POLITECNICO DI MILANO

Facoltà di Ingegneria Industriale e dell'Informazione
Corso di laurea magistrale in Ingegneria Meccanica

Comparison of optimization methods for wind turbine blades

Relatore: Prof. Alberto Zasso

Tesi di Laurea di:
Luca Bazzan Matr. 799447

Abstract

The design and optimization of wind turbine rotors is a complex and challenging problem. Both aerodynamic and structural issues must be taken into account to reach the most balanced solution and a large set of constraints must be considered to ensure the feasibility of the solution. The strict relationship between aerodynamic and structural behavior of the rotor blades poses issues on the design procedure because the two aspects must be considered together. In the context of research carried out by TU München's Lehrstuhl für Windenergie and Politecnico di Milano's POLI-Wind, the aim of this work is to build a code for the integrated aero-structural optimization of wind turbines and then to compare for this code different solving algorithms. At each iteration the code selects a blade shape in terms of chord distribution, then a twist optimization process is performed and eventually a structural optimization is carried out and the cost of energy produced by the wind turbine is calculated. Dynamic and static aeroelastic simulation are performed for loads and annual energy production estimation. Acting like this, the coupling between aerodynamic and structural aspects should be ensured.

Riassunto

Introduzione

La progettazine integrata aero-strutturale delle pale e rotori eolici sta riscuotendo sempre maggiore interesse sia nel campo della ricerca sia in quello industriale. Appare chiaro nel panorama odierno che la possibilità di dotarsi di potenti strumenti computerizzati nel processo di progettazione delle macchine eoliche, porti evidenti benefici in termini economici per la tecnologia stessa. Particolarmente interessante e studiato, è il problema di costruire macchine il cui costo dell'energia prodotta sia sempre minore. Evidentemente il processo deve passare attraverso una valutazione di aspetti sia aerodinamici sia strutturali, e ciò può essere fatto con differenti approcci. Si può procedere separando i problemi e affrontandoli in sequenza, ovvero operando in successione un' ottimizzazione aerodinamica e una strutturale, oppure considerando l'accoppiamento che esiste tra i due aspetti e integrando l'ottimizzazione in un unico ciclo. In questo contesto il presente lavoro si prefigge l'obiettivo di sviluppare un codice in ambiente MATLAB capace di gestire autonomamente un processo di ottimizzazione integrata attraverso l'uso dell'algoritmo di ricerca più adatto, da ricercarsi all'interno di una cerchia di candidati. Per lo sviluppo del programma, seguendo la logica dei precedenti lavori di POLI-Wind a Milano, il comportamento delle turbine eoliche è stato studiato attraverso software di simulazione aeroelastica per assicurare la massima affidabilità dei risultati nonostante questa scelta porti inevitabilmente ad un innalzamento dei tempi computazionali.

Sviluppo del codice

Il presente lavoro prende spunto dalle precedenti ricerche del POLI-Wind esplorando la possibilità di creare un codice iterativo che assicuri il completo accoppiamento degli effetti aerodinamici e strutturali nel progetto delle pale eoliche. Il cuore del progetto è l'algoritmo iterativo che sceglie di volta in volta la forma della pala in termini di distribuzione di corda, da sottoporre all'ottimizzazione strutturale.Per ogni pala così valutata, viene stimato il costo dell'energia prodotta che diventa la cifra di merito da minimizzare nell'ottimizzazione. La scelta dell'algoritmo è fondamentale per assicurare:

- Affidabilità del risultato: non tutti gli algoritmi sono in grado di ottimizzare il profilo in termini di costo dell'energia
- Velocità: i tempi computazionali devono rimanere per quanto possibile ridotti.

La scelta dell'algoritmo avviene rispetto a tre candidati: un metodo di ricerca globale, un modello surrogato e un metodo al gradiente. Ogni algoritmo applica la propria logica di variazione ai parametri aerodinamici, dopodichè per ogni pala viene ottimizzata la struttura e si stima l'energia prodotta annualmente. L'ottimizzatore strutturale e il processo di valutazione dell'energia prodotta, sono frutto dei precedenti lavori del POLI-Wind così come gli script per la simulazione e stima dei carichi agenti sulle pale. Il calcolo della cifra di merito che viene usata dall'algoritmo per guidare il processo di ottimizzazione, ha come variabili di input il peso della pala e l'energia prodotta dalla stessa.

Risultati

Il codice di ottimizzazione integrata è stato sviluppato e diversi algoritmi sono stati testati. Si è notato come il metodo di ricerca globale sia risultato il migliore compromesso tra affidabilità della soluzione e tempi computazionali. Il modello surrogato è apparso essere troppo esigente dal punto di vista degli oneri computazionali mentr il metodo al gradiente non è riuscito a ottenere risultato alcuno. Peraltro é emerso chiaramente il legame profondo che intercorre tra parametri aerodinamici e strutturali, valutabile in maniera schematica tramite il parametro di solidità. Si è notato una certa tendenza del modello di costo a enfatizzare piccoli miglioramenti dell'energia prodotta piuttosto che grosse riduzioni di massa.

Sviluppi futuri

Alla luce dei risultati sopra elencati, si delineano diversi possibili sviluppi futuri del codice:

- Inserire il parametro di solidità della pala nell'ottimizzazione aerodinamica per correlare sinteticamente la reciproca influenza che hanno la variazione delle due famiglie di parametri (aerodinamici e strutturali)
- Snellire la comunque onerosa procedura computazionale ad esempio parallelizzando maggiormente i processi di calcolo per ottenere risultati in tempi minori.

- Scegliere un miglior modello per la valutazione della cifra di merito la quale attualmente viene valutata senza considerare l'influenza che la variazione dei carichi sperimentati dalla macchina ha su componenti diversi dalle pale.

Contents

1 Introduction **2**
- 1.1 Introduction . . . 2
- 1.2 Aim of the Work . . . 3
- 1.3 State of Art . . . 4
- 1.4 Considerations . . . 6

2 Description of the problem **8**
- 2.1 Nature of The Problem . . . 8
- 2.2 Tools used . . . 10
  - 2.2.1 Cp-Lambda . . . 10
  - 2.2.2 ANBA . . . 10
  - 2.2.3 Cp-Max . . . 11
- 2.3 Variables . . . 12
- 2.4 Constraints . . . 14
- 2.5 Objective . . . 16
- 2.6 Scheme of The Code . . . 17
  - 2.6.1 New code . . . 17
  - 2.6.2 Twist Optimizator . . . 18
  - 2.6.3 Non structural masses update . . . 18
  - 2.6.4 Loads calculation . . . 18
  - 2.6.5 Structural Optimization . . . 19
  - 2.6.6 AEP Calculation . . . 19
  - 2.6.7 Cost of energy calculation . . . 20

3 Algorithms review **21**
- 3.1 Patternsearch . . . 21
  - 3.1.1 Operation Mode . . . 21
  - 3.1.2 Pros . . . 23
  - 3.1.3 Cons & Problems . . . 23
- 3.2 SUMO . . . 24
  - 3.2.1 Operation Mode . . . 24
  - 3.2.2 Pros . . . 27
  - 3.2.3 Cons & Problems . . . 27

3.3 Fmincon 28
  3.3.1 Operation Mode 28
  3.3.2 Pros 29
  3.3.3 Cons & Problems 30

4 Results **31**
4.1 Patternsearch 31
4.2 SUMO 34
4.3 Fmincon 37
4.4 Influence of Solidity 40
4.5 Influence of Cost Model 42

5 Conclusions **44**
5.1 Future developments 45

List of Figures

1.1 [10] optimization procedure. 5
1.2 [5] optimization procedure. 6

2.1 Internal structure scheme of the baseline blade 11
2.2 Working scheme of the code 17

3.1 Patternsearch workflow 22
3.2 SuMo Toolbox workflow 26
3.3 Value of the measure at each model creation 28
3.4 FminCon workflow 29

4.1 Result of Patternsearch optimization 32
4.2 Comparison between non structural masses in baseline and optimized blade 33
4.3 Mass and stiffness distributionof the Patternsearch solution compared to the baseline 33
4.4 3x3 Kriging Model 36
4.5 5x5 Kriging Model 36
4.6 Refined Kriging model 37
4.7 Blade Mass and AEP response surface 38
4.8 Result of fmincon with Multistart 39
4.9 Comparison between the optimal aerodynamic solution of*Cp-Max* and the result of the Patternsearch iteration 41
4.10 Tip displacement for different blade solidity 42

List of Tables

2.1 Position of the optimization station along the span 13
2.2 Comparison between masses calculated by the code before and after the structural optimization and masses extimated by the NREL scaling model in baseline and advanced configuration . 20

4.1 Comparison between baseline and patternsearch solution . . . 34
4.2 Results of SUMO without fatigue analysis 35
4.3 Gains of the optimization variables respect to the baseline configuration 38

Chapter 1

Introduction

1.1 Introduction

Wind energy has been used by the human kind as a source of power for boats and wind mills since the ancient times. The development of the electricity and the availability of low cost engines driven by fossil fuels, made wind technologies fall into disuse. However, the recent necessity of increasing the quote of green energy produced and the concerns about a future decrease of oil availability promoted the interest in energy production from renewable sources. Yhis also implies a growth in power production from wind resources. This kind of technology compared to the other renewable sources, requires lower initial investments and exploit a source of energy usually available everywhere and particularly in the zone of the globe where the most industrialized countries are located. In the last decade of the 20th century many wind turbine models have been built and tested: vertical (VAWT) or horizontal (HAWT) axis, one two or three blades, up wind or down wind etc.. The horizontal axis, up-wind, three blades wind turbine is usually considered the most suitable and efficient configuration [26]. In a context of free energy market, wind energy is collecting in the last years more interest from researchers and companies all over the world. The interest in making this resource cheaper and more competitive, is leading the investigation about the optimization of wind turbine components and wind farms. A huge amount of studies have been performed about the field of wind energy, from structural and aerodynamic analysis to controls and scaled models passing through electrical components like the generator or the safety systems. A particularly important aspect is the optimization of blades since modern machines are characterized by increasing rotor dimensions that may lead to non-negligible dynamic and structural issues. In the current world of science and engineering, ample scopes are provided by computer technology that makes possible to run on a cheap, domestic pc, accurate multi-physics simulations to better understand and forecast possible design problems. However it is important to notice that usually, extremely accurate physic models are not suitable for optimization purposes, because they need quite long computational times:it is then important to reach the perfect trade-off between accuracy of the solution and computational costs. The design of wind turbine blades is indeed a complex, multi-disciplinary process that involves both aerodynamic and structural aspects and a holistic point of view is really important to understand how a variety of factors can affect the final result. A proper design method, the availability of proper instruments and the deep understanding of the phenomena behind the wind turbine behavior, will help this technology to become more and more competitive in the free energy market, decreasing the cost of energy produced.

1.2 Aim of the Work

A big effort has been spent in the last years both from academic and the industrial players to better understand how to properly manage the problem of wind turbine optimization. During the process, engineers select a set of important parameters to optimize and then either with an algorithm or manually, a solution that minimizes a merit function is investigated. Looking at the rotor, the process can be addressed by a pure aerodynamic, a pure structural or a coupled aero-structural optimization. It is important to highlight that the choice of the parameters considered considerably affects the final result and the value of the objective function. The idea behind this work, is to get a glimpse of a new, fully integrated, multidisciplinary, aero-structural optimization method, taking into account the accurate physical description of the aeroelastic and dynamic effects using high-fidelity models. Summarizing, it is possible to outline the principal goals of this thesis

1. Develop a code for aero-structural integrated design of wind turbine rotors, using an accurate physical description of the behavior of all the principal components of the wind turbine.

2. Investigate the most suitable algorithm to ensure an automatic optimization process minimizing the human intervention and limiting the computational times.

3. Perform the integrated aero-structural design optimization of a benchmark wind turbine aiming at reduction of the cost of energy.

The ideal procedure to solve this problem must have some important features to ensure reliability of results and possibility of future improvements:

- Convergence: the first and most important feature of the new code is to be able to solve the optimization problem. Of course this property does not depend only on the procedure chosen but also on the problem itself.

Any algorithm can be suitable or not depending on the properties of the particular problem, and the convergence may be guarantede only under a certain set of conditions.

- Robustness: the code must be able to bear unpredictable errors that can arise during the execution of the simulation without crashing suddenly. This also means that the method must be able to keep iterating even if a problem occurs during a simulation or a small change in the input files is accidentally made.
- Speed: since the simulation for loads, energy production, frequencies and the structural optimization procedure, are quite long to be run, an important feature of the algorithm is to be able to identify the optimal solution within a defined tolerance, in a few number of iterations.

Regarding the choice of the proper algorithm, among all the possible methods, the present research is restricted to three candidates:

- global search method
- a surrogate model or response surface method
- gradient based, SQP method

These families of algorithms are selected among a long list of potential candidates because they have good strength regarding convergence and they generally reach the minimum of the cost function in a limited number of iterations. Moreover, Matlab is chosen as computational environment since the existing codes to perform the aerodynamic and structural optimizations are available in on this platform. Other powerful methods like genetic algorithm or particles swarm have been discarded because they need a really high number of function evaluation. All these approaches have been tested and the result compared. It is important to notice that the choice of these three algorithms out of the number of possible methods, is not absolute and general but it is correct only for the present problem solved with the particular set of instruments used here. This obviously means that these three methods are not necessarily the only methods for all the wind turbines optimization problems, but they appear the most promising for the problems investigated in this specific work

1.3 State of Art

A good number of studies during last years has been carried out in the field of wind turbine optimization. Some of them are interested in optimizing only the aerodynamic properties of the rotor starting from a set of design requirements [13] or from a baseline blade using the beam momentum theory

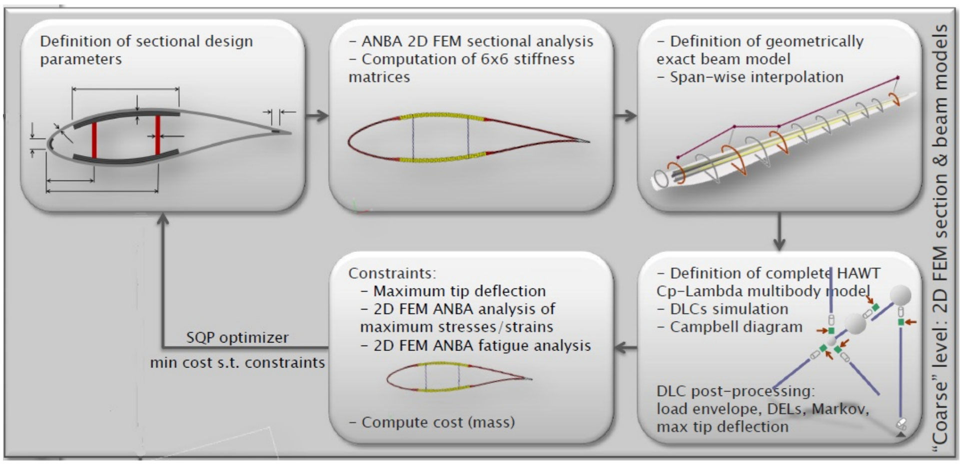


Figure 1.1: [10] optimization procedure.

(BEM) implemented in Excel worksheets [8]. A certain number of studies are focused on the pure structural analysis to select the best disposition of the composite material in the blade and the correct thicknesses of the structural elements using accurate FEM simulations [15] [16]. But the approach is collecting more interest in the last years refers to the problem of integrated aero-structural optimization. Jureczko et al. [24] optimize a blade with a multi-objective genetic algorithm calculating loads with the BEM theory and then simulating the behavior of the blade with the commercial finite element software ANSYS.

Vesel & McNamara [28] search for the minimum COE considering as variables the airfoil shapes, chord and twist distributions and the degree of bend-twist coupling as structural parameter. Airfoils are considered as optimization variable also in the software *Rotoropt* developed by L. Flusslang [4]: this tool perform a maximization of the AEP considering as variables airfoils, blade length, chord and twist distribution, tip speed, tower structure, tilt, cone, overhang and pre-bend. An estimation of the total rotor cost is carried out as well. A sequential aero-structural approach is followed by Fangfang S. et al. [20]: MATLAB's optimization function Fmincon is used to maximize the coefficient of performance calculated with the BEM theory, after that a FEM model of the blade is subjected to analysis to estimate response to loads applied and natural frequencies. Bottasso et al. [10] optimize the ratio between rotor mass (plus a constant contribution for the tower mass) and the AEP through a sequential aero-structural procedure using multi-body aeroelastic simulations for loads and annual energy production estimation (1.1). An integrated approach is followed by Ashuri T, et al., [5] to minimize the levelized cost of energy through a simultaneous optimization of rotor

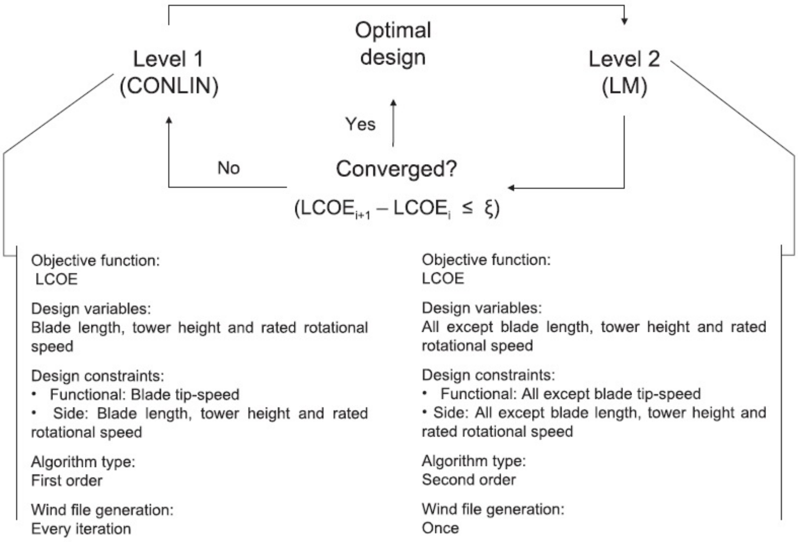


Figure 1.2: [5] optimization procedure.

and tower(1.2).

Wind turbine is optimized following a multi stage process: in the first stage, a gradient based method (CONLIN) is used to determine the principal characteristics of the machine like blade length, tower height and rated speed considering the constraint of maximum tip speed. In the second stage, Lagrange Multiplied method (LM) is enforced to find the distribution of chord, twist and structural thicknesses along the blade span. The COE is considered also by Xudong et al. [29] as a merit function but is calculated out of the ratio between rotor cost and annual energy production and the variables considered are chord and twist distribution.

1.4 Considerations

The wind turbine world is changing rapidly. The principal research direction seems to be the holistic design of wind machines. Since rotors of today are characterized by diameters up to 180 m, neglecting one of the aspects (aerodynamic or structural) may lead to heavily sub-optimal design in term of actual cost of energy. It seems that even a sequential aero-structural approach, if not properly bounded, can bring to solutions far from the optimum [14]. In general, the optimization process in many case is moving from component level to system level [5]. It remains actually quite common to consider as variables the twist and chord along the blade because their distribution mainly influence the energy production and the loads of the machine.

A number of recent studies examine also the influence of different airfoils selection in the optimization process: although it should be interesting to include in the present work even the airfoil analysis, this can be possible only by means of low fidelity models. Considering chord and twist distribution at several stages of the blade span already means generally to consider a quite high amount of variables at every iteration and depending on the loads calculation strategy, the computational times can rise in unreasonable way. That is why only a few researchers are focused on aeroelastic simulations of the whole turbine for optimization purpose and they prefer to use more simplified method like to calculate loads and energy production. This can be dangerous when large wind turbine are considered, because the influence of structural design on the dynamic behavior becomes not negligible [1] i.e. for these slender and flexible blades,the aeroelastic deformation is unavoidable, and this alters the turbine power performance. Indeed it seems difficult to properly consider constraints like maximum tip deflection without carefully simulating the elastic properties of the blade. This issue can be recovered by a following, more accurate FEM analysis that nevertheless can not be actively used in the optimization process. Because of these considerations, in this work high fidelity models are used and a trade-off needs to be found between model accuracy and computational times. For example computational fluid dynamic (CFD) wind flow simulation are not implemented because computational times for a coupled CFD-Multi-Body approach would not be suitable for this purpose [17].

Chapter 2

Description of the problem

2.1 Nature of The Problem

The optimization of wind turbine is a complex, deeply multidisciplinary, constrained problem. A high number of degrees of freedom and constraints must be taken into account,with regard to satisfy all the constraints in order to provide a feasible solution. It is possible to formulate the problem of finding the minimum cost of energy as follows:

$$\begin{gathered} find : \min(COE) \qquad (2.1) \\ COE = f(BladeMass, AEP) \\ BladeMass = f(Chord, Twist, Thicknesses) \\ AEP = f(Chord, Twist) \end{gathered}$$

s.t.

$$\begin{gathered} x = Chord, Twist, Thicknesses \\ UB_{\mathrm{i}} < x_{\mathrm{i}} < LB_{\mathrm{i}} \\ g_{\mathrm{i}}(x) \leq 0 \\ h_{\mathrm{i}}(x) = 0 \end{gathered}$$

where COE is the Cost Of Energy calculated as a function of the blade mass and the Annual Energy Production (AEP). A set of upper and lower bounds for each design variable besides a number of inequality and equality constraints is considered in the problem to limit the searching region. Thus the variables of this problem are the span-wise chord, the twist distribution and the thicknesses of all internal structural elements: it is important to notice that by changing the chord or the twist distribution, the aerodynamic loads on the blade will change and so the thicknesses of the structural elements shall adapt to satisfy the constrains. It is not possible to imagine any kind of superimposition principle for this problem because many of its aspects are highly non-linear and deeply coupled: because of the nature of the problem, among all the possible structural configurations, every time exists one that has the minimum mass and for this arrangement, at least one constrain lies on the boundary. From this consideration arises the difference between aerodynamic and structural efficiency: aerodynamic efficiency refers to a blade that due to its chord-twist configuration, is able to extract the maximum amount of energy in one year. Thus solves the problem:

$$\max AEP = f(chord, twist) \tag{2.2}$$

This is a well posed problem because for each set of chord and twist distribution, a unique value of the AEP exists and depends with continuity on the data [18].

Structural efficiency is referred to the best arrangement of blade's structural elements for a given set of loads. Blade twist and chord are not changed in this kind of analysis but among all the possible configuration of the interal structure, the one with the minimum total mass is investigate. A structural efficient blade is substantially a blade to wich are applied the same loads but has a lower total weight. The problem solved in this case is

$$\min Mass = f(thicknesses) \tag{2.3}$$

Structural efficiency is referred to the problem of finding for a certain set of loads experienced by the blade, the minimum total weight without compromising the structural integrity. In this modelization of the problem the mass is function only of the thicknesses of the internal structure. Mass is a monotone function in each variable. It starts from zero and can only grow applying a perturbation to any degree of freedom. Obviously, not all the solutions are actually feasible and then the minimum of the 2.3 is the first solution that satisfies the constraints. This is a well posed problem too [18] and hence is possible for a given blade shape to determine the minimum mass arrangement by changing thicknesses of internal structural elements with a constrained optimization approach. Solving equations 2.2 and 2.3 subsequently the user is sure to find the most structural efficient configuration for the aerodynamic most efficient blade but there is no guarantee about the final cost of energy. In fact, optimizing a merit value like the ratio of blade mass and AEP or the COE, using a sequential approach mathematically means to solve the problem

$$\min COE = f(Aero, Thicknesses) \tag{2.4}$$

where *Areo* is a vector containing all the aerodynamic parameters and *Thicknesses* is a vector of structural degrees of freedom. Equation 2.4 needs the hypothesis that *Areo* and *Thicknesses* are independent, but actually *Thicknesses* is mainly driven by the loads experienced by the machine and the loads mainly depends on the *Aero* parameters, so that

$$\frac{\partial Thicknesses}{\partial Aero} \neq 0 \tag{2.5}$$

i.e. the real problem to be solved is not 2.4 but rather

$$\min COE = f(Aero, Thicknesses(Aero)) \tag{2.6}$$

This can be done iterating the sequential approach till convergence or trying to use appropriate aerodynamic constraints in order to take into account the problem stated in 2.5. From a mathematical point of view, there are no clues to say *a-priori* that the problem is well-posed like 2.2 and 2.3 and namely one aim of this thesis is to answer to this question, and then choose the most appropriate optimization algorithm.

2.2 Tools used

2.2.1 Cp-Lambda

The software Cp-Lambda (Code for Performance, Loads and Aeroelasticity by Multi-Body Dynamic Analysis) [7] [9] is a multi-body, full finite-element software, able to handle every kind of wind turbine without modal reduction on the components of the structure. Turbine blades, towers, drive trains are modelled into beam elements and it is also possible to add point masses or complex joint models accounting for power losses, backlashes, concentrated springs dampers and so on. It is even possible to model an arbitrary wind turbine configuration using the elements (beams, joints, mechanical actuator etc.) provided by a library. Blades are modeled into beams from a reference line, and this allows to model complex shapes like pre-bended blades. Aerodynamic properties of blades, nacelle and tower, are considered using the classical 2D lifting line theory. The characteristics in terms of lift, drag and momentum coefficient of the different airfoils used along the span are given in tables for various angles of attack and Reynolds numbers. The formulation of the problem brings to a a set of non-linear, partial differential algebraic equation solved with an implicit integration procedure [12],[11]. The software is able to perform all the Dynamic Load Case (DLCs) simulations according to the international standards [2]. To generate the turbulent wind field for the dynamic simulations, the software TurbSim [23] is used. Cp-Lambda is even able to compute the cp vs. tip speed ratio curves and the Campbell diagram for the rotor. Static simulations can be run as well to estimate the Annual Energy Production (AEP) of the machine.

2.2.2 ANBA

ANBA (ANisotropic Beam Analysis) [21] is a software initially developed for rotorcraft blades analysis that is able to calculate stiffnesses and stresses in a

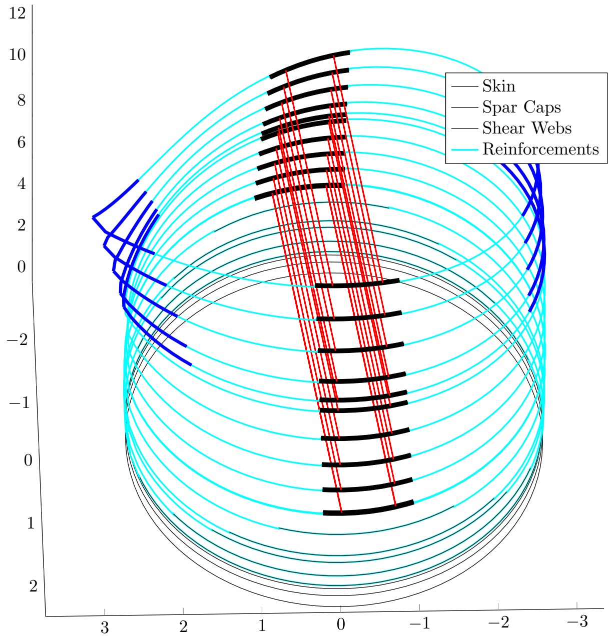


Figure 2.1: Internal structure scheme of the baseline blade

beam cross-section made of anisotropic or non homogeneous material. ANBA uses a bi-dimensional , finite element discretization and can easily handle arbitrary tailored blade sections. Moreover, the so calculated stiffness matrix is fully populated and allows to take into account all the potential couplings between flap, torsion, lag etc. The blade considered in the structural analysis has an internal structure built with two straight webs and the relative suction and pressure spar caps. Webs are placed approximately at the maximum thickness of the airfoil in order to obtain maximum flapwise bending stiffness. A third shear web is present close to the trailing edge but starting from $r = 21,8$ m. Reinforcement for the leading and trailing edge of the blade are present too. The skin is modelled as a set of plies. A scheme of the internal structure is shown in 2.1 for complete description refer to [6].

2.2.3 Cp-Max

The previous code used by the Lehrstuhl für Windenergie for the aerostructural optimization was *Cp-Max*, a set of MATLAB's scripts initially developed by Politecnico di Milano's POLI-Wind [10]. This code considers an integrated aero-structural optimization through an independent approach. Namely can be used as pure aerodynamic or pure structural optimizer. First of all the aerodynamic properties of the blade (chord and twist) are optimized, and then the aerodynamic optimal blade is subject to a structural process where the the elements are modified in order to reach the minimum weight with respect to the constrains. Concerning the aerodynamic process a number of blade station along the blade where is possible to variate chord and twist are considered. The variables considered are managed by a gradient-based method that, by varying them, seeks the maximum AEP of the machine through a number of static simulation with Cp-Lambda. Subsequently, for the aerodynamic optimized blade, Dynamic Load Cases according to the IEC regulation are computed and then extreme loads, fatigue loads and maximal strains in the elements are extracted. At this point all the loads are frozen i.e. constant loads value are kept, and a minimization of the blade mass considering a wide range of non linear constrains, upper and lower boundaries is performed. At the end of this operation, to recover possible aeroelastic effects due to variation of cross-sectional properties of the blade (thickness and stiffness), all the DLC are performed again and the blade is once more optimized with respect to the varied loads: the loop keep going till the change in the blade mass from an iteration to the next is lower than a certain tolerance user-defined. Load cases are calculated also with static wind only for the first iteration in order to provide to the following dynamic turbulent simulation, an already deformed simulation and avoid huge oscillation and big deflections in the first seconds of the simulation. At the end of the aero-structural double-loop, the Cost of Energy is calculated. It is not the leading parameter of the optimization process but only a data useful for the final comparison between different blades.

2.3 Variables

The variables considered are divided into aerodynamic and structural, the first ones are considered in an external optimization loop, the second ones in a structural sub-loop. Aerodynamic variables are:

- Length of chord in 4 stations along the span
- Value of twist in 4 stations along the span

Structural variable are:

- thicknesses of 72 structural elements

The decision to use only four chord variables comes from the necessity to compress computational time: too many variables may bring to an unreasonable computational effort or even to a non-well-posed problem. Other important aerodynamic parameters like the airfoils used or their location along the blade are not optimized in this work because the computational effort would be inappropriate. The usage of so low number of design parameters leads to the necessity to interpolate the chord values along the span, using splines to preserve a generally smooth shape for the blade. Out of

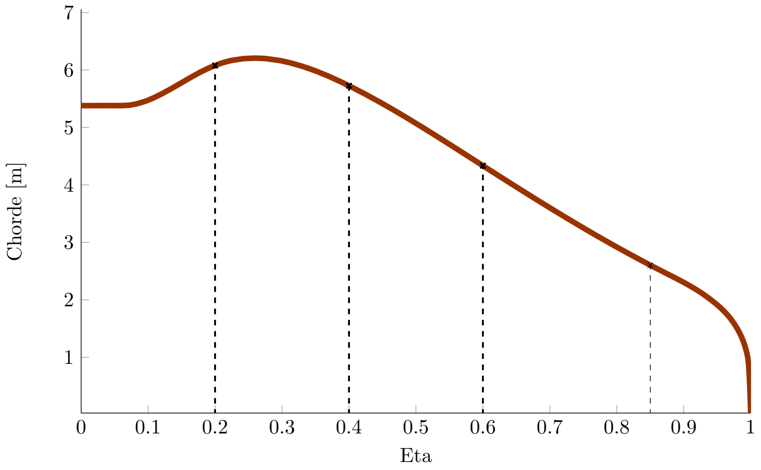
Station	spanwise adimensional position
1	0.2
2	0.4
3	0.6
4	0.85

Table 2.1: Position of the optimization station along the span

four values of chord and twist, a process of parametrization is carried out. Concerning structural variables, 14 station span wise are considered. No structural variables are taken into account for blade root but values are kept constant. For the other stations span-wise

- Skin
- Spar cap
- Trailing Edge reinforcement
- Leading Edge reinforcement
- Root reinforcement

are considered.



All the thicknesses along the span wise in the positions not coincident with considered stations, are obtained by interpolation: the user can chose to linearly interpolate the values between two consecutive stations, or keep it constant. In order to reduce the number of structural design variables, the two webs are considered to have the same thickness and only one of them is optimized. It is even possible deciding if the webs will be twisted (locally orthogonal to the chord line) or straight (orthogonal to the line of maximum chord). A number of other parameters are necessary for the analysis but they are not considered as degree of freedom during this work. The parameters needed as input that can not be optimized are for example the P_{rated} of the machine, the $v_{\text{cut-in}}$ and the $v_{\text{cut-out}}$ the rotor radius R the height of the hub h_{hub}. The analysis of cone and pre-bend infuence on the cost of energy and on the machine behavior could in the future be included in the present work.

2.4 Constraints

The set of constrains considered for the problem is necessary to be sure the solution founded will lie in a feasible region of the solution space. The applied constrains to be satisfied are:

Maximum stress the stress σ_{max} in every section for every element must not exceed a certain value corrected by an appropriate safety factor.

$$\sigma_{\text{max}} \leq \sigma_{\text{adm}} \tag{2.7}$$

Maximum strain the strainε_{max} in all structural element has to stay under a certain threshold.

$$\varepsilon_{\text{max}} \leq \varepsilon_{\text{adm}} \tag{2.8}$$

Maximum Tip Deflection the maximum tip deflection for the worst case in the worst condition must be taken into account to avoid that the blade hits the tower. In the time history of all the DLCs is searched for the worst deflective condition, among them the maximum tip deflection is founded.

$$\delta_{\text{tip max}} \leq \max \delta_{\text{tip}} \tag{2.9}$$

Fatigue fatigue is considered performing a rain-flow analysis to evaluate the damage due to cyclical loads. The counting of the rainflow analysis requires a number of dynamic turbulent simulation for different wind speeds from the cut in to the cut out. The analysis sorts the peaks of tensile and compressive stresses in the time history associating for to the maximum tensile the maximum compressive stress and hence creating the maximum range cycle. The other cycles are created associating every tensile stress with the following compressive one. Sorting in this way by amplitude, different bocks of stresses are considered and for each block amplitude the limit number of cycles admitted is calculated. Thus the Miner law is applied adding up the effect of every block on the total damage index.

$$d_(\sigma) = \sum F_v k \frac{n(\sigma_m, \sigma_a, v_k)}{N(\sigma_m, \sigma_a, v_k)} \tag{2.10}$$

where d is the damage index $F_v k$ is a coefficient used to take into account how much time the machine will operate in his life at v_k speed, n is the number of cycles at average stress σ_m ,amplitude σ and N is the number of cycles to failure corrected by an appropriate safety factor. Acording to the Miner law [25], for $d = 1$ statistically failure due to fatigue shall occur.

Natural frequencies first natural flapwise frequency $\omega_{1\ \mathrm{flap}}$ must be higher than the 3-per-revolution frequency $\omega_{3\mathrm{p}}$ at the rated rotor speed Ω_{rated} to avoid resonance phenomena. Moreover a constraint on the ratio between first flapping and first edgefrequency is considered. Special simulations for this purpose are performed.

$$\omega_{1\ \mathrm{flap}} \geq s_1 \omega_{3\mathrm{p}}(\Omega_{\mathrm{rated}}) \tag{2.11}$$

$$\frac{\omega_{1\ \mathrm{edge}}}{\omega_{1\ \mathrm{flap}}} \leq s_2 \tag{2.12}$$

Maximum tip speed because of noise issues, the turbine is imposed to work at a maximum rotational speed of the tip lower than a certain value.

$$\max v_{\mathrm{tip}} \leq s_2 v_{\mathrm{tip\ a}} \tag{2.13}$$

In order to calculate stresses, strain and fatigue loads, a set of dynamic load cases according to the IEC standard [2], are performed. This calculation is carried out with Cp-Lambda, and the usage of this tool enables to chase the intimate physic of the problem but the price to be paid is an elevate computational time. For this turbine the most critical constraints seem to be the maximum tip deflection, the placing of the first flapping frequency, and fatigue on some part of the skin. The maximum tip deflection constraint forces the designer to build an adequately stiff structure, so that the oscillations remain in a certain narrow range: increasing the stiffness, it helps even to raise the eigenfrequencies, that on the other hand, tends to decrease if the blade mass grows. For so long blades, built with big skin panels, buckling could be a serious structural issue: although a buckling constraint is not here directly considered, but is taken into account a set of empirical correlations for estimation of non structural masses used to keep under control this phenomenon. It is important to notice that all the constraints here considered are applied to the structural optimization process and not to the aerodynamic one. It is indeed possible to include additional aerodynamic constraints like maximum chord or maximum and minimum solidity $\sigma = \frac{BladeArea}{SweptArea}$ but in this work these considerations have been neglected not to loose generality.

2.5 Objective

The objective function here considered is the Cost of Energy$[\frac{\$}{kWh}]$ because is the best parameter to compare wind energy with the other technologies and evaluate its competitiveness in the market. The model used for COE has been developed by the American National Renewable Energy Laboratory (NREL) [3] with some modification to ensure more reliable results for this thesis:

$$COE = \frac{FCR * ICC}{AEP} + AOE \tag{2.14}$$

where

- COE: Cost Of Energy $[\frac{\$}{kWh}]$
- FCR: Fixed Charged Rate
- ICC: Initial Capital Cost [\$]
- AEP: Annual Energy Production $[kWh]$
- AOE: Annual Operating Expanses [\$]

Fixed Charges Rate can include insurance, salaries, utilities, vehicle payments, loan payments and mortgage payments. These charges allow to create more predictable budgets and estimate cash flows more accurately. The Initial Capital Cost is an estimation for all the machine's components cost of material and manufacturing from the tower to the rotor and blades plus a contribution from the Balance of Stations. The Annual Operating Expanses is considering the annual costs of O&M and the lease cost of the bottom. This cost model was developed as a scaling model based on a restricted amount of important parameters such as the rated power of the machine, his radius or the mass of the blades. Choosing the correct objective function in wind turbine design is an open task but it has been shown that this cost of energy model is still the most reliable tool at disposal. In effect considering directly the ratio between blade mass and AEP can be interesting but it has been shown that may overemphasize the role of the blades in the total cost of the system [27]. On the other hand, consider only the blade mass as an input for the cost model, could lead to an over estimation of the AEP contribution. In any case the NREL cost model is the best one available now for design purposes.

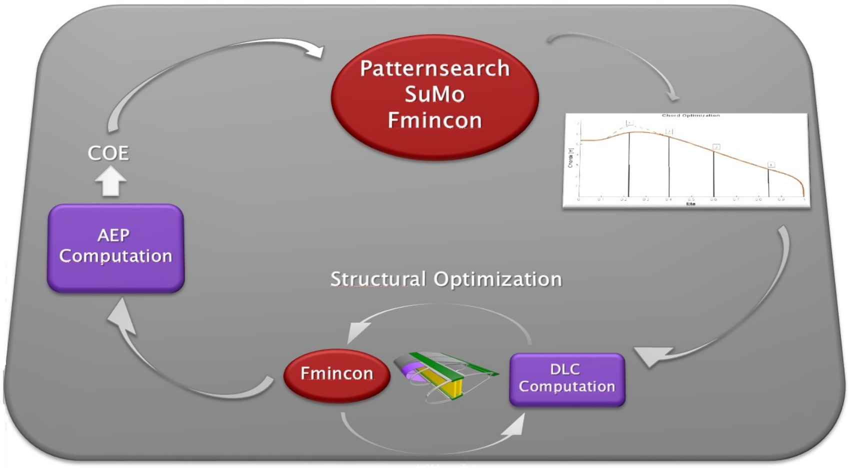


Figure 2.2: Working scheme of the code

2.6 Scheme of The Code

2.6.1 New code

From this last consideration arises the idea of the new project. The cost of energy can be a more effectively objective than the AEP and blade mass because is the parameter that synthesize the best the competitiveness of a wind turbine in the energy market.

To make this step the scheme of the previous code is completely reviewed:

- Sequential approach must be abandoned to shift to a fully integrated aero-structural implementation. This means that the process will not pass through an independent aerodynamic and structural optimization anymore.
- AEP and blade mass must lie at the same level in the optimization, that means that the best blade can be the one that does not have the best AEP.
- Choose an appropriate research algorithm to perform an external aerodynamic loop.

Surely a significant part of the previous work can be re-used in this thesis, for instance the structural optimizer or the AEP computation routine. Now the main changes to the code are outlined.

2.6.2 Twist Optimizator

Instead of considering twist and chord distribution at the same level, the twist is subject to a sub-optimization process to save computational times. The twist sub-optimization is operated searching, for every blade configuration, the twist distribution which leads to the highest coefficient of performance maintaining the pitch value and the tip speed ratio $TSR = \frac{\Omega R}{v_\mathrm{w}}$ fixed.

$$\max Cp(TSR, \beta) \tag{2.15}$$

where Ω is the rotor speed, R is the rotor radius, v_w is the far field wind speed and β is the pitch angle. This method has been selected because, using multi-body AEP simulations, is more accurate if compared with a beam element theory optimization and the computational expanses are admissible. It is important to notice that even if the twist only slightly changes during the optimization, this operation is made before the load cases computation in order to correctly evaluate the change in loads due to a change in the twist distribution. Acting like this aeroelastic effect are not taken into account in the twist procedure.

2.6.3 Non structural masses update

During the structural cycle, it is necessary to take into account in an approximate way a really important constraint for large blades: buckling. Buckling is a phenomenon of instability under compressive stresses that depends on the stiffness and inertia but also on the panel area. The biggest is the area and the lower is the inertia and the stiffness of the panel, the more probable is buckling. This event is barely predictable but with accurate FEM models that for sake of simplicity and computational times, can not be here implemented. Not to guiltily forget this part of the problem, an empirical relation between stresses in panels and the amount of non structural masses (filler) is considered. Essentially a rough design of filler to avoid buckling is implemented and every structural iteration updated. It is really important consider this parameter because for blades with high maximal chord value, the buckling problem becomes critical and providing no instrument to the code to model these phenomena could lead to a wrong computation of blade mass and so COE. Was then decided to consider an even approximate model able to scaling the non structural masses for growing blades.

2.6.4 Loads calculation

The calculation of loads acting on the blade is adopted from the previous code but changing the number of the DLC taken into account, limiting it to the most critical ones for the machine considered. In total are calculated 16 DLC necessary for fatigue analysis and extreme loads evaluation:

- 12 DLC 1.1 for normal turbulent wind with speed from 4 to 25 meters per second, used to compute fatigue analysis
- 4 DLC 1.3 for extreme turbulent wind conditions, where the worst tip deflection are shown.

The choice of considering these 16 load cases arises from preliminary considerations made using *Cp-Max*: indeed the most active constraints are every time 2.9 for the maximum tip deflection in DLC 1.3 (extreme wind turbulence) and 2.10 for fatigue analysis. It is important to notice that several optimization softwares for wind turbine blades, evaluate loads applying the beam element momentum theory or at most computing a set of static analysis using multi-body FEM simulator. If this approach was followed even in this thesis, most limiting loads would not be revealed. In fact for this machine most driving loads are observed for the extreme turbulent wind condition and for fatigue over certain part of the skin. Then the design of the very same blade considering different approaches can lead to significantly different conclusions.

2.6.5 Structural Optimization

As above briefly described, the structural optimization process is carried out using as objective the total blade mass, and as variables the thicknesses of several structural elements in 14 stations along the span. Optimization process is executed even here by gradient-based method able to minimize a function managing in the same time a big amount of constraints and checking their development. The method consists of considering each time, for every blade station, for every element in the the section, the influence that the variation has on each of these elements on total blade mass and non linear constraints. To do so an ANBA analysis in executed. After the evaluation of variations of thickness in every element, the algorithm calculates a gradient using central differences and seeking a new set of thicknesses that will be subject again to the process and so on till the tolerance on the weight change is not reached. The evaluation of a so high number of variables can lead to speed problem: even though every single perturbation evaluation is quite fast, the high number necessary widen requested times for computation. Moreover, to use ANBA it is necessary a *virtual machine* external to MATLAB and this makes quite hard to parallelize on more processor the function evaluation.

2.6.6 AEP Calculation

AEP calculation is recovered from the previous code [10] with some change to solve convergence problems that may occur in Cp-lambda simulation. To calculate AEP, a set of static simulation are run at different wind speeds from the cut in to the cut out velocity and the corresponding power coefficient

	Calculated $[kg]$	Estimated $[kg]$
INWIND 10MW Baseline	41467	64227
INWIND 10MW Advanced	41138	39189

Table 2.2: Comparison between masses calculated by the code before and after the structural optimization and masses extimated by the NREL scaling model in baseline and advanced configuration

is extracted. Then all the wind speeds considered are weighted for the probability of occurrence using a Weibull distribution [26] 2.16.

$$f(v) = k \left(\frac{v^{(k-1)}}{c^k} \right) exp^- \left(\frac{v}{c} \right)^k \tag{2.16}$$

where v is wind speed, k is a shape factor of the distribution and c is the average value of wind speeds. AEP is calculated on a mesh of TSR and β initially coarse and eventually refined near maximum AEP point. This choice does not bring to unbearably greater computational expanses because are performed only static simulations that are quite fast.

2.6.7 Cost of energy calculation

Even the cost of energy model comes from the previous code and needs as input only the blade mass and the AEP as variables. Other important parameters are taken into account but can not be varied in the optimization process like the material used for the blade, the hub height or the rated power. Particularly in this thesis the blade mass is calculated out of the structural model instead of the scaling law present in the NREL work [3] to ensure a better reliability in data and because the scaling model can easily fail in estimating the weight for such long blades as shown in 2.2. In this table are compared the masses calculated by the the code in baseline and optimized configuration and the relative extimation from the NREL scaling model.

The calculation of the COE is really fast because does not need any simulation to be run. After the evaluation of this quantity, an external file containing informations about AEP, blade mass, COE and computational time elapsed is written. This file is really important because allows to use the recovery mode if the simulation is stopped. Basically the data saved in this file are read by the code in recovery mode without running the simulations. Obviously the code is supposed to have an identical behavior for two subsequent runs, otherwise the values recovered would not be associated to the right blade shape, leading to wrong final deigns.

Chapter 3

Algorithms review

In this chapter the three different families of algorithms are considered, evaluating for each one merits and deficiencies in order to give the reader an overview of problems encountered and relative solutions adopted.

3.1 Patternsearch

The global search algorithm used in the present work is the MATLAB function Patternsearch that has in the robustness his strongest point: indeed no particular assumption has to be made to ensure a proper behavior of the method.

3.1.1 Operation Mode

The functioning and iterative evolution of the algorithm are quite easy to understand. It is here briefly described the work flow

1. Starting point: a starting point is selected by the user. In this case the start is set at the baseline blade configuration
2. Perturbation of variables: in order to choose in which direction to proceed, Patternsearch does a *Poll* i.e. variates the degrees of freedom with an intensity chosen by the user and subject to a structural optimization each blade so obtained. Different logics called *Poll strategies* can be adopted for perturbing variables:
    - GPS (General Pattern Search) Positive Basis 2N: variables are one by one increased and then decreased
    - GPS Positive Basis Np1: variables are augmented independently and then decreased all together at once
    - MADS (Mesh Adaptive Direct Search) Positive Basis 2N: like GPS Positive Basis 2N but the order of perturbation is random

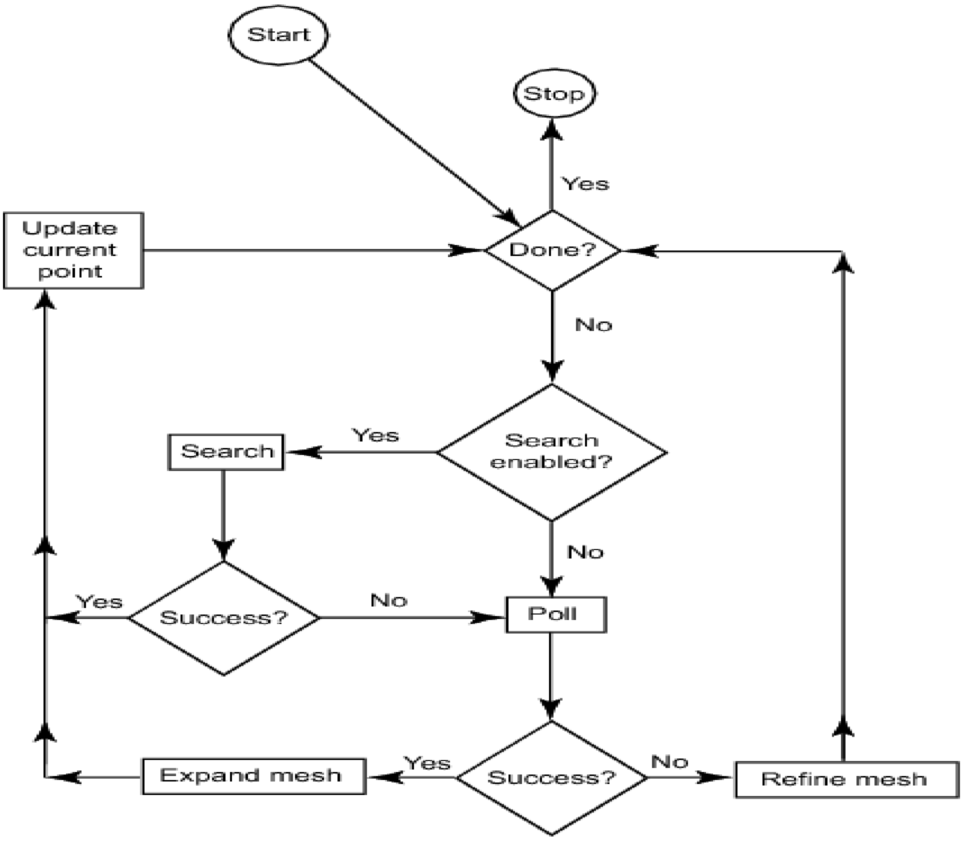


Figure 3.1: Patternsearch workflow

- MADS Positive Basis Np1: like GPS Positive Basis Np1 but the order of perturbation is random

In total every *Poll* consists in a different number of blades evaluation: the '2N' strategies make a maximum of $2(NumberofVariables)$ different guesses, the 'Np1' strategies instead, $(NumberofVariables) + 1$

3. Evaluation of obtained results: if one of new blades after the structural optimization, obtain a lower value of COE respect to the baseline, this becomes the new starting point for the next *Poll* and the range of variables perturbation (mesh) is augmented. If no candidate obtains a lower COE compared to baseline, the mesh is decreased and a new *Poll* is made. Furthermore it has been decided to consider a *Poll* successful and then to restart with a coarser mesh , as soon as the algorithm was able to find a single blade with a better COE, without waiting for the *Poll* to be completed. This decision was kept to save computational time and because does not bound the final results.

3.1.2 Pros

One of the most positive aspects of this algorithm, is its insensitivity to non-regular response surfaces, both non-continuous or not differentiable. By nature, Direct-Search algorithms work sampling the solution space and generally they have no memory of the previous iterations. In a problem like the one handled here, there is no *a-priori* assurance about the continuity of the solution: surely discontinuity shall rise at the boundaries, dividing feasible and unfeasible solutions. A not correct placing of boundaries could take to a misleading usage of optimization methods that need to calculate gradients and Hessians, because numerical results could not have any physical meaning. Patternsearch instead is able to manage within the code, errors and discontinuity in the objective function indeed because it has memory of only one result at time: in case the blade is subject to the analysis, for its particular shape, pose issues in structural calculation, a command in the code would tell Patternsearch to discard that blade considering for it a high value of COE. This is done without compromising next results. The exactly same situation would happen if the code has some internal trouble e.g. a non-convergence in the DLC or AEP calculation: if the problem is revealed, an high value of COE is set and that only means to Patternsearch that that blade will be forgotten in the next iteration. Neglecting a result if the DLC or AEP computation did not reach convergence has the disadvantage that it is impossible to know whether the blade so discarded, has in the reality an high value of objective function or not. Fortunately the blade can be simulated again after some tuning of the parameters or the simulation that did not converge can be run manually and the result recovered. Namely, since the structural optimization process is really long (up to 8 hours of computation) it becomes necessary to have a system in the code able to recover data in case the session shut down in a unwanted manner. To do so it is necessary to have a deterministic behavior of the method used: because of this the MADS *Poll* method has to be discarded because evolve in a random way. Choosing a General Pattern Search (GPS) method is instead easy to run subsequent simulations every time in the same order, and this define a further benefit of this algorithm

3.1.3 Cons & Problems

The research strategy adopted by Pattersearch is quite simple and this is even his weakest point. In fact for it is possible to move just in one direction every iteration i.e. only one chord variable changes every *Poll* and this makes the method barely suitable to manage an high number of independent variables.It is kind of a blind method because every time it considers only a point with a lower COE within the mesh and forgets all the previous results. These characteristics could let think that convergence will be quite slow and in fact it is, if a uncorrected value of initial mesh is considered. Moreover for every *Poll* considered positive (i.e. that bring to a solution with a lower value of COE), the mesh is augmented and if several *Poll* at the beginning are positive, in a few iteration it is possible to get to a situation in which the solution investigated is quite far away from the initial guess, in a zone of the space with big chord variations. This could be not desirable if was necessary at that point, only refine the solution using a fine mesh, because it would be necessary to wait several iteration before the mesh to become small enough.

3.2 SUMO

In this section is explained the usage of a SUrrogate MOdeling method (SUMO) as a tool for the optimization. This family of algorithms is usually suitable to solve difficult and computationally expensive engineering design problems. SUMO is a freeware MATLAB toolbox developed by the University of Gent [22] and it is automatically able to build accurate surrogate models (also known as metamodels or response surface models) of a given data source within the accuracy and time constraints set by the user. The models created can be even displayed on the computer monitor.

3.2.1 Operation Mode

The name Surrogate Modeling is referred to a number of optimization techniques that seek the function global minimum building an analytical model (a.k.a. surrogate) by sampling the solution space.The mathematical model behind this algorithm can be quite complex and essentially works interpolating the sampled points with a Kriging model, is a linear combination of basis functions (depending on the geometric distance between sampled points) and polynomial terms [19].

$$\hat{y}(x^*) = a + \sum_{i=1}^{n} b_\mathrm{i}\varphi(x^* - x_\mathrm{i}) \tag{3.1}$$

where

- $\hat{y}(x^*)$ is the predicted value of the function at point x^*.
- a is the polynomial term
- $\sum_{i=1}^{n} b_\mathrm{i}\varphi(x^* - x_\mathrm{i})$ is the sum of basis or error function depending on the distance between sampled point and current point.

This approach is quite different from a minimum squares or a spline interpolation. The biggest difference with a minimum square interpolation is that the solution surface created by a kriging model actually passes through the sample points and builds around them a meta-model. On the other hand the main distinction with spline interpolation is that the response surfaces are not forced to follow a polynomial shape (quadratic, cubic or so on). Indeed the surface created by the kriging interpolation uses as focal points the samples and around them builds an estimation of the real surface making no assumption on his nature. The only aspect considered in the interpolation process is the distance between a point and the samples calculated around it. It is easy to notice that the function should have for each different set of variables one and only one value of the objective, otherwise would be impossible to fit the surface for all the points. Actually, SUMO is able to bear different outputs for the same input, but the surface built in this way becomes easily irregular and barely reliable. In other worlds this method is not suitable for procedure that may observe statistical oscillations. In general the more are sampled the points, the more is reliable the solution : the logic behind the sampling is driven by two conflicting needs, the first one is to search the minimum in the place where the analytical solution is predicting it, the second is to widely explore the solution space. The algorithm then alternate a research of the expected global minimum with a random survey of the surrounding space. This last aspect gives the method the mathematical property to be ‘dense’ and guarantees the possibility to find the global optimum. This characteristic is usually highly desirable for an optimization method, but brings also to high computational expanses and a certain lack of deterministic behavior. The SUrrogate MOdeling toolbox works not exactly in an iterative way: at the beginning, a certain minimum number of samples are required from the simulator and then the first model is created. Afterwards new sample are selected near the predicted minima or in region where the calculated error function is seen to be high or in a random position.

SUMO toolbox needs as a input a MATLAB script used as sampler evaluator, the sampling algorithm chosen and the measure that is a metric to evaluate how good a model is. As matlab script was the code developed before, with some important modification to permit the software to accept as input a given chord distribution instead of using the patternsearch algorithm. As sampler evaluator was selected the extremaLOLA-Voronoi technique that performs an optimal trade off between searching properties (LOLA-Voronoi) and finding the poles of the function. A method that only search the minimum value of the function without exploration properties, could be easily trapped in local minima. The Voronoi method uses an approximation of the Voronoi tessellation of the design space, the LOLA method searches in highly non linear space regions. To evaluate the model created at every sampling iteration, a cross-validation method is used: it performs an n-fold

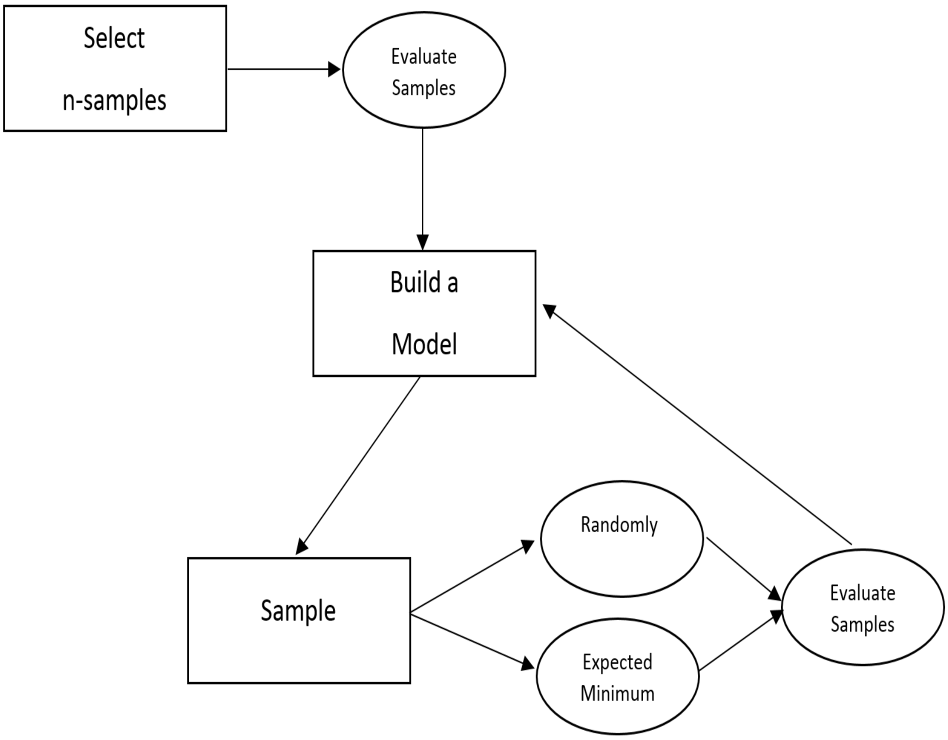


Figure 3.2: SuMo Toolbox workflow

cross validation on the model to create an efficient estimation of the accuracy. This is the only possible choice if it is impossible to make assumption on the response surface properties: actually to evaluate the current model it is only possible to compare it with the previous ones and estimate the change with the mean square error.

3.2.2 Pros

The surrogate Modeling is a very powerful instrument if corrected applied to engineering problems. The big advantage is the ability to describe a wide response space asking in input only a few samples. In this way it is possible even to plot a slice of the response surface and see it directly: this is a useful tool to understand quickly if the model built is consistent or not and to better understand which are the parameters that better describe the problem. It is moreover assured the convergence to the global minimum after a sufficient amount of sampling iterations because the method is mathematically ‘dense’.

3.2.3 Cons & Problems

This method as said before enables to make precise evaluations of the response surface but only under certain assumptions. First of all even if it is possible to create a model in a pretty low number of iterations, there is no assurance about its reliability and about the accuracy in the predicted values. A trial model was created initially not considering fatigue analysis to speed up the computation. In this case 36 iterations are needed to build the first model in 4 dimensions, but once created, it has a really coarse mesh and the so obtained surface is not representing at all the real one. Moreover the adding of new samples, brings to increases in the measure i.e. in a not reliable model

Furthermore the LOLA-Voronoi search method adopted in the toolbox uses a strategy to select the samples to be submitted to the simulator, that depends on the state of variables (SoV) in the computer’s RAM. Since the SoV is continuously changing, it becomes impossible to simulate two identical run and then there is no way to recover previous results. Moreover, required computational times have seemed to be improper to the analysis in exam because first real results (i.e. considering the full fatigue analysis) could have come only more then a month after the start of the computation. To avoid to discard prematurely this algorithm it has been decided to set aside the pure optimizer and to proceed to a manual construction of the response surface for the first two chord stages. The two stages have been selected because they proved to be the most sensible to COE variations according to the preliminary Patternsearch’s tests and they offer a good point of view over a set of different blade solidities.

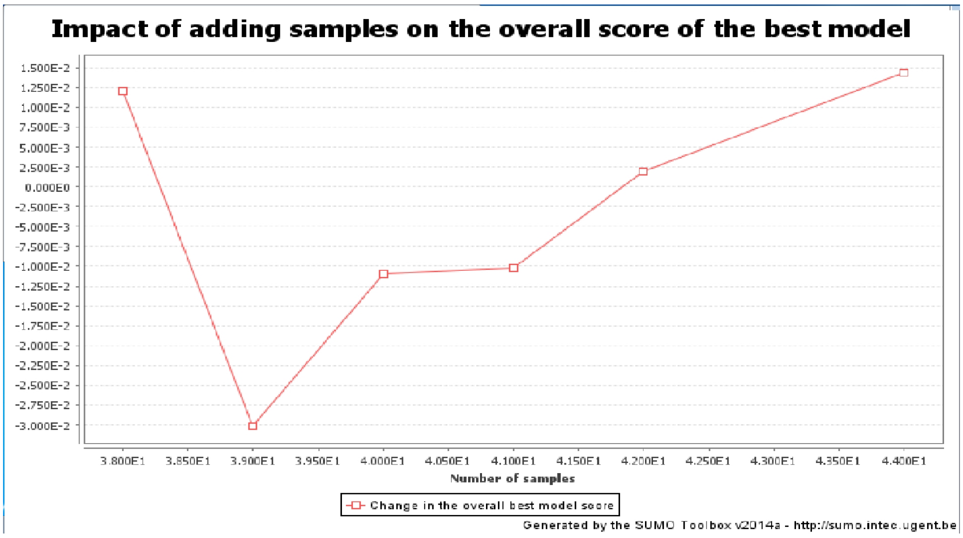


Figure 3.3: Value of the measure at each model creation

3.3 Fmincon

Last implementation considered is a gradient-based method withdifferent starting points (Multi-Start). The method chosen is MATLAB function Fmincon.

3.3.1 Operation Mode

A gradient based method is founded on the evaluation of the finite differences of the function. Starting from a base configuration COE is calculated, thus in turn, a small increment is given to each variable and the COE of the new blades is calculated. Then FminCon calculates the approximation of the partial derivative (finite differences) in all the directions following the formula:

$$u_{\mathrm{i}}'(x) = \frac{f_{\mathrm{i}}(x+h) - f_{\mathrm{i}}(x)}{h} \tag{3.2}$$

Where u' is the approximated first derivative of function f in the direction i. At this point an approximation of the gradient based on the finite differences can be evaluated

$$\bigtriangledown \bar{f} = \left(\frac{\partial f}{\partial x_1} + \frac{\partial f}{\partial x_2} + \dots \right) \approx \left(u_1' + u_2' + \dots \right) \tag{3.3}$$

The Hessian (second derivative of the function) is evaluated with a quasi-Newton method. The gradient indicates the direction of maximum grown of a function, thus changing the variable in the opposite way, it is possible to follow the path of maximum decrease in COE. In the gradient evaluation, forward differences have been used. In general this is not the most precise way

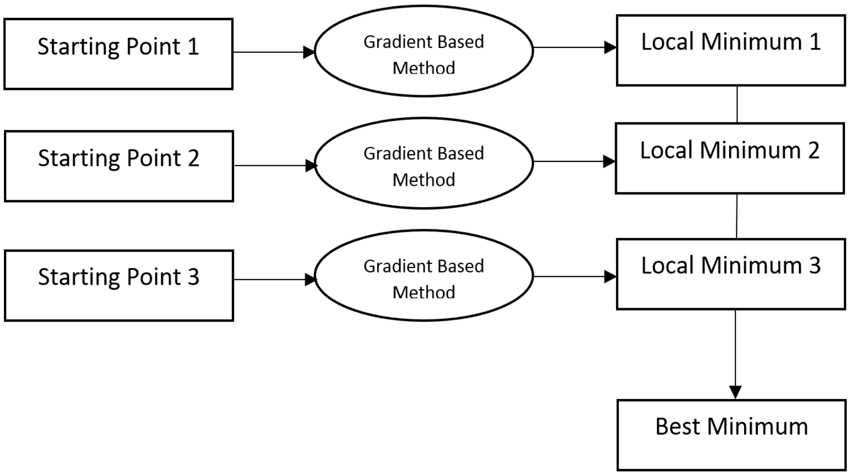


Figure 3.4: FminCon workflow

of proceeding because only the right hand side neighborhood is considered, however is the less computationally expansive method. A central-differences approach would have requested the double the function evaluations, leading to exaggerate computational times for the process. Fmincon is used with a Multi-Start approach: actually this means that different starting points are provided for the gradient procedure. The Multi-Start becomes necessary to avoid premature stops in the iterations caused by the presence of local minima. Starting from a user-provided set of points, the algorithm follows the gradient until a local minimum is found or until the imposed tolerance in the objective function variation is not reached. Then, it starts from the second point and calculating again the gradients, finds an other minimum and so on. The different starting points can be provided by the user or selected randomly by the computer: the first choice is the only available if is necessary the possibility to run deterministic simulations and then recover data.

3.3.2 Pros

In contrast with algorithms like Patternsearch, gradient-based methods are able to keep information from previous iterations by calculating the approximate derivative. Doing this they can move in the steepest descend direction changing all the variables together and not one by one. It is then probable that such a method will converge to the solution more quickly than Patternsearch. Moreover not only the direction but also the magnitude of variable change, is linked to the gradient. The bigger a calculated partial derivative is, the bigger will be the corresponding variable perturbation. Correlating the research mesh dimension with the intensity in the objective variation, allows to overtake even the second critical aspect of Patternsearch i.e. to avoid to come in a few iteration to a coarse mesh and then search only in far away regions of the solution space. Another strong point of Fmincon is to search in a *connected* set of solutions: all intermediate solutions between the starting point and the local minimum are calculated or at least they are feasible and no ‘jump’ is possible into the set of unfeasible solutions. This behavior enables the user to know the solutions field and realize in a few iterations if the results are realistic or not

3.3.3 Cons & Problems

The main limit of the present algorithm is the needed hypothesis of continuity in the response surface. In fact there is no guarantee about this assumption: a discontinuity could occur, for a determined blade, if the worst operating condition would arise in a certain DLC not critical for other blade shapes. For instance over a certain threshold of solidity the driving parameter could not be the maximum tip deflection anymore but rather the buckling constraint. By the way for the machine in exam, this assumption seems to be quite well verified. Another limit in this method is the tuning of simulation parameters: is not easy to determine without any preliminary information which are for instance the correct values of the perturbation for the finite differences evaluation. By setting a too low value, the corresponding variations in the objective function for the perturbed blade could be minimal or under the stop tolerance, and the code could consider that a minimum point, stopping the current iteration. On the other hand an exaggerate change in chord is not recommended when finite difference are evaluated because it is possible to lose important details of the response surface. But maybe the weakest point is the strong dependence of the solution by the starting point decided by the user in the Multi-Start setting. Indeed many starting position lead to very small gradient values, stopping the research prematurely. Then a correct choice of starting point is crucial for a rapid convergence. Furthermore, is highly recommendable to set starting point in feasible zones or with realistic blade shapes, otherwise the code can evolve in unpredictable ways.

Chapter 4

Results

In this chapter are presented the results of the three algorithms explained before.

4.1 Patternsearch

The first code implementation was created using MATLAB's Patternsearch. Its relative ease in the usage and the possibility to retrieve data from previous iterations make the code fast to be developed. In the first implementation a *complete Poll* strategy was adopted: before to determine the best configuration from which to re-start the research, all variables are perturbed. This approach enables to better figure out the objective function but has two main issues

- It is quite slow
- Variations in COE value are similar for different variables perturbed.

Then is better to opt for a strategy that as soon as a new better-score-value is found, restart the *poll* increasing consequently the mesh without waiting for all the perturbation to be computed. First evaluations report improvements in COE by increasing the chord value in the first and second stations. In particular in first iterations the maximum chord as well as the blade solidity value is constantly increased. As said before, after every positive *poll* that namely finds a better objective function value, perturbation mesh is increased by a user defined factor: this is a crucial parameter to keep under control because otherwise the evolution of the computation could rapidly bring the mesh to explosion. Indeed the two first iterations are successfull with the increase in the max-chord first stage. A further growth of maximum chord brings no more benefits to the COE but a better solution is found increasing the second stage. At this point the blade is probably quite near to the optimum but on the other hand the research mesh is very coarse. In

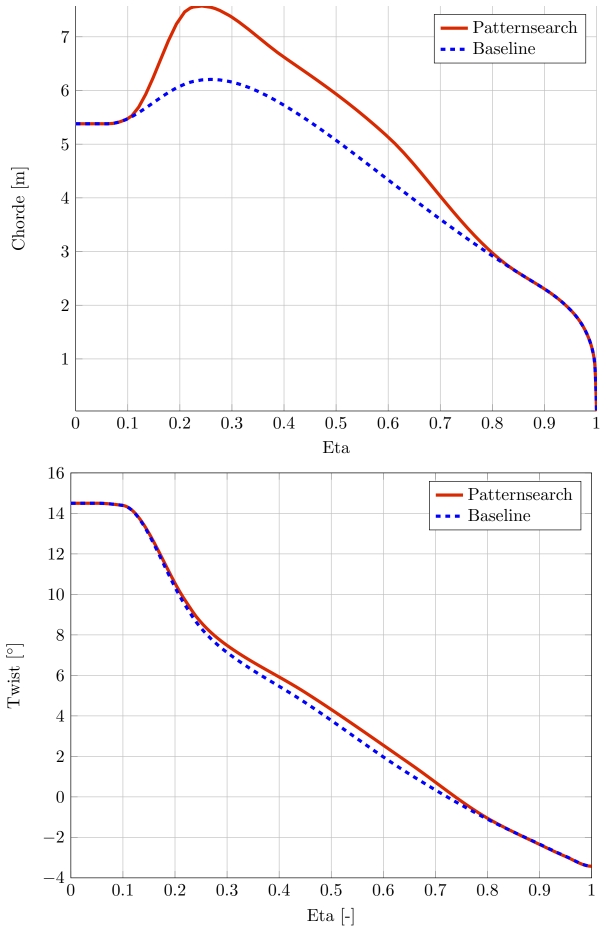


Figure 4.1: Result of Patternsearch optimization

fact the next decrease in the objective function arises after 33 iterations i.e. after three in a row mesh reductions caused by just as many negative *polls*. Last chord station is not modified from the baseline by the optimization algorithm.

Compared to the baseline the solution found is basically a blade with higher solidity, where the increase in blade area is concentrated near the blade root, in the maximum chord point as shown in figure **??**. An indiscriminate boost in maximum chord is however not positive because it is shown that non-structural masses increase consistently with the growth of the panel surface. The non structural filler is used to contain the buckling phenomena, quite important when panel size is getting bigger. In general the solution reached has aerodynamic properties wich are not pushed to the limit (in fact has an augmented solidity) and a consistent decrease in mass is shown in table 4.1.

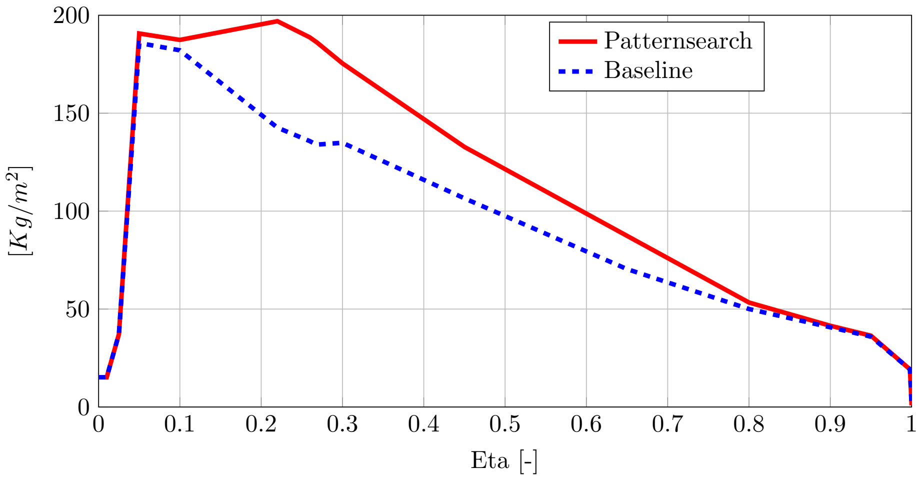


Figure 4.2: Comparison between non structural masses in baseline and optimized blade

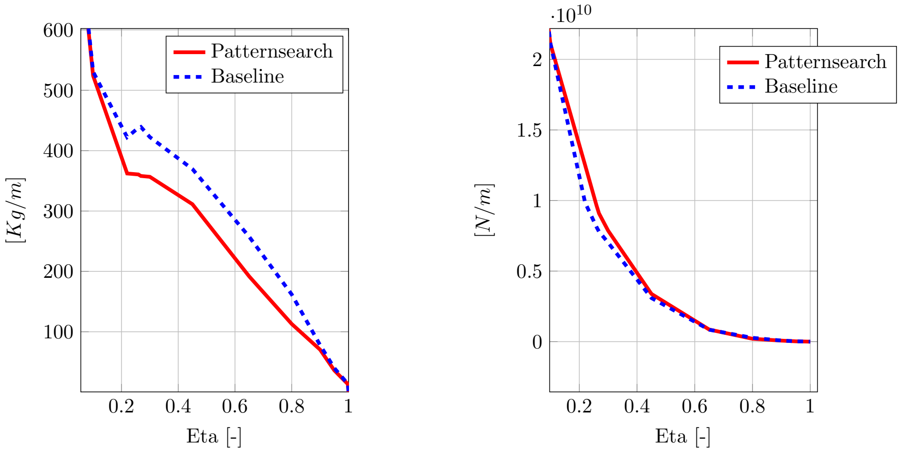


Figure 4.3: Mass and stiffness distributionof the Patternsearch solution compared to the baseline

	Baseline	Patternsearch
Mass [Kg]	41090	38246
AEP [kWh]	4,8993e10	4,8899e10
COE [$/MWh]	74,666	74,536
Solidity [-]	4,66%	5,23%

Table 4.1: Comparison between baseline and patternsearch solution

Big reduction in weight has been shown for the skin and the spar caps and to a lesser extent for other structural elements. This reduction of the structural thicknesses brings to decrease even the rotor inertia in the measure of ca. -21%. The maximum tip displacement constraint can be satisfied thanks to the increased stifness of the region between 20% and 40% of the blade span. A slight reduction of the AEP is outlined. The best C_p is found for a TSR value of 7,4827 decreased from the previous 7,615. Actually the best solution found is clearly not the most efficient from an aerodynamic point of view namely the AEP is lower than the baseline and nevertheless the COE is lower as well. It is important to notice that with a pure sequential aero-structural approach without any additional constraint, this solution would have not been possible to be found. The code reaches a minimum increasing the solidity: this parameter will be kept under observation in the next steps.

4.2 SUMO

Initially, a simplified optimization process with SUMO was tempted. This initial attempt did not implement the fatigue analysis. 36 samples were calculated initially and then the first model was created, then the search strategy selected other samples and other models were built. After every new sample was submitted, the measure was increasing constantly: that means that the previous model was pretty different from the present one. If this situation happens the simulator keeps submitting other samples and the optimization process at a certain moment is stopped because it runs out of maximum time. The impossibility to create a reliable model in acceptable times probably is caused by the irregularity of the response surface and by the elevated time needed for the computation. The relinquishment of this even promising method was then originated by two different factors:

- Elevated computational effort without any assurance about the reliability of the final solution.
- Impossibility to recover data from previous iterations caused by the

	Gain 1	Gain 2	Gain 3	Gain 4	COE [$/MWh]
Best solutions	1,1263	1,25	1,15	1,0526	74,492
	1,06316	1,176	1,097	1,0631	74,494
	1,3	1,047	1,137	1,0158	74,507
Worst solutions	1	0,9	1,15	1,2	74,779
	1,01	0,9184	0,913	1,034	74,799
	1	0,9	0,9	1,2	75,064

Table 4.2: Results of SUMO without fatigue analysis

presence in the algorithm of random parts.

Moreover the lack of a proper documentation and a non-editable source files make impossible for the user to directly intervene on the code to solve the problems. By the way the results of this partial process can be kept to better understand which are the most interesting parameters during the optimization. Results of this model principally predicted enhancement of COE by growing in the first two stages and modest variations in the other two (4.2). As said before, not to early abandon this powerful method in this thesis a Surrogate Model is developed only for the first two variables. This was made because, according to the Patternsearch analysis and the first partial SUMO results, they seem to be the most sensitive to COE variations and because they together describe a quite large field of solidity. The manual construction of a response surface has been seen as the only feasible way to profit by the potentiality of the response surface methods. To do that ooDACE Toolbox was used: it is an implementation of SUMO for the manual building of response surface. First evaluation was made with a deliberately coarse mesh: 9 sampling points (3 by 3 mesh figure 4.4) with variation of chord included in a range of 80-130% of the initial value. The result is a pretty regular and flat COE surface where the minimum is found for solidity higher than the baseline but within a not big range, in accordance to Patternsearch solution. Subsequently a 25 samples refined mesh was considered (figure 4.5). The result shows how the interpolation routine can be caught in a trap with a coarse mesh because local minima are not identified: this refined mesh clarify that there are minimum zones at the boundaries. Then it is necessary to enlarge the mesh and proceed with a new sampling considering more points and checking for the predicted minima. This third model, quite precise but computationally really heavy, shows a solution space absolutely not regular, with several local minima and, by the way, quite flat. It seems difficult to distinguish between real minima and artificial areas created by a not correct interpolation. From the analysis of this response surface it is possible to understand that the adopted cost model (even if it is the only one available in literature) only roughly represent the real objective surface, raising doubts about the real availability

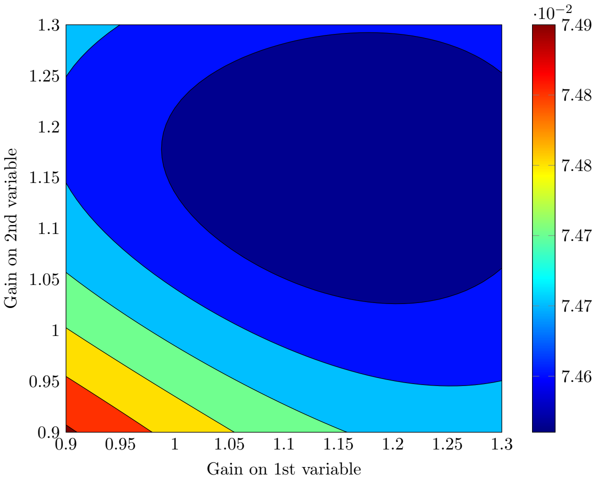


Figure 4.4: 3x3 Kriging Model

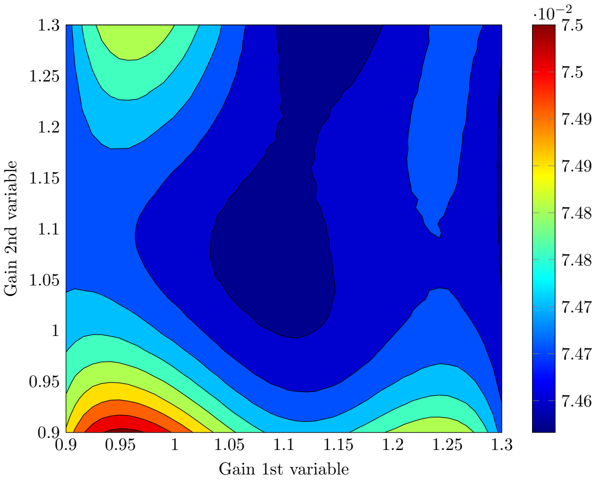


Figure 4.5: 5x5 Kriging Model

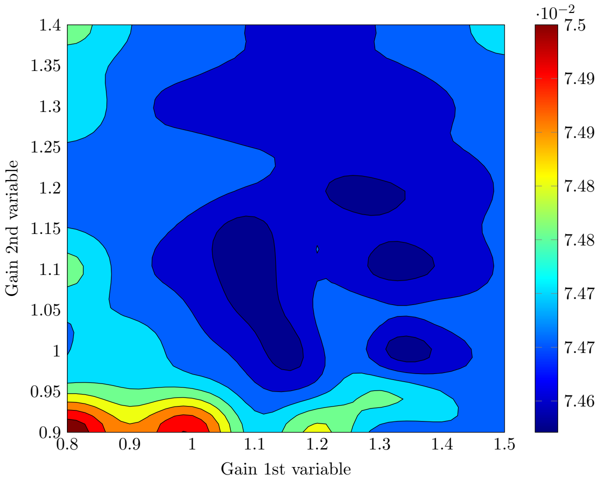


Figure 4.6: Refined Kriging model

of the method (figure 4.6). The positive aspect is that it seems clear now the parameter which synthesizes the best the blade behavior, is solidity. To better investigate this aspect is possible to evaluate separately the two values that together are used to compute the COE: Annual Energy Production and Blade Mass. The surrogate models for each of these parameters show more appreciable variations in the values and help to better understand which are the important factors in the optimization. It is interesting to notice the regularity of mass response surface: this means probably that the leading factors for this optimization are quite always the same, in terms of critical DLCs and critical constrains that are almost the same for every blade. Even the grown of the non structural masses for high-solidity blades is smooth. The most remarkable aspect is the big increase in mass for low solidity blades, caused both by a raise in the aerodynamic loads and a reduction of the maximal transversal section. On the contrary the AEP surface has a maximum next to the baseline and fall for high solidity configuration.

4.3 Fmincon

Let's consider now the gradient-based method. As said before, it proceeds calculating an approximation of partial derivatives, an then follows the steepest descend path of the objective. The starting points selected are the baseline, a configuration with a lower solidity, one slightly higher and the last quite higher.

Starting from the baseline, Fmincon is not able to find significant changes of the objective function in any direction, and in fact, it ends his computation after two gradients evaluation. Then the first aspect to point out is that the gradient method, is not able to find an optimum from the baseline configu-

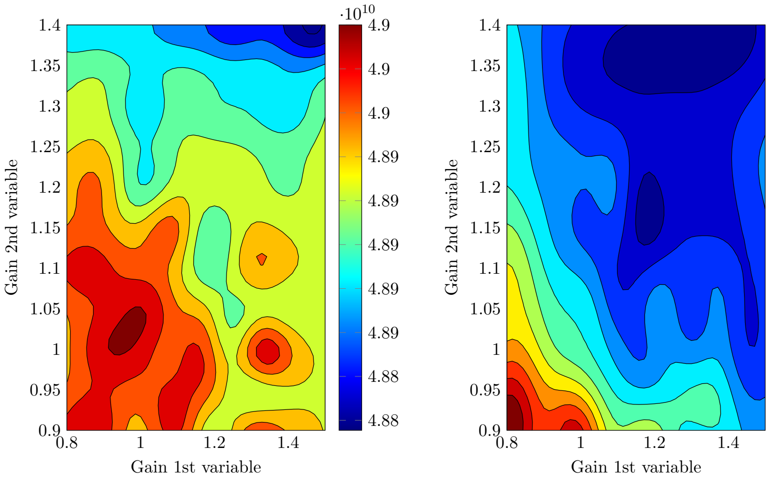


Figure 4.7: Blade Mass and AEP response surface

Starting Point	Chord 1	Chord 2	Chord 3	Chord 4
Baseline	+0%	+0%	+0%	+0%
Low Solidity	+0%	+0%	-10%	-15%
Medium Solidity	+10%	+10%	+10%	-10%
High Solidity	+30%	+20%	+10%	-10%

Table 4.3: Gains of the optimization variables respect to the baseline configuration

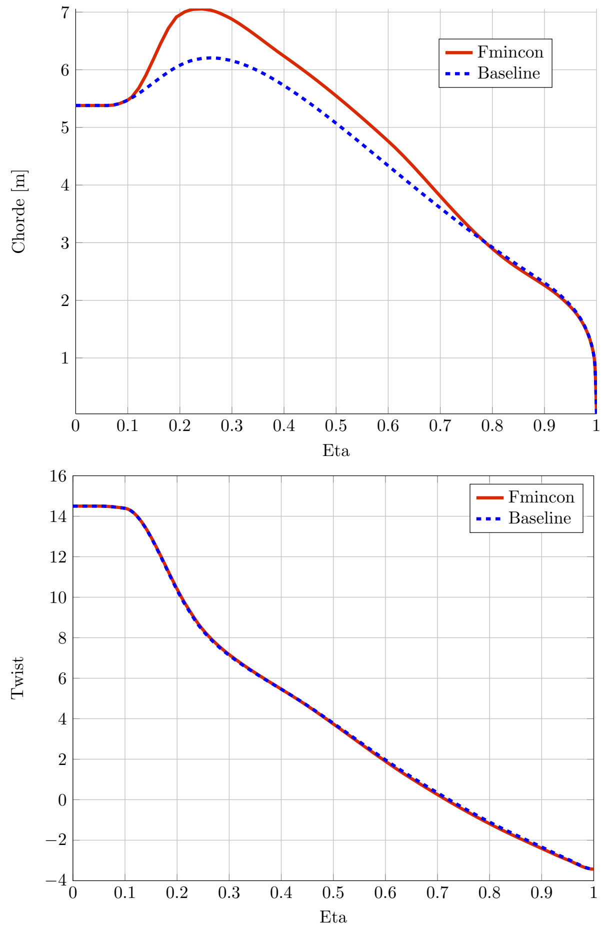


Figure 4.8: Result of fmincon with Multistart

ration (contrary to Patternsearch). The second starting point (mid-solidity) presents since the beginning a better COE value due not to the optimization procedure but to the choice operated by the user. During the optimization process, maximum chord is augmented and the chord in the other three station is slightly decreased (figure 4.8). Actually the blade solidity remains almost constant and the leading factor seems to be the distribution of the solidity along the span (*tapering*). In any case, the solution found is not dramatically different from its starting point: it becomes indeed difficult to figure out if the result is coming from a good work of the algorithm or rather from a lucky choice in the initial condition. In fact all the troubles of this method in continuing iterating are clearly shown also in the evaluation of the third starting blade, the high solidity. The process stops after one evaluation of the gradient not because the starting point is a minimum but instead because the objective surface in that region is extremely flat and the gradients are pretty low. The last starting point is the low solidity blade.

The optimization process follows the gradient quite easily because that zone of the response surface seems not to be flat but near the baseline configuration, the process stops suddenly without finding a minimum. It should be clear from these data that fmincon is not the correct method to use even with a multistart approach for this kind of problem. The small slope of the objective surface makes difficult to calculate gradients significantly bigger than zero and then follow the direction of steepest descent.

4.4 Influence of Solidity

The comparison between various results revealed some important aspect so far not completely well understood: one of these is the influence of solidity on blades performance. The optimization problem faced from a mathematical point of view was

$$COE = f(c_1, c_2, c_3, c_4) \tag{4.1}$$

i.e. the only parameters that can be changed by the optimizer are c_i cwhere c_i are the chord lengths at the i stationConsidering the results all the methods agree about some aspects of the optimum solution:

- Growth of c_1, c_2, c_3
- No variation of c_4

in particular it seems that c_1, c_2, c_3 can not vary autonomously, rather they have in some way to balance in order to have a quite regular blade shape. The c_4 chord value deserves a separate mention because an increase of it, brings to grow the aerodynamic loads and then the root flap-wise moments. To satisfy the constrains the structural elements must get thicker and this makes the blade mass raise. A parameter that condensates this chord behavior is solidity: indeed a variation in a single chord variable is difficult to correlate directly with the COE but on the other hand it is possible to do it with the solidity parameter. The motivation of this fact lies in the problem's physics and in the applied constraints. Indeed for all the blades the maximum tip deflection constraint is the most difficult to satisfy and it is also the leading factor in the structural optimization. Low solidity blades have lower area and aerodynamic properties pushed to the limit, but even smaller cross section areas and then lower bending stiffnesses that lead to higher deformation. To satisfy the maximum tip deflection constraint, structural elements are increased, and so the blade mass grows. Actually the importance of the tip deflection on the final design can be kept only using an integrated aero-structural approach and running accurate aeroelastic simulation of the dynamic behavior of the blade. It is interesting for this pupose to compare the Patternsearch result and the solution found by the pure aerodynamic optimizer of *Cp-Max* in figure 4.9. As outlined above, the pure aerodynamic

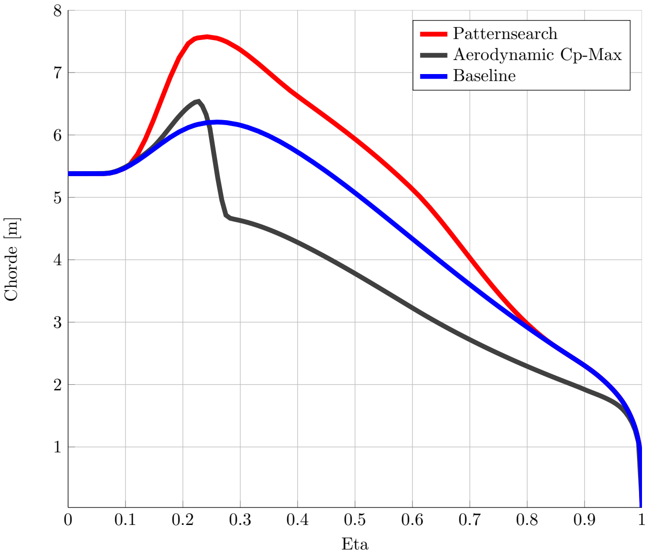


Figure 4.9: Comparison between the optimal aerodynamic solution of *Cp-Max* and the result of the Patternsearch iteration

optimizer is not able to see the influence of the aeroelastic interaction on the blade design. It searches the solution where the AEP is high i.e. where the solidity is low (figure 4.7). A similar blade shape can have big problems in satisfing the tip deflection constraint and therefore needs a big increase in structural thicknesses to reach a proper bending stiffness. For high solidity blades, the maximum tip deflection constrain is not so pressing. A comparison between blades with different solidity is shown in figure 4.10: it is clear how increasing the solidity, the maximum tip displacement drops. Then solidity seems to be an interesting parameter for the optimization process but for this purpose can not be used alone. As said before although the value of the blades total area is important, it is also important how this area is distributed along the span: for a given solidity an infinite number of different blade chord distribution is possible. For instance as outlined before increases in the tip or excessive growths in the maximum chord value are not benefical because of the problem of maximum tip deflection and buckling respectively. A second parameter that well fits with the solidity could be the *tapering* τ, a coefficient that constrains the position of the aerodynamic centers along the span. Solidity is probably a more sensitive value in comparison with chord lengths and the usage of this variable may bring to more regular response surface, easier to be evaluated by the optimization algorithm.

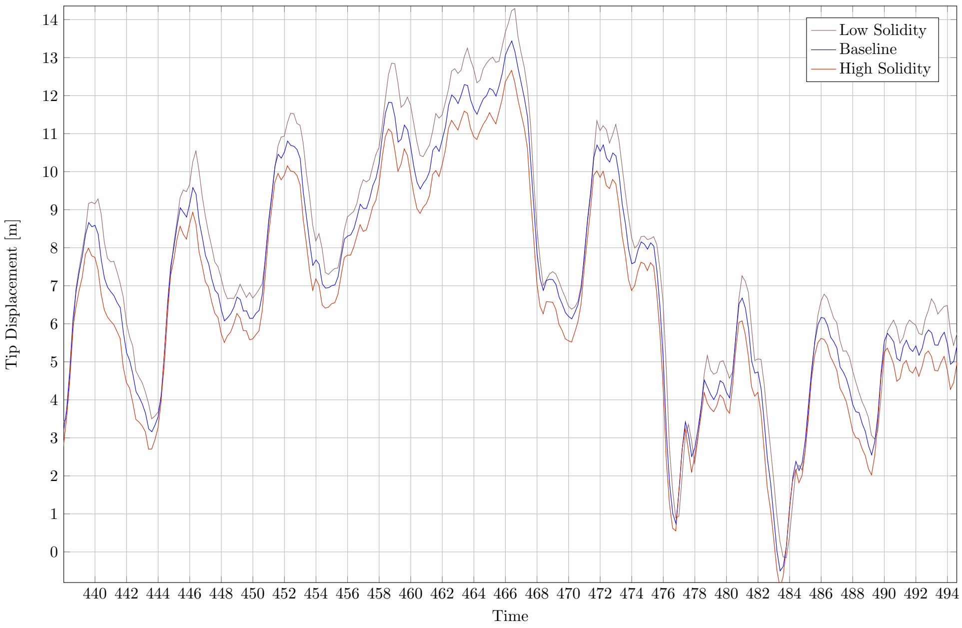


Figure 4.10: Tip displacement for different blade solidity

4.5 Influence of Cost Model

Some considerations about the cost model are necessary. The NREL mode was studied to provide simply scaling law for wind turbines, useful in optimization process. Necessary parameters to evaluate the COE are $RotorDiameter$ and $RatedPower$ but the cost of some components (blades, tower) pass through the calculation of total mass, estimated with a further empirical model. In the equation 2.14 the ICC contribution includes all the initial expanses needed for building and setting up the machine, inclusive of the cost of blades, nacelle, generator and so on. So for a given blade shape, the structural optimizer tries to satisfy all the constraints minimizing in the meanwhile the blade mass, and this changes the COE value because the blade cost is a part of it. But it is important to notice that a variation of the blade shape severely modifies all the loads the machine is subjected e.g. the maximum thrust that influence the design of several components. Hence changing the blade shape, the behavior, the loads and so the cost of many parts of the turbine should change, whereas in this implementation only the blade cost is strictly correlated with the shape. This, besides of being theoretically not much realistic, addresses the problem of marginal COE variation for big changes in blade mass. This happens mainly because the blade total cost is only a marginal part of the entire machine (around 15-20%). Then a relevant decrease of the blade weight is in part anesthetized by other component cost that remain a constant value in the ICC computation.

Moreover this cost model tents to overemphasize the role of the AEP that in fact influences the entire calculation of the cost and not only a part. So a small change in the AEP seems more important than a big decrease in blade mass and loads.

Chapter 5

Conclusions

The present work aims at comparing different algorithms for the aero-structural wind turbine blade optimization and at investigating the intricate coupling between aerodynamic and structural efficiency of a wind turbine rotor for minimum cost of energy COE. First of all a new software platform is coded starting from an existing tool named *Cp-Max*: a frozen-loads structural optimization with a set of non linear constrains and bounds is performed for every different blade design. Before the computation of the dynamic load cases, a twist sub-optimization is performed searching for the maximum power coefficient. Different algorithms for the blade optimization are tested. The first choice is MATLAB's function Patternsearch that shows to have good stability, robustness and capability to find a minimum for the COE function. Subsequently, a response surface method is tested. However, for this particular problem it proves to be onlya useful tool to produce information about the problem and not a proper optimization method. The last implemented algorithm is a gradient-based method with a multi-start approach. This method has been shown to be too sensitive to the starting point and not able on its own to reach a minimum for the COE. This process revealed some important aspects about the optimization of wind turbine blades:

- The importance of solidity as a parameter in the optimization process. Solidity is capable of accurately describe the solution field condensing many parameters in only one and therefore saving a significant amount of computational time.

- The necessity to integrate and consider other components in the optimization process to avoid distortion in the cost function evaluation.

The main goals that can be considered achieved by this work are:

- Provide a valuable and robust tool for wind turbine blade design activities.

- Highlight the influence of the solidity as an important parameter for the design process.
- Outline the necessity for high fidelity cost models to be implemented in optimization codes that take advantage of high-fidelity physical models.

5.1 Future developments

The code has been conceived and developed to be a starting point for future enhancements in all its crucial aspects: convergence properties, research speed and reliability of the solution found. During the work, several aspects and possible enhancements have been pointed out. The most important are enlisted here:

1. Many of different studies and researches use chord and twist distribution as aerodynamic optimization parameters : it is a strengthened opinion coming from years of research in the field of wind turbines. However the new machines pose new and different issues in the design process and it is becoming more and more important to perform aeroelastic simulations during the optimization procedure and this could lead to unbearable computational efforts. From the analysis made using the instruments provided by the code in this thesis, computational times could be reduced significantly considering as variables in the external loop, no more the chord lengths but rather solidity and tapering values witch seem to better represent and condense the behavior of the wind turbine blades. With half of the variables it is then possible to imagine the use of surrogate models, discarded in this work because of excessively high computational costs.
2. The choice of the right merit value for the optimization process is an open issue. In this work it has been decided to implement the NREL scaling model [3] because it is the only cost model available and reliable at disposal. Following the concepts outlined by Ashuri T, et al., [5] it will be possible in the future to implement a cost model where a change in the blade shape is reflected in a change of other turbine components cost. For instance, the tower design is mainly driven by loads experienced by the blades. A drop in blades loads due to a change in their shapes should be reflected in a change of the tower mass and hence its cost. This new approach should result in a more integrated, holistic process and even the advantages of using *high-fidelity* simulation tools should be more evident in the final cost of energy calculation because more components would be involved.

3. Since modern computers are all multi-processors and allow multi-thread calculations, it is possible to imagine a parallelization of the structural optimizer. Indeed a number of part parts in the code are already running in parallel (DLC, AEP, eigenfrequency calculations) but the computation can be further speeded up parallelizing the gradient calculation in the structural optimization process. As already outlined, this is not an immediate process, and requires to re-write several scripts and routines in the code. But the time saved in the computation would widely justify the effort.

Bibliography

[1] A., Ahlstrom: *Influence of wind turbine flexibility on loads and power production.* Wind Energy, 9:237–249, 2006.

[2] Anonymous: *Wind Turbines: Design Requirements.* International Standard IEC 61400, 2005.

[3] Anonymous: *Wind Turbines Design Cost and Scaling Model.* National Renewable Energy Laboratory, 2006.

[4] Anonymous: *Rotoropt perfect rotor design.* LM Glasfiber Newsletter, 2007.

[5] Ashuri T., Zaaijer M.B., Martins J.R.R.A. van Bussel G.J.W. van Kuik G.A.M.: *Multidisciplinary design optimization of offshore wind turbines for minimum levelized cost of energy.* Renewable energies, 00:1–13, 2014.

[6] Bak C., Zahle F., Bitsche R. Kim T. Yde A. Henriksen L.C.: *Design and perfomance of a 10 MW wind turbine.* J. Wind Energy.

[7] Bauchau O.A., Bottasso C.L., Nikishkov Y.G.: *Modelling rotorcraft dynamics with finite element multibody procedure.* Math. Comput.Model., 33:1113–1137, 2001.

[8] Bot ETG, Ceyan O.: *Blade optimization tool user manual.* ECN–E–09–092, 2009.

[9] Bottasso C.L., Croce A.: *Cp-Lambda: User's Manual.* Dipartimento di Ingegneria Aerospaziale, Politecnico di Milano, 2006–2012.

[10] Bottasso C.L., Campagnolo F., Croce A.: *Multi-disciplinary constrained optimization of wind turbine.* Multibody syst. dyn., 27:21–53, 2012.

[11] Bottasso C.L., Bauchau O.A., L. Trainelli: *Robust integration schemes for flexible multibody systems.* Comput. Methods Appl. Mech Eng., 192:395–420, 2003.

[12] Bottasso C.L., Bauchau, O.A.: *On the design of energy preserving and decaying schemes for flexible, nonlinear multibody systems.* Comput. Methods Appl. Mech Eng., 169:61–79, 1999.

[13] Bumsuk K., Woojune K., Sanglaae L. Sungyoul B. Youngho L.: *Development and verification of a performance based optimal design software for wind turbine blades.* Renewable energy, 54:166–172, 2013.

[14] Chittick I.R., Martins J.R.R.A.: *An asymmetric suboptimization approach to aerostructural optimization.* Optim. Eng., 10:133–152, 2009.

[15] Cox K., A.Echtermeyer: *Structural design and analysis of a 10 MW wind turbine blade.* Energy Procedia, 24:194–201, 2012.

[16] Dal Monte A., Raciti Castelli M., Bernini E.: *Multiobjective structural optimization of a HAWT composite blade.* Composite Structure, 106:362–373, 2013.

[17] Dong Ok Yu, Oh Joon Kwon: *Predicting wind turbine blade loads and aeroelastic response using a coupled CFDeCSD method.* Renewable Energy, 70:184–196, 2014.

[18] Dontchev A.L., Zolezzi T.: *Well-Posed Optimization Problems.* Paperback. Springer-Verlag New York, LLC, 2008.

[19] D.R., Jones: *A Taxonomy of Global Optimization Methods Based on Response Surfaces.* Journal of Global Optimization, 21:345–383, 2001.

[20] Fangfang Song, Yihua Ni, Zhiquiang Tan: *Optimization design, modeling and dynamic analysis for composite wind turbine blade.* Procedia Engeneering, 16:369–375, 2011.

[21] Giavotto V., Borri M., Mantegazza P. Ghiringhelli G.: *Anisotropic beam theory and applications.* Computer & Structures, 16:403–413, 1983.

[22] Gorissen D., Couckuyt I., Demeester P. Dhaene T.: *A Surrogate Modeling and Adaptive Sampling Toolbox for Computer Based Design.* Journal of Machine Learning Research, 11:2051–2055, 2010.

[23] Jonkman, B.J., Buhl M.L. Jr.: *TurbSim: User's Guide.* Golden, CO: National Renewable Energy Laboratory, 2007.

[24] Jureczko M., Pawlak M., Mezyk A.: *Optimization of wind turbine blade.* J. of Materials Processing Technology, 167:463–471, 2005.

[25] M.A., Miner: *Cumulative damage in fatigue.* J. of Applied Mechanics, 12:159–164, 1945.

[26] Manwell J.F., McGowan J.G., Rogerso A.L.: *Wind Energy Explained: theory, design and applications.* Wind Power. Wiley, 2009.

[27] Ning S.A., Damiani R., P.J. Moriarty: *Objective and constraints for wind turbine optimization.*

[28] Vesel R.W Jr., McNamara J.J: *Performance enhancement and load reduction of a 5 MW wind turbine blade.* Renewable energy, 66:391–401, 2014.

[29] Wang Xudong, Wen Zhong Shen, Wei Jun Zhu Jens Nørkær Sørensen Chen Jin: *Blade Optimization for Wind Turbines.*