Multi-disciplinary Design Optimization of a Rotor for an Offshore Wind Turbine

A comparison of static and dynamic models

Mihir Mehta

TU Delft

Multi-disciplinary Design Optimization of a Rotor for an Offshore Wind Turbine

A comparison of static and dynamic models

by

Mihir Mehta

in partial fulfillment of the requirements for the degree of Master of Science in Sustainable Energy Technology at Delft University of Technology

Student number:	4731948	
Project duration:	October 2, 2018 – July 10, 2019	
Thesis committee:	Dr. ir. M. B. Zaayer	TU Delft, Supervisor
	Prof. dr. S. J. Watson	TU Delft, Chairman
	Dr. J. J. E. Teuwan	TU Delft

Cover image courtesy : Tristan Stedman - MHI Vestas Offshore Wind

Keywords : Multi-disciplinary Design Optimization, Offshore wind turbines, Wind turbine optimization, NREL FAST

The Greatest Threat To Our Planet Is The Belief That Someone Else Will Save It.
- Robert Swan

Acknowledgements

My thought of us millennials, being the last generation to do anything about the environment and climate change, led me into this masters course at TU Delft. The mandatory thesis project resonated with my interest as I wanted to master certain aspects of at least one energy technology before I graduate. This master thesis project helped me develop researcher instincts and made me critical of my own work. It is of utmost importance to thank everyone who was a part of this journey.

Foremost, the credit for the success of this thesis goes to my supervisor, Michiel. His wisdom and valuable feedback steered this thesis to the correct path. Also, his meticulous comments about my report have been of enormous help. Apart from this, Michiel's availability has always been the cherry on the cake. I would also like to thank Sebastian Moreno, the chief developer of WINDOW, who helped me with the integration of FAST into the framework. Special thanks to Sebastiaan Mulders for explaining the concepts of controller design, and Julie Teuwan, for helping me understand composites and its implementation in my work. I am also grateful to Raffaello for sharing his findings of the optimization setup, Tanuj for his timely advice, and Naunidh for helping me understand aero-elasticity.

Surviving in Delft would not have been possible without the company of my beloved friends. I owe a very important debt to my constants, Reshma and Neel, for always being there throughout the journey. Their encouragement and constant support cannot be expressed here on paper. Special thanks to Nilay for being a major helping hand and for always being jocular. I am also grateful to Sukanya and Husain for making me feel at home with their sibling-like friendship and annoying comments. A final thanks to Yash and my friends from VJTI, for the amazing dinner parties and intellectual discussions. What completed me during my time here at TU Delft was my dance crew. Following my passion to the fullest during my Master's, was something that I had never imagined and I'd always be thankful to everyone at Delftse Dance Crew for the same.

Lastly, this entire journey wouldn't have been possible without the emotional and financial support from my parents and family. A mere thanks to their blind faith and assurance in me, would not suffice.

Mihir Mehta
Delft, July 2019

Summary

Wind farms are one of the most complex engineering systems with multiple stakeholders, each responsible for a particular element of the wind farm. The conventional industrial approach of designing involves sequential optimization of each of these disciplines, leading to a sub-optimal design. To reduce the costs of offshore wind farms in the future, it is important that a cohesive approach is developed wherein the elements are designed to minimize the LCOE, while capturing the interactions between all the disciplines.

Multi-disciplinary Design Analysis and Optimization (MDAO) is one such technique that has been explored in literature and has resulted in lower LCOE values compared to the existing approach. An MDAO tool at a wind farm level includes models for each of the disciplines in a wind farm : *Wake aerodynamics, Turbine, Support structure, Cabling, Costs, etc.*. It is essential that the tool allows agility to the user wherein the user can decide the variables to be optimized, so that the tool can cater to all the stakeholders involved.

At first, the missing links in the existing wind farm level tools are studied and the existing framework of the tool being developed at TU Delft is then explained. This research deals with the turbine discipline of the wind farm, with a focus on rotor optimization. It is already explored in literature that designing a rotor from a wind farm perspective differs significantly from designing a rotor from a wind turbine perspective. Designing a rotor from a wind farm perspective captures all the inter-disciplinary influences and results in a completely different rotor design.

The turbine model that is used currently in the framework is a static model that ignores the dynamic effects, lacks an aero-servo-elastic coupling in the rotor and uses a factor of 1.5 to compensate for the same. This research focuses on optimizing the rotor, after incorporating a dynamic model for the rotor in the MDAO tool, and quantifies the influence of model choice on various rotor and other wind farm parameters. To achieve the same, FAST, an aero-servo-elastic model is integrated into WINDOW (the existing MDAO tool). Also, a gradient-based (SLSQP) and a gradient-free (GA) algorithm are tried out with both the models to evaluate the influence of algorithm choice on all the parameters.

All the possible model-algorithm configurations are tried out and compared. With respect to the algorithm, it is observed that GA results in a design with a lower LCOE value for both the models, which is attributed to its better design space search abilities, while SLSQP shows a high starting point dependency with respect to some variables and has a tendency of getting stuck at a local minimum. With respect to the model choice, a huge difference in the chord and twist distributions is observed. The low-fidelity static model overestimates the constraints compared to the high fidelity dynamic model, resulting in a blade design that is stiffer than necessary.

The design details for these four configurations are evaluated next. The *chord* distribution is highly dependent on the model choice and does not show any dependence on the optimizer choice. The *thickness* always reduces to its lower bound for all the four cases, as the optimizer tries to reduce the blade mass by lowering the *thickness* and compensates for the stiffness by changing the *chord*. The *twist* distribution shows a high dependence on the starting point being used for SLSQP. The values differ significantly for all the four model-algorithm configurations. The *fine pitch angle* (the angle at which the blade is pitched for optimal performance during partial load operation) shows no dependence on the model-algorithm configuration. The *tip speed ratio* shows a decreasing trend with the SLSQP, while the GA results in designs with a higher tip speed ratio.

As the major differences in the designs are attributed to the choice of model, the best designs with the static and dynamic model are compared, where it is confirmed that the static model results in a blade that is stiffer than necessary and is highly sensitive to the amplification factor used (1.5 in this case). The optimized designs are also checked for fatigue damage where it is observed that fatigue damage is well within limits and does not emerge as a design driver. Overall, using a dynamic model, coupled with a gradient-free algorithm is observed to have the best performance in terms of optimality and is recommended for other use cases related to rotor design, in an MDAO framework.

Contents

Abbreviations

AEP	Annual Energy Production
BEM	Blade Element Momentum
CFD	Computational Fluid Dynamics
DTU	Denmark Technical University
FAST	Fatigue, Aerodynamics, Structures and Turbulence
GL	Germanischer Lloyd
HAWC2	Horizontal Axis Wind Turbine Simulation Code 2nd Generation
IEC	International Electrotechnical Commission
LCOE	Levelized Cost of Electricity
MDAO	Multi-disciplinary Design Analysis and Optimization
NREL	National Renewable Energy Laboratory
NREL5MW	5 MW Definition turbine by NREL
RNA	Rotor Nacelle Assembly
TOPFARM	Design Tool for Optimization of Wind Farm Topology and Operation
WINDOW	Windfarm Integrated Design and Optimization Workflow
WISDEM	Wind Plant Integrated System Design and Engineering Model
XDSM	eXtended Design Structure Matrix
MDF	Multidisciplinary Design Feasible
IDF	Individual Design Feasible
ESP	Extract Structural Properties
SLSQP	Sequential Least Squares Quadratic Programming
GA	Genetic Algorithm
PID	Proportional-Integral-Derivative
DLC	Design Load Case
ETM	Extreme Turbulence Model
NTM	Normal Turbulence Model
EDC	Extreme Direction Change
EWS	Extreme Wind Shear

Nomenclature

Symbols

α	Ply angle	[°]
β	Blade twist angle	[°]
δ_{tip}	Tip deflection	[m]
ϵ	Strain in the fibres	[−]
η	Efficiency	[−]
γ	Safety factor	[−]
λ	Tip speed ratio	[−]
μ	Blade mass density	[kg/m]
ν_{12}	Poisson's ratio	[−]
ω	Rotational speed of the rotor	[rad/s]
$\omega_{\phi n}$	Natural frequency	[Hz]
ϕ	Wind inflow angle	[°]
ϕ_{elm}	Angle between blade element and elastic axes	[°]
σ	Bending stress	[Pa]
θ	Blade pitch angle below rated wind speed	[°]
θ_P	Collective blade pitch angle	[°]
ζ_{ϕ}	Damping ratio	[−]
a	Annuity factor	[−]
A_{rotor}	Area of the rotor	[m^2]
a_{scale}	Scale factor	[−]
C	Cost	[€]
c	Blade chord	[m]
C_D	Drag coefficient	[−]
C_L	Lift coefficient	[−]
C_P	Power coefficient	[−]
C_Q	Torque coefficient	[−]
C_T	Thrust coefficient	[−]
$d_{fatigue}$	Fatigue damage	[−]
d_{tower}	Tower diameter distribution	[m]
E	Modulus of Elasticity	[Pa]

E_L	Modulus of Elasticity in the longitudinal direction	[Pa]
E_T	Modulus of Elasticity in the transverse direction	[Pa]
EA	Axial Stiffness	[N]
EI	Bending Stiffness	[Nm^2]
G	Shear Modulus	[Pa]
GJ	Torsional stiffness	[Nm^2]
GK	Gain correction factor	[$-$]
I	Moment of inertia	[m^4]
I_{XX}	Moment of inertia about flapwise axis	[m^4]
I_{YY}	Moment of inertia about edgewise axis	[m^4]
k	Shape factor	[$-$]
K_I	Integral gain	[$-$]
k_{opt}	Optimal mode gain	[Nms^2/rad^2]
K_P	Proportional gain	[$-$]
M	Bending moment	[Nm]
m_{rna}	Mass of the rotor nacelle assembly	[kg]
N_t	Number of turbines	[$-$]
P_{rated}	Rated power of the turbine	[W]
r	Radial distance along the blade	[m]
r_g	Gearbox ratio	[$-$]
R_{rotor}	Radius of the rotor	[m]
R_{yaw}	Radius of the yaw bearing	[m]
T	Torque	[Nm]
t	Thickness	[m]
T_{rotor}	Rotor thrust	[N]
t_{tower}	Tower thickness distribution	[m]
U	Wind speed	[m/s]
U_{rated}	Rated wind speed	[m/s]
UCS	Ultimate compressive strength	[Pa]
UTS	Ultimate tensile strength	[Pa]
V_{res}	Resultant wind velocity	[m/s]
y	Maximum distance from neutral axis	[m]

Subscripts

avg	Average
cc	Cabling
$decom$	Decommissioning
dt	Drivetrain
$edge$	Edgewise
elm	Element at a blade cross section
eq	Equivalent
$flap$	Flapwise
hss	High speed shaft
inv	Total investment costs
lss	Low speed shaft
max	Maximum
$O\&M$	Operations and Maintenance
opt	Optimum
ref	NREL 5MW reference turbine
ss	Support structure
st	Simulation time
t	Total
$te-reinf$	Trailing edge reinforcement
$trans$	Transmission

1

Introduction

1.1. Overview

Recent studies and trends show a rapid drop in the cost and drastic performance improvements in offshore wind, resulting in a massive capacity growth. According to the International Renewable Energy Agency [1], the global averaged Levelized Cost of Electricity (LCOE) reduced from USD 0.17/KWh to USD 0.14/KWh, while the projects to be commissioned in 2020-2022 in Europe and North America are in the range of 0.06 USD/KWh to 0.10 USD/KWh, as shown in Figure 1.1.

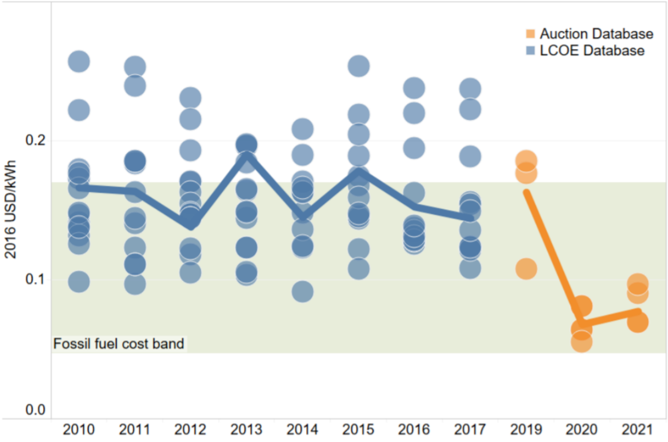


Figure 1.1: Global levelised cost of electricity from offshore wind farms by year of commissioning[1].

The drop in cost can be attributed to the industry becoming more mature, the increased turbine sizes (that help reduce support structure costs) and a streamlined project development process [1]. Europe, being the major contributor in offshore wind, now has a total installed capacity of 18,499 MW, where 2,649 MW was added in 2018 alone. The average turbine size of the turbines installed in 2018 was 6.8 MW, a 15% increase from 2017 [2].

For offshore wind energy to be viable, the design of a wind farm is as crucial as the design of a wind turbine. While wind turbine design alone can be challenging, one can imagine the complexity of designing a wind farm where multiple stakeholders are involved. The different stages of wind farm design are explained in [3], where the first phase includes choice of site for tendering by national authorities followed by layout design, cabling design, operation & maintenance strategy and installation optimization, done by the wind project developer.

1.2. Background information

Designing offshore wind farms can be quite complex due to the involvement of different aspects like the behaviour of atmosphere and water, complex seabed, wake interaction of turbines, energy production and loads, support structure design, layout optimization and electrical grid infrastructure [4]. The design of a wind turbine itself involves various disciplines such as aerodynamics, structural design, materials engineering, controller design, control systems, generator design and tower design. Different codes with varying fidelity for individual disciplines exist in literature. For instance, Fatigue, Aerodynamics, Structures and Turbulence (FAST) [5], Horizontal Axis Wind Turbine Simulation Code 2nd Generation (HAWC2) [6] and BLADED [7] are few state of the art aero-elastic codes that accurately couple the aerodynamics and structures of the rotor. A lot of work in the field of wind turbine optimization using these tools has already been done. Fuglsang *et al.* [8] optimized the rotor and tower for minimum cost of energy and for specific site conditions, using existing aero-elastic tools. Bottasso [9] used an aero-servo-elastic tool with multi-body dynamic analysis for different use cases: To optimize the turbine for maximum annual energy yield, minimize the cost of energy and optimize the internal layup, taking into account effects of bend-twist coupling. Ashuri [10] used an integrated multi-disciplinary optimization approach to optimize the rotor and tower, to minimize the turbine LCOE. Ashuri *et al.* [11] performed a complete aero-servo-elastic optimization of the wind turbine, wherein the effect of different controller parameters like generator slip, pitch rate on the LCOE was analyzed.

Currently, there are a large number of stakeholders involved, each responsible for the design of a particular system. This has made the industry partitioned in nature, which leads to a sequential design optimization that lacks the interaction between these different disciplines, leading to a sub-optimal design [12]. Multidisciplinary Design Analysis and Optimization (MDAO) is a method that involves coupling of various disciplines in an automated optimization loop where the performance of the entire system is analysed. An MDAO could help better integration of wind turbine design and wind farm design. Moreno *et al.* [12] demonstrate the superiority of MDAO over traditional sequential optimization, wherein the MDAO technique results in a lower LCOE value as compared to the sequentially optimized wind farm.

1.3. Existing tools and problem analysis

Denmark Technical University (DTU) and National Renewable Energy Laboratory (NREL) have been spearheading the research in the field of MDAO by developing tools like Design Tool for Optimization of Wind Farm Topology and Operation (TOPFARM) [13] and Wind Plant Integrated System Design and Engineering Model (WISDEM) [14], respectively. This section provides an elaborate description of the two tools and possible drawbacks, laying the foundation for the tool being developed at Delft University of Technology.

1.3.1. TOPFARM

TOPFARM is a wind farm topology optimization tool developed by DTU, built to get the optimum economic output which also takes into consideration, the costs related to component fatigue lifetime consumption. As the power production and loading patterns of a wind turbine differ when placed in a wind farm, TOPFARM includes high-fidelity instationary flow models (Dynamic wake meandering model) with dynamic effects to model the wind flow field, HAWC2 as the aero-elastic tool for load and power prediction and dedicated cost and control models in an optimization framework [13].

To include the effects of wake-deficits, added wake turbulence by upstream turbines and wake meandering, Reynolds-averaged Navier–Stokes based Computational Fluid Dynamics (CFD) models, together with experimental evidence were used to formulate and verify simplified CFD models. This flow field was fed to aero-elastic codes like HAWC2 and Bladed to get detailed information on wind turbine fatigue loads and power production. To estimate the component degradation cost, the aero-elastic codes were used to determine the lifetime equivalent fatigue loads of different wind turbine components [15]. A database of all generic load cases was made with different wind speeds, ambient turbulence intensity, distance from upstream wind turbine and different azimuth angles. Over 7436 simulations, each of 10 minutes and 6 different seeds, were carried out by combining HAWC2 and the dynamic wake meandering model, with the 5 MW UPWIND turbine [15].

Few possible drawbacks of the given tool are listed below:

1. **Feasibility:** A possible drawback of this tool would be that the generic load cases were determined for a given wind turbine. As the database is based on the UPWIND 5MW turbine, it is not directly applicable for wind farms consisting of other wind turbines. However, a first order approximation is suggested to adapt the current load and production set to different turbine types. The suggested scaling is valid for geometrically similar turbines equipped with a power and load control system comparable with that of the applied 5MW turbine. In such cases, the scaling can be simplified to depend on the rotor radius [15].

2. **Agility:** While the tool allows a variable fidelity approach, mainly in the wind climate discipline, it still lacks agility wherein the user can select a particular model for various disciplines to execute a particular use case [16]. For instance, the user cannot have a quick static model for turbine analysis. The objective of the tool in the end is to have an optimized farm layout where the main design variables translate to wind turbine positions and control strategy parameters.

3. **Computational time:** Most of the models used are high fidelity models which lead to a higher computational time.

1.3.2. WISDEM

WISDEM, developed by NREL is a multi-disciplinary analysis tool for assessing the overall plant cost of energy. The system engineering software framework includes model/workflow selection, analysis specification (optimization, sensitivity analysis), input specification for the turbine and site characteristics [17]. For the analysis of the rotor, it uses RotorSE, an aero-structural systems engineering model which includes CCBlade (a steady state aerodynamic model based on blade element momentum theory) and CurveFEM (a finite element method based model) for structural analysis of curved blades.

A possible drawback of WISDEM is that it lacks a proper time domain simulation with aero-servo-elastic coupling to accurately estimate the loads, resulting tip deflection and the fatigue damage of the blades.

WISDEM has a wrapper for FAST called AeroelasticSE but that uses a ‘template-based’ input files approach, wherein the user supplies with a working set of input files required to run FAST, which are then parsed into python dictionaries by the wrapper. The user can then modify the inputs based on the simulation requirements and the wrapper writes it back in terms of input files and WISDEM is called.

1.3.3. WINDOW

Given the need to develop a comprehensive system engineering tool for offshore wind farms that accurately captures the physics, can be tailored to cater to specific use cases and is computationally fast, TU Delft is making its own wind farm level optimization tool called ’Windfarm Integrated Design and Optimization Workflow (WINDOW)’. What is unique about this multi-discplinary analysis and optimization tool is its ability to tailor the workflow to suit a particular use case [16].

WINDOW offers feasibility in optimizing multiple parameters (at a wind farm or a wind turbine level), agility in the form of variable fidelity models and tailored use case according to the user, all of it at a low computational cost. However, similar to WISDEM, WINDOW relies on a static model in the existing framework and still lacks an aero-servo-elastic simulation of the turbine to predict extremes.

1.3.4. Problem statement

In the current WINDOW framework, the model used for the turbine analysis is a steady state model that fails to capture the essential dynamics involved. For instance, the existing aerodynamic model misses out on complex phenomena like dynamic stall, skewed inflow, etc., explained in the Aerodyn theory manual [18], which can have a significant impact on the loads. Also, the essential aero-elastic coupling needs to be taken into account, wherein the aerodynamic forces are affected by structural deflections and vice versa. Lastly, with the existing static model, the controller for the turbine cannot be modelled and the fatigue damage in the blades cannot be estimated.

The existing static model in WINDOW uses a safety factor of 1.5 to compensate for the dynamic effects, which could lead to an over or an underestimation of the extremes, resulting in a completely different design. Hence, there exists a need to integrate a computationally fast aero-servo-elastic model for turbine analysis in WINDOW and quantify the differences in the optimized rotor design resulting from the two models of different fidelities.

1.4. Research objective

In view of the problems discussed in Section 1.3.4, the main objective of this research is '*to provide a comprehensive insight into the consequences of having a static and a dynamic wind turbine model, on rotor optimization, using different optimization algorithms, in an MDAO tool for offshore wind farms.*'

1.5. Scope

In order to accomplish the main research objective, the following milestones are aimed at :

1. Selecting and integrating an aero-servo-elastic model into the existing wind farm tool (WINDOW).
2. Enhancing the capabilities of the tool in terms of pre and post-processing of turbine data.
3. Optimizing the rotor design in order to minimize the farm LCOE, using both gradient based and gradient free optimization algorithms.
4. Comparing and analyzing how the results (for e.g. LCOE or Blade internal layup) differ for the two models of different fidelities.

1.6. Report outline

This report starts with introducing the concept of MDAO and the existing framework of the wind farm optimization tool, in which the dynamic model of the turbine will be integrated. The process of model development and its integration is then elaborated. Finally, the optimization setup, problems faced and comparison of results obtained from static and dynamic models are discussed.

Chapter 2 : Provides the reader with a fundamental background of MDAO and a detailed eXtended Design Structure Matrix (XDSM) of the existing WINDOW framework.

Chapter 3 : Discusses the key changes required in the existing framework with respect to the use case.

Chapter 4 : Elaborates the entire model development procedure. Key elements of this chapter are as follows :

1. Tool selection to simulate the dynamic effects of the wind turbine.
2. Detailed explanations of the custom made pre-processing module used to extract blade mass and stiffness properties.
3. Detailed explanations of the Dynamics block, that prepares the template files required to run an aero-servo-elastic model, and post-process its output.

Chapter 5 : Explains the complete optimization setup and the challenges faced.

Chapter 6 : Presents a quantitative comparison of the optimized rotor, resulting from different model-algorithm configurations.

Chapter 7 : Lists the key findings of the research and discusses the scope for future work to further enhance the capabilities of the tool.

2

WINDOW: A Systems Engineering Tool

This chapter familiarizes the reader with the basic concepts and terminology used in systems engineering, particularly in MDAO, as a part of Section 2.1. A brief of the existing framework of the MDAO tool developed at Delft University of Technology, WINDOW is given in Section 2.2.

2.1. Background

This section gives a basic idea about the concept of Systems Engineering and its application using MDAO.

2.1.1. Systems engineering

"A system is an integrated composite of people, products or processes that provides a capability to satisfy a stated need or objective" [19]. Systems engineering can have a lot of definitions where most of them exhibit these particular characteristics [19]:

1. A logical sequence of activities or processes that transform the needs of the user into system performance parameters.

2. An interdisciplinary and collaborative approach.

3. An integrated life-cycle balanced set of system solutions.

Systems engineering has been widely used in the aerospace industry over the years [20]. As wind energy systems are equally complex with large number of stakeholders, systems engineering can play a huge role in meeting the future design and development needs of the industry. Due to its holistic and inter-disciplinary approach, the entire wind farm technical system can be simulated in order to have an integrated solution for all its stake holders [20].

One of the widely used methodologies in systems engineering is MDAO, which relates to physical system design and has been used for many years in the aircraft industry [20].

2.1.2. MDAO

A lot of work has been done with respect to applications of MDAO in wind turbine design, as explained in Section 1.2. The same technique can be extended to a wind farm level to include other aspects like O & M costs, wake interactions , wind farm energy production, etc.

The basic elements of an MDAO framework have been elaborated by Moreno [21]. There are various tools involved in simulating a system, where each tool represents a given discipline and these coupled tools are known as an analysis block. A driver at the top controls the logical sequence of execution of the disciplines in this block. The workflow determines the flow of variables and data between the analysis block and the driver, which is given for a particular use case. The effectiveness of a particular workflow can be tested with a use case. For instance, optimization of the turbine to minimize wind farm LCOE is a use case.

Moreno [21] explains the three core aspects of an MDAO workflow: System scope, Model fidelity/Driver algorithm and MDAO architecture.

System scope

It is important to define the system scope or the number of disciplines involved in a particular workflow, as the interaction between the disciplines might differ based on what the use case is. An example of this would be: For the optimization of the wind farm layout, the system should include, along with other models, wake models and cabling layout models. However, if the use case is to calculate the sensitivity of the wind farm LCOE with respect to the support structure design, it could be unnecessary to have the wake models and cabling models running in the loop every time, as the interaction between these modules and the support structure design could be negligible, which translates to not having discipline C in the loop, with respect to Figure 2.1. In this case, it can be run once, and the same value can be used in every function evaluation while checking the sensitivity of the LCOE with respect to the support structure. MDAO is hence a powerful tool that can also help in determining if there are significant interactions between different modules or not.

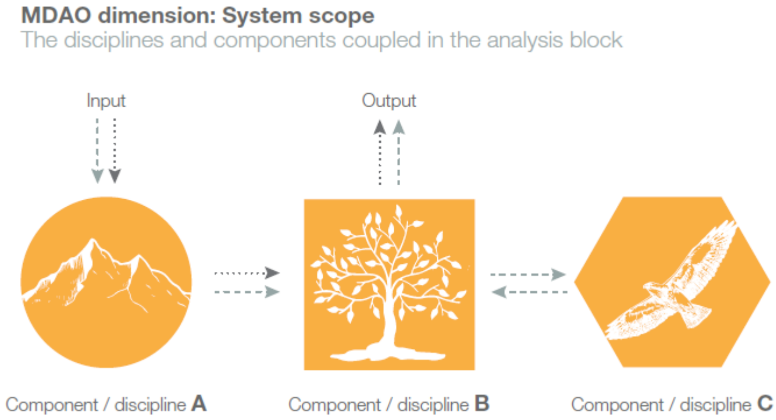

MDAO dimension: System scope
The disciplines and components coupled in the analysis block
Input
Output
Component / discipline A
Component / discipline B
Component / discipline C

Figure 2.1: System scope for two different use cases; Dotted arrows includes disciplines A and B, dashed includes all three [21]

Model fidelity/Driver algorithm

Model fidelity for each discipline is a key consideration as it determines the accuracy of the results, weighed against the computational costs. The use of a simple low-fidelity tool or a sophisticated high-fidelity tool depends on the use case. Tanmay [16] portrays a nice example of model fidelity for two different use cases, as shown in Figure 2.2. In the first use case, a steady blade element momentum model and a simple structural solver would work for a preliminary estimation of the rotor mass. However, the same would not be true for the second use case, for which, a high-fidelity aero-servo-elastic tool would be necessary to capture the system dynamics and hence, aid in controller design. An absurd example of model fidelity selection would be if one would use the solution of the full Navier-Stokes equation for the first use case, which would capture irrelevant details and result in an unnecessary increase in computational costs.

MDAO architecture

MDAO architecture is concerned with the coupling between different disciplines and between the disciplines and the driver, which is partly responsible for the performance of a particular MDAO workflow [21]. For instance, in blade optimization, aerodynamics and structures need a tight coupling, as they depend on each other. The blade outer parameters like chord and twist define its aerodynamic performance while the blade internal layup determine its ability to withstand the loads resulting from different load cases. Few of the MDAO architectures include Multidisciplinary Design Feasible (MDF) and Individual Design Feasible (IDF). In MDF, a consistent solution is passed on in every iteration of the optimization process. In IDF, there is no interaction between different disciplines while additional constraints and guess variables are added, making sure that the solution is consistent by the end of the complete optimization process.

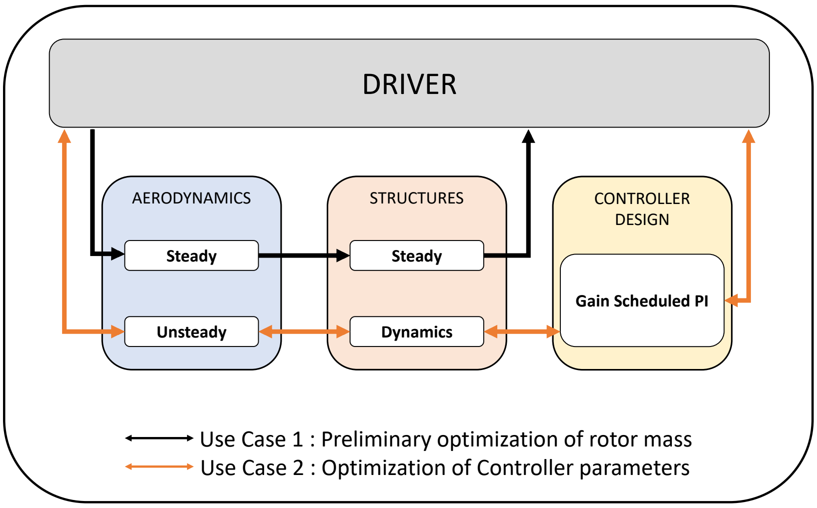

DRIVER
AERODYNAMICS
Steady
Unsteady
STRUCTURES
Steady
Dynamics
CONTROLLER DESIGN
Gain Scheduled PI
Use Case 1 : Preliminary optimization of rotor mass
Use Case 2 : Optimization of Controller parameters

Figure 2.2: Model workflow for two use cases with respect to rotor design [16]

2.2. WINDOW explained

The interactions between various disciplines in a multi-disciplinary optimization problem can be well illustrated in an XDSM. The XDSM for the existing model of WINDOW with the coupled static RNA model is shown in Figure 2.3. Each XDSM explains the workflow for a particular use case and the workflow has to be tailored according to the use case definition.

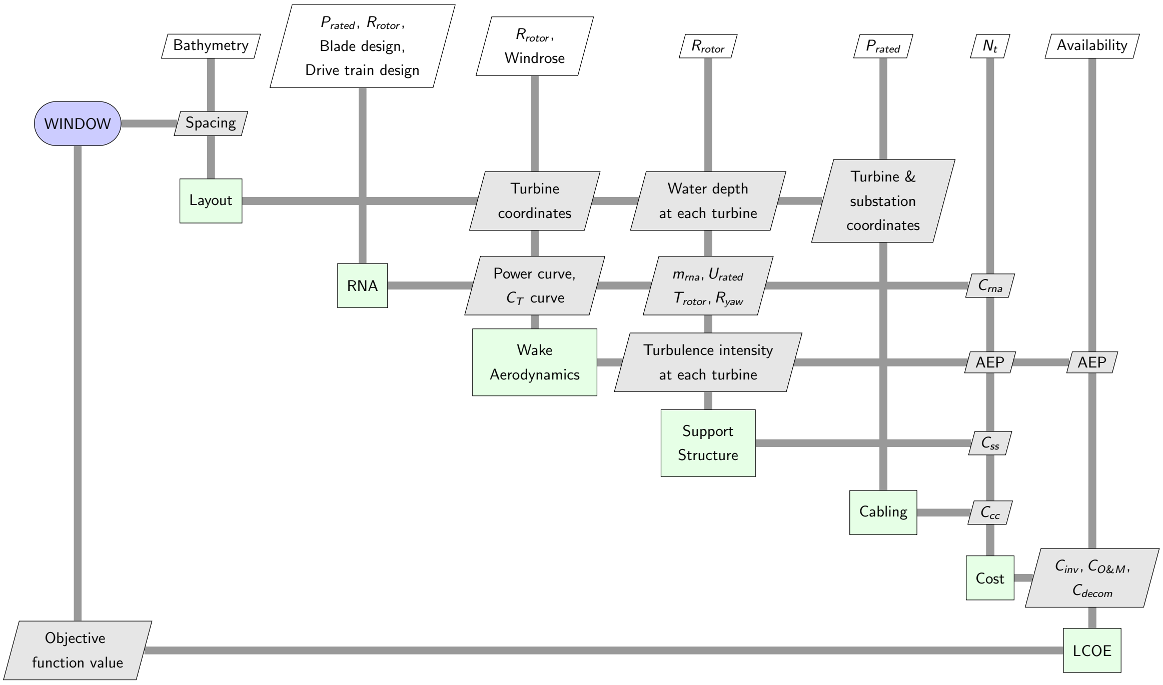


Figure 2.3: XDSM of the existing WINDOW framework [16]

The blocks in green along the diagonal represent the various disciplines involved in the given MDAO problem. The first row consists of various fixed parameters that are given as an input by the user, that remain constant throughout the optimization process.

The second row consists of all the design variables. In this simplified XDSM, the downwind and crosswind spacing between the turbines in a wind farm was optimized in order to minimize the LCOE. The grey boxes represent the connecting variables between different modules. The vertical lines and the horizontal lines represent the flow of inputs and the connections between different disciplines, respectively.

The Layout discipline takes in the spacing variables given out by the optimizer and sets the co-ordinates of the turbine and the substation. It uses the bathymetry data to calculate the water depth at each turbine location for the given site.

The Rotor Nacelle Assembly (RNA) module, which was recently integrated into WINDOW by Tanmay [16], gives an added flexibility of optimizing turbine parameters like the rated power (P_{rated}), rotor radius (R_{rotor}), blade design and drivetrain design parameters. RNA's interaction with the other modules (Wake Aerodynamics, Support Structure and Cost) is shown in the grey boxes.

The Wake Aerodynamics module uses the windrose data and samples it into discrete wind speed and direction bins. For each of these samples, the power and thrust co-efficient of the turbine is interpolated and the wind speed deficit due to all the neighbouring turbines is determined. A wake merger module then calculates the overall wind speed deficit and the corresponding power produced. It is then multiplied by the probability of the bin and summed over all samples to get the total annual energy yield.

The Support Structure module uses the RNA mass (m_{rna}), yaw bearing radius (R_{yaw}), rated wind speed (V_{rated}) and the maximum thrust (T_{max}) at the rotor from the RNA module, bathymetry data and the turbulence intensity from the wake models to design the support structure.

The Cabling module uses the co-ordinates of the turbines and the sub-station to calculate the cable layout, length and its cost, based on Esau-Williams heuristic algorithm [21]. P_{rated} is used to select the cable type from a pre-defined database based on capacities.

The Cost module takes the cost of RNA, support structure and cabling, along with the number of turbines in the farm in order to give an aggregated investment costs. It also returns the O & M costs and decommissioning costs along with total investment costs, which is then used along with the annual energy production (multiplied with an availability factor) to give the wind farm LCOE, which is sent to the optimizer as the objective function value along with constraint values. The optimizer then generates a new set of design variables and the process continues.

Moreno [21] discusses the basic framework of WINDOW in detail, while Tanmay [16] elaborates the static RNA model.

3

New WINDOW Framework

The essential modifications required in the existing framework of WINDOW in order to run and analyse a dynamic simulation have been elaborated in this chapter. Section 3.1 describes the use case for which the rotor will be optimized. Section 3.2 gives a detailed description of the existing static RNA model. Section 3.3 lists down the key modifications to be made (in WINDOW) in order to run a dynamic simulation and post process it. Lastly, Section 3.3 gives a detailed XDSM of the new WINDOW framework.

3.1. Use case

As defined in Section 2.1.2, the effectiveness of a multi-disciplinary tool can be measured with a use case. The use case for which the tool will be tested in this research is based on the study conducted by Tanmay [16], wherein the author explored the benefits of systems engineering by analyzing the effect of system scope on rotor optimization, using the existing WINDOW framework with the static RNA model.

To achieve this, Tanmay [16] optimized the rotor with respect to certain design variables: Tip speed ratio, fine pitch angle (the angle at which the entire blade is pitched below the rated wind speed) along with the chord and twist distribution. The optimization was done with an increasing system scope as shown in Table 3.1, wherein the LCOE is normalized with respect to the highest value. It should be noted that even if the objective function was different for all the cases, the optimality of that particular scope was always measured in terms of the LCOE. The study shows how the conventional approach of designing the turbine for its maximum aerodynamic efficiency results in a higher LCOE while designing the turbine for the LCOE yields the lowest value. Also, simply increasing the system scope may not result in an increasing optimality. For instance, increasing the scope from *Blade* to *RNA*, resulted in an increase in LCOE. In the case of *RNA*, the optimizer tried to reduce the mass of the RNA (dominated by the nacelle mass), which was quite sensitive to torque and not to rotor thrust. As minimizing the torque then became the implicit objective and not the rotor thrust, the increase in thrust led to an increase in wake losses, support structure costs and hence, an overall increase in the LCOE.
However, the LCOE was found to be minimum in the last case with the entire *Wind farm* as the system scope. Hence, the use case that will be analyzed in this research, after the integration of the dynamic model, will be that of the entire wind farm as system scope and minimization of LCOE as the objective. In order to make a fair comparison, the use case and the design variables will be kept consistent for both, the static and the new dynamic model.

Table 3.1: System scopes for Use Case 1 of Tanmay [16]

System scope	Objective function	Normalized LCOE [-]
Aerodynamics	Maximize aerodynamic efficiency	1.0000
Blade	Maximize ratio of aerodynamic efficiency to blade mass	0.9363
RNA	Maximize ratio of aerodynamic efficiency to RNA mass	0.9552
Wind farm	**Minimize wind farm LCOE**	**0.9346**

The rotor design will be optimized with respect to the given design variables :

1. Blade internal layup (based on a reference layup)
2. Blade chord distribution (at 3 Blade nodes)
3. Blade twist distribution (at 3 Blade nodes)
4. Tip speed ratio
5. Fine pitch angle

These design variables can have a significant impact on the design of the rotor and either affect the objective function or the constraint evaluation. An impact on the objective function (LCOE) can either be in the form of overall costs or annual energy production, while the impact on the constraints can be in the form of out of plane tip deflection or spanwise stresses. For instance, the interaction of the blade internal layup with the objective function and constraints is shown in Figure 3.1.

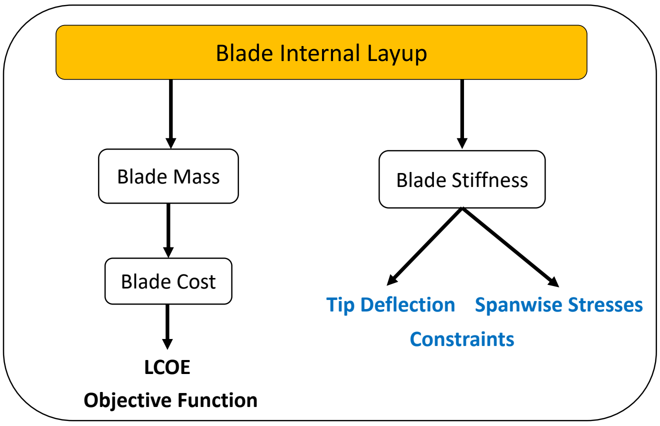


Figure 3.1: Interaction of blade internal layup with the objective function and constraints

The *blade internal layup*, which includes the placement and thickness distribution of different materials used has a direct implication on the mass of the blade that determines the cost of the blades and hence, the LCOE. Similarly, the layup also determines the spanwise flapwise and edgewise stiffness of the blade. The stiffness properties govern the tip deflection of the blade and bending stresses experienced by different materials at different sections of the blade that form the constraints in this particular optimization use case.

Having the thickness of different materials at multiple blade spanwise locations would result in a lot of design variables and ultimately, higher computational times. In order to circumvent this, a reference internal layup will be developed based on existing studies done on various 5 MW turbines. The thicknesses at multiple blade locations can then be uniformly scaled with respect to one factor, called the *thickness factor (τ)*. This thickness factor can then be used as a design variable to optimize the blade internal layup.

The *chord distribution* (c) directly affects the aerodynamic performance of the blade, the power coefficient and hence, the overall energy production. Also, blade chord has a direct influence on the thickness of the airfoil, the overall mass and stiffness and hence, on the tip deflection and stresses. Similarly, the *twist distribution* (β) directly affects the aerodynamic performance and hence, the total energy produced. Also, the twist has a direct influence on the angle of attack and hence, the loads.

The *fine pitch angle* (θ) is the angle at which the entire blade is pitched below the rated wind speed in order to achieve the ideal angle of attack for the respective airfoil throughout the blade. This fine pitch angle is needed separately, as in the existing model, the blade twist at the tip is set to zero, as a convention adopted for manufacturing ease. Similar to the twist distribution, the fine pitch angle also has an effect on the aerodynamic performance of the blade.

The *tip speed ratio* (λ) determines the rotational speed at which the rotor rotates, for a given wind speed. It directly affects the inflow conditions and hence, the power and thrust coefficients of the turbine. These coefficients have a direct impact on the power curve, farm wake losses and hence, the annual energy production. Also, it has a direct effect on the forces acting on the rotor and hence, the tip deflection and spanwise stresses.

With the integration of an aero-servo-elastic model, a better evaluation of constraints is expected in the form of tip deflection, spanwise stresses and fatigue damage, leading to a blade design which would be much more feasible and a better starting point for a detailed blade design for manufacturing purposes.

3.2. Existing RNA model

This section gives a detailed description of the existing static RNA model in WINDOW, developed by Tanmay [16].

3.2.1. Inputs

The existing static RNA model requires certain blade properties as an input to determine the loads and stresses acting on the rotor.

Blade properties
The existing model takes the blade chord and twist at 3 pegged nodes, as shown in Table 3.2.

Table 3.2: Pegged notes for chord and twist distribution in the static RNA model

Parameter	Blade sections
Chord	Root section, 70% and 90% of blade length
Twist	Transition section, 40% and 70% of blade length

For the chord profile, a linear region is considered between the root and the transition section [1], while another linear profile runs through the nodes at 70 and 90% of the blade span till the transition section. For the twist distribution, the twist at the tip is set to zero, which is also the reason why the fine pitch angle is introduced. The twist at the transition section is defined and held constant till the root section as the twist at the root has no meaning due to its rotational symmetry.

Figure 3.2 shows the spanwise chord and twist distribution obtained from the three respective pegged nodes.

The blade mass and stiffness properties are scaled from the NREL5MW reference turbine based on chord, using the equations listed below, where s is the chord scaling factor and τ is the thickness factor [16]. Also, with the existing model, no knowledge about the offsets of the center of mass, shear center and tension center is required.

$$s = \frac{c}{c_{ref}} \tag{3.1}$$

$$\mu(r) = \mu_{ref}(r) \cdot s \cdot \tau \tag{3.2}$$

[1] Transition section refers to the blade spanwise location where the blade cross section transitions from a circular shape to an airfoil shape

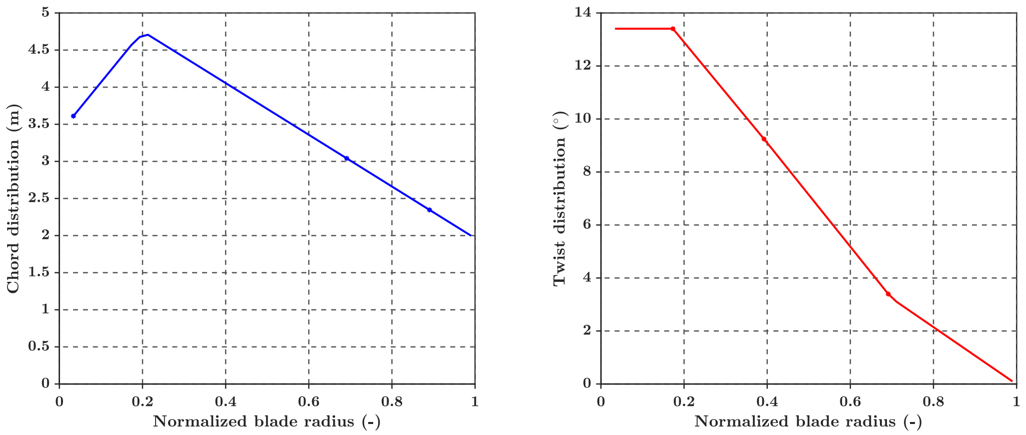


Figure 3.2: Complete chord and twist distribution obtained from pegged nodes

$$EI(r) = EI_{ref}(r) \cdot s \cdot \tau \tag{3.3}$$

Where μ is the mass density and EI is the stiffness.

3.2.2. Post-processing

The static moments and tip deflection are amplified by a factor of 1.5 to compensate for not modelling the dynamic effects.

Blade spanwise stress

The existing model assumes the blade to be characterized uniformly by glass fibre reinforced plastics with a Young's Modulus (E) of 36.23 GPa and an Ultimate Tensile Strength (UTS) of 400 MPa. The moment of inertia of the blade section ($I(r)$) is then calculated using Equation 3.4.

$$I(r) = \frac{EI(r)}{E} \tag{3.4}$$

This value of $I(r)$, along with the amplified moments, is then used in the bending equation to determine stresses at different blade spanwise locations. It should be noted that this approach would not yield accurate results as it assumes one constant value of E throughout the blade span and does not use any internal layup information. As a result, material specific stresses cannot be determined.

3.3. Aero-servo-elastic model integration requirements

It is important to understand the inputs needed to run an aero-servo-elastic (dynamic) model, and post-processing of the outputs required in order to extract meaningful information from it. This section lists down the model developments that will be done in order to meet the I/O requirements, and probable constraints in realizing the same. An aero-elastic simulation of a wind turbine captures the interaction between the aerodynamics and structural dynamics of the blade. However, an aero-servo-elastic simulation requires the additional element of controller design that also simulates the transients in the system due to the controller action. This mainly concerns the pitching action required above rated wind speeds to maintain a constant rotational speed, and hence, constant power. Unlike the existing static RNA model, running a time domain simulation using an aero-servo-elastic model requires tower properties, controller parameters and a wind field (with temporal and spatial variation) as an input, apart from rotor properties.

3.3.1. Inputs

The inputs required to run an aero-servo-elastic simulation are listed in this section.

Blade properties
The spanwise chord and twist distribution will be determined in the same way as in the existing RNA model. As an aero-servo-elastic model requires the blade offsets and mass Moment Of Inertia (M.O.I.), a module to determine the same will also be developed.

To determine the blade mass and stiffness properties, a pre-processing module is needed.

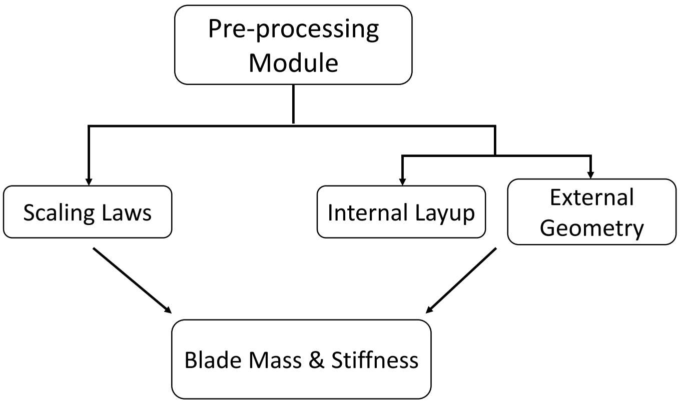


Figure 3.3: Different types of pre-processing modules to determine blade mass and stiffness

The pre-processing module can rely on scaling laws, as in the existing RNA model, or derive the blade properties using the external geometry and internal layup data, as shown in Figure 3.3. Having a pre-processing module based on an internal layup can be advantageous in many ways. At a particular blade spanwise section, as the thickness of the different materials and its location along the airfoil is known, *material specific stresses* can easily be determined. Also, the thicknesses of these materials can then be optimized by making them *design variables*, either individually, or in the form of a thickness factor that would scale values of thicknesses from a reference layup. Hence, in this research, the blade mass and stiffness properties will be derived based on an internal layup and external geometric data (chord, airfoil type and geometry), for which a reference layup will be developed.

Support structure properties
As the hydrodynamic loading will not be taken into account, only the tower properties will be used to run the aero-servo-elastic simulations. The tower wall thickness and diameter values will be used from the existing support structure design module in WINDOW. However, for the farm LCOE calculation, the costs of the monopile based support structure designed by the same design module will also be taken into account.

Controller parameters
To design a conventional PI controller for the pitching action above rated wind speeds, a full system linearization about the equilibrium points will be required. It should be noted that linearization at all wind speeds above rated could be a hindrance in the optimization process and few time reduction techniques will be devised for the same.

Wind field
The existing static RNA model calculates the moments and tip deflection at the rated wind speed. With the new aero-servo-elastic model, a time domain simulation with a turbulent wind field will be carried out. To achieve the same, a wind field generation model will have to be integrated.

Table 3.3 summarizes the specific input parameters required by an aero-servo-elastic tool and the corresponding modifications required in order to meet them.

Table 3.3: Modifications in WINDOW to incorporate the aero-servo-elastic tool

Input	Required Modifications
Blade spanwise Mass, Stiffness, Chord, Twist, Offsets, M.O.I.	Develop a reference internal layup Develop a module to calculate mass and stiffness properties based on an internal layup and external geometry Develop a module that determines blade offsets and M.O.I.
PI controller gains	Develop a module that performs full system linearization and analytically determines the controller gains
Support structure parameters	Connect the outputs of the existing support structure module to the tower inputs of the aero-servo-elastic model
Wind Field with temporal and spatial variation	Integrate a wind field generator into the existing framework

3.3.2. Post-processing

While the tip deflection output of an aero-servo-elastic simulation can be used as it is for the constraint evaluation, the moments have to be post-processed to calculate spanwise stresses and fatigue damage in the blade. It should be noted that with the existing static model, the fatigue damage in the blades cannot be evaluated. To estimate the fatigue damage, a time domain simulation with the normal turbulence model, at each wind speed is to be carried out and the stress ranges are then used to determine the fatigue damage.

Blade spanwise stress
To convert the spanwise moments to stresses in the blade, a post processing module will be developed that first calculates the strain in a particular material, at specific spots of a blade section. As the resultant Young's Modulus (E) of all the materials is known, the stresses in a particular material can be determined. Although the bending moments are maximum at the blade root, the stresses need not be highest at the root and hence, stresses at multiple blade spanwise locations will be determined. The largest stress value for each material will then be compared against the material specific tensile strength for constraint evaluation.

Fatigue damage
To estimate the fatigue damage of different materials in the blade, the traditional rainflow counting method will be used to determine the stress range along with the S-N curve data of that respective material to evaluate the fatigue damage. However, the same process is usually done at all wind speeds between cut-in and cut-out. As this would drastically increase the time for one function evaluation, the fatigue damage will not be evaluated every time and only the final optimized design will be checked for fatigue failure.

Table 3.4 summarizes the specific output parameters of an aero-servo-elastic model and the corresponding modifications required in order to convert them into constraints.

Table 3.4: Modifications in WINDOW to incorporate the aero-servo-elastic tool

Output	Required modifications
Blade spanwise bending moments	Develop a module to determine material specific spanwise stresses Develop a module to determine material specific fatigue damage based on rainflow counting

3.3.3. Load case

A time domain simulation of the wind turbine under a given set of operating conditions and environmental conditions defines one load case. In the process of certifying a wind turbine, multiple load cases with different operating and wind conditions are simulated for periods of 10 minutes. However, doing the same for a design optimization process would require unreasonably high computational power. Hence, the optimization with the dynamic model will be done for a particular load case that represents the worst case conditions for the blade tip deflection and moment values.

Also, time reduction techniques will be needed to reduce the overall time required for one function evaluation, where one function evaluation involves running all the disciplines of WINDOW.

3.4. XDSM : New WINDOW

To meet the research objective stated in Section 1.5 and the modifications required as per Section 3.3, two new modules will be developed and coupled to the existing WINDOW framework.

- Extract Structural Properties (ESP), to calculate the blade mass and stiffness properties.
- Dynamics, for other pre-processing required, running the aero-servo-elastic simulation and post-processing its outputs.

The XDSM of the new WINDOW framework can be seen in Figure 3.4. The custom made pre-processor module, ESP, will be integrated into the RNA block, and hence, is not visible in the XDSM. The interactions of the Dynamics block with the other disciplines and the optimizer can be clearly seen in the XDSM presented in Figure 3.4. The Dynamics block requires the blade spanwise properties, airfoil properties, drivetrain generator and gearbox efficiency values and several nacelle properties to prepare the input files to run the aero-servo-elastic model and post-process its outputs to give the tip deflection, spanwise stresses and fatigue damage. The Dynamics module consists of four sub-modules, based on their functionalities: *PrepSim, Controls, FAST, Post-processor*. The functions of all the newly added blocks, followed by the module/sub-module name in brackets, are elaborated below.

Integration of the ESP and Dynamics into WINDOW will now enable the tool to :

1. Estimate blade spanwise mass and stiffness properties, and also enable the inclusion of the blade internal layup as design variables. (***ESP***)
2. Determine other blade properties required to run an aero-servo-elastic simulation (blade offsets, pitch axis location, MOI, etc.), along with tower cross sectional stiffness properties. (***PrepSim***)
3. Automate the design of a PI controller that would be required to run an aero-servo-elastic simulation. Thus, it also enables the user to have controller parameters as design variables and analyse the effects at a wind farm level. (***Controls***)
4. Run an aero-servo-elastic simulation for different International Electrotechnical Commission (IEC) load cases. (***FAST***)
5. Accurately estimate the tip deflection and spanwise stresses, for different materials, at multiple blade spanwise locations. (***Post-processor***)
6. Estimate fatigue damage for different materials in the blade, as each material would have a different slope on the S-N curve. (***Post-processor***)

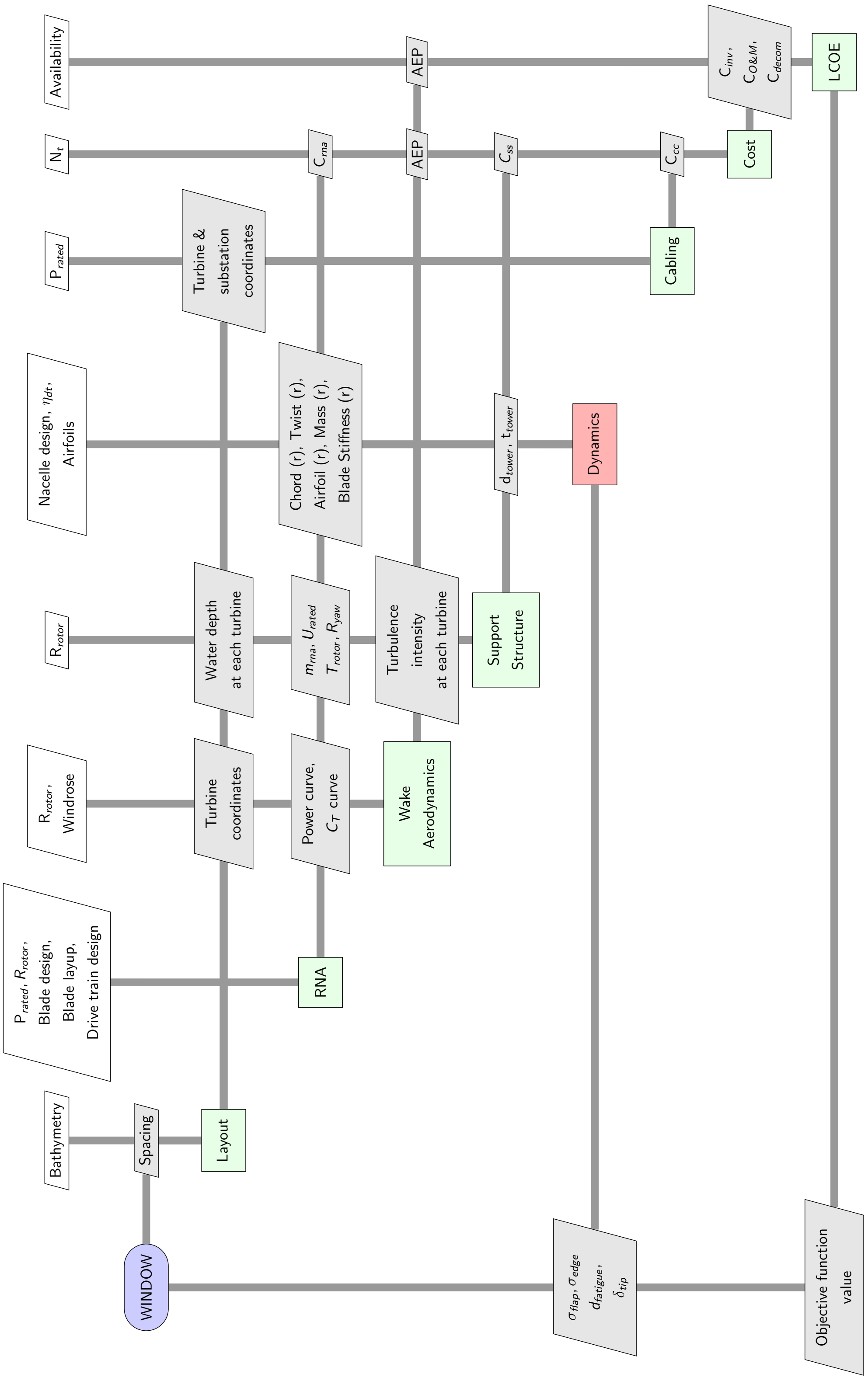

WINDOW
Bathymetry
Spacing
Layout
P_{rated}, R_{rotor}, Blade design, Blade layup, Drive train design
RNA
R_{rotor}, Windrose
Turbine coordinates
Power curve, C_T curve
Wake Aerodynamics
R_{rotor}
Water depth at each turbine
m_{rna}, U_{rated} T_{rotor}, R_{yaw}
Turbulence intensity at each turbine
Support Structure
Nacelle design, η_{dt}, Airfoils
Chord (r), Twist (r), Airfoil (r), Mass (r), Blade Stiffness (r)
d_{tower}, t_{tower}
Dynamics
P_{rated}
Turbine & substation coordinates
Cabling
N_t
C_{rna}
AEP
C_{ss}
C_{cc}
Cost
Availability
AEP
C_{inv}, $C_{O\&M}$, C_{decom}
LCOE
σ_{flap}, σ_{edge} $d_{fatigue}$, δ_{tip}
Objective function value

Figure 3.4: XDSM of the New WINDOW Framework

4 Model Development

The entire model development process can be broken down into 2 salient modules: ESP, which will be integrated into the RNA block, and the Dynamics block. The pre-processor module, and the Dynamics block, highlighted in Figure 3.4, have been elucidated in this chapter. This chapter discusses various existing tools related to different disciplines of a wind turbine, tool selection and then about its integration into the Dynamics block.

4.1. Existing tools

A crucial part of this research is the selection of the aero-servo-elastic model, along with other coupling models, to be integrated into the Dynamics module. Various models available in literature are explored, as shown in Table 4.1.

Table 4.1: Existing tools evaluated for model development

Discipline	Tool	Description	Assessment
Wind field	CFD based models [13]	Dynamic wake meandering models; Calibrated CFD models to construct simpler CFD models	High fidelity tool; High computational time
	TurbSim [22]	Stochastic, full field turbulent wind simulator	Medium-fidelity tool; Can be coupled to other aerodynamic modules
Structural properties	BECAS [23]	Based on a finite element method	Highly accurate; High computational time
	Pre-Comp [24]	Based on classical laminate theory and shear flow approach	Highly accurate; Low computational time
	ESP	Based on a simple weighting method to determine equivalent cross sectional properties	Fairly accurate; Low computational time
Aero-servo-elastic coupling	HAWC2 [6]	Uses BEM with all dynamic effects included; Structural module based on Multi-body formulation	High fidelity model; Highly accurate; High computational time
	FAST [5]	Uses BEM with few dynamic effects included; Structural module based on modal dynamics with no bend-twist coupling	Medium/High fidelity model; Fairly accurate; Low computational time

As explained in Chapter 3, an external wind field generator is required to simulate a load case using the aero-servo-elastic model. CFD based models for the wind climate, coupled with HAWC2, are used in TOPFARM. While CFD based models are computationally expensive, TurbSim can quickly generate the flow field that can directly be read by any aero-servo-elastic model. Also, a pre-processing model is required to extract blade properties used as an input by the aero-servo-elastic model. BECAS is used to calculate the blade structural properties and generate an input file that can directly be read by HAWC2. Pre-Comp is used to determine the blade structural properties that can be read by BModes, a module that determines coupled mode shapes required as an input by FAST. The structural properties can be extracted by using other simplified approaches as well. The two aero-sevo-elastic models considered for the research are HAWC2 and FAST.

4.2. Tool selection

To accurately simulate a wind field and yet be computationally fast, TurbSim is selected as the wind field generator. Also, as mentioned previously in Chapter 3, ESP, a custom module will be developed for extracting the blade structural properties.

While making the selection for the aero-servo-elastic model to be integrated into WINDOW, few selection criteria considered are explained below:

- Accurately captures the system dynamics
- Computationally fast and accurate

Few advantages that FAST exhibits over HAWC2 are :

- **Modularity**: Available as a Simulink block; Provides for input of inflow generation in multiple formats; Design of controller can be performed in Simulink or written as a seperate fortran routine.
- **Flexibility**: Other NREL tools can be easily coupled to FAST or can be replaced by simple analytical methods to generate input files.
- Free and open source in nature.

Apart from the reasons mentioned above, the Wind energy group at Delft University of Technology has built an interface on MATLAB using the Simulink version of FAST. This prior experience would contribute greatly towards expediting the research process.

Owing to the reasons stated above, **FAST** developed by NREL, will be integrated into the Dynamics block of WINDOW.

4.2.1. FAST

FAST is an open source multi-physics simulation tool built by NREL to analyse offshore and land based horizontal axis wind turbines. Figure 4.1 shows the coupling between the FAST driver and its different modules. The modules in light gray are the modules that are used for this research.

FAST comprises of the modules on the left that are tightly coupled to the main FAST driver. While FAST is capable of simulating almost all the external effects on wind turbines, many are not considered for this research. For instance, the HydroDyn module is not used as the hydrodynamics aspect will not be treated. There are various other modules that haven't been listed in the diagram but not been taken into account. To give an example, IceFlow (to simulate ice dynamics) is not shown in Figure 4.1. BeamDyn is a high fidelity structural dynamics module in FAST that requires a full 6 x 6 cross-sectional mass and stiffness matrix for the blade. In order to make BeamDyn work, an additional cross sectional analysis tool would be needed. Also, BeamDyn needs a really fine discrete time step in order to run and the capability of having a full system linearization with BeamDyn is not possible on FAST v8. As a result, the medium fidelity structural dynamics tool, ElastoDyn is used. AeroDyn is the module in FAST for aerodynamics, based on BEM, and ElastoDyn is the module in FAST for structural dynamics, based on modal dynamics. The InflowWind module in FAST reads the input wind file, which could be a steady wind field or a turbulent wind field generated by an external tool. ServoDyn is used to simulate the control and drivetrain dynamics in FAST.

The modules on the right are used to generate data that is eventually read by the modules in FAST. TurbSim is a tool developed by NREL, used to generate a turbulent wind field which can be easily read by InflowWind, making it a suitable choice. BModes is another module developed by NREL, that determines the rotating beam coupled modes for both, the blade and the tower [25] and requires their spanwise properties as an input. Elastodyn reads these mode shapes as an input and uses them to simulate the structural dynamics. To generate the spanwise stiffness and mass properties of the blade, a cross sectional analysis tool called PreComp, developed by NREL, is often used. However, PreComp requires a detailed internal layup definition (the number of plies and its orientation for each material) and generates an output file that can be directly read by BModes. A model that would better suit the given research purpose would require a more general internal layup definition (in terms of thicknesses of different materials) and also provide some data that would be used for post processing purposes. To meet these I/O requirements, a custom module called ESP will be developed.

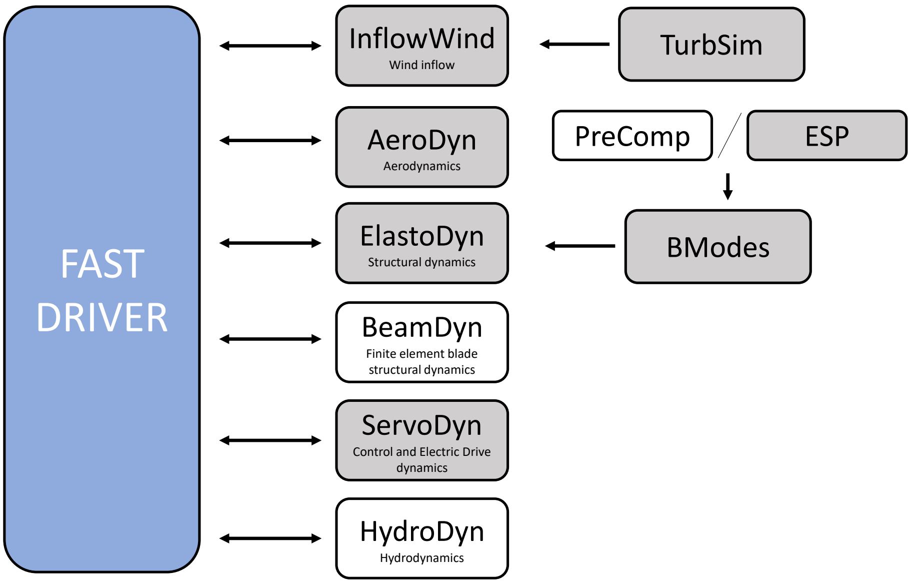


Figure 4.1: Coupling between FAST driver and its modules

4.2.2. FAST: A validated tool

A validation study of FAST against experimental data from the Siemens 2.3 MW turbine and the code BHawC used by Siemens, is presented by Guntur *et al.* [26]. It is observed that FAST (with BeamDyn) shows excellent agreement with BHawc and the experimental data. Also, the predictions from FAST (with BeamDyn) are compared with FAST (With ElastoDyn), where it is seen that the results using ElastoDyn follow the same trend as BeamDyn, with a small deviation. From the analysis presented by Guntur *et al.* [26], it can be concluded that FAST (with BeamDyn) accurately predicts real life blade conditions (tip deflection, root moments, etc.) and FAST (with ElastoDyn) shows reasonable agreement with (with BeamDyn). However, due to the various constraints mentioned in Section 4.2.1, FAST (with ElastoDyn) is chosen for the optimization process in this research.

4.3. ESP

As mentioned in Section 3.1, in this research, the blade internal layup will also be a part of the optimization process. To translate this internal layup and external blade geometry into blade spanwise mass and stiffness properties, required by the aero-servo-elastic tool (FAST), as stated in Section 3.3, the pre-processing module called ESP is developed. Also, as stated in Section 3.1, having all the thicknesses for different materials at variable blade span locations would drastically increase the number of design variables. To circumvent this, a reference layup will be developed and the *thickness factor* (τ), that can uniformly scale this reference layup, will be used as a design variable. Section 4.3.1 describes the reference layup that will be used for the optimization process. Section 4.3.2 elucidates the model used to convert the internal layup and external blade geometry into spanwise mass and stiffness properties.

An XDSM of the Pre-processor module can be seen in 4.2, where ESP is the custom pre-processing module.

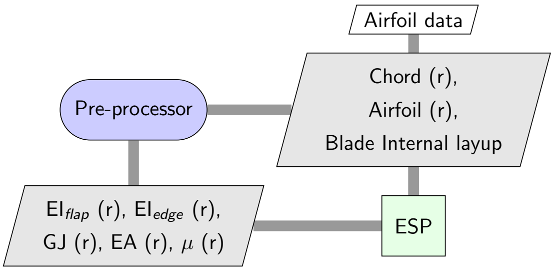


Figure 4.2: XDSM of the Pre-processor module

The spanwise distribution of mass and stiffness properties can be derived using *ESP* that uses few traits of the classical laminate theory and is based on the methodology explained by Ashuri *et al.* [27] and Wang *et al.* [28].

4.3.1. Reference blade internal layup

The reference blade internal layup is defined based on the studies carried out by SANDIA labs [29] [30] and the UPWIND 5 MW layup [31]. Table 4.2 lists down the terminology that will be used and its interpretation in the given context.

Table 4.2: Blade layup terminology

Term	Interpretation
UD	Uni-directional
LE	Leading edge
TE	Trailing edge
Blade spanwise location	Position along the length of the Blade
Blade segment	Position within a particular Blade section

Table 4.3 gives a summary of all the material properties that are used for this research, same as that of the study conducted by SANDIA labs[29][30].

Table 4.3: Summary of material properties [29][30]

Stack ID	Stack Name	Material	E_L (MPa)	E_T (MPa)	ρ (kg/m^3)	UTS (MPa)	UCS (MPa)
1	Gelcoat	Gelcoat	3440		1235		
2	Triax Skins	SNL (Triax)	27700	13650	1850	700	
3	Triax Root	SNL (Triax)	27700	13650	1850	700	
4	UD carbon	Carbon (UD)	114500	8390	1220	1546	1047
5	UD Glass TE	E-LT-5500 (UD)	41800	14000	1920	972	702
6	TE foam	Foam	256	256	200		
7	LE foam	Foam	256	256	200		
8	Double bias	Saertex (DB)	13600	13300	1780	144	213
9	Foam	Foam	256	256	200		

Figure 4.3 shows a typical blade internal layup somewhere close to mid span, which was used for the UPWIND 5 MW turbine. The arrows indicate different segments at a particular blade spanwise location and the box to its right lists the materials used in the layup at that particular blade segment. A uniform layer of gelcoat over the entire blade is considered.

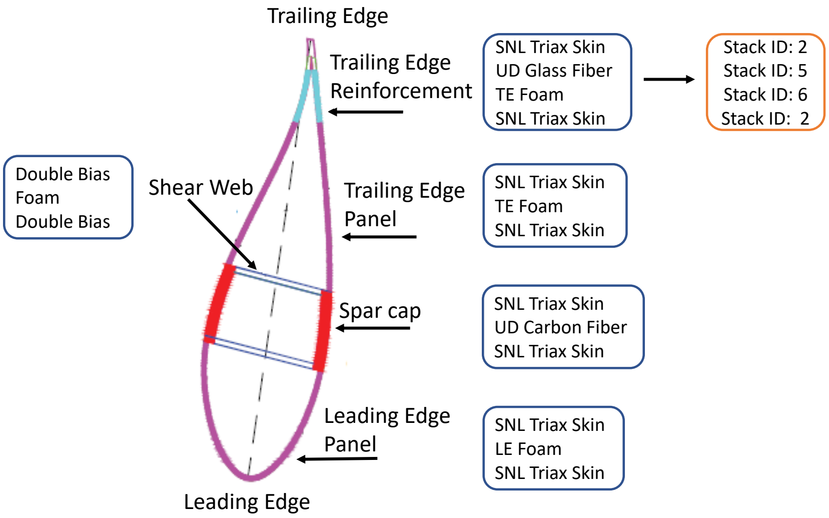


Figure 4.3: Typical blade internal layup for a wind turbine [31]

Based on these material properties, the blade internal layup at few blade spanwise locations is pre-defined, while for the locations in between, the layup is linearly interpolated. A linear interpolation ensures a smooth increase or decrease in the number of plies/fiber layers. Figure 4.3 also shows an example of the material distribution along with Stack ID's. Similarly, Table 4.4 shows the blade spanwise location at which the layup is defined [29] and the material distribution for all the segments, for each blade spanwise location.

Table 4.4: Spanwise distribution of stack layup [29]

Blade spanwise location ($\frac{r}{R}$)	**LE panel**	**Spar cap**	**TE panel**	**TE reinforcement**	**Shear web**
0	1,2,3,2	1,2,3,2	1,2,3,2	1,2,3,2	-
0.03	1,2,3,7,2	1,2,3,4,2	1,2,3,6,2	1,2,3,5,6,2	8,9,8
0.1	1,2,7,2	1,2,4,2	1,2,6,2	1,2,5,6,2	8,9,8
0.3	1,2,7,2	1,2,4,2	1,2,6,2	1,2,5,6,2	8,9,8
0.50	1,2,7,2	1,2,4,2	1,2,6,2	1,2,5,6,2	8,9,8
0.75	1,2,7,2	1,2,4,2	1,2,6,2	-	8,9,8
0.95	1,2,2	1,2,2	1,2,2	-	-

Additionally, the root Triax layer thickness is linearly terminated to 0 at 0.15 span. Shear webs begin at 0.05 span and end at 0.95 span, holding a constant foam thickness throughout. Similarly, the skin runs throughout the span with a constant thickness. The UD Glass used in the TE reinforcements terminates at 0.75 span.

The complete internal layup along the length of the blade is shown in Figure 4.4.

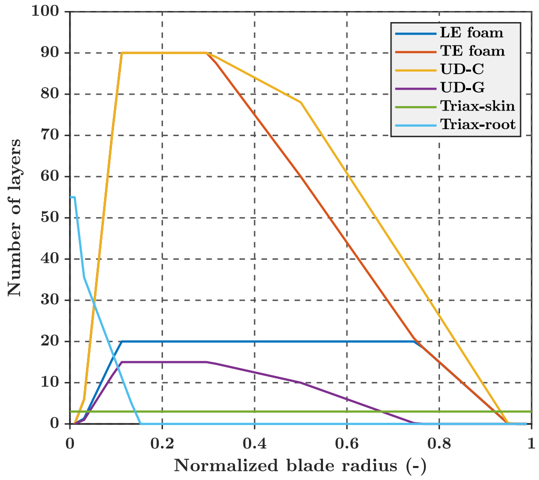


Figure 4.4: Complete blade internal layup

As stated before, this entire layup can now be scaled up or down, using the thickness factor (τ). This τ will be used as a design variable in the optimization process.

4.3.2. ESP model

For extracting different structural properties, the methodology implemented is explained below.

1. The modulus of elasticity stated in Table 4.3 are along the principle directions of the fibres, as shown in Figures 4.5a and 4.5b. However, the modulus of elasticity for angled plies can be converted to the global coordinate system along x and y, using relations as shown in Equation 4.1 [28].

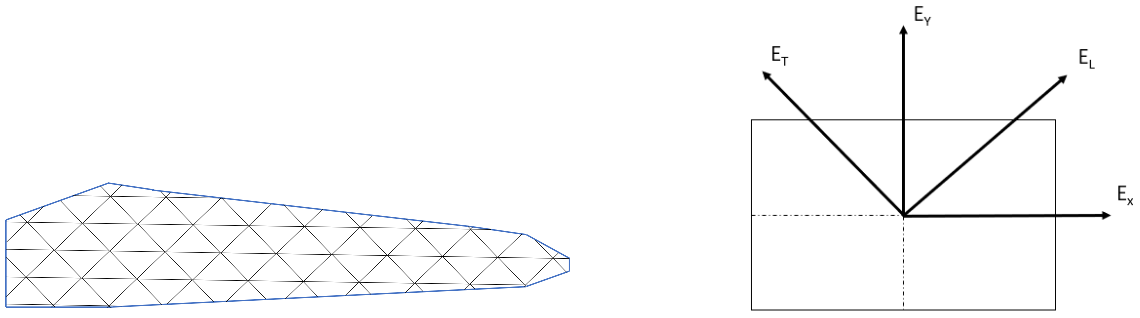
$$E_x^{ply} = \frac{1}{\frac{cos^4(\alpha)}{E_L} + \left(\frac{1}{G_{12}} - \frac{2\nu_{12}}{E_L}\right) \cdot sin^2(\alpha) \cdot cos^2(\alpha) + \frac{sin^4(\alpha)}{E_T}} \tag{4.1}$$

Where E_L and E_T are the Moduli of Elasticity in the longitudinal and transverse direction, ν_{12} is the Poisson's ratio and G_{12} is the Modulus of Rigidity.

Similarly, the shear modulus for the torsional stiffness can be calculated using Equation 4.2 [28].

$$G_x^{ply} = \frac{1}{\left(\frac{4}{E_T} + \frac{4+8\nu_{12}}{E_L} - \frac{2}{G_12}\right) \cdot sin^2(\alpha) \cdot cos^2(\alpha) + \frac{sin^4(\alpha)+cos^4(\alpha)}{G_{12}}} \tag{4.2}$$

Where, α is the ply angle. Hence, effective engineering constants for the Triax skin layers and for the double bias layers used in the shear webs are calculated. Figure 4.5a is a representation of the Triax skin, where it consists of fibres with an orientation of +45°, -45°and 0°.



(a) SNL Triax layer representation

(b) Laminar co-ordinate system

Figure 4.5: Triax Skin layer and the different laminar co-ordinate systems

2. The airfoil is divided into five different segments, as mentioned before. Each of these segments is further divided into really fine elements. An example of a fine element in the leading edge panel region can be seen in Figure 4.6. Each of these elements is assumed to be a rectangular section, having a stack of materials, based on Table 4.4. (X_E, Y_E) is the elastic center of the entire section at that particular blade spanwise location, while (X_C, Y_C) is the element centroid.

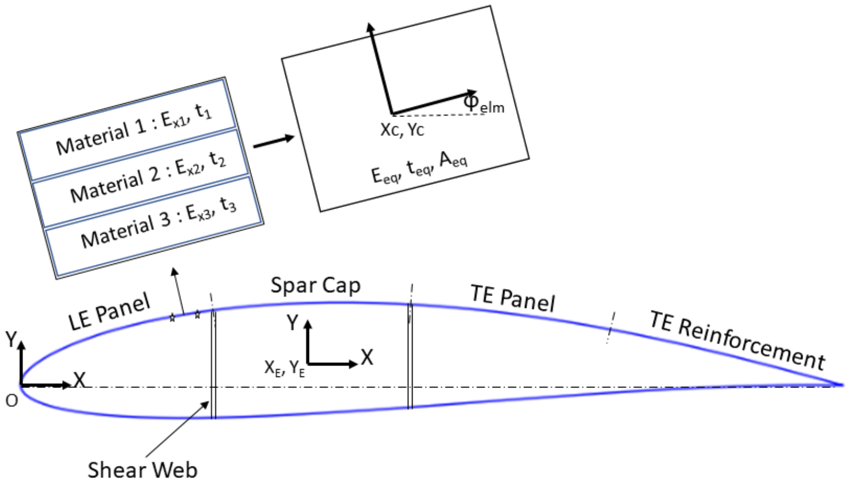


Figure 4.6: Airfoil layup at a given blade spanwise location

3. The equivalent properties for these elements is determined using a simple weighting method. For instance, the equivalent Modulus of Elasticity of an element in the LE panel segment is given by Equation 4.3 [27].

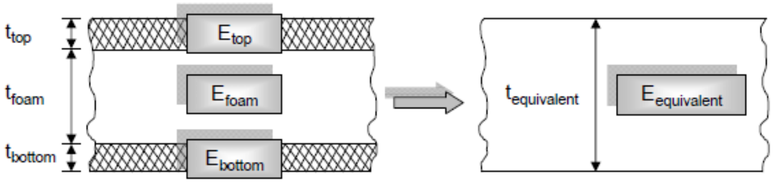


Figure 4.7: Equivalent Modulus of Elasticity of a given segment [11]

$$E_{eq}^{elm} = \frac{E_{top}^{triax} \cdot t_{top}^{triax} + E_{LE}^{foam} \cdot t_{LE}^{foam} + E_{bottom}^{triax} \cdot t_{bottom}^{triax}}{t_{top}^{triax} + t_{LE}^{foam} + t_{bottom}^{triax}} \tag{4.3}$$

4. Similarly, the equivalent Modulus of Elasticity values are calculated for the elements in the spar caps, TE panel, TE re-inforcements and shear webs. The same approach is used to determine the equivalent density, where the Modulus of Elasticity is replaced by density [27].

5. After calculating the equivalent Modulus of Elasticity, the area moment of inertia values of all these individual elements is calculated about its own centroid, using Equations 4.4 and 4.5. I_{xy} for a rectangular segment about its own centroidal axis will be zero.

$$I_{xx}^{elm} = \int y^2 dxdy \tag{4.4}$$

$$I_{yy}^{elm} = \int x^2 dxdy \tag{4.5}$$

6. The moment of Inertia for each of these segments is then rotated to a co-ordinate system along the elastic axes using Equations 4.6 and 4.7 [28], where ϕ_{elm} is the angle between an individual element and the elastic axes, as shown in Figure 4.6.

$$I_{XX}^{elm} = \frac{I_{xx}^{elm} + I_{yy}^{elm}}{2} + \frac{I_{xx}^{elm} - I_{yy}^{elm}}{2} \cdot cos(2\phi_{elm}) - I_{xy}^{elm} \cdot sin(2\phi_{elm}) \tag{4.6}$$

$$I_{YY}^{elm} = \frac{I_{xx}^{elm} + I_{yy}^{elm}}{2} - \frac{I_{xx}^{elm} - I_{yy}^{elm}}{2} \cdot cos(2\phi_{elm}) + I_{xy}^{elm} \cdot sin(2\phi_{elm}) \tag{4.7}$$

7. The overall section elastic center is then determined using Equations 4.8 and 4.9 [28].

$$X_E^{Section} = \frac{\sum_{i=1}^{N_{elm}} E_{eq}^i \cdot A_{eq}^i \cdot X_C^i}{\sum_{i=1}^{n} E_{eq}^i \cdot A_{eq}^i} \tag{4.8}$$

$$Y_E^{Section} = \frac{\sum_{i=1}^{N_{elm}} E_{eq}^i \cdot A_{eq}^i \cdot Y_C^i}{\sum_{i=1}^{n} E_{eq}^i \cdot A_{eq}^i} \tag{4.9}$$

8. After the section elastic centers are derived, the area moment of inertia values of the individual elements are transformed to the section elastic center, using the Huygens-Steiner's theorem, as shown in Equations 4.10 and 4.11.

$$I_{XX}^{elm} = I_{xx}^{elm} + A_{eq}^{elm} \cdot \left(Y_C^{elm} - \bar{Y}_E^{Section}\right)^2 \tag{4.10}$$

$$I_{YY}^{elm} = I_{yy}^{elm} + A_{eq}^{elm} \cdot \left(X_C^{elm} - \bar{X}_E^{Section}\right)^2 \tag{4.11}$$

9. The overall stiffness properties at a particular blade span location is then determined by summing up the contributions from all individual elements in the LE, spar cap, shear web, TE and TE reinforcement segments, as shown in Equations 4.12, 4.13 and 4.14 [28].

$$EI_{XX}^{Section} = \sum_{i=1}^{N_{elm}} E_{eq}^{elm,i} \cdot I_{XX}^{elm,i} \tag{4.12}$$

$$EI_{YY}^{Section} = \sum_{i=1}^{N_{elm}} E_{eq}^{elm,i} \cdot I_{YY}^{elm,i} \tag{4.13}$$

$$EA^{Section} = \sum_{i=1}^{N_{elm}} E_{eq}^{elm,i} \cdot A_{eq}^{elm,i} \tag{4.14}$$

For calculating the torsional stiffness, the modulus of elasticity is replaced by the modulus of rigidity [27].

10. The mass distribution along the blade is also be determined using a similar approach, as shown in Equation 4.15 [27].

$$\left(\frac{M}{L}\right)_{section} = \sum_{i=1}^{N_{elm}} \rho_{eq}^{elm,i} \cdot A_{eq}^{elm,i} \tag{4.15}$$

Although the NREL5MW reference turbine is widely used for research purposes, no data about the internal layup of the blade existed. Resor [29], from Sandia National laboratories, presented a study in which an internal layup was developed to best match the spanwise mass and stiffness properties of the NREL5MW reference turbine [32]. While doing so, the layup developed was designed to satisfy basic failure criteria, mainly tip deflection, fatigue damage and buckling. The analytical model developed as a part of this research (ESP), is validated against the data developed by Sandia Labs. For this, the same internal layup is used and the resulting blade mass and stiffness properties from the analytical model are compared with the values obtained by SANDIA labs. It should be noted that, for the layup developed by SANDIA labs, not all properties showed good agreement with the actual values of the NREL5MW turbine.

Figures 4.8 and 4.9 show three of the several properties extracted, where a good agreement with the values from Sandia Labs is observed. However, it can be seen how the actual values of the NREL5MW turbine (indicated by the red line) are not in complete agreement with SANDIA labs.

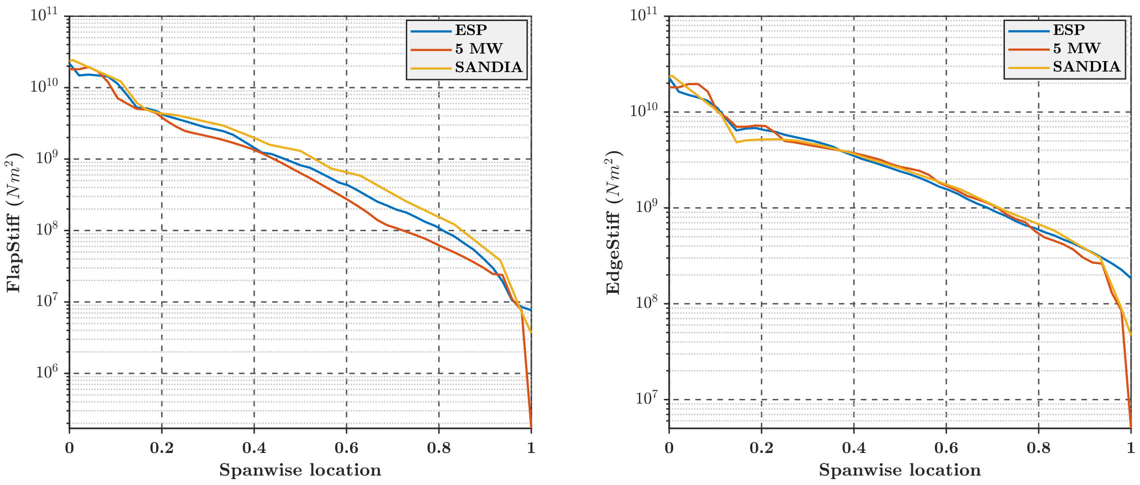


Figure 4.8: Comparison of flapwise and edgewise stiffness

Figure 4.8 show a comparison of the flapwise and edgewise stiffness values obtained from ESP, with the respective values obtained by SANDIA labs and the actual values of the NREL5MW turbine. The inconsistency in the stiffness values can be attributed to the fact that the standard layup values were defined at locations defined in Table 4.4, while the rest were linearly interpolated.

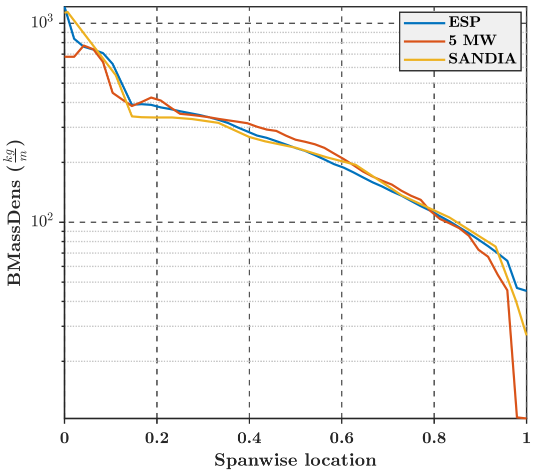


Figure 4.9: Comparison of blade mass density

A comparison of the mass densities is shown in Figure 4.9, which shows good agreement with the data from SANDIA, except for at the root. Again, the difference can be attributed to interpolation of layup values.

4.4. XDSM: Dynamics module

Figure 4.10 shows a detailed XDSM of the Dynamics module. The Dynamics module performs the necessary pre-processing to call FAST, runs an IEC load case on FAST and post-processes data to convert it into constraints.

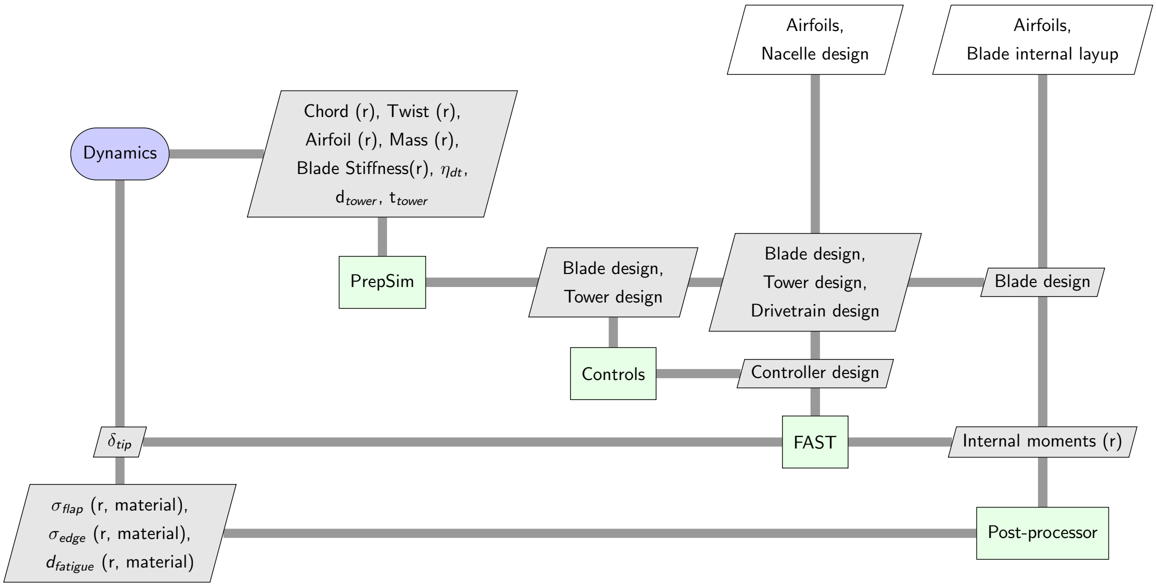

Airfoils, Nacelle design
Airfoils, Blade internal layup
Dynamics
Chord (r), Twist (r), Airfoil (r), Mass (r), Blade Stiffness(r), η_{dt}, d_{tower}, t_{tower}
PrepSim
Blade design, Tower design
Blade design, Tower design, Drivetrain design
Blade design
Controls
Controller design
δ_{tip}
FAST
Internal moments (r)
σ_{flap} (r, material), σ_{edge} (r, material), $d_{fatigue}$ (r, material)
Post-processor

Figure 4.10: XDSM of the Dynamics module

The main module can further be broken down into sub-modules that perform these tasks. The PrepSim module takes in all the available blade, airfoil, drivetrain and tower properties, from the user and from other modules, and converts them into all the template files which are required to run BModes and FAST, ensuring its integration into WINDOW. As explained in Section 4.1, BModes is a module developed by NREL that calculates the mode shapes for the blade and the tower, that serve as an input for ElastoDyn, the structural dynamics module of FAST. The Controls module performs full system linearization and calculates the PI controller gains. The FAST module makes a call to the aero-servo-elastic model, FAST for the user defined load case. The internal moment reactions from FAST are used by the Post-processor module to convert them into spanwise stresses and fatigue damage. Many of the MATLAB scripts and the analytical expressions used to build the python wrapper for FAST, have been constructed over years to make an interface for the Simulink version of FAST, by the Delft University of Technology.

4.5. Dynamics: Sub-modules

The submodules, *PrepSim*, *Controls*, *FAST* and *Post-processor* have been elaborated in this section.

4.5.1. PrepSim

PrepSim or 'Preparation for Simulation' serves as the module that fills in the missing links between the other modules in WINDOW and the Simulink version of FAST.

The spanwise mass and stiffness properties from the ESP module serve as an input for PrepSim. From all the available inputs as shown in Figure 4.10, few other **Blade** parameters that are derived have been listed below:

1. Offsets for the center of gravity, tension and shear center.
2. Pitch axis location, based on standard parameters available at few sections, using which, the values at the other stations were interpolated.
3. Blade mass moment of inertia values based on blade mass and blade thickness.

For the **Tower**, the wall thickness and diameter values at the top and bottom are taken as input from the support structure module, while tower extra mass is taken as a user-defined input. The tower extra mass compensates for the cables and transformer stationed inside the tower. The Modulus of Elasticity and the Shear Modulus for the tower are the same as that of Steel (210 GPa and 80 GPa respectively). Based on these values, the following parameters are derived :

1. Spanwise distribution of the tower diameter, which is a linear interpolation using the given values of tower top and bottom diameter.
2. Tower mass, based on the tower section details and tower density, with added tower extra mass.
3. Tower axial and torsional stiffness, using the respective modulus and tower section details.
4. Tower mass moment of inertia using tower mass and tower section details.

The airfoil, drivetrain and nacelle data is defined externally by the user. *PrepSim* takes in this data, along with the complete blade and tower design, and generates files that can be read by the MATLAB interface for the Simulink version of FAST.

4.5.2. Controls

The main purpose of having a controls module is to automate the process of calculating the proportional and the integral gain values of the PI controller, so as to make it a part of the optimization loop. The drivetrain parameters coming in as user-defined inputs are the gearbox ratio, gearbox efficiency and the generator efficiency. The blade pitch rates, startup pitch values and braking parameters are taken to be the same as that of the standard NREL5MW turbine [32]. The step-wise methodology to calculate the gains is explained in this section.

Rated wind speed

The pitching action of the controller is for wind speeds above rated. The rated wind speed, which is taken as an input from the RNA block (refer to Figure 3.4) is calculated using Equation 4.16, once the power coefficient (C_P) value is known.

$$U_{rated} = \left(\frac{P_{rated}}{0.5 \cdot C_P \cdot \rho \cdot A_{rotor}} \right)^{1/3} \tag{4.16}$$

$C_p - \lambda$ curve

The first step is to determine the operating power co-efficient at the given tip speed ratio and fine pitch angle for the given rotor design. This module uses a BEM script that is a part of the interface developed at the Delft university of technology. The $C_P - \lambda$ curve is obtained by solving the BEM theory for a given fine pitch angle and tip-speed ratio (λ), which are design variables and hence, will be guessed by the optimizer. The value of C_P at the λ guessed by the optimizer is then used to calculate the rated wind speed.

Torque-control curve

The next step in control design, is to obtain the torque control curve. A torque control curve has various regions of operation, as shown in Figure 4.11 , wherein the various rotational speed values and the corresponding torque values need to be determined.

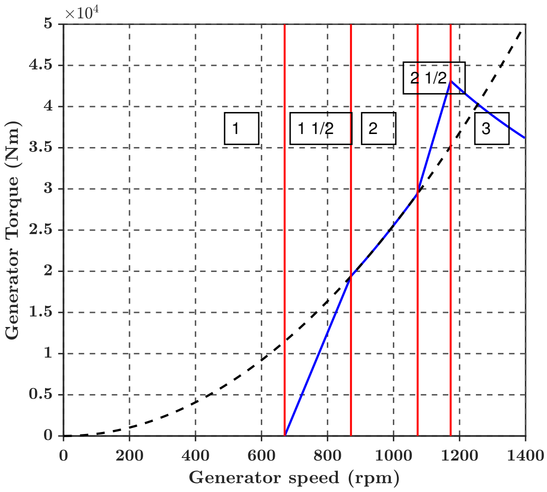


Figure 4.11: Torque control curve of the NREL5MW turbine [11]

The different control regions have been explained by Ashuri [10]. In region 1, no power is produced, as the cut-in rpm is decided by the cut-in wind speed. Region 2 is the partial load region, wherein the turbine operates at optimal power producing conditions. The relation between the torque and rotational speed in region 2 is given by Equation 4.17, where the Torque (T) and the rotational speed (ω) are related by the optimal mode gain (k_{opt}).

$$T = k_{opt} \cdot \omega^2 \tag{4.17}$$

In region 3, the turbine operates in the full load region and to keep the power constant, torque decreases with increasing rotational speed. Regions 1 1/2 and 2 1/2 represent linear transition regions, wherein the slope of region 2 1/2 represents the slip of an induction generator [10]. In the model, the relative difference in the operational range, of the rotational speed, for regions 1 1/2 and 2, is taken to be the same as that of the NREL5MW turbine [32].

Using the existing model developed at the Delft university of technology, the drivetrain high speed shaft inertia is scaled using Equation 4.18, where I_{hss}^{ref} represents the high speed shaft inertia of the NREL5MW turbine and $\omega_{hss,rated}^{ref}$ represents the high speed shaft rotational speed at rated conditions, for the NREL5MW turbine. The high speed shaft inertia is scaled so as to prevent the generator from reaching very high values of rotational speed during linearization.

$$I_{hss} = I_{hss,ref} \cdot \left(\frac{P_{rated}}{5 \cdot 10^6}\right)^{5/3} \cdot \left(\frac{\omega_{hss,rated,ref}}{\omega_{hss,rated}}\right)^2 \tag{4.18}$$

The low speed shaft inertia is then calculated using Equation 4.19, where r_g is the gearbox ratio.

$$I_{lss} = I_{hss} \cdot (r_g)^2 \tag{4.19}$$

Automated controller design

As most systems in the world are non-linear while the conventional Proportional-Integral-Derivative (PID) controllers are based on linear theory, the first step towards controller design involves linearization of the turbine at specific steady state operating points. A linearized model is only valid in some vicinity of the steady state operating point. For each of these operating points, the controller design differs, leading to different proportional and integral gain values.

Using scripts from the existing TU Delft model, the steady state rotational speed of the rotor, generator speed, generator torque and the collective blade pitch angle are determined for different windspeeds. The *Controls* module then makes a call to FAST with the linearization flag 'on', in the main input file for FAST. The system is then linearized about these steady state operating points with all structural degrees of freedom turned off and with the frozen wake toggle on [33].

The main purpose of linearization is to determine the sensitivity of aerodynamic power to rotor collective blade pitch, which is then used to calculate the proportional and integral gains. The variation of aerodynamic power sensitivity ($\frac{\partial P}{\partial \theta_P}$) with different pitch angles for the NREL5MW turbine is shown in Figure 4.12.

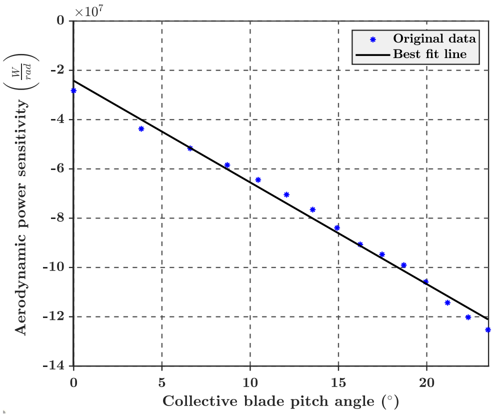


Figure 4.12: Aerodynamic power sensitivity v/s collective blade pitch angle [32]

It can be seen how the aerodynamic power sensitivity changes linearly with the pitch angle (θ_P). This dependence of the aerodynamic power sensitivity on the pitch angle explains the need for gain scheduling, as seen in Equations 4.20 and 4.21 [32]. Using these equations, the proportional and integral gains, as a function of the pitch angle, are calculated, where ideal response characteristics given by $\omega_{\phi n}$=0.6 rad/s and ζ_ϕ=0.6-0.7 are assumed, as recommended by Hansen *et al.* [34].

$$K_P(\theta_P) = \frac{2 \cdot I_{dt} \cdot \Omega_0 \cdot \zeta_\phi \cdot \omega_{\phi n}}{N_{Gear} \cdot \left[-\frac{\partial P}{\partial \theta}(\theta_P = 0)\right]} \cdot GK(\theta_P) \tag{4.20}$$

$$K_I(\theta_P) = \frac{I_{dt} \cdot \Omega_0 \cdot \omega_{\phi n}^2}{N_{Gear} \cdot \left[-\frac{\partial P}{\partial \theta}(\theta_P = 0)\right]} \cdot GK(\theta_P) \tag{4.21}$$

where $GK(\theta_P)$ is the gain correction factor, given by Equation 4.22.

$$GK(\theta_P) = \frac{1}{1 + \frac{\theta_P}{\theta_k}} \tag{4.22}$$

θ_k is the pitch angle at which the aerodynamic power sensitivity has doubled from its value at rated.

The gain and gain correction factor values for the NREL5MW turbine are shown in Figure 4.13.

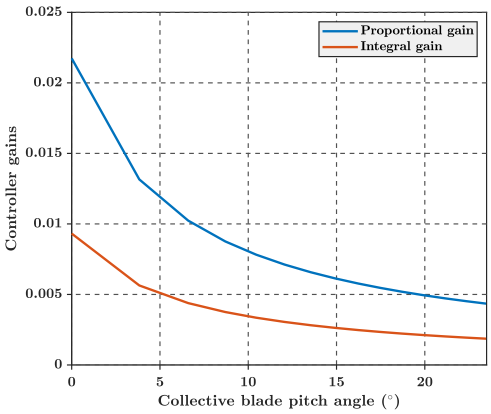


Figure 4.13: Gain and Gain correction factor values for different pitch angles [32]

4.5.3. FAST

The FAST module performs two important tasks :

- Calls the wind field generator, TurbSim, which then generators a three dimensional flow field with spatial and temporal variation. The wind conditions, namely the mean wind speed and the wind type, are load case dependent. Various options for the wind type include: steady wind speed, stepped wind profile, normal turbulence model, extreme turbulence model, etc. The load case used for this research will be defined later, in Chapter 5.
- For the defined load case, it makes a call to the Simulink version of FAST and runs the simulation for the stipulated time, generating the output file that is read by the Post-processor module.

4.5.4. Post-processor

The Post-processor module converts the internal reaction moments obtained along the span of the blade into stresses and fatigue damage, in different materials.

Spanwise stresses

Even though the moment at the root is the highest, it does not necessarily translate to maximum stress. As the blade external geometry and internal layup thicknesses are different throughout the blade span, a need to check stresses at different locations arises. For the same, 5 different spanwise locations ($\frac{r}{R}$) are chosen : 0.15, 0.3, 0.5, 0.75 and 0.95. As a result, stresses at 6 different points along the blade are calculated. The quantities that are checked at each of these locations are given in Table 4.5.

Figure 4.14 points out the exact locations along the blade where these parameters are calculated. The flapwise stresses are calculated at the maximum thickness point of the airfoil, in the spar cap segment, while the edgewise stresses are calculated at the maximum chord point, in the TE reinforcement segment. In the spar cap segment, the flapwise stresses are determined for the skin and the UD carbon fibers, while in the TE reinforcement segment, the edgewise stresses are determined for the skin and the UD glass fibers.

Table 4.5: Quantities calculated for stress check

Parameter	Material
Flapwise stress	SNL Triax
Flapwise stress	UD Carbon
Edgewise stress	UD Glass fiber
Edgewise stress	SNL Triax

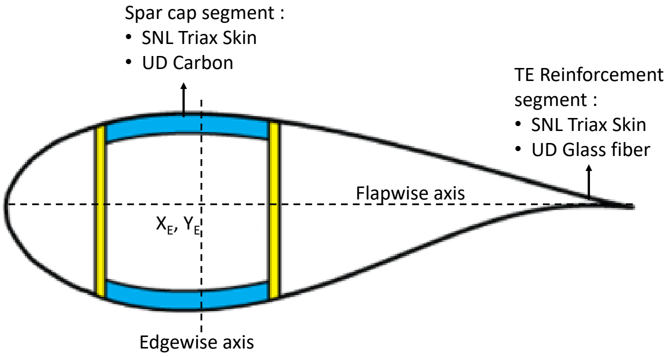


Figure 4.14: Locations selected for stress calculation

The internal bending moment values at different spanwise locations are obtained as an output of FAST, while the stiffness properties and the farthest point are taken from the pre-processing model, ESP. The strain experienced at the farthest point is then obtained using Equation 4.23.

$$\epsilon = \frac{M_{section} \cdot y}{EI} \tag{4.23}$$

The stresses resulting in a particular material is then determined from the strain in the fibers and its modulus of elasticity. For instance, the stress in the skin can be calculated using Equation 4.24.

$$\sigma_{skin} = E_{Triax} \cdot \epsilon_{skin} \tag{4.24}$$

Fatigue damage

The fatigue damage is evaluated for stresses at all the four points shown in Figure 4.14. To evaluate fatigue damage, the method adopted is the same as that used by Griffith and Ashwill [30], based on Minor's rule, given by Equation 4.25.

$$d_{fatigue} = \sum_i \frac{n_i}{N_F \cdot (\gamma_f \cdot \gamma_m \cdot S_i)} \leq 1.0 \tag{4.25}$$

where
n_i = number of cycles in the stress range S_i
N_F = number of cycles to failure
γ_f = Safety factor for consequences of failure
γ_m = Material safety factor

The detailed methodology adopted is explained below:

1. For fatigue analysis, 6 simulations with different wind speeds are considered, between cut-in and the cut-out wind speed, in steps of 4 m/s, with a normal turbulence model.
2. The spanwise stresses calculated are then converted into cyclic stress ranges using the rain flow counting method, a histogram of which is made and the number of occurrences (n_i) of each stress range is determined.
3. For each of these stress ranges, the number of cycles to failure (N_F) are obtained from the S-N curve given material, using equation 4.26. The S-N curve for a material is derived from test results, where the number of cycles to failure for a given stress range is determined and the same process is repeated for different stress ranges. A linear fit to these points, also known as the S-N curve, can be defined with the help of two important parameters, the inverse slope of the line, and the effective single cycle strength. Effective single cycle strength is the value of the y-intercept of the S-N curve, or the value of stress range, for which the material fails after one cycle.

   $$N_F = \left(\frac{1}{C} \cdot S\right)^{-b} \tag{4.26}$$

   where
   C = effective single cycle strength of the material
   b = inverse of the slope of the S-N curve obtained from test results for a particular material

   Equation 4.26 can be rearranged into a logarithmic form given by Equation 4.27.

   $$\log S = \log C - \frac{1}{b} \cdot \log N_F \tag{4.27}$$

   Different values for the slope and UCS for the blade materials are given in Table 4.6 [29].

   Table 4.6: Material properties for Fatigue

Material	b	C (Mpa)
E-LT-5500 (UD)	10	700
Carbon (UD)	14	1047
SNL Triax	10	700

4. A simple summation of $\frac{n}{N}$ values for each stress range gives the total 10 min damage due to a particular wind speed at the point where the stress is calculated.
5. To get the total damage due to a particular wind speed in a turbine's lifetime, at first, the number of 10 min periods of that particular wind speed are evaluated. The probabilities of different wind speed bins are calculated using the cumulative density function given by the IEC standards, as shown in Equation 4.28. The probability of a particular wind speed bin is found out by simply substituting the upper and lower limits of the bin, in Equation 4.28 and then taking the difference of the probabilities.

   $$P(U) = 1 - e^{-\left(\frac{U}{a_{scale}}\right)^k} \tag{4.28}$$

6. The total damage for a particular wind speed is just a product of the damage in one 10 min period and the number of ten minute periods.
7. The same procedure is followed for all the wind speed bins and the damage for each of these bins is summed up to get the total fatigue damage.

5 Optimization Setup

Before delving into the optimization setup, it is important to evaluate the function evaluation time, which is the overall time taken to run the analysis block in the new WINDOW framework, once. Running a wind farm level optimization, with FAST in the loop, for multiple load cases, would drastically increase the overall function evaluation time and prove to be computationally expensive. Hence, a need to reduce the number of load case arises. The chapter starts off with defining the load case for which FAST will be run and it also elaborates the methodology used to reduce the overall function evaluation time. The reader is then familiarized with some basic optimization terminology. Lastly, the use case specific objective function, design variables, constraints and driver algorithms used are elaborated.

5.1. Load case selection

In a usual design certification process of a wind turbine, the wind turbine manufacturers simulate all the IEC load cases, as specified in the IEC standards. However, the certification cases are ran during the detailed design phase and doing the same for preliminary rotor design optimization would not be a viable option, given the limited computational time and resources. Hence, for design optimization purposes, a need to reduce the number of load cases arises.

To reduce the number of load cases, majority of the IEC load cases are run for the NREL5MW reference turbine. Few critical load cases, with high values of tip deflection and moment, are shown in Table 5.1, where the mentioned external conditions include Extreme Turbulence Model (ETM), Normal Turbulence Model (NTM), Extreme Wind Shear (EWS) and Extreme Direction Change (EDC). The maximum value of tip deflection is observed for the normal power production mode with ETM while the highest root moments are observed in the start up mode with a change in the wind direction. The work carried out by Ashuri [35] involved using FAST to optimize the turbine LCOE, and the tip deflection was found out to be the main design driver for the rotor. Hence, of all the load cases, Design Load Case (DLC) 1.3 at the rated wind speed is chosen to be the most critical load case, based on its highest tip deflection value. Also, the stress values for the moments given in Table 5.1 are found to be way below the ultimate strength and hence, tip deflection is chosen to be the main driver behind the load case selection.

Table 5.1: Results of critical load cases, simulated for the NREL5MW turbine

DLC	Design situation	External condition	Wind Speed	Tip Deflection (m)	Resultant moment (KNm)
1.3	Power production	ETM	Rated	7.793	15689
1.5	Power production	EWS	Rated	7.634	14519
2.1	Power production + Occurrence of fault	NTM	Rated	7.147	15043
3.3	Start-up	EDC	Cut-in	0.297	15972

DLC 1.2, for the fatigue limit state, requires a 10 minute simulation for all wind speeds between cut-in and cut-out, with a normal turbulence model , in the normal production mode of the turbine. As mentioned in Section 3.3.2, including fatigue check in the main optimization loop would be highly infeasible due to limited computational resources. As a result, the fatigue limit state will only be checked for the final optimized design given out by the optimizer. To summarize, the two load cases that will be simulated are given in Table 5.2, of which, DLC 1.3 will be included in the main optimization loop and DLC 1.2 will only be checked for the final optimized turbine. The fatigue damage evaluation for the final optimized design will be done only to check the feasibility of the design with respect to fatigue. If fatigue is found out to be the main design driver, only changes in the design will be recommended and the main optimization run with the design update will not be carried out again.

Table 5.2: Final load cases to be simulated

DLC	Design situation	External condition	Wind Speed
1.3	Power production	ETM	Rated
1.2	Power production	NTM	Cut-in : 4 : Cut-out

In reality, each of these load cases is run for a period of 10 minutes and with at least 6 different seeds. Each seed gives a different realization of the wind speed profile, for the same mean wind speed. Consequently, the simulations are run for different seeds and then the maximum value is used. However, running a simulation for 10 minutes with 6 different seeds, in an optimization loop, would drastically increase the total time elapsed and hence, is not a feasible option. A need to reduce the time taken per load case arises and the techniques used in this research have been elaborated in Section 5.2.

5.2. Function evaluation time

This section mainly addresses the issue regarding the time taken for one function evaluation. In a problem as complex as wind turbine rotor optimization where in every function evaluation, a dynamic simulation is involved, the total time taken for one function evaluation becomes critical.

The total time taken to run the analysis block in the new WINDOW framework once, also known as the function evaluation time, along with the break up of time taken by individual disciplines is shown in Table 5.3. The time taken by the Dynamics block is considering a 10 minute simulation with a single seed.

Table 5.3: WINDOW function evaluation time break up

Module	Computation time (s)
RNA	85
Wake Aerodynamics	70
Support Structure	20
Dynamics	1480
Other	3
	1658

With the given function evaluation time, it would be highly infeasible to run multiple optimization runs with different drivers and settings, especially gradient free algorithms that require many more function evaluations as compared to gradient based optimizers.

At first, the time taken by different sub-modules of the Dynamics block is calculated and each time consuming sub-module is individually dealt with. The most time consuming modules identified, include the *Controls* and *FAST* sub-modules. The *Controls* sub-module involves a full system linearization at wind speeds from rated to cut-out, while the *FAST* sub-module involves running FAST for the given load case. The approach used to reduce the overall time taken by the *Controls* and *FAST* sub-modules is elaborated in this section.

5.2.1. Controller design time

In the controls module, full system linearization is the most time consuming process. Jonkman *et al.* [32] recommend linearizing at all wind speeds above rated, to determine the PI controller gains, as a function of the pitch angle. To achieve a time reduction in the linearization process, without compromising on the controller design, the following parameters are tuned : *Simulation time before linearization* and the *Linearization wind speeds*. The simulation time before linearization is the time for which FAST is run before linearizing, to get rid of the transients.

As aerodynamic power sensitivity is the only output of linearization that is used, the time taken for the power output transients to die out is plotted, as shown in Figure 5.1, and a time of 20 s is selected, instead of the recommended value of 60 s.

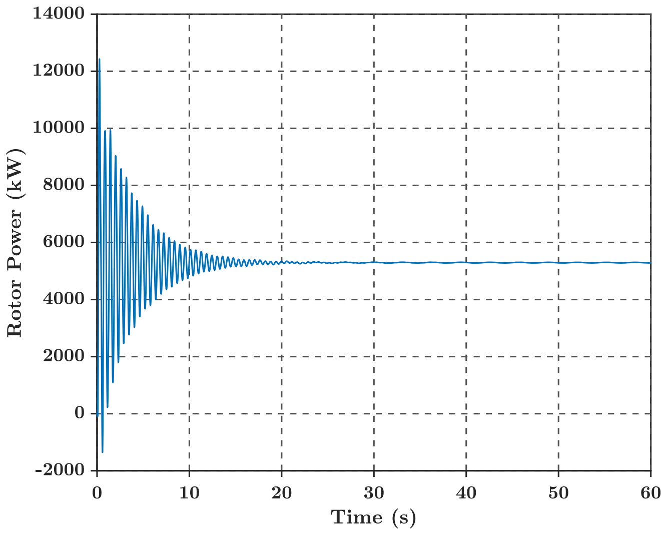


Figure 5.1: Power output during linearization at 15 m/s

Also, as shown in Figure 4.12 from Chapter 4, the aerodynamic power sensitivity varies linearly with the wind speeds. Hence, the linearization is tried out in steps of 2 m/s, instead of the recommended value of 1 m/s. The overall effect of changing the simulation time before linearization and the linearization wind speeds on the controller gains is analyzed, and it shows a good agreement with the recommended parameters, as shown in Figure 5.2.

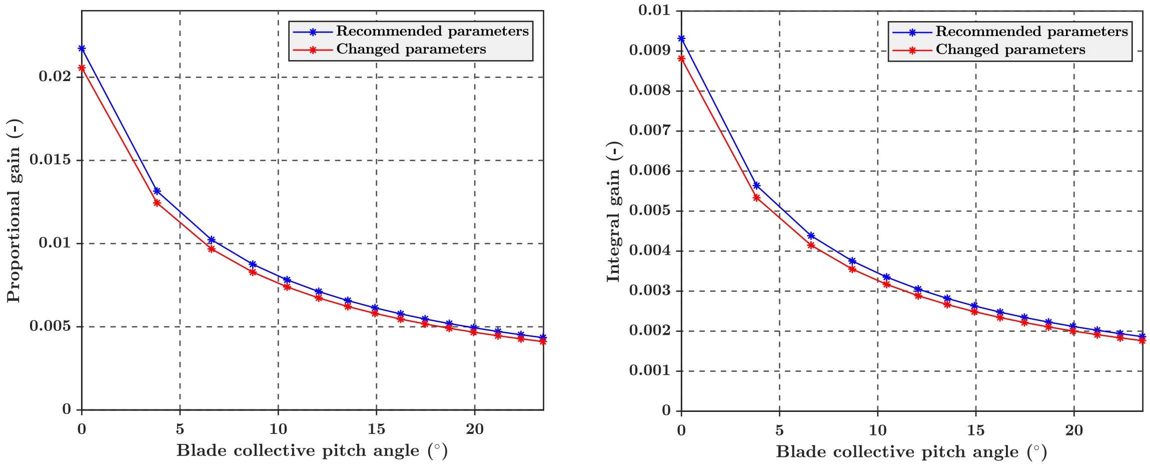


Figure 5.2: Comparison of proportional and integral gains before and after changing linearization parameters

5.2.2. Load case simulation time

The biggest hindrance while running a dynamic simulation in an optimization loop is the simulation time itself. The effect of different simulation times on the tip deflection can be seen in Figure 5.3, where a 10 minute simulation is run, for the NREL5MW reference turbine, with 10 different seeds (10 different realizations of the wind profile about the same mean wind speed). The black line indicates the mean value trend line while the red points represent the maximum deflection values for a simulation time of 60 s and 600 s. A similar trend is observed for the root moment values.

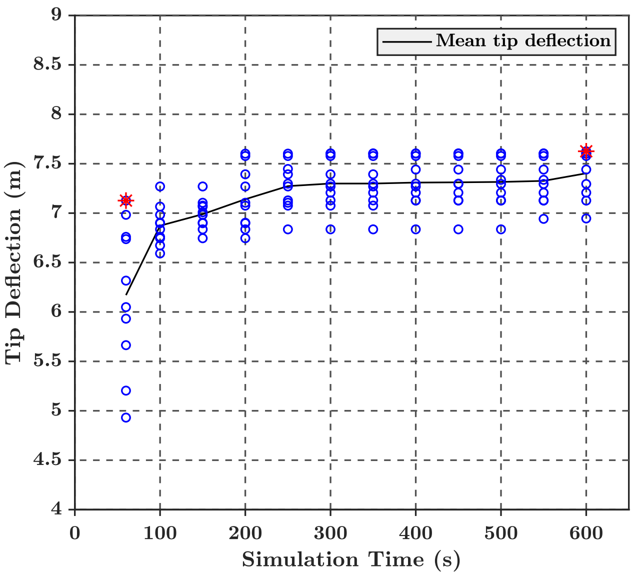


Figure 5.3: Tip deflection for different simulation time and seeds

Based on these observations, two clear conclusions can be drawn :

1. Reducing the simulation time results in missing out on an extreme which may occur later in the 10 minute simulation.
2. The effect of running multiple seeds becomes prominent as the simulation time reduces. For instance, the scatter in the points for a simulation time of 60 s is higher as compared to the scatter for 600 s.

To reduce the simulation time while still capturing the worst extreme, a simulation time of 60 s is chosen and is compensated by an additional safety factor that accounts for this reduction in simulation time. Also, the seed chosen for the optimization is the seed for which the highest value of extreme is observed, for a simulation time of 60 s.

The safety factor that compensates for the reduction in simulation time is given by Equations 5.1 and 5.2, where $\delta_{tip,600}$ is the maximum tip deflection observed for a simulation time of 600 s and $\delta_{tip,60}$ is the maximum tip deflection for a period of 60 s. Similarly $M_{root,600}$ is the maximum root moment for a period of 600 s while $M_{root,60}$ is the maximum root moment for a period of 60 s.

$$\gamma_{st}^{tip} = \frac{\delta_{tip,600}}{\delta_{tip,60}} \tag{5.1}$$

$$\gamma_{st}^{stress} = \frac{M_{root,600}}{M_{root,60}} \tag{5.2}$$

In an ideal scenario, it would always be advisable to run the load cases for a period of 10 minutes for multiple seeds and then consider the maximum value. Also, the same seed can have a different outcome when a completely different turbine design is tried out and consequently, the same compensation factor may not hold completely true. It is nevertheless assumed that the main reason for the extreme response of the design to the chosen seed for the 60 s simulation, is the shape of the corresponding wind field realization.

To analyze the effects of the methodology adopted above, a simulation with a completely different rotor design is tested and analyzed for the chosen seed and for a time of 60 s. This value, multiplied with the compensation factor, is then compared with the extreme value for 600 s, which resulted in a deviation of 0.09 m, which is within acceptable limits. However, it should be noted that this stochastic nature of the interaction between different designs and seeds cannot be completely eliminated.

Using the **same seed** throughout the optimization process produces the same wind profile every time it's run and hence, ensures a fair comparison between different turbine designs. Having a different seed in every function evaluation would induce its own stochastic nature which would again have negative implications. For instance, a stiffer design could still give a tip deflection higher than a design not as stiff, just because of a different realization of the wind profile.

5.2.3. Reduced function evaluation time

With these added changes explained in Sections 5.2.2 and 5.2.1 , the overall time taken by the Dynamics module drastically reduces, as shown in Table 5.4. With the incorporated changes in the Dynamics module, the overall function evaluation time of WINDOW is now reduced to an average value of about 420 s.

Table 5.4: Dynamics block time break up comparison

Dynamics block: **Sub-modules**	**Initial time (s)**	**Final time (s)**
PrepSim	1	1
Controls	1030	170
FAST	450	85
Post-processor	1	1

5.3. Optimization terminology

Most optimization problems are formulated to either maximize or minimize a particular quantity. For instance, maximizing profits or minimizing overall costs. The quantity to be optimized is called the objective function, given by *f*. This objective function is a function of a set of variables that can be controlled by the optimizer, known as the design variables. The design variables can be represented by a vector $x = [x_1, x_2, ...x_n]^T$, where n is the number of dimensions of the problem. The aim of an optimization problem involving minimization is to find the set of design variables, (*x**), such that *f(x*) = min(f(x))*.

In most real life problems, the design space is restricted, which can be given by a set of constraints imposed on the optimizer and in the form of bounds for the design variables . These constraints could be inequality constraints, given by $g(x) \leq 0$ or equality constraints, given by *h(x) = 0*. While for the design variables, they can only take values between the limits specified in the form of bounds. A constrained minimization optimization problem with n design variables, l inequality constraints and m equality constraints can be summarized as :

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0, \; i = 1, 2, \ldots, l. \\ & h_j(x) = 0, \; j = 1, 2, \ldots, m. \\ \text{where} \quad & x = [x_1, x_2, \ldots, x_n]^T \\ & x_{lower} \leq x \leq x_{upper} \end{aligned}$$

The constraints can be implemented in different ways, depending on the algorithm. For instance, in the

Genetic Algorithm (GA), the constraints are implemented in the form of a penalty to the objective function for violating a constraint [36]. Also, not all optimizers can handle a constrained optimization problem.

5.4. Objective function

The wellness or fitness of the optimization problem is evaluated in terms of the objective function. For a complex multifarious system like a wind farm, only minimizing the overall costs of the various elements involved or only maximizing the overall energy production, would always result in a sub-optimal design. A parameter that accurately represents the trade-offs between the overall costs and annual energy production is the LCOE of the wind farm.
As a result, for the minimization problem in this use case, the objective function is the *LCOE* of the wind farm, given by Equation 5.3. The LCOE depends on the total investment costs (C_{inv}), operation and maintenance costs ($C_{O\&M}$), decommissioning costs (C_{decomm}), the Annual Energy Production (AEP), electrical transmission efficiency (η_{trans}) and the annuity factor (a), which depends on the interest rate (r).

$$LCOE = \frac{C_{inv}}{a \cdot AEP \cdot \eta_{trans}} + \frac{C_{O\&M}}{AEP \cdot \eta_{trans}} + \frac{C_{decomm} \cdot (1+r)^{-T}}{a \cdot AEP \cdot \eta_{trans}} \tag{5.3}$$

where Annuity is given by Equation 5.4, T being the total lifetime of the project.

$$a = \frac{(1+r)^T - 1}{r} \tag{5.4}$$

5.5. Design variables

The numerical input values that are in control of the optimizer are called the design variables. In real life conditions, each of these design variables has a lower limit and an upper limit, also known as bounds. For instance, when designing the blade planform, the chord values cannot be negative or infinitely large, as that would be physically impossible and hence, has to be restricted within certain limits. Each of these design variables is expected to influence the objective function, which in this case, is the LCOE of the wind farm.

For a comparison between the static and dynamic models with different optimization algorithms, the NREL5MW reference turbine is redesigned. Hence, the NREL5MW rotor design is used as a reference and the bounds for the design variables are decided based on these reference values. As stated in Section 3.1, the design variables with respect to which the rotor will be optimized are :

1. Chord (c) at 3 pegged nodes : At the root, 70% and 90% of the blade span
2. Twist (β) at 3 pegged nodes: At the transition section, 40% and 70% of the blade span
3. Tip speed ratio (λ)
4. Fine pitch angle (θ)
5. Thickness factor (τ)

The thickness factor (τ) will be used to scale the reference layup developed in Section 4.3.1, based on the study conducted by Resor [29]. The layup developed by SANDIA labs was based on manual optimization techniques and just enough to satisfy the ultimate limit states, fatigue and buckling. As buckling is not modelled in this research, the lower bound of the thickness factor is limited to 0.7, so to avoid blade designs that might fail in buckling. While checking the response of LCOE with respect to the fine pitch angle, it is observed that values lower than -1° lead to high values of LCOE. This can be attributed to conditions similar to 'pitch to stall', resulting in a significant drop in the energy production. Consequently, the lower bound is set to -1°, and the solutions will be checked for their closeness to the lower bound while performing the optimization runs.

The design variables along with their respective bounds, are summarized in Table 5.5.

Table 5.5: Design variables for the defined use case

Design variables	Reference value	Lower bound	Upper bound
$c_{root,0.7,0.9}$	[3.54, 3.01, 2.31] m	0.75 · Reference	1.25 · Reference
$\beta_{trans,0.4,0.7}$	[13.31, 9.0, 3.13]°	0.7 · Reference	1.3 · Reference
λ	7.6	6.5	8.5
θ	0.1°	-1°	3.5°
τ	1	0.7	1.3

5.6. Constraints

The constraints that are imposed relate to the structural aspects of the blade. The spanwise internal moments from *FAST* are converted into stresses and multiplied with the safety factors, while the tip deflection values from *FAST* are simply multiplied with the safety factors, by the *Post-processor* module, and fed into the optimizer as constraints. As described in Section 4.5.4, the stresses are calculated for six different spanwise locations and for each of the three materials : SNL Triax in the skins, UD-Carbon in the spar and UD-Glass fiber in the TE-Reinforcement. The maximum stress value (σ^{max}) for all the three materials is then used for constraint evaluation and compared with the respective material's Ultimate Compressive Strength (UCS), listed in Table 4.6. The tip deflection (δ_{tip}) is compared with the maximum allowable tip deflection for the NREL5MW reference turbine (δ_{tip}^{ref}), which is calculated to be 7.07 m.

$$\text{Constraints :}\quad \gamma_t \cdot \delta_{tip} \leq \delta_{tip}^{ref}$$

$$\gamma_t \cdot \sigma_{skin}^{max} \leq UCS_{skin}$$

$$\gamma_t \cdot \sigma_{spar}^{max} \leq UCS_{spar}$$

$$\gamma_t \cdot \sigma_{te-reinf}^{max} \leq UCS_{te-reinf}$$

The γ_t factor multiplied includes both, the partial safety factors and the additional γ_{st} for reduction in simulation time. The partial safety factors used are further elaborated next.

Partial safety factors

It is a common practice in the wind turbine industry to use partial safety factors in order to account for all the discrepancies in the design and manufacturing procedures. Ashuri [10] made a comparison of the partial safety factors given in two well known certification bodies, IEC and Germanischer Lloyd (GL).
For the materials used in manufacturing process, GL provides an extensive list of safety factors that comes up to a total of 2.94, whereas IEC provides a general material factor of 1.3 [29].

Tables 5.6 and 5.7 give the IEC partial safety factors that will be used for ultimate limit states and fatigue load analysis.

Table 5.6: Partial safety factor for ultimate load analysis

Type of safety factor	IEC
Loads, γ_f	1.35
Blade consequence of failure, γ_n	1.0
Materials, γ_m	1.3
Total	**1.755**

Table 5.7: Partial safety factors for fatigue load analysis

Type of safety factor	**IEC**
Loads, γ_f	1.0
Blade consequence of failure, γ_n	1.15
Materials, γ_m	1.2
Total	**1.38**

For the fatigue damage, a material factor of 1.2 for the IEC standards is assuming SN curve data with a confidence level of 95%, adopted from Resor [29]. For this research, partial safety factors of 1.755 for the ultimate limit states and 1.38 for the fatigue damage, based on the IEC standards, are used, so as to draw a quantitative comparison with the results of Resor [29], which was also based on the IEC standards. However, it should be noted that the optimum design changes with the use of a different standard. Also, the design driver is highly dependent on the selection of safety factors. For instance, the ultimate stresses or fatigue damage might emerge as the design driver for the blade when using GL standards while the tip deflection might still be the main design driver when using IEC standards. For the critical deflection analysis, the partial safety factors as shown in Table 5.8 are the same as those used by Resor [29].

Table 5.8: Partial safety factors for critical deflection analysis

Type of safety factor	**IEC**
Loads, γ_f	1.35
Blade consequence of failure, γ_n	1.0
Materials, γ_m	1.1
Total	**1.485**

The total safety factors used for the ultimate limit states and fatigue damage, including the simulation time safety factor, are summarized in Table 5.9.

Table 5.9: Total safety factor for constraints

Constraint	γ_{psf}	γ_{st}	γ_t
Tip deflection	1.485	1.1	1.63
Stress Check	1.755	1.1	1.93
Fatigue damage	1.38	-	1.38

5.7. Driver algorithm

The choice of the optimization algorithm can have a significant impact on the final design. A gradient based algorithm performs well for problems with a smooth function but it also faces an issue of getting stuck at a local minimum. To avoid this, multiple optimization runs with different starting points are usually carried out. Gradient free algorithms have a better ability of searching the entire design space and giving a global optimum. As they lack any gradient information, they are much slower than gradient based algorithms and require a higher number of function evaluations. Hybrid configurations also exist where at first, a gradient free algorithm is used to search the design space globally and its optimum is used as the starting point for gradient based algorithms.

The choice of an optimizer always requires a trade-off between various factors namely computational time, reliability, repeatability of results, etc. A lot of work has been done in the field of wind turbine optimization with different types of optimization algorithms. Jureczko *et al.* [37] optimized the blade internal composite structure using Genetic Algorithm. Ashuri *et al.* [35] performed a comprehensive optimization of the rotor and the tower in order to minimize the turbine LCOE, for which FAST was used in the framework with a bi-level optimization using a gradient based optimizer. However, Ashuri *et al.* [35] did not test the optimizer for different starting points as the aim of the authors was to execute the overall framework and not find a global optimum.

For the static model, the objective function and the constraints are expected to have a smooth response with respect to the design variables, making it suitable for a gradient based method as compared to the dynamic model. Also, a gradient free algorithm is expected to have better design space search capabilities. For this research, a gradient based Sequential Least Squares Quadratic Programming (SLSQP) algorithm and a gradient free Genetic Algorithm (GA) will be used and the differences in the behaviour of the optimizer for the given use case will be elucidated. Also, the results for a hybrid configuration, wherein the optimum results from GA are used as the starting point for SLSQP, will be presented in Appendix A.2. A brief introduction to the optimization algorithms is given in Sections 5.7.1 and 5.7.2.

5.7.1. SLSQP

The main ingredient of a gradient-based optimizer is the derivative of the objective function with respect to the design vector. Three key elements of a gradient based optimizer are :

1. *Search Direction:* The search direction enables the optimizer to take a step in the direction of the steepest descent, where the gradient of the objective function with respect to a particular variable x is given by $\frac{\delta f}{\delta x}$ and by $\nabla f(x)$, for a multivariate system.
2. *Step size:* Once the derivative is known, the optimizer moves in that direction with a particular step size, where too large a step size can cause the optimizer to cross the point of interest (minimum) and too small a step size would mean high computational times.

   After the search direction and the step size is known, the optimizer can guess the design vector for the next iteration, given by Equation 5.5 [38], where α^k is the step size and d^k is the search direction.

$$x^{k+1} = x^k + \alpha^k \cdot d^k \tag{5.5}$$

3. *Convergence:* Finally, the optimizer reaches a local or a global minimum when $\frac{\delta f}{\delta x} = 0$ or when the change in the objective function is less than the specified tolerance.

A gradient based optimizer requires the objective function and the constraint function to be continuous and differentiable. To check for the same, the response curves of the objective function and constraints with respect to the design variables are checked, using the new WINDOW framework with the dynamic model. The response of the objective function is found to be smooth with respect to most of the variables and as an example, the response of the LCOE with the tip speed ratio can be seen in Figure 5.4.

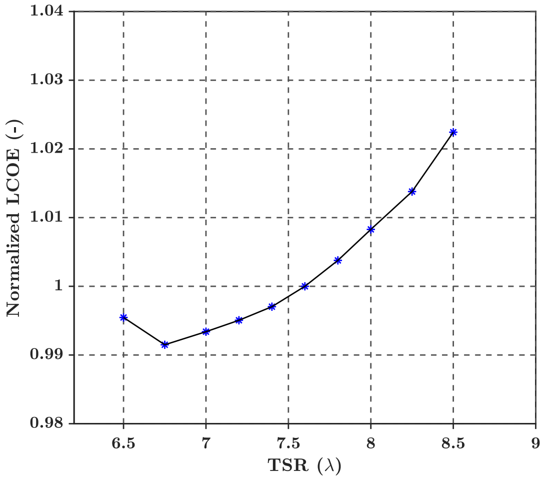


Figure 5.4: Smooth response of LCOE with respect to the tip speed ratio

However, the response of the constraints (stresses and tip deflection) with respect to few of the design variables is found to be erratic compared to the smooth response with the static model, as seen in Figure 5.5. With the dynamic model, the response of a new design to the wind profile depends on its mode shapes and cannot be predicted beforehand. Consequently, the stochastic nature of the constraints when using the dynamic model, is expected. Hence, a gradient based algorithm might not be the best option for all the cases and a need to explore a gradient free algorithm arises.

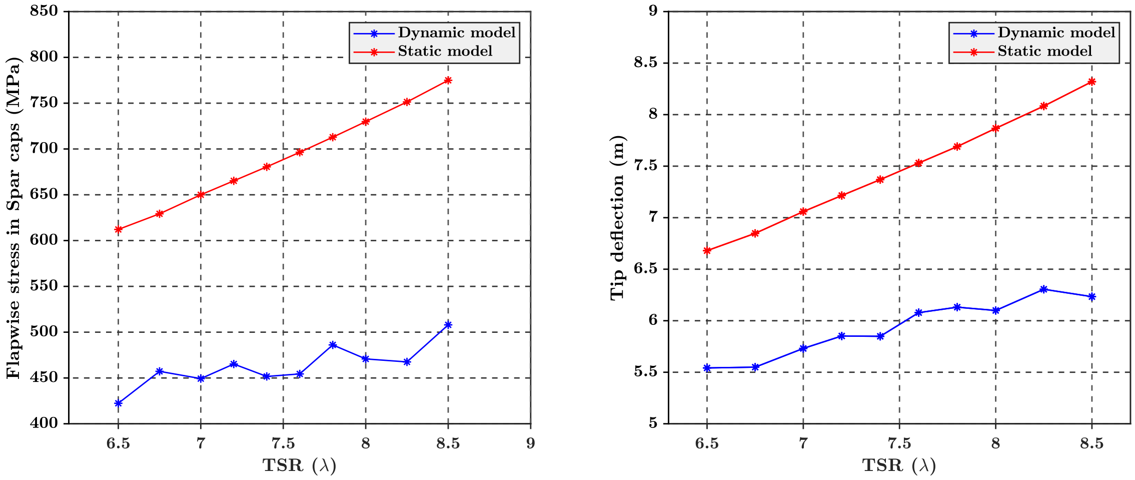


Figure 5.5: Difference in response of ultimate limit states with λ, for the static and dynamic model

5.7.2. Genetic Algorithm

For problems as complex as rotor optimization with 9 variables and some discontinuities of the constraints and the objective function with respect to the design variables, a gradient-free algorithm, like the GA may perform better than a gradient-based algorithm.

The Genetic Algorithm emulates Darwin's theory of evolution and ensures *'Survival of the fittest'*. When translated to an optimization problem, the fitness is measured in terms of the *fitness function*, $\phi(x)$. In the GA, the design variables are encoded into binary numbers, where the number of *bits* per variable are defined by the user. The number of bits define the accuracy desired with respect to a particular variable. Each of these encoded design variables, also called *genes*, are appended, to form one *chromosome*, where each chromosome represents one *individual* in the entire *population*. Also, in the GA, constraints are implemented in the form of a *penalty function*, wherein, if there is a constraint violation, a *penalty* is added to the *fitness function*. The overall *fitness function* is given by Equation 5.6, where *f(x)* is the objective function, p is the penalty co-efficient, N_g is the number of inequality constraints, g_i being the inequality constraint violation value, k is the penalty exponent, N_h is the number of equality constraints and h_j is the equality constraint violation value.

$$\phi(x) = f(x) + p \cdot \sum_{i=1}^{N_g} (\delta_i \cdot g_i^k) + p \cdot \sum_{j=1}^{N_h} |h_j|^k \tag{5.6}$$

Where $\delta_i = 0$ when the constraint is satisfied and 1 otherwise. It can be seen how the fitness function value increases whenever there is a constraint violation. For the given use case in this research, *f(x)* is the LCOE of the wind farm, g_i being the inequality constraint violation values for tip deflection and stresses, while p and k are user defined parameters.
Figure 5.6 displays a simplified example of a two-dimensional problem, where each of the variables are encoded into a 5 bit binary number. To better explore the design space, *Mutation* is introduced, which randomly flips a particular gene of an individual.

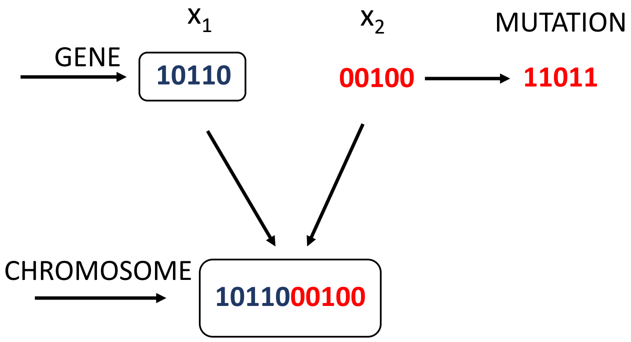


Figure 5.6: Genetic Algorithm basic terminology

At first, a random set of *individuals* is generated, based on the *population size* defined by the user. For each of these *individuals*, the objective function and the constraints are evaluated, based on which, each *individual* is assigned a *fitness* score. Based on these *fitness* values, the best *individuals* are selected to be the parents for the next generation. A *mating* process then takes place, wherein there is a *crossover* between the parents to produce the offsprings, or the new generation of *individuals*. The *crossover* method is usually defined by the user. Also, the new set of *individuals* may not necessarily be better than the previous generation, but it tends to reach the optimum set. To better explore the design space, *mutation* is introduced, defined by the user, where a gene of a particular *individual* is randomly changed. The overall process continues till the number of generations reach the maximum allowable generations or when the change in average fitness function value between the N and (N + 1)th generation is within the user defined tolerance. It explores the complete design space and hence, has a higher chance of reaching a global minima.

A Genetic Algorithm exhibits the following process :

1. *Initialization* : Random selection of individuals as the initial population, where the population size is defined by the user.

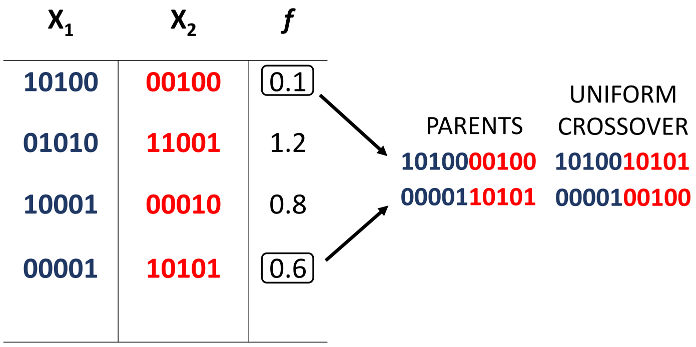
2. *Fitness function evaluation* : The entire model is run for each of these individuals and a fitness score is assigned based on the LCOE and constraint violation values.

3. *Selection* : A percentage of individuals with the best fitness score is selected for mating. Alternate selection methods also exist that are based on ranking.

4. *Crossover* : A crossover between these parents takes place, depending on the crossover probability. For a uniform crossover with 50% probability, the offspring takes half the genes from one parent and the rest from the second parent.

5. *Mutation* : If a mutation probability is specified by the user, a random individuals' gene is flipped (0 becomes 1 or 1 becomes 0). Mutation usually helps in better exploration of the design space.

The process of selection and uniform cross-over can be seen in Figure 5.7, where the best two individuals are selected as parents and a uniform crossover produces two off-springs for the next generation.



X$_1$	X$_2$	f
10100	00100	$\boxed{0.1}$
01010	11001	1.2
10001	00010	0.8
00001	10101	$\boxed{0.6}$

PARENTS	UNIFORM CROSSOVER
1010000100	1010010101
0000110101	0000100100

Figure 5.7: Genetic Algorithm selection and uniform crossover explained

5.8. Wind farm parameters

For this research, the existing framework developed by Moreno [21] will be used, in which the IEA Borssele wind farm is modelled. Figure 5.8 [39] shows the layout of the wind farm along with the water depth, where the regular configuration includes standard spacing rules for the downstream and crosswind spacing while the irregular configuration is a layout that was obtained by minimizing just the wake losses. For this research, the baseline irregular layout will be used for the optimization use case, for both, static and dynamic models.

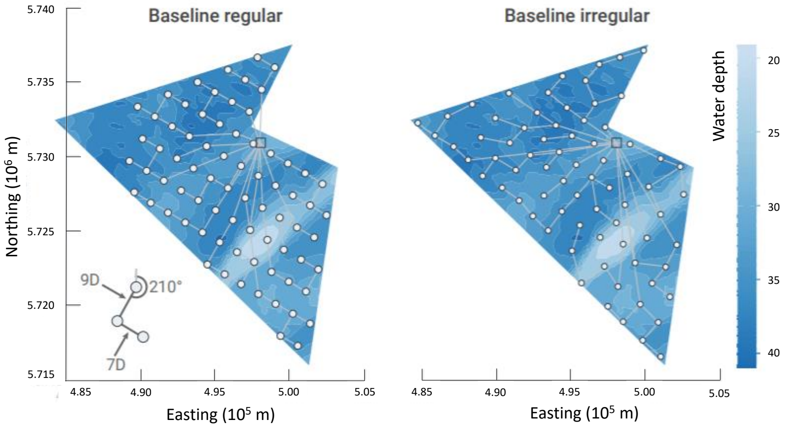


Figure 5.8: IEA Borssele irregular layout used for the research

All the modules used in WINDOW have a certain set of assumptions and a list of farm parameters has to be given as an input by the user. The assumptions made and the parameters used for all the optimization runs are listed below :

1. The support structure designed by the support structure module of WINDOW uses a monopile configuration.
2. A total of 74 turbines and 2 substations are placed in the farm.
3. A transmission efficiency of 95% and a collection electrical efficiency of 99% is assumed.
4. For the objective function evaluation, an interest rate of 7.5% and an operational lifetime of 25 years is assumed.

6

Results & Discussion

The discussion of the optimization results presented in this chapter are aimed at giving valuable insight into understanding the consequences of having a static/dynamic turbine model, coupled with the selection of the driver algorithm, in a wind farm level optimization framework. To quantify the same, the use case described in Section 3.1 is carried out for both the static RNA model and the new dynamic model. With each of these models, a gradient-based SLSQP and a gradient-free GA are tried out. Both the algorithms have been briefly explained in Chapter 5. Section 6.1 presents the optimized designs obtained using SLSQP, for both the static and the dynamic model, while Section 6.2 presents the optimized designs obtained using the GA. Section 6.3 summarizes the best designs obtained from all the four configurations and assesses the performance of each model-optimizer configuration. Lastly, in Section 6.4, the fatigue damage for the best designs resulting from the static and the dynamic model is evaluated. This is done in order to check regions sensitive to fatigue and to see if fatigue emerges as a design driver.

6.1. Gradient-based SLSQP

The SLSQP driver from the ScipyOptimizer written in openMDAO [40] is used for this research. For the convergence criteria, the number of iterations are limited to 50 and a tolerance is set to $1e^{-3}$. As an analytical form of gradient does not exist, the gradients are obtained using the finite difference method. The behaviour of the algorithm with the two models is first explored. Few typical characteristics of SLSQP observed during the initial optimization runs are listed below:

- For the dynamic model, the constraints (mainly tip deflection output of FAST) are not sensitive to extremely small changes (of the order $1e^{-6}$) in the design variables, which may have resulted in a loss of essential gradient information. This is mainly a limitation of the FAST output format.

- The optimizer has a tendency to get stuck at a local minimum or an infeasible region, which is highly dependent on the starting point of the optimization run. As a result, optimization runs with multiple starting points are performed.

- The design space explored by the gradient-based optimizer is quite limited.

It should be noted that multiple starting points with SLSQP are tried out but the two most relevant points are presented in Sections 6.1.1 and 6.1.2, where the NREL5MW design is used as the first starting point, resulting in the highest LCOE, and the second starting point presented is the point that gave the lowest value of LCOE. The two starting points are represented by sp1 and sp2. The results for the third starting point can be found in Appendix A.1. Also, the constraints presented are normalized with respect to the allowable limit.

6.1.1. Optimization results: Static model

To circumvent the problem of getting stuck at a local minimum, with the SLSQP, optimization runs with different starting points are tried out. The results of the optimized rotor design, using a static model, for two different starting points are presented in this section.

Starting point # 1

The first starting point uses the values for the NREL5MW reference turbine as the initial values. Table 6.1 compares the initial and optimum values of the design variables, some constraints, objective function and few other important parameters, where C_P and C_T are the power and thrust coefficients respectively. As mentioned previously, the stresses are calculated for all the materials throughout the blade span but only the stress with the lowest safety margin is presented. The lowest safety margin is observed for the stresses in Uni-Directional Carbon (UD-C) fibers placed in the spar caps, at 75% blade span.

Table 6.1: Initial and optimized results for Starting point # 1

	Parameter	**Units**	**Initial value**	**Optimum value**
Design variables	λ	-	7.6	6.5
	Fine Pitch	°	0.1	0.39
	τ	-	1	0.7
Constraints	Tip deflection	-	1.07	1
	Stress - spar	-	0.67	0.66
Others	Blade mass	kg	17956	13027
	RNA mass	kg	367480	308516
	C_P	-	0.481	0.459
	C_T	-	0.803	0.678
	Farm efficiency	-	0.919	0.931
	Farm energy production	kWh	1.35E+9	1.34E+9
	LCOE	€ct/kWh	8.934	8.491

The spanwise chord and distributions for the initial and optimum point can be seen in Figure 6.1.

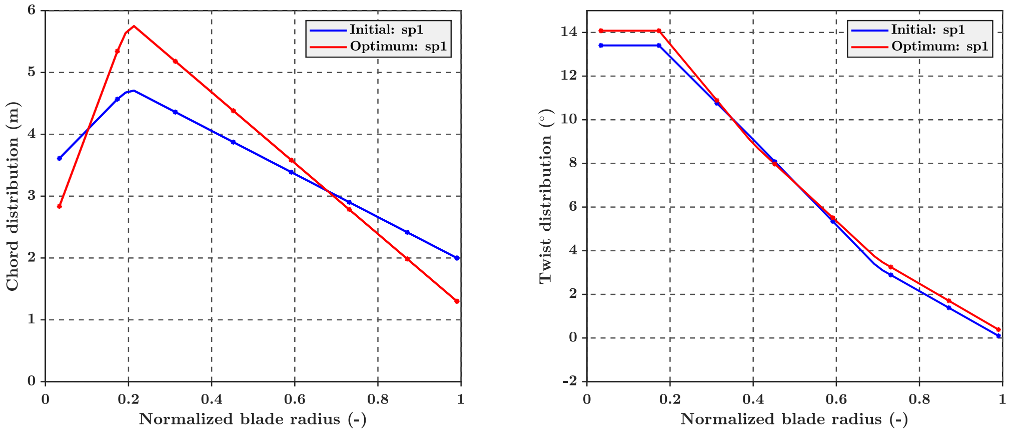
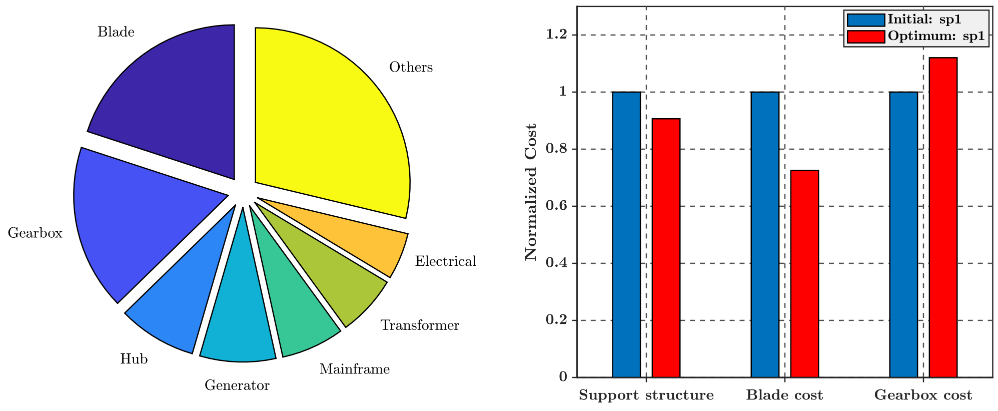


Figure 6.1: Chord and twist distribution comparison between initial and optimum point

The overall costs and energy production are the two factors that influence the LCOE of a wind farm. A reduced LCOE can either result from a higher energy production or a reduction in costs. The overall chord and twist distribution, along with the fine pitch angle and the operating tip speed ratio, govern the power and thrust coefficient values of the rotor. From Table 6.1, it can be observed that the optimum design results in a lower power and thrust coefficient. A lower C_T results in a higher farm efficiency due to lower wake losses in the farm, but a decreased overall energy production can be attributed to a lower C_P value, which has a significant impact on the power produced by an individual turbine at any given wind speed.

As the optimum design results in a lower energy production, the drop in the LCOE value can then be attributed to the reduced overall costs. The overall farm costs can be broken down into three main components: *total investment costs*, *Operations and Maintenance (O&M) costs* and *decommissioning costs*. The *total investment costs* include the capital cost of the turbine, the support structure, cabling and few other administrative costs; the *O&M costs* are a function of the energy production; the *decommissioning costs* are mainly dependent on the RNA mass.

Hence, it is clear that a reduced energy production results in lower *O&M costs* for the optimized design and a reduced RNA mass results in lower *decommissioning costs* . As the layout of the farm remains constant throughout the optimization process, the *cabling and electrical costs* of the farm do not change. The effect of the *total investment costs* can then be analyzed by zooming into the *support structure costs* and the *RNA costs*. The *support structure costs* are mainly driven by the rotor thrust, which leads to the bending of the support structure, and the RNA mass. A reduction in the rotor thrust and RNA mass for the optimized design, as shown in Table 6.1, result in lower *support structure costs*. An approximate component wise breakdown of the *RNA costs*, for the NREL5MW turbine, is shown in Figure 6.2a. It can be inferred that the *blade and the gearbox costs* are the major contributors to the overall *RNA costs*, where the *blade costs* are mainly driven by the rotor thrust and the *gearbox costs* are governed by the rotor torque. As the tip speed ratio for the optimal design decreases, a higher torque is required to produce the same power (rated power is constant), resulting in higher gearbox costs. A decrease in *blade costs* can be clearly attributed to a significant reduction in the blade mass. Figure 6.2b shows the difference in the initial design and the optimal design, for the main drivers of the *total investment costs*, where the costs are normalized with respect to the initial point costs.



(a) A component wise breakdown of the RNA costs for the NREL5MW turbine

(b) Component wise cost comparison for the initial and optimum design

Figure 6.2: Component wise cost breakdown for RNA followed by cost comparison between initial and optimum design

A clear drop in the blade mass and stiffness is evident from the reduction of the thickness factor to its lower bound and a reduced value of chord at most spanwise locations. However, the extreme tip deflection and stresses are still within limits due to an overall design that results in a lower loading, reflected in the thrust coefficient. To understand the same, Figure 6.3 illustrates all the forces acting on a blade element. The lift force (dL) acting on the element dr is given by Equation 6.1, where C_L is the lift coefficient, c is the chord and V_{res} is the resultant of the rotational velocity and the wind velocity (at the rotor). The drag force (dD), which is much lower than dL, can be obtained by simply replacing C_L with the drag coefficient (C_D).

$$dL = \frac{1}{2} \cdot C_L \cdot \rho \cdot c \cdot dr \cdot V_{res}^2 \tag{6.1}$$

The thrust force (dF_n) acting on the blade element dr is given by Equation 6.2, ϕ being the inflow angle.

$$dF_n = dL \cdot cos(\phi) + dD \cdot sin(\phi) \tag{6.2}$$

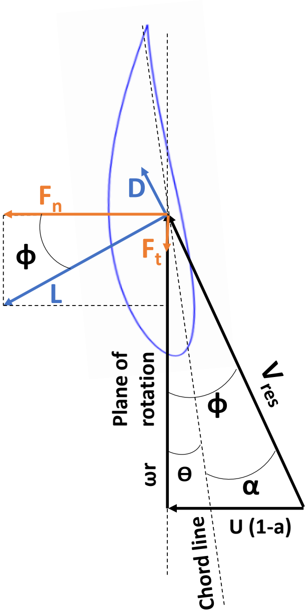


Figure 6.3: Forces acting on a given blade element

To analyze how the forces acting on an element for the initial and optimum design change, a section near the tip can be considered, where the rotational velocity component of the wind (ωr) has a much larger magnitude than the $U(1-a)$ component. A lower tip speed ratio for the optimum design reduces ωr, resulting in a much lower V_{res}. A lower ωr component increases the inflow angle. As the total twist close to the tip is nearly the same for both the designs, the angle of attack increases, resulting in a higher value of C_L. Also, the optimum design has a lower chord at the tip compared to the initial design. Consequently, a lower V_{res} and c contribute towards decreasing the lift force while a higher C_L contributes towards increasing the lift force. However, as the V_{res} is the most dominant factor, the overall lift force and hence the thrust, reduces, indicated by a lower C_T value in Table 6.1.

Starting point # 2

The second starting point presented is the point that resulted in the lowest value of LCOE and hence, the most optimum design.

Table 6.2: Initial and optimized results for Starting point # 2

	Parameter	Units	Initial value	Optimum value
Design variables	λ	-	8.36	6.55
	Pitch	°	-0.35	0.51
	τ	-	0.75	0.7
Constraints	Tip deflection	-	0.88	0.99
	Stress - spar	-	0.70	0.65
Others	Blade mass	kg	16249	13021
	RNA mass	kg	332292	307301
	C_P	-	0.474	0.456
	C_T	-	0.857	0.656
	Farm efficiency	-	0.911	0.934
	Farm energy production	kWh	1.33E+9	1.34E+9
	LCOE	€ct/KWh	9.008	8.438

The spanwise chord and distributions for the initial and optimum point can be seen in Figure 6.4.

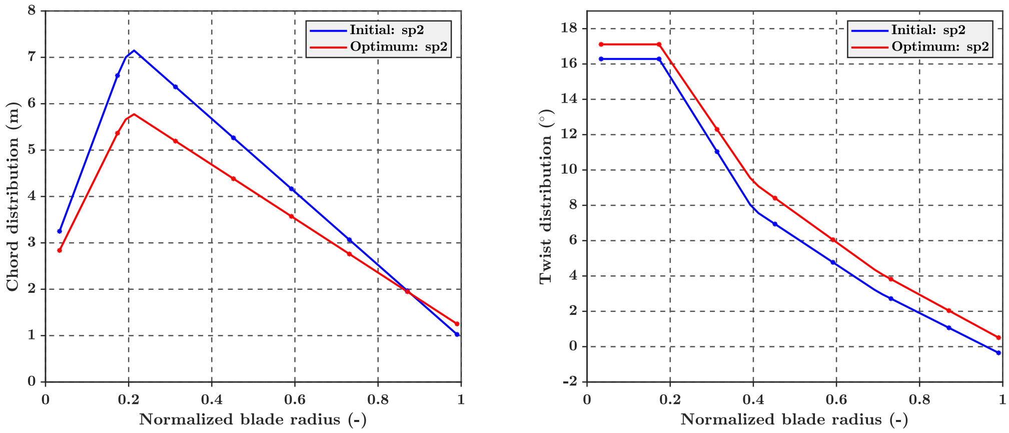


Figure 6.4: Chord and twist distribution comparison between initial and optimum point

As seen in Table 6.2, the arbitrarily chosen second starting point resulted in a lower LCOE as compared to the first starting point. The decrease of LCOE can be attributed to both decreased costs and an increased energy production. A lower C_T value for the optimum design leads to a significant increase in the overall farm efficiency, resulting in higher energy production.

As the tip speed ratio decreases, an increase in the torque is required and hence, the gearbox costs increase. Also, a decrease in the blade mass results in lower blade costs. As illustrated previously in Figure 6.2a, the blade has the highest share in the RNA costs and hence, governs the same. Also, a significant decrease in the support structure costs is observed due to a lower thrust coefficient for the optimum design. A relative comparison between the costs for the initial and optimum designs is shown in Figure 6.5.

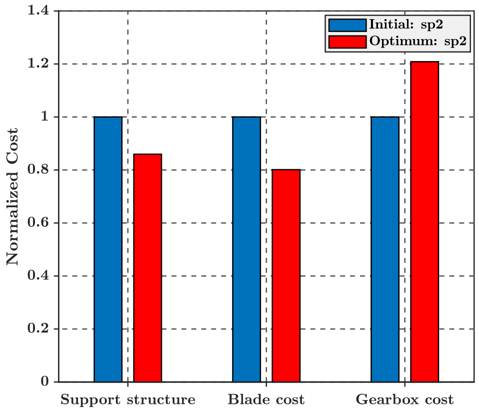


Figure 6.5: Component wise cost comparison for the initial and final design

In this case, the decrease in the blade mass is found to be the biggest contributor towards decreasing the LCOE. The decrease in blade mass led to a lower RNA mass, reducing the decommissioning costs. Also, the reduced RNA costs resulted in an overall decrease in the total investment costs.

As explained by Tanmay [16], the mass and stiffness of the blade have a dependency on the chord and the thickness factor, given by Equation 6.3 and Equation 6.4.

$$\mu \propto c^2 \cdot \tau \tag{6.3}$$

$$EI \propto c^4 \cdot \tau \tag{6.4}$$

Hence, as the mass and stiffness is sensitive to small changes in the chord, the chord is slightly increased, as seen in Figure 6.4, while the thickness factor is reduced to its lower bound to decrease the blade mass.

Comparing the optimum designs

A typical characteristic of the SLSQP algorithm is to get stuck at a local minimum and this can be better explained when the two optimum designs, resulting from different starting points, are compared. This trait can be better studied by analyzing the dependency of the design variables on the starting point.

1. The final chord distribution is found to be independent of the starting point and produces nearly similar results for both the starting points, which can be seen in Figure 6.6.
2. The twist distribution is found to be highly dependent on the starting point and hence, has a direct impact on the loading conditions and the aerodynamic performance of the rotor, as seen in Figure 6.6.

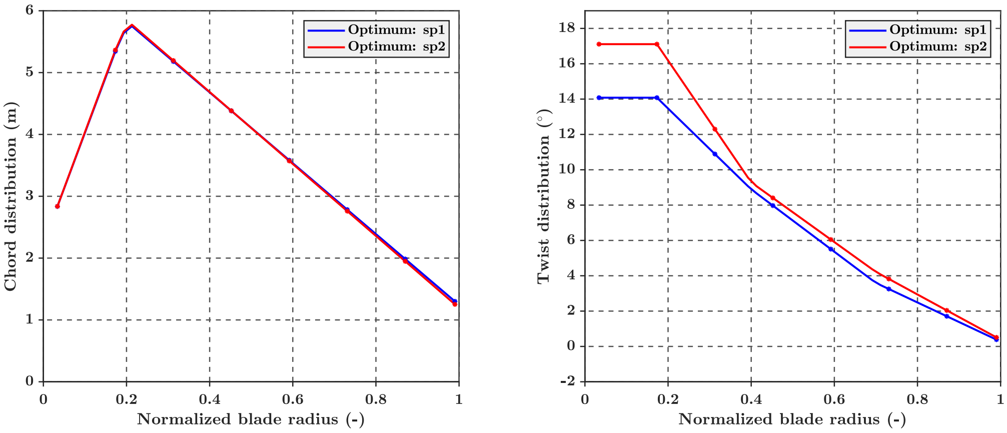


Figure 6.6: Chord and twist distribution comparison between the two optimum points

3. The fine pitch angle is also found to be dependent on the starting point.
4. The tip speed ratio (λ) always shows a decreasing trend with respect to the starting point. A decreasing λ reduces in a lower lift generation and hence a lower thrust. However, it also has a direct impact on the power coefficient and hence, the energy production.
5. The thickness factor (τ) always settles to its lower bound value to reduce the blade mass, as explained before using Equation 6.4.

6.1.2. Optimization results: Dynamic model

The results of the optimized rotor design with two starting points, using the dynamic model and SLSQP are presented in this section.

Starting point # 1

Similar to the static model, the first initial point presented is that of the NREL5MW design. A comparison between the initial design and the optimized design can be seen in Table 6.3.

Table 6.3: Initial and optimized results for Starting point # 1

	Parameter	**Units**	**Initial value**	**Optimum value**
Design variables	λ	-	7.6	6.5
	Fine Pitch	°	0.1	0.53
	τ	-	1	0.7
Constraints	Tip deflection	-	0.86	0.96
	Stress - spar	-	0.43	0.52
Others	Blade mass	kg	17956	12492
	RNA mass	kg	367480	305481
	C_P	-	0.481	0.470
	C_T	-	0.803	0.726
	Farm efficiency	-	0.919	0.927
	Farm energy production	kWh	1.35E+9	1.35E+9
	LCOE	€ct/kWh	8.934	8.524

The spanwise chord and twist distributions for the initial and optimum point can be seen in Figure 6.7, where no significant deviation of the twist, from the starting design, is observed. The reduction in C_T can be attributed to a decrease in the tip speed ratio, resulting in a lower overall lift.

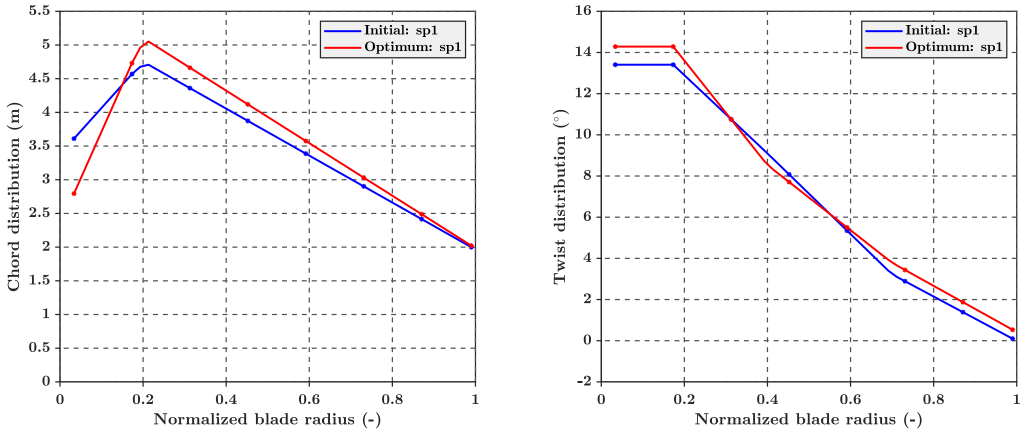


Figure 6.7: Chord and twist distribution comparison between initial and optimum point

As seen in Table 6.3, to minimize the LCOE, the optimum design favors a clear reduction in blade costs. A decrease in C_T increases the farm efficiency, which is compensated by a slight reduction in C_p, resulting in a near similar energy production.

Also, a reduction in the tip speed ratio increases the torque required and the gearbox costs. However, a reduction in LCOE can be observed because of a design that clearly favors a reduction in overall costs. A significant reduction in blade mass can be seen in Table 6.3, which also results in a reduction of the RNA mass. Also, the reduction in support structure costs can be attributed to a lower thrust force acting on it, indicated by a lower C_T value.

A comparison of costs between the initial and final design can be seen in Figure 6.8, where the drop in blade costs and support structure costs is apparent.

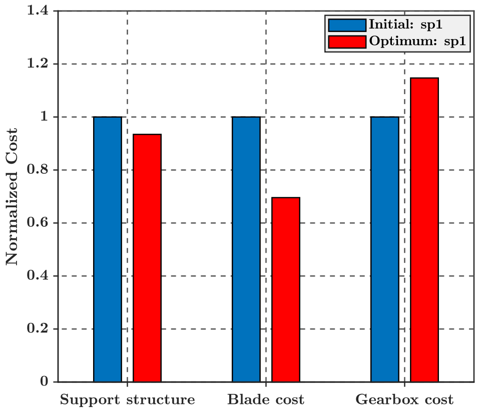


Figure 6.8: Cost comparison between the initial and optimum design

Starting point # 2

Similar to the static model, the second starting point presented is the point that resulted in the lowest value of LCOE and hence, the most optimum design among the different trials.

Table 6.4: Initial and optimized results for Starting point # 2

	Parameter	**Units**	**Initial value**	**Optimum value**
Design variables	λ	-	7.68	7.26
	Pitch	°	2.28	2.13
	τ	-	0.75	0.7
Constraints	Tip deflection	-	0.77	0.95
	Stress - spar	-	0.48	0.53
Others	Blade mass	kg	13985	12078
	RNA mass	kg	304117	292727
	C_P	-	0.460	0.456
	C_T	-	0.679	0.663
	Farm efficiency	-	0.931	0.933
	Farm energy production	kWh	1.34E+9	1.34E+9
	LCOE	€ct/KWh	8.442	8.370

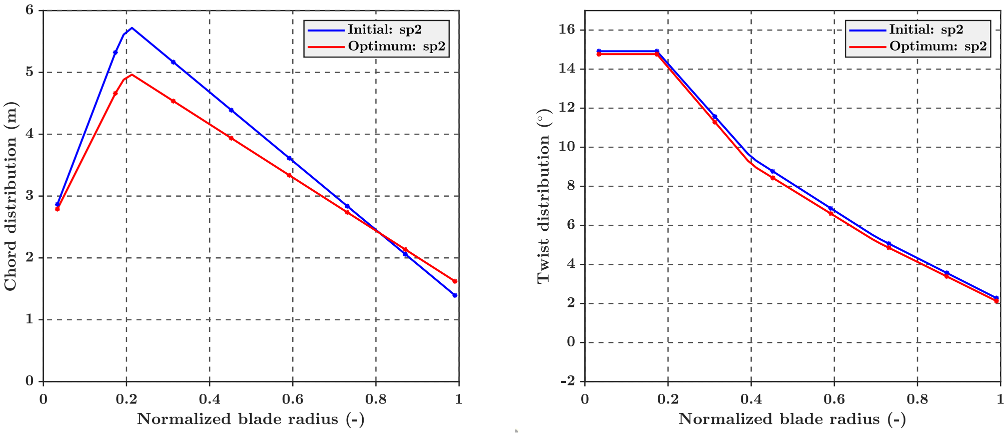


Figure 6.9: Chord and twist distribution comparison between initial and optimum point

The spanwise chord and distributions for the initial and optimum point can be seen in Figure 6.9. From Table

6.4, it can be clearly seen that the optimum design lies close to the initial design, indicating that the starting point is located near a local/global minimum. A similar value for the energy production is observed for the optimum design as the farm efficiency and C_P do not experience a significant change. A lower chord and tip speed ratio reduce the overall lift acting on the blade, resulting in a lower C_T. The only factor contributing to the reduction of LCOE is the blade mass. The lower thickness factor results in a lower blade mass and hence, RNA mass. This results in lowering of the blade costs and support structure costs.

The component wise cost comparison is shown in Figure 6.10, where a higher drop in the blade costs, compared to the increase in gearbox costs, can be seen.

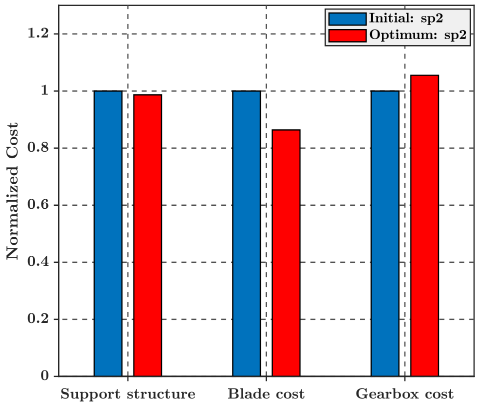


Figure 6.10: Component wise cost comparison for the initial and final design

Comparison of two optimums

Different starting points used resulted in completely different optimum designs, which is expected when using a gradient-based algorithm. Similar to the comparison made for the static model, the dependency of the variables on the starting point is evaluated for the results with the dynamic model.

The design vector shows a similar trend with both the static and the dynamic model. The *chord* distribution obtained using the dynamic model, also shows a low dependency on the starting point. The *twist* distribution and *fine pitch angle* are highly dependent on the starting point and do not show a significant deviation from the initial design. A decreasing trend for the *tip speed ratio (λ)* is observed, so as to reduce the overall lift force and hence, the thrust generated. The *thickness factor (τ)* always reaches its lower bound, irrespective of the initial point, to reduce the blade mass.

6.2. Gradient-free GA

As observed in Section 6.1, a gradient-based algorithm needs to be tested at multiple starting points and the optimum designs show a significant difference for different initial points. Also, many variables showed a strong dependency on the initial design point. Consequently, the results of a gradient-based algorithm for the given use case cannot be completely trusted and a need to test the models with a gradient-free algorithm arises.

Based on the observations from initial optimization runs, a few typical characteristics of how the GA behaves with both the models, are listed below:

- Constraint implementation is challenging as the value for penalty co-efficient and penalty exponent have to be tuned for a particular optimization problem.
- The convergence of the solution depends on the initial population size, number of generations allowed, crossover, mutation rate, etc. Hence, the solution depends on a lot of parameters, for which, no rule of thumb exists and the parameters have to be tuned for a particular type of a problem. A high pop-

ulation size with a low mutation probability always gives a better result but leads to a high number of generations to converge, increasing the computational time.

- Exploration of the design space is much better with the GA, as compared to SLSQP.
- A degree of randomness that exists with a GA could also lead to an unfair comparison. Starting with the same initial population always may not result in the exact same solution.

This section discusses the different user defined parameters specific to the genetic algorithm and the values selected for this research. Once these parameters are established, the optimum designs for the static and the dynamic model are discussed.

6.2.1. GA parameters

As mentioned above, there are certain problem specific parameters in the Genetic Algorithm that need to be tuned. The parameters that are dealt with in this research are listed below:

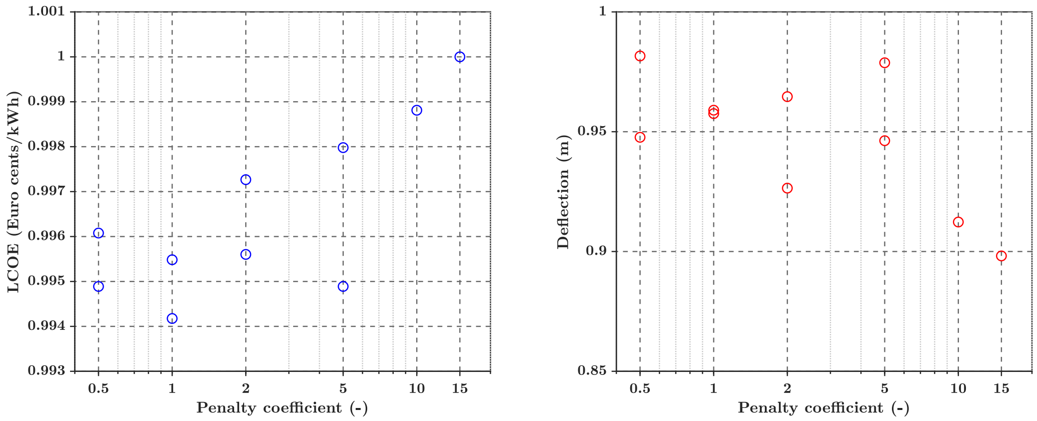
1. *Penalty coefficient*: The multiplication factor for the constraint violation, that gets added to the overall fitness function.
2. *Population size*: The number of individuals per generation used throughout the optimization process.
3. *Mutation rate*: The probability with which a gene of an arbitrarily selected individual is randomly flipped.

Multiple optimization runs are done to study the effect of these parameters on the objective function, constraints and the resulting rotor design. As running the entire optimization with a new value of a particular parameter is a computationally expensive process, the optimization runs are carried out using the static model. The final parameters are then used to run the optimization use case with the dynamic model as well. It should be noted that the tuned parameters selected may slightly differ for the dynamic model due to a difference in response for the two models. However, as the effect of these algorithm-specific parameters on the response of the objective function and the constraints is mainly dependent on the nature of the design problem and not the model fidelity per se, the values obtained using the static model are assumed to hold true for the dynamic model as well.

Penalty coefficient

The formulation of the penalty coefficient is already described before in Section 5.7.2, where an additional term $p \cdot \delta \cdot g^k$ is added to the objective function $(f(x))$, for every inequality constraint. The penalty parameter (p) has to be tuned in such a way that its product with the constraint violation (g) would add a value that is at least of the same order of magnitude as the objective function. Also, values that are too high or too low lead the optimizer to be stuck in the feasible or the infeasible region respectively. Penalties could be implemented either in the form of a static penalty coefficient, that remains constant throughout the optimization process, or a dynamic penalty coefficient, that changes over generations [41]. In this research, a static penalty coefficient, supported by the SimpleGADriver in openMDAO, is implemented, while the penalty exponent (k) holds a constant value of 1 for all the optimization runs. Also, a *population size* of 20 and a maximum number of 15 *generations* are used for all the runs made to determine the optimal penalty coefficient. A low population size of 20 is selected so as to try multiple penalty coefficients with the given computational resources. Also, from the initial runs, it is observed that the value of the best individual converges within 15 generations.

For the optimization runs with different penalty coefficients, the objective function and the constraint values are checked to evaluate the performance of the algorithm, as shown in Figures 6.11a and 6.11b. The tip deflection values are normalized with respect to the maximum allowable value, while the LCOE is normalized with respect to the highest value obtained.



(a) LCOE values of optimum designs for different penalty coefficients

(b) Tip deflection of optimum designs for different penalty coefficients

Figure 6.11: Variation of objective function and constraints of the optimum design, for different values of the penalty coefficient

It can be observed that the LCOE shows a decreasing trend as the penalty coefficient is lowered. As the LCOE values observed for large penalty coefficients (10 and 15) are quite high, only one optimization run is performed for each. Higher LCOE values for higher penalty coefficients are expected, as the optimizer has a higher chance of getting locked in the feasible region and not have any individuals in the infeasible region, that might have violated the constraint even by a small margin. This explains why lower penalty coefficients perform slightly better. Also, some variations in the LCOE value observed with multiple runs can be attributed to GA's inherent degree of randomness.

Another performance indicator for the penalty coefficient is the tip deflection constraint. The margin between the allowable deflection and the tip deflection for the optimum design is also a measure of how close the optimum design is to the constraint boundary. For high penalty coefficients, the optimizer directly discards any individual in the infeasible region and only contains individuals that are far away from the constraint boundary, in the feasible region. This can be avoided with having a high number of individuals so that at least few of them have a feasible solution that lies close to the constraint boundary. This can be seen in Figure 6.11b, where a penalty coefficient of 15 gives a solution with a tip deflection much lower than the allowable deflection value. However, for such a low population size, the differences in LCOE and constraint margin, for different penalty coefficients, are minimal. Based on the performance with respect to LCOE, distance from the constraint boundary and the magnitude added by the penalty coefficient relative to the LCOE value, a penalty coefficient of 0.5 is selected and used in all subsequent optimizations.

Population size

In this research, two population sizes of 20 and 40 are tried out to evaluate the differences in the resulting optimum design. Obviously, a higher number of individuals results in a better design because of better design space exploration. However, running the dynamic model later on, with a higher number of individuals would be computationally expensive. Hence, the deviation in the optimum results for these two population sizes is analyzed. The maximum number of generations are kept at 15 for both the cases. However, a population size of 40 did not converge within the given generation size and could have resulted in a better design if the maximum number of generations were increased.

Table 6.5 compares the optimum designs resulting from the two different population sizes of 20 and 40, represented by $Optimum_{20}$ and $Optimum_{40}$ respectively.

Table 6.5: Comparison of optimum designs for different population sizes

	Parameter	Units	$Optimum_{20}$	$Optimum_{40}$
Design variables	λ	-	8.11	7.47
	Pitch	°	2.34	2.34
	τ	-	0.7	0.7
Constraints	Tip deflection	-	0.95	0.91
	Stress - spar	-	0.65	0.61
Others	Blade mass	kg	13489	13489
	RNA mass	kg	297006	301691
	C_P	-	0.461	0.461
	C_T	-	0.687	0.673
	Farm efficiency	-	0.931	0.932
	Farm energy production	kWh	1.34E+9	1.34E+9
	LCOE	€ct/KWh	8.416	8.406

The chord and the twist distribution for the two optimum designs are shown in Figure 6.12.

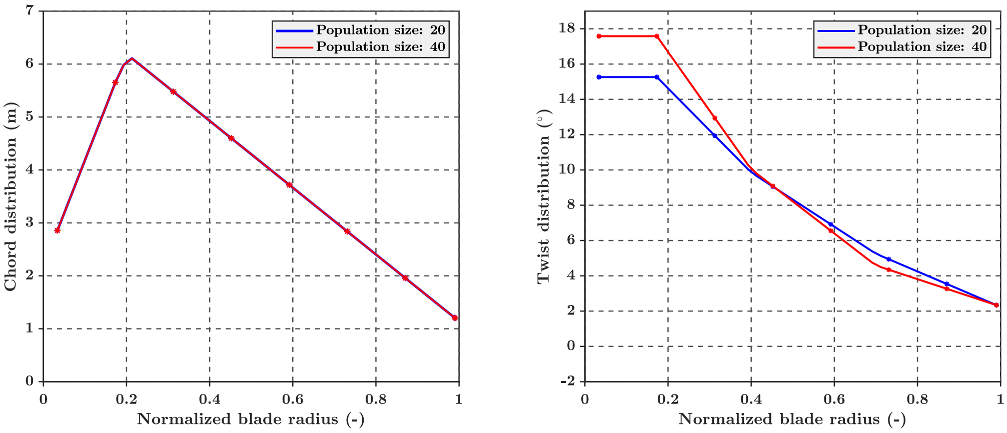


Figure 6.12: Chord and twist distribution comparison for different population sizes

It can be seen that the optimum design for a higher population size of 40 does not differ significantly. The exact same points for certain design variables can be seen because every variable can only take certain values between its bounds governed by the number of bits assigned. As a significant deviation in results is not observed, a population size of 20 is selected for all subsequent optimization runs.

Mutation rate

The mutation rate allows the algorithm to better search the design space but at the same time, also requires higher number of generations to converge. While mechanisms like cross-over are meant to converge towards one solution (the global optimum), mutation rate tends to diverge from same, at the cost of better design space exploration. Consequently, mutation rates are usually low for most optimizations using GA. Three different configurations tried out for the given problem are shown in Table 6.6, where the cases with a finite mutation rate are allowed 20 generations to converge. It should be noted that Case 1 does not include mutation and hence requires a lesser number of generations to converge compared to the other two cases.

Table 6.6: Cases with different mutation rates

Parameter	**Case 1**	**Case 2**	**Case 3**
No. of generations	15	20	20
Mutation rate	0	0.005	0.05

Optimum designs for the three mutation cases are presented in Table 6.7. The comparison of the chord and twist distribution is shown in Figure 6.13.

Table 6.7: Comparison of optimum designs for different mutation rates

	Parameter	**Units**	**Case 1**	**Case 2**	**Case 3**
Design variables	λ	-	8.11	7.85	8.11
	Pitch	°	2.34	2.77	3.06
	τ	-	0.7	0.7	0.7
Constraints	Tip deflection	-	0.95	1	0.98
	Stress - spar	-	0.65	0.65	0.68
Others	Blade mass	kg	13489	13098	13230
	RNA mass	kg	297006	295465	293733
	C_P	-	0.461	0.462	0.448
	C_T	-	0.687	0.685	0.658
	Farm efficiency	-	0.931	0.931	0.932
	Farm energy production	kWh	1.34E+9	1.34E+9	1.33E+9
	LCOE	€ct/KWh	8.416	8.396	8.431

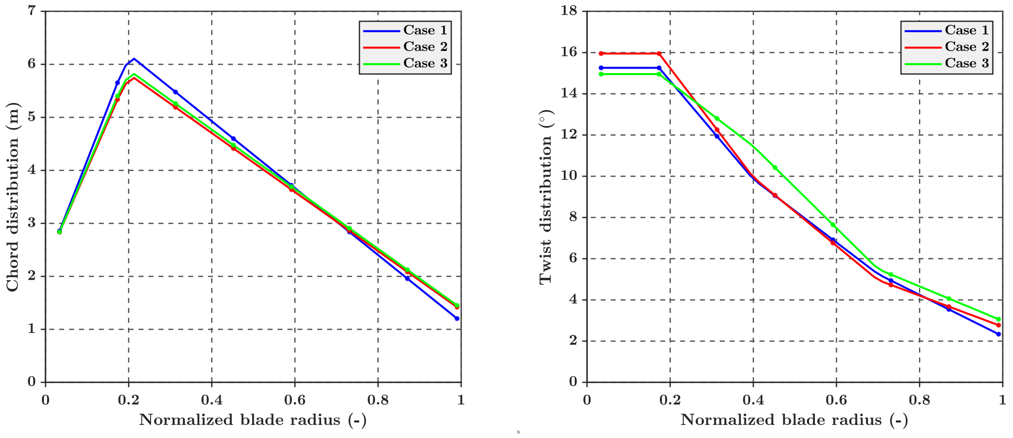


Figure 6.13: Chord and twist distribution comparison for different mutation rates

It is observed that a mutation rate of 0.5 % gives a lower LCOE value compared to the other two cases. Also, having some mutation results in slightly higher twist distributions, that are not observed before in earlier designs. As mentioned before during the analysis with a gradient-based algorithm, the chord distribution shows a similar trend for most cases. The differences observed even with a slight variation in design variables can be attributed to the fact that the pegged nodes are defined at the root, 70% and 90% of the blade span. Consequently, even a small difference in the last 2 values of chord, can lead to a significant difference in the overall chord distribution. The value of thickness factor and fine pitch angle observed is nearly the same for all the designs. Because of a lower LCOE achieved using a mutation rate of 0.5% along with better design space exploration, compared to case 1 and 3, a mutation rate of 0.5% is chosen to carry out the final optimization run.

A summary of the optimal parameters selected for this research is presented in Table 6.8.

The next two sections present the optimization results obtained using GA with the static and the dynamic model. Also, the results will be compared against the NREL5MW reference design.

6.2.2. Optimization results: Static Model

The results of the optimized rotor design with the static model and the tuned GA parameters are the same as Case 3 presented in Table 6.7 along with Figure 6.13. The optimum values are compared against the

Table 6.8: GA tuned parameters

Parameter	Value
Bits	5
Population size	20
No. of generations	20
Mutation rate	0.005
Penalty coefficient	0.5
Penalty exponent	1

NREL5MW reference design in Table 6.9 and Figure 6.14.

Table 6.9: Comparison of NREL5MW design vs optimum design obtained using GA with the static model

	Parameter	Units	NREL5MW	Optimum
Design variables	λ	-	7.6	7.85
	Pitch	°	0.1	2.77
	τ	-	1	0.7
Constraints	Tip deflection	-	1.07	1
	Stress - spar	-	0.67	0.65
Others	Blade mass	kg	17956	13098
	RNA mass	kg	367480	295465
	C_P	-	0.481	0.462
	C_T	-	0.803	0.685
	Farm efficiency	-	0.919	0.931
	Farm energy production	kWh	1.35E+9	1.34E+9
	LCOE	€ct/KWh	8.934	8.396

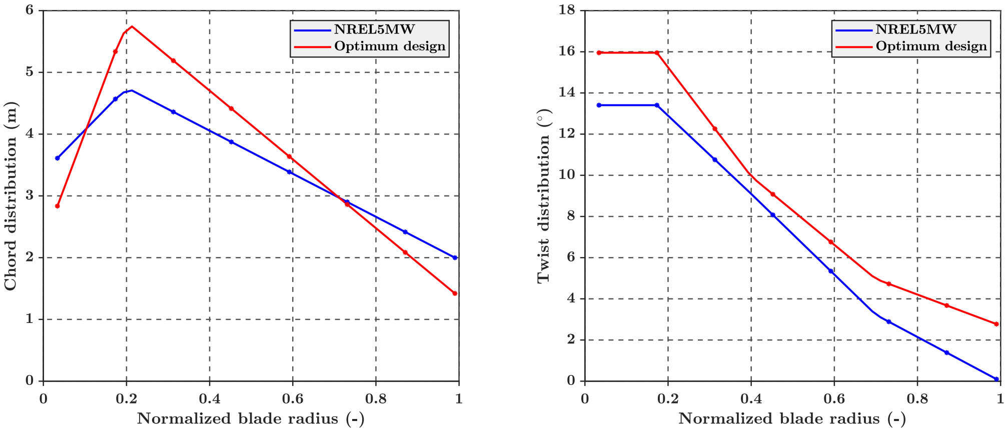


Figure 6.14: Chord and twist distribution comparison between NREL5MW and optimum design obtained using GA

It can be seen that with the GA, a significant decrease in the LCOE is achieved. The design shifts towards a rotor that has a lower loading (indicated by the lower C_T), while showing a notable decrease in the blade mass. This reduction in blade mass leads to a lower RNA mass, resulting in lower RNA costs. Also, a significant decrease in the support structure costs can be attributed to a lower C_T. It should be noted that a similar trend was also observed with SLSQP.

6.2.3. Optimization results: Dynamic Model

The GA parameters that are tuned with the static model, listed in Table 6.8, are also used to run the optimization with the dynamic model. Table 6.10, along with Figure 6.15 compares the optimum values with the NREL5MW reference design.

Table 6.10: Comparison of NREL5MW design vs optimum design obtained using GA with the dynamic model

	Parameter	Units	NREL5MW	Optimum
Design variables	λ	-	7.6	7.79
	Pitch	°	0.1	1.18
	τ	-	1	0.7
Constraints	Tip deflection	-	0.86	0.95
	Stress - spar	-	0.43	0.55
Others	Blade mass	kg	17956	12278
	RNA mass	kg	367480	284978
	C_P	-	0.481	0.463
	C_T	-	0.803	0.683
	Farm efficiency	-	0.919	0.931
	Farm energy production	kWh	1.35E+9	1.34E+9
	LCOE	€ct/KWh	8.934	8.350

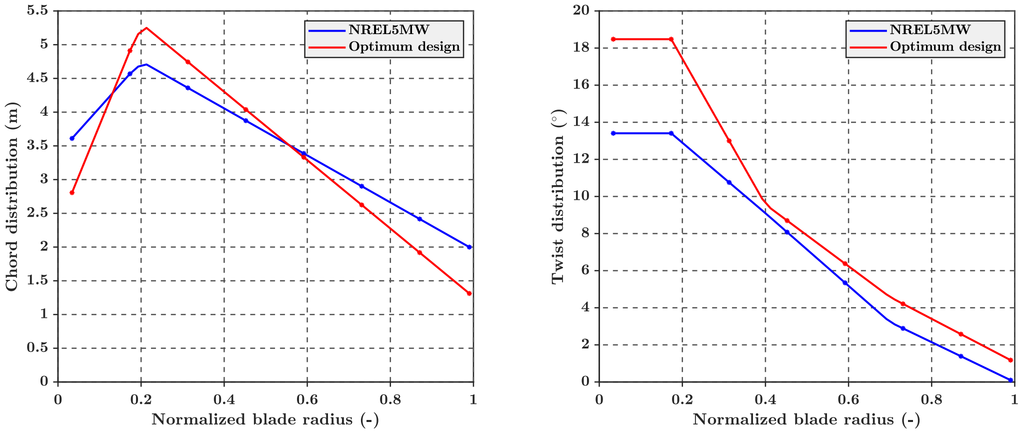


Figure 6.15: Chord and twist distribution comparison between NREL5MW and optimum design obtained using GA

The dynamic model with GA leads to a twist distribution and tip speed ratio that is unexplored before, owing to the design space search capabilities of the GA. A high tip speed ratio increases the lift and reduces the inflow angle. However, a higher twist reduces the angle of attack and hence the C_L. This reduced C_L, along with lower chord profiles, reduce the overall lift generated and hence, the thrust, indicated by the decreased C_T value in Table 6.10. However, it shows a trend similar to the static model wherein lower blade costs, gearbox costs, and support structure costs are observed.

6.3. Performance of the model-algorithm configuration

In the results presented so far, the static and dynamic model are optimized using both a gradient-based and a gradient-free algorithm. The performance of each of these model-optimizer configurations will be evaluated using some key performance indicators, in Section 6.3.1. After that, several design details resulting from the four model-optimizer configurations will be compared in Section 6.3.2.

6.3.1. Key performance indicators

The defined performance indicators used are enumerated below :

1. *Optimality* : A measure of the overall performance of the algorithm-model configuration. This can simply be based on the lowest value of objective function, which in this case is the LCOE.
2. *Starting point dependence* : A measure of how the design changes if the initial point is changed. A parameter that mainly depends on the optimization algorithm.
3. *Repeatability* : The extent to which the algorithm gives the same result every time it is run with the same intial conditions (For SLSQP) or settings (For GA), where a high repeatability indicates high reliability on the optimizer.
4. *Constraint margin* : A parameter that indicates how close the optimum design is to the design driving constraint boundary. In this use case, it is indicated by the difference between the tip deflection constraint value and the allowable tip deflection, normalized with the allowable tip deflection limit.

The behaviour of the optimization algorithms in combination with the turbine model, based on the optimization results presented earlier, can be summarized in a concise design matrix as shown in Table 6.11.

Table 6.11: Design matrix to summarize optimum designs for different algorithms and turbine models

	SLSQP		**GA**	
Static	Optimality (LCOE)	8.438	Optimality (LCOE)	8.396
	Starting point dependence	High	Starting point dependence	Low
	Repeatability	High	Repeatability	Low
	Constraint margin	Low (0 - 0.01)	Constraint margin	Low (0-0.01)
Dynamic	Optimality (LCOE)	8.370	Optimality (LCOE)	8.350
	Starting point dependence	High	Starting point dependence	Uncertain
	Repeatability	High	Repeatability	Low
	Constraint margin	Uncertain (0-0.05)	Constraint margin	Uncertain

Clearly, the GA is proven to have a better *optimality* for both static and dynamic models, compared to SLSQP. This can be attributed to GA's design space search capibilities which leads to combinations that are unexplored by SLSQP. SLSQP, being a gradient-based algorithm, shows a typical characteristic of getting stuck at a local minimum, closest to the *starting point* defined by the user. The GA implemented in this research takes one point out of the entire population as an input from the user. Although not as high as SLSQP, the optimum results with the GA show some dependence on the user defined point. In terms of *repeatability*, SLSQP performs better than the GA, for both, static and dynamic models as it returns the exact same results with the same initial conditions, every time it is run. For the GA, the results show a slight deviation for every run made with the same settings, which can be attributed to GA's inherent degree of randomness. However, if a higher number of generations are allowed for convergence, this trait of GA can be mitigated.

SLSQP with the static model performs the best when it comes to the *constraint margin*. This can be attributed to the smooth response of the constraints, with the static model, to the small steps in design variables taken by SLSQP, for gradient information. A similar trait is observed with the static model when using GA. It should be noted that the constraint margin mentioned for GA is after all the user defined parameters have been tuned to fit the given optimization problem. However, the stochastic nature of the dynamic model and its rough response result in a design that may be close to the constraint boundary or far away in the feasible region. The constraint margin for the dynamic model with GA is observed to be 0.05. As the dynamic model, coupled with GA, is run only once, the constraint margin and starting point dependence parameters are not defined with certainty.

6.3.2. Design details

The optimality performance indicator discussed in the design matrix is an end result of the overall design of the rotor. Hence, it is important to analyze not just the key performance indicators, but also the differences in the design resulting from each of these model-algorithm configurations. Table 6.12 compares the design parameters for all the four configurations.

Table 6.12: Design comparison of all algorithm-model configurations

	Parameter	Units	Static SLSQP	Static GA	Dynamic SLSQP	Dynamic GA
Design variables	λ	-	6.55	7.85	7.26	7.79
	Pitch	°	0.51	2.77	2.13	1.18
	τ	-	0.7	0.7	0.7	0.7
Constraints	Tip deflection	-	0.99	1	0.95	0.95
	Stress-spar	Mpa	0.65	0.65	0.53	0.55
Others	Blade mass	kg	13021	13098	12078	12278
	RNA mass	kg	307301	295465	292727	289432
	C_P	-	0.456	0.462	0.456	0.463
	C_T	-	0.656	0.685	0.663	0.683
	Farm efficiency	-	0.934	0.931	0.933	0.931
	Farm Energy production	kWh	1.34E+9	1.34E+9	1.34E+9	1.34E+9
	LCOE	**€ct/KWh**	**8.438**	**8.396**	**8.370**	**8.350**

The chord and twist distributions for all the four configurations can be seen in Figure 6.16.

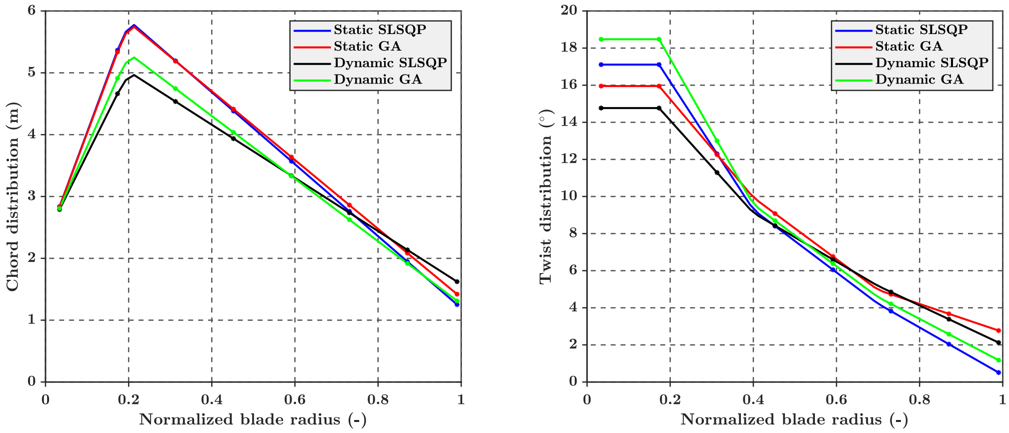


Figure 6.16: Chord and twist distribution comparison between all the four configurations

As shown in Section 6.1, the twist distribution for SLSQP is largely dependent on the starting point and does not deviate much from the same, while the GA results in completely different twist distributions due to its better search capabilities. However, this dependence on the algorithm can be seen for both the models where the difference in twist distributions resulting from SLSQP and GA is consistent. From Figure 6.16, a large difference in the chord distributions resulting from the static and dynamic model is apparent, while the dependence of chord distribution on the used optimizer is not observed. Table 6.12 shows a dependency between the blade mass and the choice of model, where the dynamic model results in a design with lower blade and RNA costs. As the thickness factor values are the same for the two models, the lower chord distributions from the dynamic model result in a reduced blade mass and a lower LCOE. As many differences in the configurations are dependent on the choice of the model, the variations can be better explained by analyzing the best designs (lowest LCOE) resulting from the static and the dynamic model, which are both obtained using the GA.

The consequences of the choice of turbine model can be better explained if the optimum design resulting from the low fidelity static model is run using the dynamic model and then compared with the optimum design resulting from the dynamic model. In this section, the optimum design resulting from the static model (using GA) will be referred to as $static_{opt}$ while the optimum design resulting from the dynamic model (using GA) will be referred to as $dynamic_{opt}$. Figure 6.17 compares the rotor thrust and the tip deflection of the $static_{opt}$, run with the dynamic model, and $dynamic_{opt}$.

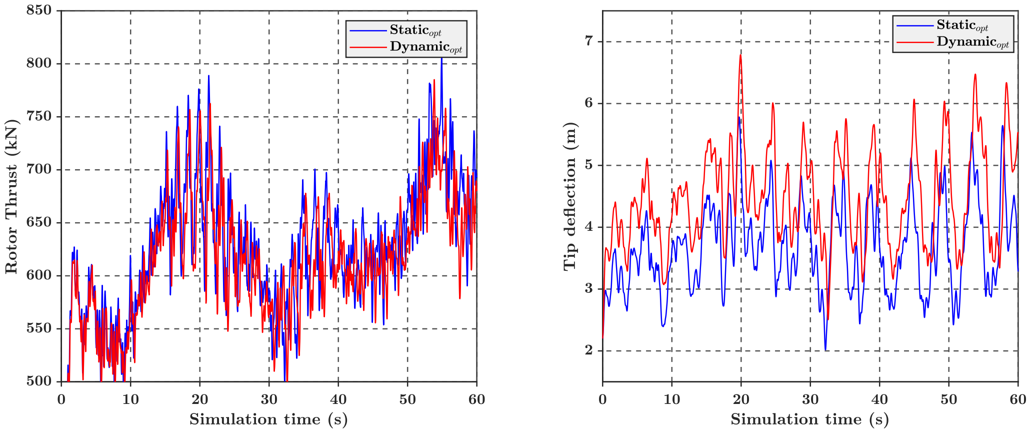


Figure 6.17: Rotor thrust and tip deflection comparison between optimum designs: Static vs Dynamic

When $static_{opt}$ is run using the dynamic model, it can be seen that a similar thrust is experienced by both the designs, which is apparent from the similar thrust coefficient values listed in Table 6.12. However, the tip deflection values for $static_{opt}$ are much lower than $dynamic_{opt}$, which can be attributed to the higher blade stiffness of $static_{opt}$. As the thickness factor for both the designs is the same, the higher blade stiffness of $static_{opt}$ can be explained by its larger chord distributions, seen in Figure 6.16. It is obvious that the static model estimates higher tip deflection values compared to the dynamic model, leading to an over-designed stiffer blade, as seen in Table 6.12. The higher chords produced by $static_{opt}$ are to meet the overestimated tip deflection values resulting from the static model.

The differences in the ultimate limit states are highlighted in Table 6.13, where the constraints are normalized with respect to their allowable limits.

Table 6.13: Results for running $static_{opt}$ with the static and dynamic model

$Static_{opt}$		
Model	**Tip deflection**	**Spar stress**
Static	1	0.65
Dynamic	0.80	0.53

The static model used for this research uses a factor of 1.5 to account for not modeling the dynamic effects. It is clear that $static_{opt}$, obtained by using the static model, leads to a stiffer blade and behaves differently when run with the dynamic model. A factor lower than 1.5 could also lead to an underestimation of the tip deflection, resulting in more flexible blades. Consequently, a direct implementation of a factor with the static model may not be valid for many designs. To summarize, it is apparent that rotor optimization using a dynamic model leads to a better evaluation of the constraints and hence, an accurate rotor design.

6.4. Fatigue damage

Based on the methodology to estimate fatigue damage, along with the material properties, elaborated in Section 4.5.4, the fatigue damages for the optimum designs, resulting from the static and dynamic model, are presented in this section. This is done in order to assess the implications of not including fatigue check in the constraint evaluation for all the optimization runs.

At first, the blade spanwise locations with maximum stresses are determined for all the materials. For the Uni-directional carbon fibers (UD-C), placed in the spar caps, the flapwise stresses are of interest while for the Uni-directional glass fibers (UD-G), in the TE reinforcements, the edgewise stresses are of interest, due to their respective positions with respect to the two neutral planes. Figure 6.18 shows the spanwise distribution of stresses for UD-C and UD-G respectively, for $dynamic_{opt}$.

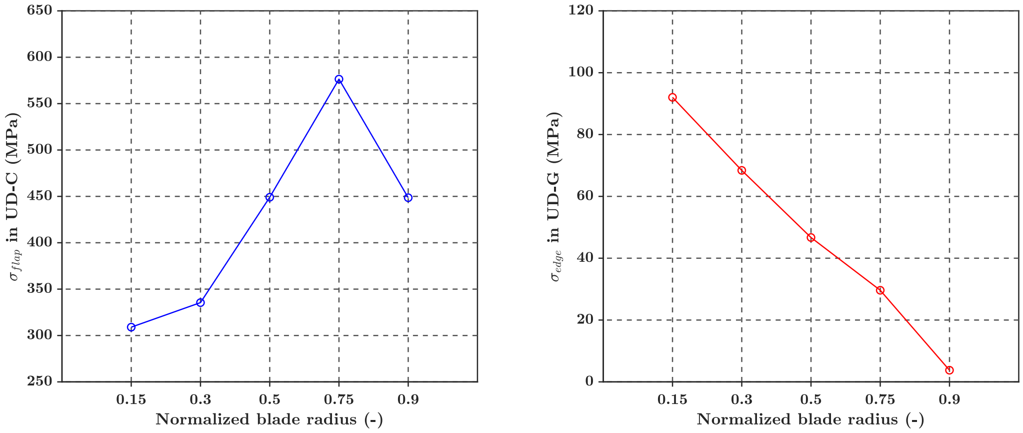


Figure 6.18: Stresses in UD-C and UD-G fibres

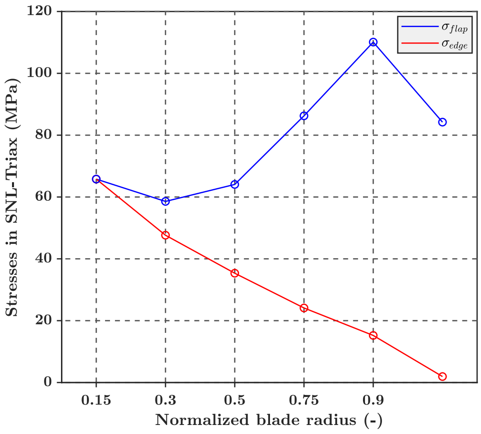


Figure 6.19: Stresses in SNL Triax

Figure 6.19 shows the flapwise stresses in the spar cap region and edgewise stresses in the trailing edge reinforcement region, for the SNL-Triax skins present throughout the blade span. From the stresses determined for all the materials, it can be noted that the maximum flapwise stress for UD-C and SNL-Triax is observed at 75% blade spanwise location while the edgewise stress for UD-G and SNL-Triax is observed at 15% blade spanwise location. As a result, these locations are selected for fatigue evaluation.

The mean wind speed at the hub height follows a Rayleigh distribution, according to the IEC standards.

The distribution is given by Equation 6.5, where an average velocity (V_{avg}) of 10 m/s is defined for the NREL5MW turbine based on its wind turbine class (I B).

$$P(V_{hub}) = 1 - e^{-\pi\cdot\left(\frac{V_{hub}}{2\cdot V_{avg}}\right)^2} \tag{6.5}$$

Table 6.14 and 6.15 show the fatigue damage for $static_{opt}$ and $dynamic_{opt}$ respectively, where Node 7 and Node 37 indicate 15% and 75% of the blade length respectively. As SNL Triax is present in the skins throughout the blade span, the fatigue damage is evaluated at the root and the two nodes. For the other two materials, the fatigue damage is only determined at the node where it experiences the maximum stress.

Table 6.14: Fatigue damage for optimum design obtained using the static model

$Static_{opt}$			
	SNL Triax	**UD-G**	**UD-C**
Root	3.73E-06	-	-
Node 7	3.10E-07	2.27E-04	-
Node 37	3.27E-06	-	0.0059

Table 6.15: Fatigue damage for optimum design obtained using the dynamic model

$Dynamic_{opt}$			
	SNL Triax	**UD-G**	**UD-C**
Root	3.97E-06	-	-
Node 7	6.43E-07	4.68E-04	-
Node 37	2.7E-06	-	0.0035

It is clear that UD-C and UD-G, in the spar caps and the TE reinforcement sections respectively, show higher fatigue damage values compared to the other materials. However, they are far below the allowable value of 1 for both the optimum designs, clearly indicating that fatigue is not the main design driver for the tested use case.

When optimizing the rotor in a wind farm level MDAO tool, including fatigue in the loop as a constraint would prove to be computationally expensive and can be avoided in the preliminary design phase. For the use case tested in this research, the tip deflection is found to be the main design driver and not fatigue. However, the same might not hold true for a different use case, wherein the layup definition is modified or larger blades are optimized. For larger blades, gravity loads can cause fatigue to be the main design driver for blade design [42]. Hence, it is yet important to check the blade design resulting from the optimization run for fatigue damage.

7

Conclusions & Recommendations

To recapitulate the research objective of this project, the aim was to provide a wind turbine designer with valuable insights into the consequences of the rotor model and optimization algorithm choice, on the final rotor design, in an MDAO tool for offshore wind farms. To accomplish the same, the milestones listed in Section 1.5 were achieved. This chapter presents the key findings of this research along with the implications of the assumptions, in Section 7.1, and recommendations for future research, in Section 7.2.

7.1. Key findings

It is clear that MDAO, as an approach applied in the wind energy industry, results in lower LCOE values compared to the existing sequential execution of each wind farm discipline. However, how the choice of turbine model and optimization algorithm influence the optimal rotor design and the resulting LCOE, has not been explored earlier. This research paints a clear picture of how the model-algorithm configuration affects rotor design by comparing the results for the same. To achieve the same, optimization runs with the existing static rotor model, and with the newly integrated dynamic model are performed. Each of these models is optimized using a gradient-based (SLSQP) and a gradient-free (GA) algorithm. The key findings can be classified into two broad categories : *Algorithm specific* and *Model specific.*

7.1.1. Algorithm specific

The differences in results, with the static and dynamic model, that can be attributed to the choice of the algorithm are listed below :

1. For the optimum values of twist (β), tip speed ratio (λ) and fine pitch angle (θ), a high dependency on the initial design point is observed with a gradient-based SLSQP . This starting point dependency leads the optimizer to the nearest local minimum instead of the global minimum. As a consequence, a gradient-based algorithm, for the given or a similar use case, has higher chances of resulting in a design that is a local optimum. Nevertheless, it does lead to a design that follows a trend similar to a gradient free algorithm, with respect to the loading, costs, and energy production. Hence, it can be used to develop insights into rotor optimization in an MDAO framework.
2. With both the models, the GA performs better in terms of LCOE, even with a low population size. Increasing the population size, while keeping the maximum number of generations constant, does not lead to a significant improvement in results for the static model. However, given the low population size and number of generations allowed to converge, the optimum values show a small deviation in results for different runs even with the same settings. Given this degree of randomness within a certain range, even higher penalty coefficients might give results similar to lower penalty coefficients.
3. When using the GA, having mutation is highly beneficial as it better explores the design space and also prevents the algorithm from converging into a local minimum. With respect to rotor design, the effect of mutation can be mainly observed in better exploration of twist distributions. However, a low

mutation rate is observed to give the best results. A higher mutation rate leads to a better diversity but prevents the optimizer from converging to a global optimum.

7.1.2. Model specific

The differences in results that can be attributed to the choice of the model, are listed below :

1. For the static model, the constraints show a smooth response with respect to the design variables, unlike for the dynamic model. With the dynamic model, the interaction between a given rotor design and the wind field leads to an erratic response of the constraints to the changes in the design variables. The effect can be directly observed in the constraint margin parameter, where a high degree of uncertainty can be seen with the dynamic model. With the GA, a high population size can mitigate this uncertainty in the constraint margin to a certain extent.
2. The optimal rotor design with the static model is highly sensitive to the amplification factor considered to account for not modelling the dynamic effects. This safety factor leads to an overestimation of the constraints, resulting in a blade design that is stiffer than necessary. However, it is difficult to establish a calibrated value for the same as it may differ from turbine to turbine. To conclude, when optimizing the rotor in an MDAO framework, the dynamic model performs better in terms of the optimality, giving a lower LCOE value compared to the static model (as used in this case study).

7.1.3. Concluding remarks

Designing a wind turbine rotor at a wind farm level captures all the inter-disciplinary interactions and gives deeper insights into the trade-offs between loading, costs and energy production. A commonality observed in all the model-algorithm configurations, relates to an optimum design with a lower thrust coefficient (C_T), indicating a design with lower loading. This lower loading can then allow a lower blade mass and hence, RNA mass. Also, a lower C_T leads to a significant reduction in the support structure costs and an increase in the energy production due to lower wake losses. Time reduction techniques for the dynamic model, similar to the ones used in this research, can be adopted to get reasonable results with limited computational resources. Also, the GA parameters used in this research may have to be tuned again for a different use case. Overall, integrating a dynamic model in an MDAO framework, coupled with a gradient-free algorithm, results in the most optimal rotor design.

7.2. Recommendations

To further enhance the capabilities of the existing WINDOW framework, with the newly integrated dynamic model, the recommended changes in the model are enumerated in this section.

1. Implementation of buckling check for the blade. The buckling check is an important criterion for the sandwich foam used in most parts of the blade. With buckling check being implemented, the lower bound limit on the thickness factor can be further lowered and its effect on chord distribution can also be analyzed.
2. Tuning the GA parameters using a higher population size (twice or thrice the chromosome length) with a higher number of generations to converge. This should then be compared with the results for the parameters obtained using a low population size.
3. Implementing a method to accurately estimate the stresses. This would be of importance especially for turbines with large blades, where the stresses due to gravity loading can be the design drivers. A better estimation of stresses would also give valuable insights into the fatigue damage for blade locations sensitive to fatigue.
4. Using the dynamic model for purposes other than constraint evaluation. For instance, the average power produced over a given time period can be determined, for each wind speed, and used for energy production calculations in the objective function.
5. Implementing a Campbell diagram check wherein for every design, the natural frequencies are determined and checked for resonance.

Bibliography

[1] IRENA (2018), "Offshore innovation widens renewable energy options: Opportunities, challenges and the vital role of international co-operation to spur the global energy transformation" (Brief to G7 policy makers), International Renewable Energy Agency, Abu Dhabi." Tech. Rep., 2018. [Online]. Available: www.irena.org

[2] Wind Europe, "Offshore Wind in Europe. Key trends and statistics 2018," Tech. Rep., 2018. [Online]. Available: https://windeurope.org/wp-content/uploads/files/about-wind/statistics/WindEurope-Annual-Offshore-Statistics-2018.pdf

[3] G. Giebel and C. B. Hasager, "An overview of offshore wind farm design," in *MARE-WINT*. Springer, Cham, 2016, pp. 337–346.

[4] S. S. Perez-Moreno, M. Zaaijer, C. L. Bottasso, K. Dykes, K. O. Merz, P.-E. Réthoré, and F. Zahle, "Roadmap to the multidisciplinary design analysis and optimisation of wind energy systems," in *Journal of Physics: Conference Series*, vol. 753, no. 6. IOP Publishing, 2016, p. 062011.

[5] B. Jonkman and J. Jonkman, "Fast v8. 16.00 a-bjj," *National Renewable Energy Laboratory*, 2016.

[6] T. J. Larsen and A. M. Hansen, *How 2 HAWC2 , the user ' s manual Risø-R-1597(ver. 4-5)(EN)*, 2014, vol. 1597, no. July.

[7] DNV GL, "Wind turbine design software | Bladed." [Online]. Available: https://www.dnvgl.com/services/wind-turbine-design-software-bladed-3775

[8] P. Fuglsang, C. Bak, J. Schepers, B. Bulder, T. Cockerill, P. Claiden, A. Olesen, and R. van Rossen, "Site-specific design optimization of wind turbines," *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, vol. 5, no. 4, pp. 261–279, 2002.

[9] C. Bottasso, "Integrated design optimization of wind turbines: Challenges, methods, applications," in *Functional lightweight construction for the wind energy requirements, requirements, benefits, DLR*, 2016.

[10] T. Ashuri, M. B. Zaaijer, J. R. Martins, and J. Zhang, "Multidisciplinary design optimization of large wind turbines—technical, economic, and design challenges," *Energy conversion and management*, vol. 123, pp. 56–70, 2016.

[11] T. Ashuri, G. J. Van Bussel, M. Zaayer, and G. A. Van Kuik, "Controller design automation for aeroservoelastic design optimization of wind turbines," in *3rd EWEA Conference-Torque 2010: The Science of making Torque from Wind, Heraklion, Crete, Greece, 28-30 June 2010*. European Wind Energy Association, 2010.

[12] S. S. Perez-Moreno, K. Dykes, K. O. Merz, and M. B. Zaaijer, "Multidisciplinary design analysis and optimisation of a reference offshore wind plant," in *Journal of Physics: Conference Series*, vol. 1037, no. 4. IOP Publishing, 2018, p. 042004.

[13] G. C. Larsen, H. A. Madsen, N. Troldborg, T. J. Larsen, P.-E. Réthoré, P. Fuglsang, S. Ott, J. Mann, and T. Buhl, "Topfarm-next generation design tool for optimisation," *Wind Energy*, vol. 13, pp. 86–99, 2011.

[14] K. L. Dykes, R. R. Damiani, P. A. Graf, G. N. Scott, R. N. King, Y. Guo, J. Quick, L. Sethuraman, P. S. Veers, and A. Ning, "Wind turbine optimization with wisdem," National Renewable Energy Lab.(NREL), Golden, CO (United States), Tech. Rep., 2018.

[15] P.-E. Réthoré, P. Fuglsang, G. C. Larsen, T. Buhl, T. J. Larsen, and H. A. Madsen, "Topfarm: Multi-fidelity optimization of wind farms," *Wind Energy*, vol. 17, no. 12, pp. 1797–1816, 2014.

[16] T. Tanmay, “Multi-disciplinary optimization of rotor nacelle assemblies for offshore wind farms: An agile systems engineering approach,” 2018.

[17] K. Dykes, P. Graf, G. Scott, A. Ning, R. King, Y. Guo, T. Parsons, R. Damiani, F. Felker, and P. Veers, “Introducing wisdem: An integrated system modeling for wind turbines and plant (presentation),” National Renewable Energy Lab.(NREL), Golden, CO (United States), Tech. Rep., 2015.

[18] P. J. Moriarty and A. C. Hansen, “Aerodyn theory manual (no. nrel/tp-500-36881),” *National Renewable Energy Lab., Golden, CO (US)*, 2005.

[19] J. Leonard, *Systems engineering fundamentals: Supplementary text.* DIANE Publishing, 1999.

[20] K. Dykes, R. Meadows, F. Felker, P. Graf, M. Hand, M. Lunacek, J. Michalakes, P. Moriarty, W. Musial, and P. Veers, “Applications of systems engineering to the research, design, and development of wind energy systems,” National Renewable Energy Lab.(NREL), Golden, CO (United States), Tech. Rep., 2011.

[21] S. Sanchez Perez Moreno, “A guideline for selecting mdao workflows with an application in offshore wind energy,” 2019.

[22] N. Kelley and B. Jonkman, “Overview of the turbsim stochastic inflow turbulence simulator: Version 1.21 (revised february 1, 2001),” National Renewable Energy Lab.(NREL), Golden, CO (United States), Tech. Rep., 2007.

[23] J. P. A. A. Blasques, “User’s manual for becas: A cross section analysis tool for anisotropic and inhomogeneous beam sections of arbitrary geometry,” 2012.

[24] G. S. Bir, “User’s guide to precomp (pre-processor for computing composite blade properties),” National Renewable Energy Lab.(NREL), Golden, CO (United States), Tech. Rep., 2006.

[25] G. Bir, “User’s guide to bmodes (software for computing rotating beam-coupled modes),” National Renewable Energy Lab.(NREL), Golden, CO (United States), Tech. Rep., 2005.

[26] S. Guntur, J. M. Jonkman, B. Jonkman, Q. Wang, M. A. Sprague, M. Hind, R. Sievers, and S. J. Schreck, “Fast v8 verification and validation for a mw-scale wind turbine with aeroelastically tailored blades,” in *34th Wind Energy Symposium*, 2016, p. 1008.

[27] T. Ashuri, G. J. Van Bussel, M. B. Zaayer, and G. A. Van Kuik, “An analytical model to extract wind turbine blade structural properties for optimization and up-scaling studies,” in *3rd EWEA Conference-Torque 2010: The Science of making Torque from Wind, Heraklion, Crete, Greece, 28-30 June 2010.* European Wind Energy Association, 2010.

[28] L. Wang, X. Liu, L. Guo, N. Renevier, and M. Stables, “A mathematical model for calculating cross-sectional properties of modern wind turbine composite blades,” *Renewable energy*, vol. 64, pp. 52–60, 2014.

[29] B. R. Resor, “Definition of a 5mw/61.5 m wind turbine blade reference model,” *Albuquerque, New Mexico, USA, Sandia National Laboratories, SAND2013-2569*, vol. 2013, 2013.

[30] D. T. Griffith and T. D. Ashwill, “The sandia 100-meter all-glass baseline wind turbine blade: Snl100-00,” *Sandia National Laboratories, Albuquerque, Report No. SAND2011-3779*, vol. 67, 2011.

[31] R. Nijssen, G. de Winkel, J. Peeringa, and R. DATE, “Wmc5mw laminate lay-out of reference blade for wp 3,” *Upwind Integrated Wind Turbine Design*, 2007.

[32] J. Jonkman, S. Butterfield, W. Musial, and G. Scott, “Definition of a 5-mw reference wind turbine for offshore system development,” National Renewable Energy Lab.(NREL), Golden, CO (United States), Tech. Rep., 2009.

[33] J. M. Jonkman and B. J. Jonkman, “Fast modularization framework for wind turbine simulation: full-system linearization,” in *Journal of Physics: Conference Series*, vol. 753, no. 8. IOP Publishing, 2016, p. 082010.

[34] M. H. Hansen, A. D. Hansen, T. J. Larsen, S. Øye, P. Sørensen, and P. Fuglsang, "Control design for a pitch-regulated, variable speed wind turbine," 2005.

[35] T. Ashuri, M. B. Zaaijer, J. R. Martins, G. J. Van Bussel, and G. A. Van Kuik, "Multidisciplinary design optimization of offshore wind turbines for minimum levelized cost of energy," *Renewable energy*, vol. 68, pp. 893–905, 2014.

[36] A. E. Smith and D. W. Coit, "Penalty functions," *Handbook of Evolutionary Computation*, vol. 97, no. 1, p. C5, 1995.

[37] M. Jureczko, M. Pawlak, and A. Mężyk, "Optimisation of wind turbine blades," *Journal of materials processing technology*, vol. 167, no. 2-3, pp. 463–471, 2005.

[38] D. Kraft, "A software package for sequential quadratic programming," *Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt fur Luft- und Raumfahrt*, 1988.

[39] S. S. Perez-Moreno, K. Dykes, K. O. Merz, and M. B. Zaaijer, "Multidisciplinary design analysis and optimisation of a reference offshore wind plant," in *Journal of Physics: Conference Series*, vol. 1037, no. 4. IOP Publishing, 2018, p. 042004.

[40] J. S. Gray, J. T. Hwang, J. R. R. A. Martins, K. T. Moore, and B. A. Naylor, "OpenMDAO: An Open-Source Framework for Multidisciplinary Design, Analysis, and Optimization," *Structural and Multidisciplinary Optimization*, 2019.

[41] A. Chehouri, R. Younes, J. Perron, and A. Ilinca, "A constraint-handling technique for genetic algorithms using a violation factor," *arXiv preprint arXiv:1610.00976*, 2016.

[42] P. Brøndsted and R. P. Nijssen, *Advances in wind turbine blade design and materials.* Elsevier, 2013.

A

Appendix

A.1. SLSQP: Starting point #3

Results for the third starting point with SLSQP are presented in this section, where the design shows a trend similar to the one observed with the first two starting points.

A.1.1. Static model

Table A.1 compares the initial and optimum values for the third starting point with the static model. Similar to the first two starting points, a decrease in the thrust coefficient (C_T) can be seen, which allows a lower blade mass and hence, a lower RNA mass.

Table A.1: Initial and optimized results for Starting point # 3

	Parameter	**Units**	**Initial value**	**Optimum value**
Design variables	λ	-	7.83	6.50
	Fine Pitch	°	2.28	1.92
	τ	-	1	0.7
Constraints	Tip deflection	-	0.96	1
	Stress - spar	-	0.57	0.64
Others	Blade mass	kg	16795	12723
	RNA mass	kg	324591	304603
	C_P	-	0.449	0.448
	C_T	-	0.650	0.636
	Farm efficiency	-	0.933	0.935
	Farm energy production	kWh	1.33E+9	1.33E+9
	LCOE	€ct/kWh	8.575	8.442

The spanwise chord and distributions for the initial and optimum point can be seen in Figure A.1.

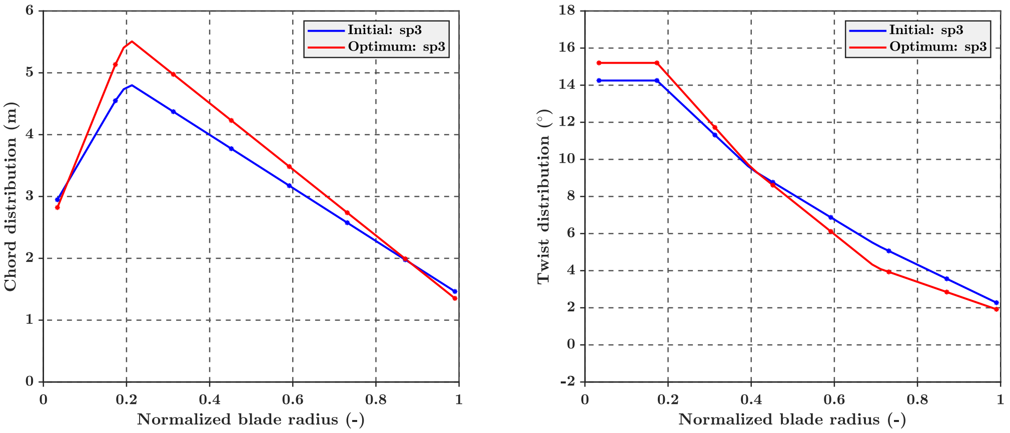


Figure A.1: Chord and twist distribution comparison between initial and optimum point

A.1.2. Dynamic model

Table A.2 compares the initial and optimum values for the third starting point with the dynamic model.

Table A.2: Initial and optimized results for Starting point # 3

	Parameter	Units	Initial value	Optimum value
Design variables	λ	-	7.83	7.58
	Fine Pitch	°	2.45	1.95
	τ	-	0.75	0.7
Constraints	Tip deflection	-	0.76	0.90
	Stress - spar	-	0.49	0.51
Others	Blade mass	kg	14230	12508
	RNA mass	kg	308006	292316
	C_P	-	0.431	0.452
	C_T	-	0.614	0.658
	Farm efficiency	-	0.935	0.933
	Farm energy production	kWh	1.31E+9	1.33E+9
	LCOE	€ct/kWh	8.510	8.389

The spanwise chord and twist distributions for the initial and optimum point can be seen in Figure A.2.

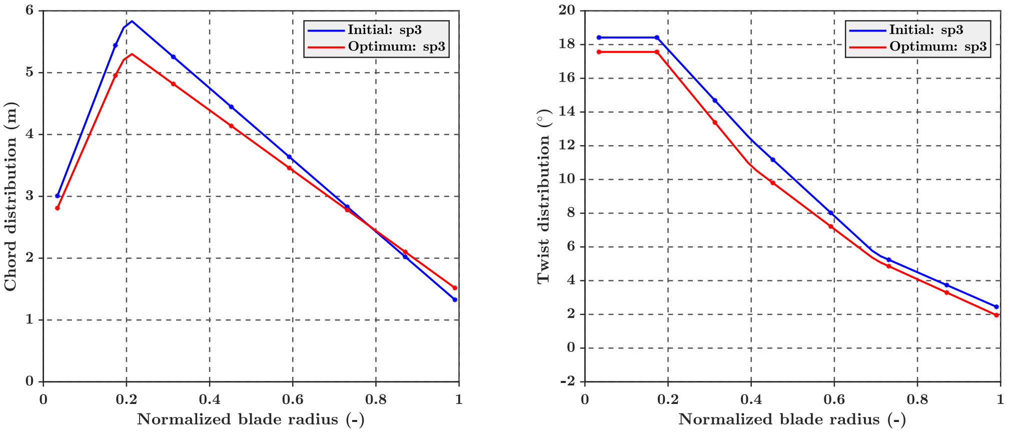


Figure A.2: Chord and twist distribution comparison between initial and optimum point

A higher tip deflection is observed because of an increase in the thrust coefficient. However, the reduction in LCOE can be observed due to a significant reduction in the blade costs and an increase in the annual energy production.

A.2. Hybrid configuration

For the hybrid configuration, optimum results obtained using the GA are used as a starting point for SLSQP. The results for the hybrid configuration, with the static and dynamic model, are presented in this section. It is observed that this configuration shows minimal improvement over the optimum results obtained using GA.

A.2.1. Static model

The results for the static model are summarized in Table A.3, where the optimum design resulting from GA is used as the starting point for SLSQP.

Table A.3: Improvement in results obtained using the hybrid configuration

	Parameter	Units	Initial value	Optimum value
Design variables	λ	-	7.85	7.58
	Fine Pitch	°	2.77	2.70
	τ	-	0.7	0.7
Constraints	Tip deflection	-	1	1
	Stress - spar	-	0.65	0.65
Others	Blade mass	kg	13098	13009
	RNA mass	kg	295465	296403
	C_P	-	0.462	0.462
	C_T	-	0.685	0.679
	Farm efficiency	-	0.931	0.932
	Farm energy production	kWh	1.34E+9	1.34E+9
	LCOE	€ct/kWh	8.396	8.391

The spanwise chord and twist distributions for the optimal design from GA and the hybrid configuration can be seen in Figure A.3.

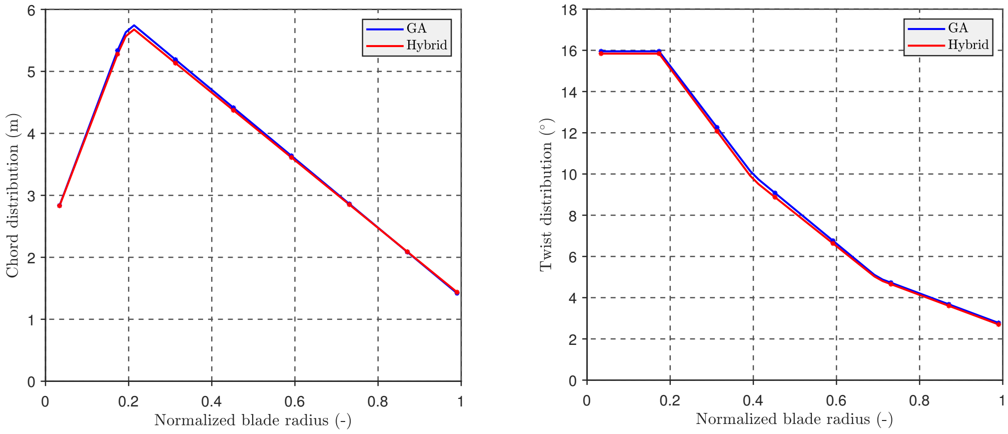


Figure A.3: Chord and twist distribution comparison between initial and optimum point

A.2.2. Dynamic model

The results for the dynamic model are summarized in Table A.4, where a minimal change in the design is observed.

Table A.4: Improvement in results obtained using the hybrid configuration

	Parameter	Units	Initial value	Optimum value
Design variables	λ	-	7.79	7.77
	Fine Pitch	°	1.18	1.17
	τ	-	0.7	0.7
Constraints	Tip deflection	-	0.95	0.96
	Stress - spar	-	0.55	0.56
Others	Blade mass	kg	12278	12277
	RNA mass	kg	289432	289566
	C_P	-	0.463	0.463
	C_T	-	0.683	0.683
	Farm efficiency	-	0.931	0.931
	Farm energy production	kWh	1.34E+9	1.34E+9
	LCOE	€ct/kWh	8.350	8.348

The optimal design from GA and the hybrid configuration produce nearly the same spanwise chord and twist distribution, as seen in Figure A.4.

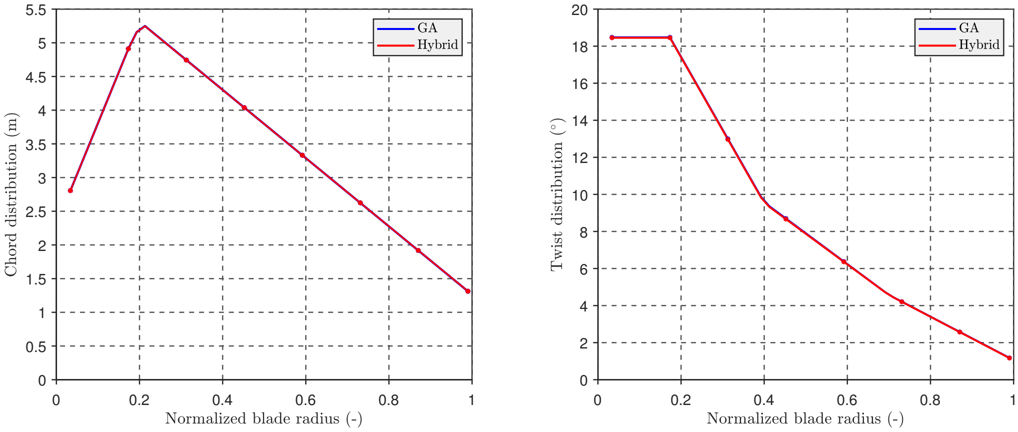


Figure A.4: Chord and twist distribution comparison between initial and optimum point