's rotation. The TensorFlow profile is consistently set to a high camber with little fluctuation, so doesn't create the same local acceleration phenomena, allowing the flow to remain better attached to the aerofoil profile.

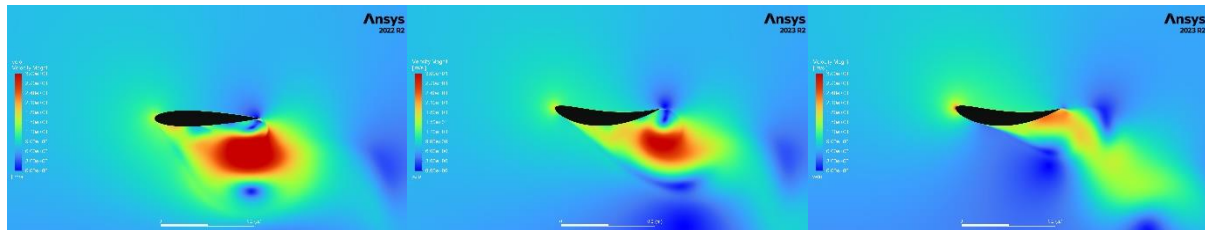


Figure 6-31: Velocity Contour at 150° Azimuth. a) NACA0015, b) NeuralFoil Based Optimisation Profile, c) TensorFlow Based Optimisation Profile

The flow profile is very similar between the NeuralFoil profile and TensorFlow profile at 225°, seen in Figure 6-32, which explains the reason for the small difference in power generation at that point in rotation. Looking in the bottom right corner of Figure 6-32, it is clear the effect of the earlier separation vortex is still influencing the flow field around the blade, which would quantify the small increase in power due to a likely increased lift from higher vorticity in the incoming flow to the blade. There is very little separation in all 3 profiles, but looking at the angle of the shear layer at the wake of the aerofoil coming from the trailing edge, there is very little circulation around the symmetric aerofoil (as the shear layer is almost parallel to the chord line), which suggests that very little lift is being produced and as such, very little torque, which in turn causes very little power generation. The much steeper angles of the optimised profiles reinforce the theory that they are producing more lift at that azimuthal angle, due to the larger circulation that is being imparted onto the flow.

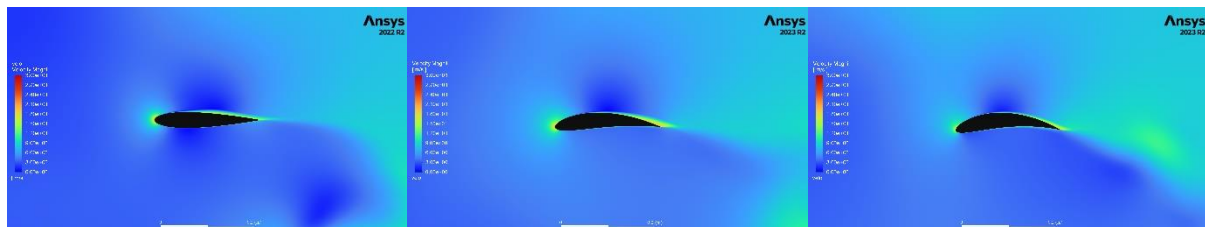


Figure 6-32: Velocity Contour at 225° Azimuth. a) NACA0015, b) NeuralFoil Based Optimisation Profile, c) TensorFlow Based Optimisation Profile

6.5.1 Testing with Alternative Turbine Conditions

To establish the validity of utilising the artificial neural network technique at various turbine parameters, it was tested at two additional tip speed ratios: one, which is low at a tip speed ratio of 1.5 and one scenario with a higher tip speed ratio of 3. The lower tip speed ratio would be the case of most significance, as this is part of the startup period of the turbine, which is one of the most problematic areas that vertical axis wind turbines face.

Looking at the higher tip speed ratio of 3 allows us to understand if the reduced peak angle of attack (which is inversely proportional to tip speed) will be utilised in maximising the camber throughout the turbine rotation. Similarly, the lower tip speed ratio case will effectively demonstrate the ability of the machine learning models to predict stall at high angles of attack and lower Reynolds numbers.

Interestingly, at the lower tip speed ratio, the average torque value is marginally lower for the NeuralFoil optimised GA profile, seen in Figure 6-33: Torque of NeuralFoil GA Optimised Blade Deformation profile TSR = 1.5 and Figure 6-34: Torque of MLVAWT ANN GA Optimised Blade Deformation profile for TSR = 1. The flow field images for low tip speed NACA0015 shows a peak in torque generation at around 200° , as the vortex that was shedded at 150° is carried downstream, only for the turbine blade to cut into it again. This is just one example of the flow physics that would not be considered in the ANN, due to the data it is trained on. The data in our case is trained on a steady-state flow over an aerofoil, so it doesn't have the level of complexity needed to model the circulating flow that the aerofoil blade travels through. Implementing this into the model would require significant complex data generation. The MLVawt model however did show significant improvement in the average tangential force coefficient, particularly in the upstream section (which is somewhat expected due to the general lack of complex flow phenomena that the blade experiences in this region) but less performant from 180 - 216° than the NACA0015 as it was encountering that vortex which sheds around 144° and is carried downstream, before recovering to perform better than the NACA0015 again.

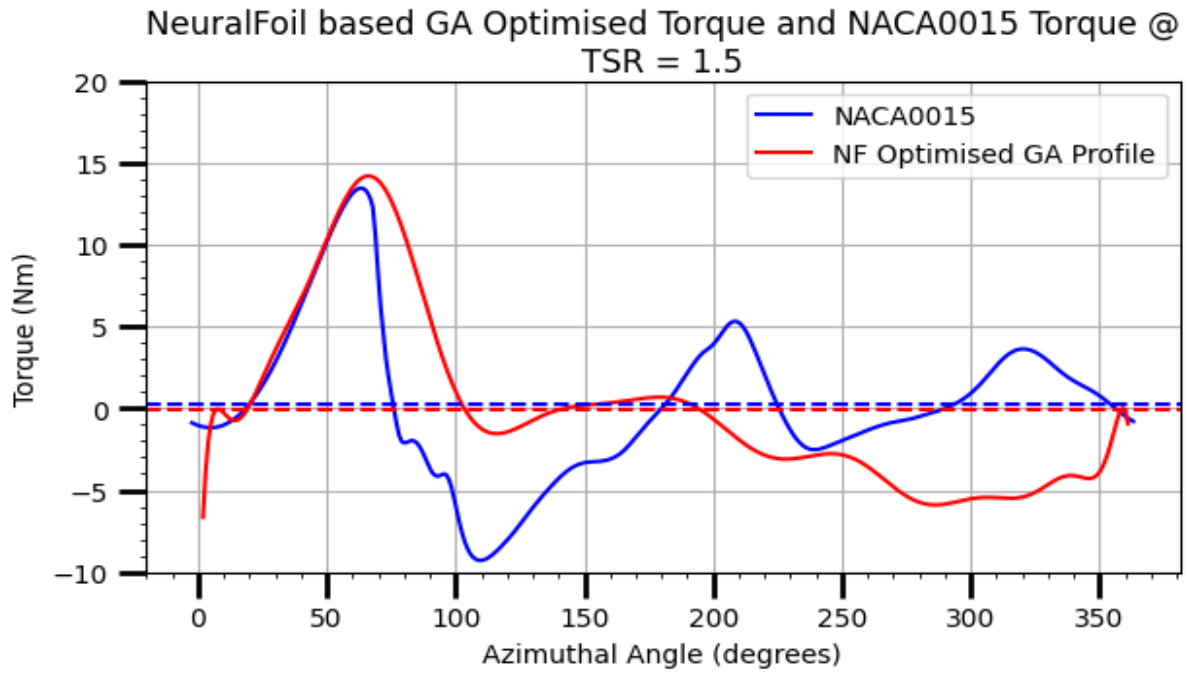


Figure 6-33: Torque of NeuralFoil GA Optimised Blade Deformation profile TSR = 1.5

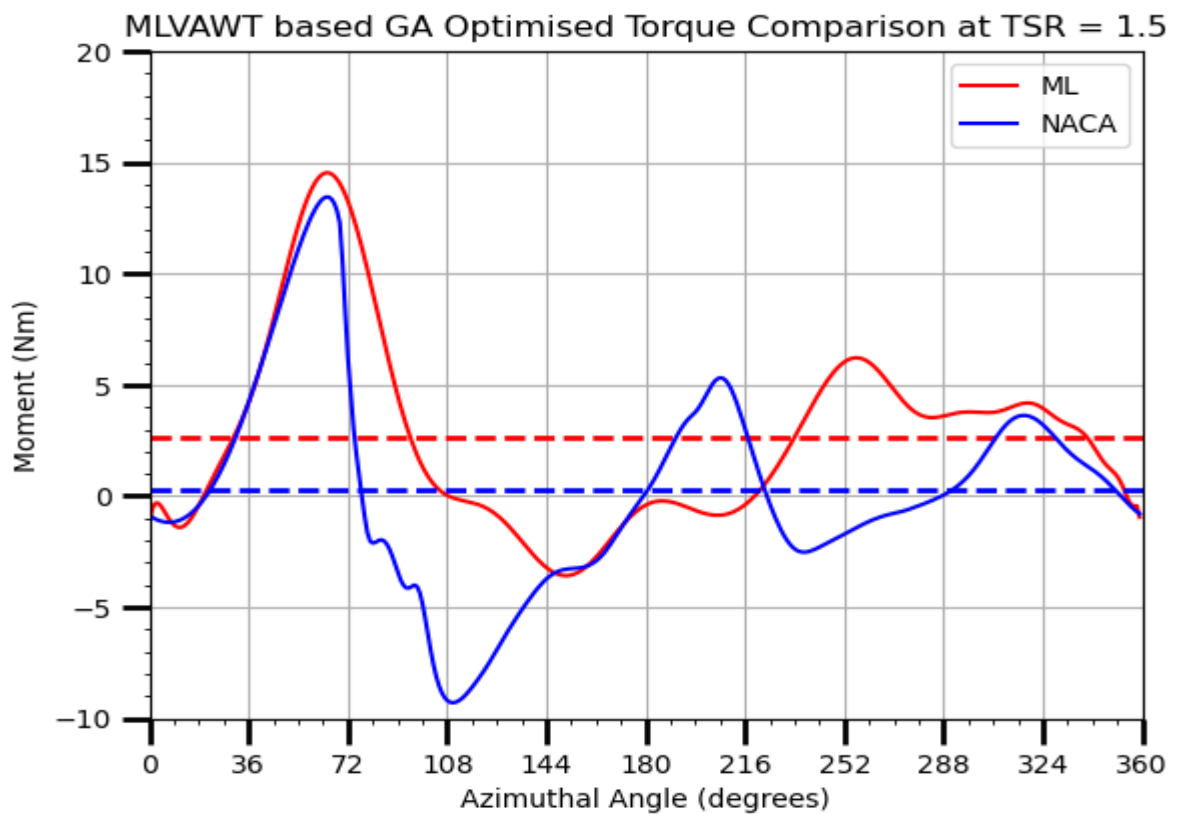


Figure 6-34: Torque of MLVAWT ANN GA Optimised Blade Deformation profile for TSR = 1.5

The higher tip speed ratio also somewhat demonstrates this susceptibility to not knowing the true flow conditions that the blade is experiencing. At the high tip speed

ratio, the upstream stroke behaves much closer to the theoretical profile for the ML model, but it stalls significantly earlier in the NF optimised model, seen in Figure 6-35. Additionally, the NF model has a negligible average blade torque, whereas the NACA0015 is substantially better, this could be a limit of NeuralFoil's training data or the inherent characteristics of XFOIL.

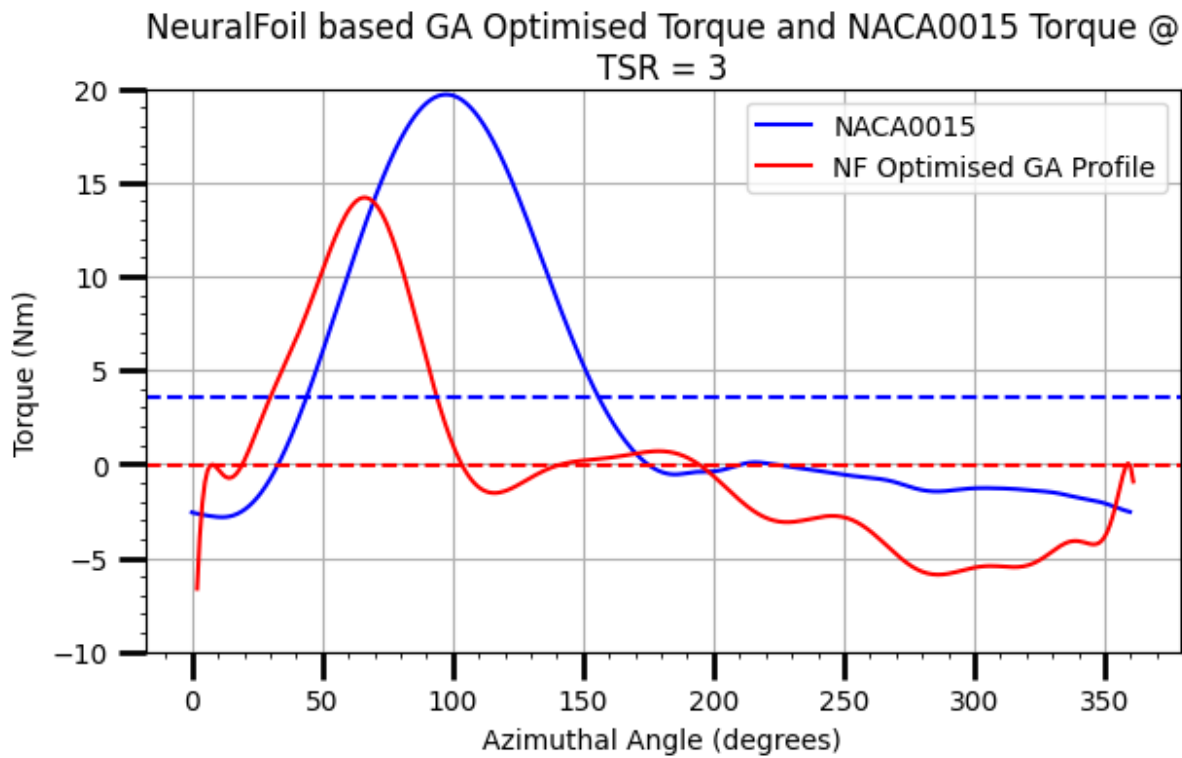


Figure 6-35 : NeuralFoil Based GA compared with NACA0015 at TSR=3

The MLVawt model has significantly better results than the NACA0015 almost throughout the turbine's rotation, as is shown in Figure 6-36. The NACA0015 is slightly more performant from $108^\circ \rightarrow 180^\circ$, but this marginal difference does not offset the significant improvements in the downstream rotation and slight improvement from $18 \rightarrow 108^\circ$ allowing the ML average moment to be better overall. The primary benefit of the ML GA model is the increased camber in the downstream section, allowing for more lift to be produced, and improving the tangential force of the blade in this region.

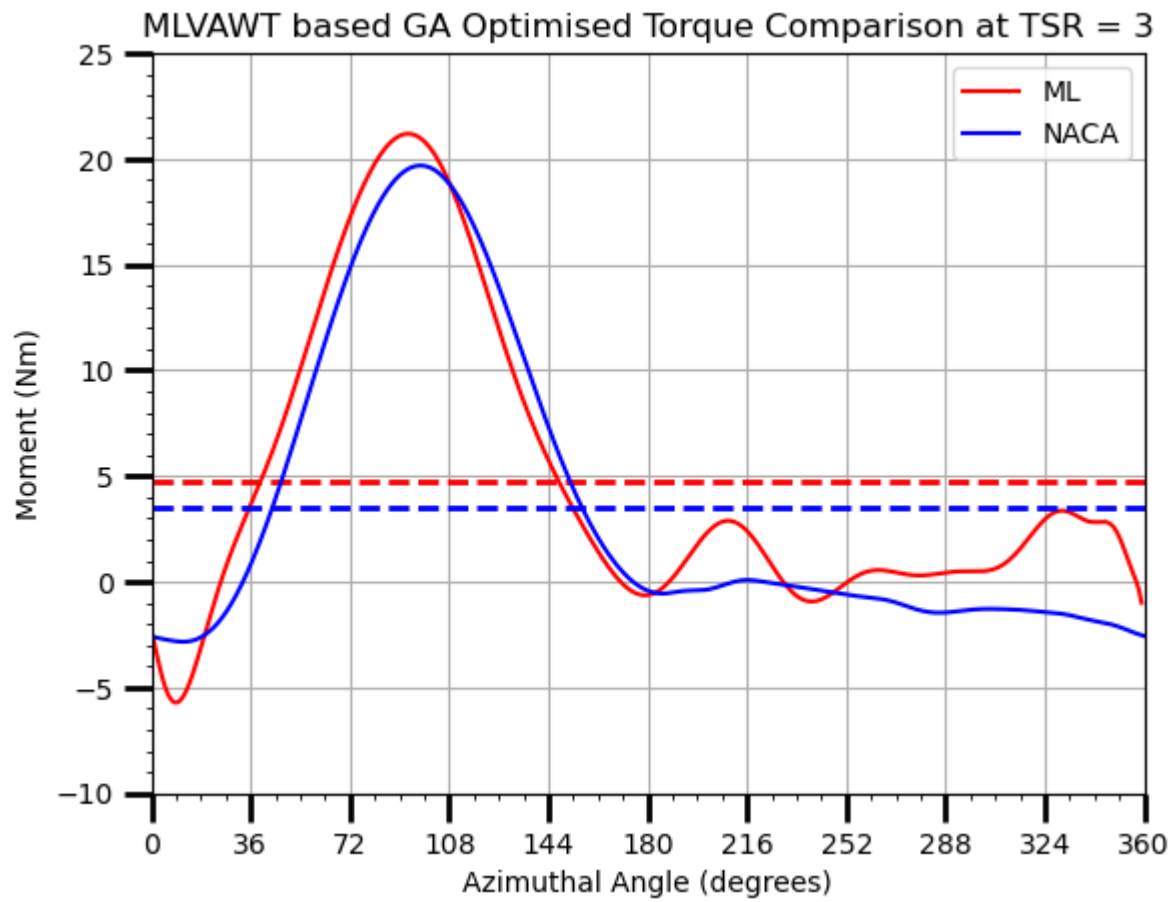


Figure 6-36 : TensorFlow Based GA compared with NACA0015 at TSR=3

6.6 Summary of Findings

This chapter leverages a genetic algorithm integrated with a machine learning model trained on a dataset derived from two-dimensional computational fluid dynamics (CFD) simulations of aerofoils, to optimise the camber profile of a transiently deforming vertical axis wind turbine blade. Whilst there have been very few studies on using optimisation to create a transiently deforming turbine blade profile, none have utilised a genetic algorithm, and none have used a machine learning model to predict the aerofoil performance characteristics. The newly created machine learning model is also compared to a well-established and trusted machine learning model called NeuralFoil for predicting aerofoil performance, which is trained on millions of XFOIL simulations.

It has been shown that using Machine Learning models can be utilised to substantially improve the time taken for the optimisation of a VAWT blade camber profile. In the case of the deforming blade camber magnitude optimisation, it was found that each genotype's torque evaluation took only around 4ms on NeuralFoil [148] and around 1ms with the new proposed machine learning regression model on a single thread of a machine equipped with an AMD Ryzen 5950X CPU. When compared to the few seconds per XFOIL run (with 2 cores) and minutes per CFD run (with 32 CPU cores), its time-savings are substantial and impressive, but one must also make sure to consider the time and computational resources required to train the initial prediction model. Even with only 4 input parameters and one output parameter, the predictions of the blade torque were made with good accuracy between the predicted and trained values, but this can be easily expanded to consider other parameters such as the aerofoil thickness as an input and separation point as an output. The average torque increased significantly for all tested TSRs when using the MLVAWT Model, but was only better at a TSR of 2.29 for the NeuralFoil optimiser, and even performed worse at the lower and higher tip speed ratios.

The machine learning model trained in this chapter was trained on the tip speed ratio, wind speed, azimuthal angle, camber magnitude and torque magnitude, to

quantify the flow conditions for a two-dimensional steady-state CFD simulation of an aerofoil. As a result, it should be relatively straightforward to create different optimisation profiles for any wind speed and tip-speed ratio combination of the designated turbine.

Future work could include training on Reynolds number instead and then simply calculating the Reynolds number that correlates with specific operating conditions, which would allow the model to be applied to any NACAXX15 based VAWT. Additionally, increasing the dataset size by using more samples and testing other positions of max camber would help increase the breadth of application further still. Implementing an additional model to predict incoming flow characteristics would further improve the accuracy of the tangential force prediction, alternatively creating a coupled CFD-in-the-loop optimisation could allow for better prediction of the actual incoming flow angle, velocity and vorticity to better replicate the flow field that is experienced by a VAWT blade, by extracting flow angles and magnitudes from the CFD simulation to feed into the ML model for each optimisation step, as a step between the work in Section 5 and this chapter.

The optimised camber profile for the standard turbine operating conditions (TSR=2.29) is found in Figure 6-21 and Figure 6-24 respectively.

7 Thesis Conclusions and Future Work

7.1 Conclusion

In summary, this thesis has demonstrated how to implement a mesh deformation for the blade of a rotating turbine simulation, has displayed the strengths and weaknesses of the different methods of generating the optimal camber values for the bending profile of the blade, and has provided source code to allow the reader to readily implement and adapt these techniques as they see fit.

An initial study was undertaken to utilise XFOIL to predict the tangential force coefficient of each candidate aerofoil profile within the genetic algorithm, which showed the strength of the genetic algorithm optimisation technique due to significant improvements in turbine torque generation, and was verified using a 2D CFD simulation of a rotating VAWT. This work also demonstrated the weakness of XFOIL when trying to calculate lift and drag coefficients at high angles of attack and/or large magnitudes of camber deflection, as it would often be unable to complete the calculation, demonstrating the limitations to utilising it for the VAWT optimisation.

The optimal Genetic Algorithm settings was found to be for the turbine consisting of a 10%c bound, with the Normalised XFOIL geometry setup and the theoretical angle of attack. The corresponding tangential force coefficient and camber profile is seen in Figure 4-5 and Figure 4-6 respectively. The lowest improvement in standard deviation of the moment is 19.9% and the lowest improvement in mean moment is 59.1%!

Building from the findings of the XFOIL based genetic algorithm, a more complex and intensive method of calculating the torque was investigated utilising a transient VAWT simulation to gather the lift and drag coefficient. This was achieved by marching the rotating VAWT simulation up to the desired optimisation position, linearly interpolating the difference between the prior optimised position and the current optimised position as it rotated, and then taking the value for torque at the

optimised position. This technique showed excellent improvement in the torque generation, particularly in the downwind half of rotation, by capturing and considering the transient effects on the turbine blade as it progressed to the optimisation position. A multitude of variations in turbine parameters such as position of maximum camber, number of optimisation positions, the size of each optimisation generation and the impact of various crossover functions were tested to develop a better understanding of their influence on the optimised camber profile's power generation performance.

The final optimal design was the 5 Phenotype, 10 Sampling points Genetic Algorithm applied with a 50% position of max camber. The corresponding moment and camber plots are found in Figure 5-9 and Figure 5-10 respectively. This resulted in the mean moment increasing from 1.425 Nm to 3.559 Nm or an increase of 150%, and the standard deviation reducing from 6.036 Nm to 4.97 Nm or a reduction by 22%!

Finally, an alternative to XFOIL is investigated, to alleviate the issues that were found in using XFOIL, but also to avoid the high computational cost of running transient genetic algorithm optimisation studies. This was achieved by utilising an Artificial Neural Network as a surrogate for the CFD procedure. A pre-established model which is publicly available named NeuralFoil, which is trained on XFOIL simulations, was investigated as one potential source for the surrogate model. Additionally, 2000 Static two-dimensional CFD simulations of a two-dimensional aerofoil were performed using flow conditions to emulate the theoretical characteristics that would be experienced by a VAWT blade based on the tip speed ratio, windspeed, azimuthal angle and camber magnitude. Both ANNs were fed into the genetic algorithm as the tool for torque estimation and showed incredible speedup in data gathering, being anywhere from fifteen times faster than XFOIL and over 1000 times faster than the transient CFD method. This technique did highlight the dependence of high-quality data to train the machine learning model, and also the fact that a well architected model will also reflect the characteristics of the training data.

The optimised camber profile for the standard turbine operating conditions (TSR=2.29) is found in Figure 6-21 and Figure 6-24 respectively. This camber profile

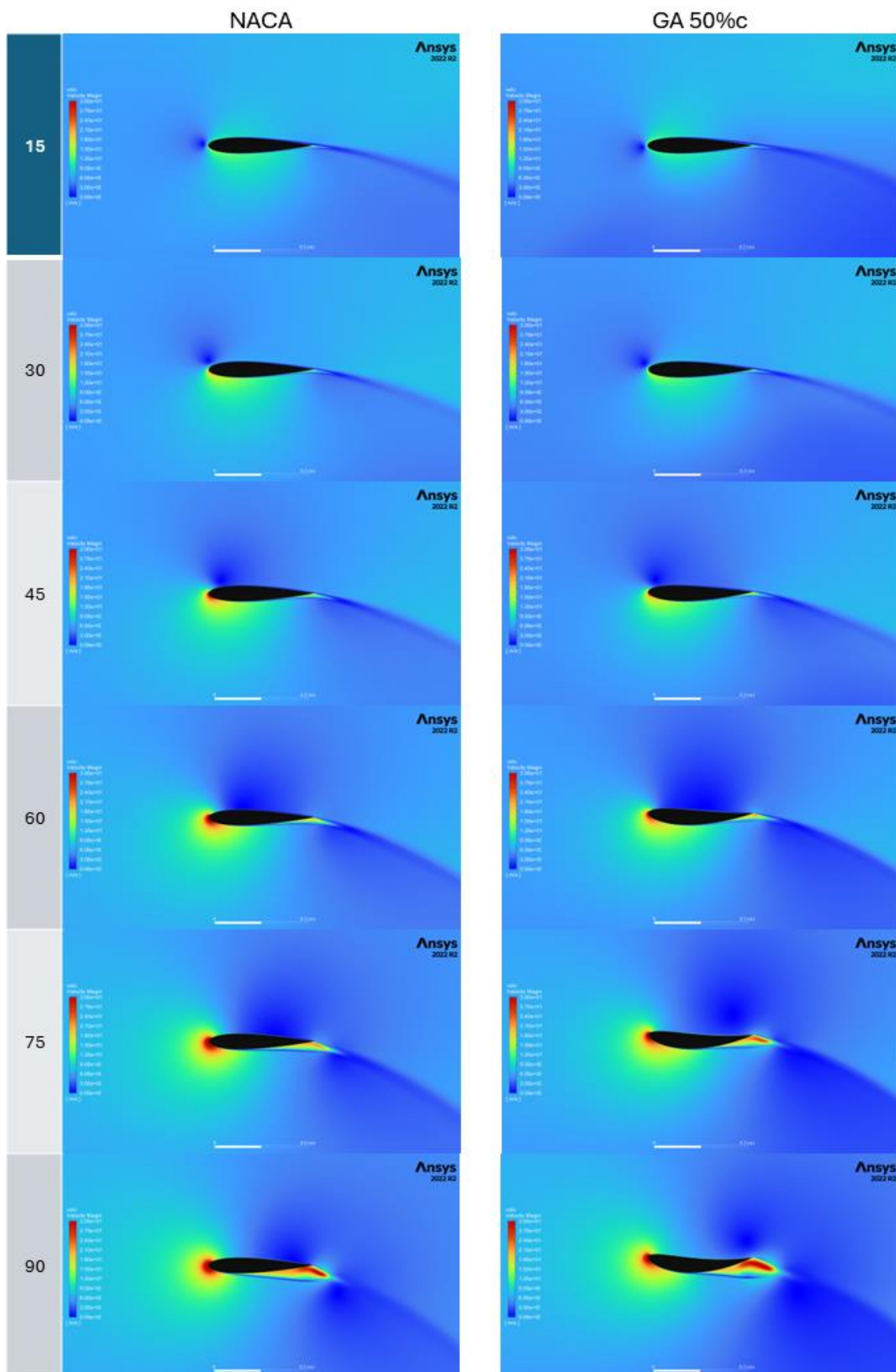
results in an increase in mean from 1.43 Nm to 3.78 Nm, an increase of 164% and a reduction in standard deviation from 6.04 Nm to 5.40 Nm, a reduction by 11.9%!

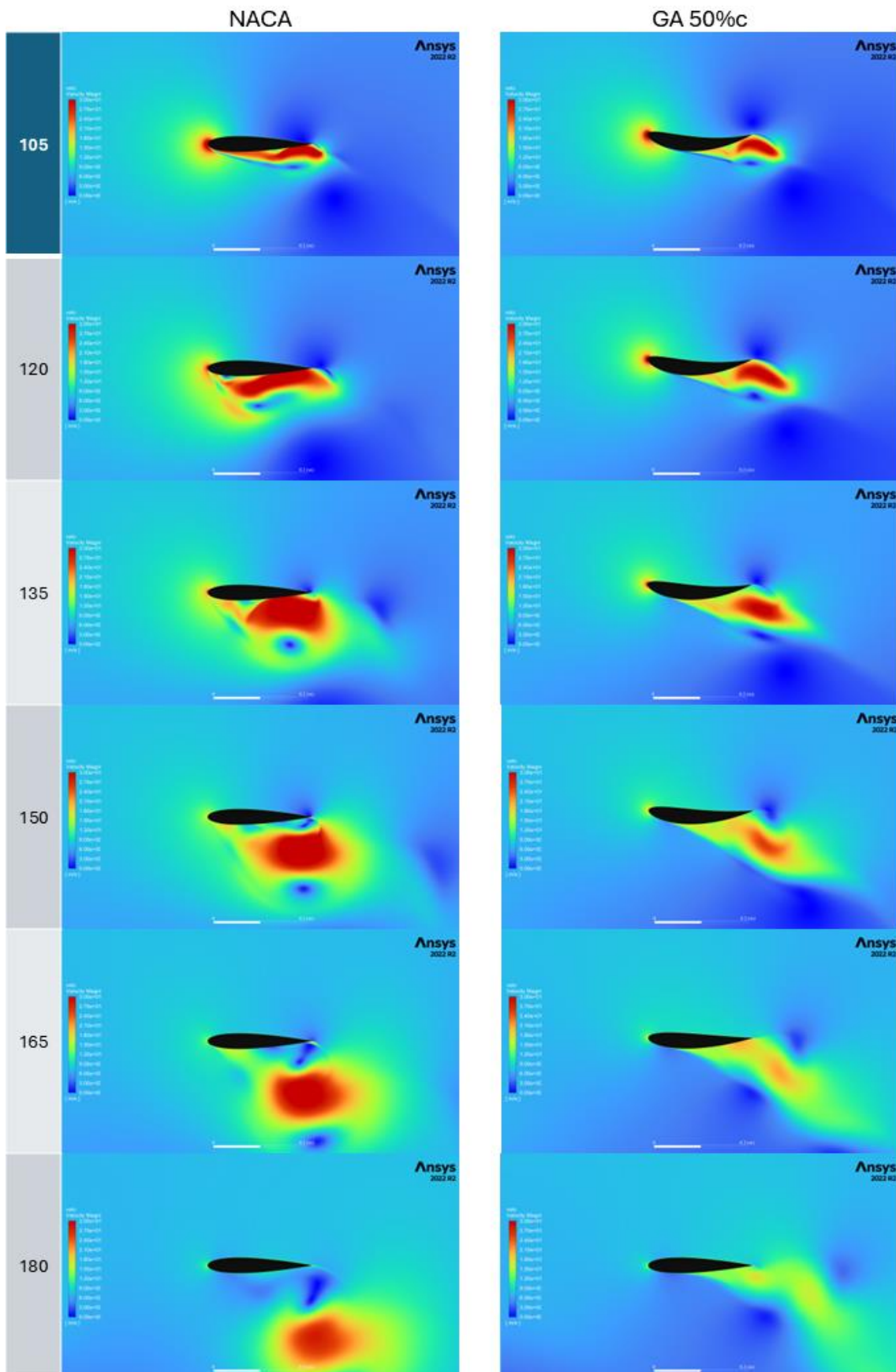
7.2 Future Work

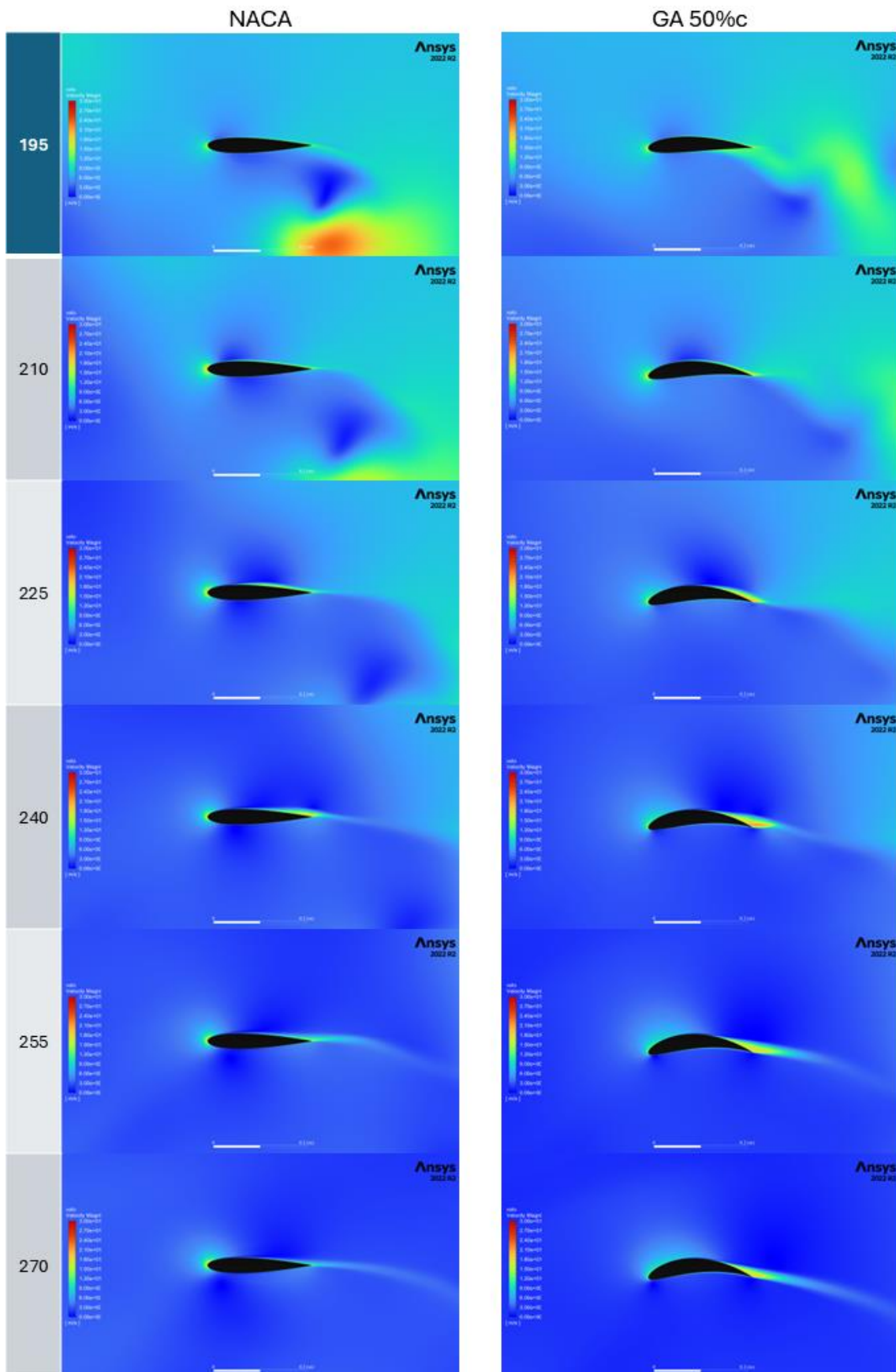
Based on the findings and conclusions of this thesis, there a small number of logical steps for continuing this research avenue. The most impactful and interesting addition would be to implement a multi-cycle optimisation loop. This would be implemented by optimising the turbine for one full rotation, running a few rotations to stabilise the periodicity of the solution, optimising the turbine in one rotation again, and repeating this process until the average period-averaged turbine power or moment coefficient no longer changes between the optimisation loops. The primary cause for this is to alleviate the discrepancy seen in Figure 5-19, where the upstream rotation ($0^\circ < \phi < 180^\circ$) is lower moment generation in the optimised turbine than in the standard NACA0015.

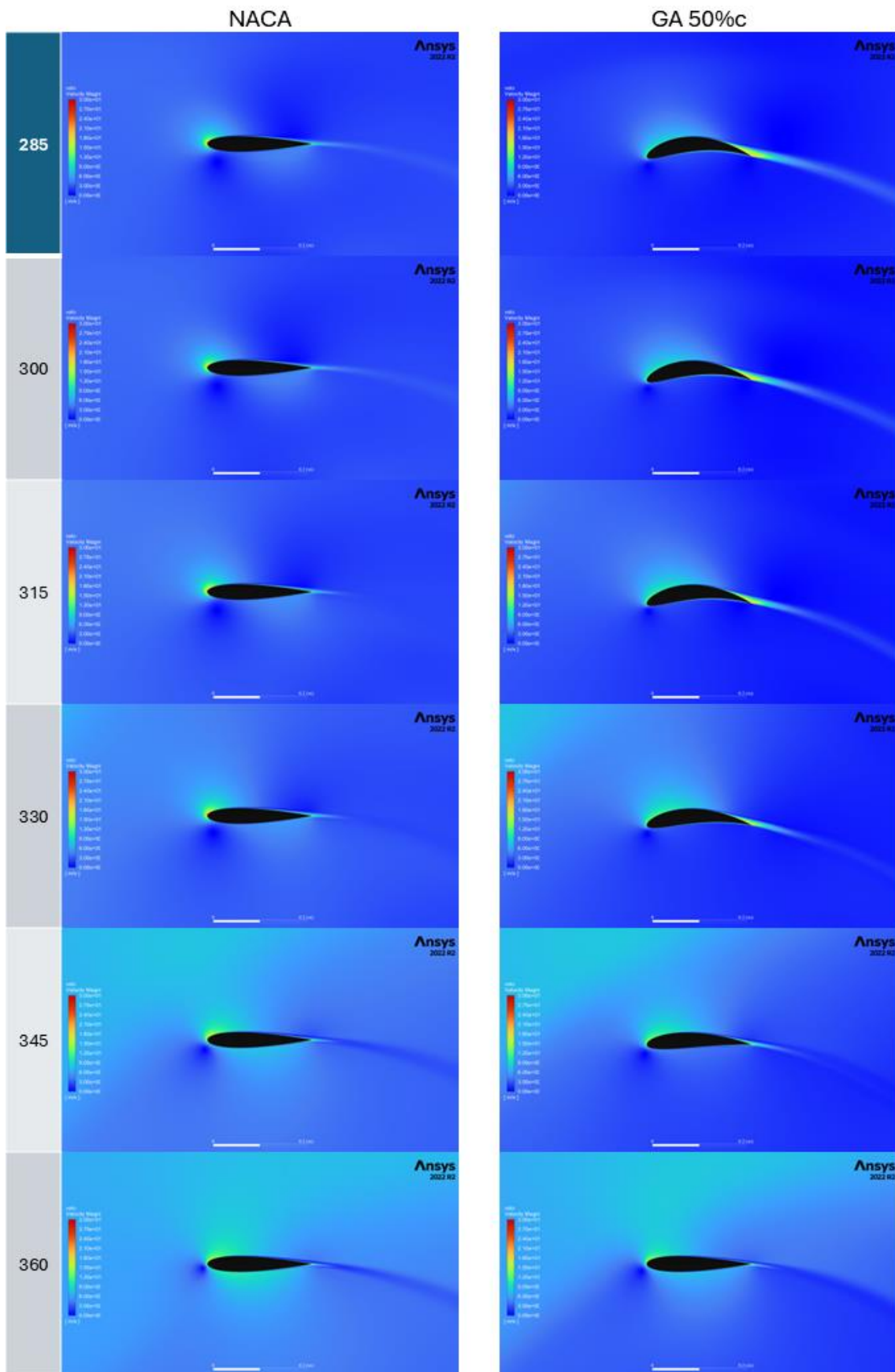
8 Appendix A

The following pages are the images of the velocity contours from $0 \rightarrow 30$ m/s for the 50%c Position of Max Camber, 5 Genotypes, 10 Optimisation Positions case.









9 Appendix B – AI Regression Based Optimised VAWT Camber Profile Velocity Field Images

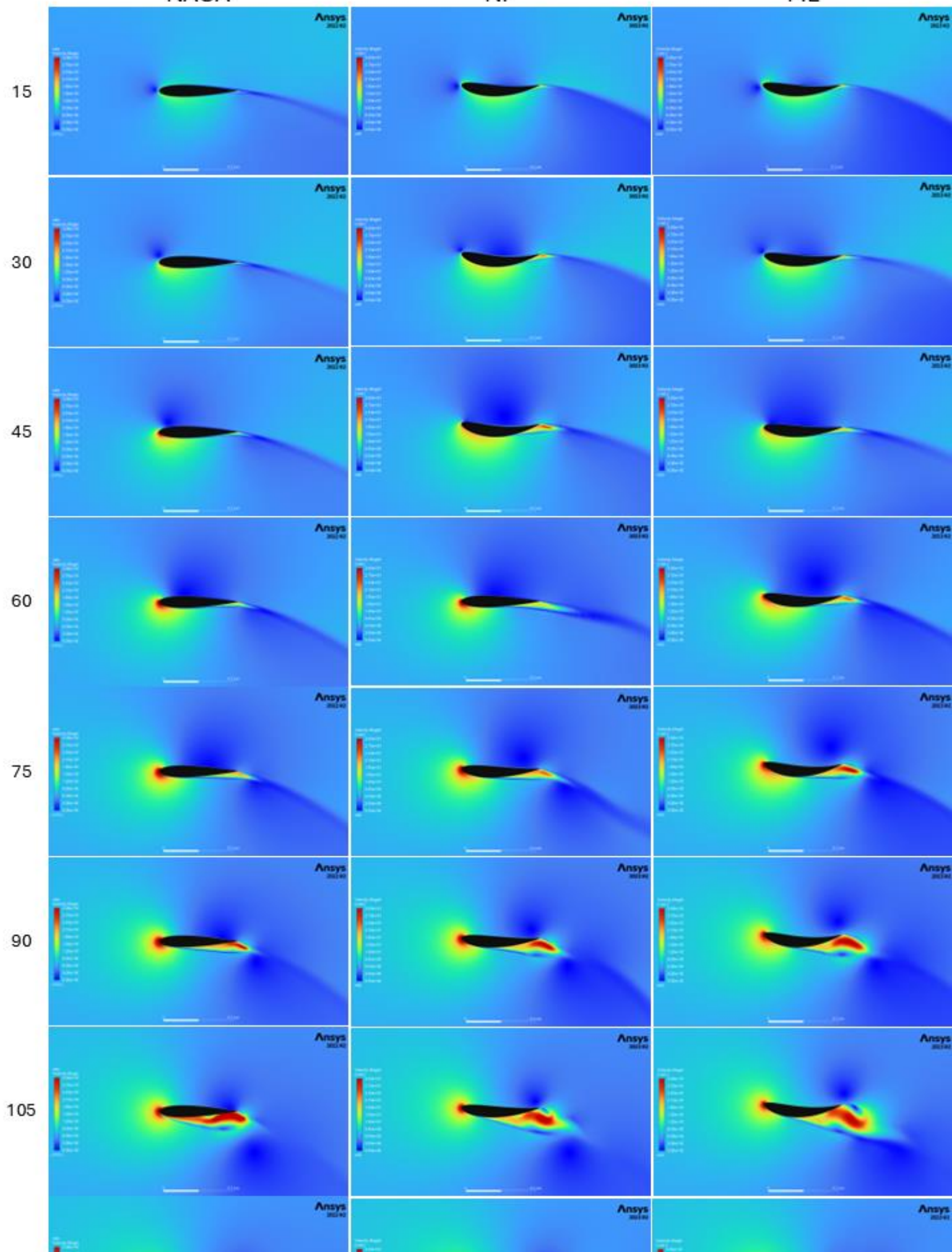
The three following pages contain the velocity contours at all optimised positions, taken from the rotating VAWT simulations. It contains the images for the cases:

- NACA0015 Symmetrical Standard VAWT Blade with no deflection
- NeuralFoil Based Genetic Algorithm Optimised VAWT Blade Camber Profile
- TensorFlow CFD Trained Genetic Algorithm Optimised VAWT Blade Camber Profile

NACA

NF

ML

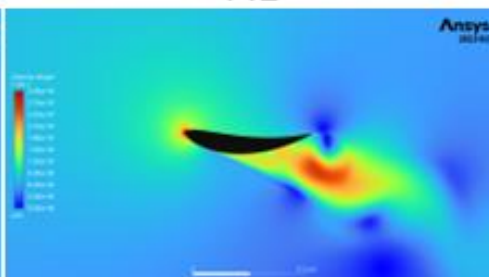
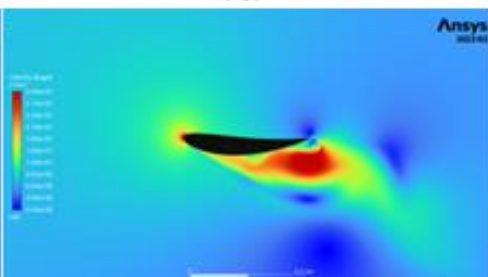
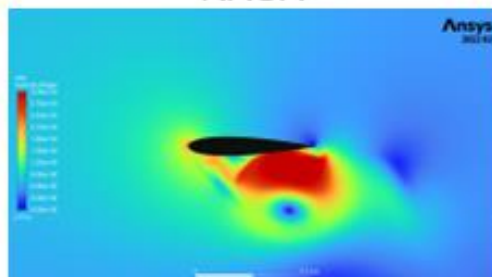


NACA

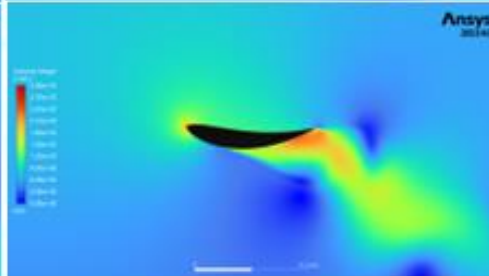
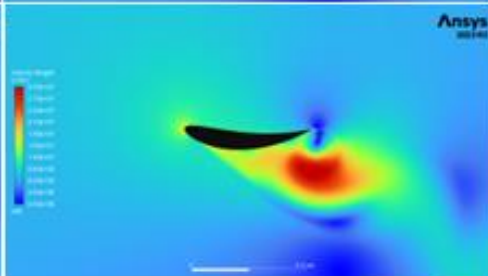
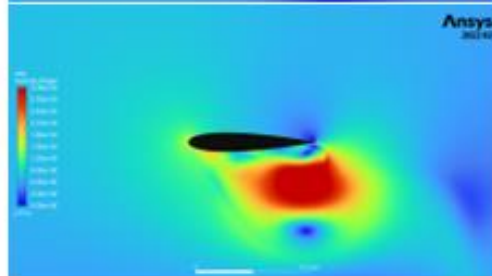
NF

ML

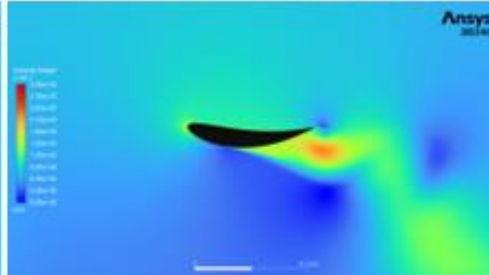
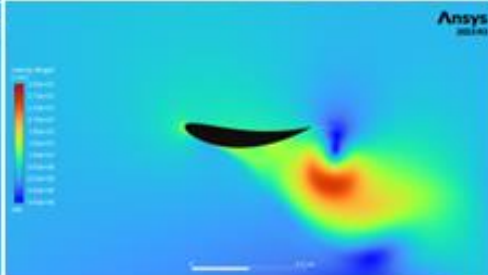
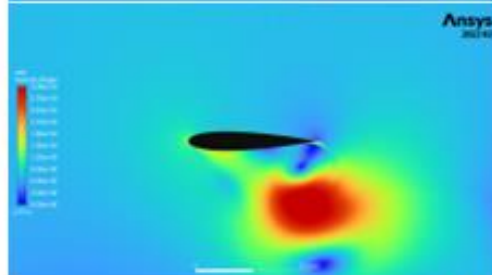
135



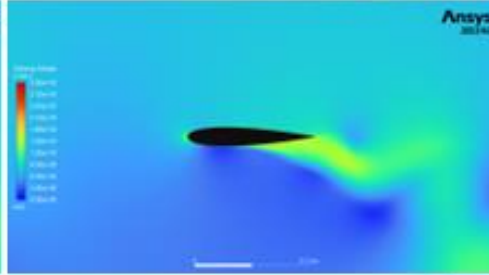
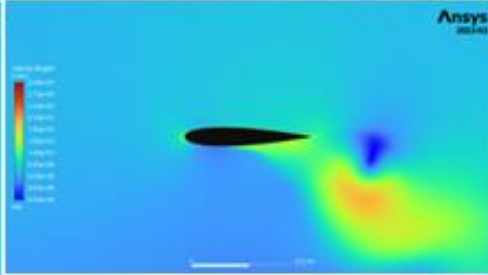
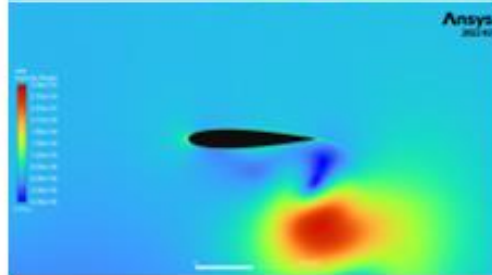
150



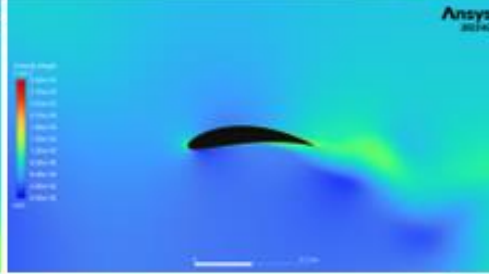
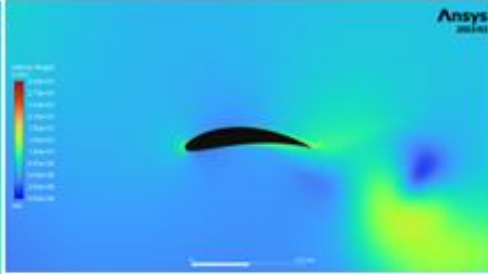
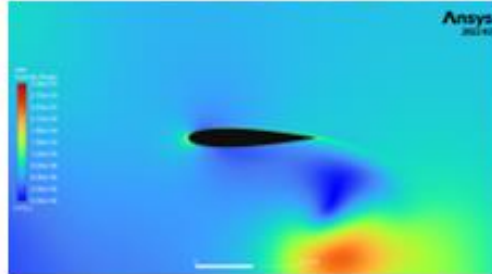
165



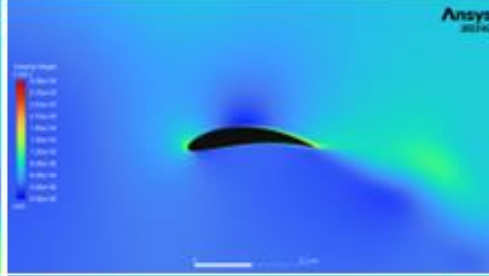
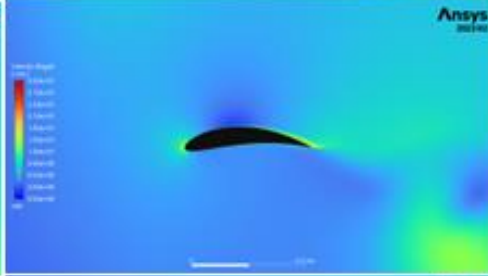
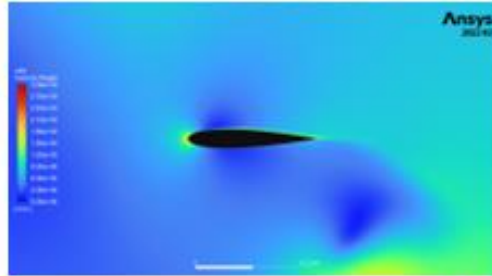
180



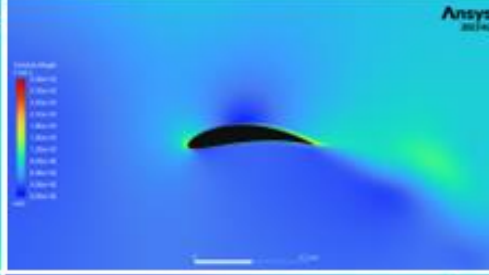
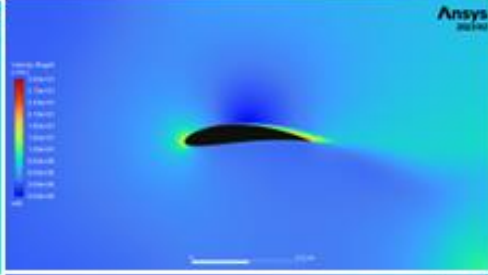
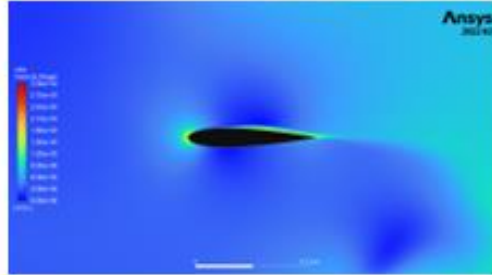
195



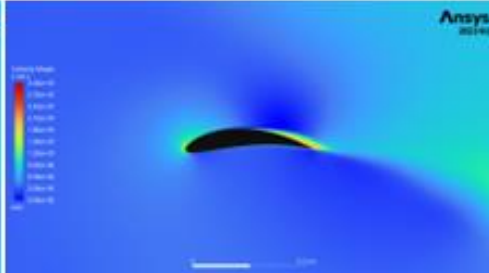
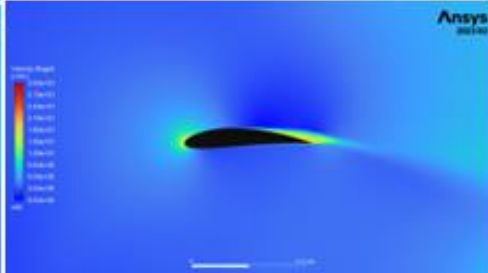
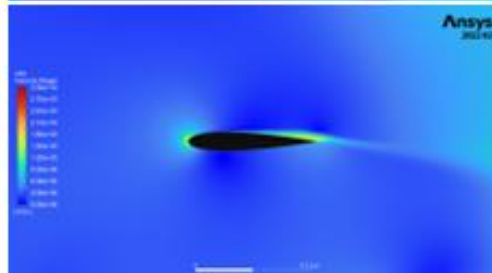
210



225



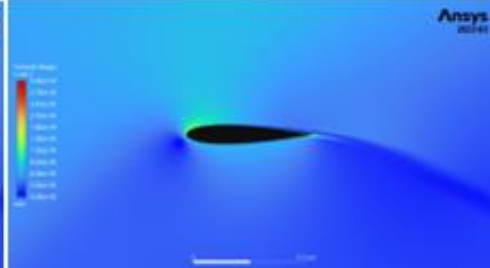
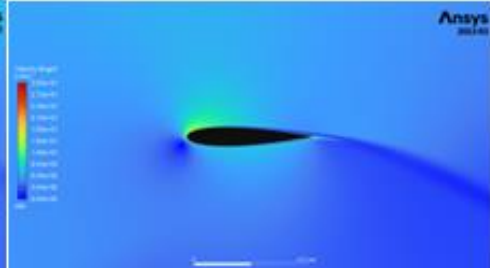
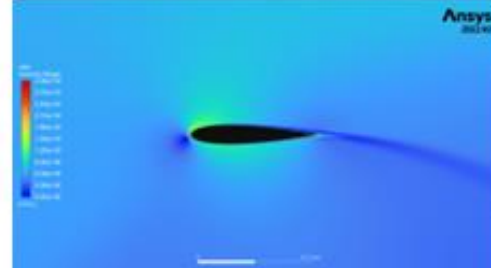
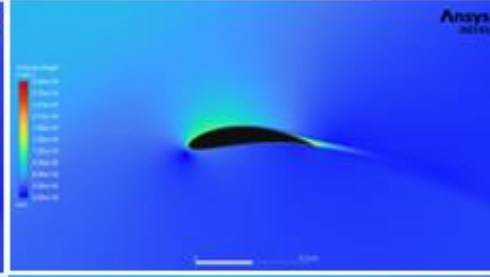
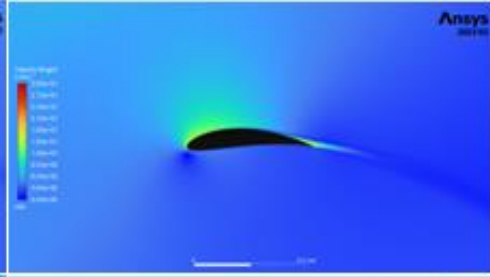
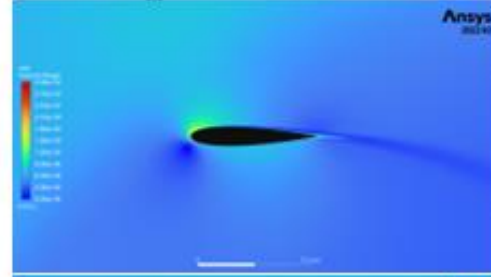
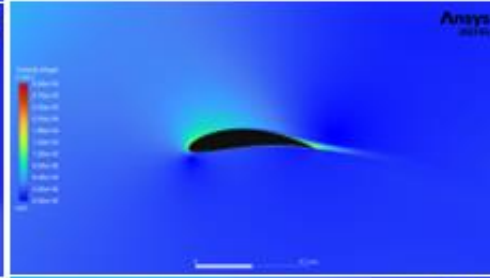
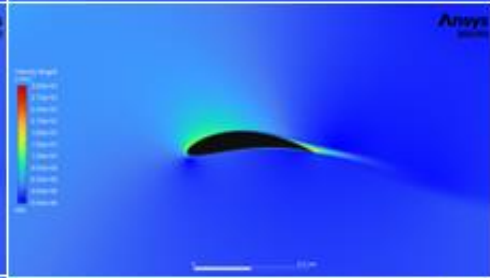
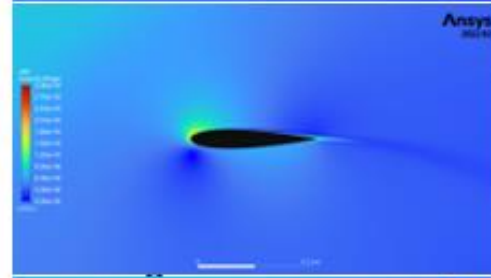
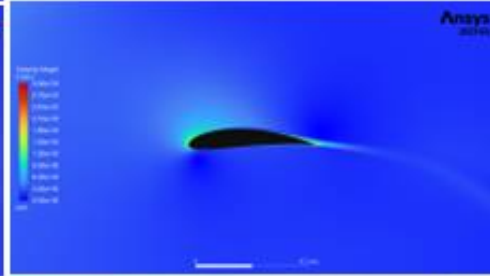
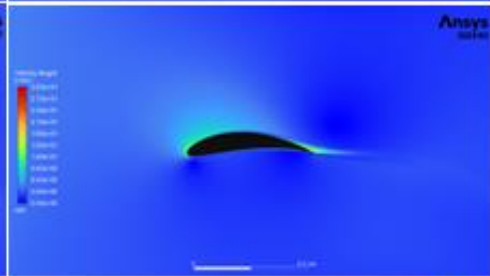
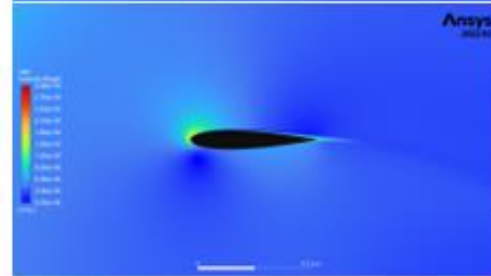
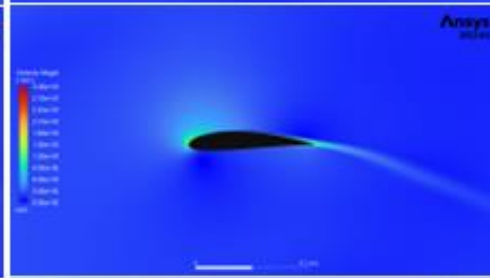
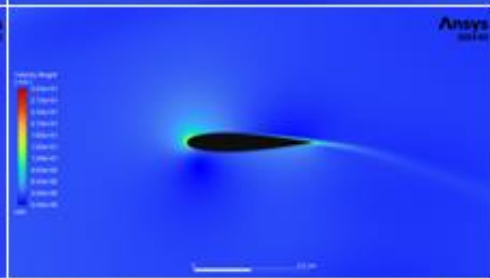
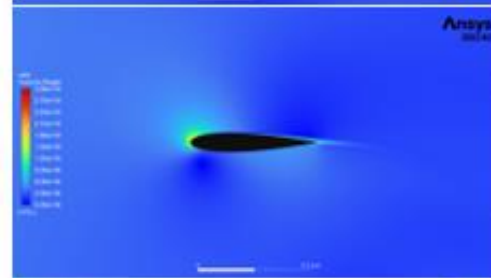
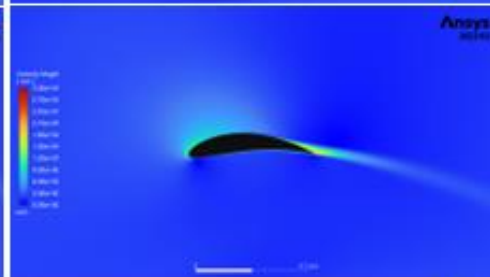
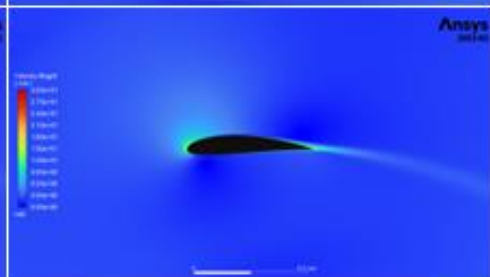
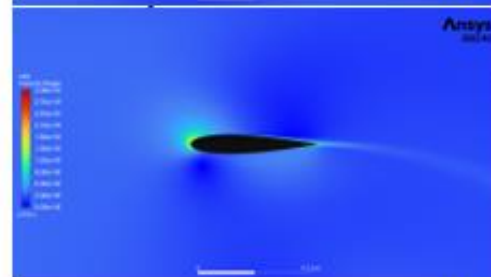
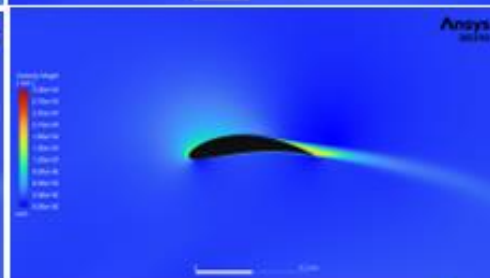
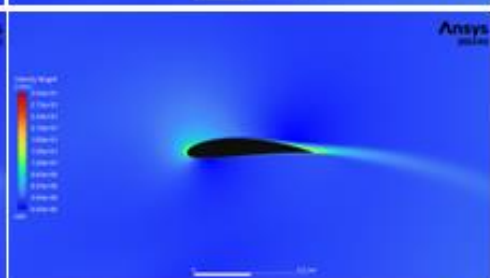
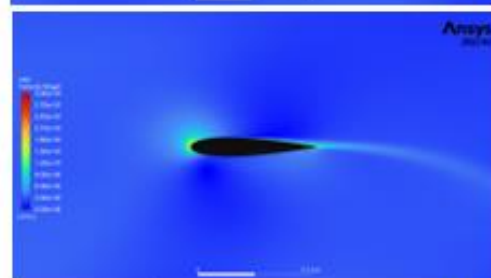
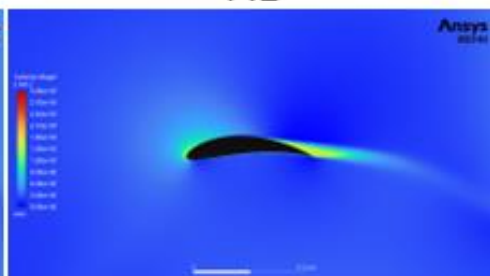
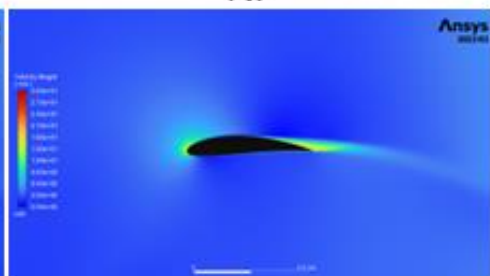
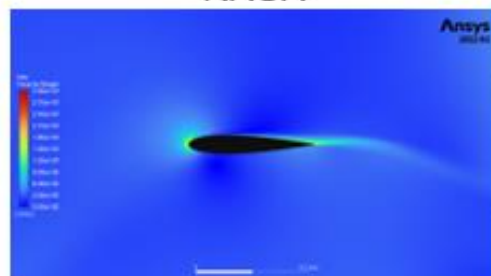
240



NACA

NF

ML



10 Appendix C – Parametric Deforming

VAWT Blade UDF

```
#include "udf.h"
DEFINE_GRID_MOTION(blade0, domain, dt, time, dtime)
{
    Thread* tf = DT_THREAD(dt);
    face_t f;
    Node* v;

    real lamtest, Xqrt, Yqrt, dXqn1, dYqn1, dXqn2, dYqn2, dx, x, Xc, Yc, Xr,
    Yr, r, dy, y, newYc, newYr, newXc, newXr, theta;
    real ts1, ts0, y0, y1, x0, x1, m1, m0, m, yNACA;
    #define chord 0.225 // chord length
    #define Rturb 0.85 // Turbine Radiuses
    #define TSR 3.0 // Turbine Tip Speed Ratio
    #define Vflow 7.0 // Free flow velocity
    #define offset 0.25 // %c distance between LE and global origin in X
    direction. Positive offset => LE < y(0)
    #define count 360.0 // %c Number of timesteps per rotation 1degree/Ts =
    360, 0.5deg/Ts = 720 etc.
    int n;
    real camber_0 = Get_Input_Parameter("camber_0"); // 36
    real camber_1 = Get_Input_Parameter("camber_1"); // 36
    real camber_2 = Get_Input_Parameter("camber_2"); // 72
    real camber_3 = Get_Input_Parameter("camber_3"); // 108
    real camber_4 = Get_Input_Parameter("camber_4"); // 144
    real camber_5 = Get_Input_Parameter("camber_5"); // 216
    real camber_6 = Get_Input_Parameter("camber_6"); // 252
    real camber_7 = Get_Input_Parameter("camber_7"); // 288
    real camber_8 = Get_Input_Parameter("camber_8"); // 324
    real camber_9 = Get_Input_Parameter("camber_9"); // 36
    real camber_10 = Get_Input_Parameter("camber_10"); // 72
    real camber_11 = Get_Input_Parameter("camber_11"); // 108
    real camber_12 = Get_Input_Parameter("camber_12"); // 144
    real camber_13 = Get_Input_Parameter("camber_13"); // 216
    real camber_14 = Get_Input_Parameter("camber_14"); // 252
    real camber_15 = Get_Input_Parameter("camber_15"); // 288
    real camber_16 = Get_Input_Parameter("camber_16"); // 324
    real camber_17 = Get_Input_Parameter("camber_17"); // 288
    real camber_18 = Get_Input_Parameter("camber_18"); // 324
    real camber_19 = Get_Input_Parameter("camber_19"); // 324
    real camber_20 = Get_Input_Parameter("camber_20"); // 324
    real camber_21 = Get_Input_Parameter("camber_21"); // 288
    real camber_22 = Get_Input_Parameter("camber_22"); // 324
```

```

real camber_23 = Get_Input_Parameter("camber_23"); // 324
real camber_24 = Get_Input_Parameter("camber_24"); // 324

int opt_count = Get_Input_Parameter("opt_count");
real p = Get_Input_Parameter("p");

real origin = p * chord;
real omega = (Vflow * TSR) / Rturb;
real T = 2 * M_PI / omega;
real ts = dtime * omega * 180/M_PI;

real flowtime = time;
real phi = (flowtime * omega) * 180/M_PI;

int rev = phi/360;

double currentTime = CURRENT_TIME;
double previousTime = PREVIOUS_TIME;
double timeUDF = time;

double phiCurrent = omega * currentTime * 180/M_PI;
double phiPrevious = omega * previousTime * 180/M_PI;
double phiTime = omega * timeUDF * 180/M_PI;

ts1 = flowtime*omega*180/M_PI - 360 * rev;
ts0 = ts1-ts; // Previous timestep calculation
if (ts0 < 0)
{
    ts0 +=360;
}

// Create array containing all Camber Values
float camber_array[] = {camber_0, camber_1, camber_2, camber_3, camber_4,
camber_5, camber_6, camber_7, camber_8, camber_9, camber_10, camber_11,
camber_12, camber_13, camber_14, camber_15, camber_16, camber_17, camber_18,
camber_19, camber_20, camber_21, camber_22, camber_23, camber_24};
int opt_phi_count = 360/opt_count; // Calculate azimuth spacing between
each optimisation position

int phi_array[opt_count]; // Create Array of Correct Size for Phi
Distribution
int length = sizeof(phi_array) / sizeof(phi_array[0]); // Calculate Length
of Phi Array

for (int k=0; k<length+1; k++)
{
    phi_array[k] = k * opt_phi_count; // Fill Phi Array with Appropriate
Values

```

```

        // printf("%d ", phi[k]);
    }

    int l1 = ts1/opt_phi_count; // Check which section of rotation following
timestep is in / Optimisation Position
    //printf("\nSection of Optimisation: %d", l1);

    // Parameters for linear interpolation
    int x0_1 = phi_array[l1];
    int x1_1 = phi_array[l1+1];
    float y0_1 = camber_array[l1];
    float y1_1 = camber_array[l1+1];

    int l0 = ts0/opt_phi_count; // Check which section of rotation current
timestep is in
    //printf("\nSection of Optimisation: %d", l0);
    // Parameters for linear interpolation
    int x0_0 = phi_array[l0];
    int x1_0 = phi_array[l0+1];
    float y0_0 = camber_array[l0];
    float y1_0 = camber_array[l0+1];

    m1 = ((y1_1 - y0_1) / (x1_1 - x0_1)) * (ts1 - x0_1) + y0_1;
    m0 = ((y1_0 - y0_0) / (x1_0 - x0_0)) * (ts0 - x0_0) + y0_0;

    //printf("\nm0 Camber Value: %f", m0);
    //printf("\nm1 Camber Value: %f", m1);

    m = (m1 - m0)/100;
    // Message0("\nRevolution: %i    UDF Phi: %f    ts1: %f    ts0: %f    m1:
%f    m0 %f", rev, phi, ts1, ts0, m1, m0);
    // Message0("phi: %f ,time: %g, current time: %g ts0: %f , ts1: %f , m0:
%f , m1: %f \n", phi, ts0, ts1, m0, m1, flowtime, CURRENT_TIME); // Print
Variables

    SET_DEFORMING_THREAD_FLAG(THREAD_T0(tf));

    begin_f_loop(f,tf) // Begin looping across the wall i select in qsh after
enabling deform mesh
    {

        f_node_loop(f,tf,n) // Begin looping across every node on face "f"
        {

            Xqrt = 0 - Rturb * sin(omega * time);
            Yqrt = Rturb - Rturb * (1 - cos(omega * time));
            v = F_NODE(f,tf,n); // Get node properties for node n on face t
            Xr = NODE_X(v);

```

```

    Yr = NODE_Y(v);
    dXqn1 = Xr - Xqrt;
    dYqn1 = Yr - Yqrt;
    lamtest = (dYqn1 * sin(omega * time)) + (dXqn1 * cos(omega * time)) +
0.25 * chord;

    if (lamtest < origin && NODE_POS_NEED_UPDATE (v)) // Deflection
happens if node is in aileron zone ie x coord is greater than pin. this is ok
if blade is stationary
    {
        NODE_POS_UPDATED(v); // Check if node has been updated in loop
        Xc = (dYqn1 * sin(omega * time)) + (dXqn1 * cos(omega * time)) +
0.25 * chord;
        Yc = (dYqn1 * cos(omega * time)) - (dXqn1 * sin(omega * time));
        x = Xc; // Variables for x-Coordinate of Node 'v'
        y = Yc;
        yNACA = m * chord * pow(origin - x, 2) / pow(origin,2);
        dy = yNACA;
        dx = 0;
        newXc = Xc + dx - 0.25 * chord;
        newYc = Yc + dy;
        dXqn2 = (newXc * cos(omega * time)) - (newYc * sin(omega * time));
        dYqn2 = (newXc * sin(omega * time)) + (newYc * cos(omega * time));
        newXr = Xqrt + dXqn2;
        newYr = Yqrt + dYqn2;
        NODE_X(v) = newXr;
        NODE_Y(v) = newYr;
    }

    if (lamtest > origin && NODE_POS_NEED_UPDATE (v)) // Deflection
happens if node is in aileron zone ie x coord is greater than pin. this is ok
if blade is stationary
    {
        NODE_POS_UPDATED(v); // Check if node has been updated in loop
        Xc = (dYqn1 * sin(omega * time)) + (dXqn1 * cos(omega * time)) +
0.25 * chord;
        Yc = (dYqn1 * cos(omega * time)) - (dXqn1 * sin(omega * time));
        x = Xc; // Variables for x-Coordinate of Node 'v'
        y = Yc;
        yNACA = m * chord * pow(x-origin, 2) / pow(chord - origin,2);
        dy = yNACA;
        dx = 0;
        newXc = Xc + dx - 0.25 * chord;
        newYc = Yc + dy;
        dXqn2 = (newXc * cos(omega * time)) - (newYc * sin(omega * time));
        dYqn2 = (newXc * sin(omega * time)) + (newYc * cos(omega * time));
        newXr = Xqrt + dXqn2;
        newYr = Yqrt + dYqn2;

```

```

        NODE_X(v) = newXr;
        NODE_Y(v) = newYr; // Update new Y position. ill need to have node
x update as well, so a new line for /newX and a new line for NODE_X(v)
    }
}
}
end_f_loop(f,tf);
}

DEFINE_GRID_MOTION(blade180, domain, dt, time, dtime)
{
    Thread *tf = DT_THREAD(dt);
    face_t f;
    Node *v;

    real lamtest, Xqrt, Yqrt, dXqn1, dYqn1, dXqn2, dYqn2, dx, x, Xc, Yc, Xr,
Yr, r, dy, y, newYc, newYr, newXc, newXr, theta;
    real ts1, ts0, y0, y1, x0, x1, m1, m0, m, yNACA;
    #define chord 0.225 // chord length
    #define Rturb 0.85 // Turbine Radiues
    #define TSR 3.0 // Turbine Tip Speed Ratio
    #define Vflow 7.0 // Free flow velocity
    #define offset 0.25 // %c distance between LE and global origin in X
direction. Positive offset => LE < y(0)
    int n;

    // Message0("Period of rotation: %g, Timetep size: %g, Deg per timestep:
%g\n", T, CURRENT_TIMESTEP, ts);
    real camber_0 = Get_Input_Parameter("camber_0");
    real camber_1 = Get_Input_Parameter("camber_1");
    real camber_2 = Get_Input_Parameter("camber_2");
    real camber_3 = Get_Input_Parameter("camber_3");
    real camber_4 = Get_Input_Parameter("camber_4");
    real camber_5 = Get_Input_Parameter("camber_5");
    real camber_6 = Get_Input_Parameter("camber_6");
    real camber_7 = Get_Input_Parameter("camber_7");
    real camber_8 = Get_Input_Parameter("camber_8");
    real camber_9 = Get_Input_Parameter("camber_9");
    real camber_10 = Get_Input_Parameter("camber_10");
    real camber_11 = Get_Input_Parameter("camber_11");
    real camber_12 = Get_Input_Parameter("camber_12");
    real camber_13 = Get_Input_Parameter("camber_13");
    real camber_14 = Get_Input_Parameter("camber_14");
    real camber_15 = Get_Input_Parameter("camber_15");
    real camber_16 = Get_Input_Parameter("camber_16");
    real camber_17 = Get_Input_Parameter("camber_17");
    real camber_18 = Get_Input_Parameter("camber_18");
    real camber_19 = Get_Input_Parameter("camber_19");

```

```

real camber_20 = Get_Input_Parameter("camber_20");
real camber_21 = Get_Input_Parameter("camber_21");
real camber_22 = Get_Input_Parameter("camber_22");
real camber_23 = Get_Input_Parameter("camber_23");
real camber_24 = Get_Input_Parameter("camber_24");

int opt_count = Get_Input_Parameter("opt_count");
real p = Get_Input_Parameter("p");

real origin = p * chord;
real omega = (Vflow * TSR) / Rturb;
real T = 2 * M_PI / omega;
real ts = dtime * omega * 180/M_PI;
// real rev;

real flowtime = time;
real phi = (flowtime * omega) * 180/M_PI + 180;

int rev = phi/360;

ts1 = flowtime*omega*180/M_PI - 360 * rev + 180;
ts0 = ts1-ts; // Previous timestep calculation
if (ts0 < 0)
{
    ts0 +=360;
}
// Create array containing all Camber Values
float camber_array[] = {camber_0, camber_1, camber_2, camber_3, camber_4,
camber_5, camber_6, camber_7, camber_8, camber_9, camber_10, camber_11,
camber_12, camber_13, camber_14, camber_15, camber_16, camber_17, camber_18,
camber_19, camber_20, camber_21, camber_22, camber_23, camber_24};
int opt_phi_count = 360/opt_count; // Calculate azimuth spacing between
each optimisation position

int phi_array[opt_count]; // Create Array of Correct Size for Phi
Distribution
int length = sizeof(phi_array) / sizeof(phi_array[0]); // Calculate Length
of Phi Array

for (int k=0; k<length+1; k++)
{
    phi_array[k] = k * opt_phi_count; // Fill Phi Array with Appropriate
Values
    // printf("%d ", phi[k]);
}

int l1 = ts1/opt_phi_count; // Check which section of rotation following
timestep is in / Optimisation Position

```

```

//printf("\nSection of Optimisation: %d", l1);

// Parameters for linear interpolation
int x0_1 = phi_array[l1];
int x1_1 = phi_array[l1+1];
float y0_1 = camber_array[l1];
float y1_1 = camber_array[l1+1];

int l0 = ts0/opt_phi_count; // Check which section of rotation current
timestep is in
//printf("\nSection of Optimisation: %d", l0);
// Parameters for linear interpolation
int x0_0 = phi_array[l0];
int x1_0 = phi_array[l0+1];
float y0_0 = camber_array[l0];
float y1_0 = camber_array[l0+1];

m1 = ((y1_1 - y0_1) / (x1_1 - x0_1)) * (ts1 - x0_1) + y0_1;
m0 = ((y1_0 - y0_0) / (x1_0 - x0_0)) * (ts0 - x0_0) + y0_0;

//printf("\nm0 Camber Value: %f", m0);
//printf("\nm1 Camber Value: %f", m1);

m = (m1 - m0)/100;

SET_DEFORMING_THREAD_FLAG(THREAD_T0(tf));

begin_f_loop(f,tf) // Begin looping across the wall i select in qsh after
enabling deform mesh
{
    f_node_loop(f,tf,n) // Begin looping across every node on face "f"
    {
        // omega = 83.7; // angular rotation rate of aileron linking to how
many 'i' that im morphing with so it indirectly decides the max angle achieved
        Xqrt = 0 - Rturb * sin(omega * time + M_PI);
        Yqrt = Rturb - Rturb * (1 - cos(omega * time + M_PI));
        v = F_NODE(f,tf,n); // Get node properties for node n on face t
        Xr = NODE_X(v);
        Yr = NODE_Y(v);
        dXqn1 = Xr - Xqrt;
        dYqn1 = Yr - Yqrt;
        lamtest = (dYqn1 * sin(omega * time + M_PI)) + (dXqn1 * cos(omega *
time + M_PI)) + 0.25 * chord;

        if (lamtest < origin && NODE_POS_NEED_UPDATE (v)) // Deflection
happens if node is in aileron zone ie x coord is greater than pin. this is ok
if blade is stationary

```

```

    {
        NODE_POS_UPDATED(v); // Check if node has been updated in loop
        Xc = (dYqn1 * sin(omega * time + M_PI)) + (dXqn1 * cos(omega *
time + M_PI)) + 0.25 * chord;
        Yc = (dYqn1 * cos(omega * time + M_PI)) - (dXqn1 * sin(omega *
time + M_PI));
        x = Xc; // Variables for x-Coordinate of Node 'v'
        y = Yc;
        yNACA = m * chord * pow(origin - x, 2) / pow(origin,2);
        dy = yNACA;
        dx = 0;
        newXc = Xc + dx - 0.25 * chord;
        newYc = Yc + dy;
        dXqn2 = (newXc * cos(omega * time + M_PI)) - (newYc * sin(omega *
time + M_PI));
        dYqn2 = (newXc * sin(omega * time + M_PI)) + (newYc * cos(omega *
time + M_PI));
        newXr = Xqrt + dXqn2;
        newYr = Yqrt + dYqn2;
        NODE_X(v) = newXr;
        NODE_Y(v) = newYr; // Update new Y position. ill need to have node
x update as well, so a new line for /newX and a new line for NODE_X(v)
    }

    if (lamtest > origin && NODE_POS_NEED_UPDATE (v)) // Deflection
happens if node is in aileron zone ie x coord is greater than pin. this is ok
if blade is stationary
    {
        NODE_POS_UPDATED(v); // Check if node has been updated in loop
        Xc = (dYqn1 * sin(omega * time + M_PI)) + (dXqn1 * cos(omega *
time + M_PI)) + 0.25 * chord;
        Yc = (dYqn1 * cos(omega * time + M_PI)) - (dXqn1 * sin(omega *
time + M_PI));
        x = Xc; // Variables for x-Coordinate of Node 'v'
        y = Yc;
        yNACA = m * chord * pow(x-origin, 2) / pow(chord - origin,2);
        dy = yNACA;
        dx = 0;
        newXc = Xc + dx - 0.25 * chord;
        newYc = Yc + dy;
        dXqn2 = (newXc * cos(omega * time + M_PI)) - (newYc * sin(omega *
time + M_PI));
        dYqn2 = (newXc * sin(omega * time + M_PI)) + (newYc * cos(omega *
time + M_PI));
        newXr = Xqrt + dXqn2;
        newYr = Yqrt + dYqn2;
        NODE_X(v) = newXr;

```

```
        NODE_Y(v) = newYr; // Update new Y position. ill need to have node
x update as well, so a new line for /newX and a new line for NODE_X(v)
    }
}
}
end_f_loop(f,tf);
}
```

11 References

- [1] Directorate-General for the Environment (European Commission), “Living well, within the limits of our planet. 7th EAP – the new general Union environment action programme to 2020,” *EU Publ.*, p. 37, 2014, doi: 10.2779/57220.
- [2] United Nations, “Summary of the Paris Agreement,” *United Nations Framew. Conv. Clim. Chang.*, pp. 27–52, 2015, [Online]. Available: <http://bigpicture.unfccc.int/#content-the-paris-agreemen>
- [3] Wind Europe, “Wind energy in Europe in 2018 - Trends and statistics.” [Online]. Available: <https://windeurope.org/wp-content/uploads/files/about-wind/statistics/WindEurope-Annual-Statistics-2018.pdf>
- [4] M. Scungio, F. Arpino, V. Focanti, M. Profili, and M. Rotondi, “Wind tunnel testing of scaled models of a newly developed Darrieus-style vertical axis wind turbine with auxiliary straight blades,” *Energy Convers. Manag.*, vol. 130, pp. 60–70, 2016, doi: 10.1016/j.enconman.2016.10.033.
- [5] S. S. Johannes, “Rotor adapted to be driven by wind or flowing water,” 1929, *Google Patents*. [Online]. Available: <https://patents.google.com/patent/US1697574>
- [6] Z. Li, R. Han, P. Gao, and C. Wang, “Analysis and implementation of a drag-type vertical-axis wind turbine for small distributed wind energy systems,” *Adv. Mech. Eng.*, vol. 11, no. 1, p. 1687814019825709, Jan. 2019, doi: 10.1177/1687814019825709.
- [7] D. G. J. Marie, “Turbine having its rotating shaft transverse to the flow of the current,” 1931, *Google Patents*. [Online]. Available: <https://patents.google.com/patent/US1835018A/en>
- [8] T. J. Price, “UK large-scale wind power programme from 1970 to 1990: the Carmarthen Bay experiments and the musgrove vertical-axis turbines,” *Wind Eng.*, vol. 30, no. 3, pp. 225–242, 2006.

- [9] I. D. Mays, C. A. Morgan, M. B. Anderson, and S. J. R. Powles, "Experience with the VAWT 850 demonstration project," in *Proceedings of European Community Wind Energy Conference*, 1990, pp. 482–487.
- [10] T. M. Letcher, *Wind Energy Engineering: A Handbook for Onshore and Offshore Wind Turbines*. Elsevier Inc., 2017.
- [11] R. E. Sheldahl and P. C. Klimas, "Aerodynamic characteristics of seven symmetrical airfoil sections through 180-degree angle of attack for use in aerodynamic analysis of vertical axis wind turbines," 1981. [Online]. Available: <https://www.osti.gov/biblio/6548367-aerodynamic-characteristics-seven-symmetrical-airfoil-sections-through-degree-angle-attack-use-aerodynamic-analysis-vertical-axis-wind-turbines>
- [12] S. Ellora and M. Mitra, *Design, and Analysis of Bio-Inspired Tricopter with Delta Manipulator for Industrial Applications*. 2021. doi: 10.13140/RG.2.2.32249.54881.
- [13] T. Benson, "Incorrect Lift Theory #1 - Equal Transit Theory." [Online]. Available: <https://www.grc.nasa.gov/www/k-12/VirtualAero/BottleRocket/airplane/wrong1.html>
- [14] T. Benson, "Incorrect Lift Theory #2 - Skipping Stone Theory." [Online]. Available: <https://www.grc.nasa.gov/www/k-12/VirtualAero/BottleRocket/airplane/wrong2.html>
- [15] N. Hall, "Incorrect Lift Theory #3 - Venturi Theory." [Online]. Available: <https://www1.grc.nasa.gov/beginners-guide-to-aeronautics/venturi-theory/>
- [16] F. M. White, *Fluid Mechanics*, Eighth Edi. New York, NY: McGraw-Hill Education, 2016.
- [17] T. Benson, "Effect of Shape on Lift," Apr. 2018, NASA.
- [18] N. Hall, "Inclination Effects on Lift," Apr. 2018, NASA.
- [19] D. Ma, Y. Zhao, Y. Qiao, and G. Li, "Effects of relative thickness on aerodynamic characteristics of airfoil at a low Reynolds number," *Chinese J. Aeronaut.*, vol. 28, no. 4, pp. 1003–1015, Aug. 2015, doi: 10.1016/j.cja.2015.05.012.

- [20] L. A. M. Danao, N. Qin, and R. J. Howell, "A Numerical Study of Blade Thickness and Camber Effects on Vertical Axis Wind Turbines," *Proc. Inst. Mech. Eng. Part a J. Power Energy*, vol. 226, no. 7, pp. 867–881, 2012, doi: 10.1177/0957650912454403.
- [21] H. Beri and Y. Yao, "Effect of Camber Airfoil on Self Starting of Vertical Axis Wind Turbine," *J. Environ. Sci. Technol.*, vol. 4, no. 3, pp. 302–312, 2011, doi: 10.3923/jest.2011.302.312.
- [22] A. Meana-Fernández, I. Solís-Gallego, J. M. F. Oro, K. M. A. Díaz, and S. Velarde-Suárez, "Parametrical Evaluation of the Aerodynamic Performance of Vertical Axis Wind Turbines for the Proposal of Optimized Designs," *Energy*, vol. 147, pp. 504–517, 2018, doi: 10.1016/j.energy.2018.01.062.
- [23] H. Day, D. Ingham, L. Ma, and M. Pourkashanian, "Adjoint based optimisation for efficient VAWT blade aerodynamics using CFD," *J. Wind Eng. Ind. Aerodyn.*, vol. 208, p. 104431, Jan. 2021, doi: 10.1016/J.JWEIA.2020.104431.
- [24] R. E. Brown and F. Scheurich, "Effect of Dynamic Stall on the Aerodynamics of Vertical-Axis Wind Turbines," *Aiaa J.*, vol. 49, no. 11, pp. 2511–2521, 2011, doi: 10.2514/1.j051060.
- [25] F. Scheurich, T. M. Fletcher, and R. E. Brown, "Simulating the Aerodynamic Performance and Wake Dynamics of a Vertical-axis Wind Turbine," *Wind Energy*, vol. 14, no. 2, pp. 159–177, 2011, doi: 10.1002/we.409.
- [26] M. T. Bouzaher, B. Guerira, and M. Hadid, "Performance analysis of a vertical axis tidal turbine with flexible blades," *J. Mar. Sci. Appl.*, vol. 16, no. 1, pp. 73–80, 2017, doi: 10.1007/s11804-017-1391-0.
- [27] M. Baghdadi, S. Elkoush, B. Akle, and M. Elkhoury, "Dynamic shape optimization of a vertical-axis wind turbine via blade morphing technique," *Renew. Energy*, vol. 154, pp. 239–251, 2020, [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0960148120303293>
- [28] O. Eboibi, L. A. M. Danao, and R. J. Howell, "Experimental investigation of the influence of solidity on the performance and flow field aerodynamics of

- vertical axis wind turbines at low Reynolds numbers,” *Renew. Energy*, vol. 92, pp. 474–483, 2016, doi: 10.1016/j.renene.2016.02.028.
- [29] H. Yusoff, M. Arifin, N. Shaari, M. Zainuddin, R. Mahmud, S. Suhaimi, and W. A. N. W. Mohamed, “Lift and Drag Performance Based on Varying Flapping Wing Camber at Low Reynolds Number of Micro Air Vehicles (MAVs),” *J. Soc. Automot. Eng. Malaysia*, vol. 5, no. 3, pp. 348–354, 2021, doi: 10.56381/jsaem.v5i3.177.
- [30] A. E. Rivero, S. Fournier, M. Μανωλέσος, J. E. Cooper, and B. K. S. Woods, “Experimental Aerodynamic Comparison of Active Camber Morphing and Trailing-Edge Flaps,” *Aiaa J.*, vol. 59, no. 7, pp. 2627–2640, 2021, doi: 10.2514/1.j059606.
- [31] A.-J. Buchner, M. W. Lohry, L. Martinelli, J. Soria, and A. J. Smits, “Dynamic stall in vertical axis wind turbines: Comparing experiments and computations,” *J. Wind Eng. Ind. Aerodyn.*, vol. 146, pp. 163–171, 2015, doi: <https://doi.org/10.1016/j.jweia.2015.09.001>.
- [32] B. Jay M., “Dynamic Stall Effects and Applications To High Performance Aircraft,” NASA Langley Technical Report Server, 2003.
- [33] A. T. Conlisk, “Modern helicopter rotor aerodynamics,” *Prog. Aerosp. Sci.*, vol. 37, no. 5, pp. 419–476, 2001.
- [34] G. J. Leishman, *Principles of helicopter aerodynamics with CD extra*. Cambridge university press, 2006.
- [35] J. Yen and N. A. Ahmed, “Enhancing vertical axis wind turbine by dynamic stall control using synthetic jets,” *J. Wind Eng. Ind. Aerodyn.*, vol. 114, pp. 12–17, 2013.
- [36] G. Lachmann, C. Wieselsberger, R. Katzmayer, and L. Kirste, “Experiments with Slotted Wings,” *Tech. Notes - Natl. Advis. Comm. Aeronaut.*, no. 71, pp. 1–43, 1921, [Online]. Available: <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Experiments+with+Slotted+Wings#0>

- [37] (John David) Anderson John D., *Fundamentals of Aerodynamics*, Sixth Inte. in McGraw-Hill Series in Aeronautical and Aerospace Engineering. New York, NY: McGraw-Hill Education, 2017.
- [38] F. H. Page, "Developments in Aircraft Design by the Use of Slotted Wings," *Flight*, vol. 13, no. 51, pp. 844–860, Dec. 1921.
- [39] P. D. Chougule, L. Rosendahl, and S. R. K. Nielsen, "Experimental study of the effect of a slat angle on double-element airfoil and application in vertical axis wind turbine," *Ships Offshore Struct.*, vol. 10, no. 2, pp. 176–182, 2015, doi: 10.1080/17445302.2014.918685.
- [40] M. Scungio, F. Arpino, V. Focanti, M. Profili, and M. Rotondi, "Wind tunnel testing of scaled models of a newly developed Darrieus-style vertical axis wind turbine with auxiliary straight blades," *Energy Convers. Manag.*, vol. 130, pp. 60–70, 2016, doi: 10.1016/j.enconman.2016.10.033.
- [41] "A Real Fairey Story," *Flight*, vol. 39, no. 1678, p. 153, Feb. 1941.
- [42] Y. Li, C. Zhao, C. Qu, S. Zhao, F. Feng, and K. Tagawa, "Effect of Auxiliary Blade on Aerodynamic Characteristics of Vertical Axis Wind Turbine by Numerical Simulation," *Int. J. Rotating Mach.*, vol. 2019, 2019, doi: 10.1155/2019/8098160.
- [43] Y. Wei and C. Li, "Investigation of the two-element airfoil with flap structure for the vertical axis wind turbine," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 52, no. 5, pp. 1–6, 2013, [Online]. Available: <http://search.proquest.com/docview/1762069715/>
- [44] ANSYS Inc., "ANSYS Fluent Tutorial Guide 18," vol. 15317, no. April, pp. 724–746, 2018, doi: 10.1016/0140-3664(87)90311-2.
- [45] Y. Yang, C. Li, W. Zhang, X. Guo, and Q. Yuan, "Investigation on aerodynamics and active flow control of a vertical axis wind turbine with flapped airfoil," *J. Mech. Sci. Technol.*, vol. 31, no. 4, pp. 1645–1655, 2017, doi: 10.1007/s12206-017-0312-0.
- [46] Q. Liu, W. Miao, C. Li, W. Hao, H. Zhu, and Y. Deng, "Effects of trailing-edge

- movable flap on aerodynamic performance and noise characteristics of VAWT,” *Energy*, no. 189, p. 116271, 2019, doi: 10.1016/j.energy.2019.116271.
- [47] R. Myose, M. Papadakis, and I. Heron, “Gurney flap experiments on airfoils, wings, and reflection plane model,” *J. Aircr.*, vol. 35, no. 2, pp. 206–211, 1998, doi: 10.2514/2.2309.
- [48] A. Bianchini, F. Balduzzi, D. Di Rosa, and G. Ferrara, “On the use of Gurney Flaps for the aerodynamic performance augmentation of Darrieus wind turbines,” *Energy Convers. Manag.*, vol. 184, no. February, pp. 402–415, 2019, doi: 10.1016/j.enconman.2019.01.068.
- [49] H. Zhu, W. Hao, C. Li, and Q. Ding, “Numerical study of effect of solidity on vertical axis wind turbine with Gurney flap,” *J. Wind Eng. Ind. Aerodyn.*, vol. 186, no. August 2018, pp. 17–31, 2019, doi: 10.1016/j.jweia.2018.12.016.
- [50] A. Rezaeiha, H. Montazeri, and B. Blocken, “Active flow control for power enhancement of vertical axis wind turbines: Leading-edge slot suction,” *Energy*, vol. 189, p. 116131, 2019, doi: 10.1016/j.energy.2019.116131.
- [51] X. Sun, Y. Xu, and D. Huang, “Numerical simulation and research on improving aerodynamic performance of vertical axis wind turbine by co-flow jet,” *J. Renew. Sustain. Energy*, vol. 11, no. 1, 2019, [Online]. Available: <http://search.proquest.com/docview/2169981216/>
- [52] C. Song, G. Wu, W. Zhu, X. Zhang, and J. Zhao, “Numerical Investigation on the Effects of Airfoil Leading Edge Radius on the Aerodynamic Performance of H-Rotor Darrieus Vertical Axis Wind Turbine,” *Energies*, vol. 12, no. 19, p. 3794, 2019, doi: 10.3390/en12193794.
- [53] Z. Wang and M. Zhuang, “Leading-edge serrations for performance improvement on a vertical-axis wind turbine at low tip-speed-ratios,” *Appl. Energy*, vol. 208, no. June, pp. 1184–1197, 2017, doi: 10.1016/j.apenergy.2017.09.034.
- [54] Y. Wang, S. Shen, G. Li, D. Huang, and Z. Zheng, “Investigation on aerodynamic performance of vertical axis wind turbine with different series airfoil shapes,”

- Renew. Energy*, vol. 126, pp. 801–818, 2018, doi: <https://doi.org/10.1016/j.renene.2018.02.095>.
- [55] W. Liu and Q. Xiao, “Investigation on Darrieus type straight blade vertical axis wind turbine with flexible blade,” *Ocean Eng.*, vol. 110, pp. 339–356, 2015, doi: 10.1016/j.oceaneng.2015.10.027.
 - [56] D. H. Zeiner-Gundersen, “A vertical axis hydrodynamic turbine with flexible foils, passive pitching, and low tip speed ratio achieves near constant RPM,” *Energy*, vol. 77, pp. 297–304, 2014, doi: 10.1016/j.energy.2014.08.008.
 - [57] D. H. Zeiner-Gundersen, “A novel flexible foil vertical axis turbine for river, ocean, and tidal applications,” *Appl. Energy*, vol. 151, pp. 60–66, 2015, doi: 10.1016/j.apenergy.2015.04.005.
 - [58] J. M. Miao and M. H. Ho, “Effect of flexure on aerodynamic propulsive efficiency of flapping flexible airfoil,” *J. Fluids Struct.*, vol. 22, no. 3, pp. 401–419, 2006, doi: 10.1016/j.jfluidstructs.2005.11.004.
 - [59] C. Zhuang, G. Yang, Y. Zhu, and D. Hu, “Effect of morphed trailing-edge flap on aerodynamic load control for a wind turbine blade section,” *Renew. Energy*, no. xxxx, 2019, doi: 10.1016/j.renene.2019.10.082.
 - [60] W. Liu, Q. Xiao, and F. Cheng, “A bio-inspired study on tidal energy extraction with flexible flapping wings,” *Bioinspir. Biomim.*, vol. 8, no. 3, p. 36011, Aug. 2013, doi: 10.1088/1748-3182/8/3/036011.
 - [61] Y. Wang, H. Tong, H. Sima, J. Wang, J. Sun, and D. Huang, “Experimental study on aerodynamic performance of deformable blade for vertical axis wind turbine,” *Energy*, vol. 181, pp. 187–201, 2019, doi: 10.1016/j.energy.2019.03.181.
 - [62] H. M. S. M. Mazarbhuiya, A. Biswas, and K. K. Sharma, “A 2D Numerical Simulation of Blade Twist Effect on the Aerodynamic Performance of an Asymmetric Blade Vertical Axis Wind Turbine in Low Wind Speed,” *Eai Endorsed Trans. Energy Web*, vol. 0, no. 0, p. 162828, 2018, doi: 10.4108/eai.13-7-2018.162828.

- [63] M. R. Tirandaz, A. Rezaeiha, and D. Micallef, "Towards smart blades for vertical axis wind turbines: different airfoil shapes and tip speed ratios," *Wind Energy Sci.*, vol. 8, no. 9, pp. 1403–1424, 2023, doi: 10.5194/wes-8-1403-2023.
- [64] M. R. Tirandaz and A. Rezaeiha, "Effect of airfoil shape on power performance of vertical axis wind turbines in dynamic stall: Symmetric Airfoils," *Renew. Energy*, vol. 173, pp. 422–441, 2021, doi: <https://doi.org/10.1016/j.renene.2021.03.142>.
- [65] M. Mifsud, D. G. MacManus, and S. Shaw, "A Variable-fidelity Aerodynamic Model Using Proper Orthogonal Decomposition," *Int. J. Numer. Methods Fluids*, vol. 82, no. 10, pp. 646–663, 2016, doi: 10.1002/fld.4234.
- [66] P. LeGresley and J. Alonso, "Investigation of non-linear projection for POD based reduced order models for aerodynamics," in *39th aerospace sciences meeting and exhibit*, 2001, p. 926.
- [67] B. I. Epureanu, E. H. Dowell, and K. C. Hall, "A parametric analysis of reduced order models of potential flows in turbomachinery using proper orthogonal decomposition," in *Turbo Expo: Power for Land, Sea, and Air*, American Society of Mechanical Engineers, 2001, p. V001T03A056.
- [68] H. M. Day, "Aerodynamic Optimisation of Vertical Axis Wind Turbines Using Adjoint Methods in CFD," no. November, 2021.
- [69] C. S. Ferreira and B. Geurts, "Aerofoil optimization for vertical-axis wind turbines," *Wind energy (Chichester, England)*, vol. 18, no. 8, pp. 1371–1385, 2015, doi: 10.1002/we.1762.
- [70] M. E. Tejeda-del-Cueto, M. A. Flores-Alfaro, M. Toledo-Velázquez, L. D. Santos-Cortes, J. Hernández-Hernández, and M. O. Vigueras-Zúñiga, "Airfoil Lift Coefficient Optimization Using Genetic Algorithm and IGP Parameterization: Volume 1," 2024. doi: 10.3390/aerospace11010044.
- [71] M.-Y. (吴明雨) Wu, X.-Y. (袁心怡) Yuan, Z.-H. (陈志华) Chen, W.-T. (吴威涛) Wu, Y. (华越) Hua, and N. Aubry, "Airfoil shape optimization using genetic

- algorithm coupled deep neural networks,” *Phys. Fluids*, vol. 35, no. 8, p. 85140, Aug. 2023, doi: 10.1063/5.0160954.
- [72] S. Ahmed, K. Kamal, T. A. Ratlamwala, S. Mathavan, G. Hussain, M. Alkahtani, and M. B. Alsultan, “Aerodynamic Analyses of Airfoils Using Machine Learning as an Alternative to RANS Simulation,” 2022. doi: 10.3390/app12105194.
- [73] Q. (杜秋晚) Du, T. (刘天源) Liu, L. (杨立克) Yang, L. (李良梁) Li, D. (张荻) Zhang, and Y. (谢永慧) Xie, “Airfoil design and surrogate modeling for performance prediction based on deep learning method,” *Phys. Fluids*, vol. 34, no. 1, p. 15111, Jan. 2022, doi: 10.1063/5.0075784.
- [74] Y. Sun, U. Sengupta, and M. Juniper, “Physics-informed deep learning for simultaneous surrogate modeling and PDE-constrained optimization of an airfoil geometry,” *Comput. Methods Appl. Mech. Eng.*, vol. 411, p. 116042, 2023, doi: <https://doi.org/10.1016/j.cma.2023.116042>.
- [75] H. R. Karbasian and W. Rees, *A Deep-Learning Surrogate Model Approach for Optimization of Morphing Airfoils*. 2023. doi: 10.2514/6.2023-1619.
- [76] ANSYS Inc, “Conservation of Momentum Equations.”
- [77] COMSOL, “What Are the Navier-Stokes Equations?,” 2015. [Online]. Available: <https://www.comsol.com/multiphysics/navier-stokes-equations>
- [78] ANSYS Inc., “Fluent Theory Guide,” *Fluent Theory Guide* 23.2.
- [79] A. Barnes, D. Marshall-Cross, and B. R. Hughes, “Validation and comparison of turbulence models for predicting wakes of vertical axis wind turbines,” *J. Ocean Eng. Mar. Energy*, vol. 7, no. 4, pp. 339–362, 2021, doi: 10.1007/s40722-021-00204-z.
- [80] M. M. Elsakka, D. B. Ingham, L. Ma, and M. Pourkashanian, “CFD analysis of the angle of attack for a vertical axis wind turbine blade,” *Energy Convers. Manag.*, vol. 182, no. December 2018, pp. 154–165, Feb. 2019, doi: 10.1016/j.enconman.2018.12.054.

- [81] Q. Li *et al.*, “Wind tunnel and numerical study of a straight-bladed vertical axis wind turbine in three-dimensional analysis (Part I: For predicting aerodynamic loads and performance),” *Energy*, vol. 106, pp. 443–452, 2016, doi: 10.1016/j.energy.2016.03.089.
- [82] F. R. Menter, “Improved two-equation k-turbulence models for aerodynamic flows,” *NASA Tech. Memo.*, vol. 103975, no. 1_, p. 3t, 1992.
- [83] Autodesk, “What Y+ values should be used for the ke and SST turbulence models? | CFD | Autodesk Knowledge Network.” [Online]. Available: <https://knowledge.autodesk.com/support/cfd/learn-explore/caas/sfdcarticles/sfdcarticles/What-Y-values-should-be-used-for-the-ke-and-SST-turbulence-models.html>
- [84] J.-P. Marx, “PhD Example Files - Genetic Algorithm Optimisation of VAWT Deforming Blade Profile,” 2025, *GitHub*. [Online]. Available: <https://github.com/NaclTea/PhD-Example-Files/>
- [85] X. Li, M. J. Garzarán, and D. Padua, “Optimizing sorting with genetic algorithms,” *Proc. 2005 Int. Symp. Code Gener. Optim. CGO 2005*, vol. 2005, pp. 99–110, 2005, doi: 10.1109/cgo.2005.24.
- [86] “10 real-life applications of Genetic Optimization,” Mar. 22, 2022. [Online]. Available: <https://analyticsindiamag.com/10-real-life-applications-of-genetic-optimization/>
- [87] C. Darwin, *On the origin of species by means of natural selection, or preservation of favoured races in the struggle for life*. London : John Murray, 1859, 1859. [Online]. Available: <https://search.library.wisc.edu/catalog/9934839413602122>
- [88] H. Spencer, *The Principles of Biology*, no. v. 1. in System of synthetic philosophy. Williams and Norgate, 1864. [Online]. Available: <https://books.google.co.uk/books?id=3yYCAAAAQAAJ>
- [89] M. Drela, “XFOIL: An Analysis and Design System for Low Reynolds Number Airfoils BT - Low Reynolds Number Aerodynamics,” T. J. Mueller, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 1989, pp. 1–12.

- [90] K. Källström, “Computational and Experimental Analysis of a Rotor in Hover Using CFD and XFOIL,” in *VFS Aeromechanics for Advanced Vertical Flight Technical Meeting, San Jose, CA, Jan. 25–27, 2022*, Moffett Field, CA: NASA Ames Research Center, 2022. [Online]. Available: https://rotorcrafter.arc.nasa.gov/Publications/files/Kristin_Kallstrom_Final_Paper_18-Jan-2022.pdf
- [91] J.-P. Marx, “PhD Example Files - Chapter 1: XFOIL Based GA,” 2025, *GitHub*. [Online]. Available: [https://github.com/NaclTea/PhD-Example-Files/tree/main/Chapter 1 - XFOIL Based GA](https://github.com/NaclTea/PhD-Example-Files/tree/main/Chapter%201%20-%20XFOIL%20Based%20GA)
- [92] J. Morgado, R. Vizinho, M. A. R. Silvestre, and J. C. Páscoa, “XFOIL vs CFD performance predictions for high lift low Reynolds number airfoils,” *Aerosp. Sci. Technol.*, vol. 52, pp. 207–214, 2016, doi: <https://doi.org/10.1016/j.ast.2016.02.031>.
- [93] M. Elsakka, “The Aerodynamics of Fixed and Variable Pitch Vertical Axis Wind Turbines,” 2020, [Online]. Available: <http://etheses.whiterose.ac.uk/27424/>
- [94] M. M. Elsakka, D. B. Ingham, L. Ma, and M. Pourkashanian, “Comparison of the computational fluid dynamics predictions of vertical axis wind turbine performance against detailed pressure measurements,” *Int. J. Renew. Energy Res.*, vol. 11, no. 1, pp. 276–293, 2021.
- [95] S. Jain, N. Sitaram, and S. Krishnaswamy, “Effect of Reynolds Number on Aerodynamics of Airfoil with Gurney Flap,” *Int. J. Rotating Mach.*, vol. 2015, pp. 1–10, 2015, doi: [10.1155/2015/628632](https://doi.org/10.1155/2015/628632).
- [96] J. R. Spreiter and P. J. Steffen, “Effect of Mach and Reynolds Numbers on Maximum Lift Coefficient,” 1946. [Online]. Available: <https://ntrs.nasa.gov/citations/19930084610>
- [97] M. Drela, “Xfoil_Sv.Pdf,” 1989. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-84010-4_1
- [98] M. Drela and H. Youngren, “XFOIL 6.9 User Primer,” 2001. [Online]. Available: https://web.mit.edu/drela/Public/web/xfoil/xfoil_doc.txt

- [99] M. Reid and J. Kozak, "Thin/Cambered/Reflexed Airfoil Development for Micro Air Vehicle Applications at Reynolds Numbers of 60,000 to 100,000," 2006, doi: 10.2514/6.2006-6508.
- [100] A. Bakar, K. Li, H. Liu, Z. Xu, M. Alessandrini, and D. Wen, "Multi-Objective Optimization of Low Reynolds Number Airfoil Using Convolutional Neural Network and Non-Dominated Sorting Genetic Algorithm," *Aerospace*, vol. 9, no. 1, p. 35, 2022, doi: 10.3390/aerospace9010035.
- [101] A. Arshad, L. B. Rodrigues, and I. M. López, "Design Optimization and Investigation of Aerodynamic Characteristics of Low Reynolds Number Airfoils," *Int. J. Aeronaut. Sp. Sci.*, vol. 22, no. 4, pp. 751–764, 2021, doi: 10.1007/s42405-021-00362-2.
- [102] C. Heteyi, I. Molnar, and F. Szlivka, "Comparing different CFD software with NACA 2412 airfoil," *Prog. Agric. Eng. Sci.*, vol. 16, no. 1, pp. 25–40, 2020, doi: 10.1556/446.2020.00004.
- [103] A. Fazylova, B. Tultayev, T. Iliev, I. Stoyanov, and I. Beloev, "Development of a Control Unit for the Angle of Attack of a Vertically Axial Wind Turbine," *Energies*, vol. 16, no. 13, p. 5202, 2023, doi: 10.3390/en16135202.
- [104] N. N. Davis *et al.*, "The Global Wind Atlas: A High-Resolution Dataset of Climatologies and Associated Web-Based Application," *Bull. Am. Meteorol. Soc.*, vol. 104, no. 8, pp. E1507–E1525, 2023, doi: <https://doi.org/10.1175/BAMS-D-21-0075.1>.
- [105] J. Winslow, H. Otsuka, B. Govindarajan, and I. Chopra, "Basic Understanding of Airfoil Characteristics at Low Reynolds Numbers (104–105)," *J. Aircr.*, vol. 55, no. 3, pp. 1050–1061, Dec. 2017, doi: 10.2514/1.C034415.
- [106] S. Worasinchai, G. Ingram, and R. Dominy, "A Low-Reynolds-Number, High-Angle-of-Attack Investigation of Wind Turbine Aerofoils," *Proc. Inst. Mech. Eng. Part a J. Power Energy*, vol. 225, no. 6, pp. 748–763, 2011, doi: 10.1177/0957650911405411.
- [107] C. E. Brunner, J. Kiefer, M. O. L. Hansen, and M. Hultmark, "Study of Reynolds

- number effects on the aerodynamics of a moderately thick airfoil using a high-pressure wind tunnel,” *Exp. Fluids*, vol. 62, no. 8, p. 178, 2021, doi: 10.1007/s00348-021-03267-8.
- [108] N. Rosado Hau, “Unsteady Aerodynamics of Vertical Axis Wind Turbines,” University of Sheffield, Jan. 2021. [Online]. Available: <https://etheses.whiterose.ac.uk/28359/>
- [109] G. Ren, J. Jin, J. Mei, X.-Y. Jia, J. Han, and Y. Wang, “An Improved Variational Mode Decomposition Method and Its Application in Diesel Engine Fault Diagnosis,” *J. Vibroengineering*, vol. 20, no. 6, pp. 2363–2378, 2018, doi: 10.21595/jve.2018.19694.
- [110] A. Rezaeiha, H. Montazeri, and B. Blocken, “Characterization of aerodynamic performance of vertical axis wind turbines: Impact of operational parameters,” *Energy Convers. Manag.*, vol. 169, pp. 45–77, 2018, doi: <https://doi.org/10.1016/j.enconman.2018.05.042>.
- [111] M. G. Adams, P. F. Beard, M. Stokes, F. Wallin, K. Chana, and T. Povey, “Effect of a Combined Hot-Streak and Swirl Profile on Cooled 1.5-Stage Turbine Aerodynamics: An Experimental and Computational Study,” *J. Turbomach.*, vol. 143, no. 2, 2021, doi: 10.1115/1.4049103.
- [112] I. A. Hamakhan and T. Korakianitis, “Aerodynamic performance effects of leading-edge geometry in gas-turbine blades,” *Appl. Energy*, vol. 87, no. 5, pp. 1591–1601, 2010, doi: <https://doi.org/10.1016/j.apenergy.2009.09.017>.
- [113] Q. Cheng, X. Liu, H. S. Ji, K. C. Kim, and B. Yang, “Aerodynamic Analysis of a Helical Vertical Axis Wind Turbine,” 2017. doi: 10.3390/en10040575.
- [114] P. B. S. Lissaman, “Low-Reynolds-number airfoils,” *Annu. Rev. Fluid Mech.*, vol. 15, no. 1, pp. 223–239, 1983.
- [115] P. D. Berger, R. E. Maurer, and G. B. Celli, “Introduction to Response-Surface Methodology,” in *Experimental Design: With Application in Management, Engineering, and the Sciences.*, Cham: Springer International Publishing, 2018, pp. 533–584. doi: 10.1007/978-3-319-64583-4_16.

- [116] H. R. Maier *et al.*, “Exploding the myths: An introduction to artificial neural networks for prediction and forecasting,” *Environ. Model. Softw.*, vol. 167, p. 105776, 2023, doi: <https://doi.org/10.1016/j.envsoft.2023.105776>.
- [117] C. M. Bishop, “Pattern recognition and machine learning,” *Springer google Sch.*, vol. 2, pp. 1122–1128, 2006.
- [118] G. Cloud, “What is Supervised Learning?” [Online]. Available: <https://cloud.google.com/discover/what-is-supervised-learning>
- [119] M. F. Mokbel, W. G. Aref, and I. Kamel, “Performance of Multi-Dimensional Space-Filling Curves,” pp. 149–154, 2002, doi: 10.1145/585147.585179.
- [120] S. Gupta, “Some New Methods for Constructing Block Designs Having Orthogonal Factorial Structure,” *J. R. Stat. Soc. Ser. B (Statistical Methodol.*, vol. 45, no. 2, pp. 297–307, 1983, doi: 10.1111/j.2517-6161.1983.tb01253.x.
- [121] M. D. McKay, R. J. Beckman, and W. J. Conover, “A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output From a Computer Code,” *Technometrics*, vol. 42, no. 1, p. 55, 2000, doi: 10.2307/1271432.
- [122] I. ANSYS, “ANSYS SpaceClaim 23R2 Software,” 2023. [Online]. Available: <https://www.ansys.com/products/3d-design/ansys-spaceclaim>
- [123] L. J. Ayton, J. Gill, and N. Peake, “The Importance of the Unsteady Kutta Condition When Modelling Gust–aerofoil Interaction,” *J. Sound Vib.*, vol. 378, pp. 28–37, 2016, doi: 10.1016/j.jsv.2016.05.036.
- [124] Y. T. Lee, K. Ramesh, and A. Gopalarathnam, “Effect of Rounded Trailing Edges on Unsteady Airfoil Loading at Low Reynolds Numbers,” 2024, doi: 10.21203/rs.3.rs-4611607/v1.
- [125] X. Zhang, W. Li, and H. Liu, “Numerical Simulation of the Effect of Relative Thickness on Aerodynamic Performance Improvement of Asymmetrical Blunt Trailing-Edge Modification,” *Renew. Energy*, vol. 80, pp. 489–497, 2015, doi: 10.1016/j.renene.2015.02.038.

- [126] H. Kim, K. T. K. S, L. S, Y. S. Baik, and L. G, “Numerical Analysis of Unsymmetric Flatbak Trailing Edge Airfoil to Reduce Turbomachinery Noise in Power Generation Cycle,” 2017, doi: 10.20944/preprints201705.0146.v1.
- [127] L. CFD TEAM, “What y+ should I use? Part 3 – Understanding impact of Y+ and number of prism layers on flow resolution | Computational Fluid Dynamics (CFD) Blog – LEAP Australia & New Zealand,” 2020. [Online]. Available: https://www.computationalfluiddynamics.com.au/y-plus_part3_understanding-impact-of-y-and-number-of-prism-layers-on-flow-resolution/
- [128] T. J. Baker, “Mesh generation: Art or science?,” *Prog. Aerosp. Sci.*, vol. 41, no. 1, pp. 29–63, 2005, doi: <https://doi.org/10.1016/j.paerosci.2005.02.002>.
- [129] H. K. Versteeg, *An introduction to computational fluid dynamics: the finite volume method*, 2nd ed. Pearson Education, 2007.
- [130] Y. Wang, S. Shen, G. Li, D. Huang, and Z. Zheng, “Investigation on aerodynamic performance of vertical axis wind turbine with different series airfoil shapes,” *Renew. Energy*, vol. 126, pp. 801–818, 2018, doi: 10.1016/j.renene.2018.02.095.
- [131] Z. Yang, F. L. Haan, H. Hu, and H. Ma, “An Experimental Investigation on the Flow Separation on a Low-Reynolds-Number Airfoil,” 2007, doi: 10.2514/6.2007-275.
- [132] M. J. Baines and K. W. Morton, *Numerical Methods for Fluid Dynamics*. Oxford University Press, 1994. doi: 10.1093/oso/9780198536963.001.0001.
- [133] A. Teimourian, D. Rohacs, K. Dimililer, H. Teimourian, M. Yildiz, and U. Kale, “Airfoil aerodynamic performance prediction using machine learning and surrogate modeling,” *Heliyon*, vol. 10, no. 8, p. e29377, 2024, doi: <https://doi.org/10.1016/j.heliyon.2024.e29377>.
- [134] J. N. Kutz, “Deep learning in fluid dynamics,” *J. Fluid Mech.*, vol. 814, pp. 1–4, 2017.
- [135] J. Rabault, M. Kuchta, A. Jensen, U. Réglade, and N. Cerardi, “Artificial neural

- networks trained through deep reinforcement learning discover control strategies for active flow control,” *J. Fluid Mech.*, vol. 865, pp. 281–302, 2019.
- [136] L. Buitinck *et al.*, “{API} design for machine learning software: experiences from the scikit-learn project,” in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013, pp. 108–122.
- [137] T. Developers, “TensorFlow,” Sep. 2024, *Zenodo*. doi: 10.5281/zenodo.13863481.
- [138] A. Paszke *et al.*, “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” in *Advances in Neural Information Processing Systems 32*, Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>
- [139] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
- [140] J. Gerretzen, E. Szymańska, J. J. Jansen, J. Bart, H. van Manen, E. R. van den Heuvel, and L. M. C. Buydens, “Simple and Effective Way for Data Preprocessing Selection Based on Design of Experiments,” *Anal. Chem.*, vol. 87, no. 24, pp. 12096–12103, 2015, doi: 10.1021/acs.analchem.5b02832.
- [141] J. Chen, “What is a Neural Network?,” 2022. [Online]. Available: <https://www.investopedia.com/terms/n/neuralnetwork.asp>
- [142] G. Wang, R. Wang, W. Kong, and J. Zhang, “Simulation of Retinal Ganglion Cell Response Using Fast Independent Component Analysis,” *Cogn. Neurodyn.*, vol. 12, no. 6, pp. 615–624, 2018, doi: 10.1007/s11571-018-9490-4.
- [143] G. Gorospe, M. Daigle, S. Sankararaman, C. S. Kulkarni, and E. Ng, “GPU Accelerated Prognostics,” *Annu. Conf. PHM Soc.*, vol. 9, no. 1, 2017, doi: 10.36001/phmconf.2017.v9i1.2437.
- [144] J. Wu, W. Shi, and B. N. Hong, “Dynamic Kernel/Device Mapping Strategies for GPU-Assisted HPC Systems,” pp. 96–113, 2013, doi: 10.1007/978-3-642-35867-8_6.

- [145] S. Yu, M. S. Pritchard, P. Ma, B. Singh, and S. J. Silva, “Two-Step Hyperparameter Optimization Method: Accelerating Hyperparameter Search by Using a Fraction of a Training Dataset,” 2023, doi: 10.48550/arxiv.2302.03845.
- [146] M. Thamsanqa, H. van Deventer, and A. S. Bosman, “Cauchy Loss Function: Robustness Under Gaussian and Cauchy Noise,” 2023, doi: 10.48550/arxiv.2302.07238.
- [147] M. Kim and P. Kang, “Text Embedding Augmentation Based on Retraining With Pseudo-Labeled Adversarial Embedding,” *Ieee Access*, vol. 10, pp. 8363–8376, 2022, doi: 10.1109/access.2022.3142843.
- [148] P. Sharpe and R. J. Hansman, “NeuralFoil: An Airfoil Aerodynamics Analysis Tool Using Physics-Informed Machine Learning,” 2023.