E.T.S. de Ingeniería Industrial,
Informática y de Telecomunicación

Computer assisted aerodynamic design of a 10 kW HAWT blade

Máster Universitario en
Ingeniería Industrial

Trabajo Fin de Máster

Alvaro Olcoz Alonso
Aitor Plaza Puértolas
Pamplona, Junio de 2019

upna Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

Nothing in life is to be feared,
it is only to be understood.
Now is the time to understand more,
so that we may fear less.
Marie Curie

ABSTRACT

This **Final Master Degree Work** proposes the computer aided aerodynamic design of a 10 kW small-scale horizontal axis wind turbine blade based on the **Blade Element Momentum Theory** (BEMT). The method relies on the discretization of the blade span into several elements, for which the main equations need to be solved iteratively. Some improvements in order to increase the accuracy of the method, such as the **tip and hub-loss correction** factors and the correction of the axial induction factor due to the **break-down of the momentum theory**, have been implemented. Thereby, the so called Improved BEMT has been implemented in MATLAB®.

The aim of the project is to design aerodynamically a blade as an element of a small-scale wind turbine that will be placed in the province of Navarre. Indeed, a **wind resource assessment** of the location by statistical Weibull PDF is part of this work. A review of the **power control methods** available for small-scale wind turbines, which results in the selection of the Variable-Speed Fixed-Pitch one is relevant in the design process.

Research has been needed in order to identify the potential airfoils that work properly and are currently being used in small-scale wind turbine blade layouts, where Reynolds numbers are low. Aerodynamic coefficients of the airfoils are obtained with the simulation tool **XFOIL**. In order to use these data in BEMT computations, they are corrected to account for the **rotational effects** of the flow in a wind turbine rotor following the model of Snel et *al.* Then, these data which are limited to a narrow angle of attack range are **extrapolated to 360 °** using the Viterna-Corrigan model.

The blade design implementation is divided in **2 main MATLAB® scripts.** The first one discovers the chord length and twist angle that maximise the glide-ratio (C_l/C_d) at each section in order to obtain the maximum power coefficient at the design point. Once the geometry is fixed the 2nd main script will simulate the **off-design behaviour** of the wind turbine for a working wind velocity range, leading to the power curve of the wind turbine. 4 different airfoils and 2 chord variations are used, which means $4 \cdot 2 =$ **8 different blade designs** (each blade is composed by a cylindrical root followed by a single airfoil until the tip). From the 8 alternatives, 1 is finally chosen to continue towards the work that complements the current one, where the blade will be mechanically designed and tested.

Finally, the designed geometry and the wind turbine model will be simulated with the software **WT_Perf** (NREL) and its results compared with the MATLAB® ones. Hence, the written MATLAB® code will be somehow validated with the existing WT_Perf code.

KEYWORDS: Blade, Design, Wind, Energy, HAWT, BEMT, BEM, MATLAB®

RESUMEN

El objeto del presente **Trabajo Fin de Master** es el diseño aerodinámico asistido por ordenador de una pala de aerogenerador de 10 kW mediante la teoría de **Blade Element Momentum** (BEMT). Este método se basa en la discretización de la pala, dividiéndola así en diversos elementos para los cuales se resuelven las ecuaciones de manera iterativa. Es preciso incluir en la implementación efectos como los de las **pérdidas en la punta de o en la raíz de la pala** no contemplados en el modelo básico para aumentar la precisión en el cálculo de cargas aerodinámicas. También se incluye la **corrección del factor de inducción axial y coeficiente de empuje** cuando el aerogenerador se encuentra en el estado de estela turbulenta. Estas correcciones se implementan con modelos numéricos, y la teoría BEMT pasa a llamarse BEMT "mejorada". Las bases del presente trabajo son la implementación de este modelo en MATLAB® así como su aplicación.

Se ha querido estudiar el diseño de la pala para un emplazamiento en Navarra, incluyéndose así un **estudio básico del recurso eólico** basado en la función densidad de probabilidad (PDF) de Weibull para un pueblo de la provincia. También se han revisado los diferentes sistemas de **control de potencia** disponibles para aerogeneradores de pequeña escala, escogiéndose el control de **velocidad variable y paso fijo** para el aerogenerador cuyas palas son el objeto del trabajo.

Después de investigar sobre los perfiles aerodinámicos con buen comportamiento para números de Re bajos, 4 perfiles diferentes han sido identificados y sus coeficientes de sustentación y arrastre han sido obtenidos mediante simulaciones con **XFOIL**. Estos valores son relativos a modelos bidimensionales, pero debido al movimiento radial del flujo de aire al atravesar el plano del rotor deben de ser corregidos pues en la realidad se experimentan coeficientes de sustentación mayores, sobre todo cerca de la raíz de la pala. Esta corrección se ha llevado a cabo mediante la implementación del modelo de **Snel *et al.***. Posteriormente los coeficientes corregidos se han **extrapolado a un rango de 360 °** de ángulo de ataque mediante el modelo de **Viterna-Corrigan**.

La implementación del diseño de la pala se ha dividido en **2 scripts de MATLAB® principales.** Mediante el primero se obtienen valores de la cuerda y twist en cada sección radial de la pala de forma que el coeficiente de potencia sea máximo. El segundo obtiene las curvas características del aerogenerador en el rango de velocidades de viento deseado. Se generarán **8 alternativas de diseño** (4 perfiles aerodinámicos x 2 distribuciones de cuerdas) pero finalmente **se elegirá 1** para seguir adelante con el trabajo y finalizar su diseño mecánico en el Complemento al Trabajo Fin de Master.

Los resultados obtenidos mediante el código MATLAB® generado se contrastarán con los que se obtienen con el software WT_Perf (desarrollado por el NREL).

KEYWORDS: Diseño, Pala, Aerogenerador, HAWT, BEMT, BEM, MATLAB®

ACKNOWLEDGMENTS

I would like to use this space to thank all who have trusted on me during the time I have been doing this project, even when I did not. Your support has been essential and a piece of this work belongs to all of you. Special mention deserves...

Nire tutorea Aitor Plazak, lehenengo egunatik proiektuarekin laguntzeko prest egoteagatik, nire ideak beti entzun eta gomendioak eskaintzeagatik eta prioektuan duen interesarengatik, MILA ESKER AITOR!

A Paulita, por quererme y apoyarme de manera incondicional en todo momento. ¡Gracias por estar ahí!

Toda mi familia, en especial mis padres, hermana y tío Miguel. Gracias por vuestro interés y por todo el apoyo que me habéis brindado.

Anche il mio proffessore di *Wind and HydraulicTurbines* nella UNIPD, Giorgio Pavesi, non devo dimenticare che è stato lui chi mi ha fatto scoprire il ruolo dell'ingegneria nel disegno dei componenti legati alle energie rinnovabili. Grazie mille!

CONTENTS

INTRODUCTION

Background and motivation

The current work is presented as a **Master's Thesis** (Trabajo Fin de Master) in Industrial Engineering, with a corresponding workload of 18 ECTS. The motivation and knowledge background about wind turbine aerodynamics and design was obtained during the international mobility stance at the University of Padua (UNIPD), within the subject Wind and Hydraulic Turbines.

Scope statement

The project consists mainly in the aerodynamic design, this is, the definition of the surface in contact with incoming air, of a 10 kW small-scale Horizontal Axis Wind Turbine. The design process is carried out in the programming environment MATLAB®, where the iterative algorithm on which the Blade Element Momentum Theory (BEMT) is based has been implemented.

The classic expressions of BEMT do not respect some of the real physical phenomena on wind turbines due to the suppositions and limitations of the theories in which it is based on. Some of them, such as the tip and hub loss effects and the correction of thrust coefficient and axial induction factor can be overcome with proper correction models, which are defined and implemented in this work: Prandtl's tip/hub loss correction factor and the empirical correction in the advanced brake state by Buhl have been used.

The design of a WT blade requires further information, such us the power control type of the wind turbine, the wind resource quality in the location, and several characteristics of the wind turbine components. Therefore, apart from the main objective which is the proper design of a wind turbine blade, some more goals are defined:

- Characterise and assess the wind resource for a suitable location in the province of Navarre. The statistical analysis based on Weibull PDF is a useful tool in order to evaluate a potential wind turbine site location.
- Discover airfoils with good behaviour in small-scale wind turbine applications, where Re numbers tend to be low. Then, obtain aerodynamic coefficient data for those airfoils within a certain angle of attack range. For this task the simulation tool XFOIL was used, which gives data for angles of attack between -25° and 25°.
- Apply correct modifications to the previously obtained aerodynamic coefficients: On the one hand, data need to be corrected to account for the 3D effects of the rotating flow when crossing the rotor section. The radial movement of the air lead to an

increase in lift coefficients, which has been addressed by the implementation of the correction model by Snel et *al*. On the second hand, the corrected data needs to be extrapolated to a wider angle of attack range than XFOIL allows. Viterna-Corrigan model has been implemented to extrapolate the data to 360°.

- Study the power control systems used in small-scale wind turbine industry, especially for a power rating around 10 kW. Select the most suitable one taking into account the wind resource in the desired location.
- Implement the improved BEMT algorithm in MATLAB® in order to create the blade design by discretizing the blade into several elements. These elements are delimited by blade sections, for which optimum values of chord and twist are found. The size of the rotor needs to be determined to give a certain power rating (10 kW). 8 blade design alternatives are created, each one composed by a cylindrical root and a single airfoil, using a total amount of 4 airfoils and 2 different chord distributions. Finally, differences among the blade alternatives are discussed and 1 blade design chosen.
- Simulate the designed blade's performance with the software WT_Perf, developed by NREL, and compare the results in order to validate the written code.

The boundaries of the work also need to be defined. The following task or topics will **not** be discussed in this work and therefore are out of scope:

- The design refinement of the root and tip of the blade, which is usually addressed by CFD analysis in order to reduce the creation of vortices and emission of sound.
- Economical tradeoffs between captured energy and the impact of aerodynamic loads in the overall increase of wind turbine manufacturing cost, which is a task that involves different departments in a real company.

Content

The work is organised with a succession of 9 chapters which are recommended to be read in order. Chapter 1 is a mere introduction to wind turbines and energy and Chapter 2 performs the wind resource assessment for 2 locations in Navarre. Chapter 3 introduces the reader to the airfoil concept and describes the aerodynamic coefficient obtaining and their required modifications. Chapter 4 summarises the theoretical approach of BEMT as well as the implemented numerical corrections. In Chapter 5 power control modes for small-scale HAWTs are discussed. Chapter 6 gives a detailed description of the implemented algorithms with the help of flowcharts, whereas in Chapter 7 all the simulation's results are plotted. In Chapter 8 the results from the implemented MATLAB® scripts are contrasted with those from WT_Perf. Finally, in the last chapter the author's conclusions about the work and future work lines are discussed.

The MATLAB® codes and WT_Perf input and output files are collected in the Appendices A and B, respectively.

Chapter 1

WIND ENERGY CONVERSION SYSTEMS

Energy is a critical ingredient regarding the socio-economic growth of a country. Renewable energy sources, such us wind, solar (photovoltaic or thermal) or hydraulic energy can be a major alternative when it comes to the reduction of the fossil fuels energetic dependency.

Wind energy represents an environmentally friendly and safe option of energy production at a time when global fossil fuel reserves are substantially decreasing. Assuming the actual level of extraction, coal will be depleted in 111 years, natural gas in 50, and oil in 48 [2]. In addition, it is estimated that nearly 10 million MW of wind power are continuously available in the earth's surface wind [3].

A Wind Energy Conversion System (WECS) can be constituted by several subsystems or elements and transforms the kinetic energy of the wind into a different energy form. This energy form could be mechanical (windmills and pumping systems) or electrical (wind turbines). In the latter case an electrical generator is needed.

This theoretical chapter intends to make the reader familiar with WECS, more specifically with wind turbines. A brief historical review, the current world and national status of wind energy, different wind turbine classification criteria and a short description of the constituting elements of a modern wind turbine will be shown.

1.1 Historical review, and current status of WECS

1.1.1 Historical development of WECS

Humans have been using wind energy to propel ships and boats for at least 3000 years. Later, wind energy started to be used in order to generate mechanical power in a shaft. However, experts disagree on the location where this concept was born.

The earliest documented windmill design was found in the Persian empire, in the year 200 B.C. It was used to grind the grains and consisted in a vertical axis to which some sails were attached. An approximation of the shape of this ancient windmill is shown in Figure 1.1

During several decades the use of wind energy was only used for sailing and the use of wind for grain grinding expanded slowly.

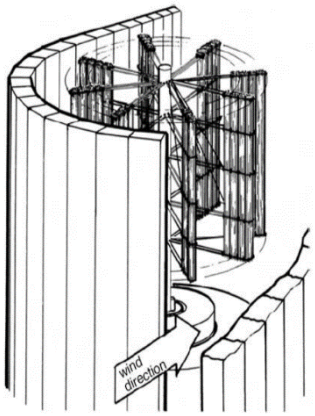


Figure 1.1 Illustration of ancient Persian wind mill [4]

By the 13th century grain grinding mills were popular in most Europe. These European windmills differed from the ones used by the Persians because the orientation of the shaft was horizontal. The tower was usually cylindrical and built of bricks.

In the following years, this windmill configuration was developed, mostly in Denmark, where first twisted blades appeared. This idea of twisted airfoils is used nowadays in modern high technology wind turbines as they make the aerodynamic efficiency of the blades higher. These Dutch windmill model arrived to America by mid-17th century.

In the mid-18th a metallic multi bladed wind turbine for water pumping (Figure 1.2) was launched.

Figure 1.2 Typical American multi-blade windmill

The first modern wind turbine for energy generation was constructed in Denmark in 1890. However, during the 19th century grid connection was not developed and the use of these DC turbines was, manly, battery charging. Once the access to the grid was more

available, new rotor design procedures appeared, leading to low solidity innovative rotors. The blades were aerodynamically designed and the performance was impressive.

The first utility scale (100 kW) system was installed in Russia in 1931. After this, several experimental wind plants were constructed in USA, Denmark, France, Germany and UK. New electrical technologies, improvements in material mechanical resistance and innovative design procedures boosted the research ambition of these countries.

By 1970, cost of electricity from fossil fuels was much cheaper that the one from wind turbines, and some nuclear projects were embarked on. Consequently, interest in wind turbines declined gradually. Nevertheless, with the oil crisis in 1973 investment and research in wind turbines was recovered.

Relevant research was carried out in USA by NASA. In 1975 they constructed the MOD-0 wind turbine. It was a 38 m diameter 100 kW and 2 blade wind turbine. They used top technological advancements for the design. After MOD-0, NASA launched the MOD-0A (200 kW), MOD-1 (2000 kW) and MOD-2 (2500 kW) wind turbines. Rated power and rotor diameter consequently were augmented from one wind turbine to the following.

In the 80s a three bladed, fixed-speed stall-regulated turbine with a simple induction machine configuration became popular and was widely installed. The aesthetic of these kind of wind turbine may appear similar to the current ones, but technologically they have several differences.

Nowadays, large wind turbines work at variable rotational speed at wind velocity below rated, following the optimal power curve (MPPT with a controller). Above rated, the power is regulated by pitching the blades. This leads to an active control of the angle of attack on the blade.

Figure 1.3 Siemens Gamesa 3.4 MW modern wind turbine [5]

Variable speed can be achieved by placing an electronic converter between the grid and the generator. Thus, rotational speed and the frequency imposed by the grid are decoupled. State of the art multi MW machines usually use a Doubly Fed Induction Generator (DFIG) so that only a part of the electricity passes through the converter reducing losses. The stator of the machine is directly connected to the grid (see Figure 1.4). The chief drawback of this configuration is that only a variation of around 30 % each side of the synchronous speed can be achieved. However, most advantages of variable speed operation can be achieved in this range [3, p.174]

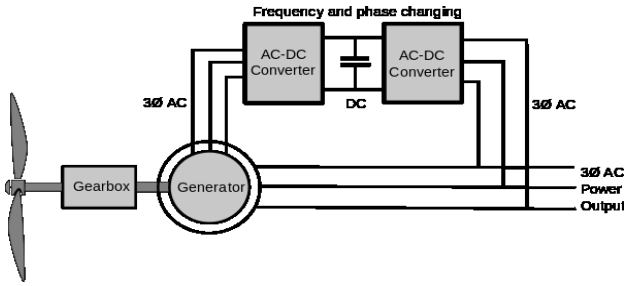


Figure 1.4 *DFIG machine connection scheme*

1.1.2 Wind energy in the world

Installed wind power in the world has increased impressively over the last years. In fact, Figure 1.5 shows that the installed wind power in the world is increasing exponentially. The total value of installed power (2017) is of **540 GW** approximately.

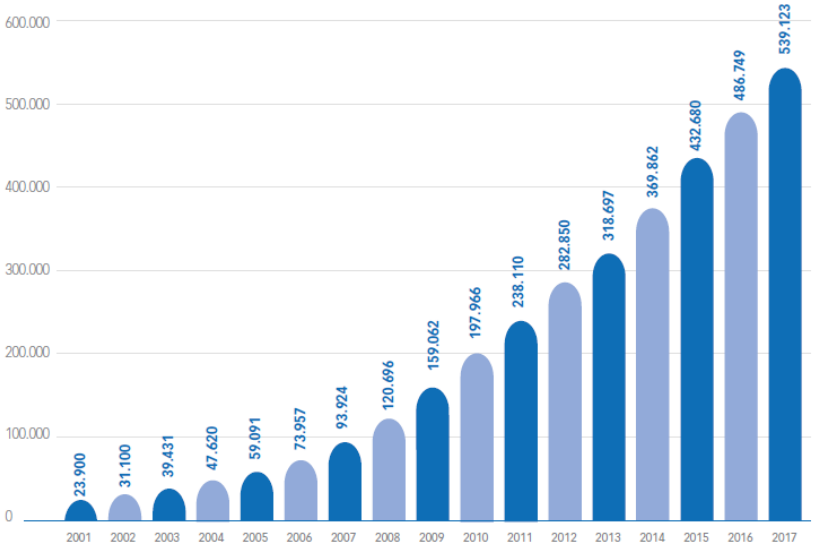


Figure 1.5 Evolution of the installed world wind power, measured in MW [6]

Countries that contribute most to the world installed wind power are China, USA, Germany, India, Spain and UK respectively [6]. How installed power is distributed among countries is depicted in Figure 1.6 . It can be seen that China leads the world ranking, with an installed power even higher than the sum of the following 3 countries.

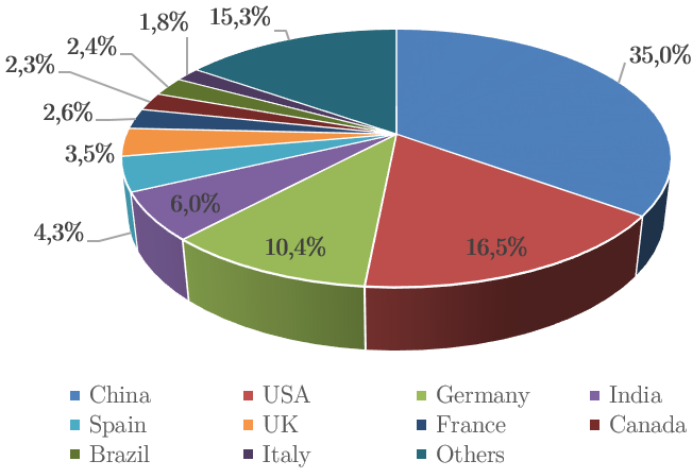


Figure 1.6 Installed wind power country ranking (2017)

Nevertheless, Spain is very well positioned as it has a much smaller surface than the countries positioned above. In the next section the current status of wind energy in Spain will be analysed, not only in terms of installed power, but also in the fraction of annual produced energy that comes from wind energy

1.1.3 Wind energy in Spain

The following figure depicts the evolution of the installed wind power in Spain over the last years. Dark blue bars represent the total wind power that was installed in Spain each year whereas light blue bars represent the new wind power that was installed that year. The picture shows a great increase rate between 1998 and 2012, with a maximum of new installed power in 2007, but the curve appears flat from 2012 to this day. A reason might be that wind energy is currently quite exploited in Spain, and this makes impossible to maintain that past years' increase. However, political decisions could have also been somehow responsible of this stagnation.

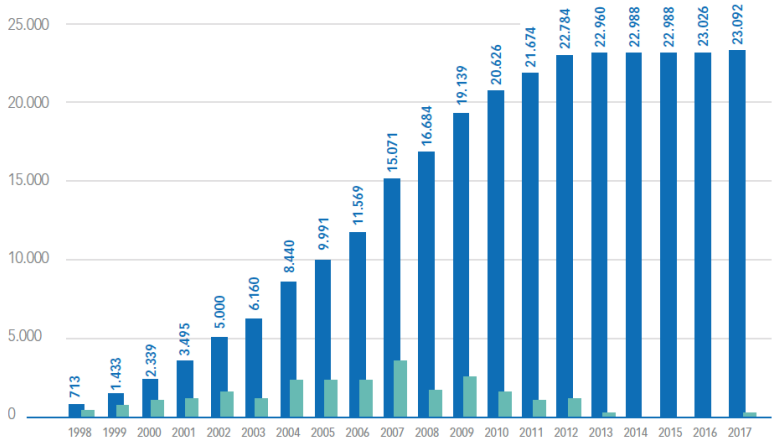


Figure 1.7 Annually installed and cumulative wind power evolution in Spain (MW) [6]

At the end of 2018, *Red Eléctrica Española (REE)* has made a final prevision of about **262 000 GWh** in the national annual energy production [7]. Figure 1.8 shows the fraction of this amount of energy that comes from each energy source. Wind energy has produced the **19 %** of the annual energy production.

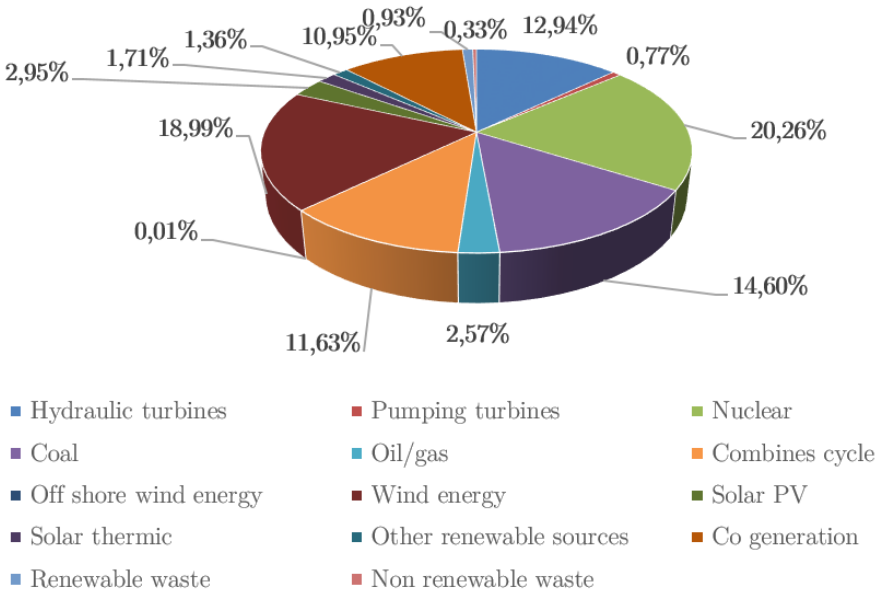


Figure 1.8 Electric energy generation coming from each source in 2018

Regarding the overall installed power in Spain, *REE* estimates a value of **104 031 MW** at the end of 2018 [7]. Figure 1.9 shows the weight of each energy source in the overall installed power. It can be seen that **22,47 %** of the total installed power comes from wind power (on shore + off shore).

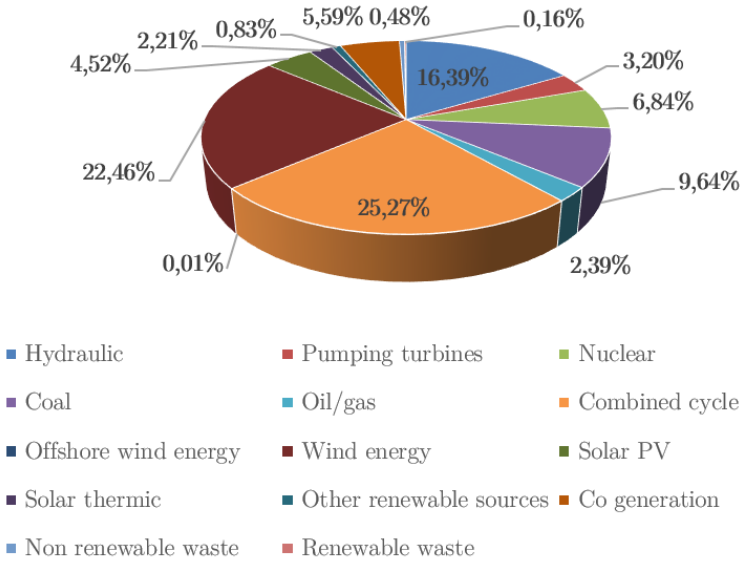


Figure 1.9 Energy source composition of the total installed power in Spain (2018)

1.2 Wind turbine classification

In this section, a short wind turbine classification review according to several criteria will be shown. Apart from the criteria discussed below there are some other than can be considered when one wants to classify a wind turbine, such us the power control method or the generator type. These topics and their importance on the blade design are treated deeper in Chapter 5.

1.2.1 Classification according to the design

1.2.1.a Horizontal Axis Wind Turbines (HAWT)

A wind turbine is defined as a HAWT when rotor blades are connected to a horizontal shaft. These kind of turbines are very common for commercial use. An example can be seen in Figure 1.3.

HAWT can have the rotor **upwind**, when it faces the incoming wind, or **downwind**, when the air passes the tower and nacelle before it interacts with the rotor. Nowadays, upwind configurations dominate the market, as they are easier to orientate in wind direction [8].

The advantages of HAWTs are the high efficiency, the pitch angle capacity and the possibility to have a high tower (note that available power increases with the cubic of the velocity, and velocity increases with height). There are also some downsides, as generation of noise, killing of birds or visual impacts [3, Ch.11]

1.2.1.b Vertical Axis Wind Turbines (VAWT)

Wind turbines with vertical axis configuration (VAWTs) are less common in the energy production sector but they also have some advantages. One of the major upsides is that they are cross-flow devices so they accept wind from any direction and need no yawing control system. Also, the drive train is located at the ground of the base of the tower, making maintenance easier. The most popular configurations are the Darrieus and Savonius rotors.

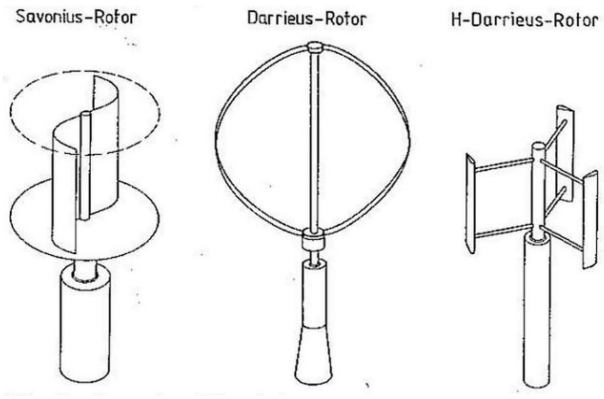


Figure 1.10 VAWT rotor types [8]

1.2.2 Classification according to the size

1.2.2.a Small scale wind turbines

Small scale wind turbines are those whose power rating is lower than 50 kW, and are usually intended to supply electricity to single houses, factories, farms or buildings [9]. The current project aim is the design of a blade for this kind of turbines.

However, there is no worldwide consensus about the maximum rating of a small wind turbine. For instance, the Canadian Wind Energy Association considers a small scale wind turbine until it reaches 300 kW rated [10]. Alternatively, the standard IEC 61400-2:2006 defines as small scale a wind turbine with a rotor swept area < 200 m^2 [11].

Figure 1.11 Small-scale (10 kW) wind turbine example [12]

1.2.2.b Large scale wind turbines

The utility scale wind turbines are those whose aim is the massive electric power generation. They are connected to the general electric grid so that the generated electric energy can be transported. Their rating moves from a few hundreds of kW to some MW and they are usually grouped in what is called a windfarm in a location with a good wind resource. An example of this kind of wind turbine can be seen in Figure 1.3

1.2.3 Classification according to the location

1.2.3.a Onshore wind turbines

The wind turbines that are placed in the land are the most typical. All the figures above in this chapter depict onshore wind turbines.

1.2.3.b Offshore wind turbines

Offshore wind turbines are placed in water bodies (generally in the ocean) and also organized in windfarms. One of the advantages of the offshore wind energy is the availability of higher wind velocities. However, the construction and the maintenance are much more difficult.

1.3 Modern HAWT component description

The majority of the components of a modern large scale HAWT are shown in Figure 1.12. Nevertheless, some of them are not present in some small scale HAWT layouts, such us the tower hollow, the transformer or the pitch or yaw systems. Some other components are not specified in the figure and may exist in a wind turbine configuration.

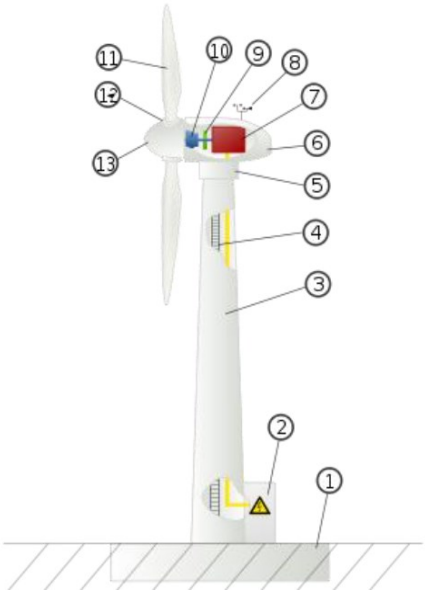


Figure 1.12 Modern HAWT component scheme

1) Foundation:

Aerodynamic forces and the system weight is reacted in the ground by means of the foundation. The foundation generates a proper stress distribution in the ground and has to be correctly designed to avoid the tilt of the turbine.

2) Grid connection

The generated electricity is linked to the electric grid by means of a transformer that elevates the voltage in order to reduce losses in the energy transport lines.

3) Tower

The tower increases the height of the rotor in order to interact with higher velocity winds. Even so, the height of the tower has to be based on an economic tradeoff between increased energy capture and the tower cost [8]. The tower is the link between the nacelle and the foundation. There are several types of towers, which will be chosen depending on the turbine characteristics.

4) Tower hollow

The tower hollow allows workers to go up to the nacelle for maintenance reasons.

5) Yaw system

Turbines with yaw systems can orientate towards the incoming wind direction.

6) Nacelle

The nacelle is the enclosed envelop where the drive train, generator, inverter etc. are supported. It offers them protection from outside conditions.

7) Electric generator

The electric generator converts the mechanical energy of the high speed shaft into electric energy based on electromagnetic induction principles.

8) Anemometer

The anemometer is usually positioned at the top of the nacelle. The anemometer measures wind velocity and are usually accompanied by a wind vane that measures the wind direction. There are several types of anemometers, being cup anemometers the most used in wind turbines.

9) Gearbox

Rotational speed is increased between the low speed shaft and the high speed shaft in order to reach the velocity (or velocity ranges) imposed by the generator.

10) Low speed shaft

The low speed shaft receives the torque from the hub. It rotates at low rotational speed. As rotational speed is low, torque is high so it needs to be robust in order to resist the high tangential stress.

11) Blade

Blades constitute a very important part in wind turbine design, and are the object of this work. When air interacts with blades, forces appear in the blade, being the ones in tangential direction responsible of creating a torque that is transmitted to the slow shaft.

12) Pitch system

Modern turbines have a variable pitch power regulation system. Thus, when rated power is achieved blades are rotated with respect of their own axis, controlling the angles of attack and the consequent power generation.

13) Hub

The hub connects the blades to the slow shaft and drive train and must resist all loads from the blades. It is usually made of ductile iron and covered by a cone shape in order to enhance aesthetics and reduce visual impact [8].

Others:

When the machine works at variable speed power electronics are needed, and a **converter** is placed between the generator and the grid to decouple rotational speed from grid frequency.

Chapter 2

WIND RESOURCE ASSESSMENT

An air mass m flowing upstream of the wind turbine has a kinetic energy given by the following expression:

$$E_c = \frac{1}{2} m {U_\infty}^2 \tag{2.1}$$

This equation can be written in power terms, leading to the well-known equation that expresses the available power in the wind:

$$P_{wind} = \frac{1}{2} \rho A {U_\infty}^3 \tag{2.2}$$

As it can be seen, available power in wind depends on the air density ρ, the air section through which air is flowing A, and the cubic of the free stream velocity U_∞. In a HAWT, section A is the area swept by the rotor blades, calculated as $A = \pi R^2$, where R is the radius of the rotor.

Equation (2.2) shows the high importance of wind velocity on available power in wind. As it will be seen, wind velocity varies both spatially and temporally among the world, and this means that some emplacements will be better than others in order to install a wind turbine. In practice, statistical methods are used in order to study the potential of a future wind turbine site.

In this chapter the most important variations of wind will be briefly examined. Then, statistical tools used to evaluate a wind emplacement will be reviewed. Finally, this tools will be applied in order to find a proper wind site in Navarre for the small wind turbine of the current project. A MATLAB® script has been written with this purpose.

2.1 Nature and variations of the wind

The elemental driving force of air is a pressure difference between two air regions. The air pressure can be described by the ideal gas law:

$$pV = nRT \tag{2.3}$$

So, air pressure is affected by the temperature and as global heating is not uniform, different temperatures in the atmosphere lead to pressure gradients, and therefore, to the movement of air mass, which is called wind. Solar radiation, water evaporation, cloud covering and surface roughness play an important role in the meteorological study of wind [4].

As wind is highly affected by surface solar heating, it varies both **spatially** and **temporally.**

On spatial large scale, there are some climatic regions that are windier than others. Among the same climatic region, wind is affected by geographical elements as mountains or plains. Locally, wind can be significantly reduced by obstacles.

On time scale, wind varies from one year to the others in a manner that is not well understood and is difficult to make accurate economic predictions of windfarms. However, the seasonal variation within a year are much more predictable. Diurnal variations are also fairly predictable. Short-term variations such us turbulence or gusts are also important for the mechanical design of a wind turbine.

- **Wind velocity variation with height**

In practice, the variations mentioned above are captured by meteorological stations, which measure continuously wind speed and direction, and then storage values for the mean wind speed (usually each 10 min) with the standard deviation of all the values within the 10 minutes from the mean. These measurements are made at a reference height, which is usually **2** or **10 meters.**

The airflow above ground is retarded due to friction with earth surface, known as boundary layer effect. Therefore, the wind velocity profile increases when so does the height above ground. From an engineering point of view, the potential law (Equation (2.4)) offers a good relation between wind velocity at a certain reference height $u(z_1)$ and wind velocity at a different, arbitrary height $u(z_2)$.

$$\frac{u(z_2)}{u(z_1)} = \left(\frac{z_2}{z_1}\right)^{\alpha} \tag{2.4}$$

Taking the meteorological station data as reference, the velocity at the wind turbine hub height can be calculated easily as follows:

$$u(h_{hub}) = u(h_{ref}) \left(\frac{h_{hub}}{h_{ref}}\right)^{\alpha} \tag{2.5}$$

The exponent α depends on the surface roughness and the atmospheric stability. For neutral atmospheric stability a value of 1/7 is typically taken [13]. However, the value of **0.2** suggested in the **IEC 61400-1:2005** standard will be taken [9, p.26].

Equation (2.5) is very useful in wind site resource assessments, when velocity at hub height is required. It will be applied in Section 2.3.

2.2 Analysis of wind data. Weibull statistics

For estimating the wind energy potential of a site, the collected wind data from the possible location needs to be properly analysed. Modern wind measurement systems usually placed in meteorological stations give us the mean wind speed at the site, averaged over a fixed time period, which is usually 10 minutes. Then, this data need to be arranged into the desirable total time. A minimum of a year analysis is recommended due to seasonal variations of wind. Once the data is selected and properly scaled to hub height according to Equation (2.5), statistical tools are used in order to perform the wind resource assessment of the site.

Over all the wind velocity measurements the mean of the set and the standard deviation of the data are given by Equations (2.6) and (2.7).

$$\bar{u} = \frac{1}{n}\sum_{i=1}^{n} u_i \tag{2.6}$$

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(u_i - \bar{u})^2} \tag{2.7}$$

Wind data is usually organized by dividing the measured velocity range in certain width bins (usually 1 m/s or less) and counting the frequency with which velocities in this range happen. Thus, data is organized in frequency diagrams, where wind hours for each span are always plotted. Data can also be organized in a cumulative way.

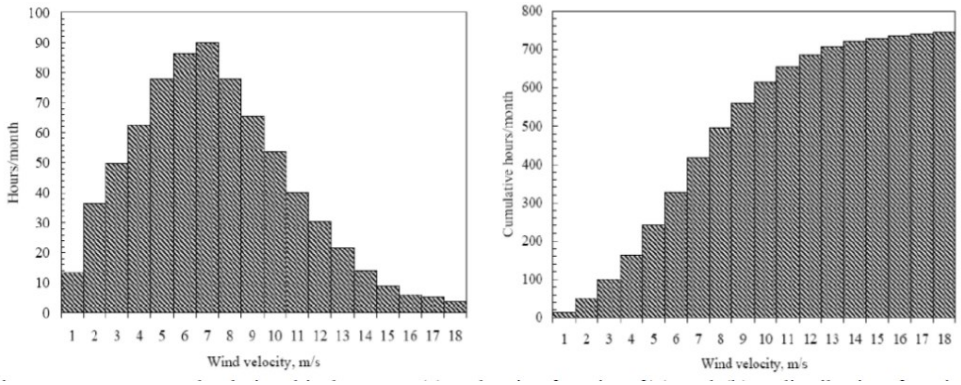


Figure 2.1 Frequency (left) and cumulative (right) diagrams

The same study can be done from a **probabilistic** point of view. Let m_i be the number of observations of the specific velocity u_i . The probability of this event to happen will be:

$$p(u_i) = \frac{m_i}{n} \tag{2.8}$$

Where n is the number of total recorded events. The sum of all probabilities will be unity.

In this case, the cumulative values F(u_i) represent the probability of having a wind speed equal or lower than u_i.Therefore, it can be seen that F(∞)=1.

$$F(u_i) = \sum_{j=1}^{i} p(u_i) \tag{2.9}$$

This approach gives a good perspective of the wind characteristics but it is not convenient at all because a lot of data needs to be managed. We could think now in a way to approach the previous data with a continuous function.

This is done by means of the **Probability Density Function** (PDF), noted as f(u). Mathematically it represents the probability that the wind is in a 1 m/s interval centred in u. The area under the density curve is always 1.

$$\int_0^\infty f(u)\ du = 1 \tag{2.10}$$

There are several density functions which can be used to describe wind velocity frequency curve. The most used are the **Weibull** (2 parameters) and Rayleigh (1 parameter) functions. In this work the former one will be used due to its versatility.

THE WEIBULL DISTRIBUTION

The Weibull probability density function is given by:

$$f(u) = \frac{k}{c}\left(\frac{u}{c}\right)^{k-1} exp\left[-\left(\frac{u}{c}\right)^{k}\right] \tag{2.11}$$

Valid for positive values of u, k and c. Where k is the **shape parameter** and c is the **scale parameter** of the distribution function. Figure 2.2 shows how the curves vary for constant c = 1 and variable shape parameter k. It can be seen that when k increases it becomes more narrow and the peak moves rightwards. If c is different from 1, values of the y axis are divided by c. A higher than 1 values of c makes the curve wider and shorter.

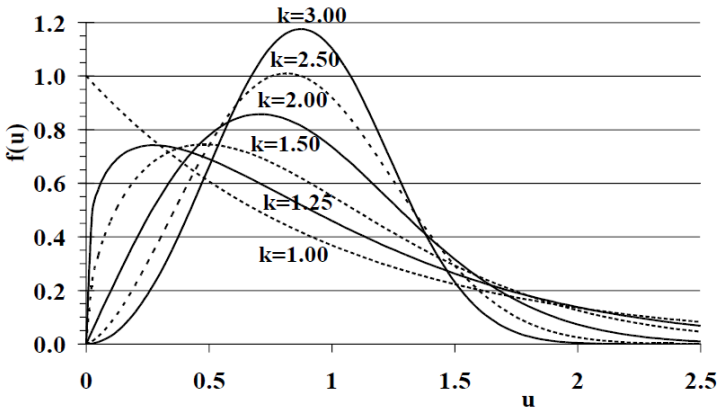


Figure 2.2 Weibull density function f(u) for scale parameter c = 1 [4]

In order to determine the Weibull parameters several methods are available in the literature: the gamma function method, linear regression method, momentum method, maximum likelihood method… [4], [13].

In this work the MATLAB® function ***wblfit*** (data_vector) is used, which returns the values of the scale and the shape parameter respectively. For the computation, MATLAB® uses the maximum likelihood method:

$$k = \left[\frac{\sum_{i=1}^{n} u_i^k \ln(u_i)}{\sum_{i=1}^{n} u_i^k} - \frac{\sum_{i=1}^{n} u_i^k}{n} \right]^{-1} \tag{2.12}$$

$$c = \left[\sum_{i=1}^{n} u_i^k \right]^{1/k} \tag{2.13}$$

Once the fit of the data is done (which usually is quite acceptable) the potential annual energy content of each wind velocity can be studied.

The energy probability density curve is defined as the power available in wind times the PDF:

$$P_u = \tfrac{1}{2} \rho A u^3 f(u) \quad [\mathrm{W}] \tag{2.14}$$

The area under this curve times the number of hours in a year represents the annual energy available in wind for that site. The energy available within a velocity range can be calculated setting the proper integration limits for the curve.

Among all the velocities there is one, $\mathbf{u_{me}}$, which contains more probable energy than the others. This velocity can be easily identified looking at the maximum of P_u. The value can be obtained by equating the derivative of the function to 0.

$$d\frac{P_u}{du} = d\frac{\frac{1}{2}\rho A u^3 f(u)}{du} = \frac{1}{2}\rho A \; d\frac{f(u)u^3}{du} = 0 \tag{2.15}$$

Note that $\mathbf{u_{me}}$ depends neither on the rotor swept area nor the density. It can be seen that in order to calculate $\mathbf{u_{me}}$ it is enough to solve u for:

$$d\frac{f(u)u^3}{du} = 0 \tag{2.16}$$

A good turbine design should include this velocity with maximum energy content into the turbine best operating wind speed range (where $C_p = C_{p_{max}}$) [4].

The use of a PDF to characterise wind is very useful in order to design a proper turbine for a wind emplacement or alternatively for a purchaser who knows the wind characteristics of the land and wants to select an already existing commercial wind turbine. Knowing the PDF of the site and the power curve of the turbine, Annual Energy Production (AEP) can be estimated.

2.3 Wind turbine site selection

One of the objectives of this work is to **choose a proper site** in the province of Navarre in order to place the small wind turbine with the designed blades. AEP and mechanical validation of wind turbines depend strongly on the site wind characteristics.

All the followed steps in the wind resource assessment are shown in the flowchart and described below.

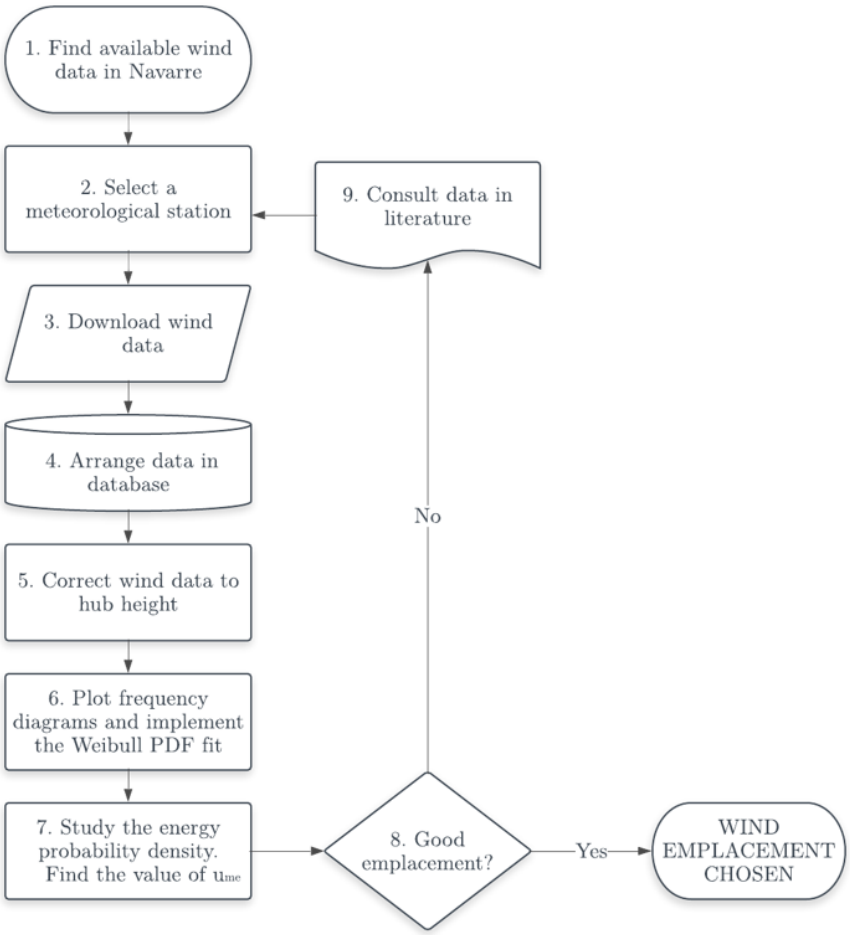


Figure 2.3 Process diagram of the wind resource assessment

- Steps 1-4: Data acquisition and storage

Meteorological data from several stations in Navarre is public and available. It can be found in the **Rural Development and Environment section** of the Navarre's Government webpage [15].

The next step is to select a meteorological station among all the available ones. Initially, the station of the Public University of Navarre (placed on ETSIA) was chosen. Each station is pointed in Navarre's politic map and provides important data such as the height of reference, the latitude, longitude and altitude.

Average wind speed each 10 minutes needs to be downloaded (.xlsx format). Due to strange reasons data can only be downloaded for 1 month each time. Therefore, data is then arranged in a column placing each monthly vector one behind the other (Wind_Data.xlsx). Data from November 2017 to November 2018 was used.

- Steps 5-8: MATLAB® aided statistical analysis

Once data is collected it needs to be treated as explained in 2.1 and 2.2. A MATLAB® script has been written for the wind resource assessment which can be found in the Appendix A.1.

First, the velocity data is scaled to the hub height using the power law (Equation (2.5)). A hub height of **25 m** is taken, imitating the configuration of the 10 kW rated SWRT [16] wind turbine developed by the National Renewable Energy Laboratory (NREL) of USA.

The corrected data is organized into frequency and cumulative diagrams in an absolute and also probabilistic way. Then, the discrete probabilities are adjusted with a Weibull PDF.

Finally, multiplying the available power of the wind times the time in a year that each power occurs the energy probability density function is shown. For this analysis a base (sea level) density value of 1.225 was taken. However, as it will be seen in Section 6.1.5, air density depends on the altitude of the wind turbine site and needs to be properly analysed as it affects proportionally the power production.

The maximum point of the energy probability density curve ($\mathbf{u_{me}}$) represents the wind velocity with higher annual energy content.

- Step 9: Literature on wind distributions in Navarre

In order to find a suitable wind site (with enough wind potential but not in a hill, because the production is for domestically use), a work where several wind emplacements in Navarre were analysed [13] has been used. In this work wind data was fit with Weibull PDFs. The work helped to identify a wind location with a reasonable wind potential. The meteorological section of Navarre's government also provides information about potential wind sites [17]. All the results obtained are illustrated in the following section.

2.3.1 Public University of Navarre (ETSIA)

The first thought when thinking of a suitable place to install a wind turbine with the designed blades was the same Public University of Navarre. Thus, 1-year wind data was collected from the meteorological station of the Superior Technical School of Agronomy Engineering.

Wind data was arranged in 1 m/s width bins and plotted in frequency and cumulative bar diagrams:

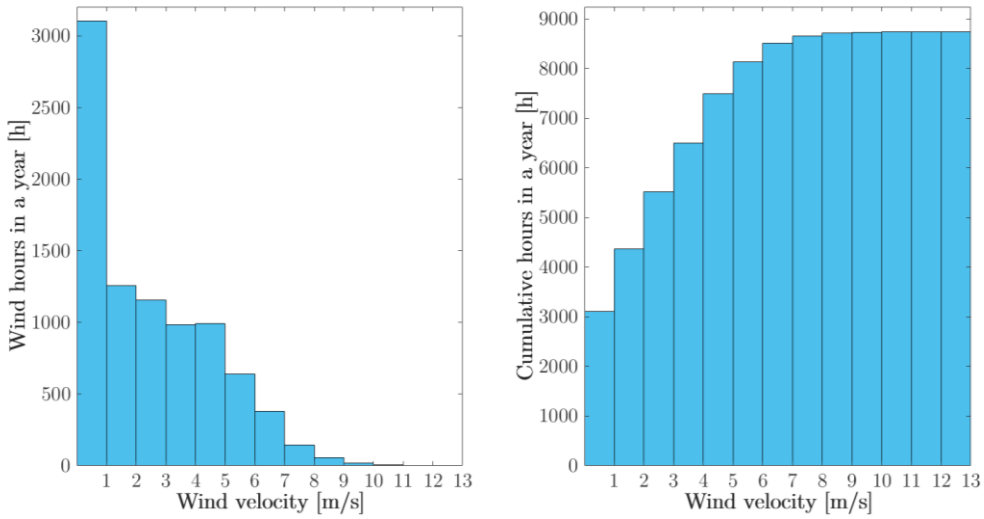


Figure 2.4 Wind hours and cumulative hours for ETSIA

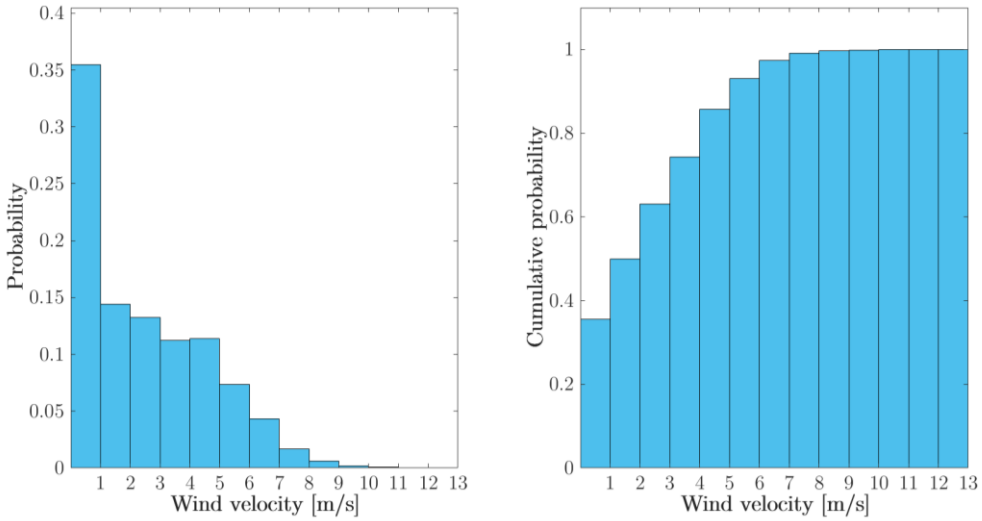


Figure 2.5 Wind probability and cumulative probability for ETSIA

The diagram above shows a very high probability of calms in this location, specially between 0 and 1 m/s. In wind turbines, when the wind is below a certain velocity called cut-in velocity the rotor does not rotate and power is not produced. This velocity is usually between 2 and 4 m/s. Taking 3 m/s as cut-in velocity reference, a wind turbine in this emplacement would be off **63.1 %** of the year, and the AEP would be very low.

The Weibull parameters of the PDF are then calculated in order to fit both the discrete probability density and the cumulative distribution:

$$c = 2.47$$

$$k = 0.98$$

The average wind velocity is **2.47 m/s**.

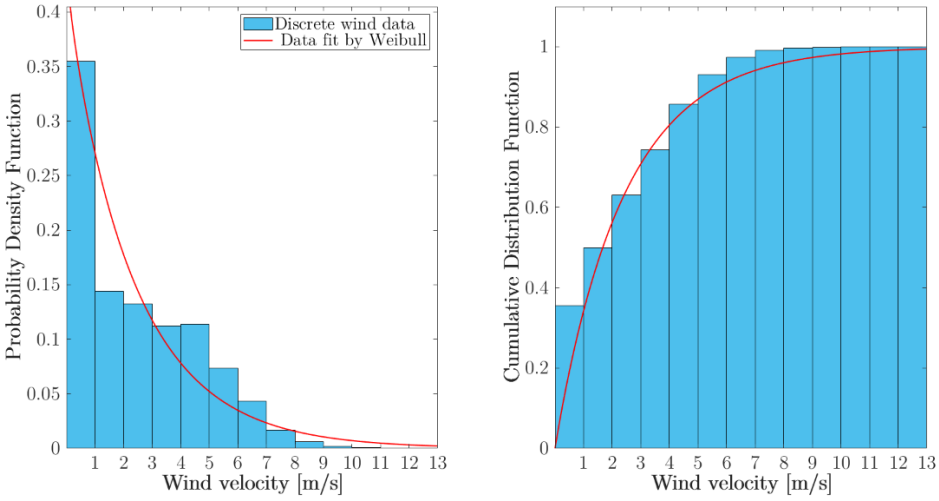


Figure 2.6 Probability Density and Cumulative Distribution of the wind

The shape parameter k is lower than 1, so the curve tends to ∞ when wind velocity approaches 0. The fit of the data by the Weibull PDF is not good at all because of the very high probability density of the first bin winds. This curve shape is never desirable for wind turbine sites.

Multiplying the Weibull PDF with the available power in wind per surface unit we find the energy probability density curve (per surface unit):

$$\frac{P_u}{A} = \frac{1}{2}\rho u^3 f(u) \tag{2.17}$$

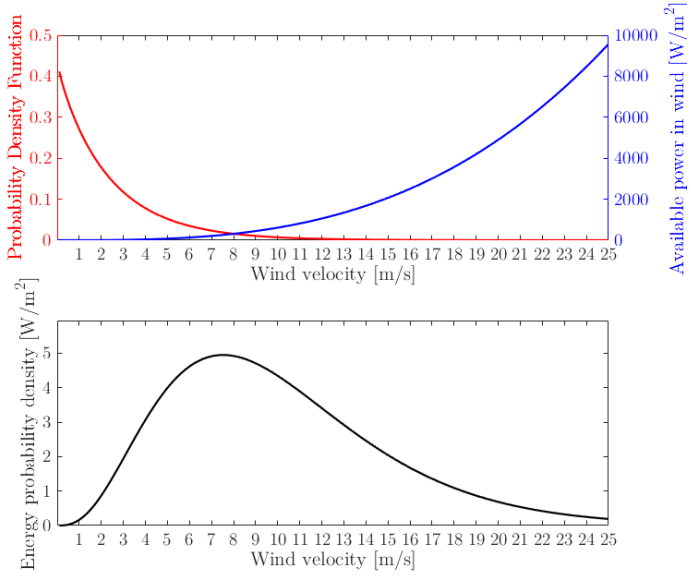


Figure 2.7 Energy probability density for ETSIA

The velocity that contains more energy is u_{me}= 7.7 m/s.

It is not easy to stablish which is the limit above which installing a wind turbine is economically justified. For a small scale wind turbine a value of **4.5 m/s** of average wind velocity and **100 W/m²** of wind power density (WPD) could be taken as a lower limit reference [18].

The wind resource of a site can be described roughly by its average wind speed, but the WPD gives a much more accurate idea of the quality of the location [19]. The WPD can be defined using the PDF, $f(u)$:

$$WPD = \frac{1}{2}\rho \int_0^\infty u^3 f(u)\, du \qquad \left[\frac{W}{m^2}\right] \tag{2.18}$$

Graphically the *WPD* is calculated as the area under the curve of the energy probability density described in Equation (2.17). It can be easily calculated using a numerical integration tool such as the trapezoidal rule (*trapz ()* in MATLAB®).

WPD for this location is equal to 58.9 W/m², and the site has **no wind potential**.

2.3.2 Aguilar de Codes

Aguilar de Codes is a village in the central west part of Navarre. It is located in the region of Estella. The Weibull curves of the PDF in the work in [13] show an acceptable shape and the WPD estimated in [17] is of 105 W/m². The study of the wind resource is analogous to the previous one.

The frequency and cumulative diagrams are:

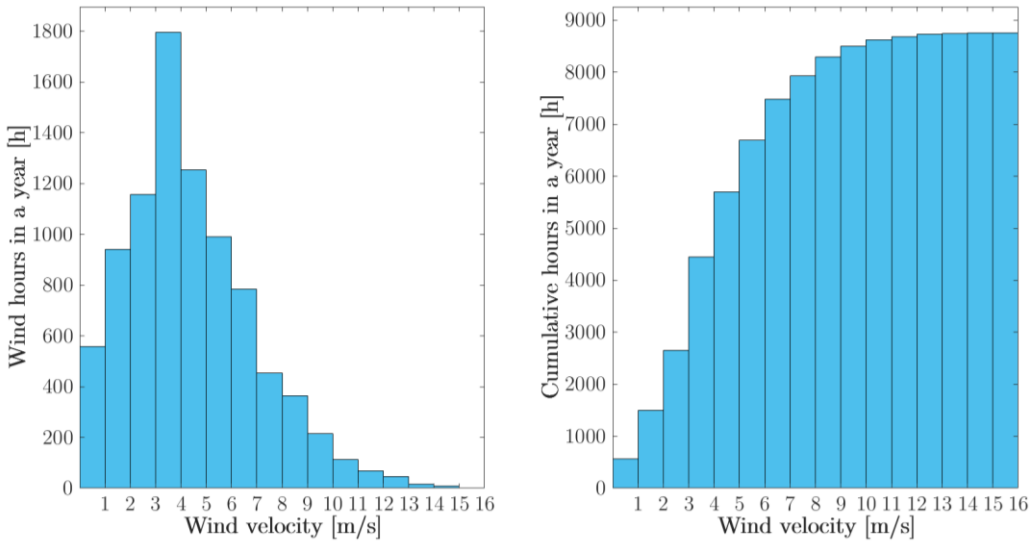


Figure 2.8 Wind hours and cumulative hours for Aguilar de Codes

In this case, the wind distribution is different. The predominant wind velocity is between 3 and 4 m/s. The average wind velocity is **4.37 m/s**.

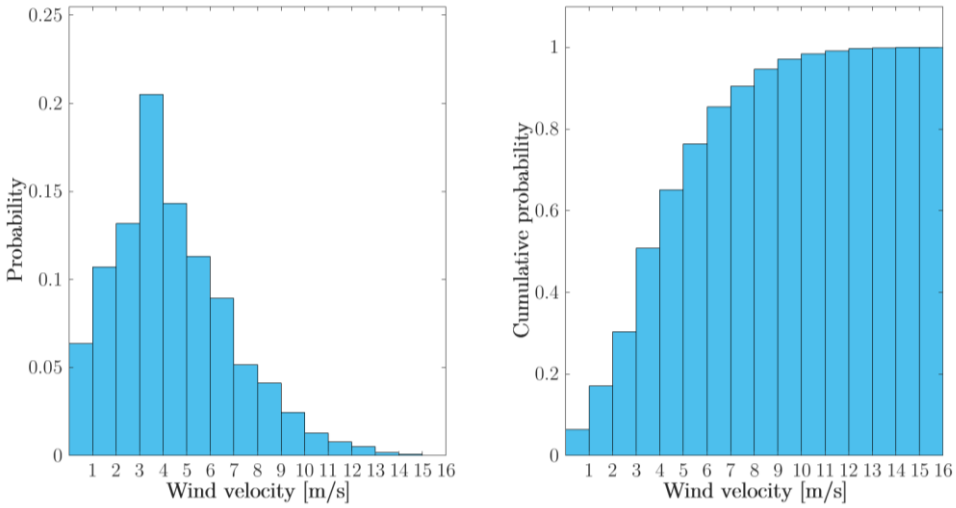


Figure 2.9 Wind probability and cumulative probability for Aguilar de Codes

The Weibull parameters of the PDF are:

$$c = 4.9$$

$$k = 1.78$$

The Weibull fit of the probability density and the cumulative distribution of wind is shown below.

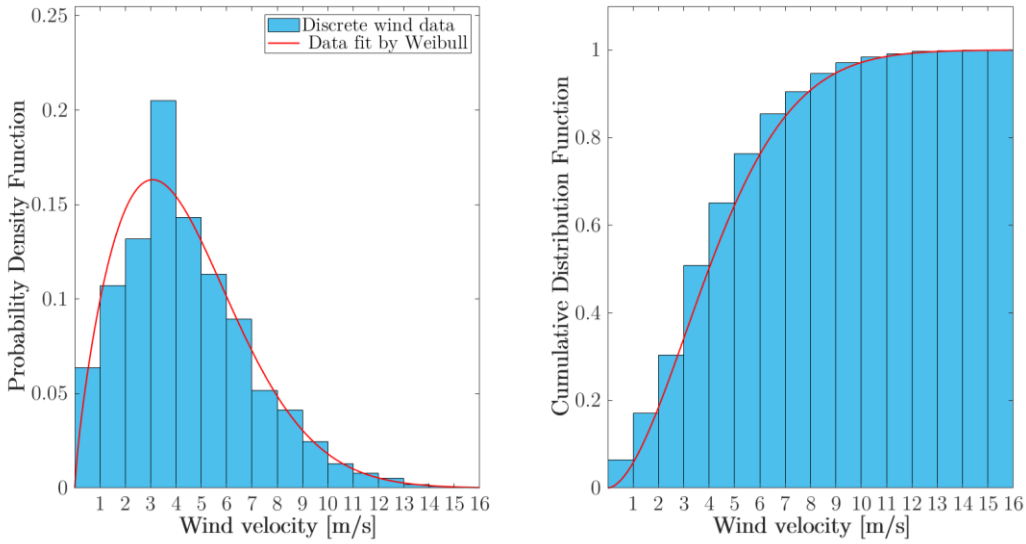


Figure 2.10 Probability Density and Cumulative Distribution of the wind

The fit of the curves is quite accurate, and only differs from the discrete probability density the value in the value of the 3rd and 4th bin. The fit of the cumulative distribution function is almost perfect.

The energy probability density is depicted combining the PDF and the power available in wind.

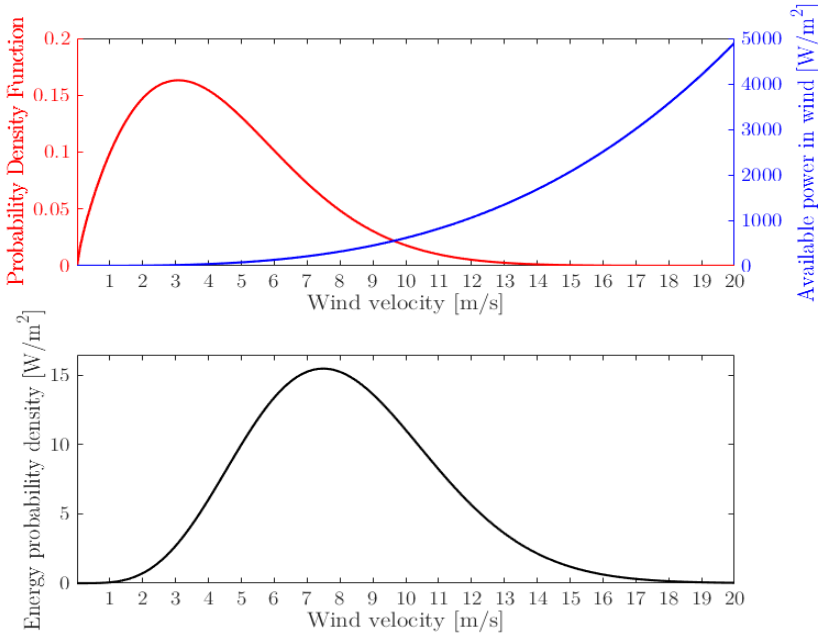


Figure 2.11 Energy probability density for ETSIA

The velocity that contains more energy is u_{me}= 7.5 m/s. A customized design would include this velocity in its best performance wind velocity region.

The WPD is given by the area under the curve above and is **110.2 W/m²**. As WDP is higher than 100 W/m² we could consider installing a wind turbine in this location. However, the wind resource is considered moderate for this value of WPD.

The characteristics of the meteorological stations, as well as the wind parameters of both studied emplacements are shown in the table below.

Table 2.1 *Meteorological data for both locations*

	Location	
METEOROLOGICAL STATION	**UPNA (ETSIA)**	**Aguilar de Codes**
Altitude [m]	433	736
Reference height [m]	10	10
Data period	11/2017 to 11/2018	11/2017 to 11/2018
WIND CHARACTERISTICS (h=25 m)		
Average velocity [m/s]	2.49	4.37
Standard deviation [m/s]	2.16	2.51
Shape parameter, k	0.98	1.78
Scale parameter, c	2.47	4.9
% time OFF (u<3m/s)	63.1	30.3
Max. Energy velocity u_{me}	7.7	7.5
WPD [W/m^2]	58.9	110.2
Wind resource potential	Very poor	Moderate

Chapter 3

AIRFOIL CHARACTERISTICS AND SELECTION

In this chapter, the most important element of the wind turbine blade will be studied: the **airfoil**. The airfoil represents the geometric shape of the blade cross-section.

The shape of the airfoil (or airfoils) in a blade has a very high impact, not merely in the overall wind turbine performance, but also in the aerodynamic loads acting on the blade. Thus, for each wind turbine blade the constituting airfoils have to be carefully selected.

In the following sections the basic terminology of airfoils will be reviewed and then the different airfoils used in this project will be discussed. Airfoil characteristics (lift and drag) will be determined in function of the angle of attack and Re number using the computational tool XFOIL implemented in QBlade.

Finally, a 3D correction due to the rotational nature of the flow in wind turbines will be applied to the airfoil data. Also, due to the lack of data over the 360º angle of attack a numeric extrapolation model will be implemented. These corrections of the XFOIL data will make the computational implementation of the Blade Element Momentum Theory (BEMT) reliable and accurate.

3.1 Preliminary concepts of airfoils

3.1.1 Airfoil geometry

Different terms are used in order to characterise the airfoil geometry as it is shown in Figure 3.1. The airfoil geometry has a high impact on the aerodynamic performance of the blade.

The mean camber line defines the halfway between upper and lower surfaces. The most forward and backward points of this line are on the leading and trailing edges, respectively. The straight line that connects both edges is the called the chord line of the airfoil, and the distance from the leading to the trailing edge, measured along the chord line, is the chord of the airfoil.

The camber of the airfoil is the distance between the mean camber line and the chord line, perpendicular to the chord line direction. The thickness of the airfoil is the distance between the upper and lower surfaces, also measured in a direction perpendicular to the chord line. Both the camber and the airfoil thickness are usually expressed as a percentage of the blade chord.

Finally, the angle of attack α defines the angle between the chord line of the airfoil and the incoming relative wind direction. The length of the airfoil in a direction perpendicular to the cross-section is called the airfoil span.

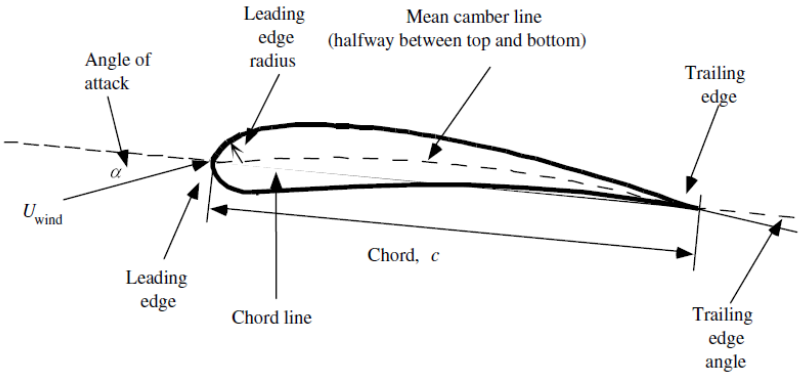


Figure 3.1 Airfoil nomenclature [20]

3.1.2 Lift, drag and non-dimensional parameters

The airflow passing through an airfoil produces a pressure difference between upper and lower surface. The flow velocity over the airfoil increases over the convex surface of the airfoil resulting in a lower average pressure on the 'suction' side compared to the concave 'pressure' side of the airfoil. On the other hand, viscous forces on the surface slow down the fluid.

These pressure and friction forces are usually resolved into two forces (Lift and Drag) and a pitching moment that acts at a distance of c/4 from the leading edge. The lift force, L, acts in a perpendicular direction to incoming wind, drag force, D, acts in parallel direction to the wind and pitching moment, M, is defined around and axis perpendicular to the airfoil cross-section (see Figure 3.2).

The 2 dimensional lift, drag and pitching moment coefficients are defined as:

$$C_l = \frac{L/l}{\frac{1}{2}\rho U^2 c} \tag{3.1}$$

$$C_d = \frac{D/l}{\frac{1}{2}\rho U^2 c} \tag{3.2}$$

$$C_m = \frac{M}{\frac{1}{2}\rho U^2 A c} \tag{3.3}$$

Where ρ is the density, l the airfoil span, c the chord and A the projected area (chord x span). The pitching moment is related to the structural aspect of the blade and its relevance in power generation is neglected in steady state BEMT codes.

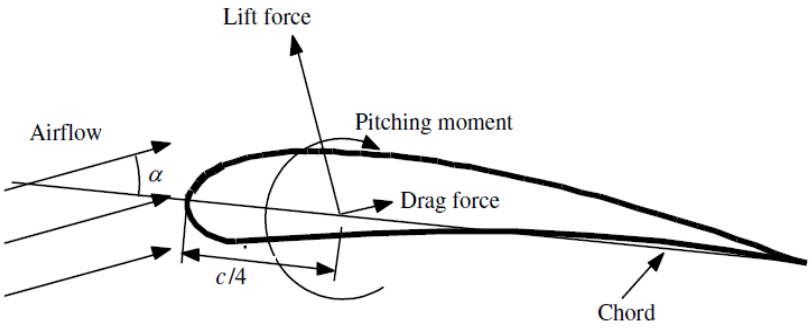


Figure 3.2 Lift and drag forces and pitching moment on airfoil [20]

These coefficients are usually tabulated and can be found using experimental analysis, CFD simulations or other computational methods such us *XFOIL*, which has been used in this work.

Lift, drag and pitching moment coefficients are functions of the angle of attack α, the Reynolds number and the Match number, which has no important relevance in wind turbine application.

The chordal Reynolds number is defined as follows:

$$Re = \frac{W\rho c}{\mu} = \frac{Wc}{\nu} \tag{3.4}$$

Where W is the relative incoming wind velocity, ρ is the air density, c is the chord of the airfoil and μ is the dynamic viscosity [Pa · s]. The expression can also be written in terms of the kinematic viscosity ν [$m^2 \cdot s^{-1}$].

3.1.3 Airfoil behaviour and stall phenomenon

Ideally lift coefficient increases linearly with the angle of attack α by:

$$C_l = 2\pi(\alpha - \alpha_0) \tag{3.5}$$

Where α_0 is the zero lift angle of attack (0 for symmetric airfoils).

However, as it can be seen in Figure 3.3, when a certain angle of attack is reached, the lift coefficient separates and no longer follows the ideal straight line. If α continues increasing, a drop in lift coefficient with a simultaneous rise in drag coefficient happens. This phenomenon is called **stall** and acts in a very geometrically dependent manner. As it will be seen in Chapter 5, this phenomenon is commonly used in real wind turbines to regulate and limit the power generation both actively or passively (as it is done in this work).

The transition point between the ideal airfoil behaviour and the real one, and, in general, the lift and drag coefficients for a specific airfoil depend on the Re number. This means that rotor designers must make sure that appropriate Reynolds number data is available for an accurate analysis of the wind turbine performance.

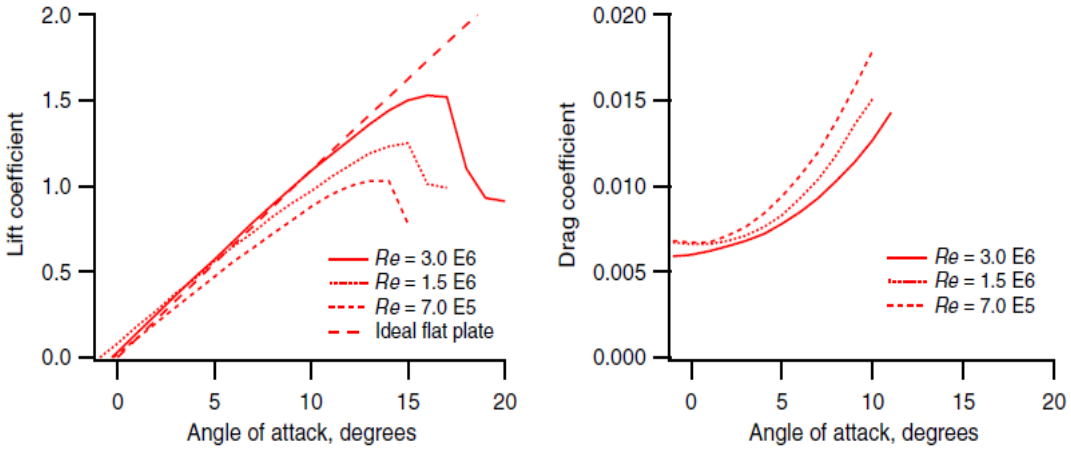


Figure 3.3 Lift and drag coefficients for NACA 0012 for several Re numbers [20]

The stall phenomenon occurs due to 2 reasons: flow separation and flow reversal. This separation can occur both on the leading or on the trailing edge.

The first one is considered a laminar separation. When the angle of attack is quite high, the leading edge curvature is high and kinetic energy is lost by shear. It arrives and standstill point where flow detaches. A suction pressure downstream causes the reversal of the flow and the formation of bubbles. This stall type happens when Re numbers are low, usually in rotors below 5m diameter. It is less common in cambered airfoils, which are typical in wind turbine blade design.

The trailing edge separation starts at the trailing edge and moves gradually along the upper surface while angle of attack increases. It is more common in rounded nose cambered airfoils.

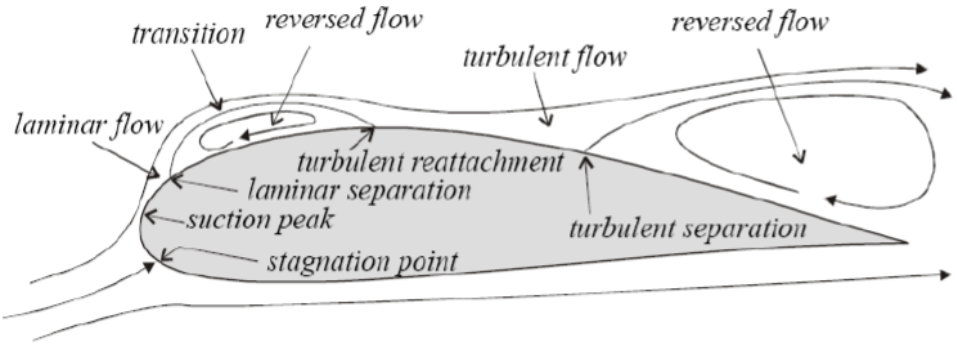


Figure 3.4 Qualitative representation of flow separation types [4]

3.2 Airfoil selection: Low Reynolds number airfoils

A difficult task when it comes to the design of a wind turbine blade is the selection of a proper airfoil layout. Depending on the geometry some airfoils will behave better for large scale wind turbines and others will work better for small scale ones. It is also very common to find different airfoils in the same blade span, using a certain type for the root region, another in the middle section and a different one near the tip. However, in small scale wind turbines a single airfoil solution is usually taken [21]–[24] and so will be done in this work.

Besides, the majority of the small scale wind turbines are regulated by passive stall. This means that the power limitation above rated relies on the aerodynamic design of the blade and therefore in the capacity of the airfoil to stall.

The first step followed in order to find proper airfoils for the wind turbine design has been checking the literature. This has helped to identify some low Re number airfoils that are, or have been used in the current research and manufacture of real small scale wind turbines. In order to choose the proper airfoil, it is desired that it has a good glide-ratio (C_l/C_d) for low values of the Re number [25, p. 378]. A particularity of low Re number airfoils is the high dependence of the airfoil characteristics with Re numbers.

Once the proper airfoils have been identified, their aerodynamic characteristics need to be determined. Chord-scaled airfoil coordinates are available in several internet websites (*AirfoilTools*, for example [26]). Then, airfoil coordinates have been loaded in the software *QBlade*, which has internally implemented an XFOIL analysis module.

XFOIL simulations have been performed in batch for a Re number range from 100000 to 1 million, with a ΔRe of 100000. Note that there is no exact way of knowing which will be the Re number range in which blade sections will work within all the operative velocity range. Practically, the BEMT design algorithm was run for a unique Re number of 400000 and then, after discovering the values of Re numbers in the designed blade stations within the operation range the blade design has been refined to a multi Re analysis.

XFOIL data is usually only available within a limited range of angles of attack α. In this work, aerodynamic characteristics analysis was performed in an angle of attack range of [-25,25] ° since above 25° convergence is difficult to achieve. The value of $\Delta\alpha$ used was 0.25°.

In this work, different blade designs will be tested using **4 different airfoils**.

The airfoil creator, uses, geometrical shape, geometrical parameters and aerodynamic characteristics given by the simulations are depicted in the next sub-section.

3.2.1 S822 airfoil (NREL)

S822 airfoil, as well as S823, was developed by NREL between 1992 and 1993 and they form part of a family of thick airfoils (S809 included) designed for 3 to 10-meter diameter, stall-regulated, horizontal axis wind turbines [27].

This airfoil has been used in several research projects for small scale horizontal axis wind turbine blade design due to its high glide-ratio [22], [28].

The geometrical shape and parameters of the airfoil are shown in Figure 3.5 and Table 3.1 respectively.

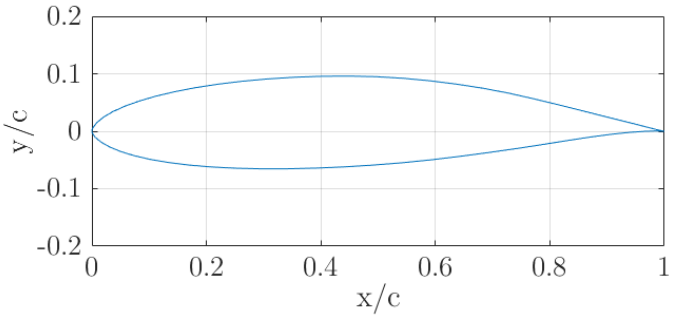


Figure 3.5 S822 airfoil shape

As it can be seen, the S822 airfoil has a very aerodynamical shape, with very high curvature angles, which is the reason of having so high values of glide-ratio.

Table 3.1 Geometrical parameters of S822 airfoil

	Airfoil
PROPERTY	S822 (NREL)
Max. Thickness (chord %)	16
Max. Thickness position (chord %)	39.2
Max. Camber (chord %)	1.9
Max. Camber position (chord %)	59.2
Number of geometric points	66

Aerodynamic coefficients of the airfoil for different Re numbers are depicted in Figure 3.6. 4 graphs have been created: Cl vs alpha, Cd vs alpha, Cl/Cd vs alpha and Cl vs Cd. These four graphs are usually the most characteristic ones.

It should be noticed how the maximum glide-ratio (Cl/Cd) angle of attack varies with Re numbers in a manner that when Re number increases, the angle of attack for maximum Cl/Cd decreases. Also, when Re number increases polar curves are more confined, closer one to each other, whereas for lower Re numbers they appear to be further from each other. These facts are important when it comes to small scale wind turbine blade design.

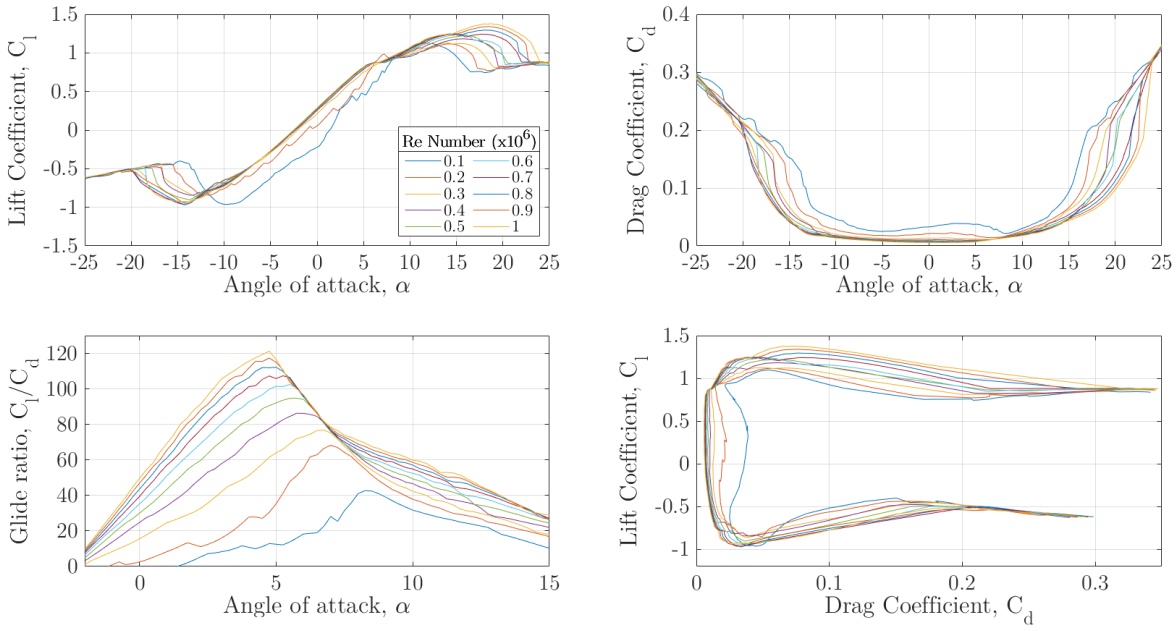


Figure 3.6 Aerodynamic coefficients for diferent Re numbers, S822 airfoil

3.2.2 S823 airfoil (NREL)

The S823 airfoil was also developed by NREL and has been used for the design of small scale wind turbine blades [28].

The geometrical shape and parameters of the S823 airfoil are shown in Figure 3.7 and Table 3.2 respectively.

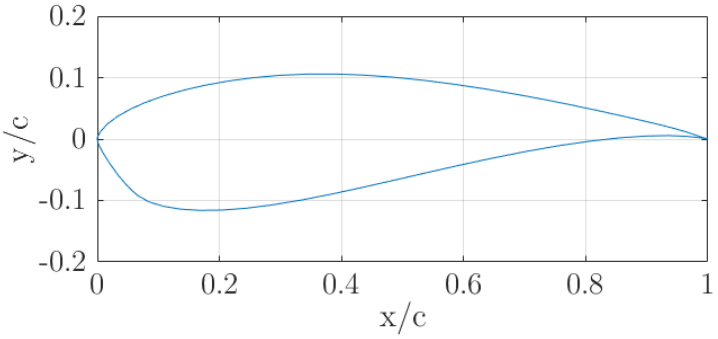


Figure 3.7 S823 airfoil shape

This is a thicker airfoil, with the maximum thickness point closer to the leading edge than in the previous airfoil. This gives to the airfoil better structural characteristics and is usually a practical solution for the root region of the blade when loads are high and more inertia moment is required.

However, as it can be seen in Figure 3.8, generally lower glide-ratios than in the previous case are achieved, even if they are still quite acceptable.

Table 3.2 Geometrical parameters of S823 airfoil

	Airfoil
PROPERTY	S823 (NREL)
Max. Thickness (chord %)	21.18
Max. Thickness position (chord %)	25.5
Max. Camber (chord %)	2.5
Max. Camber position (chord %)	70.5
Number of geometric points	66

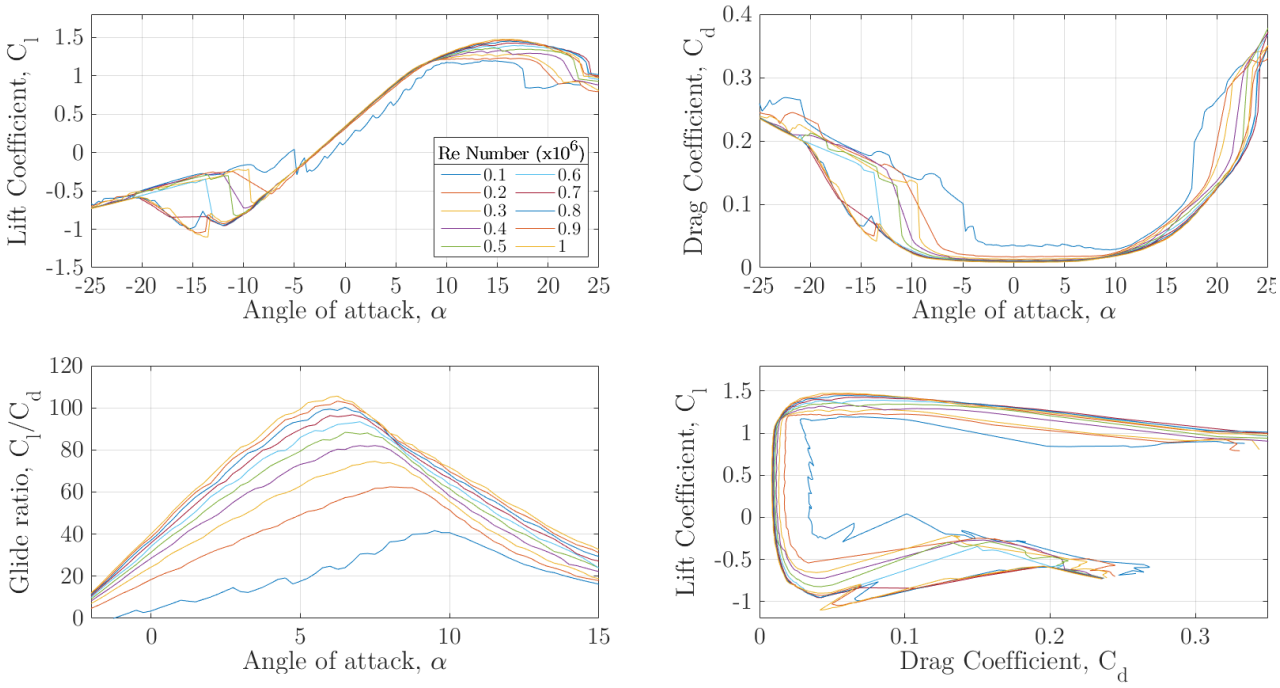


Figure 3.8 Aerodynamic coefficients for diferent Re numbers, S823 airfoil

It can be noted that for Re=100 000 simulated curves present subtle oscilations and differ from the others in the ideal zone of the Cl vs alpha graph.

3.2.3 S809 airfoil (NREL)

The S809 airfoil, also created by NREL, is a very tested airfoil which was used for the blade layout of NREL Phase VI experimental wind turbine (20 kW, stall-regulated) [29], [30].

The geometrical shape and parameters of the S809 airfoil are shown in Figure 3.9 and Table 3.3 respectively. This airfoil presents a maximum thickness (21%) similar to the one of S823, but it appears further from the leading edge.

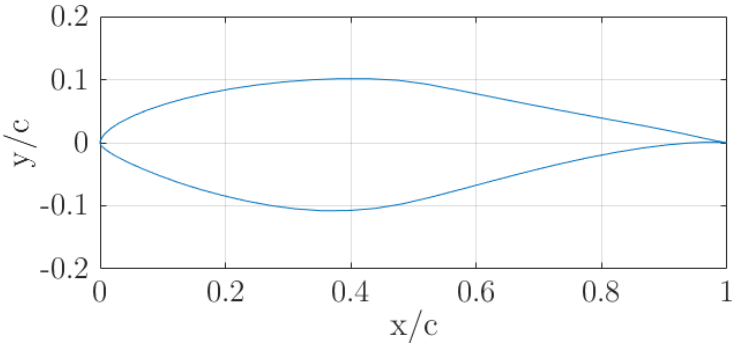


Figure 3.9 S809 airfoil shape

Table 3.3 Geometrical parameters of S809 airfoil

	Airfoil
PROPERTY	S809 (NREL)
Max. Thickness (chord %)	21
Max. Thickness position (chord %)	38.3
Max. Camber (chord %)	1
Max. Camber position (chord %)	82.3
Number of geometric points	66

Aerodynamic coefficients for the several Re numbers studied are depited in Figure 3.10. The Cl/Cd vs alpha graph shows a lower aerodynamic performance of the airfoil than in the previous airfoils, since maximum glide-ratios are lower. Thus, a lower power coefficient is expected to be achieved at the design point.

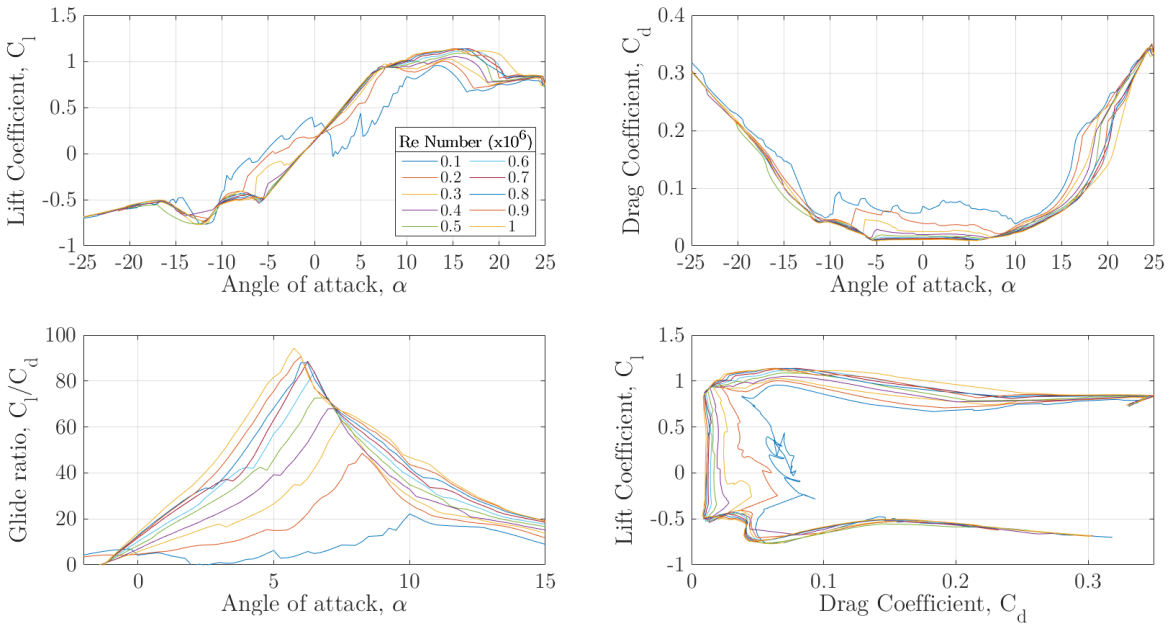


Figure 3.10 Aerodynamic coefficients for diferent Re numbers, S809 airfoil

3.2.4 SG 6040 airfoil

The SG 60-XX airfoil family was designed and tested by Selig and Giguère in 1998 [31]. These airfoils present a very high aerodynamic performance for low Re numbers. The SG 6040 airfoil is commonly used for low power rating wind turbine blade design due to its high glide-ratio and acceptable thickness [24].

The geometrical shape and parameters of the SG 6040 airfoil are shown in Figure 3.11 and Table 3.4 respectively.

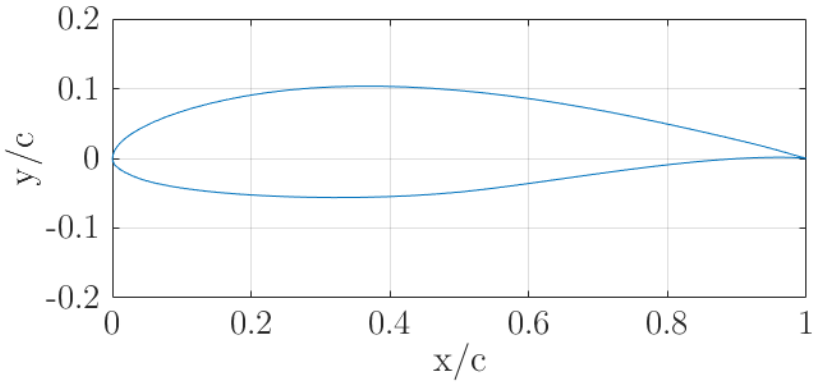


Figure 3.11 SG 6040 airfoil shape

Table 3.4 Geometrical parameters of SG 6040 airfoil

	Airfoil
PROPERTY	**SG 6040**
Max. Thickness (chord %)	16
Max. Thickness position (chord %)	35.3
Max. Camber (chord %)	2.5
Max. Camber position (chord %)	60.4
Number of geometric points	81

Note that this airfoil is geometrically very similar to S822: they have the same maximum thickness, it appears more or less at the same point of the airfoil and they present very high curvature radius on their geometry. SG 6040 has a slightly higher camber, but both are expected to give a high aerodynamic performance on the design point.

Figure 3.12 shows the aerodynamic coefficients of the airfoil. As in the previous cases, the curves for Re number of 100 000 show much lower performance and more oscillation. Aerodynamic performance is usually higher for higher Re numbers. Besides, curves are more confined for higher Re number range and Re dependence becomes lower, as it happens in large scale wind turbines.

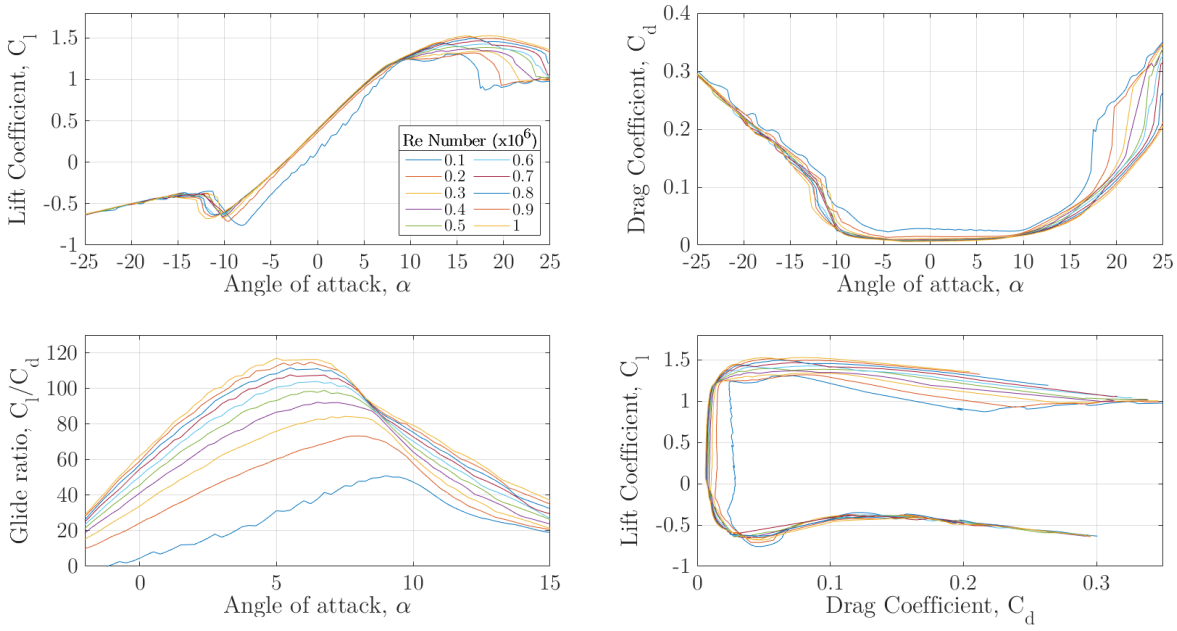


Figure 3.12 Aerodynamic coefficients for diferent Re numbers, SG 6040 airfoil

3.3 Airfoil data 3D corrections

In the middle of the 20th century it was observed in aircraft propellers that lift coefficients were increased due to the rotational effects that imply radial flows over the blade span. It was also observed that the stall was occurring at higher angles of attack, a phenomenon known as **"Stall Delay".**

It was seen that laminar flow was hardly influenced by these rotational effects. However, when a certain angle of attack is reached the flow detaches from the upper surface of the airfoil and the stall of the blade begins (as explained in the previous section). Under this circumstance the fluid is moved outwards by the centrifugal force. This effect is called "centrifugal pumping".

The direct consequence of this phenomenon is an **increase of lift** in the inner part of the blade. The quantification of this increase is difficult and highly geometry dependant. Regarding the drag coefficient, there is not great consensus among researchers and while some say that there is no significant effect on drag, others claim that a slight decrease in drag should be taken into account.

The physical understanding of this phenomenon is the following: The centrifugal loads act on the separated flow, which is accelerated towards the blade tip. This radial velocity induces a **Coriolis acceleration** towards the trailing edge, acting as a favourable pressure gradient that gives stability to the boundary layer and moves it outboard to the trailing edge [32]. Consequently, the lift coefficient increases.

Therefore, the 2D airfoil data measured from wind tunnel, by CFD simulations or panel methods like XFOIL, are usually corrected in order to use in BEMT or aeroelastic codes.

Actually there are several 3D correction models available in the literature. The most known models are the ones developed by Snel *et al.* [33], Lindenburg [32] and Du and Selig [1]. All the correction models are expressed as:

$$C_{l,3D} = C_{l,2D} + f_{C_l}(\frac{c}{r}, \dots)\Delta C_l \tag{3.6}$$

$$C_{d,3D} = C_{d,2D} + f_{C_d}(\frac{c}{r}, \dots)\Delta C_d \tag{3.7}$$

Where the subscript 2D refers to measurements from 2D wind tunnels or 2D simulations, $\frac{c}{r}$ is the ratio between the blade chord length and the radial position of the blade element and $f(\frac{c}{r}, \dots)$ means that all the proposed models are function of $\frac{c}{r}$ but also that they can be a function of other parameters. ΔC_l and ΔC_d are the differences between the C_l and C_d that the airfoil section would experience if the flow did not separate, this is, the lift and drag coefficients of inviscid flow, and their steady 2D value, where obviously separation will occur.

The functions $f(\frac{c}{r}, \dots)$ vary from one model to the other, being for some authors equal to 0 the correction factor of the drag coefficient. The proposed values for these coefficients are depicted in the figure below:

Table 3.5 Correction of airfoil coefficients for 3D effects

MODEL	f_{C_l}	f_{C_d}
Snel *et al.* [33]	$3\left(\frac{c}{r}\right)^2$	0
Lindenburg [32]	$3.1\left(\frac{\omega r}{W}\frac{c}{r}\right)^2$	0
Du and Selig [1]	$\frac{1}{2\pi}\left[\frac{1.6(\frac{c}{r})}{0.1267}\frac{a-(\frac{c}{r})^{\frac{d}{\Lambda}\frac{R}{r}}}{b+(\frac{c}{r})^{\frac{d}{\Lambda}\frac{R}{r}}}-1\right]$	$\frac{1}{2\pi}\left[\frac{1.6(\frac{c}{r})}{0.1267}\frac{a-(\frac{c}{r})^{\frac{d}{2\Lambda}\frac{R}{r}}}{b+(\frac{c}{r})^{\frac{d}{2\Lambda}\frac{R}{r}}}-1\right]$
	with $\Lambda = \frac{\omega R}{\sqrt{{U_\infty}^2+(\omega r)^2}}$ and $a = b = d = 1$	

In this work the model of **Snel *et al.*** will be implemented due to several reasons:

1. Accuracy in small scale wind turbine modelling:

Authors in [34] proposed a new 3D correction model and also compared the existing ones with measurements in existing wind turbines. This work showed how for the *NREL/NASA Ames* rotor (10 m diameter, stall-regulated wind turbine with S809 airfoil layout) results from Snel *et ali.*'s model and their own model where the closest ones from the real performance of the wind turbine.

2. Simplicity:

The National Wind Technology Center (NWTC) recommends the generation of the corrected and extrapolated airfoil aerodynamic characteristics as a pre-process to BEMT codes and so they do for their code software [35]. Lindenburg and Du and Selig corrections require the velocities acting on the blade sections, which vary within the inflow wind velocity range. This would lead to the generation of a very high amount of airfoil polars in order to study the performance over a wind velocity range, or alternatively, to perform a sensibility study of the model towards relative velocity in the sections. However, this is avoided in Snel *el al.*'s model due to its mathematical simplicity: f_{C_l} only depends on the chord and radial position of the blade element.

MODEL IMPLEMENTATION IN MATLAB®

In this work an own MATLAB® code has been written, which can be found in APPENDIX A (A.2 Snel_3D_corrections.m). In this script, 2D airfoil characteristics generated with XFOIL and the blade design (chords and radial positions) are loaded and lift coefficient is corrected as follows:

1. For each Re number the inviscid lift line is drawn as a straight line.
2. For each blade element node along blade span the ratio $\frac{c}{r}$ is calculated.
3. Once $\frac{c}{r}$ is calculated, lift coefficient is corrected when 2D data detaches from the straight inviscid line using Equation (5.1). Data for negative angles of attack are not corrected due to the uncertainty of the model's accuracy in the negative angle of attack stall condition (which is not experienced during normal operation of a wind turbine). Drag coefficient is not corrected.
4. Corrected data is then saved into an EXCEL file for each airfoil. Inside each EXCEL file, data for each element node is written in a different sheet. In each sheet, data is stored in 3-column blocks (α, C_l, C_d), one for each Re number.

An example of the corrected lift coefficients is depicted in Figure 3.12. The figure shows very high values in the blade element nodes that are closer to the blade root. However, when going outboard in the blade span the values start to confine until near the blade tip, where they are very similar to the 2D XFOIL values.

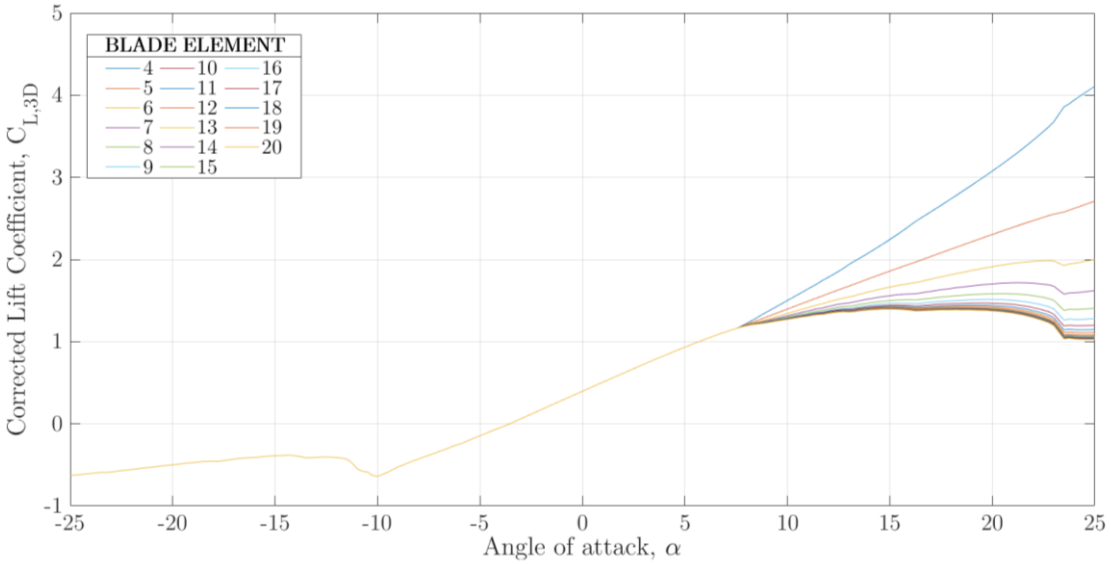


Figure 3.13 3D correction model results' example (SG 6040, Re=$5 \cdot 10^5$)

A more detailed explanation of the code can be found in the MATLAB file comments.

3.4 Polar data 360 ° extrapolation.

Stall regulated wind turbine blade sections experience higher angles of attack when incoming wind velocity increases. Unfortunately, data for large angle of attacks are usually not available from wind tunnel or 2D simulations and therefore extrapolation techniques are often used in order to obtain the wind turbine performance over a wind velocity range. Beyond the stall or the last data point available, **Viterna-Corrigan extrapolation method** [36] is commonly used.

In this work Viterna-Corrigan extrapolation method is used to predict the lift and drag coefficients from the last data point to 90 ° and from the first data point to -90 ° (negative angle of attack side). The extrapolated values of lift and drag are given by the following expressions:

$$C_l = \frac{C_{d_{max}}}{2} \sin 2\alpha + A_2 \frac{\cos^2 \alpha}{\sin \alpha} \tag{3.8}$$

where

$$A_2 = (C_{l_s} - C_{d_{max}} \sin \alpha_s \cos \alpha_s) \frac{\sin \alpha_s}{\cos^2 \alpha_s} \tag{3.9}$$

and

$$C_d = C_{d_{max}} \sin^2 \alpha + B_2 \cos \alpha \tag{3.10}$$

where

$$B_2 = \frac{C_{d_s} - C_{d_{max}} \sin^2 \alpha_s}{\cos \alpha_s} \tag{3.11}$$

The subscript s defines the stall point or alternatively the last data point available, called the matching point.

The value of $C_{d_{max}}$ is defined as:

$$C_{d_{max}} = 1.11 + 0.018AR \tag{3.12}$$

Where AR is the aspect ratio of the blade, given by:

$$AR = \frac{b}{SMC} = \frac{b}{\frac{S_{projected}}{b}} = \frac{b^2}{S_{projected}} \tag{3.13}$$

Where b is the blade span, SMC the standard mean chord and $S_{projected}$ the projected area of the wind turbine blade.

FLAT PLATE THEORY

For high incidence angles flow separates and behaves in a non-geometric dependant way, like a flat plate. In this work flat plate theory has been assumed for $\alpha > 90°$ and $\alpha < -90°$ instead of reflecting Viterna-Corrigan curves, which would lead to a discontinuity at $\alpha = 180°$ for non-symmetric foils. Besides, the results are close for lift coefficient extrapolation and nearly identical for the drag coefficient [37]. The curves of lift and drag coefficients are given the flat plate theory expressions:

$$C_l = 2C_{l_{max}} \sin \alpha \cos \alpha \tag{3.14}$$

and

$$C_d = C_{d_{max}} \sin^2 \alpha \tag{3.15}$$

For $C_{l_{max}}$, the 70% of the value of the lift coefficient at 45° is usually taken. The 70% is applied in order to consider the asymmetry between the leading and the trailing edge.

It is important to note that wind turbines will hardly work under this angle of attack range. However, full extrapolation is usually required in BEMT and aeroelastic codes in order to obtain convergence and avoid computational problems.

MODEL IMPLEMENTATION IN MATLAB®

In the present work an own MATLAB® code has been written, which can be found in the APPENDIX A (1A.3 Viterna_360_extrapolation.m). In this script, the already corrected airfoil characteristics are extrapolated over 360°. The extrapolated data was computed with a $\Delta\alpha = \mathbf{1°}$. The script also allows to extrapolate the non-corrected 2D airfoil data in order to perform simulations without takin into account the rotational

effects of the flow. This will be useful in order to compare the impact of the rotational effects in the generated power of the wind turbine.

The airfoil polar extrapolations have been done element by element of the blade span (due to the different coefficients along blade span) and within the range of Re numbers defined previously (from 100 000 to 1 million). In order to clarify the extrapolation process, 5 different regions where defined:

- REGION I: The region goes from [-25, 25] °. Data in this range is already known, and the is no need of extrapolation.
- REGION II: This region goes from [25, 90] °. Extrapolation is done using the Viterna-Corrigan model, with a matching point angle of attack of $\alpha_s = 25°$.
- REGION III: This region goes from [-90, -25] °. Extrapolation is done using the Viterna-Corrigan model, with a matching point angle of attack of $\alpha_s = -25°$.
- REGIONS IV and V: These regions are extended from [90, 180] º and [-180, -90] ° respectively. In these regions the flat plate theory equations are used.

An example of the polar extrapolation over the 360° is depicted below. The 360° are divided into the previously explained regions:

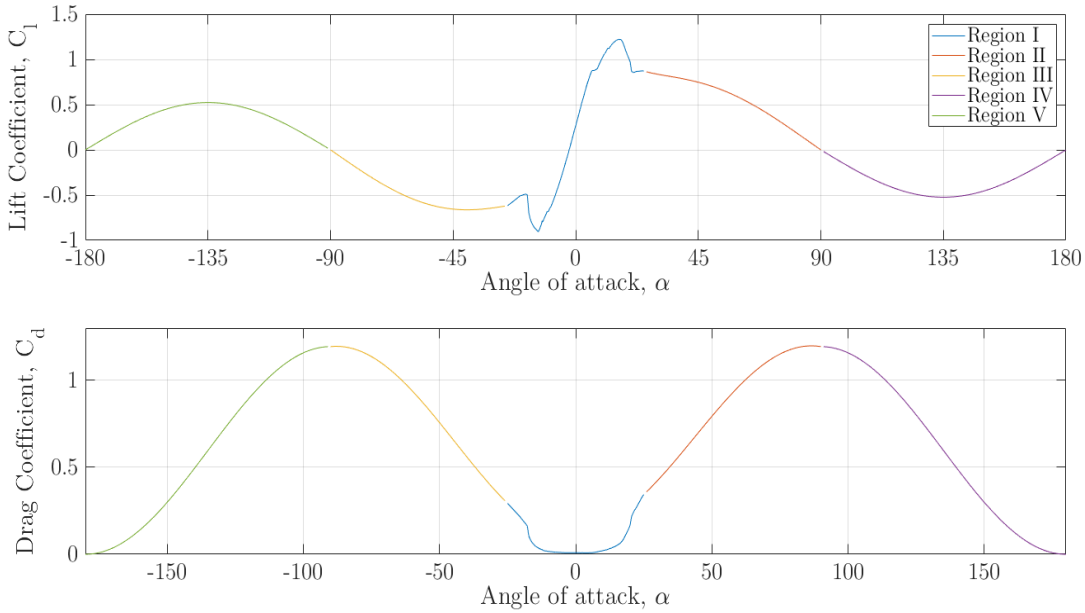
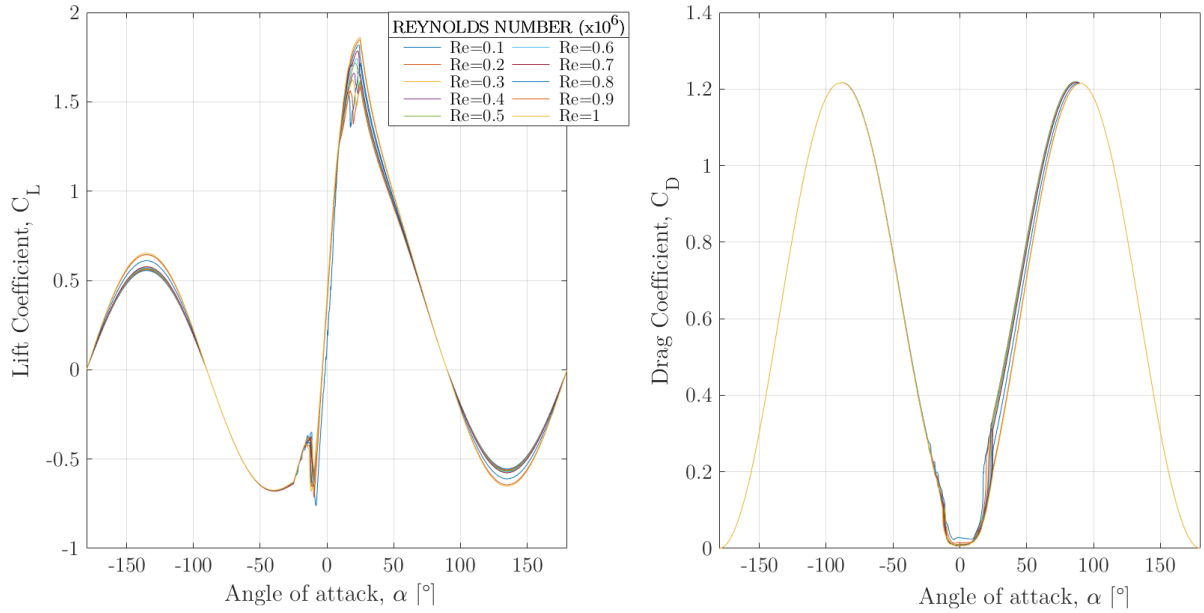


Figure 3.14 Airfoil coefficient 360° extrapolations (S822 airfoil, Re=5.10^5, 2D flow)

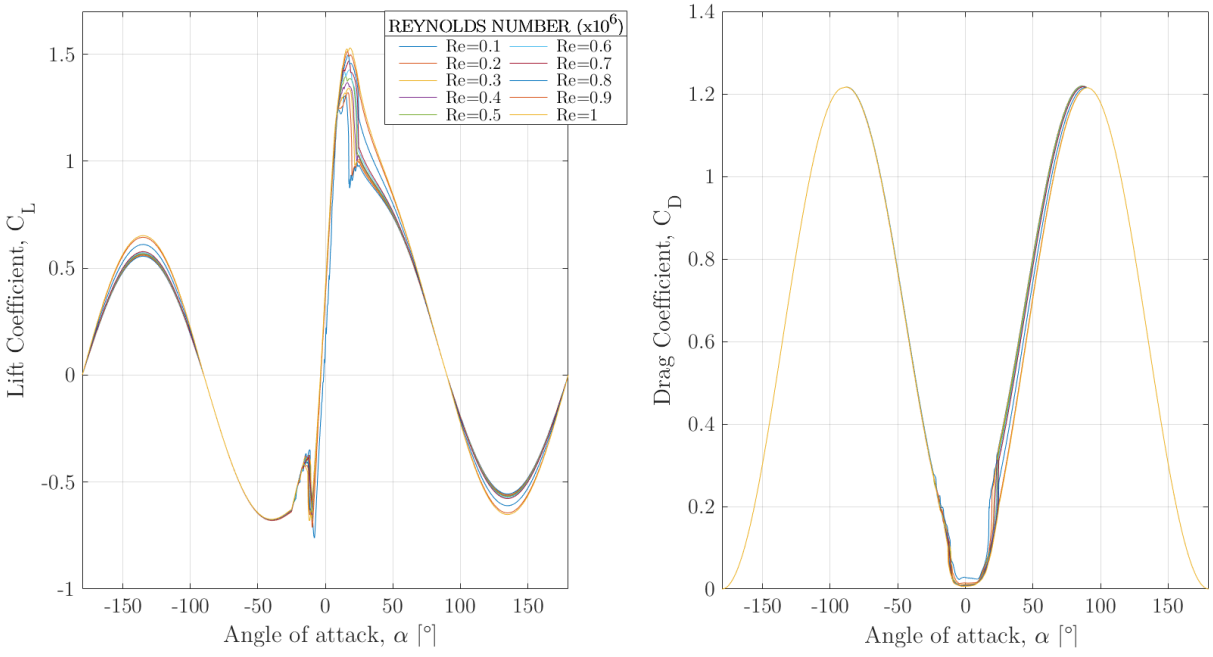
Regarding the corrections for 3D corrected data, the curves will be identical to the ones generated from 2D data in Region III, due to the fact that airfoil coefficients were not corrected in the negative angle of attack stall zone. However, this has no effect at all in the BEMT simulation power output. 3D extrapolated data is organised in the same way that in the 3D correction algorithm: One EXCEL file is used for each airfoil and a

different sheet is reserved for each blade element node along the blade span. In each sheet, data is stored in 3-column blocks (α, C_l, C_d), one for each Re number.

Some examples of the extrapolated 3D data are shown below:



a) Extrapolation of the 7th element airfoil data



b) Extrapolation of the 20th element airfoil data

Figure 3.15 360° extrapolation of aerodynamic coefficients (SG6040 airfoil, multi Re number)

Further explanation about the code can be found in the MATLAB® file (1A.3 Viterna_360_extrapolation.m) comments in APPENDIX A.

Chapter 4

HORIZONTAL AXIS WIND TURBINE AERODYNAMICS

In this chapter all the equations implemented in the MATLAB® model are described. The chief aim is not to deduce all the expressions needed to reach the final forms, but to approach the reader in a physical and analytical manner to the equations of the Blade Element Momentum Theory (BEMT).

BEMT results from the combination of the Momentum Theory (MT) and the Blade Element Theory (BET) and after applying some correction factors to the original expressions in order to overcome some limitations of the theory, it is widely used among researchers and blade designers [21]–[24], [28], [38]–[42] . The reason is that after applying proper corrections to BEMT and to the estimated airfoil polars the results are in good agreement with CFD models [21], [28], [40], [42] consuming much less computational time.

A more detailed deduction and description of HAWT aerodynamics and BEMT can be easily found in the literature [20], [43], [44], with subtle changes in the nomenclature.

4.1 The Momentum Theory

The Momentum Theory (MT) states that the axial force and the torque in the wind turbine blades are equal to the momentum and angular momentum change of the flow passing through the turbine.

4.1.1 1D Momentum Theory

The most simplified model is derived from the **actuator disc concept**. This actuator has infinite blades, no drag and no rotation of the wake. Pressure far upstream and far downstream are equal to ambient pressure.

When wind touches the actuator a reversed flow $-aU_\infty$ appears, where a is called the **axial induction factor**. The velocity on the disc plane is, thus, $U_d = U_\infty(1-a)$. The overall change of momentum is given by:

$$Momentum\ change = (U_\infty - U_w)\dot{m} = (U_\infty - U_w)\rho A_d U_d = F \tag{4.1}$$

Were U_w is the velocity downstream.

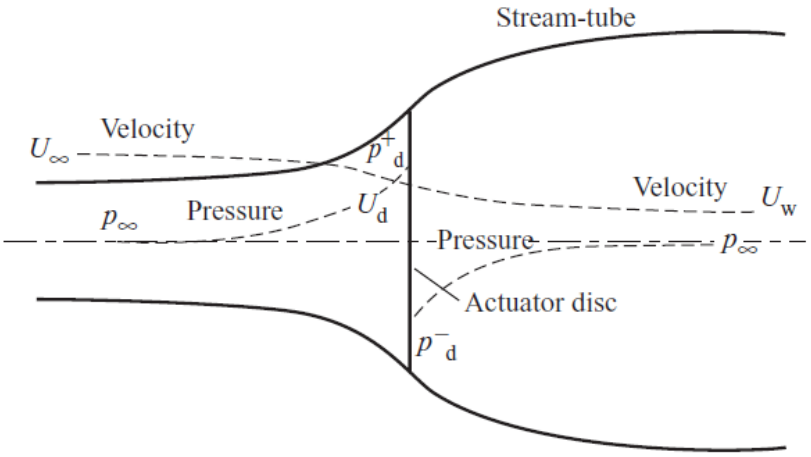


Figure 4.1 An Energy Extracting Actuator Disc and Stream-tube [44]

Knowing that pressure difference in the disk sides is causing that change of momentum and applying Bernoulli's energy conservation in both sides of the disc [44] the value of the velocity of the wake $U_w = U_\infty(1-2a)$ is obtained. After some rearranging, Equation (4.1) is written as:

$$Momentum\ change = 2\rho A_d {U_\infty}^2 a(1-a) = F \tag{4.2}$$

The power developed in the actuator disk will be, thus:

$$P = FU_d = 2\rho A_d {U_\infty}^3 a(1-a)^2 \tag{4.3}$$

And the respective power coefficient:

$$C_P = \frac{P}{\frac{1}{2}\rho A_d {U_\infty}^3} = \frac{2\rho A_d {U_\infty}^3 a(1-a)^2}{\frac{1}{2}\rho A_d {U_\infty}^3} = 4a(1-a)^2 \tag{4.4}$$

If one derives the previous expression respect to a, equals to 0 and solves, the maximum for the power coefficient is found for $a = \frac{1}{3}$ with $C_{P,max} = \frac{16}{27}$. This is known as the **Betz limit**, and describes a theoretical (and impossible) limit of power extraction from wind.

Now, if we suppose a differential annular ring of radius r and width dr, we can apply Equation (4.2) in order to calculate the differential thrust force generated on the annulus.

$$dT = 2\rho dA_d {U_\infty}^2 a(1-a) = 2\rho(2\pi r dr){U_\infty}^2 a(1-a)$$

$$dT = 4\pi\rho {U_\infty}^2 a(1-a) r dr \tag{4.5}$$

This expression will be used later on this chapter.

4.1.2 2D Momentum Theory

In a real wind turbine air exerts a torque on the rotor disc when it passes through the rotor plane and consequently it receives a reaction torque that makes the air rotate in a direction opposite to that of the rotor. Thus, the air gains angular momentum and has an axial and tangential component of velocity after crossing the disc.

As it is seen in Figure 4.2, air enters the rotor with no rotational motion and the rotational motion transfer happens entirely across the thickness of the disc. This change is described in terms of a', called the **tangential induction factor**. At the middle of the disc width the tangential velocity of air is $a'\omega r$ and immediately downstream the disc $2a'\omega r$.

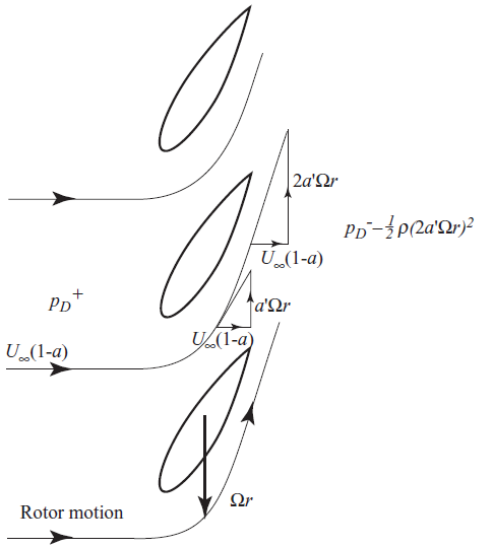


Figure 4.2 Tangential Velocity increase across the disc thickness [44]

Supposing an annular ring equal to the one on the previous section, the torque acting on the annulus will be the responsible of the change of angular momentum of the air. The whole disc will be composed of several annuli, and each one will be only responsible of the angular momentum of the air crossing it.

Torque = rate of change of angular momentum
= mass flow rate x change of tangential velocity x radius

Thus:

$$\begin{aligned} dQ = d\dot{m}(2a'\omega r)r = \rho dA_d U_\infty (1-a)2a'\omega r^2 \\ = \rho(2\pi r dr)U_\infty(1-a)2a'\omega r^2 \end{aligned}$$

Grouping terms, we reach the following expression:

$$dQ = 4\pi\rho U_\infty (1-a)a'\omega r^3 dr \tag{4.6}$$

4.2 The Blade Element Theory

The Blade Element Theory (BET) assumes that forces on a span-wise blade element can be calculated by means of the already seen (Section 3.1.2) 2D airfoil characteristics. One assumption of the BET is that flow does not move in a radial direction. 3D effects are also ignored and that is why it is so important to correct airfoil characteristic data (Section 3.3). Each blade element will sweep out an annular ring when it is rotating.

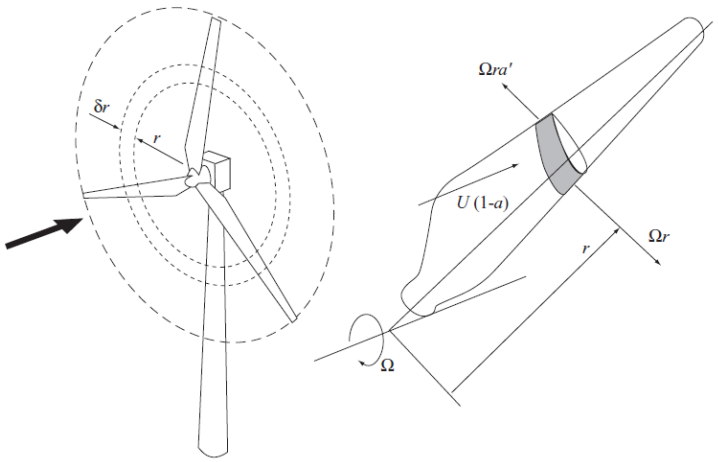


Figure 4.3 Rotating blade element [44]

The relative velocity on the blade section is composed by the axial velocity in the rotor and the tangential relative velocity, which seen from the blade is the sum of the induced velocity in the wind $a'\omega r$ (middle point of the disc) and the rotor velocity ωr.

$$W = \sqrt{{U_\infty}^2(1-a)^2 + (\omega r)^2(1+a')^2} \tag{4.7}$$

The angle between relative velocity W and the rotor plane is called the inflow angle φ and is the sum of the angle of attack α and the pitch + twist angle, β , in the section.

$$\varphi = \alpha + \beta \tag{4.8}$$

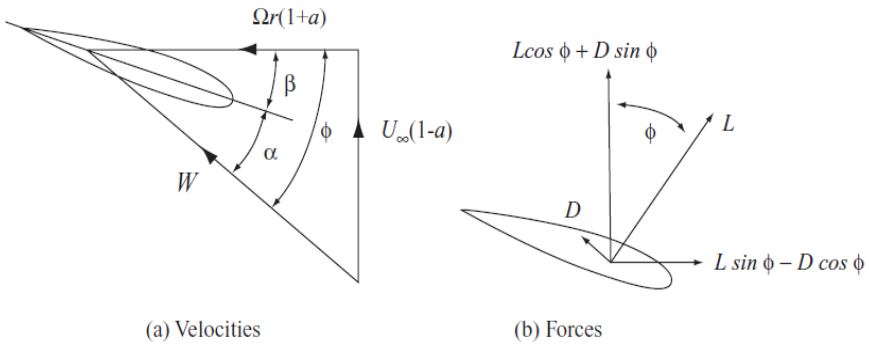


Figure 4.4 Blade element velocity triangle and forces [44]

As it is already known, a drag force (parallel to W) and a lift force (perpendicular to W) appear when the airflow attacks the airfoil leading edge. For an elemental span-wise length dr their value is:

$$dL = \frac{1}{2}\rho W^2 c C_l dr \tag{4.9}$$

$$dD = \frac{1}{2}\rho W^2 c C_d dr \tag{4.10}$$

These forces can be decomposed into the rotor axial and tangential axis, and multiplying the terms times the number of blades (B) of the rotor, the axial force (thrust) and the torque over the rotor annulus can be expressed:

$$dT = B\ (dL\cos\varphi + dD\sin\varphi) = B\frac{1}{2}\rho W^2 c C_n dr \tag{4.11}$$

$$dQ = B\ (dL\sin\varphi - dD\cos\varphi)r = B\frac{1}{2}\rho W^2 c C_t r dr \tag{4.12}$$

Where:

$$C_n = C_l\cos\varphi + \mathrm{C}_d\sin\varphi \tag{4.13}$$

$$C_t = C_l\sin\varphi - C_d\cos\varphi \tag{4.14}$$

4.3 The Blade Element Momentum Theory (BEMT)

The Blade Element Momentum Theory combines the MT and BET and equalizes the expressions of the differential thrust and torque over an elemental annulus in order to obtain the span-wise values of the induction factors a and a'. This theory shares the limitations of both theories, such us an infinite number of blades and no radial flow.

For the **thrust forces**, equations (4.5) and (4.11) are equalized giving:

$$4\pi\rho {U_\infty}^2 a(1-a)r dr = B\frac{1}{2}\rho W^2 c C_n dr \tag{4.15}$$

Similarly, for the **torque**, equations (4.6) and (4.12) are used:

$$4\pi\rho U_\infty (1-a)a'\omega r^3 dr = B\frac{1}{2}\rho W^2 c C_t r dr \tag{4.16}$$

The local solidity is defined as the ratio blade/air (Figure 4.5) of the swept annulus at the given radial position:

$$\sigma' = \frac{Bcdr}{2\pi r dr} = \frac{Bc}{2\pi r} \tag{4.17}$$

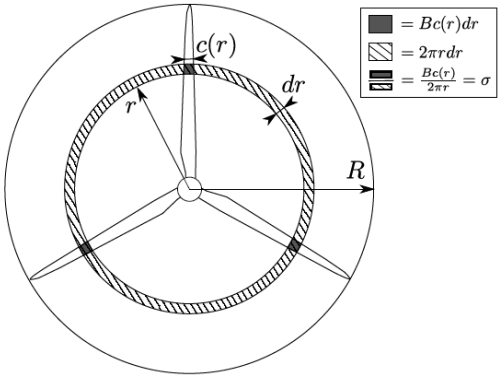


Figure 4.5 Local solidity description [45]

With the definition of local solidity, the use of the triangle of velocities (Figure 4.4) and simple algebraic manipulation Equations (4.15) and (4.16) become, respectively:

$$\frac{a}{1-a} = \frac{\sigma' C_n}{4\sin^2\varphi} \tag{4.18}$$

$$\frac{a'}{1+a'} = \frac{\sigma' C_t}{4\sin\varphi\cos\varphi} \tag{4.19}$$

And finally, letting induction factors free:

$$a = \frac{1}{\frac{4\sin^2\varphi}{\sigma' C_n} + 1} \tag{4.20}$$

$$a' = \frac{1}{\frac{4\sin\varphi\cos\varphi}{\sigma' C_t} - 1} \tag{4.21}$$

Also, derived from the velocity triangle:

$$\tan\varphi = \frac{U_\infty(1-a)}{\omega r(1+a')} \tag{4.22}$$

With these 3 equations span-wise values of the induction factors a and a' can be calculated iteratively if we know the values of lift and drag coefficients of the airfoil as a function of the angle of attack α. Initial values of the induction factor will be supposed and after calculating C_l, C_d, φ, C_n, C_t ,σ'etc. the new a and a' will be computed. Old and new values are compared and while the difference is higher than a certain tolerance a loop will continue running proving each time with different induction values, closer to the convergence ones. This iterative procedure is the basis of all the BEMT codes and will be explained carefully in Chapter 6.

4.4 The improved BEMT

The improved BEMT is the name that is given among researchers to the BEMT when some corrections are applied to the equations in order to overcome some of its limitations (infinite number of blades and no radial flow, among others).

In this section tip and hub loss effects and the correction for the induction factor in the turbulent wake state will be analysed. Among the correction alternatives, the ones written in *Aerodyn Theory Manual* [46] will be chosen.

4.4.1 Tip losses: Physical phenomena and correction model

The tip loss effect is the power reduction on the outermost part of the blade that is mainly derived from having a finite number of blades in the rotor. This leads to a non-uniformity in the flow crossing the rotor.

For a finite number of blades a will vary azimuthally, because air particles close to the blades will be affected by the pressure field of the blade, but those between 2 blades will not have the same interaction.

Close to the rotor center, when r is low, solidity σ' is high, and the wind turbine behaves like a continuous disc: almost all particles will interact with the blade and the pressure of the particles upwind is high. This makes the flow move radially and at the tip of the blade a tip vortex is shed downstream into the wake describing a helicoidally path. This vortex creates a high value of the induction factor locally, and thus, a fall in the tangential force acting on the blade that leads to a torque decrease.

As the induction factor varies both azimuthally and radially $a = a(r,\psi)$, for a given r two induction factors are defined: the induction factor local to the blade $a_b(r)$ and the azimuthally averaged induction factor $\bar{a}(r)$.

With uniform circulation, the average induction factor would be radially uniform and only a single vortex would be emitted from the tip (and other from the hub). However, this leads to a discontinuity of axial velocity in the wake boundary ($a = 0$). In reality the circulation varies along the span, and consequently a vortex sheet is continuously emitted from the trailing edge into the wake with the same velocity.

- PRANDTL TIP LOSS CORRECTION FACTOR

The easiest to implement and more straightforward numerical correction model is the one developed by Prandtl. He assumed that the helical sheets in the wake can be replaced by successions of vortex discs moving with a constant speed with the central wake velocity.

Practically, Prandtl's factor F_{tip} defines the ratio between the azimuthally averaged induction value and the one local to the blade, $F_{tip} = \frac{\bar{a}}{a_b}$. Then, this factor will be included into the BEM equations. Prandtl's tip loss correction factor is defined as:

$$F_{tip}(r) = \frac{2}{\pi}\cos^{-1} e^{-\frac{B}{2}\left(\frac{R-r}{r\sin\varphi}\right)} \tag{4.23}$$

As it can be seen in Figure 4.6, the tip loss factor remains near 1 within the most part of the radial position but falls to 0 it the outer part of the blade, when approaching to the blade tip.

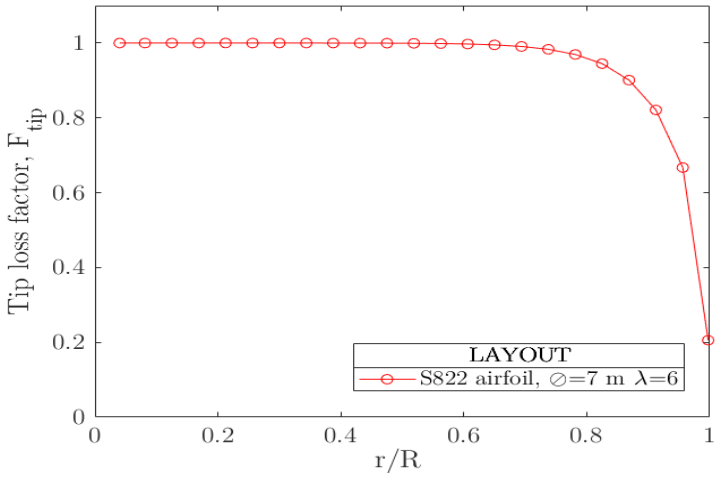


Figure 4.6 Span-wise values of the tip loss correction factor

This correction gives an easy and quite accurate correction to the tip losses on the blade, especially when airfoil data has been properly corrected to 3D effects.

Nowadays there are several different tip loss correction factors available in the literature. Besides, current research is focused in the study and development of new tip loss corrections [45], [47].

4.4.2 Hub losses: Physical phenomena and correction model

At the hub the blade circulation also falls to 0 and another vortex is shed downstream to the wake. As the process is similar to the previous case, it is also modelled by means of a correction factor:

$$F_{hub}(r) = \frac{2}{\pi}\cos^{-1} e^{-\frac{B}{2}\left(\frac{r-R_{hub}}{R_{hub}\sin\varphi}\right)} \tag{4.24}$$

The hub correction factor remains nearly equal to 1 on the majority of the blade span but falls to 0 when $r \to R_{hub}$.

4.4.3 Overall loss correction factor, F

As both the tip loss correction factor and the hub loss correction factor are introduced in order to correct the effect of a vortex on the induced velocities, they can be combined into an overall loss correction factor:

$$F = F_{tip}F_{hub} \tag{4.25}$$

This factor is nearly 1 on the mid span zone of the blade and falls to 0 when approaching the root (due to hub losses) and the tip (due to tip losses).

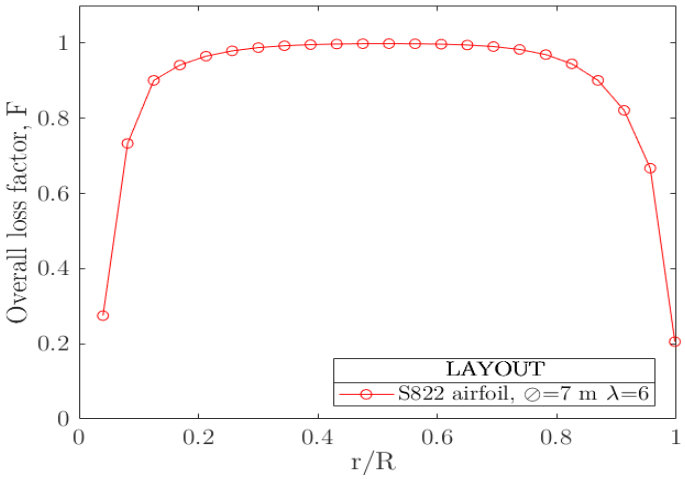


Figure 4.7 Span-wise values of the overall loss correction factor

Now, this term has to be introduced intro BEMT equations. Blade element forces have to be calculated taking into account the local velocity at the blade, so they have to be computed using the local induction factor a_b. On the other hand, in the momentum equations the averaged value of the induction factor, $\bar{a}$, has to be used. This means that we can write the corrected BEM equations referring to one or the other. Typically, and so does AeroDyn, equations are written calling a to the induction factor local to the blade, a_b. So, from now on in this work, a will represent the local axial induction factor a_b.

Thus, momentum equations (4.5) and (4.6) are modified into:

$$dT = 4\pi\rho {U_\infty}^2 aF(1-a)rdr \tag{4.26}$$

$$dQ = 4\pi\rho U_\infty (1-a)a'F\omega r^3 dr \tag{4.27}$$

Note that in these equations only the induction term that comes from the change of velocity between upstream and wake has been changed and not the induction term of the velocity at the disc. This was stated by Glauert [48] and widely applied in the literature, but authors like Wilson and Lissaman suggested to correct also the induction term of the wind velocity at the disc plane [29, p.39].

Developing the equations above we reach the corrected BEMT expressions:

$$a = \frac{1}{\frac{4F\sin^2\varphi}{\sigma' C_n} + 1} \tag{4.28}$$

$$a' = \frac{1}{\frac{4F\sin\varphi\cos\varphi}{\sigma' C_t} - 1} \tag{4.29}$$

4.4.4 Breakdown of the Momentum Theory. Glauert correction

Wind turbines sometimes experience the turbulent windmill state, usually during startup or shutdown. This rarely happens in steady state conditions but is always implemented in BEMT codes [46]. When a is high, the MT predicts a flow reversal and lower thrust coefficient. The value of thrust coefficient is defined as:

$$C_T = \frac{Thrust}{\frac{1}{2}\rho A_d {U_\infty}^2} = \frac{2\rho A_d {U_\infty}^2 aF(1-a)}{\frac{1}{2}\rho A_d {U_\infty}^2} = 4aF(1-a) \tag{4.30}$$

In reality, what happens is that the wake becomes turbulent by mixing with the outside. When a is higher that a certain value a_c the MT becomes invalid. Glauert (1926) fit a parabola to some experimental data of rotors operating in this state. However, the fit was not accurate enough and did not account for tip/hub losses. Several authors have shown different empirical correlations, with different values of the transition point a_c.

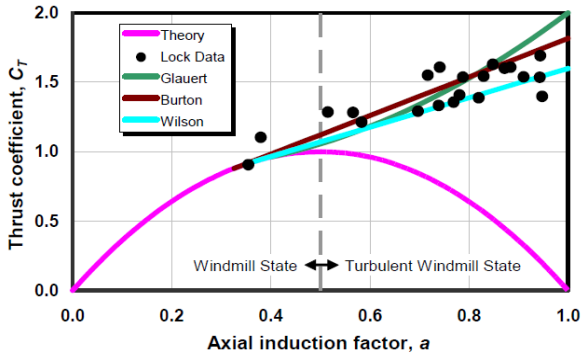


Figure 4.8 Different classical empirical corrections for the turbulent windmill state [50]

In this work the correction proposed by **Buhl** (2005) [50] is used, which is also regarded in *Aerodyn Theory Manual.* The author proposes the following parabolic relation:

$$C_T = \frac{8}{9} + \left(4F - \frac{40}{9}\right) a + \left(\frac{50}{9} - 4F\right) a^2 \tag{4.31}$$

For $a > a_c = 0.4$ or, which is the same, $C_T > 0.96F$. If we free a:

$$a = \frac{18F - 20 - 3\sqrt{C_T(50-36F) + 12F(3F-4)}}{36F - 50} \tag{4.32}$$

For each iteration in a BEMT code C_T is calculated with the BET expression:

$$C_T = \frac{\sigma'(1-a)^2 C_n}{\sin^2 \varphi} \tag{4.33}$$

If C_T is higher than 0.96F, Equation (4.32) will be used to calculate the new value of a . If not, the standard BEMT is used to calculate the new value of the axial induction factor (Equation (4.28)).

Chapter 5

SMALL WIND TURBINE POWER CONTROL SYSTEMS

A wind turbine is basically a device that captures part of the wind energy and converts it into useful electric energy. Nowadays, due to the development of the electric generators, power electronics and control theories it is possible to act into some elements of the wind turbine in order to maximise **energy capture**, reduce **dynamic mechanical loads** and improve the generated **energy quality**. However, it should be noticed that these pursued objectives are sometimes conflicting among them and that the applied control system has to pay off in terms of investment. That is the main reason why full control strategies are never used for small scale wind turbines.

In this chapter a brief review on the control strategies that are used in small scale wind turbines is done. The typical power curves obtained from the implementation of those control strategies will be depicted against an ideal power curve. This curve can be understood as the curve of a turbine that maximises energy capture below nominal speed (region I) and limits power above rated (region III). Region II is a transition zone.

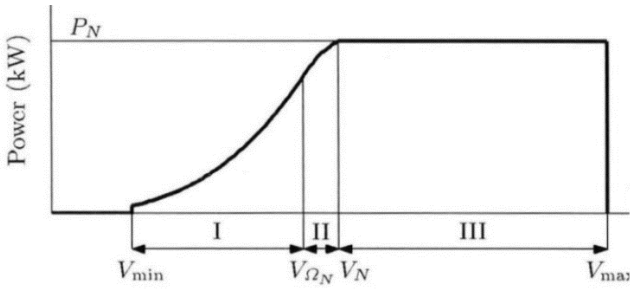


Figure 5.1 Ideal Power Curve [4]

The control strategy chosen (VS-FP) for the object wind turbine of the project will be described with more detail. The chapter ends with a list of the different electric generators used for variable speed wind turbines.

5.1 Fixed-Speed Fixed-Pitch (FS-FP)

In this scheme the asynchronous electric machine is directly coupled to the power network. This means that the generator speed is locked by the grid frequency and consequently the rotor works at a single rotational speed ($\omega = const$).

The blades of the wind turbine will be designed to work optimally at certain angles of attack that will appear on the blade when the turbine works at optimum tip-speed ratio, λ_{design}. As rotational speed ω is constant, optimum TSR and maximum power coefficient $C_{p_{max}}$ will only be achieved at one wind velocity (point E in Figure 5.3). So, this control strategy does not maximise the energy capture in region I.

Regarding power limitation, there is **no active control of the blade pitch** (rotation respect to the longitudinal axis of the blade). The generation of power is therefore limited by the geometry dependent stall phenomenon. That is why these turbines are said to be **stall regulated**. This regulation mechanism is easily understood looking at the velocity triangle of a blade section (Figure 5.2) : when incoming wind velocity increases, as ω is fixed, the inflow angle φ rises and with a constant β, the angle of attack α increases. This leads to a drop in the lift coefficient and an increase of the drag coefficient due to stall. Consequently, the power generation decreases.

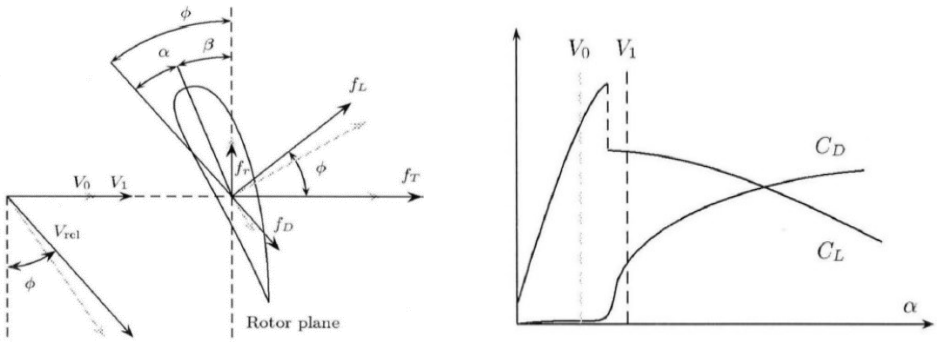


Figure 5.2 Passive stall strategy for power limitation [4]

The power curve of a typical FS-FP wind turbine is depicted below:

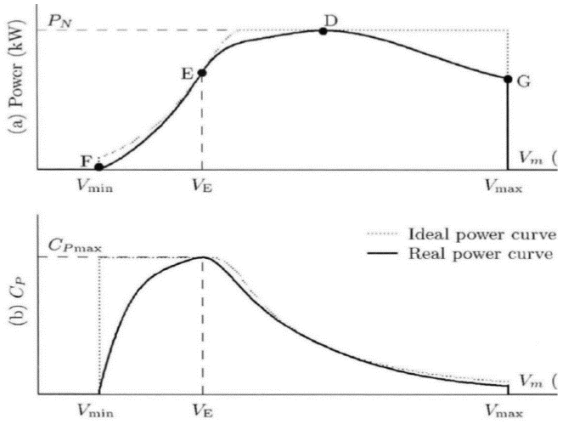


Figure 5.3 Basic FS-FP control strategy: power and Cp curves [4]

FS-FP control scheme is simple and low cost, but there is neither energy capture optimization nor energy quality improvement. That is why even if it was used decades ago, now is only used for very small wind turbines (near 1 kW). This control method is sometimes complemented with yaw and tilt disorientation to reduce power and torque.

5.2 Fixed-Speed Variable-Pitch (FS-VP)

Wind turbines with FS-VP control strategy also work at a single rotational speed ω. Thus, as it has been said in the previous section, maximum efficiency will only be achieved at one wind velocity and energy capture will not be maximised over region I.

Regarding power limitation (region III), **active pitch control** limits power above rated wind speed with the help of the controller by rotating each blade about is longitudinal axis. The main benefits of the pitch control are the increased energy capture, the load alleviation at shut down and the possibility of aerodynamic braking.

There are **2 different techniques** to achieve pitch control power limitation. The most usual is called *Pitch-to-feather* and consists in feathering the blades (increasing β) when incoming wind increases so that lift coefficient is reduced by reducing the angle of attack α. Thrust forces over the turbine are lower with this method but large positive pitch angles are required. The alternative method is called *Pitch-to-stall* and consists in pitching the blade in the opposite direction that in the previous case so that the angle of attack α increases. The blade is then brought to a controlled stall condition. Forces on the blade are higher with this method, but the control effort is lower because lower blade pitch angles are needed.

The power curve of a typical FS-VP wind turbine is shown in the figure below:

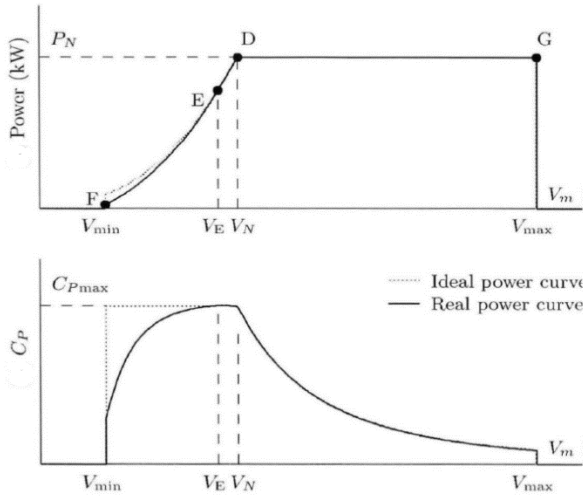


Figure 5.4 Basic FS-FP control strategy: power and Cp curves [4]

As it can be seen, power is limited and attached to the ideal curve shape above rated. However, this strategy has some drawbacks such us the extra cost of the pitching system, the energy needed to pitch the blades and the higher need of maintenance.

This control strategy is **not commonly used for small scale wind turbines** placed in low wind velocity locations (case of this work) since its major upside is the power limitation at high wind velocities.

An example of a 10 kW rated FS-VP wind turbine is the *Enair E200L* model [51].

5.3 Variable-Speed Fixed-Pitch (VS-FP): Case of this work

The variable rotational speed alternative has become popular in modern wind turbines, particularly in **low wind velocity** operation. The chief benefits of this strategy are the larger energy capture in Region I, the alleviation of dynamic loads and the power quality enhancement. The **design of the blade** object of this work will be done regarding the blade as a component of a **Variable-Speed Fixed-Pitch** wind turbine.

The maximum conversion efficiency ($C_{p_{max}}$) of a wind turbine is given when angles of attack over the blade are those of design and the tip speed ratio is on the design value , λ_{design}. If we want to maintain design angles of attack, TSR needs to remain constant at its design value within a velocity range, and therefore rotational speed will need to vary proportionally to wind velocity:

$$\omega = \frac{U_\infty \lambda_{design}}{R_{tip}} \tag{5.1}$$

As it is known, in order to achieve variable rotational speed in the wind turbine rotor the generator needs to be decoupled from the grid frequency. This is solved by placing an electronic converter between the generator and the grid. Thus, the wind turbine can be controlled to track the optimum speed ($C_{p_{max}}$ locus) as wind velocity fluctuates.

Regarding power limitation **2 different VS-FP approaches** exist: active stall regulation and passive stall regulation. The former one uses variable velocity control throughout all the velocity range and can maintain a constant power output above rated. However, this is rarely done up to date in wind turbines and would be considered only for large scale machines due to the high cost of the needed generators [44, p. 484].

- **VS-FP with passive stall regulation**

The passive stall approach is much more common and the vast majority of low power rating wind turbines are regulated following this control strategy. In low speed wind range the turbine is programed to follow the $C_{p_{max}}$ locus by changing the rotational speed proportionally with incoming wind.

When it reaches a certain point (point E in Figure 5.5) the rotational speed reaches its maximum and remains constant. Therefore, optimum angles of attack are no longer those of design, and the power curve will detach from the optimum cubic one. Above this point, the turbine will behave exactly as the previously described FS-FP wind turbine and the power will be limited by **stall phenomenon**.

One of the major drawbacks of the passive stall limitation is that power above rated is highly geometry dependent and that power output is not always easy to characterise.

The power curve of a typical VS-FP wind turbine is shown in the figure below:

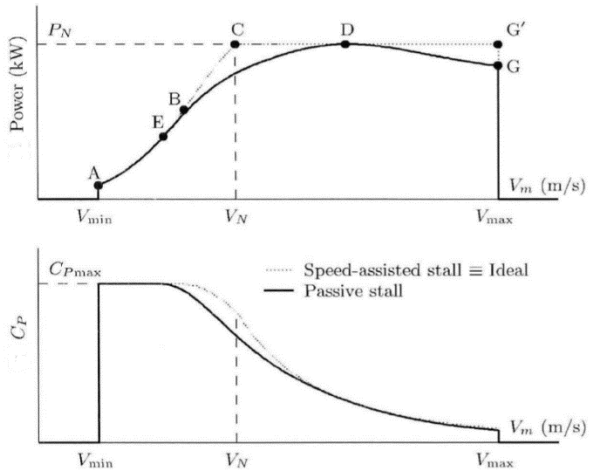


Figure 5.5 VS-FP control strategy: power and Cp curves [4]

Note that the shape of the curve between points E and G is geometrically dependent and it will rarely maintain that subtle slope between points D and G.

- **Variable-Speed range limitation**

From the previous picture it is clear that the wider the variable speed range is (from A to E) the higher the energy capture will be. However, variable speed range is typically limited due 2 main reasons:

1. Noise constraint:

The blade tip velocity is a key parameter in the emission of aerodynamic sound from the wind turbine. That is why commonly the rotational velocity is limited to a maximum value ω_{max} [44, p. 484]. Large modern wind turbines limit the rotor rotation speeds to keep the tip speeds under about 65 m/s [52, p. 11].

As small wind turbines are installed closer to people and their blade tip design does not reduce noise emission, the tip speed in this work will be set to:

$$U_{tip,max} = \omega_{max} R_{tip} = \mathbf{45\ m/s} \tag{5.2}$$

2. Avoidance of resonances:

Rotational speed is also limited due to structural reasons. In order to avoid that the wind excites natural frequencies of the structure elements it is preferred a narrow variable speed range.

This issue is usually addressed through the use of a Campbell diagram that shows the natural frequencies of the wind turbine as a function of rotor rotational speed. Due to rotational sampling of turbulence, the blades will be excited by 0P, 1P, 2P etc. effects and the support structure will be excited by 0P, 1P, 3P, 6P, etc. effects. These lines of

constant frequency are added to the Campbell diagram in order to identify areas of possible resonance. Thus, the wider the range of rotational speed, the harder will be to avoid resonances in the structure [35].

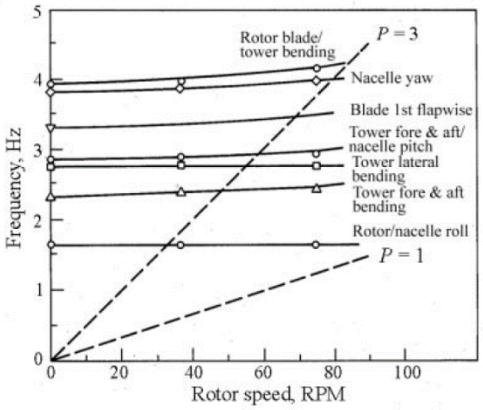


Figure 5.6 Example of a Campbell diagram for a wind turbine [20]

- ## Electric generators for variable speed small wind turbines

Two main competing generator technologies for small scale variable speed wind turbines are the classic induction generators (IG) and the permanent magnet synchronous generator (PMSG). Some IGs such us the squirrelcage type can be operated by direct-grid connection but they allow a minimal variable speed range.

PMSGs have high efficiency, low maintenance costs due to the absence of slippery rings and brushes, and do not require an additional power supply for the field excitation. That is reason why the **PMSG** is the preferred generator for small scale wind turbines [43, p. 227], [53, p. 6].

The scheme of a PMSG connected to the grid is shown in the figure below:

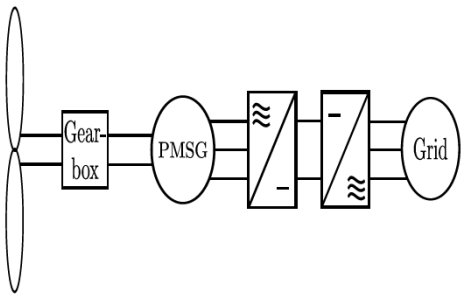


Figure 5.7 Simple scheme of a PMSG wind turbine feeding a grid [53]

Chapter 6

BLADE DESIGN: PARAMETERS, VARIABLES AND ALGORITHMS

This chapter is the most important one in order to understand the blade design process that has been followed. It has been divided from the simulation results, which are shown in the following chapter, in order to make everything clearer and easier to understand.

The current chapter is divided in 4 sections: The first one establishes the different **parameters** used during the simulations, which do not change from one simulation to the other. In the second section the design **variables** are explained, those which are different from one simulation to the other. The third one describes the overall **method** followed to generate the different blade layouts. In the last section, the practical application of the BEMT in order to reach a proper blade design is specified. This section is divided into 2 sub sections: one for the blade design **algorithm**, from which the shape of the blade is obtained, and other one for the off-design behaviour of the wind turbine from which the power curve of the wind turbine can be found.

The whole code for the blade design has been written in MATLAB®. Thus, during this chapter several MATLAB® scripts will be mentioned and referenced. All of them can be found in the APPENDIX A, commented and with proper references to the equations in this document.

6.1 Blade design parameters

All the parameters needed to perform the design simulations are described below:

6.1.1 Rated power, P_{rated}

For a VS-FP stall-regulated wind turbine the electric rated power is traditionally considered as the maximum electric power output of the turbine. The aim of this work is the design of a **10 kW rated** wind turbine. Figure 5.5 shows the relationship between a stall-regulated and a pitch-regulated wind turbine rated power. The velocity at which rated power is achieved is called rated velocity, U_{rated}.

6.1.2 Number of blades, B

The higher it is the number of blades of the wind turbine the higher will be the solidity of the rotor. The relation between the number of blades of a typical wind turbine and the power coefficient is shown in Figure 6.1 as a function of the tip speed ratio (TSR). It appears that optimum solidity is reached with 3 blades (2 is also a good choice because the curve is wider and can result in a quite high energy capture). Also, due to aesthetics 3-bladed wind turbines are the most common choice nowadays. Therefore, in this work the number of blades will be $\mathbf{B = 3}$.

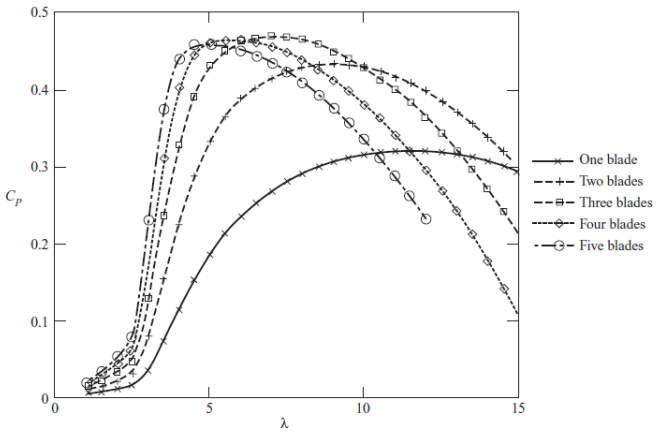


Figure 6.1 Effect of the number of blades in the power coefficient [44]

6.1.3 Design Tip Speed Ratio, λ_{design}

Another important parameter regarding the power of wind turbines is the optimal or design tip speed ratio. The tip speed ratio is defined as the ratio of the blade tip speed to the free stream wind speed:

$$\lambda = \frac{\omega R_{tip}}{U_\infty} \tag{6.1}$$

As explained in the previous chapter the turbine will operate with variable speed within a wind velocity range. In this range the tip speed ratio will have its design value, λ_{design} and the power coefficient will be maximum (nearly maximum in reality, due to Re number dependence). This means that the design value of the tip speed ratio has to be chosen carefully. A low value leads to a low interaction between blade and air passing through the rotor plane and this means a loss of efficiency. On the other hand, high tip speed ratios lead to blade erosion, high noise emission and vibration.

Some suggested values for the design tip speed ratio are shown in Table 6.1 in function of the number of blades. It shows how for lower blade number higher tip speed ratios are needed, in order to avoid air passing through the turbine without interacting with the blades.

Table 6.1 Suggested blade number, B, for different tip speed ratios, λ [20]

λ	B
1	8–24
2	6–12
3	3–6
4	3–4
>4	1–3

The optimal value of the tip speed ratio is related with the time needed by the air stream to reestablish itself ($\frac{s}{U_\infty}$) and with the time needed for a blade to reach the position of the preceding one ($\frac{2\pi}{B\omega}$) . The maximum power extraction occurs when these times are approximately equal [54]. Thus, and optimal rotational speed and tip speed ratio can be defined as:

$$\lambda_{design} = \frac{\omega_{opt} R_{tip}}{U_\infty} = \frac{\frac{2\pi U_\infty}{Bs} R_{tip}}{U_\infty} = \frac{2\pi}{B}\left(\frac{R_{tip}}{s}\right) \tag{6.2}$$

Where B is the number of blades and s the length of the disturbed wind stream. The ratio R_{tip}/s has been empirically proved to be nearly 2. Thus:

$$\lambda_{design} \approx \frac{4\pi}{B} = 4.19$$

With proper airfoil design, the optimal TSR values may be approximately 30-35 percent above these values [54]. As the blade design will be with variable chord and twist, the design value in this work is taken as:

$$\lambda_{design} = 1.35\frac{4\pi}{B} = 5.65 \rightarrow \mathbf{6} \tag{6.3}$$

6.1.4 Design wind velocity, U_{design}

The design velocity is used to determine the Re number in each blade section. It needs to be chosen a value inside the variable speed wind velocity range (which, as is explained below in this section arrives until 7.5 m/s). If the blade airfoils were not sensible to Re numbers, the design velocity would not be determinant. However, for low Re number airfoils it happens that their performance changes with Re number and the blade will be only working at ideal angle of attack at 1 velocity of the whole variable speed velocity range. This is the reason why the C_p vs U curves are not straight horizontal lines whenever $\lambda = \lambda_{design}$. In order to maximise the power generation in *Aguilar de Codes,* the wind velocity with more annual energy content will be taken as design velocity:

$$U_{design} = \mathbf{7.5}\ \frac{\boldsymbol{m}}{\boldsymbol{s}} \tag{6.4}$$

6.1.5 Air density, ρ

The power available in the wind, and therefore the electric power generated by the wind turbine are directly proportional to the air density.

$$P_{out} = P_{wind} C_p \eta = \frac{1}{2} \rho {U_\infty}^3 \pi {R_{tip}}^2 C_p \eta \tag{6.5}$$

Where C_p is the power coefficient of the blades and η the overall efficiency of the drive train, generator and converter.

If the power output of a wind turbine is needed for a certain wind turbine location, the air density at that place needs to be estimated. The air density is considered to be $\rho_0 = 1.225\ kg/m^3$ at sea level ($h = 0$), and it decreases when altitude increases. In this work, the *International Standard Atmosphere* model has been used in order to estimate the density in Aguilar de Codes and at the hub height($h = 736 + 25\ m$).

The value of the density at an altitude h is given by:

$$\rho(h) = \rho_0 \left(\frac{T_0 + a(h - h_0)}{T_0} \right)^{\frac{-g}{aR}-1} \tag{6.6}$$

Where $\rho_0 = 1.225\ kg/m^3$ and $T_0 = 288.15\ K$ are the standard values of density and temperature for the base altitude of the troposphere ($h_0 = 0$), $g = 9.81\ m/s^2$ is the acceleration of the gravity, $R = 287\ m^2/(s^2 \cdot K)$ is the individual constant of the air and $a = -6.5\ K/km$ is the temperature gradient. Substituting the data:

$$\rho(761) = 1.225 \frac{kg}{m^3} \left(\frac{288.15\ K - 6.5 \cdot 10^{-3} \frac{K}{m} \cdot (761\ m)}{288.15\ K} \right)^{\frac{-9.81\ \frac{m}{s^2}}{-6.5 \cdot 10^{-3}\ \frac{K}{m}\ 287\ \frac{m^2}{s^2 K}} - 1} = \mathbf{1.138\ \frac{kg}{m^3}} \tag{6.7}$$

As it can be seen, the density will be lower than the value at sea level. This is traduced into a lower power generation for the same wind turbine ($\frac{1.138}{1.225} \cdot 100 \approx 93\%$).

6.1.6 Dynamic viscosity, μ

The dynamic viscosity of the air is $\mu = 1.74 \cdot 10^{-5}\ Pa \cdot s$. It is only used to compute the Re number (Equation (3.4)).

6.1.7 Geometrical parameters

- **Hub diameter, D_{hub}**

A hub diameter value of **0.3 m** has been taken. It has been taken the same hub diameter that in the SWRT (10 kW rated, developed by the NREL) wind turbine [16].

- **Root diameter, D_{root}**

The blade root will have a **cylindrical** shape. The root diameter depends on structural loads and blade constructive issues. It needs to be lower than the hub diameter.

A root diameter of **0.2 m** has been taken. In future advancements of the project the blade root will need to be mechanically validated.

- **Number of blade elements, N_e**

One of the principles of the BEMT is that the blade has to be **discretised** into several elements. Each of this element is delimited by 2 boundaries and has a node in the centre. The higher the number of blade elements, the more accurate will be the blade design simulation, but the computational cost will also increase. In practice, the number of elements is taken between 15 and 20. In this work **20 elements** will be used for the blade discretisation.

- **Number of blade boundaries, N_b**

The number of boundaries is $N_b = N_e + 1 = \mathbf{21}$. The first boundary touches the hub and the last one coincides with the blade tip. Figure 6.2 illustrates the blade discretisation scheme for BEMT: blade elements are represented in black and blade boundaries in red.

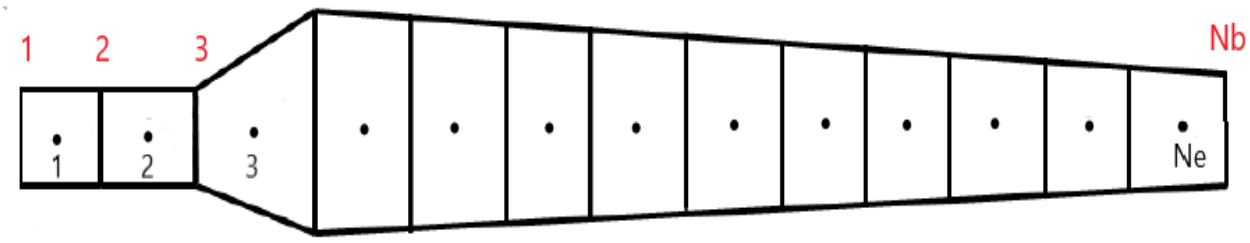


Figure 6.2 Blade span discretisation: elements and boundaries

- **Root length, L_{root}**

The root length is considered as the distance between the hub boundary and the first non-cylindrical airfoil, this is, the transition element is also considered as the blade root. Generally, the root length is close to the **15 % of the whole blade length**. As the blade will be divided into 20 elements, 3 will belong to the blade root (the first 2 will be purely cylindrical while the 3rd one will be a transition element). Thus, the first non-cylindrical airfoil will coincide with the 4th boundary.

6.1.8 Turbine operating characteristics

A wind turbine blade design requires the understanding of the blade as an element of the whole wind turbine system. Regarding the wind turbine control system, for instance, two variable speed stall-regulated wind turbines will require different rotor sizes to reach the same rating if their variable speed range is different. The parameters that define the wind turbine behaviour in this project are the followings:

- **Cut-in velocity, $U_{cut\text{-}in}$**

The cut-in velocity of wind turbine is the minimum wind velocity at which it starts to produce power. The value varies from turbine to turbine and is usually between 3-5 m/s. A value of **3 m/s** is reasonable for small wind turbines.

- **Cut-out velocity, $U_{cut\text{-}out}$**

At wind velocities higher that $U_{cut\text{-}out}$, the turbine is shut down to protect the rotor and drive train from damage due to very high loads. The value of the cut-out velocity varies among wind turbines. For small wind turbines, which are usually located in low wind sites, little energy is gained by having a high cut-out speed. In the work a cut-out velocity of **20 m/s** has been considered.

- **Power control system**

As it has been argued in Chapter 5, the blade will be designed for a **Fixed-Pitch Variable-Speed passive-stall-regulated** wind turbine.

- **Variable speed range and maximum tip speed**

The effect of the variable velocity range on a wind turbine has been discussed in Section 5.3. It was argued that the variable speed range is going to be limited due to noise and vibration constraints. The turbine will work at variable speed regimen **between 3 m/s and 7.5 m/s** of wind velocity. The value of 7.5 is included as the upper limit in order to cover the maximum energy content wind velocity in *Aguilar de Codes*.

Remembering that optimum TSR is 6, the maximum tip speed will be limited to:

$$U_{tip,max} = \lambda_{design} 7.5 = \mathbf{45\ m/s} \tag{6.8}$$

which is a low and acceptable for a small wind turbine.

- **Turbine efficiency**

A wind turbine is a system that converts wind power into useful electric power by means of different elements. The wind power is converted following some steps, which can be characterised by local efficiencies. Normally, the different subsystems are the **blades**, which transform the kinetic energy of the wind into mechanical power, the **drive train**, usually composed by a low speed shaft, a gearbox and a high speed shaft, the **electric generator** and the **converter** (if the machine works at variable velocity).

In this work the different subsystem efficiencies have been grouped in one and a value of $\eta = \eta_{dt}\eta_{gen}\eta_{conv} = \mathbf{0.85}$ has been assigned to it.

Another relevant aspect is the definition of Power Coefficient. Traditionally, books define it as the aerodynamic efficiency of the blades, the ratio between the mechanical power in the low speed shaft and the power available in the wind. However, some manufactures define it as the ratio between the produced electric power and the power in the

wind, this is, an overall efficiency of the whole machine. In this work, in order to avoid confusion, two power coefficient are defined:

- Power coefficient (without losses) = Traditional C_p, aerodynamic efficiency
- Power coefficient (with losses) = Manufacturers C_p, overall efficiency

Thus:

$$\text{Power coefficient (with losses)} = \text{Power coefficient (without losses)} \cdot \eta \tag{6.9}$$

The following figure describes the wind turbine power flow as a block diagram. In it, the nomenclature of this work has been used.

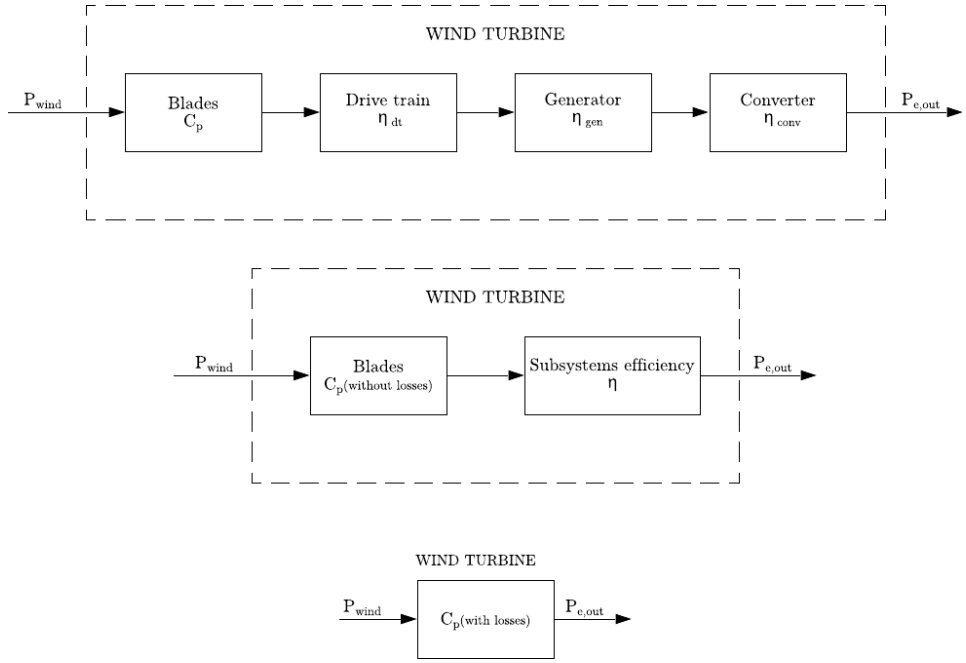


Figure 6.3 Power flow in a wind turbine

A summary of all the parameters used in this work is given in the following table:

Table 6.2 Blade design and turbine characteristics

PARAMETER	
Rated Power, P_{rated} [kW]	10
Number of blades, B	3
Design TSR, λ_{design}	6
Design velocity, U_{design} [m/s]	7.5
Air density, ρ [kg/m^3]	1.138
Dynamic viscosity, μ [Pa · s]	1.7410-5

GEOMETRICAL PARAMETERS	
Hub diameter, D_{hub} [m]	0.3
Root diameter, D_{root} [m]	0.2
Number of elements, N_e	20
Number of boundaries, N_b	21
Root length, L_{root} [m]	$0.15L_{blade}$
TURBINE OPERATING CHARACTERISTICS	
Cut-in velocity, $U_{cut\text{-}in}$ [m/s]	3
Cut-out velocity, $U_{cut\text{-}out}$ [m/s]	20
Power control system	VS-FP, passive stall
Variable speed wind range [m/s]	3-7.5
Maximum tip speed [m/s]	45
Drive train, generator and converter efficiency [%]	85

6.2 Blade design variables

The design variables are those that change from one simulation to the others.

6.2.1 Airfoil layout

As it has been explained in Chapter 3, 4 different airfoils have been used in the BEMT simulations: S822, S823, S809 and SG6040. In Chapter 7 the results of the simulations show the differences between the blades with these different airfoil layouts.

All the blades will have a common cylindrical part (2 elements) and a transition one (1 element). Then, each one will be designed with one airfoil type.

6.2.2 Chord variation

The blade chord appears in BEMT equations, inside the solidity term, but equations do not show which will be its value. In fact, the chord at each blade element is considered as an 'input' in BEMT codes. However, in the blade design process it is usually calculated inside the BEMT simulation. 3 main alternatives can be chosen:

- **Using a constant chord:**

Using a constant chord leads to a cheap and simple design, but the efficiency of the blade will be low.

- **Using a nonlinear optimization algorithm**

This method is based on nonlinear optimization algorithms which try to maximise the power coefficient expression in each section by setting some restrictions. However, it is usually bound to research, since one of the objectives of the blade design is to obtain a chord variation as smooth as possible. The computational cost is also too high with this method.

- **Using an optimal chord expression**

Using the momentum and blade element mathematical expressions the optimal chord variation can be deduced. These expressions can be found in textbooks and scientific papers and are always derived under some assumptions. They are reached by deriving the C_p expression and equating to 0. This is the alternative followed in this project.

In this work **2 different optimal chord** expressions have been studied. Thus, 8 blade design alternatives will be presented in Chapter 7 (4 airfoils x 2 chord variations). In this document the resulting designs from each chord layout will be referred as 'A chord layout design' and 'B chord layout design'.

- Optimal chord expression A

The optimal chord in this case is derived from a with wake rotation ($a' \neq 0$) by maximisig the power coefficient expression . Drag and tip losses are neglected in this model. The value of the optimal chord is given by [20, p. 123]:

$$c_{opt,A} = \frac{8\pi r}{BC_l}(1 - \cos\varphi) \tag{6.10}$$

Where r is the radial position of the element, B the number of blades, C_l the lift coefficient and φ the inflow angle. Note that the optimum value of the chord varies within the blade span. In fact, the value decreases when approaching to the blade tip.

- Optimal chord expression B

Another optimal blade chord distribution for variable-speed operating machines is proposed by the authors in *Wind Energy Handbook* [44]. The authors obtained an expression for the optimal chord by maximizing the aerodynamic torque in the variable speed range:

$$c_{opt,B} = \frac{2\pi R_{tip}}{BC_l\lambda_{design}} \frac{{}^{8}\!/_{9}}{\sqrt{{}^{4}\!/_{9} + \lambda_r{}^{2}\left(1 + \frac{{}^{2}\!/_{9}}{\lambda_r{}^{2}}\right)^{2}}} \tag{6.11}$$

Where $\lambda_r = \frac{wr}{U_\infty}$ is the local speed ratio at the blade station, B the number of blades, C_l the lift coefficient and λ_{design} the design speed ratio. This configuration leads to higher chords in the inner part of the blade than with A chord layout.

Other chord expressions such us the ones proposed by *Betz* or *Schmitz* have been discarded because the lead to very high chord values near the hub.

6.2.3 Twist angle along blade span

The twist angle β at each blade station also defines the blade geometry and will change from one design simulation to the other. However, it will be calculated inside the on-design BEMT script (A.4 Main_DESIGN.m) using the values of the resulting inflow angle and optimum angle of attack (Equation (4.8)).

6.2.4 Rotor diameter

The area swept by the wind turbine blades is function of the rotor diameter:

$$A = \frac{\pi}{4} {D_{rotor}}^2 \tag{6.12}$$

where

$$D_{rotor} = 2L_{blade} + D_{hub} \tag{6.13}$$

Hence, the power production will increase with the **square** of the rotor diameter and will be determinant in order to reach the required power generation. In this work several blade design alternatives will be proposed and each of them will need a different rotor diameter in order to reach the desired rating.

In commercial wind turbines, especially those with variable-speed variable-pitch control system, power rating and rotor size are highly related. In those cases, rotor size can be approximated because rated power is usually reached at maximum C_p, and this can be, somehow guessed. However, passive-stall-regulated wind turbines usually reach their power peak at lower turbine efficiencies. In this work an iterative procedure has been followed in order to size the blades.

6.3 Blade design method

In this section the overall blade design process is explained from a wider perspective. However, blade design algorithms will be carefully described in the next section. The fluid chart in Figure 6.4 depicts the global design scheme in this work.

Behold that words in blue denote the written MATLAB® script for each task. Some of them have already been explained, such as those related to 3D correction and extrapolation of airfoil data. Excel files where airfoil data is stored are represented in green. When *AF* appears means that one EXCEL file exists for each airfoil, since the described process is done for each airfoil, reaching to 4 different designs for each of the 2 chord distributions.

The designed algorithm can be mainly **divided in 2 important blocks**: the **design part**, from which element boundary radial position, and twist angle and chord length at those points are derived, and the **off-design analysis** of the wind turbine as a whole, from where

the values of the electric power, power coefficient, turbine thrust, shaft torque and root flap-wise bending moment are calculated for the working velocity range.

The whole implementation could be decomposed in 5 steps:

1) Design algorithm

The blade design algorithm is written in 1A.4 Main_DESIGN.m. In it, some of the already described input parameters are read and BEMT iteration method is applied through the different blade elements. An initial guess of the diameter is also needed.

During iterations the optimum chord and angle of attack (1A.5 Alpha_opt.m) are calculated. Airfoil data is loaded from Airfoil_Data.xlsx, which contains 2D data from XFOIL. Then, lift and drag coefficients are computed by the functions 1A.6 Clift_and_CDrag.m and 1A.7 CL_and_CD.m. The outputs of the design algorithm are the **radial position** of blade boundaries, and the **chord** and **twist** value at those sections.

2) Blade drawing

A MATLAB® script (1A.8 Drawing.m) has been written in order to draw the geometry of the blade alternatives. The blade design is shown by a 3D plot and also a polar view of the different airfoil sections. This script reads the output from the design algorithm (radial position, chord and twist) and the airfoil chord-scaled coordinates. For each radial position, airfoils are scaled to the chord value, and their pressure center (c/4) moved to coincide with the span axis of the blade. The section is then rotated a β angle. Coordinates of each section can be exported to SolidWorks in order to make a blade 3D model.

3) Intermediate step: Airfoil characteristic correction and extrapolation

As it has been explained in Chapter 3, airfoil data needs to be corrected to account for the 3D flow effects and then extrapolated to 360 degrees angle of attack. This is done by the scripts 1A.2 Snel_3D_corrections.m and 1A.3 Viterna_360_extrapolation.m which read the blade geometry and store the new polars in Airfoil_Data_*AF*_3D.xlsx and Airfoil_Data_*AF*_3D_extrapolated.xlsx respectively.

4) Off-design analysis: Turbine characteristic curves

Once the blade is physically designed, its performance in a wind turbine system over a wind velocity range is studied. This is done with the script 1A.9 Main_OFFDESIGN.m which perform BEMT iterations over different wind velocities. Airfoil data is this stage is loaded from Airfoil_Data_*AF*_3D_extrapolated.xlsx. Wind turbine characteristics are also needed as an input for the off-design analysis.

Once convergence is achieved axial and tangential loads in each blade element are calculated. When loads are known over the blade span the torque (tangential loads) and root flap-wise bending moment (RFWBM) contribution of each element is calculated.

By numerical integration over the blade span the turbine torque and total RFWBM are calculated. Also, by numerical integration of axial loads the thrust force that blades cause on the turbine can be calculated. From the shaft torque the mechanical power developed by the blades can be calculated. These values for a wind velocity range from the performance curves of a wind turbine and are the main output of this script. Once the power curves are obtained the AEP can be calculated.

5) Rotor diameter iteration

The first rotor diameter guess will lead to a power curve that will probably not reach or exceed the desired rating (10 kW). Therefore, the above describe process should be iterative, and, thus, a new value for the rotor diameter will be deduced from each trial.

The value of the new diameter guess is derived from the basic expression of produced power (Equation (6.5)), forced to be 10 kW and the supposition of invariant C_p curve with different rotor diameters.

Hence,

$$D_{new} = \sqrt{\frac{10000 \cdot 8}{\rho {U_{P,max}}^3 \pi C_p(with\ losses)_{U_{P,max}}}} \tag{6.14}$$

Where $U_{P,max}$ is the velocity at which maximum power is reached, and $C_p(with\ losses)_{U_{P,max}}$ the turbine overall efficiency at that velocity. A tolerance gap of $+ - 50$ W has been applied to the rated value of 10 kW in order to speed up the convergence and reduce computational time. As a first trial a diameter value of $D_i = \mathbf{7\ m}$ has been taken, similar to the ones of the same rating commercial wind turbines (usually between 6 m and 8 m).

The new diameter calculation is implemented inside the off-design analysis script.

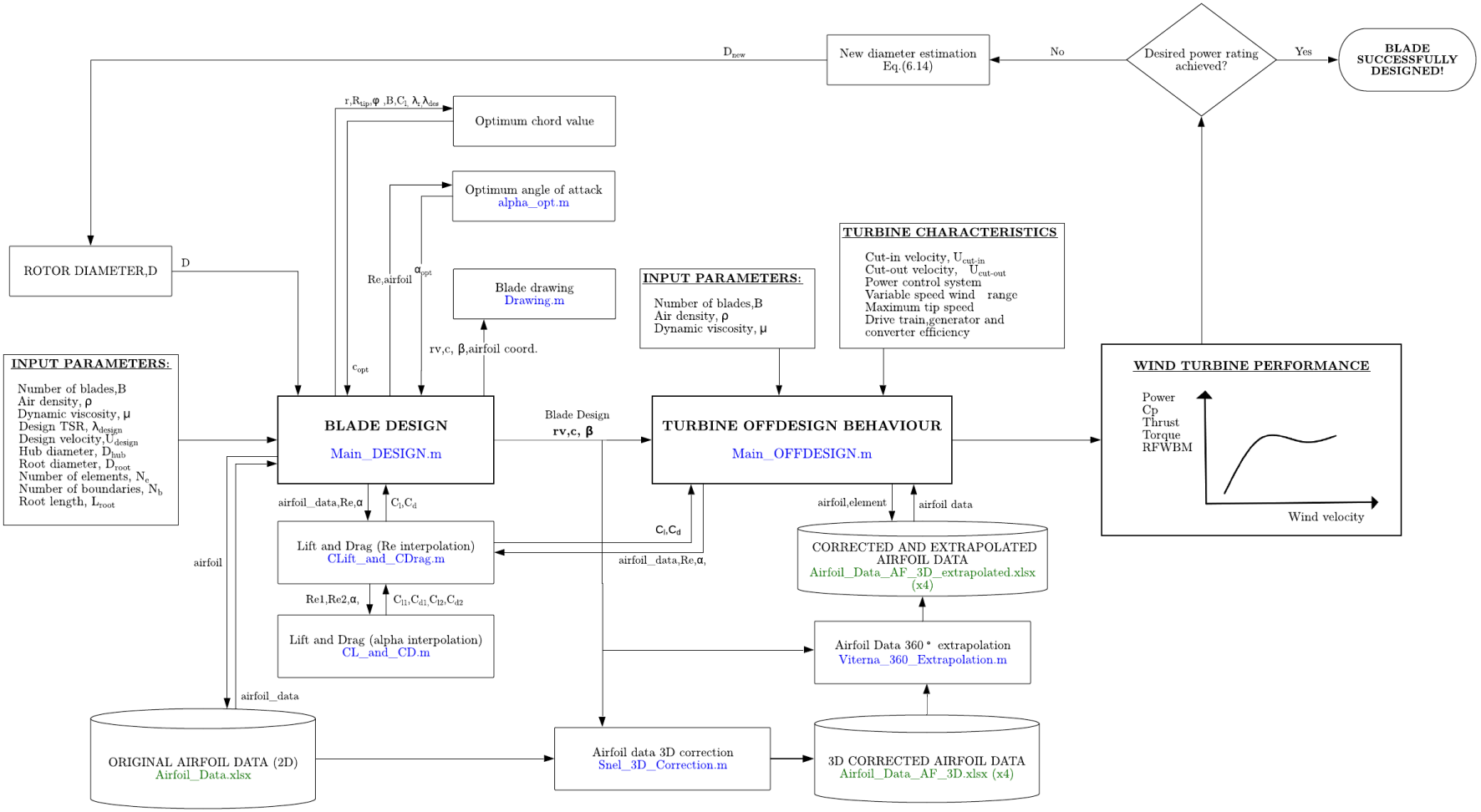


Figure 6.4 Schematic view of the overall blade design algorithm

6.4 Design algorithms

Both blade design point algorithm (on-design) and off-design turbine performance prediction algorithm use BEMT iterative equations, the first one in order to achieve a blade design with high aerodynamic efficiency and the other one to study the performance of the wind turbine over a wind velocity range from cut-in velocity to cut-out velocity.

The iterative process of each one is explained with more detail in this section.

6.4.1 On-design algorithm

The flowchart in Figure 6.5 describes the iterative procedure followed in the on-design algorithm. This process is followed for each airfoil layout. Below the flowchart each step will be briefly described. However, for further detail the written MATLAB® code could be consulted, which can be found in APPENDIX A (1A.4 Main_DESIGN.m).

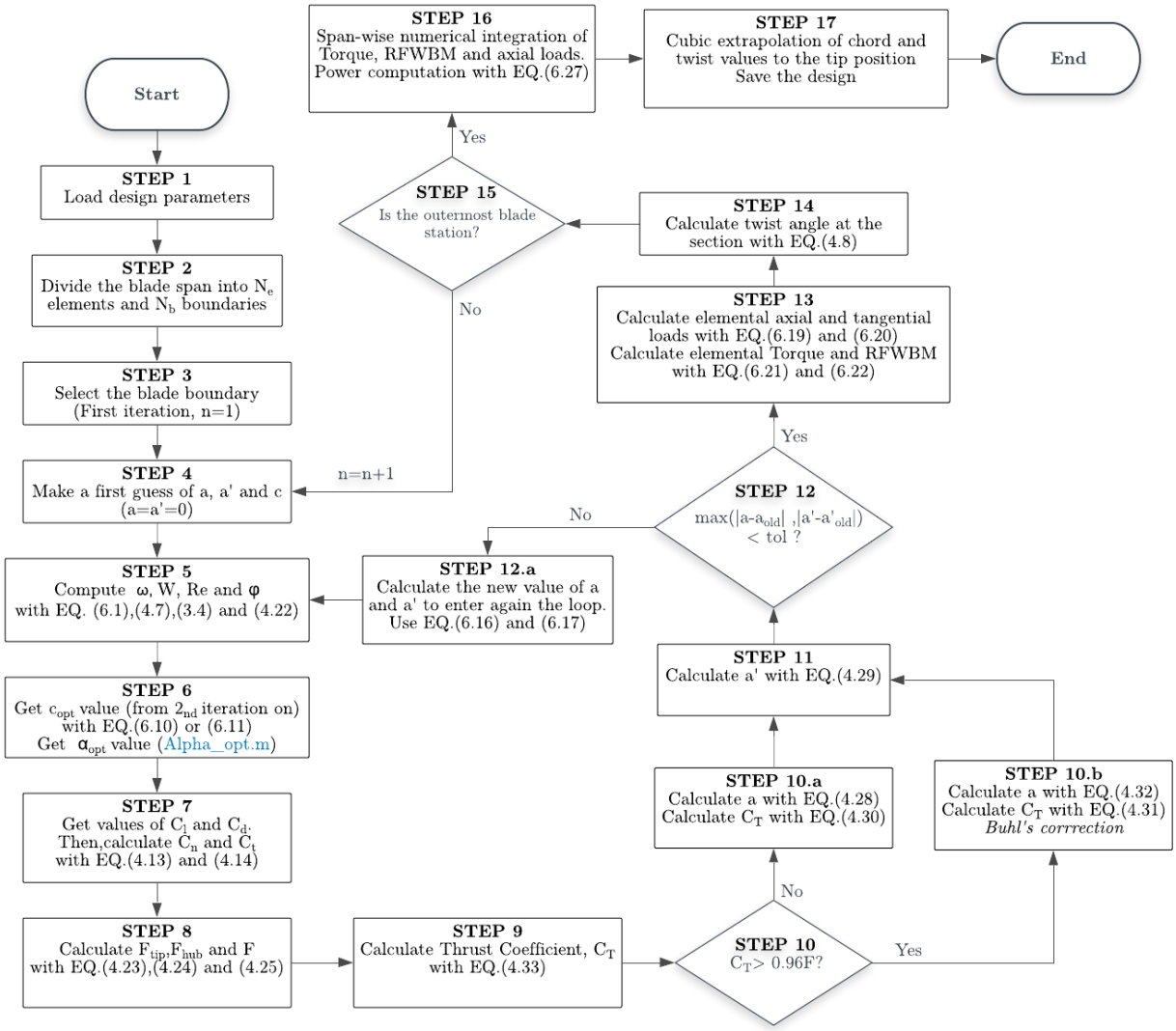


Figure 6.5 On-design algorithm flowchart

- **STEP 1:**

As a first step the necessary parameters are loaded. They have been described above in this chapter and are shown in Figure 6.4. The rotor diameter is also needed.

- **STEP 2:**

The blade span is divided into **20 segments of same length** (21 blade boundaries) using the MATLAB® command *linespace()*. As the chord and twist are to be designed also for the tip of the blade, this algorithm will apply BEMT equations on the blade **boundarie**s.

This leads to a numerical problem due to the fact that at the blade hub and tip the loss-factor $F = 0$ and therefore $a = 1$ and $a' = -1$ leading to a numerical non-convergence when calculating φ. In order to solve that and ensure convergence, the blade innermost (hub) and outermost (tip) boundaries have been moved 1 cm outward and inward respectively. When blade design is finished, chord and twist angle values are extrapolated to the blade tip position (1 cm outwards). The innermost part does not need extrapolation because the blade root is a cylindrical, constant chord part.

- **STEP 3:**

BEMT is applied for each blade section, starting from the innermost one ($n = 1$). The first 3 boundaries will belong to the root. They will be cylindrical with a constant chord of 0.2 m.

- **STEP 4:**

For the first iteration in each blade section a value of **0** will be assigned to the induction factors and a value of 1 m to the chord value (when section is not cylindrical).

- **STEP 5:**

Rotational speed ω, relative velocity W, Re number and inflow angle φ are calculated with Equations (6.1),(4.7), (3.4) and (4.22) respectively.

- **STEP 6:**

With the value of Re number, optimum value (max C_l/C_d) of angle of attack is found using the MATLAB® script 1A.5 Alpha_opt.m. Optimum value for the chord length is also calculated (from second iteration on) with Equation (6.10) or (6.11).

- **STEP 7:**

With α and Re the values of Cl and Cd are obtained by means of the MATLAB® scripts 1A.6 Clift_and_CDrag.m and 1A.7 CL_and_CD.m. Then, discompose them in axial and tangential axis with Equations (4.13) and (4.14). 2D XFOIL data is used, which is stored in Airfoil_Data.xlsx.

- **STEP 8:**

Calculate tip-loss factor, hub-loss factor and overall loss factor using Equations (4.23) (4.24) and (4.25) respectively.

- **STEP 9:**

Thrust coefficient is calculated from Blade Element Theory with Equation (4.33).

- **STEP 10:**

If C_T is higher than 0.96F, Equation (4.32) will be used to calculate the new value of a . If not, the standard BEMT is used to calculate the new value of the axial induction factor (Equation (4.28)).

- **STEP 11:**

Tangential induction factor is calculated with Equation (4.29).

- **STEP 12:**

At this point it is proved whether the difference between the new and the old values of the induction factors are lower than a certain tolerance, which has been set equal to 10^{-5}. If the difference is lower for both factors, convergence is achieved in the blade section. It will happen when:

$$\max(|a - a_{old}|, |a' - a'_{old}|) < \varepsilon = 10^{-5} \tag{6.15}$$

If not, new values for the induction factors are determined using a relaxation factor ω_{relax} and the previous values of the induction factors:

$$a_{new} = \omega_{relax} a + (1 - \omega_{relax}) a_{old} \; ; \quad 0 < \omega_{relax} < 1 \tag{6.16}$$

$$a'_{new} = \omega_{relax} a' + (1 - \omega_{relax}) a'_{old} \; ; \quad 0 < \omega_{relax} < 1 \tag{6.17}$$

A value of $\omega_{relax} = \mathbf{0.25}$ has been used. Also, a maximum number of iterations of **10000** has been set in order to prevent the algorithm from getting stuck.

- **STEP 13:**

Once convergence is achieved for a radial position the **axial and tangential loads** in the section can be obtained through BEMT equations. This loads are expressed per unit length [N/m]. The axial loads appear in the turbine shaft direction and contribute to a flap-wise bending of the blade and a thrust in the wind turbine. Tangential loads contribute to the rotation of the rotor and are perpendicular to axial forces and the blade axis at the same time. They generate a torque in the wind turbine shaft.

The Thrust and Torque per unit width of the element annulus are given by BEMT:

$$\frac{dT}{dr} = C_T \frac{1}{2} \rho {U_\infty}^2 (2\pi r) = C_T \rho {U_\infty}^2 \pi r \qquad \left[\frac{N}{m}\right] \tag{6.18}$$

$$\frac{dQ}{dr} = 4\pi \rho U_\infty F (1 - a) a' \omega r^3 \qquad \left[\frac{N \cdot m}{m}\right] \tag{6.19}$$

Hence, the axial and tangential loads (per unit length) for each blade element are calculated as:

$$\frac{dF_a}{dr} = \frac{dT}{dr}\frac{1}{B} = C_T \rho {U_\infty}^2 \pi r \frac{1}{B} \qquad \left[\frac{N}{m}\right] \tag{6.20}$$

$$\frac{dF_t}{dr} = \frac{dQ}{dr}\frac{1}{Br} = 4\pi\rho U_\infty F(1-a)a'\omega r^2 \frac{1}{B} \qquad \left[\frac{N}{m}\right] \tag{6.21}$$

Where $B = 3$ is the number of blades of the wind turbine.

So, the elemental torque developed by each blade section will be:

$$\frac{dQ_b}{dr} = \frac{dF_t}{dr} r = \frac{dQ}{dr}\frac{1}{B} \qquad \left[\frac{N \cdot m}{m}\right] \tag{6.22}$$

The Root Flap-Wise Bending Moment (RFWBM) contribution of each element is also calculated:

$$\frac{dM_{y,ROOT}}{dr} = \frac{dF_a}{dr}(r - R_{hub}) \qquad \left[\frac{N \cdot m}{m}\right] \tag{6.23}$$

Where $(r - R_{hub})$ is the lever arm from the axial load application point to the start of the blade root.

- **STEP 14:**

When convergence is achieved a, a' and φ are known. The value of the twist angle β that leads to an optimum angle of attack α_{opt} is calculated with Equation (4.8).

- **STEP 15:**

If the blade section for which convergence has been achieved is the outermost one, the blade span loop ends. If not, the convergence process is done for the next blade element boundary $(n = n + 1)$.

- **STEP 16:**

When the span-wise loop is finished, data and loads are calculated all over the blade sections. Now, the elemental loads and moments per unit of length are integrated along the blade span in order to calculate the turbine Thrust (due to blade exposition), the shaft Torque, the RFWBM:

$$Thrust = \int_{R_{hub}}^{R_{tip}} C_T \rho {U_\infty}^2 \pi r dr \tag{6.24}$$

$$Torque = \int_{R_{hub}}^{R_{tip}} 4\pi\rho U_\infty F(1-a)a'\omega r^3 dr \tag{6.25}$$

$$RFWBM = \int_{R_{hub}}^{R_{tip}} C_T \rho {U_\infty}^2 \pi r \frac{1}{B}(r - R_{hub}) dr \tag{6.26}$$

We have not continuous functions but discrete values along blade span. Therefore, **numerical trapezoidal integration** has been used to estimate the values of the integrals above. The MATLAB® function *trapz()* is used in order to perform the calculation. Trapezoidal integration assumes linear distribution of forces and moments between sections. The minor contribution of the 1 cm at the tip and hub will be also accounted.

Figure 6.6 shows a typical span-wise distribution of annular differential torque per unit length. The overall torque in the shaft is the sum between the green area and the red area (negative).

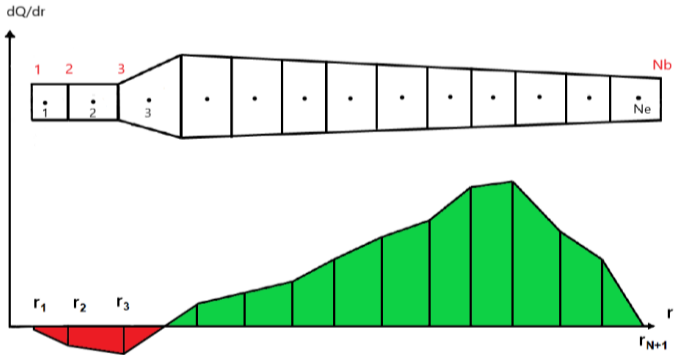


Figure 6.6 Span-wise variation of annular Torque [N · m/m]

The mechanical power developed by the blades will be:

$$Mechanical\ Power = Torque \cdot \omega \tag{6.27}$$

- **STEP 17:**

As explained in STEP 2, first and last boundaries were moved 1 cm outboard and inboard in order to achieve convergence and obtain the chord and twist values. When iterations along the blade span are finished, the chord and twist values on the tip position are obtained by **cubic extrapolation** of the chord and twist values along the blade span, as it is shown in Figure 6.7. Then, radial position, chord and twist are saved in .txt files.

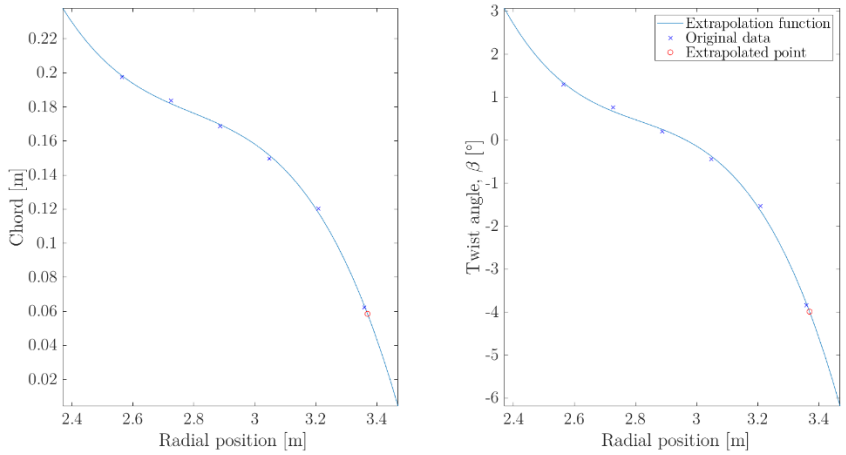


Figure 6.7 Outermost blade station chord and twist extrapolation (SG 6040 airfoil blade)

6.4.2 Off-design algorithm

The aim of the off-design analysis is to obtain the rotor performance over the wind turbine working velocity range. At this point the blade has to be understood as an element of a whole wind turbine. The followed BEMT iterative method is very similar to the previously explained one, with some subtle differences, that are explained below the flowchart in Figure 6.8.

The off-design algorithm performs calculations for several wind velocities, so another external loop needs to be used. Also, chord lengths and twist angles at radial positions are known, and after reaching convergence at a section, the angle of attack is the key factor determining the rotor performance. The algorithm is implemented in the MATLAB® script 1A.9 Main_OFFDESIGN.m.

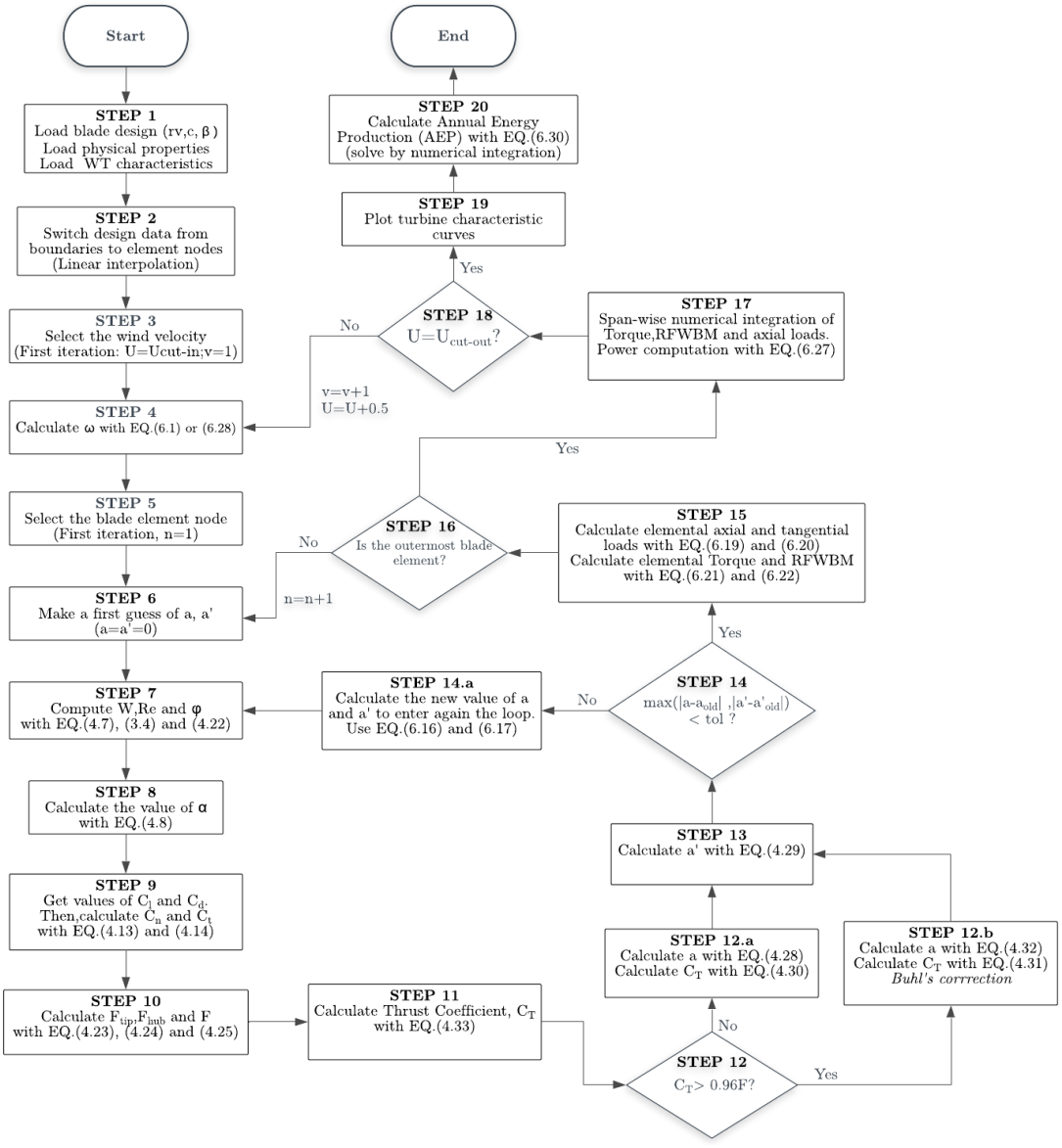


Figure 6.8 Off-design algorithm flowchart

The majority of steps are common to the on-design algorithm. However, some of them differ and are explained below. The common ones will not be re-described.

- **STEP 1:**

The design data (radial position vector, chord lengths and twist angles) which have been saved from the design algorithm are loaded. Additionally, the physical parameters of the air and the turbine characteristics are introduced as an input.

- **STEP 2:**

One of the differences between the two main algorithms of this work is that in the current one the blade element calculations are computed at the element nodes, this is, in the midpoint of two boundaries. Commercial software based on BEMT usually perform calculations at blade nodes and not at the boundaries in order to avoid non-convergence. So does, for example, the code software WT_Perf, developed by NREL [55].

Hence, values of chord and twist need to be found at node's radial positions. This is done by a simple linear interpolation.

- **STEP 3:**

The wind velocity is now chosen. A velocity vector that goes from U_{cut_in} to U_{cut_out} is created, with $\Delta U = 0.5\ m/s$. So, an outer loop will calculate rotor performance for each velocity.

- **STEP 4:**

The rotational speed needs to be established. If the wind velocity is between 3 m/s and 7.5 m/s, rotational speed will vary following the controller to maintain optimum TSR and must be calculated with Equation (6.1). If wind velocity is higher than 7.5, the machine will rotate at its maximum speed:

$$\omega = \omega_{max} = \frac{\lambda_{design} \cdot 7.5}{R_{tip}} = \frac{45}{R_{tip}} \tag{6.28}$$

- **STEP 5:**

The first node is selected to start computations. First two nodes belong to the cylindrical part of the blade. The 3rd one is in the middle of the transition between the cylindrical root and the first airfoil. For simplification, its aerodynamic coefficients have been set to $C_l = 0.7$ and $C_d = 0.8$ and are supposed constant (as the circular foils).

- **STEP 8:**

In this algorithm twist angles are known, so, when φ is computed, α is found by applying Equation (4.8).

- **STEP 17:**

In this algorithm the numerical integration has been performed employing the **rectangle rule,** which is simple and more convenient when BEMT iterations are solved at the element nodes. WT_Perf also uses this numeric integration method.

- **STEP 18:**

When blade iterations are finished for the cut-out velocity, the external loop ends. If the velocity is lower than the cut-out velocity, a new wind velocity (0.5 m/s higher) is selected and iterations follow.

- **STEP 19:**

When Mechanical Power, Turbine Thrust, Shaft Torque and Root Flap-Wise Bending Moment are calculated for each velocity they are plotted against wind velocity. The velocity at which rated (maximum) power is achieved is now discovered. The final electric power is given by:

$$Electric\ Power = Mechanical\ Power \cdot \eta \tag{6.29}$$

Where $\eta = 0.85$ is the efficiency of the drive train, generator and converter.

- **STEP 20:**

Knowing the power curve of the wind turbine and also the Weibull PDF at the wind turbine location the **Annual Energy Production (AEP)** can be calculated as:

$$AEP = 8760 \int_{U_{cut-in}}^{U_{cut-out}} P(u) f(u)\, du \tag{6.30}$$

Where:

$$f(u) = \frac{k}{c}\left(\frac{u}{c}\right)^{k-1} exp\left[-\left(\frac{u}{c}\right)^{k}\right] \tag{6.31}$$

is the Weibull PDF, k and c the shape and scale parameter respectively, P(u) the electric power, function of wind velocity and 8760 the number of hours in a year.

Practically, as we do not have the electric power as a continuous function of wind velocity, the integral above is solved numerically. Weibull PDF is evaluated for the previously defined velocity vector (3:0.5:20) and multiplied element by element with electric power vector. Then, numerical integration is performed using the trapezoidal rule (*trapz()* in MATLAB®).

Figure 6.9 shows graphically the above described procedure. The blue curve represents the electric power generation curve of the wind turbine and the red one the Weibull PDF at the selected location. Multiplying both we reach the Energy Probability Density curve (black).

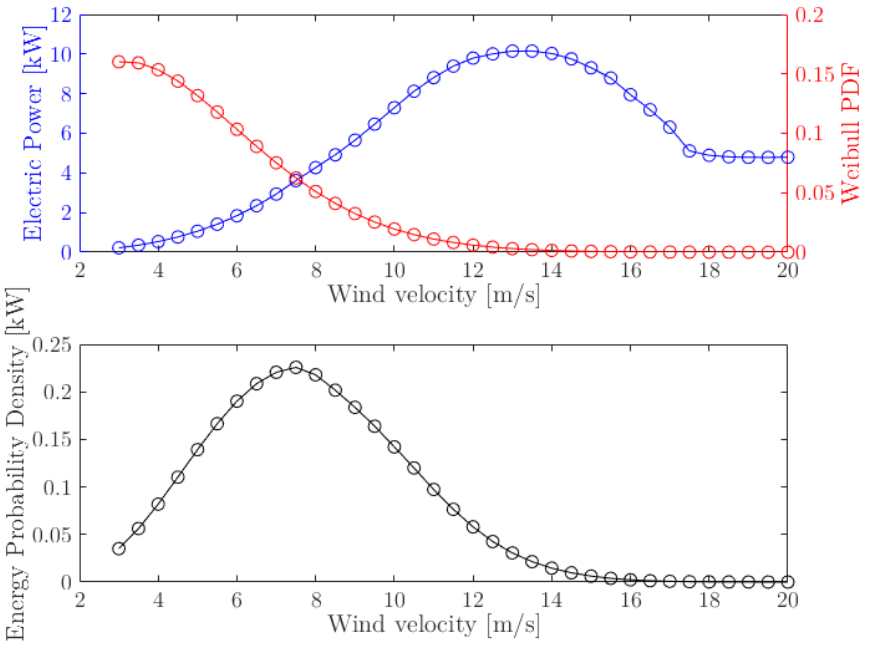


Figure 6.9 Energy Probability Density of a wind turbine in Aguilar de Codes

The area under the black curve represents the average power that the wind turbine with the designed blades will develop in a year basis. Multiplying this value by the number of hours in a year we obtain the annual production of electric energy in kWh.

The Annual Energy Production is a key factor for the purchaser (as it justifies or not the investment) and is often given by the manufacturers as function of the average wind velocity of the location.

Chapter 7

BLADE DESIGN: SIMULATIONS AND RESULTS

In this chapter all the **design results** derived from the procedure explained in Chapter 6 are shown. The objective of the chapter is to present the generated **8 blade alternatives** (4 airfoil layouts x 2 chord variation expressions) that lead to a wind turbine producing a peak electric power of 10 kW. As a stall-regulated machine usually do, the peak power will not be obtained at maximum efficiency, so the turbine rotors will need to be properly sized.

For each chord layout type a first design will be performed with a guessed rotor diameter. Performance curves will be shown for that first trial. Then, the rotor diameter will be iteratively adjusted until after a few iterations, the desired power rating is achieved. When the blade design finally fulfils the power rating, apart from the **performance curves** of the blades, the blade element data at the design point and the 3D view of the blades will also be plotted.

From the 8 blade alternatives 1 will be chosen in order to follow the work with. This is not a straightforward decision, because it involves a tradeoff between energy generation, blade size and aerodynamic loads that will need to be studied under proper economical valuation since loads involve the sizing and price of wind turbine components.

Finally, the resulting geometry of the chosen blade will be summarised and modelled in the CAD software SolidWorks 2018.

7.1 Blade design alternative generation

BEMT simulations are divided in 2 in function of the chord distribution used.

7.1.1 Blade design alternatives with chord layout A

In this case the chord length at each blade section is calculated using Equation (6.10).

7.1.1.a First rotor diameter guess

A first guess of the rotor diameter is set to **7 m.** With this as an input the design script (1A.4 Main_DESIGN.m) is run. When geometry at the blade sections is defined,

airfoil aerodynamic coefficients are corrected to 3D rotational effects (1A.2 Snel_3D_corrections.m) and extrapolated to 360 ° (1A.3 Viterna_360_extrapolation.m). When the processes are finished, the performance of each blade is simulated over the velocity range with the script (1A.9 Main_OFFDESIGN.m).

The chord and twist values for this first design guess are shown below:

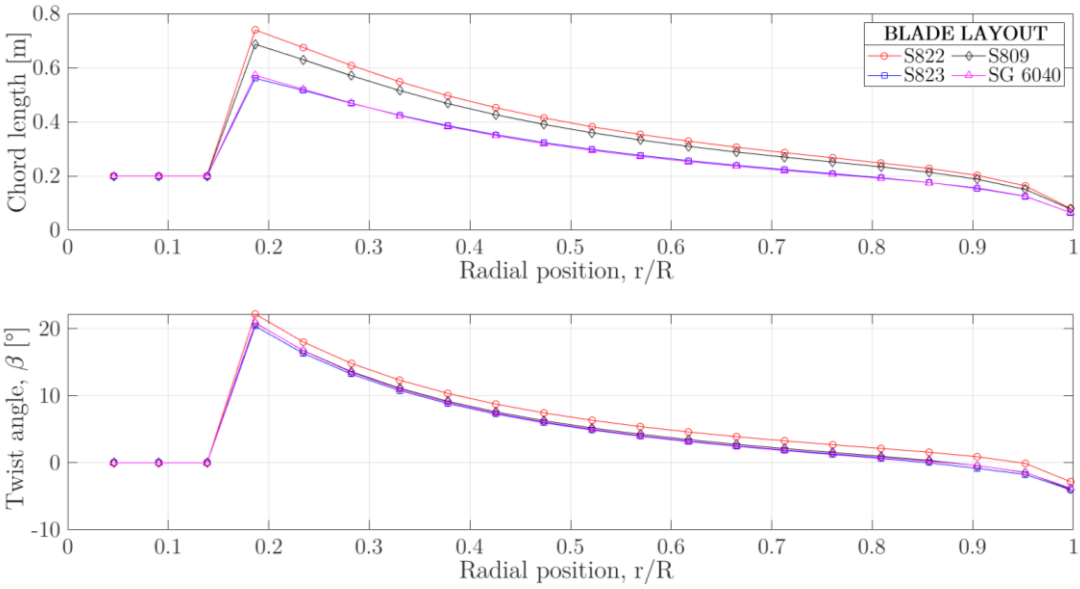


Figure 7.1 Chord length and twist angle distribution (D=7m, A chord dist.)

The wind turbine performance curves of the designed blades are shown in Figure 7.2.

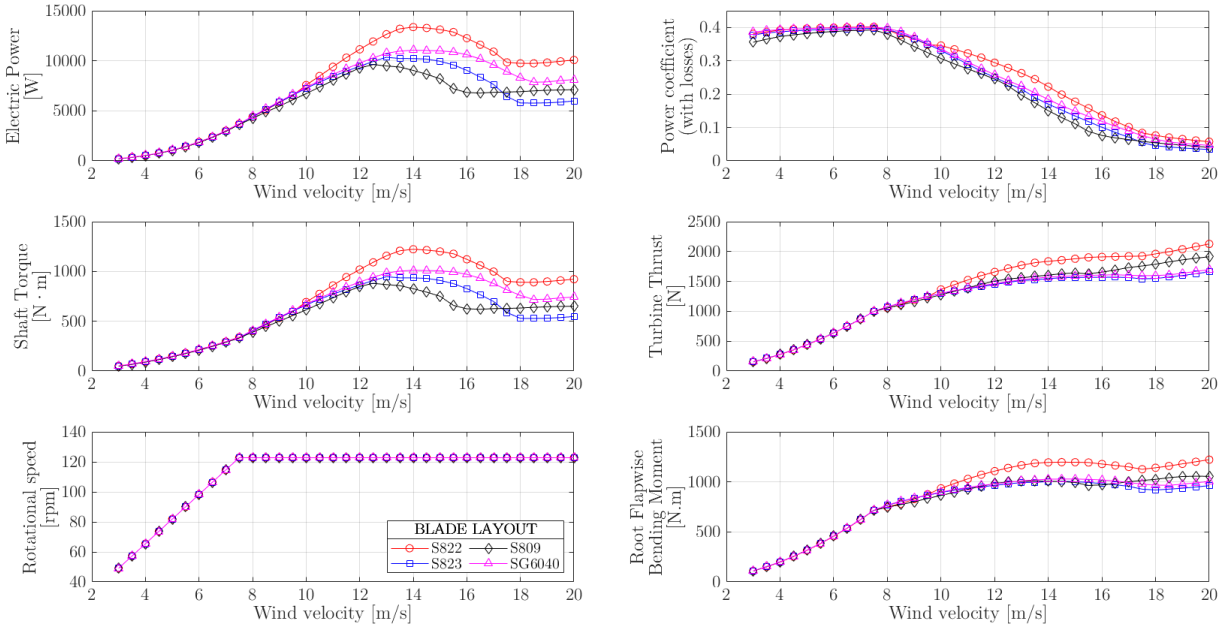


Figure 7.2 Wind turbine performance curves for a first size guess (D=7 m, A chord dist.)

The AEP in this case is:

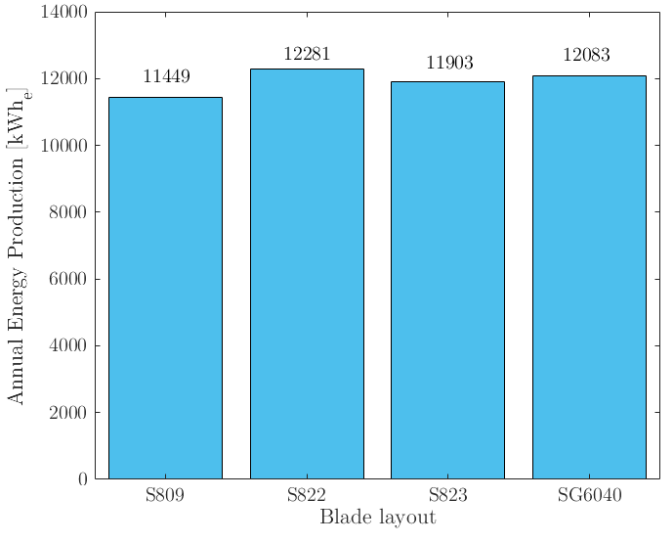


Figure 7.3 Annual Energy Production for a first size guess (D=7 m)

Analysing the results from this first simulation it is found that the blade with S822 airfoil layout reaches a higher peak power, followed by the SG 6040, S823 and S809 respectively. The same happens with the AEP.

Concerning the power coefficient, the S809 design shows the poorest values within the velocity range among all. The other 3 designs have a quite good performance curves. SG 6040 design has a slightly higher power coefficient that S823 all over the working range. Finally, the blade design with S822 airfoils outperforms the other when wind velocity is above 9.5 m/s.

With regard to resulting loads, S822 design suffers the highest values of turbine thrust force (due to blade exposition) and also the highest root flap-wise bending moment. Similar values are observed for the S823 and SG 6040 design. For velocities higher than 16 m/s the S809 design shows higher root bending moment and turbine thrust than these last 2.

Finally, in relation to blade shape S822 blade is the higher solidity rotor followed by the S809 one. Chord distribution for S823 and SG 6040 design is very similar, as Figure 7.1 depicts. A higher projected surface will be somehow related to the blade manufacturing cost.

All in all, there is no doubt that S809 design is the poorest one, while SG 6040 shows subtle higher performance than S823 for a similar rotor size and resulting loads. On the other hand, there is a **tradeoff** between S822 and SG 6040 designs, as S822 reaches more power peak, AEP and a higher efficiency above 9.5 m/s, but generates higher loads that will make the blade, tower and foundation design more expensive. Besides, the projected area of the S822 blade is much higher.

Nevertheless, discussion will be retaken after the sizing of the rotors.

7.1.1.b Rotor sizing: iterative process

The curves above represented the performance of blade designs created for a guessed rotor dimeter of 7 m. Logically none of the curves reaches an exact maximum power of 10 kW. In fact, the designs with S822, SG 6040 and S823 airfoils reach a higher power peak whereas the one with S809 remains below.

Now, the rotor diameter iteration explained in Section 6.3 is applied in order to size the blades and reach a maximum electric power output of 10 kW. Figure 7.4 illustrates the process that has been followed in order to reach convergence. Behold that as blade geometry changes, airfoil data 3D corrections need to be actualized in each iteration for maximum accuracy.

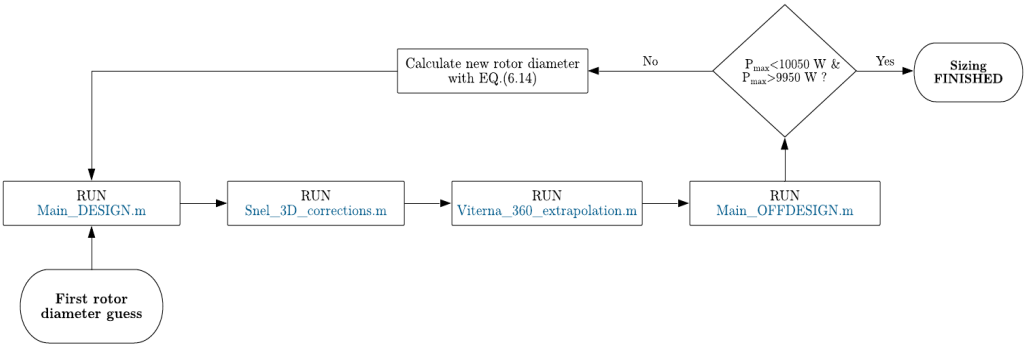


Figure 7.4 Rotor sizing iterative process in MATLAB®

Table 7.1 shows all the iteration results. Note that a tolerance of ±50 W around 10000 W (1%) is used to reduce the number of iterations, and also due to the sensibility of the rotor diameter on produced power.

Table 7.1 Rotor diameter sizing iteration process results (A)

	Iteration 1				
BLADE LAYOUT	D_0 [m]	P_{max} [W]	$U_{P,max}$ [m/s]	$CP_{P,max}$	D_{new} [m]
S822	7	13367	14	0.223	6.06
S823	7	10355	13	0.215	6.89
S809	7	9626	12.5	0.225	7.12
SG 6040	7	11061	14	0.184	6.67
	Iteration 2				
BLADE LAYOUT	D_0 [m]	P_{max} [W]	$U_{P,max}$ [m/s]	$CP_{P,max}$	D_{new} [m]
S822	6.06	8962	14	0.199	6.39
S823	6.89	9808	13	0.210	6.95
S809	**7.12**	**10025**	**12.5**	**0.227**	CONV
SG 6040	6.67	9763	13.5	0.2	6.74

	Iteration 3				
BLADE LAYOUT	D_0 [m]	P_{max} [W]	$U_{P,max}$ [m/s]	$CP_{P,max}$	D_{new} [m]
S822	6.39	10228	14	0.204	6.33
S823	6.95	10149	13	0.214	6.92
SG 6040	**6.74**	**10022**	**13.5**	**0.201**	**CONV**
	Iteration 4				
BLADE LAYOUT	D_0 [m]	P_{max} [W]	$U_{P,max}$ [m/s]	$CP_{P,max}$	D_{new} [m]
S822	**6.33**	**9976**	**13**	**0.254**	**CONV**
S823	**6.92**	**10025**	**13**	**0.213**	**CONV**

7.1.1.c Final results for A chord distribution

In this section the results for the already-sized blades are shown. The final rotor diameters are shown in Table 7.1 The results are divided in: blade element data on the design point, geometric representation of the blades (3D plots and section polar diagrams) and the off-design turbine characteristic curves.

- **Blade element data (on design point)**

All the curves below are an output of the MATLAB® script 1A.4 Main_DESIGN.m and show how the parameters involved in BEMT algorithm vary span-wise on the blade design point ($U = 7.5\frac{m}{s}; \lambda = 6$).

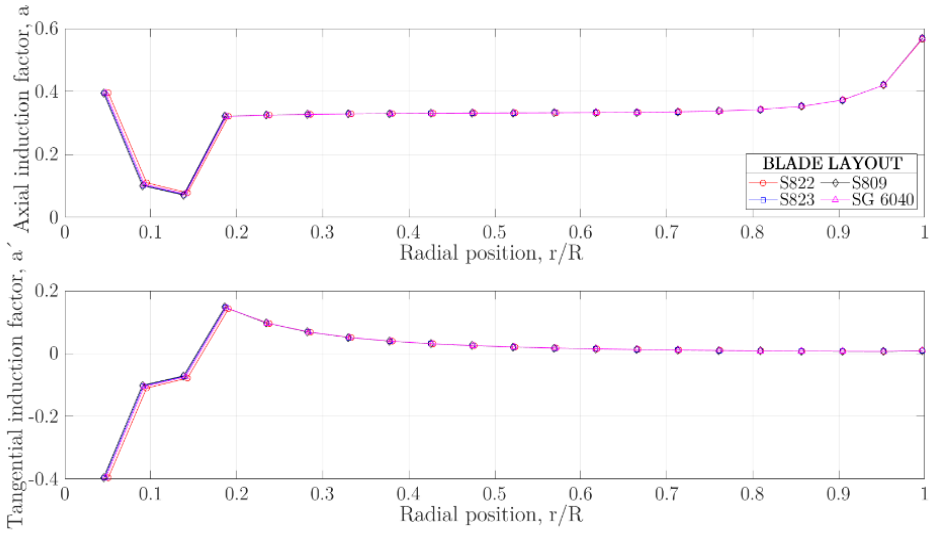


Figure 7.5 Span-wise variation of induction factors (local to the blade) (A)

Figure 7.5 shows how the axial induction factors increase near the tip and the hub, due to the existence of tip and hub vortices. Near the root, values of axial induction factor are lowers due to the lower chord value of cylindrical root. Then, the value is

nearly 1/3 (obtainable by hand, supposing too low drag, loss factor equal to 1 and substituting Equation (6.10) in Equation (4.28)). The high glide-ratio is the reason why curves are nearly equal for all the blade designs. On the other hand, tangential induction factor remains negative in the cylindrical part (due to the fact that it has no lift coefficient). Then, for the part of the blade with airfoils it is positive and tends to 0 when moving radially towards the tip.

The following figure shows the span-wise variation of chord length and twist angle.

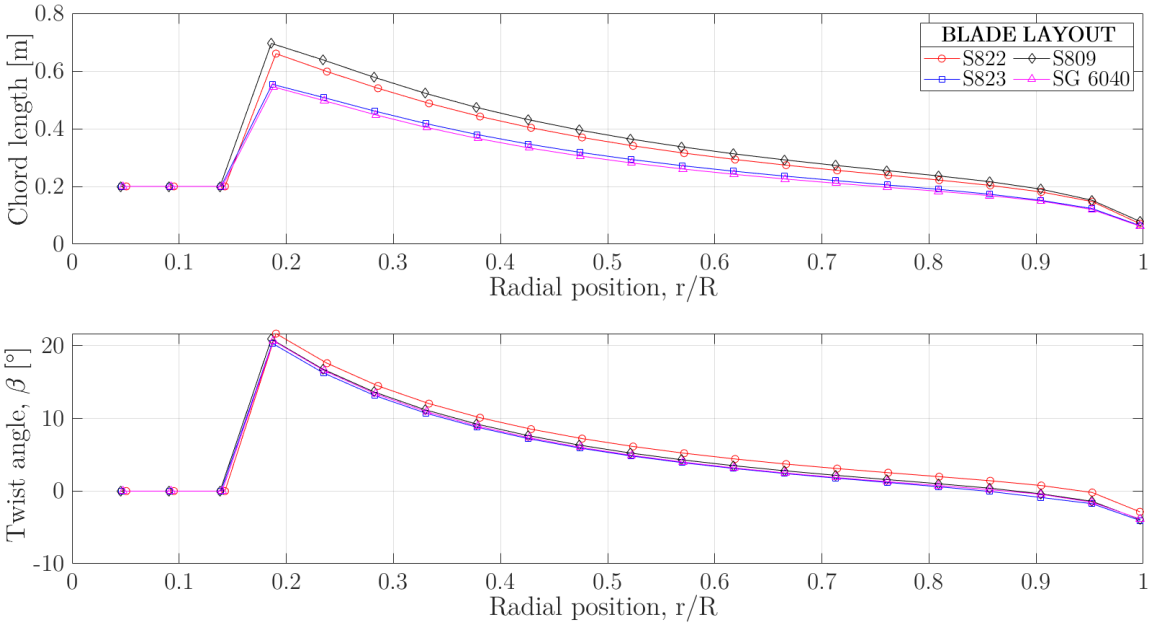


Figure 7.6 Span-wise chord and twist distributions (A)

The highest chord length is achieved in the first airfoil section by the S809 design, closely followed by the S822's one. The other 2 present lower values. Anyway, chord length deceases when radial position increases.

When radial position increases the tangential component of the wind velocity gains more strength than the axial part, which is constant, and therefore inflow angle decreases. Hence, twist angle needs to be adapted to maintain the angle of attack in its optimal value. This is the reason why the twist angle decreases radially.

In fact, the optimum value of angle of attack depends upon the Re number, and this one, on the relative velocity in the section and the chord length. Re numbers appears to be nearly constant in the major part of the blade span. Even so, near the tip the chord reduction is stronger than the increase in relative velocity, and Re number drops. It was argued in Chapter 3 how optimum angles of attack were higher for lower Re numbers. Thus, optimum angles of attack are higher near the tip (see Figure 7.7).

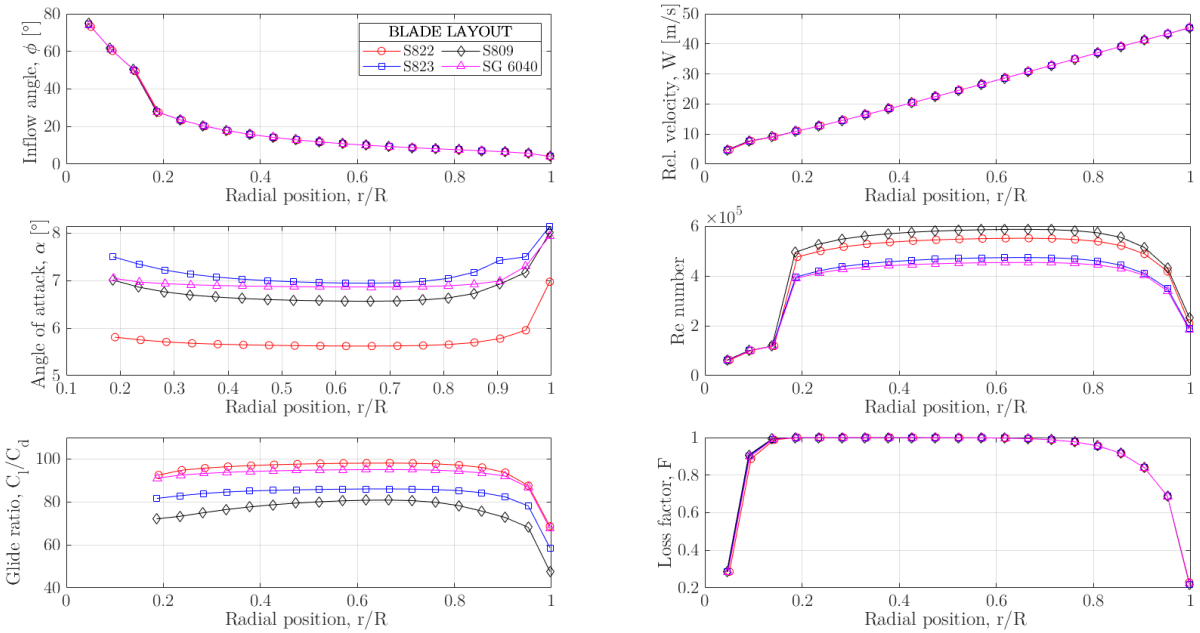


Figure 7.7 Span-wise variation of several BEMT algorithm parameters (A)

Regarding the glide-ratio distribution, S822 and SG 6040 airfoil blades take the lead, reaching values of almost 100. The poorest values are obtained by S809 airfoil blade, which is, clearly, the less aerodynamically efficient choice. The glide-ratio falls when approaching the tip as a consequence of lower Re numbers, for which, as seen in Chapter 3, lower glide ratios are obtained (see Figure 3.6, for example).

Figure 7.8 shows how lift coefficient remains almost constant over the blade span. However, it is the increase in drag coefficient which generates the drop in the glide-ratio. Thrust coefficient is lower in the circular sections (lower chord), then it remains constant and finally near the tip it falls, due to the low annular averaged axial induction factor.

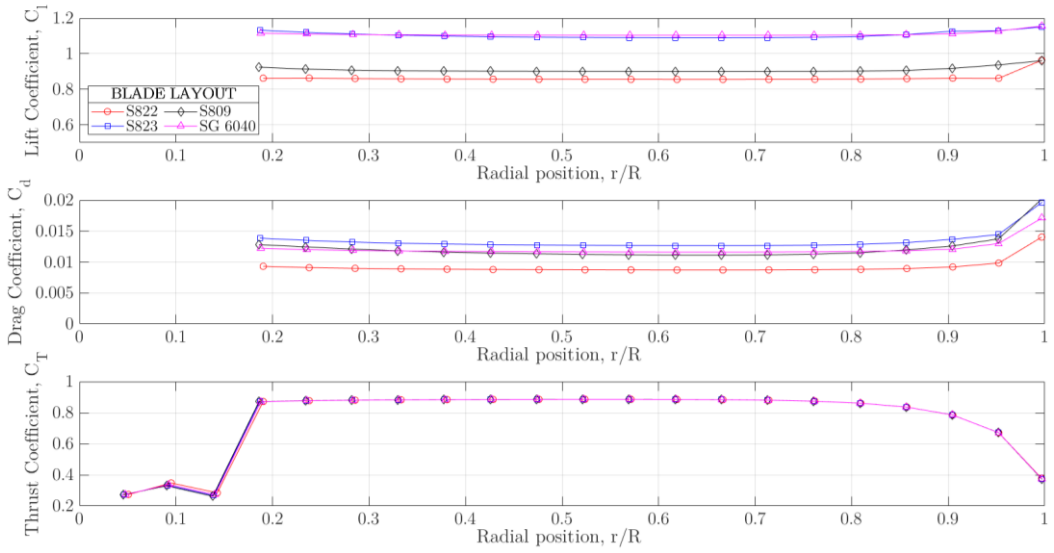


Figure 7.8 Lift, Drag and Thrust coefficient variation over the blade span (A)

Figure 7.9 shows the axial and tangential loads distribution over the blade span. **Two main differences** need to be remarked: on the one hand, unless near the tip that both drop, the distribution of axial loads is linear with radius whereas tangential loads are almost constant. Therefore, the root flap-wise bending moment curve has a quadratic shape, and the torque curve is linear with respect to the radius. On the other hand, the maximum values of the axial loads are approximately 8 times higher than the tangential ones.

The numeric integral of the 3rd row curves, which account for the elemental thrust and torque generated by the 3 blades, give the turbine overall thrust force due to the blades and the total torque on the shaft. As a peculiarity, in the root section the tangential loads are negative, since there is no lift, and air acts an aerodynamic brake for the blade. This is traduced in a power loss, which is, as it can be seen, minimal.

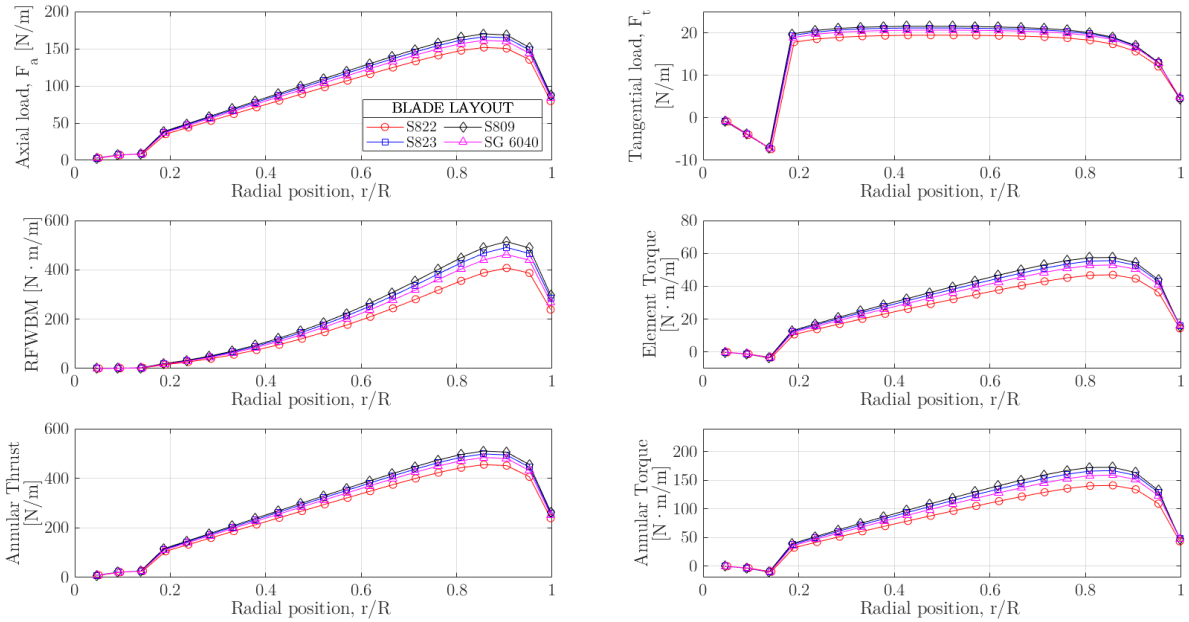


Figure 7.9 Aerodynamic forces and moments over the blade span (A)

The results regarding the whole wind turbine are shown in the following table

Table 7.2 Power, Torque Cp, Thrust, and RFWBM in the design point (A)

BLADE LAYOUT	Mechanical Power [W]	Electric Power [W]	Thrust [N]	Torque [N · m]	RFWBM [N · m]	C_P (w/o loss)	C_P (with loss)
S822	3551	3018	816	250	525	0.469	0.399
S823	4216	3584	976	324	693	0.467	0.397
S809	4422	3759	1033	350	754	0.462	0.393
SG 6040	4031	3426	926	302	639	0.471	0.400

- **Geometric representation of the blades**

The 3D shape and polar section view (from the right side of the blade) of the constituting airfoils are illustrated below.

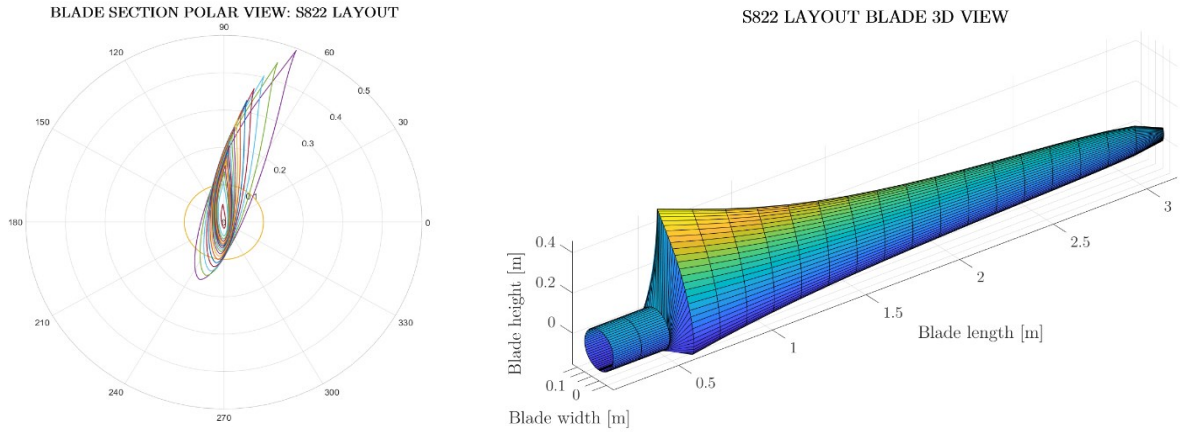


Figure 7.10 S822 airfoil blade design: 3D and polar view (A)

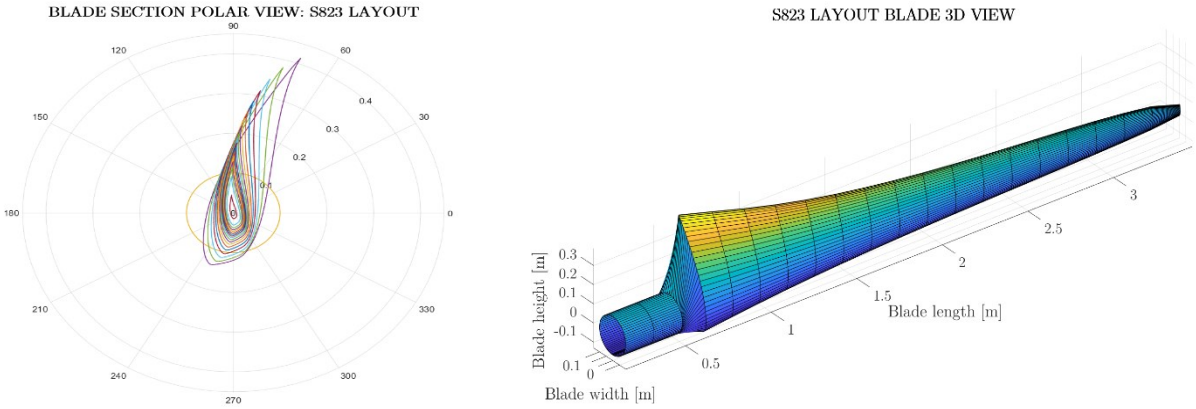


Figure 7.11 S823 airfoil blade design: 3D and polar view (A)

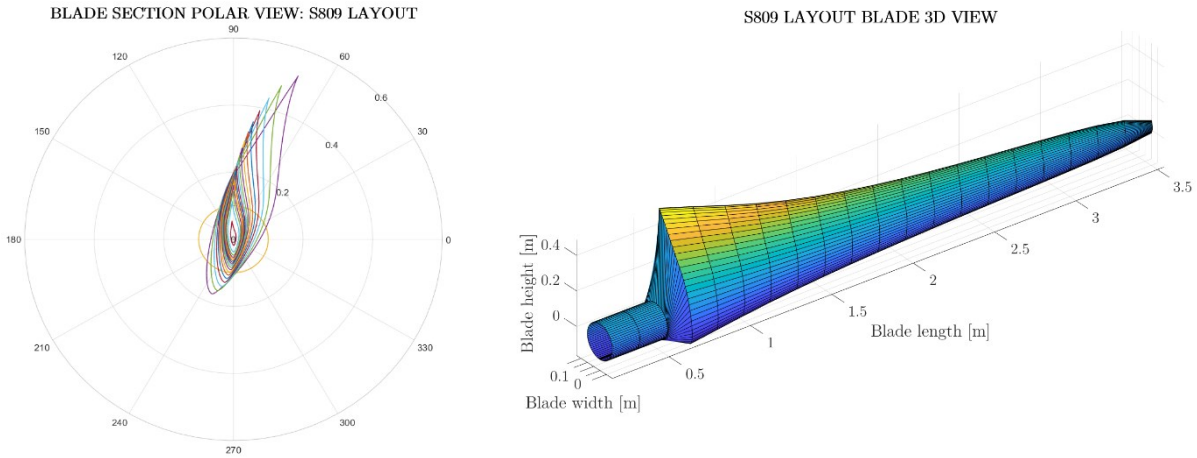


Figure 7.12 S809 airfoil blade design: 3D and polar view (A)

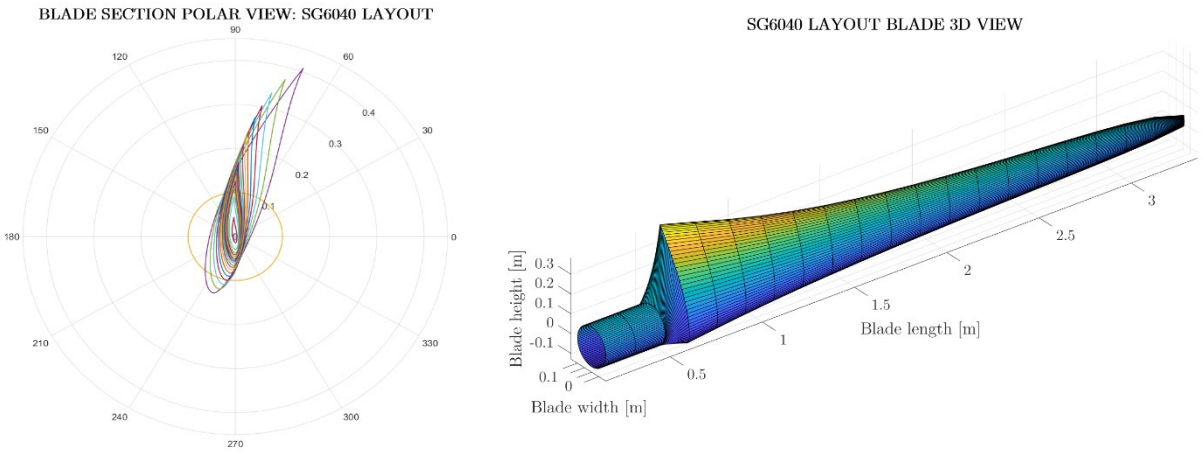


Figure 7.13 SG 6040 airfoil blade design: 3D and polar view (A)

The blade with S809 airfoil is the longest and highest blade; S822 is the shortest one but it is the second in height. S823 and SG 6040 blades are slenderer, being the SG 6040 10 cm shorter.

- **Off-design behaviour of the wind turbines**

Figure 7.14 shows the off-design behaviour of the designed blades "working" in the full wind turbine model that has been defined in the work. Curves show the electric generated power, power coefficient, thrust, torque, blade root flap-wise bending moment and rotational speed.

Regarding the power curve, all the blade designs **reach the rating of 10 kW**. The iteration process has been successful because C_p vs U curves are not highly size dependent.

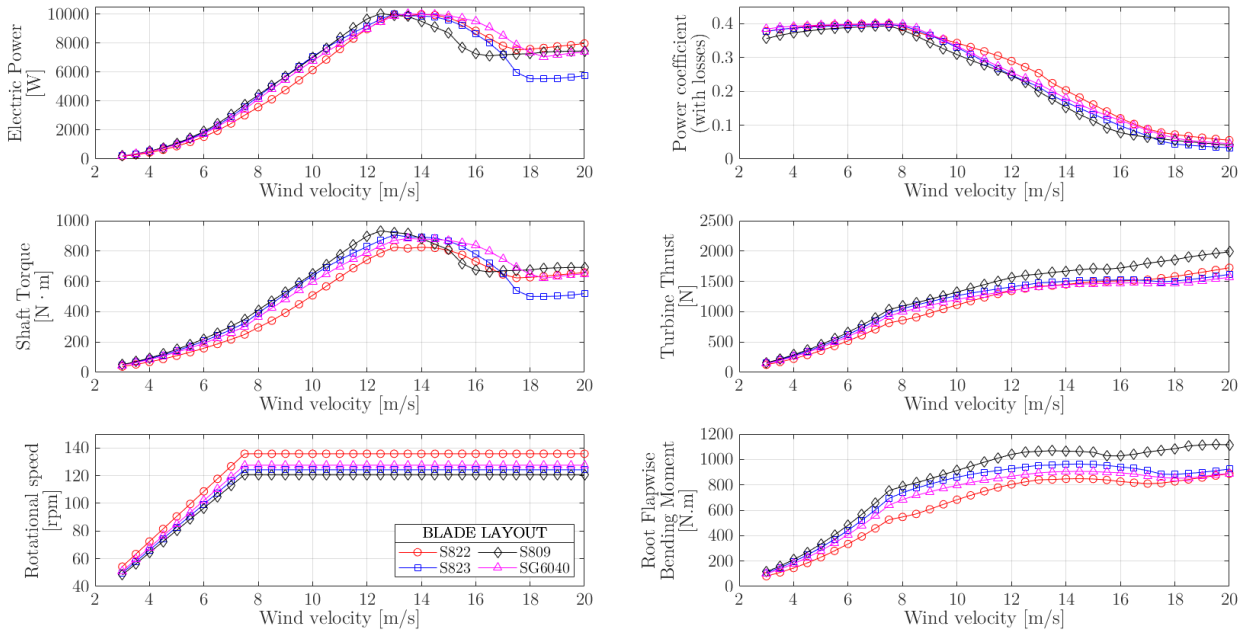


Figure 7.14 Wind turbine performance curves for the sized blades (A)

When all the blades had the same length the highest loads were experimented with the S822 design. However, now that design needs a lower diameter to reach the power rating, and experiences lower loads that others generally. In contrast, the S809 airfoil blade design requires a higher rotor diameter to reach the desired power and the rotor experiences high loads. Also, the root flap-wise bending moment is higher.

In order to decide the most suitable design with the A chord variation model, another 2 parameters have been taken into consideration. Figure 7.15 depicts the AEP that would be yielded with each design and also a parameter called the folded projected area. This represents the frontal projected area of the wind turbine if we untwisted the twisted blades and gives an idea of the overall dimension. The manufacturing cost could be somehow related to the size.

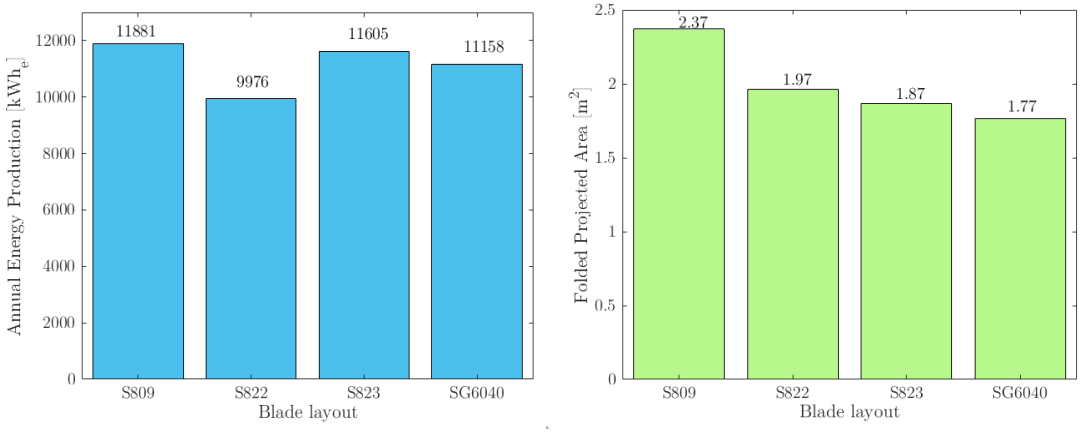


Figure 7.15 AEP and blade folded projected area for the blade design alternatives (A)

- S809 blade design generates slightly more energy than S823 but is much bigger, the power coefficient is lower and the thrust and RFWBM are the highest. S809 reaches the peak power for a lower wind velocity than the others.
- Between S823 and SG 6040, it is not a clear choice: S823 generates more energy, but SG 6040 has bit lower projected area, a subtle higher efficiency all over the wind velocity range and slightly lower maximum loads: torque, root bending moment and turbine thrust.
- S822 design is the shortest blade, but has a higher projected area than the previous 2 designs due to its high chord lengths. Its energy generation is much lower than the others and surprisingly the thrust forces on the rotor are higher than for S823 and SG 6040 designs. It works under the maximum variable velocity range, which can difficult the design due to vibration and higher centrifugal forces. Even though above 9.5 m/s it has a higher power coefficient, it will not be the chosen alternative.

Finally, the preferred design is: **SG 6040 AIRFOIL BLADE**

7.1.2 Blade design alternatives with chord layout B

In the second case the chord length at each blade section is calculated using Equation (6.11). The followed process is identical to the previous one, so it will be illustrated more summarised.

The differences found with the results in the previous pages will be described and justified.

7.1.2.a First rotor diameter guess

As for the previous chord distribution, first guess of the rotor diameter is set to **7 m.**

The chord and twist values obtained from the design MATLAB® script (1A.4 Main_DESIGN.m) in this first design guess are shown below:

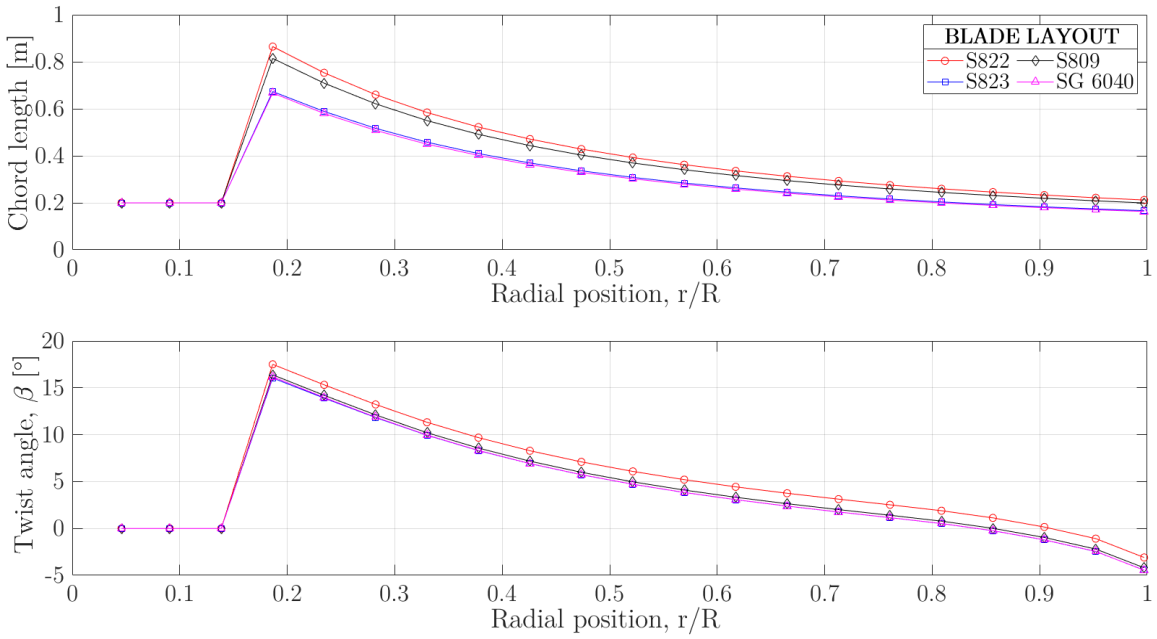


Figure 7.16 Chord length and twist angle distribution (D=7m, B chord dist.)

The figure shows a different chord span-wise variation than in the first attempt with A chord distribution. The main differences are the higher values of chord length on the first airfoil section and a more linear distribution close to the tip. An important fact is that the chord length at the blade tip is much higher than in the previous case was. As in the previous case S822 and S809 airfoil designs require higher chords and therefore, the rotor has a higher solidity. Chord distribution for S823 and SG 6040 design is very similar, as it is shown.

The blade twisting in the first sections of the blade is a bit lower than for the A chord distribution first try. Higher twist angles are required for S822 airfoil blade design.

The wind turbine performance curves of the current blades are shown in Figure 7.17.

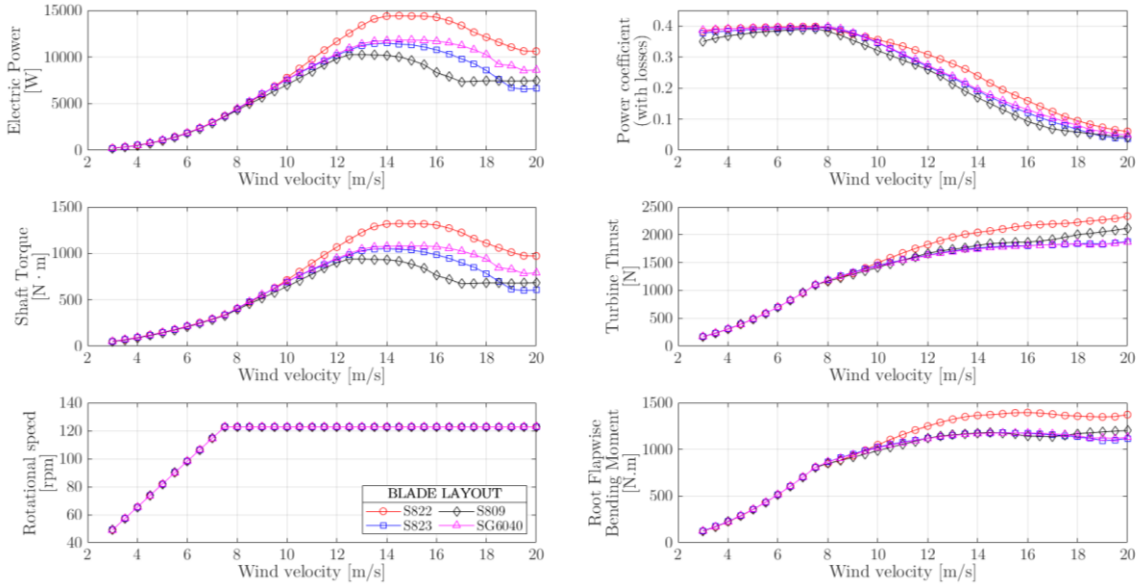


Figure 7.17 Wind turbine performance curves for a first size guess (D=7 m, B chord dist.)

The AEP in this case is:

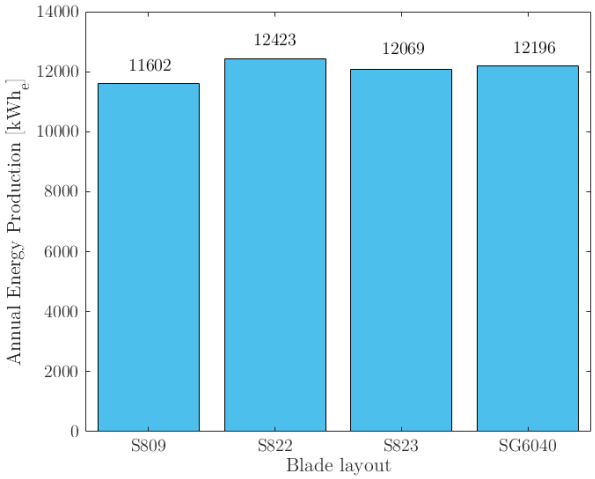


Figure 7.18 Annual Energy Production for a first size guess (D=7 m, B chord dist.)

Analysing the results from this simulation it is found that the blade with S822 airfoil layout reaches a higher peak power, followed by the SG 6040, S823 and S809 respectively. The same happens with the AEP. Similar happened with A chord distribution, although in the current case maximum peak powers are higher. The AEP, however, is quite similar.

In this simulation the S809 design also shows the poorest C_P values within the velocity range among all. The other 3 designs have a quite good performance curves. The blade design with S822 airfoils outperforms the others when wind velocity is above 9.5 m/s. There is no much contrast with the A chord distribution first guess results unless that in the variable speed range the power coefficient is more linear for S822, S823 and SG6040 blade designs.

Regarding loads, as in the first case S822 design suffers the highest values of turbine thrust force root flap-wise bending moment. S823 and SG 6040 airfoils present similar values. For velocities higher than 16 m/s the S809 design shows higher root bending moment and turbine thrust than these last 2. In general, it is noticed that turbine thrust and RFWBM are higher than with the other chord distribution, and it is mostly due to the higher chords in these last blade designs.

In conclusion, the S809 design is the poorest one, while SG 6040 shows subtle higher performance than S823 for a similar rotor size and resulting loads. S822 airfoil based design is the one that generates more energy for the same blade length but has also the highest loads and solidity.

After giving the rotors the required size the discussion about the preferred design alternative will be retaken.

7.1.2.b Rotor sizing: iterative process

With the guessed rotor diameter value of 7 m designs with S822, SG 6040 and S823 airfoils reach a higher power peak than the desired 10 kW, whereas the one with S809 is very close to the limit. Now, the rotor diameter iteration process is applied as explained in Section 6.3. MATLAB® scripts are run as detailed in Figure 7.4 and the results are shown in Table 7.3.

Table 7.3 Rotor diameter sizing iteration process results (B)

	Iteration 1				
BLADE LAYOUT	D_0 [m]	P_{max} [W]	$U_{P,max}$ [m/s]	$CP_{P,max}$	D_{new} [m]
S822	7	14428	14.5	0.216	5.83
S823	7	11487	14	0.191	6.53
S809	7	10264	13	0.213	6.91
SG 6040	7	11801	14.5	0.177	6.44
	Iteration 2				
BLADE LAYOUT	D_0 [m]	P_{max} [W]	$U_{P,max}$ [m/s]	$CP_{P,max}$	D_{new} [m]
S822	5.83	8651	13.5	0.232	6.27
S823	6.53	9014	13	0.216	6.88
S809	**6.91**	**9957**	**13**	**0.212**	**CONV**
SG 6040	6.44	9599	14.5	0.170	6.57
	Iteration 3				
BLADE LAYOUT	D_0 [m]	P_{max} [W]	$U_{P,max}$ [m/s]	$CP_{P,max}$	D_{new} [m]
S822	6.27	10483	14	0.218	6.12
S823	6.88	10869	14	0.187	6.60
SG 6040	6.57	10102	14.5	0.172	6.54
	Iteration 4				
BLADE LAYOUT	D_0 [m]	P_{max} [W]	$U_{P,max}$ [m/s]	$CP_{P,max}$	D_{new} [m]

S822	6.12	9772	13.5	0.237	6.19
S823	6.60	9312	13	0.214	6.80
SG 6040	**6.54**	**9985**	**14.5**	**0.171**	**CONV**
	Iteration 5				
BLADE LAYOUT	**D_0 [m]**	**P_{max} [W]**	**$U_{P,max}$ [m/s]**	**$CP_{P,max}$**	**D_{new} [m]**
S822	6.19	10058	13.5	0.239	6.18
S823	6.80	10425	14	0.184	6.66
	Iteration 6				
BLADE LAYOUT	**D_0 [m]**	**P_{max} [W]**	**$U_{P,max}$ [m/s]**	**$CP_{P,max}$**	**D_{new} [m]**
S822	**6.18**	**10016**	**13.5**	**0.239**	**CONV**
S823	6.66	9740	13.5	0.2	6.75
	Iteration 7				
BLADE LAYOUT	**D_0 [m]**	**P_{max} [W]**	**$U_{P,max}$ [m/s]**	**$CP_{P,max}$**	**D_{new} [m]**
S823	6.75	10155	13	0.225	6.72
	Iteration 8				
BLADE LAYOUT	**D_0 [m]**	**P_{max} [W]**	**$U_{P,max}$ [m/s]**	**$CP_{P,max}$**	**D_{new} [m]**
S823	**6.73**	**10007**	**14**	**0.181**	**CONV**

7.1.2.c Final results for B chord distribution

In this section the results for the already-sized blades are illustrated. The final rotor diameters are shown in Table 7.3. The results for the ultimate designs with B chord distribution are shown in the same way that in the previous section.

- Blade element data (on design point)

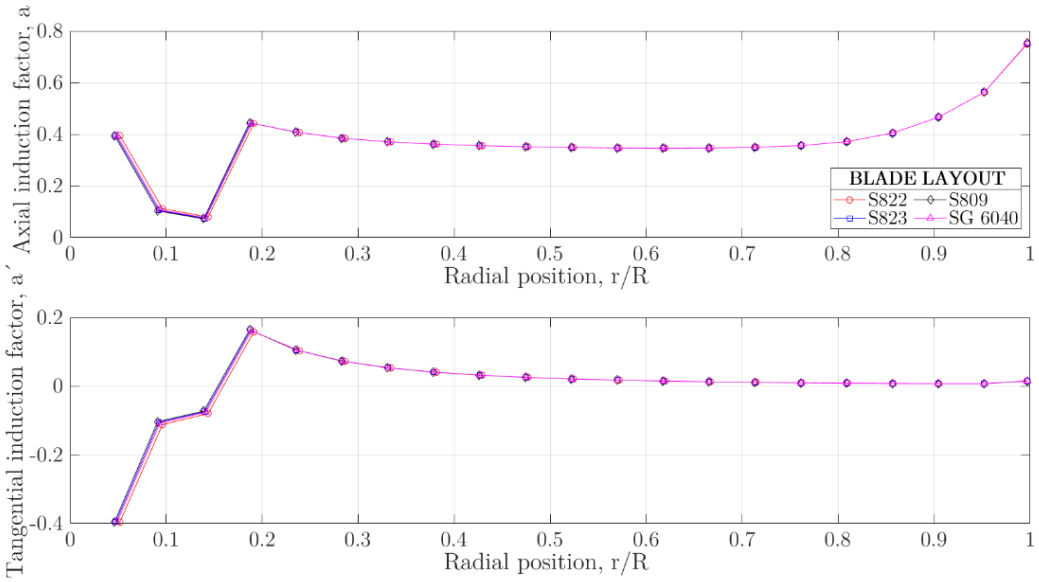


Figure 7.19 Span-wise variation of induction factors (local to the blade) (B)

Figure 7.19 shows how the axial induction factors increase near the tip and the hub, due to the existence of tip and hub vortices. The only difference with the analogous

results for chord distribution A is that near the tip higher values of axial induction factors are obtained. This is due to the much higher solidity at the blade tip in this case, which happens due to the higher chord lengths at that point.

The next figure shows the span-wise variation of chord length and twist angle (the parameters that define geometrically the blade).

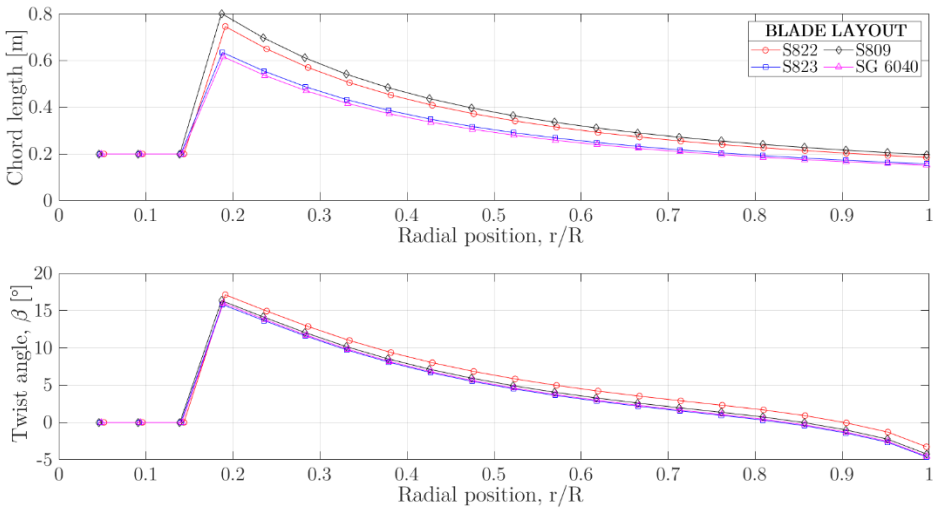


Figure 7.20 Span-wise chord and twist distributions (B)

The highest chord length is achieved in the first airfoil section by the S809 design, closely followed by the S822's one. The other designs 2 present lower values. Anyway, chord length decreases when radial position increases in a more linear way at the outer part of the blade than at the inner part. The peculiarity of this design is that on the tip of the blade the chord does not become smaller as in A chord distribution case.

Twist angles are adapted to make appear in the section an optimum angle of attack that maximises the glide ratio. Twist angles are, in the inner part of the blade 5-6 degrees lower that in the previous designs.

Figure 7.21 shows how the parameters relative to the BEMT algorithm vary with the blade span on the design point. Re number shows a linear pattern over the blade span. This happens because the linear ascendant relative velocity contribution to Re number is equalized with the quite linear descendant chord length. Having a constant Re number over the blade span leads to constant optimum angles of attack and constant values of the glide-ratio. The S822 works optimally at quite low angles of attack but its glide-ratio is the highest over all the airfoils, reaching a value slightly lower than 100. The less aerodynamic airfoil is again the S809 one, with glide-rations near 85. The loss-factor, as in the previous case, drops to 0 when approaching the blade tip and hub, whereas its value is 1 in the mid-span zone.

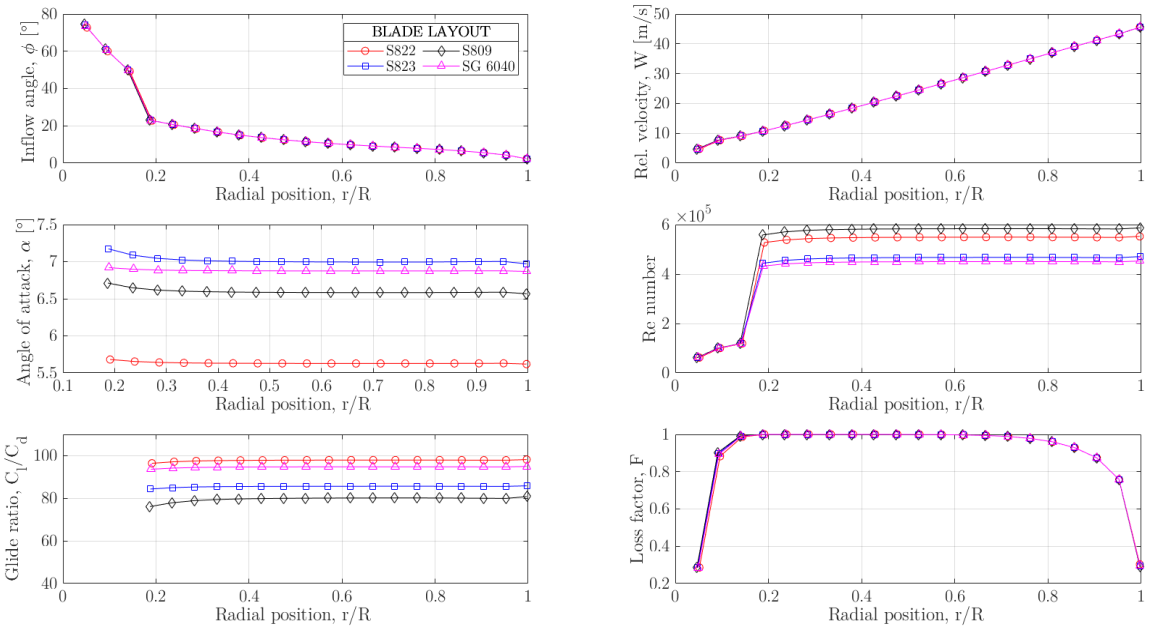


Figure 7.21 Span-wise variation of several BEMT algorithm parameters (B)

Since Re number and angle of attack are constant over the blade span, so are the lift and drag coefficients (see Figure 7.22). With respect to the thrust coefficient, it is lower in the circular sections (lower chord), then increases due to the higher chord in the first airfoil sections, entering into the high loaded turbine mode (so Buhl's correction is applied). Then it remains constant and at nearly 0.9 until the tip of the blade. This time thrust coefficient does not fall near the tip due to its higher chord (it works in the advanced brake state, explained in Section 4.4.4).

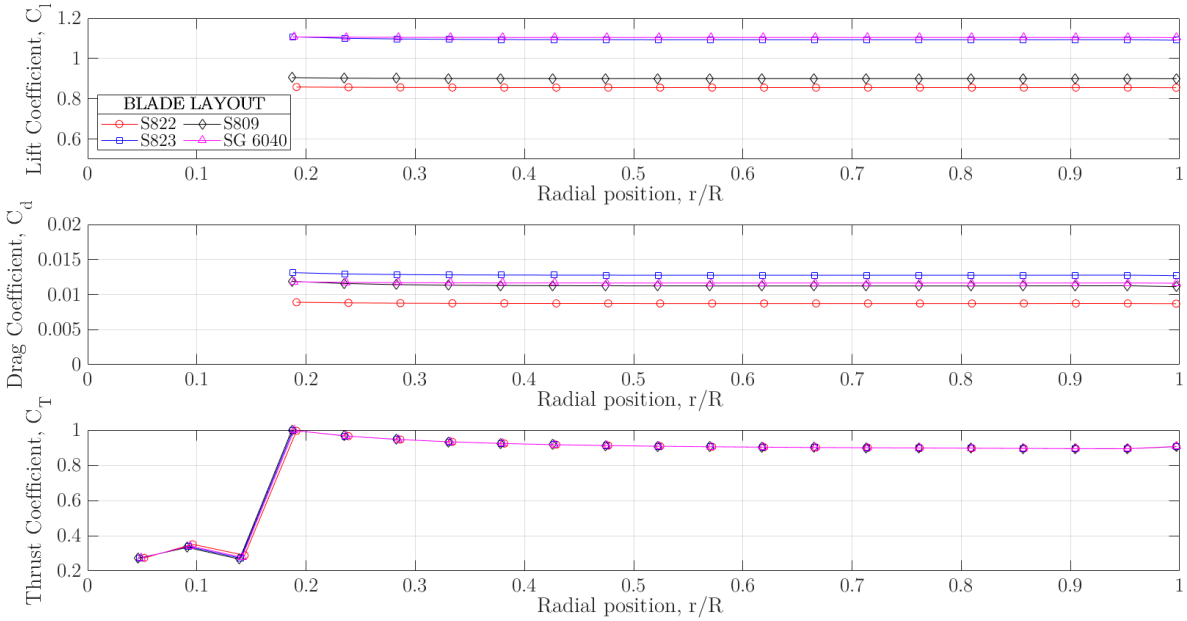


Figure 7.22 Lift, Drag and Thrust coefficient variation over the blade span (B)

Figure 7.23 shows the axial and tangential loads distribution over the blade span. Axial loads vary linearly with radius whereas tangential loads are almost constant (unless in the tip, that they drop due to tip-loss effect). A remarkable difference between chord distribution A and B designs is that in the current one axial loads do not fall near the blade tip. This loads are not desirable since they increase the turbine thrust, the RFWBM and the axial displacement of the tip points (out-of-plane). It is an effect of the higher chord values at the tip. By the way, this fact answers the question of why the tip of the blades in large scale wind turbines are sharp.

The root flap-wise bending moment curve has a quadratic shape, and the torque curve is mainly linear with respect to the radius. The maximum values of the axial loads are approximately 10 times higher than the maximum tangential ones.

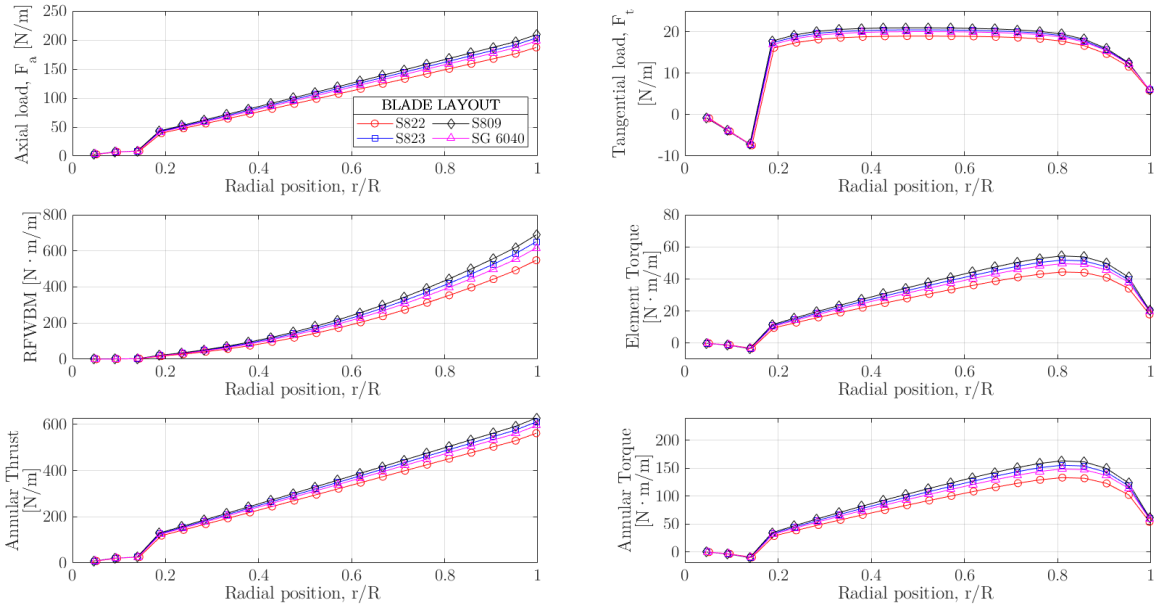


Figure 7.23 Aerodynamic forces and moments over the blade span (B)

The results regarding the whole wind turbine are shown in the following table.

Table 7.4 Power, Torque Cp, Thrust, and RFWBM in the design point (B)

BLADE LAYOUT	Mechanical Power [W]	Electric Power [W]	Thrust [N]	Torque [N · m]	RFWBM [N · m]	C_P (w/o loss)	C_P (with loss)
S822	3348	2846	856	230	551	0.465	0.395
S823	3927	3338	1011	293	713	0.461	0.392
S809	4122	3504	1071	317	776	0.457	0.389
SG 6040	3749	3187	957.7	273	656	0.465	0.395

- Geometric representation of the blades

The 3D shape and polar section view of the constituting airfoils are illustrated below.

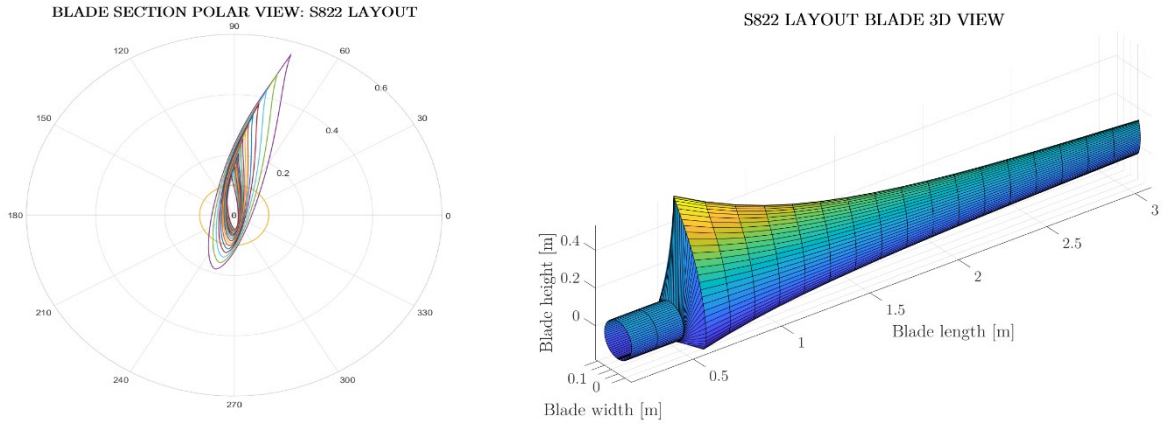


Figure 7.24 S822 airfoil blade design: 3D and polar view (B)

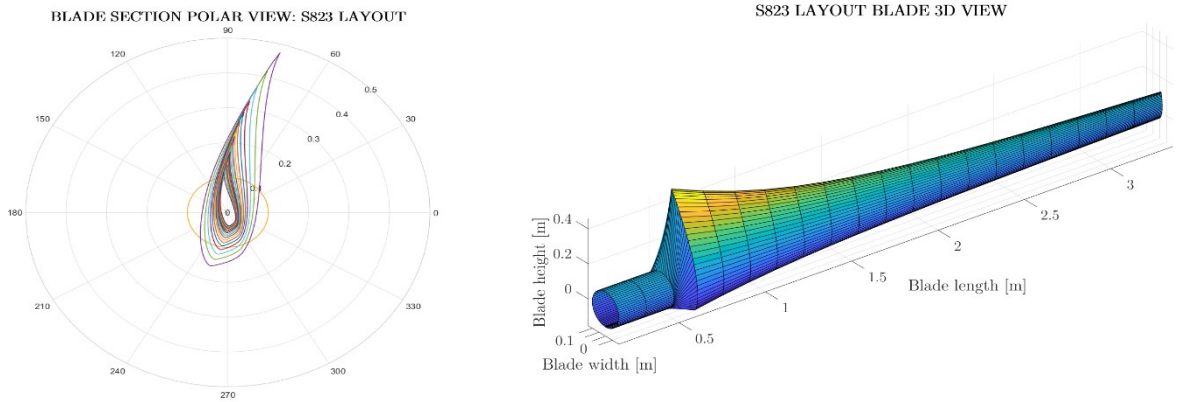


Figure 7.25 S823 airfoil blade design: 3D and polar view (B)

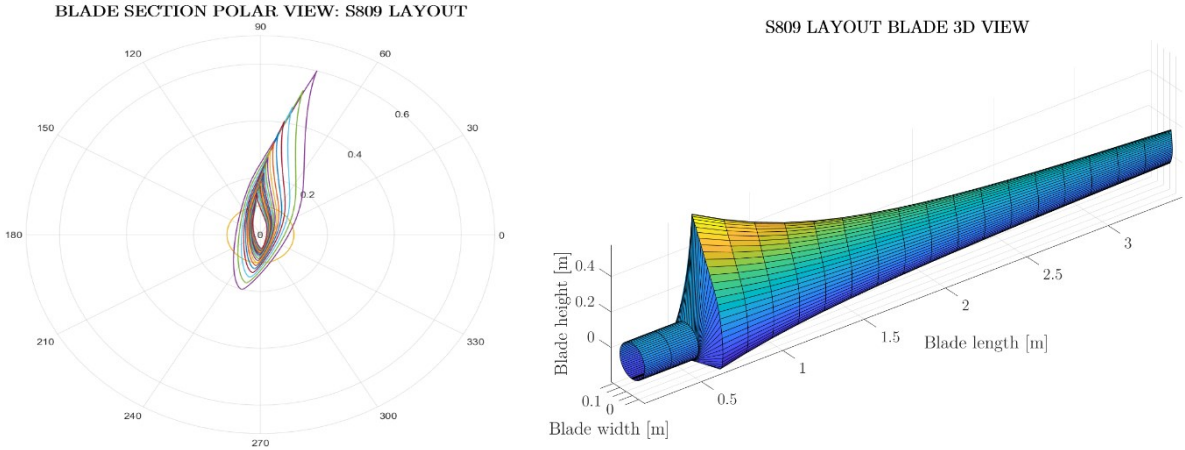


Figure 7.26 S809 airfoil blade design: 3D and polar view (B)

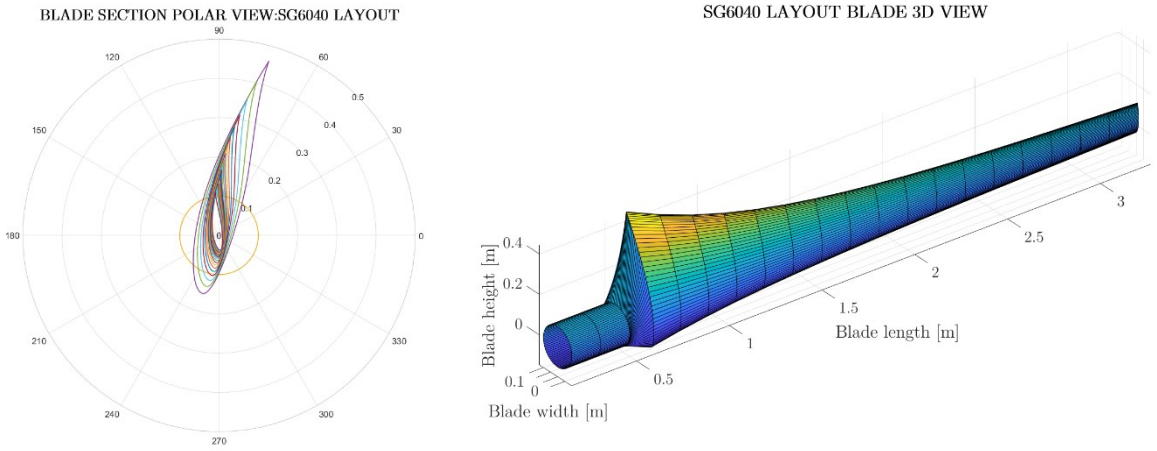


Figure 7.27 SG 6040 airfoil blade design: 3D and polar view (B)

As well as in the previous study, the blade with S809 airfoil is the longest and highest one; S822 blade is the shortest but it is the second in height. S823 and SG 6040 blades are slenderer, being the SG 6040 9.5 cm shorter.

- **Off-design behaviour of the wind turbines**

Figure 7.28 shows the off-design performance curves of the designed blades "working" in the wind turbine model that has been defined in the work. Electric generated power, power coefficient, thrust, torque, blade root flap-wise bending moment and rotational speed are shown.

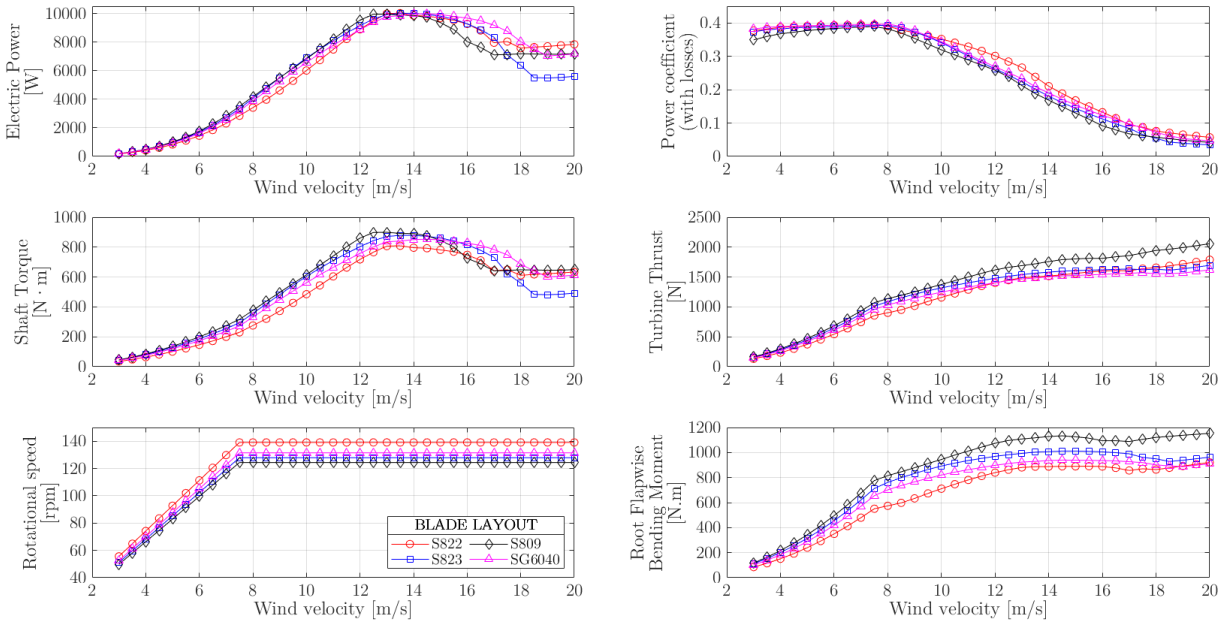


Figure 7.28 Wind turbine performance curves for the sized blades (B)

With this new design the S809 airfoil blade is also the one with higher loads; which in part are due to its high size. In contrast, loads in S822 design are usually below the others.

As in the previous section, the AEP and the projected area of the blades have been quantified and taken into consideration. Figure 7.29 depicts the obtained values for the current blade designs.

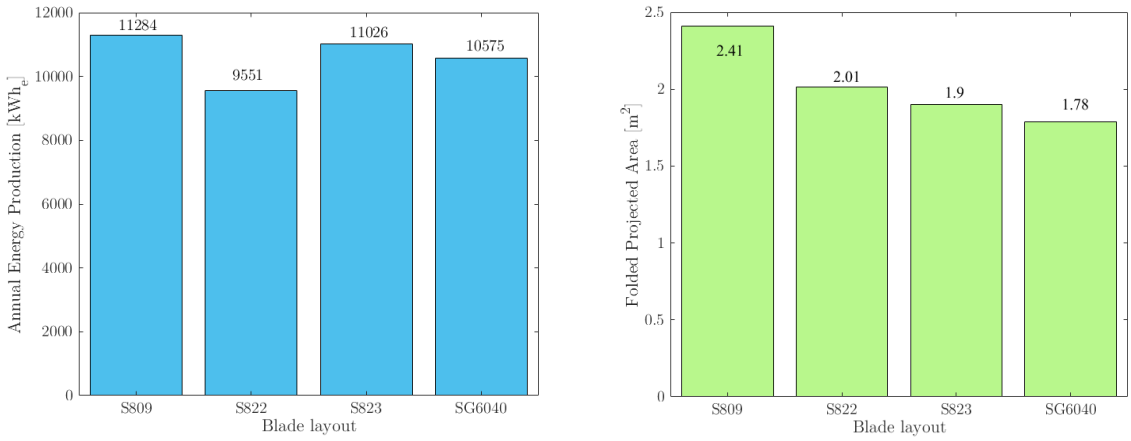


Figure 7.29 AEP and blade folded projected area for the blade design alternatives (B)

The selection of the preferred alternative comes justified below. The comparison and the tradeoffs are very similar to the previous case:

- S809 blade design generates slightly more annual energy than S823, but it has a much higher projected area, the power coefficient is lower and the maximum loads are much higher than all the others.
- Between S823 and SG 6040 the choice is not as clear: S823 generates more energy, but SG 6040 has bit lower projected area, a slightly higher efficiency all over the wind velocity range and lower maximum loads: torque, root bending moment and turbine thrust. Specially the maximum value of RFWBM is lower for the SG 6040 airfoil design.
- S822 design is the shortest one, but has a considerably higher projected area than the previous 2 designs due to its high chord lengths. Its energy generation is much lower than the others and surprisingly the maximum thrust forces on the rotor are higher than for S823 and SG 6040 designs. It works under the maximum variable velocity range, which can difficult the design due to vibration and also experiences higher centrifugal forces. As an upside, above 9.5 m/s it has the highest power coefficient.

Finally, the chosen design is: **SG 6040 AIRFOIL BLADE**

7.2 Final blade design selection

In the previous section we have compared each of the 4 blade design alternatives with the same (A or B) chord distribution among them, and one alternative has been selected from each chord distribution simulation. It has been decided the design made with SG 6040 airfoils for both cases.

Now, both of the previously selected SG 6040 designs (with A and B chord distributions) will be compared and a final blade design will be decided. Figure 7.30 Wind turbine performance curves for SG 6040 airfoil blades (A and B distr.)shows the wind turbine performance curves of the two designs in the same plot.

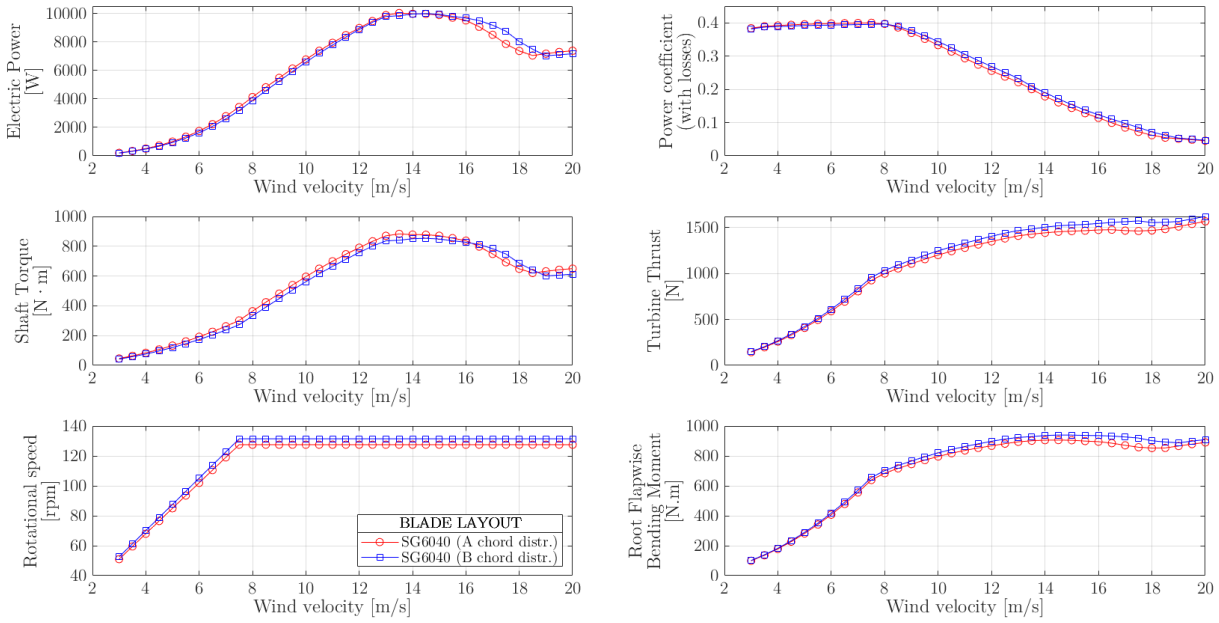


Figure 7.30 Wind turbine performance curves for SG 6040 airfoil blades (A and B distr.)

Also, Figure 7.31 depicts the AEP and projected area of the blades under study.

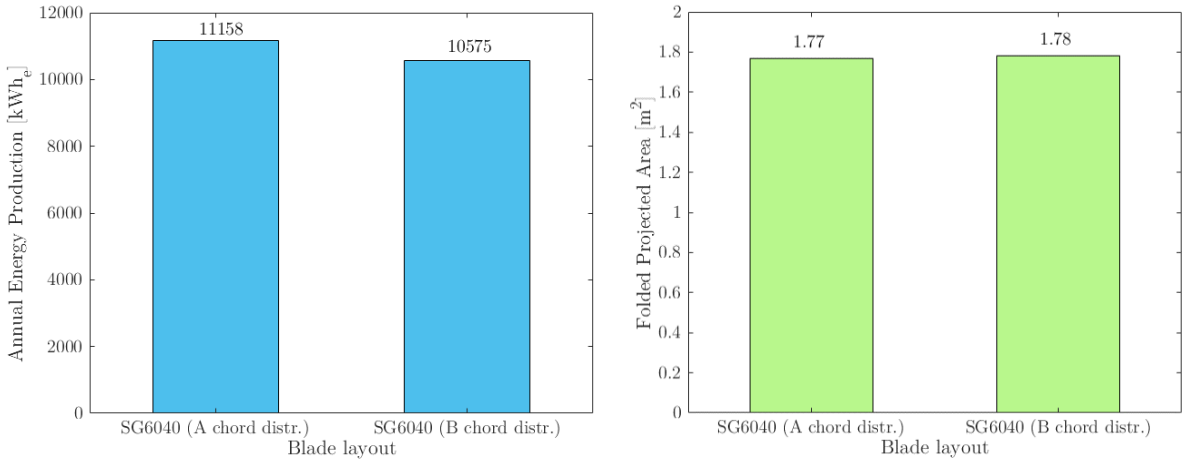


Figure 7.31 AEP and blade folded projected area for SG 6040 airfoil blades (A and B distr.)

Once the curves have been plotted in the same graph the behaviour differences of the wind turbines can be studied:

- In terms of size, A solution has a slightly larger rotor diameter. In fact, A chord distribution blade is 10 cm larger than the other one (0.2 m larger in rotor diameter). However, chord lengths are higher for B solution, which makes the overall blade projected area (1.78 m^2) nearly equal to the former alternative (1.77 m^2).
- The AEP is higher (aprox. 5.36 %) for the A chord distribution design (11158 kWh/year) than for the second, B chord distribution, case (10575 kWh/year). This greater energy capture is mainly due to the higher rotor diameter.
- The power curves are quite similar: for low wind velocities larger energy capture is achieved with the A alternative, but between 16 and 19 m/s the opposite happens. However, wind events in this velocity range are not frequent in the studied location.
- Power coefficients in the variable speed range (between 3 and 7.5 m/s) are almost equal but subtly higher for the A chord distribution alternative. In the constant rotational speed range, B chord distribution blade shows a clear better aerodynamic performance.
- Regarding aerodynamic loads, B chord variation alternative shows higher turbine thrust and RFWBM all over the wind velocity operating range. All the same, the maximum shaft torque is lower for the B chord layout alternative due to the higher rotational speed (lower rotor radius). Even though a higher rotational speed decreases the torque in the shaft, the variable rotational speed range is usually preferred to be as narrow as possible in order to avoid vibration problems (excitation of eigenfrequencies). A narrower variable rotational speed gap is achieved for the A design, which will lead to lower centrifugal loads.

After analysing these parameters, it has been decided to choose the **A chord variation blade design** alternative to continue the project with. However, this kind of tradeoffs should be studied deeper because the blade design involves three main pillars: Aerodynamic performance, mechanical aspects and component manufacture and cost.

7.3 Final blade design: geometry and CAD model

Table 7.5 shows the selected blade design geometry by means of the radial coordinate, chord length, twist angle and airfoil type in each of the 21 blade sections (boundaries) that define the whole blade span.

Table 7.5 Selected blade design layout

Section number	Radial position [m]	Twist angle [°]	Chord length [m]	Airfoil
1	0.150	0.00	0.200	Cylinder
2	0.311	0.00	0.200	Cylinder
3	0.472	0.00	0.200	Cylinder
4	0.633	20.70	0.547	SG 6040
5	0.794	16.57	0.499	SG 6040
6	0.955	13.39	0.450	SG 6040
7	1.116	10.91	0.405	SG 6040
8	1.277	8.94	0.367	SG 6040
9	1.438	7.35	0.335	SG 6040
10	1.599	6.03	0.307	SG 6040
11	1.760	4.93	0.282	SG 6040
12	1.921	4.00	0.261	SG 6040
13	2.082	3.20	0.243	SG 6040
14	2.243	2.49	0.227	SG 6040
15	2.404	1.87	0.212	SG 6040
16	2.565	1.30	0.198	SG 6040
17	2.726	0.76	0.184	SG 6040
18	2.887	0.20	0.169	SG 6040
19	3.048	-0.44	0.150	SG 6040
20	3.209	-1.54	0.120	SG 6040
21	3.370	-3.99	0.059	SG 6040

Knowing the chord and twist value of the airfoil at each blade section and also having the chord-scaled coordinates of the SG 6040 airfoil, these coordinates can be scaled and rotated to those values. Also, the pressure center of the airfoil can be made coincident with the span axis of the blade. All this is done by the MATLAB® script1A.8 Drawing.m ,which, apart from the 3D and polar plots, gives the XYZ coordinates of the points that define the blade sections (where X is the axis parallel to the shaft, Z the span-wise axis and Y a perpendicular axis to the previous ones, fulfilling the right hand rule).

These coordinates are saved into .txt files, one for each blade section, that means that all the points share the Z coordinate, and then are loaded into the CAD software SolidWorks 2018, defining curves by those points. Figure 7.32 shows the span-wise distribution of the blade sections.

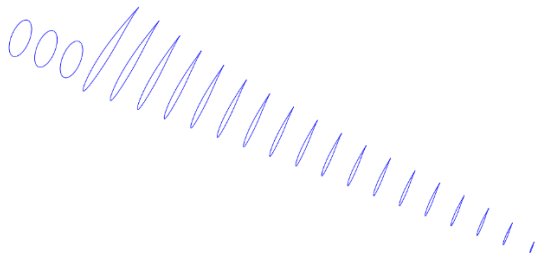


Figure 7.32 Blade section layout view in SolidWorks

With the loft tool of SolidWorks, a solid blade can be obtained. The result is shown in the following figure.

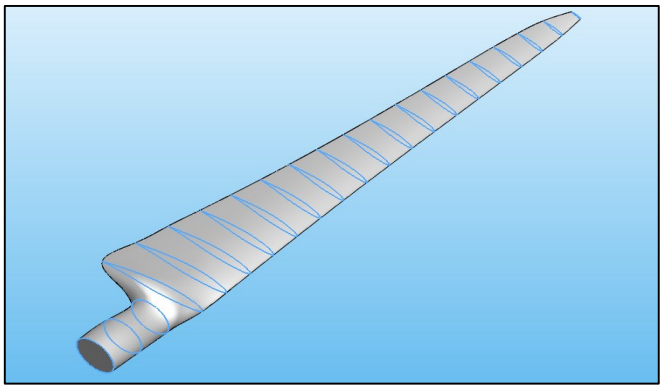


Figure 7.33 Solid blade after lofting process of the blade sections

With this tool, the surface takes the necessary geometry to adhere to the given sections. For instance, in the root zone the surface has to take the necessary curvature radius to fit the airfoil curves.

Besides, Figure 7.34 shows how the three blades have been added forming an assembly that models the wind turbine rotor (blades and hub).

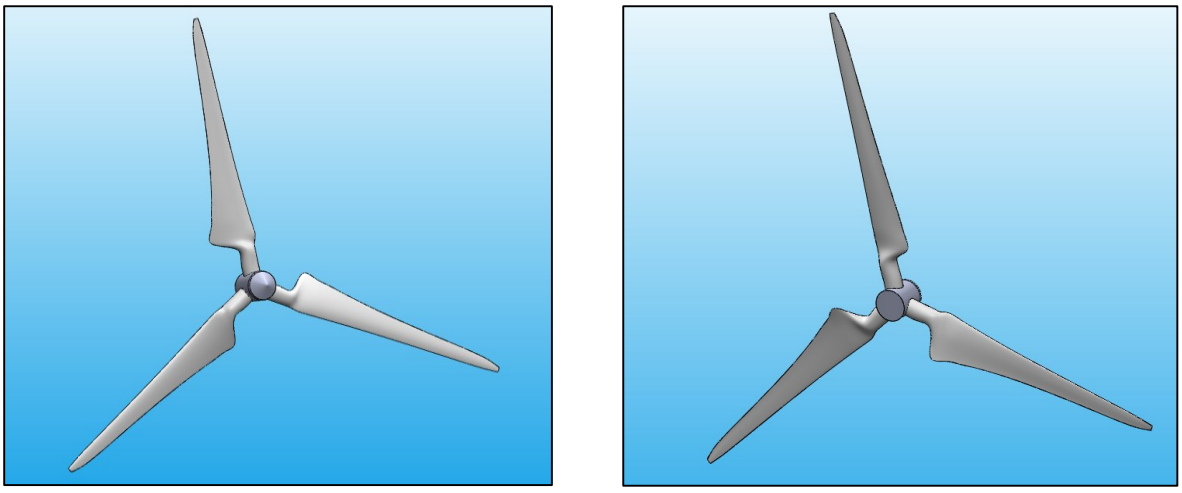


Figure 7.34 Upwind (left) and downwind (right) view of the wind turbine rotor

At this point a table that summarises the characteristics of the whole wind turbine can be made. This kind of table is always given by wind turbine manufacturers.

Table 7.6 Designed wind turbine characteristics and parameters

WIND TURBINE CHARACTERISTICS	
Rated Power [kW]	10
Number of blades	3
Rotor diameter [m]	6.74
Tower height [m]	25
Hub diameter [m]	0.3
Swept area [m^2]	35.7
Blade length [m]	3.22
Cut-in velocity [m/s]	3
Cut-out velocity [m/s]	20
Rated velocity [m/s]	13.5
Power control system	VS-FP, passive stall
Rotational speed range [rpm]	0 - 127.5
Maximum tip speed [m/s]	45
Maximum Power Coefficient (w/o losses)	0.471
Drive train, generator and converter efficiency [%]	85

Chapter 8

CODE VALIDATION AND WIND TURBINE ANALYSIS

The aim of this chapter can be divided into 3 different points:

1) The validation of the written MATLAB® codes, which is performed by simulating the rotor performance with the designed blade layout and wind turbine characteristics with the software WT_Perf, that was developed by the NREL. The results obtained with the own MATLAB® scripts are compared with those of WT_Perf.
2) The comparison and quantification of the effect that 3D corrections have into the rotor performance parameters, such as the electric power output, the power coefficient, the thrust force or the RFWBM.
3) The comparison between the designed rotor and a commercial wind turbine of the same size, rating and power control system. Study of the energy that would be produced in the selected location with that machine and comparison with the own rotor result. The chosen machine is the BERGEY EXCEL 10 [56].

The chapter is organised into 3 different sections. In the first one the WT_Perf environment and model creation are briefly explained. In the second section, rotor simulation results are shown using both 2D and 3D data, performing the simulations with the own codes and also WT_Perf, in such a way that not only MATLAB® and WT_Perf codes are compared but also the influence of the 3D corrections of airfoil coefficients is observed. In the last section the designed wind turbine rotor is compared with the BERGEY EXCEL 10 machine.

8.1 Wind Turbine model creation in WT_Perf

WT_Perf is a software developed by the NREL and needs to be executed from the computer terminal. The wind turbine model is read from an input file with .wtp extension which at the same time makes reference to .dat files where aerodynamic coefficients of the airfoils are stored. Hence, this input file needs to specify all the parameters and characteristics required to perform BEMT simulations over a wind velocity range for a wind turbine with the specifications and blade layout defined in the previous section.

The next table describes some of the most important parameters that need to be set in the input files and also the given value, which can be numeric or logic (True/False). Nevertheless, the whole WT_Perf input and output files are available in the APPENDIX B below in this document.

Table 8.1 WT_Perf input file flag description and given values

Configuration	Flag	Description	Value
INPUT	**DimenInp**	Tells WT_Perf to expect dimensional input parameters.	True
	Metric	Tells whether Metric or English units are used	True
MODEL	**MaxIter**	Maximum iterations in the BEMT induction loop	10000
	Atol	Tolerance to accept induction factors convergence	1.0e-005
ALGORITHM	**TipLoss**	Tells the code to turn on the Prandtl tip-loss model	True
	HubLoss	Tells the code to turn on the Prandtl hub-loss model	True
	SWIRL	Tells the code to enable the calculation of the tangential induction factor	True
	SkewWake	Tells the code to correct the induction factor for a skewed wake.	False
	AdvBrake	Uses the advanced brake state model (Buhl)	True
	AIDrag	Inclusion of the drag term in the axial-induction algorithm.	True
	TIDrag	Inclusion of the drag term in the tangential-induction algorithm.	True

TURBINE DATA	**NumBlade**	Number of blades	3
	RotorRad	Rotor radius	3.37
	HubRad	Hub radius	0.15
	HutHt	Hub height	25
	NumSeg	Number of blade segments	20
	[RElm; Twist; Chord; Afile; PrntElem]	Blade geometry: **NumSeg x 5 table** Relm= Radial coordinate of element Twist= Twist angle of element Chord=Chord length at element Afile=Airfoil coefficient .dat file PrntElem= Save blade element data?	see APPENDIX B
AERODYNAMIC DATA	**Rho**	Air density	1.138
	KinVisc	Kinematic viscosity (for Re computation)	1.5289e-005
	NumAF	Number of airfoil coefficient files. *Under this line list the files paths*	3 (if 2D data) 19 (if 3D data)
I/O SETTINGS	See input files (APPENDIX B) for further detail		
CASE ANALYSIS	**NumCses**	Number of simulation points	35 (from 3 m/s to 20, each 0.5 m/s)
	[Wind vel.; Rot. speed; Pitch]	WT case study table: **NumCases x 3** Enter wind velocity in m/s, the rotational speed in r.p.m and the (pitch angle) in degrees	see APPENDIX B

The airfoil coefficient data files have the AeroDyn 14 format and allow to introduce data for several different Re numbers. For 2D airfoil data simulation only 3 files are required (root, transition element and SG 6040 airfoil). However, if 3D corrections are to be taken into consideration each blade section airfoil needs to be linked to 1 airfoil data file.

The main output of the simulation is a table (.oup) that gives the values of mechanic power, torque, turbine thrust, root bending moment (flap-wise) and power coefficient (without losses), as well as wind velocity, rotational speed and TSR. These output files are shown in the APPENDIX B (B.3 and B.4).

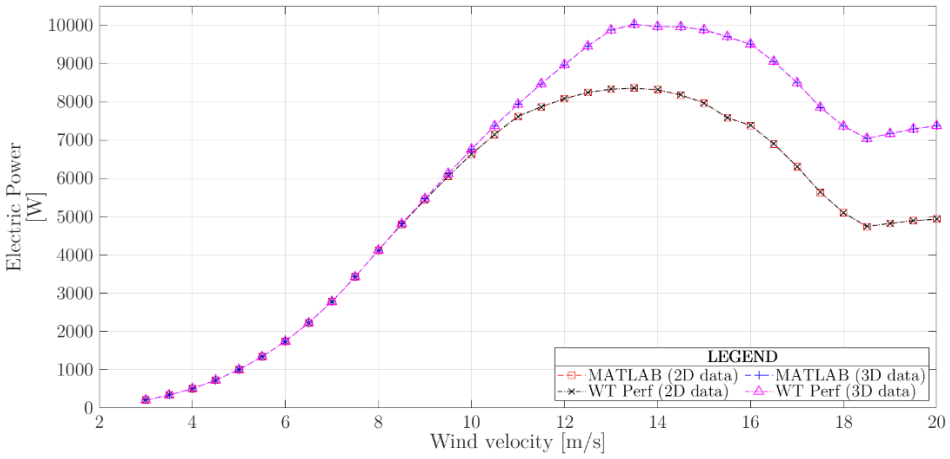
8.2 Simulation results: Code validation and 3D correction's influence

In this section the results of 4 different BEMT simulations are shown:

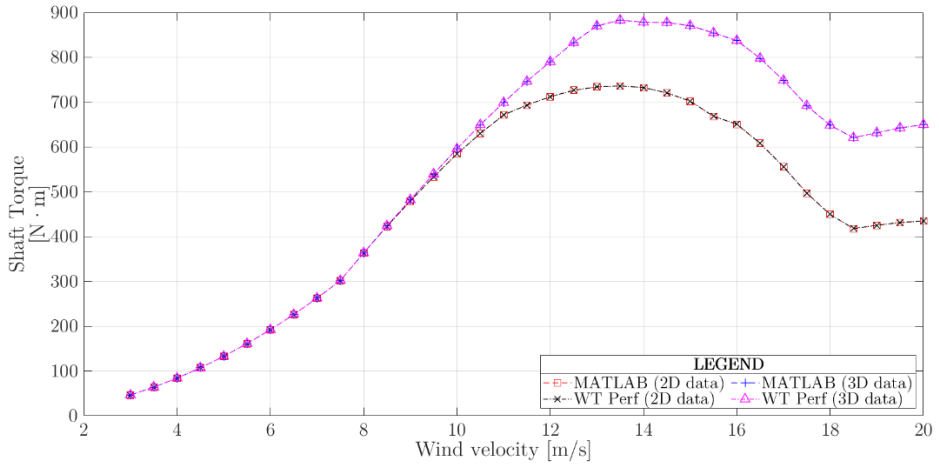
- MATLAB® (own code) simulation with 2D airfoil data
- WT_Perf simulation with 2D airfoil data
- MATLAB® (own code) simulation with 3D airfoil data
- WT_Perf simulation with 3D airfoil data

The results have been plotted into the same graphs in order to make comparisons and reach conclusions. Not only the correlation between the own code results and WT_Perf can be determined but also the importance, influence or effect of the 3D correction of airfoil aerodynamic coefficients which was made with the model of Snel et *al.* in Chapter 3.

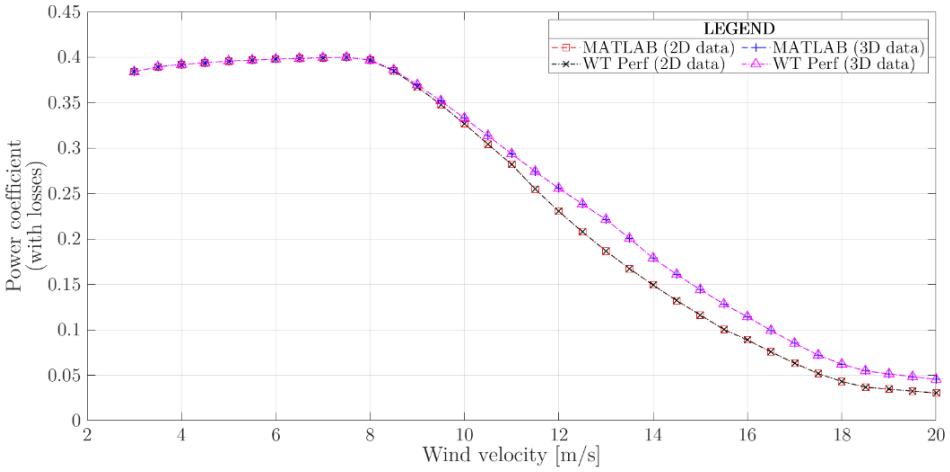
8.2.1 Power, Torque and Power Coefficient



a) Electric Power



b) Shaft Torque



c) Power coefficiet (with losses)

Figure 8.1 Simulations' results for the electric power (a), torque (b) and power coefficient (c)

Two main facts can be highlighted from the figures above: the first one is the **perfect agreement** between the curves performed with MATLAB® and those with WT_Perf, for both, 2D and 3D data simulations. It is clearly shown how the curves attach to each other. Figure 8.2 shows a zoom view of the power curves at rated velocity (13.5 m/s) for 3D data simulations. The difference between both curves is of approximately **1 W,** which corresponds to a percentage difference of about **0.01 %**, that is negligible and may be due to number rounding during the iterative process.

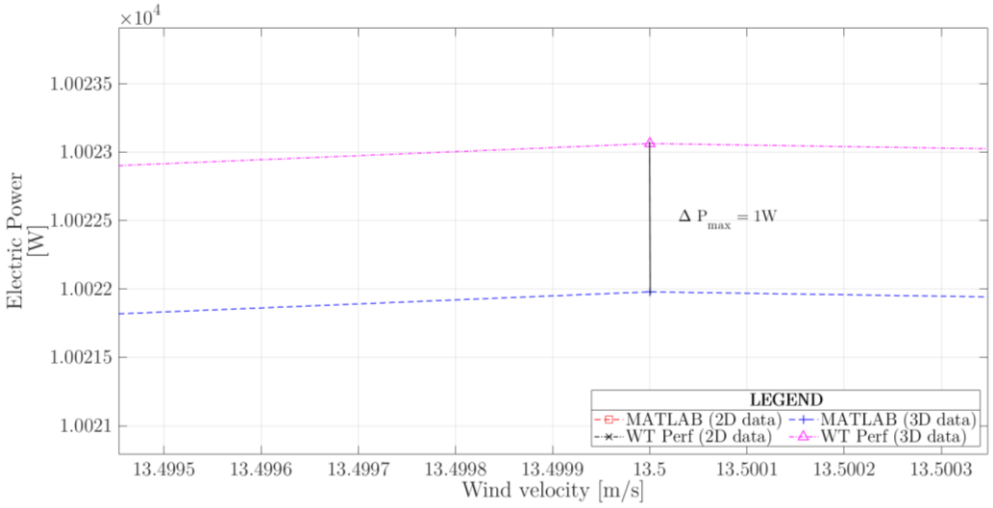


Figure 8.2 Zoom view of electric power at rated velocity (3D aerodynamic data)

The second important issue that is shown is how the curves derived from 3D corrected data detach from the ones obtained from 2D data above the velocity of 8 m/s, more or less. This happens due to the fact that between 3 and 7.5 m/s the airfoil sections are working at design angles of attack, where flow is attached to the airfoil upper surface and does not experience radial movement (so airfoil coefficients are not corrected). When

the turbine starts to work at constant rotational speed, angles of attack become higher and the flow starts to detach from the surface, moving outwards radially and experiencing an acceleration that increases the lift locally, leading to the separation of the curves.

That increase in lift coefficient leads to an augmentation of the tangential loads and therefore, in the turbine torque and power:

$$dF_t = \frac{1}{2}\rho W^2 c C_t r dr \tag{8.1}$$

where

$$C_t = C_l \sin\varphi - C_d \cos\varphi \tag{8.2}$$

The main consequence of the higher peak power and torque is related with the design of the wind turbine components. If one based the electric generator selection on the peak power resultant from a 2D data based simulation, the power would be underestimated and the generator components could be burnt under real conditions. Similarly, if the design and calculation of the drive train components was based on simulations performed with 2D airfoil data, the real loads, which are much higher could lead to mechanical fractures. That is the reason why it is so important to model accurately the behaviour of the blade airfoils. The maximum value of electric power (10022 W) is around 1665 W higher if we take into account 3D effects on the blade, and 147 N · m the shaft torque (3D simulation max values are **20 % higher** than those of 2D data approximately).

Above the point where the curves are divided the power coefficient will be higher using 3D corrected data, since the generated power is higher and the rotor diameter equal.

8.2.2 Turbine Thrust

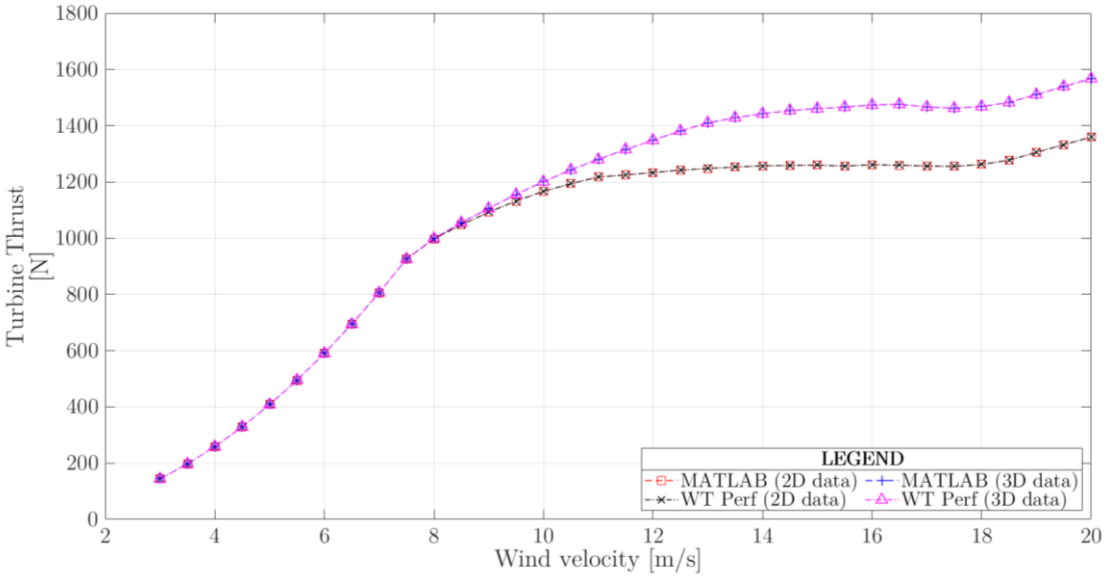


Figure 8.3 Simulations' results for the turbine thrust

As in the previous graphs, Figure 8.3 also shows a **perfect coincidence** between MATLAB® and WT_Perf results. If we analyse the maximum thrust value in simulations where 3D airfoil coefficient data was used WT_Perf result is only 0.15 N higher than in the own code results (which corresponds to a difference of **0.0095 %**).

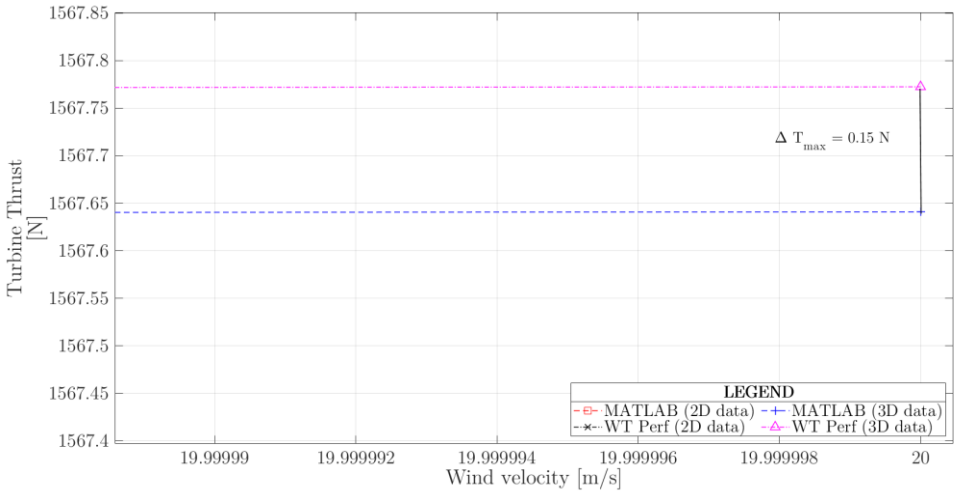


Figure 8.4 Zoom view of turbine thrust at cut-out velocity (3D aerodynamic data)

Regarding the difference between 2D and 3D data simulations, the same than with power, torque and power coefficient happens: curves start to differ at about 8 m/s of wind velocity. This happens due to the lift force increase in airfoil sections where flow detaches. The turbine thrust is the result of the span-wise integration of axial loads, which can be written as:

$$dF_a = \frac{1}{2}\rho W^2 c C_n dr \tag{8.3}$$

where

$$C_n = C_l \cos\varphi + \mathrm{C}_d \sin\varphi \tag{8.4}$$

As a consequence, when lift increases not only the tangential loads are higher but also the loads in axial direction. The maximum thrust value (1568 N) is approximately 209 N higher if we take into account 3D effects on the blade, which means that 3D simulations max. values are **15.4% higher** than in 2D airfoil data simulations. This underlines how important is a proper caracterisation of airfoil aerodynamic coefficients used in BEMT codes, in order to obtain realistic loads acting on the turbine components. The thrust over the turbine is important in order to size the wind turbine tower and foundation.

8.2.3 Root Flap-Wise Bending Moment (RFWBM)

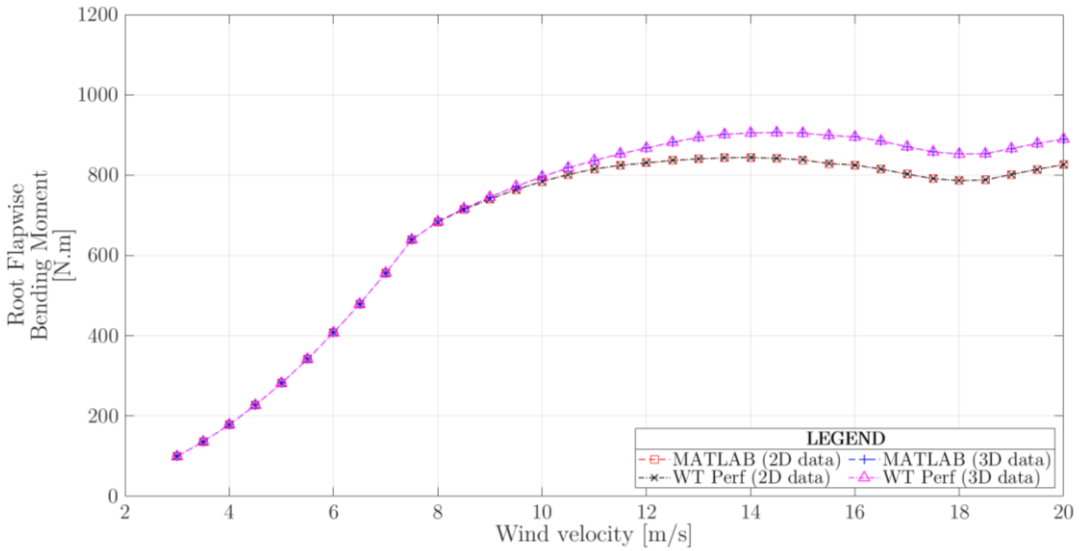


Figure 8.5 Simulations' results for the RFWBM

Figure 8.5 shows how the **perfect correlation** between MATLAB® and WT_Perf simulations is also present in the RFWBM computation. A peculiarity is that the peak value **does not appear at the cut-out** velocity (as for turbine thrust) but at 14.5 m/s. Figure 8.6 depicts how for the maximum values in the simulations where 3D airfoil coefficient data was used, WT_Perf result is only 0.1 N · m higher than the MATLAB® own code result (which corresponds to a percentage difference of **0.011 %**).

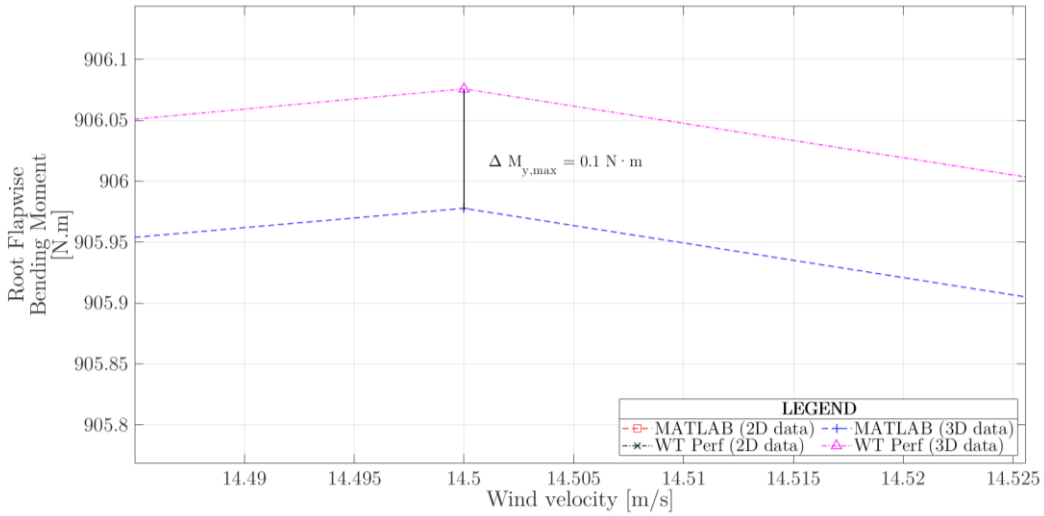


Figure 8.6 Zoom view of RFWBM at 14.5 m/s (when value=max) (3D aerodynamic data)

With regard to the difference between 2D and 3D data-based simulations, while in the second one the maximum value (906 N · m) is obtained at 14.5 m/s, in the former one the maximum (844 N · m) is achieved at 14 m/s. This absolute difference of 62 N · m

represents an increase of **7.35 %** with respect to 2D simulation's maximum RFWBM. This increase in the 3D data-based simulations is, as for turbine thrust, due to the rise in axial loads when flow separation occurs. The root maximum bending moment is important for the mechanical design of the blade root.

8.2.4 Annual Energy Production (AEP)

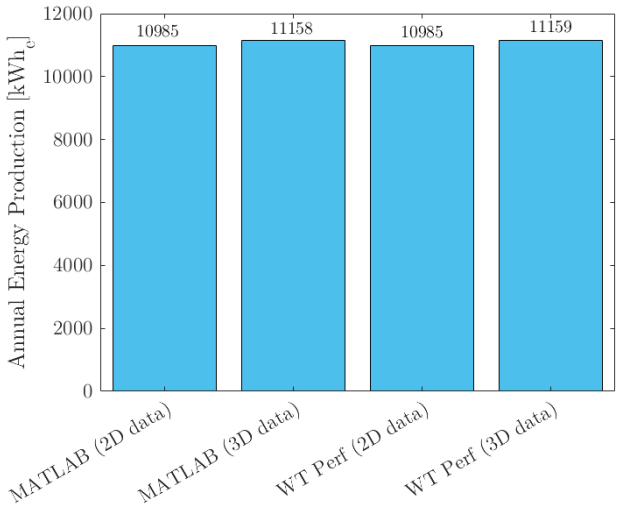


Figure 8.7 Simulations' results for the AEP

Regarding the energy yield, results are also **practically equal** for MATLAB® and WT_Perf computations. More interesting is to notice that in this case the impact of the aerodynamic coefficients' correction is much lower than in the previous parameters. Performing a simulation where airfoil data is corrected considering 3D effects an AEP of 11158 kWh is obtained whereas the obtained value is 10985 kWh for a computation with pure 2D airfoil coefficients taken from XFOIL. The 3D simulation value is only **1.58 %** higher than if we did not consider the radial flow in the rotor plane. This soft effect is due to one reason: the wind distribution of the studied location, being the most probable events on the power curve side where all the curves are equal or very similar (where flow does not dettach).

8.2.5 Section conclusions

1) The **results** obtained from MATLAB® and WT_Perf simulations are in perfect agreement for both 2D and 3D airfoil-data-based simulations. However, it needs to be remarked that the computational time in WT_Perf simulations is much lower.
2) The **proper caracterisation of airflow rotational 3D effects** in order to model the realistic behaviour of a wind turbine blade has a very high importance regarding the sizing, selection and calculation of other elements of wind turbines, such as the generator, drive train, tower or foundation, among others.

8.3 Wind Turbine comparison

The designed wind turbine has been compared with the EXCEL 10 (Figure 8.8) commercial 10 kW rated small-scale wind turbine, which is manufactured by BERGEY WindPower [56]. The main features of the wind turbine are illustrated in Table 8.2.

Figure 8.8 BERGEY EXCEL 10 wind turbine (10 kW) [12]

Table 8.2 Main characteristics of the BERGEY EXCEL 10

Rated Power [kW]	10
Number of blades	3
Rotor diameter [m]	7
Cut-in velocity [m/s]	2.5
Cut-out velocity [m/s]	None
Tower height [m]	24 - 49
Weight [kg]	545
Power control system	VS-FP, passive stall
Over speed protection	Autofurl
Rotational speed range [rpm]	0 - 400
Gearbox	None
Generator	Neo Permanent Magnet
Inverter	12 kW Powersync II

The manufacturer also gives the values of the Annual Energy Production as a function of the average wind velocity (hub height) at the selected location.

Table 8.3 AEP of BERGEY EXCEL 10 wind turbine based on average velocity [56]

AVERAGE WIND SPEED	AEP [kWh]
8 mph (3.57 m/s)	4910
10 mph (4.47 m/s)	9850
12 mph (5.36 m/s)	16530
14 mph (6.25 m/s)	24330
16 mph (7.15 m/s)	32388

These values can be extended to other velocities by cubic fitting of the data points.

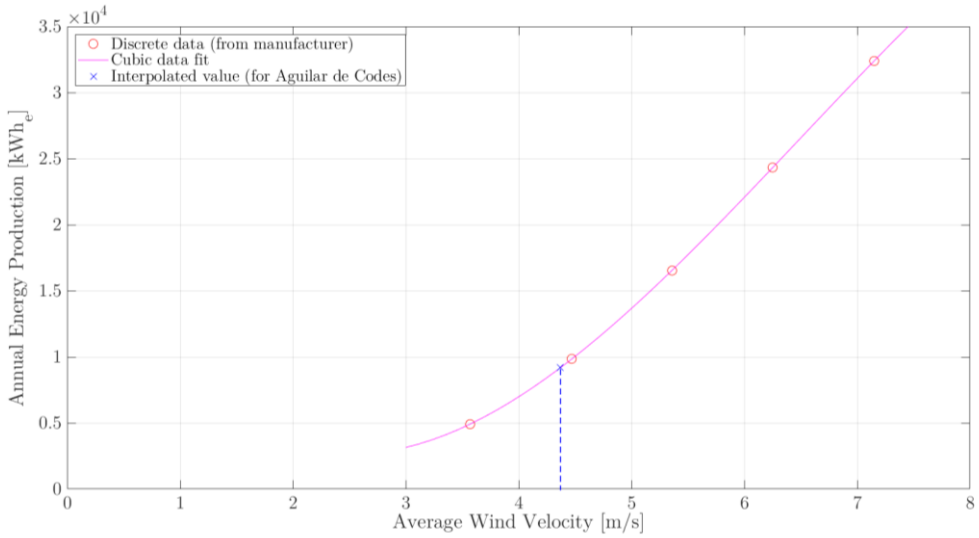


Figure 8.9 Annual Energy Production with BERGEY EXCEL 10 wind turbine

Therefore, from Figure 8.9 it is derived that the annual amount of energy generated at Aguilar de Codes (where the average wind velocity is equal to 4.37 m/s) would be **9179 kWh** with this machine.

With the designed wind turbine, the annual energy capture is **11158 kWh**, **1979 kWh** more than with the BERGEY EXCEL 10 wind turbine. This represents an increase of **21.5 %** on the energy produced with this commercial wind turbine. Even if the efficiencies of the wind turbine's elements apart from the blades have been supposed, it gives an idea of a good blade design with respect to energy capture.

The main reason lies on the simplified blade design in the BERGEY EXCEL 10 turbine: the chord is constant along the blade span and the same seems to happen with the chord twist. This simplified blade leads to a cheaper blade manufacture, though. On the contrary, the designed blade has an optimum chord length and twist angle distribution over the blade span. Its manufacture will obviously be more expensive.

Chapter 9

CONCLUSIONS AND FUTURE WORK

All the works and projects, independently of their difficulty, make the author learn and become aware of the further work that the project offers. So, in this last chapter, the work conclusions and future work lines are shown.

9.1 Main conclusions

9.1.1 Fulfilment of objectives

In the introduction section, at the start of the project, several objectives were proposed and they have been successfully achieved.

A wind location in Navarre was found and characterised by the Weibull PDF shape and scale parameters. Also, its wind power density was calculated. Then, the airfoils used in low Re number applications have been identified and their aerodynamic coefficients simulated with XFOIL. These values were then corrected to take into account the radial movement of the air when crossing the rotor plane. After it, the values were extrapolated to all the angle of attack range. With this purpose the models of Snel et *al.* and Viterna-Corrigan were implemented with success in MATLAB®.

With the implementation of the BEMT algorithm in MATLAB® 8 blade design alternatives were created and finally 1 is selected and modelled in CAD.

9.1.2 Good agreement with WT_Perf results

The results of the MATLAB® simulations over the designed blade and wind turbine showed a very good agreement with the computations in WT_Perf. The percentage of the difference in the maximum values of power, torque, twist and RFWBM for the 3D data-based simulations is around **0.01 %**. This is mainly due to the fact that both the own code and WT_Perf code are based on the AeroDyn standard. Also, WT_Perf places the calculation nodes in the center of the elements, and so has been done in the off-design own script (1A.9 Main_OFFDESIGN.m). With the same convergence criteria for the induction factor loops, the results are almost identical.

9.1.3 Importance of accurate aerodynamic coefficients' data

Chapter 8 showed how the characteristic curves of the wind turbine differ at a certain wind velocity when 2D and 3D data are used for BEMT computations. The peak power and torque increase, and if the airfoil coefficients are not well caracterised, the selection of the generator, gearbox, and inverter can be wrong, due to underestimations.

On the other hand, aerodynamic loads are always higher than for 2D data simulations, and the proper caracterisation of data is essential to discover the real loads over the blade and turbine.

All in all, one of the major drawbacks of BEMT algorithms is that the airfoil behaviour needs to be numerically caracterised in a proper way. Even though (as seen in Section 3.3) several models are available in the literature, it is very difficult to discover which one is the best approach in each case.

9.1.4 High energy capture of the wind turbine

In the last section of Chapter 8 the AEP of the designed blade, understood as a part of a wind turbine, was compared with the energy that a commercial machine of the same rating (the BERGEY EXCEL 10) would produce if it was placed in the selected location. It was discovered that the designed wind turbine, with the aerodynamically optimized blade, captures **21.5 %** more energy with respect to the commercial one, whose blades are simple and lower in manufacturing cost. This shows the high energy capture of the design, with regarding its size, power rating and average velocity.

9.1.5 Need of multidisciplinary knowledge

During the work, several decisions and tradeoff solving require a multidisciplinary work team. The blade design needs to be understood as a synergy work among several disciplines and the wind turbine companies are always organised into several departments in order to accomplish it.

9.1.6 Viability of the investment in small-scale wind turbines

The downside of the small-scale wind turbines is always the economic viability of the project. The low wind speeds that are usually present in these wind turbine locations lead to quite low energy capture, which makes it difficult to justify the initial investment.

The kWh price depends on the contract with the energy supply company and also on the use time frame. Furthermore, it oscillates and it can vary trough years. In fact, the price will probably rise. Supposing a constant kWh price of 0.15 €, the yearly saving in the bill would be:

$$\frac{€_{saved}}{year} = 11158\ \frac{kWh}{year} \cdot 0.15 \frac{€}{kWh} = \mathbf{1673.7} \frac{€}{\boldsymbol{year}} \tag{9.1}$$

The overall price for 10 kW rated wind turbines is usually between 48000 and 65000 USD [57]. With 56500 USD as an average value, and converting it to €:

$$WT\ Price = 56500\ USD \frac{0.88\ €}{1\ USD} = \mathbf{49720\ €} \tag{9.2}$$

With the rough supposition of a constant electricity price and therefore constant annual saving, the payback period of the investment would be:

$$Payback\ period = \frac{49720\ €}{1673.7 \frac{€}{year}} = \boldsymbol{29.7\ years} \tag{9.3}$$

Even if the increase in the kWh price or the possible governmental helps to renewable energy have not been taken into consideration, **the payback period is very high** and the investment is not straight forward. The reason of the low small-scale wind turbine installation is mainly due to the difficulty that the farm, factory or house owners find to have a good wind resource on their land.

9.2 Future work lines

There are 3 main future work lines which could be taken as continuation of the actual work but that due to time constraints are not part of it.

9.2.1 Chord and twist linerisation study

With the BEMT implementation that is used in this work a certain blade geometry can be linerised. In the present project the twist angle and chord length along the blade span have been optimised in order to get the maximum glide-ratio and power coefficient at a certain design point.

From this resultant geometry, chord and twist values can be linerised span-wise. Three cases can be studied:

1) Optimum twist with linerised chord
2) Linearised twist with optimum chord
3) Linearised chord and twist

This values can be compared with the both optimum chord and twist blade layout in terms of power coefficient, energy production and aerodynamic loads. Power coefficient and hence energy production will be lower, but the blade manufacture cost will be lower too, leading to a certain tradeoff between the aerodynamic design of the blade and the manufacturing cost. Also, it can be discovered which of the variables (chord or twist) affects more to the energy capture in a wind turbine.

9.2.2 Computational Fluid Dynamics (CFD) studies

The CFD is a numerical analysis tool that helps solving complicated problems related to fluid flows. 2 kind of tasks derived from this work could be performed with this kind of computational tool:

1) Comparison between BEMT and CFD results

A very common practice is to model the 3D geometry of a blade with a CAD software and perform after a CFD simulation, in order to obtain the torque and power developed in the shaft. These results can be then compared with the ones from BEMT numerical analysis. Most studies and research work find that the results' agreement is quite good.

2) Blade root and tip geometry refinement with CFD analysis

Blade root and tip geometry can be refined with CFD analysis, which screens the stream lines and pressure field of the air, in order to reduce the noise emission from the tip and the power loss due to the creation of vortices.

9.2.3 Mechanical design and validation of the blade

This work is focused on the aerodynamic design of the blade, this is, the outer surface. However, a real wind turbine blade is defined by the materials of which it is made and their thickness. This gives the blade a certain weight, resistance and rigidity, which will need to put up with the loads acting on it.

This task is carried out in the work that complements this one, called *Complemento al Trabajo Fin de Master* (12 ECTS), where the blade materials will be specified and loads will be calculated with the aeroelastic tool FAST, taking as a reference the **IEC 61400-2:2006** standard.

APPENDIX A: MATLAB® SCRIPTS

A.1 Wind_resource_assesment.m

```
clear all
close all
clc

%% SCRIPT DESCRIPTION:
% This script reads the wind data collected from several meteorological
stations in the province of Navarre . The data is scaled to the hub height
and organized in a columns diagram. Frequency and cumulative graphs are
shown.
% This data is then fit by the Weibull PDF (2 parameters: k and c), which
gives a quite accurate idea of the quality of the emnplacement as well as
help in  the selection of the WT power control type (Chapter 5) and the
selection of the design point.
%% SCRIPT START
%% WIND DATA LOADING
location = 'Aguilar de Codes'; % Choose the location of the HAWT: ETSIA,
Aguilar de Codes

    if strcmp(location,'ETSIA')
    velocity_data=xlsread('Wind_Data','ETSIA','B14:B52573'); % Loading wind
velocity every 10 min
    h_ref= 10; % Reference height for the wind data [m]

    elseif strcmp(location,'Aguilar de Codes')
    velocity_data=xlsread('Wind_Data','Aguilar de Codes','B14:B52573');
    h_ref= 10;
    end

h_hub= 25; % Hub height of the HAWT [m]
alpha= 0.2; % Parameter that depends on the rugosity of the ground [IEC
61400-1-2005]

velocity_data_corrected= velocity_data*(h_hub/h_ref)^alpha; % Velocities
extrapolated to the hub height (Potential law)
vel_max=max(velocity_data_corrected); % Maximum of the corrected values

txt=['The average velocity is: ', num2str(mean(velocity_data_corrected)),
'm/s'];
disp(txt)

txt=['The standard deviation of the data is: ',
num2str(std(velocity_data_corrected)), 'm/s'];
disp(txt)
%% FREQUENCY COUNTER AND DISPLAY
```

```
close all
bins=0:1:ceil(vel_max); % Bin n°1=frequencies for 0<U<1 m/s
h=histogram(velocity_data_corrected,bins);
frequency=h.Values; % Frequency vector
wind_hours=frequency*(1/6); % Each recorded event is equal to 10 min of
wind at that velocity (mean)
% Cumulative wind hours
cumulative_hours(1)=wind_hours(1);
for u=2:length(frequency)
    cumulative_hours(u)=cumulative_hours(u-1)+ wind_hours(u);
end
total_hours=cumulative_hours(end); % Total hours recorded from the station
% Wind probability and cumulative probability
probability=wind_hours/total_hours;
cumulative_probability=cumulative_hours/total_hours;

% Generation of graphics
velocities=0.5:1:(ceil(vel_max)-0.5);
font='LM Roman 12';

figure('DefaultTextFontName', font, 'DefaultAxesFontName',
font,'DefaultAxesFontSize',20,'DefaultTextFontSize',20);
subplot(1,2,1)
bar(velocities,wind_hours,1,'FaceColor','0.30,0.75,0.93'),
hold on
xlabel('Wind velocity [m/s]','FontWeight','bold')
xticks(1:1:ceil(vel_max))
ylabel('Wind hours in a year [h]','FontWeight','bold')
axis([0 ceil(vel_max) 0 max(wind_hours)+100]);

subplot(1,2,2)
bar(velocities,cumulative_hours,1,'FaceColor','0.30,0.75,0.93')
hold on
xlabel('Wind velocity [m/s]','FontWeight','bold')
xticks(1:1:ceil(vel_max))
ylabel('Cumulative hours in a year [h]','FontWeight','bold')
axis([0 ceil(vel_max) 0 max(cumulative_hours)+500]);

figure('DefaultTextFontName', font, 'DefaultAxesFontName',
font,'DefaultAxesFontSize',20,'DefaultTextFontSize',20);
subplot(1,2,1)
bar(velocities,probability,1,'FaceColor','0.30,0.75,0.93')
hold on
xlabel('Wind velocity [m/s]','FontWeight','bold')
xticks(1:1:ceil(vel_max))
ylabel('Probability','FontWeight','bold')
axis([0 ceil(vel_max) 0 max(probability)+0.05]);

subplot(1,2,2)
bar(velocities,cumulative_probability,1,'FaceColor','0.30,0.75,0.93')
hold on
xlabel('Wind velocity [m/s]','FontWeight','bold')
xticks(1:1:ceil(vel_max))
ylabel('Cumulative probability','FontWeight','bold')
axis([0 ceil(vel_max) 0 1.1]);

%% DEFINITION OF THE WEIBULL DISTRIBUTION
% Removal of 0 values of wind speed
for i=1:length(velocity_data_corrected)
```

```
    if velocity_data_corrected(i)<0.00001
       velocity_data_corrected(i)=0.00001;
    end
end
% Calculation of the shape and scale parameters (k and c respectively)

parmhat = wblfit(velocity_data_corrected); % Weibull parameters using
maximum likelihood method
c= parmhat(1) % Scale parameter of the Weibull distr.
k= parmhat(2) % Shape parameter of the weibull distr.

velocity_vector=0:0.1:20;

weibull_pdf = wblpdf(velocity_vector,c,k); % Velocity values are evaluated
in the Weibull PDF
weibull_cdf= wblcdf(velocity_vector,c,k); % Velocity values are evaluated
in the Weibull CDF

%% Probability Density Function (PDF) and Cumulative Distribution Function
(CDF)
% Discrete probability density
probability_density=probability; % Bin width = 1 m/s, then P(ui)=f(ui)

% Generation of graphs
figure('DefaultTextFontName', font, 'DefaultAxesFontName',
font,'DefaultAxesFontSize',20,'DefaultTextFontSize',20);
subplot(1,2,1)
bar(velocities,probability_density,1,'FaceColor','0.30,0.75,0.93')
hold on
plot(velocity_vector,weibull_pdf,'r','LineWidth',1.3)
xlabel('Wind velocity [m/s]','FontWeight','bold')
xticks(1:1:ceil(vel_max))
ylabel('Probability Density Function','FontWeight','bold')
axis([0 ceil(vel_max) 0 max(probability)+0.05])
legend('Discrete wind data',' Data fit by Weibull','location','northeast');

subplot(1,2,2)
bar(velocities,cumulative_probability,1,'FaceColor','0.30,0.75,0.93')
hold on
plot(velocity_vector,weibull_cdf,'r','LineWidth',1.3)
xlabel('Wind velocity [m/s]','FontWeight','bold')
xticks(1:1:ceil(vel_max))
ylabel('Cumulative Distribution Function','FontWeight','bold')
axis([0 ceil(vel_max) 0 1.1]);

%% DEFINITION OF THE ENERGY PROBABILITY DENSITY
% Wind power in a surface unit
rho = 1.225; % Reference value at sea level altitude
P= 0.5*rho*velocity_vector.^3;
% Energy available in the wind
fig=figure('DefaultTextFontName', font, 'DefaultAxesFontName',
font,'DefaultAxesFontSize',12,'DefaultTextFontSize',12);
left_color = [1 0 0];
right_color = [0 0 1];
set(fig,'defaultAxesColorOrder',[left_color; right_color]);
subplot(2,1,1)
yyaxis left
plot(velocity_vector,weibull_pdf,'r','LineWidth',1.3)
xlabel('Wind velocity [m/s]');
```

```
xticks(1:1:20)
ylabel('Probability Density Function','FontWeight','bold')
yyaxis right
plot(velocity_vector,P,'b-','LineWidth',1.3);
ylabel('Available power in wind [W/m^{2}]');
xticks(1:1:20)
% Energy probability density [W/m2]
probable_energy=P.*weibull_pdf; %EQ. 2.17
subplot(2,1,2) % Energy probability density
plot(velocity_vector,probable_energy,'black','LineWidth',1.3);
xlabel('Wind velocity [m/s]');
axis([0 20 0 (max(probable_energy)+1)])
xticks(1:1:20)
ylabel('Energy probability density [W/m^{2}]');

% Velocity with maximum energy content, ume:
pos=find(probable_energy==max(probable_energy)); %Locating the maximum of
the curve
num2str(velocity_vector(pos));
txt=['The wind velocity that contains more energy is:
',num2str(velocity_vector(pos)),' m/s'];
disp(txt)
% Wind power density WPD (Average)
WPD=trapz(velocity_vector(2:end),probable_energy(2:end))
```

A.2 Snel_3D_corrections.m

```
clear all
close all
clc
%% SCRIPT DESCRIPTION:
% This script implements the airfoil aerodynamic characteristic's
correction to 3D radial flow usign the model of Snel et al. 2D airfoil data
generated from XFOIL is read and corrected for each of the blade element
nodes (different output data for each blade element).
% Data is saved into a .xlsx file for each airfoil. Different sheets are
used for each blade element.

%% SCRIPT START
% Loading necessary data

for f=1:4     % External loop for each airfoil

    if f==1
    airfoil='S822';
    load Chord_S822.txt % Chord value at the boundaries
    load Radial_position_S822.txt % Radial position of the design stations
    chord=Chord_S822';
    rv=Radial_position_S822;
    airfoil_data=xlsread('Airfoil_Data','S822','B7:AK204'); % Original (2D,
XFOIL) airfoil polar data
    writting_document='Airfoil_Data_S822_3D.xlsx'; % New file for corrected
data
    alpha_crit=6; % The point where lift coefficient no longer follows the
ideal straight line. Different for each airfoil and suposed constant for
diverse Re number

    elseif f==2
    airfoil='S823';
    load Chord_S823.txt
    load Radial_position_S823.txt
    chord=Chord_S823';
    rv=Radial_position_S823;
    airfoil_data=xlsread('Airfoil_Data','S823','B7:AK199');
    writting_document='Airfoil_Data_S823_3D.xlsx';
    alpha_crit=7;

    elseif f==3
    airfoil='S809';
    load Chord_S809.txt
    load Radial_position_S809.txt
    chord=Chord_S809';
    rv=Radial_position_S809;
    airfoil_data=xlsread('Airfoil_Data','S809','B7:AK205');
    writting_document='Airfoil_Data_S809_3D.xlsx';
    alpha_crit=6;

    elseif f==4
    airfoil='SG6040';
    load Chord_SG6040.txt
    load Radial_position_SG6040.txt
    chord=Chord_SG6040';
    rv=Radial_position_SG6040;
    airfoil_data=xlsread('Airfoil_Data','SG6040','B7:AK205');
```

```
    writting_document='Airfoil_Data_SG6040_3D.xlsx';
    alpha_crit=7.5;
    end


Ne=20; % Number of blade elements
for k=1:Ne % Definition of radial position and chord in the element nodes
(mid-point of boundaries)
rv_nodes(k)=(rv(k)+rv(k+1))/2;
chord_nodes(k)=(chord(k)+chord(k+1))/2;
end

c=chord_nodes;
r=rv_nodes;

for re=1:10 % Loop for the diverse Re numbers in each blade element

% 3 column data loading for each Re, alpha = [-25,25]°
columns = (3*re-2):1:3*re;
if f==3 && re==1 % for S809,due to the inestability of the graph at Re=100
000, data from Re=200 000 is used
columns=4:1:6;
end
data_3columns=airfoil_data(:,columns);
data_3columns= data_3columns(all(~isnan(data_3columns),2),:); % Removal of
NaN values of the matrix

alpha=data_3columns(:,1);
lift_coeffcient=data_3columns(:,2);
drag_coefficient=data_3columns(:,3);

CL_corrected=zeros(1,length(alpha));

alpha_crit_pos=find(alpha==alpha_crit);

alpha0=find(alpha==0);
Cl_alpha0=lift_coeffcient(alpha0);
n=Cl_alpha0; % Cut of the line with y axis
m= (lift_coeffcient(alpha_crit_pos)-Cl_alpha0)/alpha_crit; % Slope of the
straight line
y=@(x) m*x+n ;
Cl_linear= y(alpha);

for element=4:Ne % Loop that runs span-wise, element by element
     f_CL=3*(c(element)/r(element))^2; % Correction coefficient by Snel et
al. (Table 3.5)
    for aoa=1:length(alpha) % Correction of Snel et ali. when boundary
layer separation starts
    angle_of_attack=alpha(aoa);
        if angle_of_attack <= alpha_crit
        CL_corrected(aoa)=lift_coeffcient(aoa); % Cl is not changed for low
angles of attack
        else
        CL_2D= lift_coeffcient(aoa); % 2D values of lift coeffficient
        % Calculation of Delta CL
        delta_CL=Cl_linear(aoa)-CL_2D; % Difference between inviscid flow
Cl and 2D Cl (Delta Cl)
        CL_3D=CL_2D+f_CL*delta_CL; % EQ.(3.6)
        CL_corrected(aoa)=CL_3D;
```

```
        end
    end
    MATRIX=[alpha CL_corrected' drag_coefficient] % Matrix containing the
corrected lift coefficients

%% Writing corrected data into an EXCEL sheet (One file for each airfoil,
one sheet for each element)
if re==1 % Location of the writing cells for each Re computation
    cell='B7';
elseif re==2
    cell='E7';
elseif re==3
    cell='H7';
elseif re==4
    cell='K7';
elseif re==5
    cell='N7';
elseif re==6
    cell='Q7';
elseif re==7
    cell='T7';
elseif re==8
    cell='W7';
elseif re==9
    cell='Z7';
elseif re==10
    cell='AC7';
 end
    xlswrite(writting_document,MATRIX,num2str(element),cell) % Writting
corrected polars: Different files for each airfoil,one sheet for each blade
station.
% Graphic representation
figure(re)
plot(alpha,CL_corrected)
hold on
grid on
xlabel('Angle of attack, \alpha')
ylabel('Corrected Lift coefficient, C_L_,_3_D')

lgd=legend('4','5','6','7','8','9','10','11','12','13','14','15','16','17',
'18','19','20','Location','NorthWest');
lgd.NumColumns=3;
lgd.Title.String='Blade element';
end
end
end
```

A.3 Viterna_360_extrapolation.m

```
clear all
clc
close all
%% SCRIPT DESCRIPTION:
% This script reads the 3D corrected airfoil coefficients and makes a 360°
extrapolation using the Viterna-Corrigan model nd also the flat plate
theory.
% The script also allows the extrapolation of the 2D XFOIL data (not
% corrected), for comparisons in Chapter 8

%% SCRIPT START
simulation='3D'; % Choose 3D to extrapolate 3D corrected data. Choose 2D to
extrapolate 2D XFOIL data (not corrected)
for f=1:4 % Extrapolation is done for each different airfoil

% Loading the original data created with XFOIL (from QBlade) and corrected
with Snel model:
    if f==1
    airfoil='S822';
    load Chord_S822.txt
    load Beta_S822.txt
    load Radial_position_S822.txt
    Chord=Chord_S822;
    Beta=Beta_S822*pi/180;
    Rad_pos=Radial_position_S822;
    file_read='Airfoil_Data_S822_3D.xlsx'; % 3D corrected polars in [-
25,25] range
    file_write='Airfoil_Data_S822_3D_360_extrapolated.xlsx';
    range='B7:AK204';
    original_data_matrix = xlsread('Airfoil_Data.xlsx','S822','B7:AK204');
% S822 Airfoil original polars for several Re numbers (NOT CORRECTED)
    elseif f==2
    airfoil='S823';
    load Chord_S823.txt
    load Beta_S823.txt
    load Radial_position_S823.txt
    Chord=Chord_S823;
    Beta=Beta_S823*pi/180;
    Rad_pos=Radial_position_S823;
    original_data_matrix = xlsread('Airfoil_Data.xlsx','S823','B7:AK206');
% S823 Airfoil original polars for several Re numbers (NOT CORRECTED)
    file_read='Airfoil_Data_S823_3D.xlsx';
    file_write='Airfoil_Data_S823_3D_360_extrapolated.xlsx';
    range='B7:AK206';

    elseif f==3
    airfoil='S809';
    load Chord_S809.txt
    load Beta_S809.txt
    load Radial_position_S809.txt
    Chord=Chord_S809;
    Beta=Beta_S809*pi/180;
    Rad_pos=Radial_position_S809;
    original_data_matrix = xlsread('Airfoil_Data.xlsx','S809','B7:AK205');
% S809 Airfoil original polars for several Re numbers (NOT CORRECTED)
    file_read='Airfoil_Data_S809_3D.xlsx';
    file_write='Airfoil_Data_S809_3D_360_extrapolated.xlsx';
```

```
    range='B7:AK205';

    elseif f==4
    airfoil='SG6040';
    load Chord_SG6040.txt
    load Beta_SG6040.txt
    load Radial_position_SG6040.txt
    Chord=Chord_SG6040;
    Beta=Beta_SG6040*pi/180;
    Rad_pos=Radial_position_SG6040;
    original_data_matrix =
xlsread('Airfoil_Data.xlsx','SG6040','B7:AK205'); % SG6040 Airfoil original
polars for several Re numbers (NOT CORRECTED)
    file_read='Airfoil_Data_SG6040_3D.xlsx'; % 3D corrected polars in [-
25,25] range
    file_write='Airfoil_Data_SG6040_3D_360_extrapolated.xlsx';
    range='B7:AK205';
    end
%% VITERNA EXTRAPOLATION: FOR EACH BLADE ELEMENT AND 10 DIFFERENT RE
NUMBERS (100000:1000000)

R=Rad_pos(end); % Rotor radius
N=21; % Boundary number. Element number= N-1
R_hub=0.15; %Hub radius

% Blade aspect ratio calculation (AR=b/SMC=b^2/Sp) For Viterna model
Projected_chord_vector=[ Chord(1).*cos(Beta(1)); 2*Chord(2:(end-
1)).*cos(Beta(2:(end-1))); Chord(end).*cos(Beta(end))];
suma=sum(Projected_chord_vector);
gap=(R*2-R_hub)/(N-1); %Element width
b=R-R_hub; % Blade length
Sprojected=suma*gap/2; % [m2]
AR=b^2/Sprojected; % Blade aspect ratio EQ.(3.13)

if strcmp(simulation,'3D')
num=N-1; % If 3D effects are taken into account, polar extrapolations need
to be done over the blade span
elseif strcmp(simulation,'2D') % If 3D effects are not taken into account,
airfoil polars are equal within the blade span
num=3;
end

for element=4:num % Loop to run over the blade elements. For 2D data, only
1 calculation is needed

if strcmp(simulation,'3D')
airfoil_data_matrix = xlsread(file_read,num2str(element),range); % Getting
corrected polars of each airfoil file and each blade element
end

for re=1:10 %10 diferent Re numbers

% 3 column data loading for each Re, alpha = [-25,25]° (REGION 1)
columns = (3*re-2):1:3*re;

if strcmp(simulation,'3D')
airfoil_data=airfoil_data_matrix(:,columns);
elseif strcmp(simulation,'2D')
airfoil_data=original_data_matrix(:,columns);
```

```
end

airfoil_data= airfoil_data(all(~isnan(airfoil_data),2),:); % Removal of NaN
values of the matrix

alpha_vector_Z1=airfoil_data(:,1);
CL_values_Z1=airfoil_data(:,2);
CD_values_Z1=airfoil_data(:,3);

% VITERNA & CORRIGAN EXTRAPOLATION MODEL (REGIONS II and III)
% Model parameter
CD_max=1.11+0.018*AR; % EQ.(3.12)

% Linking points for ZONE I
alpha_s=alpha_vector_Z1(end)*pi/180;
CD_s=CD_values_Z1(end);
CL_s=CL_values_Z1(end);
% Viterna extrapolation functions
B2=(CD_s-CD_max*sin(alpha_s)^2)/cos(alpha_s); % EQ.(3.11)
CD=@(alpha) CD_max.*sin(alpha).^2+B2.*cos(alpha); %Drag coefficient
function, EQ.(3.10)

A2=(CL_s-CD_max*sin(alpha_s)*cos(alpha_s))*sin(alpha_s)/(cos(alpha_s)^2); %
EQ.(3.9)
CL=@(alpha) CD_max.*sin(2*alpha)./2+A2.*(cos(alpha).^2)./sin(alpha); %Lift
coefficient function, EQ.(3.8)

% Evaluation of points between alpha = [26°,90°] (REGION 2): Viterna-
Corrigan equations
alpha_vector_Z2=(26:1:90)';
alpha_vector_Z2_rad=alpha_vector_Z2*pi/180;
CL_values_Z2=CL(alpha_vector_Z2_rad);
CD_values_Z2=CD(alpha_vector_Z2_rad);

% Linking points for ZONE III (negative angles of attack)
alpha_s=alpha_vector_Z1(1)*pi/180;
CD_s=CD_values_Z1(1);
CL_s=CL_values_Z1(1);
% Viterna-Corrigan extrapolation functions
B2=(CD_s-CD_max*sin(alpha_s)^2)/cos(alpha_s);
CD=@(alpha) CD_max.*sin(alpha).^2+B2.*cos(alpha); %Drag coefficient
function

A2=(CL_s-CD_max*sin(alpha_s)*cos(alpha_s))*sin(alpha_s)/(cos(alpha_s)^2);
CL=@(alpha) CD_max.*sin(2*alpha)./2+A2.*(cos(alpha).^2)./sin(alpha); %Lift
coefficient function

% Evaluation of points between alpha [-90°,-26°] (REGION 3): Viterna-
Corrigan equations

alpha_vector_Z3=(-90:1:-26)';
alpha_vector_Z3_rad=alpha_vector_Z3*pi/180;
CL_values_Z3=CL(alpha_vector_Z3_rad);
CD_values_Z3=CD(alpha_vector_Z3_rad);

% ZONES 4 AND 5: FLAT PLATE THEORY EQUATIONS (2D DATA WILL BE ALWAYS USED)
% Setting of peak Cl value
if strcmp(simulation,'2D')
pos=find(alpha_vector_Z2==45);
```

```
CL_peak=0.7*CL_values_Z2(pos); % It is considered to be the 70% of the
CL( when alpha=45°)
elseif strcmp(simulation,'3D') %2D Cl and Cd at alpha=45° will be taken as
reference
extrapolated_2D_data=xlsread('Airfoil_Data_360_Extrapolated_2D.xlsx',airfoi
l,'B7:AK516');
columns = (3*re-2):1:3*re;
data_3columns=extrapolated_2D_data(:,columns);
data_3columns= data_3columns(all(~isnan(data_3columns),2),:); % Removal of
NaN values of the matrix

alpha=data_3columns(:,1);
lift_coeffcient=data_3columns(:,2);
drag_coefficient=data_3columns(:,3);
pos=find(alpha==45);
CL_peak=0.7*lift_coeffcient(pos);
end
CD_peak=CD_max; %  CD peak value for the flat plate (needs to match with
Viterna-Corrigan curves)axim

% Flat plate theory functions
CL_flatplate=@(alpha) 2.*CL_peak.*sin(alpha).*cos(alpha); % Lift
coefficient function, EQ.(3.14)
CD_flatplate=@(alpha) 2.*(CD_peak./2.)*sin(alpha).^2; % Drag coefficient
function, EQ.(3.15)

% Evaluation of points between alpha = [91°,180°] (REGION 4):
alpha_vector_Z4=(91:1:180)';
alpha_vector_Z4_rad=alpha_vector_Z4*pi/180;
CL_values_Z4=CL_flatplate(alpha_vector_Z4_rad);
CD_values_Z4=CD_flatplate(alpha_vector_Z4_rad);

% Evaluation of points between alpha = [-180°,-91°] (REGION 5):
alpha_vector_Z5=(-180:1:-91)';
alpha_vector_Z5_rad=alpha_vector_Z5*pi/180;
CL_values_Z5=CL_flatplate(alpha_vector_Z5_rad);
CD_values_Z5=CD_flatplate(alpha_vector_Z5_rad);

% Implementation test plotting: Example of different zones
font='LM Roman 12';
% figure('DefaultTextFontName', font, 'DefaultAxesFontName',
font,'DefaultAxesFontSize',20,'DefaultTextFontSize',20);

% GENERATION OF THE 360° POLARS (Columns and row concatenations)
AOA =
[alpha_vector_Z5;alpha_vector_Z3;alpha_vector_Z1;alpha_vector_Z2;alpha_vect
or_Z4]; % Vector containing all the angles of attack in the alpha = [-
180°,+180°] range
CL = [CL_values_Z5;CL_values_Z3;CL_values_Z1;CL_values_Z2;CL_values_Z4]; %
Vector containing all the lift coefficients in the alpha = [-180°,+180°]
range
CD = [CD_values_Z5;CD_values_Z3;CD_values_Z1;CD_values_Z2;CD_values_Z4]; %
Vector containing all the drag coefficients in the alpha = [-180°,+180°]
range
POLAR_MATRIX = [AOA, CL, CD ]

% Writting extrapolated data into an EXCEL file
if re==1 % Location of the writing cells for each Re computation
    cell='B7';
```

```
elseif re==2
    cell='E7';
elseif re==3
    cell='H7';
elseif re==4
    cell='K7';
elseif re==5
    cell='N7';
elseif re==6
    cell='Q7';
elseif re==7
    cell='T7';
elseif re==8
    cell='W7';
elseif re==9
    cell='Z7';
elseif re==10
    cell='AC7';
 end

if strcmp(simulation,'3D')
xlswrite(file_write,POLAR_MATRIX,num2str(element),cell) % Writting data
into a EXCEL file for each airfoil
elseif strcmp(simulation,'2D')
%xlswrite('Airfoil_Data_360_Extrapolated_2D.xlsx',POLAR_MATRIX,airfoil,cell
) % Writting data into a unique EXCEL file
end

% Plotting the airfoil extrapolated 360 polars

if strcmp(simulation,'3D')
number=element;
elseif strcmp(simulation,'2D')
number=f;
end

figure (number)
subplot(1,2,1)
plot(AOA,CL)
hold on
axis ([-180 180 -1 1.7])
grid on
xlabel(' Angle of attack, \alpha [°]','FontSize',14);
ylabel(' Lift Coefficient, C_{L} ','FontSize',14);
str= ['LIFT COEFFICIENT  360° EXTRAPOLATION (',airfoil,' AIRFOIL)'];
title({str;''},'FontSize',12);

subplot (1,2,2)
plot (AOA,CD)
hold on
axis ([-180 180 0 1.4])
grid on
xlabel(' Angle of attack, \alpha [°]','FontSize',14);
ylabel(' Drag Coefficient, C_{D} ','FontSize',14);
str= ['DRAG COEFFICIENT  360° EXTRAPOLATION (',airfoil,' AIRFOIL)'];
title({str;''},'FontSize',12);

figure (number)
```

```
lgd=legend(' Re=0.1', ' Re=0.2', ' Re=0.3', ' Re=0.4', ' Re=0.5', '
Re=0.6',' Re=0.7', ' Re=0.8',  'Re=0.9', ' Re=1');
lgd.Title.String = 'REYNOLDS NUMBER (x10^6)';
lgd.NumColumns = 2;

end
end
end
```

A.4 Main_DESIGN.m

```
clear all
close all
clc
%% SCRIPT DESCRIPTION:
% This is one of the 2 main scripts in this work. In it, the BEMT algorithm
explained in Section 6.1 is implemented. Reading airfoil 2D polar data
convergence is achieved over the blade span discretised airfoil boundaries.
Twist angle at each section is calculated in order to achieve optimum angle
of attack at the section (obtaining maximum glide ratio (Cl/Cd). Optimum
chord is calculated for each iteration by EQ. (6.10) or (6.11).
% The output of this script is the radial position of blade boundaries, and
the chord length and twist angle and those points.

%% SCRIPT START

for f=1:4 % External loop for each of the 4 airfoils
    if f==1
        airfoil='S822';
        airfoil_data=xlsread('Airfoil_Data','S822','B7:AE204'); % Original
2D polars generated with XFOIL for S822 airfoil
        D_tip= 6.33; % Rotor diameter for the S822 airfoil layout blade [m]
    elseif f==2
        airfoil='S823';
        airfoil_data=xlsread('Airfoil_Data','S823','B7:AE206'); % "" S823
        D_tip= 6.92; % "" S823
    elseif f==3
        airfoil='S809';
        airfoil_data=xlsread('Airfoil_Data','S809','B7:AE205'); % "" S809
        D_tip= 7.12; % "" S809
    elseif f==4
        airfoil='SG6040';
        airfoil_data=xlsread('Airfoil_Data','SG6040','B7:AE205'); % ""
SG6040
        D_tip= 6.74;  % "" SG6040
    end

%% BLADE DESIGN PARAMETERS ( DESIGN VALUES, AIR PROPERTIES AND GEOMETRICAL
PARAMETERS) AND BLADE DISCRETISATION
U_des = 7.5; % Design wind velocity [m/s]
B = 3; % Number of blades of the HAWT
lambda_design = 6; % Design value of the Tip Speed Ratio, TSR
rho = 1.138; % Air density [kg/m3]
visc = 1.74e-5; % Air dynamic viscosity [Pa·s]
D_hub = 0.3 ; % Diameter of the hub [m]
R_hub=D_hub/2; % Hub radius [m]
R_tip = D_tip/2; % Rotor radius [m]

chord_distribution='A'; % Choose A or B chord distribution (see Section
6.2.2)

omega_mat(f,1)= lambda_design*U_des/R_tip; % Rotational speed at the design
point [rad/s], EQ.(6.1)

% BLADE DISCRETIZATION:
Nb=21; % Number of sections (boundaries) to discretize the blade span. N°
of segments/elements = Nb-1
```

```
rv(f,:)= linspace(D_hub/2,D_tip/2,Nb); % Generation of the radial position
vector: from hub radius to tip
rv_conv(f,:)=[rv(f,1)+0.01,rv(f,2:(end-1)),rv(f,end)-0.01]; % Radial
position vector to ensure convergence: innermost and outermost boundaries
moved 1 cm inside
element_width(f)=(rv(f,end)-rv(f,1))/(Nb-1); % Width of each blade element:
distance from 1 boundary to the other
counter=0; % Control counter: last value has to be equal to Nb

%% LOOP FOR CALCULATING PARAMETERS MOVING SPAN-WISE THROUGH THE BOUNDARIES
(SPAN-WISE LOOP)

for n=1:length(rv) % for each of the blade boundaries do...
omega=omega_mat(f,1);
a0=0; % First guess of axial induction factor = 0
r=rv_conv(f,n); % Radial position at the blade boundary [m]
a_p0=0; % First guess of tangential induction factor = 0
if n==1 || n==2 || n==3
c0=0.20; % 3 first boundaries are cylindrical
else
c0=1; % First guess for airfoil chord
end
lambda = omega*r/U_des; % Local speed ratio
sigma_r = B*c0/(2*pi*r); % Chord solidity at the section, EQ.(4.17)
W = sqrt(U_des^2*(1-a0)^2+omega^2*r^2*(1+a_p0)^2); % Relative velocity
[m/s], EQ.(4.7)
Re = W*c0*rho/visc; % Reynolds number, EQ.(3.4)
phi=atan(U_des*(1-a0)/(omega*r*(1+a_p0))); % Inflow angle [rad], EQ.(4.22)
% Getting aerodynamic coefficients:
if n==1 || n==2 || n==3  % Sections with circular foil
alpha=phi*180/pi;
C_l=0; % Lift coefficient
C_d=1.2; % Drag coefficient
else
% Alpha_opt function call
alpha = alpha_opt(Re,airfoil);
% CLift_and_CDrag function call
[C_l,C_d] = CLift_and_CDrag(Re,alpha,airfoil_data);
end
% Decomposition of the lift and drag coefficients into normal and
% tangential axes
C_n=C_l*cos(phi)+C_d*sin(phi); % EQ.(4.13)
C_t=C_l*sin(phi)-C_d*cos(phi); % EQ.(4.14)

% CALCULATION OF LOSS CORRECTION FACTORS
f_tip=B*(R_tip-r)/(2*r*sin(phi));
F_tip=2/pi*acos(exp(-f_tip)); % Prandtl Tip-loss correction factor,
EQ.(4.23)

f_hub=B*(r-R_hub)/(2*R_hub*sin(phi));
F_hub=2/pi*acos(exp(-f_hub)); %Hub-loss correction factor, EQ.(4.24)

F=F_tip*F_hub; % Overall correction factor, EQ.(4.25)

% CALCULATION OF INDUCTION FACTORS (GLAUERT CORRECTION IF NEEDED)

C_T=sigma_r*((1-a0)^2)*C_n/sin(phi)^2; % Thrust coefficient calculation,
EQ.(4.33)
if C_T>0.96*F % The element is considered highly loaded
```

```
    a1=(18*F-20-3*sqrt(C_T*(50-36*F)+12*F*(3*F-4)))/(36*F-50); % Buhl's
correction EQ.(4.32)
    C_T=8/9+(4*F-40/9)*a1+(50/9-4*F)*a1^2; % Buhl's correction EQ.(4.31)
else
a1=(1+4*F*sin(phi)^2/(sigma_r*C_n))^(-1); % EQ.(4.28)
C_T=4*a1*(1-a1)*F; % EQ.(4.30)
end
a_p1=(4*F*sin(phi)*cos(phi)/(sigma_r*C_t)-1)^(-1); % EQ.(4.29)

delta_a=abs(a0-a1); % Difference between a and a old
delta_a_p=abs(a_p0-a_p1); % Difference between a' and a' old

epsilon=10^-5; % Tolerance for convergence
w_relax= 0.25; % Relaxation factor

%% ITERATIVE PROCESS FOR EACH SINGLE RADIAL POSITION ( a and a' must
converge)
count=1; % Counter to see after how many iterations convergence is achieved
countmax=10^4; % Maximum number of iterations

while (max(delta_a,delta_a_p) > epsilon) && count<countmax % EQ.(6.15)
convergence criteria
    count=count+1;
    a_new=w_relax*a1+(1-w_relax)*a0; % EQ.(6.16)
    a_new_p=w_relax*a_p1+(1-w_relax)*a_p0; % EQ.(6.17)
    a0=a_new;
    a_p0=a_new_p;

W = sqrt(U_des^2*(1-a0)^2+omega^2*r^2*(1+a_p0)^2);
if n==1 || n==2 || n==3
c0=0.20;
elseif strcmp(chord_distribution,'A') % A chord distribution
     c_opt= (8*pi*r)*(1-cos(phi))/(B*C_l); % EQ.(6.10)
     c0=c_opt;
elseif strcmp(chord_distribution,'B') % B chord distribution
     c_opt=(8/9)/sqrt(4/9+lambda^2*(1+2/(9*lambda^2))^2); % EQ.(6.11)
     c_opt=c_opt*(2*pi*R_tip)/(B*lambda_design*C_l);
     c0=c_opt;
end
Re = W*c0*rho/visc;
phi=atan(U_des*(1-a0)/(omega*r*(1+a_p0)));
if n==1 || n==2 || n==3
alpha=phi*180/pi;
C_l=0;
C_d=1.2;
else
alpha=alpha_opt(Re,airfoil);
[C_l,C_d] = CLift_and_CDrag(Re,alpha,airfoil_data);
end
sigma_r = B*c0/(2*pi*r);

C_n=C_l.*cos(phi)+C_d.*sin(phi);
C_t=C_l.*sin(phi)-C_d.*cos(phi);

% CALCULATION OF LOSS CORRECTION FACTORS
f_tip=B*(R_tip-r)/(2*r*sin(phi));
F_tip=2/pi*acos(exp(-f_tip)); % Tip-loss correction factor

f_hub=B*(r-R_hub)/(2*R_hub*sin(phi));
```

```
F_hub=2/pi*acos(exp(-f_hub)); % Hub-loss correction factor

F=F_tip*F_hub; % Overall correction factor

% CALCULATION OF INDUCTION FACTORS (GLAUERT CORRECTION IF NEEDED)

C_T=sigma_r*((1-a0)^2)*C_n/sin(phi)^2; % Thrust coefficient calculation
if C_T>0.96*F % The element is considered highly loaded
       a1=(18*F-20-3*sqrt(C_T*(50-36*F)+12*F*(3*F-4)))/(36*F-50); % Buhl's
correction
       C_T=8/9+(4*F-40/9)*a1+(50/9-4*F)*a1^2; % Buhl's correction
else
a1=(1+4*F*sin(phi)^2/(sigma_r*C_n))^(-1);
C_T=4*a1*(1-a1)*F;
end
a_p1=(4*F*sin(phi)*cos(phi)/(sigma_r*C_t)-1)^(-1);

delta_a=abs(a0-a1);
delta_a_p=abs(a_p0-a_p1);
end

%% CONVERGENCE ACHIEVED FOR A RADIAL POSITION: CALCULATION OF AERODYNAMIC
FORCES/MOMENTS AND DATA SAVING
% Axial direction:
Elemental_Thrust=C_T*0.5*rho*U_des^2*2*pi*r; % Elemental thrust in the
turbine annulus [N/m], EQ.(6.18)
Elemental_Axial_Load=Elemental_Thrust/3; % Axial load in each blade element
[N/m], EQ.(6.20)
Root_Flapwise_Bending_Moment=(Elemental_Axial_Load)*(r-R_hub); % Elemental
flapwise bending moment respect to the blade root [N.m/m], EQ.(6.23)

% Tangential direction
Elemental_Torque=4*F*pi*rho*U_des*(omega*r)*a_p1*(1-a1)*r^2; % Elemental
torque in the turbine annulus [N.m/m], EQ.(6.19)
Elemental_Tangential_Load=Elemental_Torque/(B*r); % Tangential load in each
blade element [N/m], EQ.(6.21)
Elemental_Torque_1blade=Elemental_Torque/3; % Torque generated by each
elemental tangential load (1 blade) [N.m/m], EQ.(6.22)

% Data for each radial position is stored in matrixes ( f rows x Nb
columns)
count_mat(f,n)=count;
glide_ratio(f,n)=C_l/C_d; % Glide ratio at blade setions
a(f,n)=a1; % Axial induction coefficient
a_p(f,n)=a_p1; % Tangential induction coefficient
W_mat(f,n)=W; % Relative velocity at section
c(f,n)=c0; % Chord length matrix
Re_mat(f,n)=Re; % Re number matrix
alpha_mat(f,n)=alpha; % Angle of attack matrix
phi_mat(f,n)=phi*180/pi; % Inflow angle (in degrees)
beta_mat(f,n)=phi_mat(f,n)-alpha_mat(f,n); %angolo di calettamento
F_mat(f,n)=F; % Overall loss factor

C_L_mat(f,n)=C_l; % Lift coefficient matrix
C_D_mat(f,n)=C_d; % Drag coefficient matrix
CT_mat(f,n)=C_T; % Thrust coefficient matrix

Thrust_mat(f,n)=Elemental_Thrust; % Elemental thrust matrix
Axial_Load_mat(f,n)=Elemental_Axial_Load; % Elemental axial load matrix
```

```
RFWBM_mat(f,n)=Root_Flapwise_Bending_Moment; % Element contribution to Root
Flap-wise Bending Moment matrix

Torque_mat(f,n)=Elemental_Torque; % Elemental torque matrix
Tangential_Load_mat(f,n)=Elemental_Tangential_Load; % Elemental tangential
load matrix
Torque_1blade_mat(f,n)=Elemental_Torque_1blade; % Elemental torque by each
blade matrix

counter=counter+1; % Blade section counter
end
P_available(f)=0.5*rho*U_des^3*pi*rv(f,end)^2; % Available power in wind
for each design [W], EQ.(2.2)

%% END OF THE FOR LOOP THAT RUNS SPAN WISE

end

%% ALL ITERATIONS ARE FINISHED AND DATA IS ALREADY WRITTEN IN MATRIXES
%% DATA IS MOVED FROM MATRIXES TO VECTORS TO AVOID INDEXING

% S822 AIRFOIL LAYOUT
glide_ratio_S822=glide_ratio(1,:)';
a_S822=a(1,:)';
a_p_S822=a_p(1,:)';
W_S822=W_mat(1,:)';
chord_S822=c(1,:)';
Re_S822=Re_mat(1,:)';
alpha_S822=alpha_mat(1,:)';
phi_S822=phi_mat(1,:)';
beta_S822=beta_mat(1,:)';
F_S822=F_mat(1,:)';
CL_S822=C_L_mat(1,:)';
CD_S822=C_D_mat(1,:)';
CT_S822=CT_mat(1,:)';
Thrust_S822=Thrust_mat(1,:)';
Axial_Load_S822=Axial_Load_mat(1,:)';
Rfwbm_S822=RFWBM_mat(1,:)';
Torque_S822=Torque_mat(1,:)';
Tangential_Load_S822=Tangential_Load_mat(1,:)';
Torque_1blade_S822=Torque_1blade_mat(1,:)';
rv_S822=rv(1,:)';
rv_conv_S822=rv_conv(1,:)';
R_tip_S822=rv_S822(end);

% S823 AIRFOIL LAYOUT
glide_ratio_S823=glide_ratio(2,:)';
a_S823=a(2,:)';
a_p_S823=a_p(2,:)';
W_S823=W_mat(2,:)';
chord_S823=c(2,:)';
Re_S823=Re_mat(2,:)';
alpha_S823=alpha_mat(2,:)';
phi_S823=phi_mat(2,:)';
beta_S823=beta_mat(2,:)';
F_S823=F_mat(2,:)';
CL_S823=C_L_mat(2,:)';
CD_S823=C_D_mat(2,:)';
CT_S823=CT_mat(2,:)';
```

```
Thrust_S823=Thrust_mat(2,:)';
Axial_Load_S823=Axial_Load_mat(2,:)';
Rfwbm_S823=RFWBM_mat(2,:)';
Torque_S823=Torque_mat(2,:)';
Tangential_Load_S823=Tangential_Load_mat(2,:)';
Torque_1blade_S823=Torque_1blade_mat(2,:)';
rv_S823=rv(2,:)';
rv_conv_S823=rv_conv(2,:)';
R_tip_S823=rv_S823(end);

% S809 AIRFOIL LAYOUT
glide_ratio_S809=glide_ratio(3,:)';
a_S809=a(3,:)';
a_p_S809=a_p(3,:)';
W_S809=W_mat(3,:)';
chord_S809=c(3,:)';
Re_S809=Re_mat(3,:)';
alpha_S809=alpha_mat(3,:)';
phi_S809=phi_mat(3,:)';
beta_S809=beta_mat(3,:)';
F_S809=F_mat(3,:)';
CL_S809=C_L_mat(3,:)';
CD_S809=C_D_mat(3,:)';
CT_S809=CT_mat(3,:)';
Thrust_S809=Thrust_mat(3,:)';
Axial_Load_S809=Axial_Load_mat(3,:)';
Rfwbm_S809=RFWBM_mat(3,:)';
Torque_S809=Torque_mat(3,:)';
Tangential_Load_S809=Tangential_Load_mat(3,:)';
Torque_1blade_S809=Torque_1blade_mat(3,:)';
rv_S809=rv(3,:)';
rv_conv_S809=rv_conv(3,:)';
R_tip_S809=rv_S809(end);

% SG6040 AIRFOIL LAYOUT
glide_ratio_SG6040=glide_ratio(4,:)';
a_SG6040=a(4,:)';
a_p_SG6040=a_p(4,:)';
W_SG6040=W_mat(4,:)';
chord_SG6040=c(4,:)';
Re_SG6040=Re_mat(4,:)';
alpha_SG6040=alpha_mat(4,:)';
phi_SG6040=phi_mat(4,:)';
beta_SG6040=beta_mat(4,:)';
F_SG6040=F_mat(4,:)';
CL_SG6040=C_L_mat(4,:)';
CD_SG6040=C_D_mat(4,:)';
CT_SG6040=CT_mat(4,:)';
Thrust_SG6040=Thrust_mat(4,:)';
Axial_Load_SG6040=Axial_Load_mat(4,:)';
Rfwbm_SG6040=RFWBM_mat(4,:)';
Torque_SG6040=Torque_mat(4,:)';
Tangential_Load_SG6040=Tangential_Load_mat(4,:)';
Torque_1blade_SG6040=Torque_1blade_mat(4,:)';
rv_SG6040=rv(4,:)';
rv_conv_SG6040=rv_conv(4,:)';
R_tip_SG6040=rv_SG6040(end);

%% PLOTTING OF ALL THE VARIABLES AGAINST THE RADIAL POSITION OF THE ELEMENT
```

```
col_radius_S822=rv_conv(1,:)'; % Radial position in columns
col_radius_S823=rv_conv(2,:)';
col_radius_S809=rv_conv(3,:)';
col_radius_SG6040=rv_conv(4,:)';

r_R_S822=rv_conv_S822/R_tip_S822; % Radial position divided by rotor radius
(Rtip)
r_R_S823=rv_conv_S823/R_tip_S823;
r_R_S809=rv_conv_S809/R_tip_S809;
r_R_SG6040=rv_conv_SG6040/R_tip_SG6040;

%% FIGURE 1: Axial and tangential induction factors
font='LM Roman 12';
figure('DefaultTextFontName', font, 'DefaultAxesFontName',
font,'DefaultAxesFontSize',20,'DefaultTextFontSize',20)
subplot(2,1,1)
plot (r_R_S822,a_S822,'ro-')
hold on;
plot (r_R_S823,a_S823,'bs-')
hold on;
plot (r_R_S809,a_S809,'kd-')
hold on
plot (r_R_SG6040,a_SG6040,'m^-')
grid on
xlabel('Radial position, r/R ')
ylabel(' Axial induction factor, a')
lgd=legend('S822','S823','S809','SG 6040','location','southeast');
lgd.NumColumns=2;
lgd.Title.String='BLADE LAYOUT';

subplot(2,1,2)
plot (r_R_S822,a_p_S822,'ro-')
hold on
plot (r_R_S823,a_p_S823,'bs-')
hold on
plot (r_R_S809,a_p_S809,'kd-')
hold on
plot (r_R_SG6040,a_p_SG6040,'m^-')
grid on
xlabel('Radial position, r/R ')
ylabel('Tangential induction factor, a´')

%% FIGURE 2:  Chord length and Twist angle
figure('DefaultTextFontName', font, 'DefaultAxesFontName',
font,'DefaultAxesFontSize',20,'DefaultTextFontSize',20)
subplot(2,1,1)
plot (r_R_S822,chord_S822,'ro-')
hold on
plot (r_R_S823,chord_S823,'bs-')
hold on
plot (r_R_S809,chord_S809,'kd-')
hold on
plot (r_R_SG6040,chord_SG6040,'m^-')
grid on
xlabel('Radial position, r/R ')
ylabel('Chord length [m]')
lgd=legend('S822','S823','S809','SG 6040','location','northeast');
lgd.NumColumns=2;
```

```
lgd.Title.String='BLADE LAYOUT';

subplot(2,1,2)
plot (r_R_S822,beta_S822,'ro-')
hold on
plot (r_R_S823,beta_S823,'bs-')
hold on
plot (r_R_S809,beta_S809,'kd-')
hold on
plot (r_R_SG6040,beta_SG6040,'m^-')
grid on
xlabel('Radial position, r/R ')
ylabel(' Twist angle, \beta [°]')

%% FIGURE: 3 Inflow angle, angle of attack, glide-ratio, relative velocity,
Re number and loss factor
figure('DefaultTextFontName', font, 'DefaultAxesFontName',
font,'DefaultAxesFontSize',14,'DefaultTextFontSize',14)
subplot(3,2,1) % phi vs r/R
plot (r_R_S822,phi_S822,'ro-')
hold on
plot (r_R_S823,phi_S823,'bs-')
hold on
plot (r_R_S809,phi_S809,'kd-')
hold on
plot (r_R_SG6040,phi_SG6040,'m^-')
grid on
xlabel('Radial position, r/R ')
ylabel('Inflow angle, \phi [°] ')
lgd=legend('S822','S823','S809','SG 6040','location','northeast');
lgd.NumColumns=2;
lgd.Title.String='BLADE LAYOUT';

subplot(3,2,3) % alpha vs r/R
plot (r_R_S822(4:end,1),alpha_S822(4:end,1),'ro-')
hold on
plot (r_R_S823(4:end,1),alpha_S823(4:end,1),'bs-')
hold on
plot (r_R_S809(4:end,1),alpha_S809(4:end,1),'kd-')
hold on
plot (r_R_SG6040(4:end,1),alpha_SG6040(4:end,1),'m^-')
grid on
xlabel('Radial position, r/R ')
ylabel('Angle of attack, \alpha [°] ')
grid on

subplot(3,2,5) % Glide-ratio vs r/R
plot (r_R_S822(4:end,1),glide_ratio_S822(4:end,1),'ro-')
hold on
plot (r_R_S823(4:end,1),glide_ratio_S823(4:end,1),'bs-')
hold on
plot (r_R_S809(4:end,1),glide_ratio_S809(4:end,1),'kd-')
hold on
plot (r_R_SG6040(4:end,1),glide_ratio_SG6040(4:end,1),'m^-')
grid on
xlabel('Radial position, r/R ')
ylabel('Glide ratio, C_l/C_d')
axis([0 1 40 110])
```

```
subplot(3,2,2) % W vs r/R
plot (r_R_S822,W_S822,'ro-')
hold on
plot (r_R_S823,W_S823,'bs-')
hold on
plot (r_R_S809,W_S809,'kd-')
hold on
plot (r_R_SG6040,W_SG6040,'m^-')
grid on
xlabel('Radial position, r/R ')
ylabel('Rel. velocity, W [m/s] ')

subplot(3,2,4) % Re vs r/R
plot (r_R_S822,Re_S822,'ro-')
hold on
plot (r_R_S823,Re_S823,'bs-')
hold on
plot (r_R_S809,Re_S809,'kd-')
hold on
plot (r_R_SG6040,Re_SG6040,'m^-')
grid on
xlabel('Radial position, r/R ')
ylabel('Re number');

subplot(3,2,6) % Loss factor vs r/R
plot (r_R_S822,F_S822,'ro-')
hold on
plot (r_R_S823,F_S823,'bs-')
hold on
plot (r_R_S809,F_S809,'kd-')
hold on
plot (r_R_SG6040,F_SG6040,'m^-')
grid on
xlabel('Radial position, r/R ')
ylabel('Loss factor, F')

%% FIGURE 4 Lift, drag and thrust coefficients
figure('DefaultTextFontName', font, 'DefaultAxesFontName', font,'DefaultA-
xesFontSize',17,'DefaultTextFontSize',17)
subplot(3,1,1) % Cl vs r/R
plot (r_R_S822(4:end,1),CL_S822(4:end,1),'ro-')
hold on
plot (r_R_S823(4:end,1),CL_S823(4:end,1),'bs-')
hold on
plot (r_R_S809(4:end,1),CL_S809(4:end,1),'kd-')
hold on
plot (r_R_SG6040(4:end,1),CL_SG6040(4:end,1),'m^-')
grid on
xlabel('Radial position, r/R ')
ylabel('Lift Coefficient, C_l')
axis([0 1 0.5 1.2])
lgd=legend('S822','S823','S809','SG 6040','location','SOUTHWEST');
lgd.NumColumns=2;
lgd.Title.String='BLADE LAYOUT';

subplot(3,1,2) % Cd vs r/R
plot (r_R_S822(4:end,1),CD_S822(4:end,1),'ro-')
hold on
plot (r_R_S823(4:end,1),CD_S823(4:end,1),'bs-')
```

```
hold on
plot (r_R_S809(4:end,1),CD_S809(4:end,1),'kd-')
hold on
plot (r_R_SG6040(4:end,1),CD_SG6040(4:end,1),'m^-')
grid on
xlabel('Radial position, r/R ')
ylabel('Drag Coefficient, C_d')
axis([0 1 0 0.02])

subplot(3,1,3) % CT vs r/R
plot (r_R_S822,CT_S822,'ro-')
hold on
plot (r_R_S823,CT_S823,'bs-')
hold on
plot (r_R_S809,CT_S809,'kd-')
hold on
plot (r_R_SG6040,CT_SG6040,'m^-')
grid on
xlabel('Radial position, r/R ')
ylabel('Thrust Coefficient, C_T')

%% FIGURE 5 % Aerodynamic Loads and Moments over the blade span
figure('DefaultTextFontName', font, 'DefaultAxesFontName', font,'DefaultA-
xesFontSize',14,'DefaultTextFontSize',14)
subplot(3,2,1) % Axial Load vs r/R
plot (r_R_S822,Axial_Load_S822,'ro-')
hold on
plot (r_R_S823,Axial_Load_S823,'bs-')
hold on
plot (r_R_S809,Axial_Load_S809,'kd-')
hold on
plot (r_R_SG6040,Axial_Load_SG6040,'m^-')
grid on
xlabel('Radial position, r/R ')
ylabel('Axial load, F_a [N/m]')
lgd=legend('S822','S823','S809','SG 6040','location','southeast');
lgd.NumColumns=2;
lgd.Title.String='BLADE LAYOUT';

subplot(3,2,3) % RFWBM vs r/R
plot (r_R_S822,Rfwbm_S822,'ro-')
hold on
plot (r_R_S823,Rfwbm_S823,'bs-')
hold on
plot (r_R_S809,Rfwbm_S809,'kd-')
hold on
plot (r_R_SG6040,Rfwbm_SG6040,'m^-')
grid on
xlabel('Radial position, r/R ')
ylabel('RFWBM [N·m/m]')

subplot(3,2,5) % Annular Thrust vs r/R
plot (r_R_S822,Thrust_S822,'ro-')
hold on
plot (r_R_S823,Thrust_S823,'bs-')
hold on
plot (r_R_S809,Thrust_S809,'kd-')
hold on
plot (r_R_SG6040,Thrust_SG6040,'m^-')
```

```
grid on
xlabel('Radial position, r/R ')
ylabel({'Annular Thrust';'[N/m]'})

subplot(3,2,2) % Tangential load vs r/R
plot (r_R_S822,Tangential_Load_S822,'ro-')
hold on
plot (r_R_S823,Tangential_Load_S823,'bs-')
hold on
plot (r_R_S809,Tangential_Load_S809,'kd-')
hold on
plot (r_R_SG6040,Tangential_Load_SG6040,'m^-')
grid on
axis([0 1 -10 25])
xlabel('Radial position, r/R ')
ylabel({'Tangential load, F_t'; '[N/m]'})

subplot(3,2,4) % Element Torque vs r/R
plot (r_R_S822,Torque_1blade_S822,'ro-')
hold on
plot (r_R_S823,Torque_1blade_S823,'bs-')
hold on
plot (r_R_S809,Torque_1blade_S809,'kd-')
hold on
plot (r_R_SG6040,Torque_1blade_SG6040,'m^-')
grid on
xlabel('Radial position, r/R ')
ylabel({'Element Torque'; '[N·m/m]'})
axis([0 1 -10 80])

subplot(3,2,6) % Annular Torque vs r/R
plot (r_R_S822,Torque_S822,'ro-')
hold on
plot (r_R_S823,Torque_S823,'bs-')
hold on
plot (r_R_S809,Torque_S809,'kd-')
hold on
plot (r_R_SG6040,Torque_SG6040,'m^-')
grid on
xlabel('Radial position, r/R ')
ylabel({'Annular Torque'; '[N·m/m]'})
axis([0 1 -20 240])

%% TORQUE, THRUST, RFWBM, POWER AND CP CALCULATION (DESIGN POINT)
% TORQUE [N·m]:
TORQUE_S822 = trapz(col_radius_S822,Torque_S822); % Trapezoidal numerical
integration of annular torque EQ.(6.25)
TORQUE_S822=TORQUE_S822+0.01*(Torque_S822(end)+Torque_S822(1))/2 % Contri-
bution of the cm at tip and hub
TORQUE_S823= trapz(col_radius_S823,Torque_S823);
TORQUE_S823=TORQUE_S823+0.01*(Torque_S823(end)+Torque_S823(1))/2
TORQUE_S809= trapz(col_radius_S809,Torque_S809);
TORQUE_S809=TORQUE_S809+0.01*(Torque_S809(end)+Torque_S809(1))/2
TORQUE_SG6040= trapz(col_radius_SG6040,Torque_SG6040);
TORQUE_SG6040=TORQUE_SG6040+0.01*(Torque_SG6040(end)+Torque_SG6040(1))/2
% THRUST [N]:
THRUST_S822 = trapz(col_radius_S822,Thrust_S822); % Trapezoidal numerical
integration of annular thrust EQ.(6.24)
THRUST_S822=THRUST_S822+0.01*(Thrust_S822(end)+Thrust_S822(1))/2
```

```
THRUST_S823 = trapz(col_radius_S823,Thrust_S823);
THRUST_S823=THRUST_S823+0.01*(Thrust_S823(end)+Thrust_S823(1))/2
THRUST_S809 = trapz(col_radius_S809,Thrust_S809);
THRUST_S809=THRUST_S809+0.01*(Thrust_S809(end)+Thrust_S809(1))/2
THRUST_SG6040 = trapz(col_radius_SG6040,Thrust_SG6040);
THRUST_SG6040=THRUST_SG6040+0.01*(Thrust_SG6040(end)+Thrust_SG6040(1))/2

% Mechanical Power computation  EQ.(6.27) [W]
Power_S822=omega_mat(1)*TORQUE_S822
Power_S823=omega_mat(2)*TORQUE_S823
Power_S809=omega_mat(3)*TORQUE_S809
Power_SG6040=omega_mat(4)*TORQUE_SG6040

% Power coefficient (without losses)
Cp_S822=Power_S822/P_available(1);
Cp_S823=Power_S823/P_available(2);
Cp_S809=Power_S809/P_available(3);
Cp_SG6040=Power_SG6040/P_available(4);

% Root Flap-Wise Bending Moment [N.m]
RFWBM_S822=trapz(col_radius_S822,Rfwbm_S822);
RFWBM_S822=RFWBM_S822+0.01*(Rfwbm_S822(end)+Rfwbm_S822(1))/2
RFWBM_S823=trapz(col_radius_S823,Rfwbm_S823);
RFWBM_S823=RFWBM_S823+0.01*(Rfwbm_S823(end)+Rfwbm_S823(1))/2
RFWBM_S809=trapz(col_radius_S809,Rfwbm_S809);
RFWBM_S809=RFWBM_S809+0.01*(Rfwbm_S809(end)+Rfwbm_S809(1))/2
RFWBM_SG6040=trapz(col_radius_SG6040,Rfwbm_SG6040);
RFWBM_SG6040=RFWBM_SG6040+0.01*(Rfwbm_SG6040(end)+Rfwbm_SG6040(1))/2

%% GEOMETRIC EXTRAPOLATION OF INNERMOST AND OUTERMOST STATIONS
% Primary declared blade first and last stations are stablished by
% extrapolation of data. Direct design is not posible due to the fact that
% Floss equals 0 in the extreme positions (a=1 a_p=-1).

% The first station, as it is circular, is mantained equal.
% The last station is modified as follows:

for f=1:4
    if f==1
     chord=chord_S822;
     beta_mat=beta_S822;
     airfoil='S822';
    elseif f==2
     chord=chord_S823;
     beta_mat=beta_S823;
     airfoil='S823';
     elseif f==3
     chord=chord_S809;
     beta_mat=beta_S809;
     airfoil='S809';
     elseif f==4
     chord=chord_SG6040;
     beta_mat=beta_SG6040;
     airfoil='SG6040';
    end
 R_tip=rv(f,end);
% 3rd degree polynomic curves to extrapolate chord and twist

pchord=polyfit(rv_conv(f,end-5:end),chord(end-5:end)',3);
```

```
f1=@(xx) polyval(pchord,xx);

ptwist=polyfit(rv_conv(f,end-5:end),beta_mat(end-5:end)',3);
f2=@(xx) polyval(ptwist,xx);

fig=figure('DefaultTextFontName', font, 'DefaultAxesFontName', font,'De-
faultAxesFontSize',14,'DefaultTextFontSize',14);% str=['OUTERMOST BLADE
STATION CHORD AND TWIST EXTRAPOLATION (',profilo,')'];
% title({str;''})
subplot(1,2,1)
fplot(f1,[R_tip-1,R_tip+0.1])
hold on
plot(rv_conv(f,end-5:end),chord(end-5:end)','bx')
hold on
chord(end)=f1(R_tip);
plot(R_tip,chord(end),'ro')
xlabel(' Radial position [m]')
ylabel('Chord [m]')

subplot(1,2,2)
fplot(f2,[R_tip-1,R_tip+0.1])
hold on
plot(rv_conv(f,end-5:end),beta_mat(end-5:end)','bx')
hold on
beta_mat(end)=f2(R_tip);
plot(R_tip,beta_mat(end),'ro')
xlabel(' Radial position [m]')
ylabel('Twist angle, \beta [°]')

str=['OUTERMOST BLADE STATION CHORD AND TWIST EXTRAPOLATION (',air-
foil,')'];
sgtitle({str;''})
legend('Extrapolation function','Original data','Extrapolated point')

% Writing extrapolated geometry into .txt files
    if f==1
    chord_S822=chord;
    beta_S822=beta_mat;
    save Chord_S822.txt chord_S822 -ASCII -TABS
    save Beta_S822.txt beta_S822 -ASCII -TABS
    save Radial_position_S822.txt rv_S822 -ASCII -TABS
    elseif f==2
    chord_S823=chord;
    beta_S823=beta_mat;
    save Chord_S823.txt chord_S823 -ASCII -TABS
    save Beta_S823.txt beta_S823 -ASCII -TABS
    save Radial_position_S823.txt rv_S823 -ASCII -TABS
    elseif f==3
    chord_S809=chord;
    beta_S809=beta_mat;
    save Chord_S809.txt chord_S809 -ASCII -TABS
    save Beta_S809.txt beta_S809 -ASCII -TABS
    save Radial_position_S809.txt rv_S809 -ASCII -TABS
    elseif f==4
    chord_SG6040=chord;
    beta_SG6040=beta_mat;
    save Chord_SG6040.txt chord_SG6040 -ASCII -TABS
    save Beta_SG6040.txt beta_SG6040 -ASCII -TABS
    save Radial_position_SG6040.txt rv_SG6040 -ASCII -TABS
```

```
    end
end
 %% FOLDED PROJECTED AREA CALCULATION
 % To asses the dimension of each blade
 for f=1:4
chord_vector=[ c(f,1); 2*c(f,2:(end-1))'; c(f,end)];
suma(f)=sum(chord_vector);
Sfolded(f)=suma(f)*element_width(f); % Folded projected area of the blade
[m2]
 end
Sfolded_S822=Sfolded(1)
Sfolded_S823=Sfolded(2)
Sfolded_S809=Sfolded(3)
Sfolded_SG6040=Sfolded(4)
% Generation of a bar diagram for the projected area
c = categorical({'S822','S823','S809','SG6040'});
figure('DefaultTextFontName', font, 'DefaultAxesFontName', font,'DefaultA-
xesFontSize',12,'DefaultTextFontSize',12);
bar(c,Sfolded,'FaceColor','0.30,0.75,0.93')
xlabel('Blade layout')
ylabel('Folded Projected Area [m^2]' )
```

A.5 Alpha_opt.m

```
function [alpha_opt] = alpha_opt(Re,profilo)

%% DESCRIPTION:
% The aim of this function is to identify which is the optimum angle of
attack for each airfoil for a given Re number. The optimum angle of attack
is the one that gives the maximum glide ratio (Cl/Cd).
% Low Re airfoils are sensible to Re number so the value of alpha_opt
changes within the Re number. Therefore, optimum value of alpha will change
in the different blade stations. Alpha optimum value decreases when Re
increaes.

if strcmp(profilo,'S822')
    if Re < 100000 % Lower limit for the Re number
        alpha_opt = 8.75; % Optimum value of angle of attack
    elseif Re > 1200000 % Upper limit for the Re number
        alpha_opt = 4.5 ;
    else
        x = [ 100000, 200000, 300000, 400000, 500000, 600000, 700000,
800000, 900000, 1000000, 1100000, 1200000 ]; % Re values in a vector
        v = [ 8.25, 7, 6.75, 6, 5.75, 5.5, 5.25, 5, 4.75, 4.75, 4.5, 4.5 ];
% alpha_opt value for each Re number. Identified from Cl/Cd vs alpha graph
        alpha_opt = interp1(x,v,Re,'linear'); % Linear interpolation
between Re numbers
    end

elseif strcmp(profilo,'S823')
    if Re < 100000
        alpha_opt = 9.5 ;
    elseif Re > 1200000
        alpha_opt = 6 ;
    else
        x = [ 100000, 200000, 300000, 400000, 500000, 600000, 700000,
800000, 900000, 1000000, 1100000, 1200000 ];
        v = [ 9.5, 8, 7.5, 7.5, 6.75, 7, 6.75, 6.5, 6.25, 6.25, 6, 6 ];
        alpha_opt = interp1(x,v,Re,'linear');
    end

elseif strcmp(profilo,'S809')
    if Re < 100000
        alpha_opt = 10 ;
    elseif Re > 1200000
        alpha_opt = 5.5 ;
    else
        x = [ 100000, 200000, 300000, 400000, 500000, 600000, 700000,
800000, 900000, 1000000, 1100000, 1200000 ];
        v = [ 10, 8.25, 7.5, 7.25, 7, 6.5, 6.25, 6.25, 6, 5.75, 5.75,
5.5 ];
        alpha_opt = interp1(x,v,Re,'linear');
    end

elseif strcmp(profilo,'SG6040')
    if Re < 100000
        alpha_opt = 9 ;
    elseif Re > 1200000
        alpha_opt = 5.25 ;
```

```
    else
        x = [ 100000, 200000, 300000, 400000, 500000, 600000, 700000,
800000, 900000, 1000000, 1100000, 1200000 ];
        v = [ 9, 7.75, 7.5, 7, 6.75, 6.5, 6.75, 6, 6.25, 6, 5.25, 5.25 ];
        alpha_opt = interp1(x,v,Re,'linear');
    end
end
end
```

A.6 Clift_and_CDrag.m

```
function [C_lift,C_drag] = CLift_and_CDrag(Re,alpha,airfoil_data)

%% DESCRIPTION:
% This function gives back the lift and drag coefficients for a Re number
and an angle of attack.
% Inside this function, the function CL_and_CD is called, which returns
lift and drag coefficients at the desired angle of attack for the upper and
lower closest 100000-multiple Re numbers. Then,this function performs a
linear interpolation between Re numbers.

if Re >= 100000 && Re <= 1000000
    Re1 = floor(Re/100000)*100000;
    Re2 = ceil(Re/100000)*100000;

    if Re1 == Re2 %in this case we do not need an interpoltion i.e
Re=500000
    [C_lift,C_drag] = CL_and_CD(Re1,alpha,airfoil_data);

    else
   % Evaluation of CL and CD at Re1 and Re2
   [CL_Re1,CD_Re1]=CL_and_CD(Re1,alpha,airfoil_data); % CL_and_CD function
call
   [CL_Re2,CD_Re2]=CL_and_CD(Re2,alpha,airfoil_data); % CL_and_CD function
call

   % Linear interpolation in order to find the values of CL and CD at the
   % desired Reynolds number
   % Vector creation:
   reynolds = [Re1, Re2];
   lift = [CL_Re1, CL_Re2];
   drag = [CD_Re1, CD_Re2];

   C_lift=interp1(reynolds,lift,Re,'linear'); % Linear interpolation
between Re numbers
   C_drag=interp1(reynolds,drag,Re,'linear'); % Linear interpolation
between Re numbers
    end

elseif Re < 100000 % Lower Re number limit
    [C_lift,C_drag] = CL_and_CD(100000,alpha,airfoil_data);

elseif Re > 1000000 % Upper Re number limit
    [C_lift,C_drag] = CL_and_CD(1000000,alpha,airfoil_data);
end
end
```

A.7 CL_and_CD.m

```
function [C_l,C_d] = CL_and_CD(Re,alpha,airfoil_data)

%% DESCRIPTION:
% This function gives back the lift and drag coefficients for Re numbers
which are multiples of 100 000. It reads the angle of attack and computes a
linear interpolation in order to find lift and drag.
% This function is called from the function CLift_and_CDrag, which
interpolates lift and drag coefficients between 2 Re numbers.

if Re == 100000 || Re == 200000 || Re == 300000 || Re == 400000 || Re ==
500000 || Re == 600000 || Re == 700000 || Re == 800000 || Re == 900000 ||
Re == 1000000

    if Re == 100000
          data_3columns=airfoil_data(:,1:3); % First block of airfoil data:
3 columns: angle of attack, lift coefficient and drag coefficient
          data_3columns= data_3columns(all(~isnan(data_3columns),2),:); %
removal of NaN values from the matrix

    elseif Re == 200000
          data_3columns=airfoil_data(:,4:6);
          data_3columns= data_3columns(all(~isnan(data_3columns),2),:);

    elseif Re == 300000
        data_3columns=airfoil_data(:,7:9);
        data_3columns= data_3columns(all(~isnan(data_3columns),2),:);

    elseif Re == 400000
        data_3columns=airfoil_data(:,10:12);
        data_3columns= data_3columns(all(~isnan(data_3columns),2),:);

    elseif Re == 500000
        data_3columns=airfoil_data(:,13:15);
        data_3columns= data_3columns(all(~isnan(data_3columns),2),:);

    elseif Re == 600000
        data_3columns=airfoil_data(:,16:18);
        data_3columns= data_3columns(all(~isnan(data_3columns),2),:);

    elseif Re == 700000
        data_3columns=airfoil_data(:,19:21);
        data_3columns= data_3columns(all(~isnan(data_3columns),2),:);

    elseif Re == 800000
        data_3columns=airfoil_data(:,22:24);
        data_3columns= data_3columns(all(~isnan(data_3columns),2),:);

    elseif Re == 900000
        data_3columns=airfoil_data(:,25:27);
        data_3columns= data_3columns(all(~isnan(data_3columns),2),:);

    elseif Re == 1000000
        data_3columns=airfoil_data(:,28:30);
        data_3columns= data_3columns(all(~isnan(data_3columns),2),:);

    end
        angle_of_attack = data_3columns(:,1); % First column of data block
```

```
        lift_coefficient = data_3columns(:,2); % Second column of data
block
        drag_coefficient = data_3columns(:,3);  % Third column of data
block
end

if alpha<=angle_of_attack(1)
    C_l = lift_coefficient(1);
    C_d= drag_coefficient(1);
elseif alpha>=angle_of_attack(end)
    C_l = lift_coefficient(end);
    C_d= drag_coefficient(end);
else
    C_l = interp1(angle_of_attack,lift_coefficient,alpha,'linear'); %
Linear interpolation between angles of attack
    C_d = interp1(angle_of_attack,drag_coefficient,alpha,'linear'); %
Linear interpolation between angles of attack

end
end
```

A.8 Drawing.m

```
clear all
close all
clc
%% SCRIPT DESCRIPTION:
% Loading the radial position, chord length and twist angle of the blade
designs, as well as the coordinates of the airfoils, this script scales and
rotates the coordinate points and places them at the proper radial posi-
tion, leading to both a 3D view and a section polar plot of the blade's la-
yout.
% It also traslates this points into XYZ coordinates in order to export to
SolidWorks the desired final design.

%% SCRIPT START
%% Loading .txt files (Airfoil coordinates,chord and twist values)
for f=1:4
if f==1
profilo='S822';
load Coordinates_S822.txt; % Coordinates for the selected profile
Coordinates=Coordinates_S822;
load Chord_S822.txt; % Chord values for the the chosen design
Chord=Chord_S822;
load Beta_S822.txt; % Twist angle values for the chosen design
Beta=Beta_S822;
load Radial_position_S822.txt
rv=Radial_position_S822;
elseif f==2
profilo='S822';
load Coordinates_S823.txt; % Coordinates for the selected profile
Coordinates=Coordinates_S823;
load Chord_S823.txt; % Chord values for the the chosen design
Chord=Chord_S823;
load Beta_S823.txt; % Twist angle values for the chosen design
Beta=Beta_S823;
load Radial_position_S823.txt
rv=Radial_position_S823;
elseif f==3
profilo='S809';
load Coordinates_S809.txt; % Coordinates for the selected profile
Coordinates=Coordinates_S809;
load Chord_S809.txt; % Chord values for the the chosen design
Chord=Chord_S809;
load Beta_S809.txt; % Twist angle values for the chosen design
Beta=Beta_S809;
load Radial_position_S809.txt
rv=Radial_position_S809;
elseif f==4
profilo='SG6040';
load Coordinates_SG6040.txt; % Coordinates for the selected profile
Coordinates=Coordinates_SG6040;
load Chord_SG6040.txt; % Chord values for the the chosen design
Chord=Chord_SG6040;
load Beta_SG6040.txt; % Twist angle values for the chosen design
Beta=Beta_SG6040;
load Radial_position_SG6040.txt
rv=Radial_position_SG6040;
end
```

```
%% UNITARY CIRCLE COORDINATES GENERATOR (CENTER AT X=0.5): 66 VALUES FOR
S8XX, 81 FOR SG6040
if f==4 % Airfoil= SG6040
    a=42;
else
    a=33;
end
pos_up= Coordinates(1:a,1);
pos_down= Coordinates(a+1:end,1);
% Circle mathematic continuous function
f1=@(x) sqrt(0.5.^2-(x-0.5).^2);
f2=@(x) -sqrt(0.5.^2-(x-0.5).^2);
positives=f1(pos_up); % Evaluation of the function in the x-positions of
the airfoil coordinates
negatives=f2(pos_down);
% Circle coordinate matrix
X= [pos_up;pos_down];
Y= [positives;negatives];
Coordinates_Circle= [X Y];

%% GENERATION OF THE BLADE GEOMETRY

x=Coordinates(:,1); % X coordinates of the airfoil geometry (divided by
chord length x/c)
y=Coordinates(:,2); % Y coordinates of the airfoil geometry (divided by
chord length x/c)
L=length(x);

xcir=Coordinates_Circle(:,1);
ycir=Coordinates_Circle(:,2);

r=zeros(L,1);

% TRANSFORMATION OF AIRFOIL POINTS: CORRECTION OF SCALE AND ROTATION
for n=1:length(rv)
    r(:,1)=rv(n);
    c=Chord(n); % Chord value of the n-th element
  % AIRFOIL COORDINATE SYSTEM: x1-y1
if n==1 || n==2 || n==3
  x1=xcir*c;
  y1=ycir*c;
  x1=x1-c/2; % Reference system moved to the circle axis
else
    x1=x*c; % Coordinates are scaled to the chord value
    y1=y*c;
    x1=x1+(Chord(1)/2-c/4); % Cordinates moved to make presure center coin-
cident with X axis (span axis of blade)
    x1=x1-Chord(1)/2;       % Reference frame moved to the presure center
of the airfoil (to rotate it)(already aligned with span axis)
end

    beta=Beta(n); % Twist angle value of the n-th element
    beta_rad=beta*pi/180;

    [theta,radius]=cart2pol(x1,y1); % Cartesian coordites are switched to
polars

    % 90° rotation of the coordinates for a side view of the turbine
    theta=theta+pi/2;
```

```
    % Twisting the blade setions:
    theta=theta-beta_rad; % Rotation of all the points of the section

    if f==1 || f==2 || f==3
    theta_mat(:,n,f)=theta;
    radius_mat(:,n,f)=radius;
    else % Another matrix is needed to store SG6040 coordinates, since dif-
ferent number of points are used for each section
    theta_mat_SG6040(:,n)=theta;
    radius_mat_SG6040(:,n)=radius;
    end
 % Transformation from polar to cartesian frame of the translated/rotated
airfoils
    [x2,y2]=pol2cart(theta,radius);

 %DATA SAVING: Overall coordinate matrixes (cartesian). Blade reference
system: XYZ
    if f==1 || f==2 || f==3
    X_mat(:,n,f)=x2; % X axis: Blade width axis
    Y_mat(:,n,f)=y2; % Y axis: Blade height axis
    Z_mat(:,n,f)=r(:,1); % Z axis: Blade span axis (Blade length)
    else
    X_mat_SG6040(:,n)=x2;
    Y_mat_SG6040(:,n)=y2;
    Z_mat_SG6040(:,n)=r(:,1);
    end
end

end
% COORDINATE DATA VECTORS
% S822 AIRFOIL
    X_mat_S822=X_mat(:,:,1);
    Y_mat_S822=Y_mat(:,:,1);
    Z_mat_S822=Z_mat(:,:,1);
    theta_mat_S822=theta_mat(:,:,1);
    radius_mat_S822=radius_mat(:,:,1);
% S823 AIRFOIL
    X_mat_S823=X_mat(:,:,2);
    Y_mat_S823=Y_mat(:,:,2);
    Z_mat_S823=Z_mat(:,:,2);
    theta_mat_S823=theta_mat(:,:,2);
    radius_mat_S823=radius_mat(:,:,2);
% S809 AIRFOIL
    X_mat_S809=X_mat(:,:,3);
    Y_mat_S809=Y_mat(:,:,3);
    Z_mat_S809=Z_mat(:,:,3);
    theta_mat_S809=theta_mat(:,:,3);
    radius_mat_S809=radius_mat(:,:,3);
% SG6040 AIRFOIL
%     X_mat_SG6040=X_mat(:,:,4);
%     Y_mat_SG6040=Y_mat(:,:,4);
%     Z_mat_SG6040=Z_mat(:,:,4);
%     theta_mat_SG6040=theta_mat(:,:,4);
%     radius_mat_SG6040=radius_mat(:,:,4);

%% 3D PLOTS OF THE BLADES
font='LM Roman 12';
% S822 AIRFOIL
```

```
figure('DefaultTextFontName', font, 'DefaultAxesFontName', font,'DefaultA-
xesFontSize',20,'DefaultTextFontSize',20);
surf(Z_mat_S822,X_mat_S822,Y_mat_S822);
xlabel('Blade length [m]')
ylabel('Blade width [m]')
zlabel('Blade height [m]')
title('S822 LAYOUT BLADE 3D VIEW');
axis equal;
% S823 AIRFOIL
figure('DefaultTextFontName', font, 'DefaultAxesFontName', font,'DefaultA-
xesFontSize',20,'DefaultTextFontSize',20);
surf(Z_mat_S823,X_mat_S823,Y_mat_S823);
xlabel('Blade length [m]')
ylabel('Blade width [m]')
zlabel('Blade height [m]')
title('S823 LAYOUT BLADE 3D VIEW');
axis equal;
% S809 AIRFOIL
figure('DefaultTextFontName', font, 'DefaultAxesFontName', font,'DefaultA-
xesFontSize',20,'DefaultTextFontSize',20);
surf(Z_mat_S809,X_mat_S809,Y_mat_S809);
xlabel('Blade length [m]')
ylabel('Blade width [m]')
zlabel('Blade height [m]')
title('S809 LAYOUT BLADE 3D VIEW');
axis equal;
% SG6040 AIRFOIL
figure('DefaultTextFontName', font, 'DefaultAxesFontName', font,'DefaultA-
xesFontSize',20,'DefaultTextFontSize',20);
surf(Z_mat_SG6040,X_mat_SG6040,Y_mat_SG6040);
xlabel('Blade length [m]')
ylabel('Blade width [m]')
zlabel('Blade height [m]')
title('SG6040 LAYOUT BLADE 3D VIEW');
axis equal;

%% Polar view of the blade sections
% S822 AIRFOIL
figure('DefaultTextFontName', font, 'DefaultAxesFontName', font,'DefaultA-
xesFontSize',20,'DefaultTextFontSize',20);
polarplot(theta_mat_S822,radius_mat_S822,'LineWidth',1)
title('BLADE SECTION POLAR VIEW: S822 LAYOUT');

% S822 AIRFOIL
figure('DefaultTextFontName', font, 'DefaultAxesFontName', font,'DefaultA-
xesFontSize',20,'DefaultTextFontSize',20);
polarplot(theta_mat_S823,radius_mat_S823,'LineWidth',1)
title('BLADE SECTION POLAR VIEW: S823 LAYOUT');

% S822 AIRFOIL
figure('DefaultTextFontName', font, 'DefaultAxesFontName', font,'DefaultA-
xesFontSize',20,'DefaultTextFontSize',20);
polarplot(theta_mat_S809,radius_mat_S809,'LineWidth',1)
title('BLADE SECTION POLAR VIEW: S809 LAYOUT');

% SG6040 AIRFOIL
figure('DefaultTextFontName', font, 'DefaultAxesFontName', font,'DefaultA-
xesFontSize',20,'DefaultTextFontSize',20);
polarplot(theta_mat_SG6040,radius_mat_SG6040,'LineWidth',1)
```

```
title('BLADE SECTION POLAR VIEW:SG6040 LAYOUT');

%% EXPORTING BLADE GEOMETRY INTO SOLIDWORKS
% The blade geometry is saved by the coordinates of each blade section
% point. It is then loaded into solidworks in order to create a CAD model.
X_mat_SG6040=X_mat_SG6040*1000;
Y_mat_SG6040=Y_mat_SG6040*1000;
Z_mat_SG6040=Z_mat_SG6040*1000;
coordinates_solidworks1=
[X_mat_SG6040(:,1),Y_mat_SG6040(:,1),Z_mat_SG6040(:,1)];
coordinates_solidworks2=
[X_mat_SG6040(:,2),Y_mat_SG6040(:,2),Z_mat_SG6040(:,2)];
coordinates_solidworks3=
[X_mat_SG6040(:,3),Y_mat_SG6040(:,3),Z_mat_SG6040(:,3)];
coordinates_solidworks4=
[X_mat_SG6040(:,4),Y_mat_SG6040(:,4),Z_mat_SG6040(:,4)];
coordinates_solidworks5=
[X_mat_SG6040(:,5),Y_mat_SG6040(:,5),Z_mat_SG6040(:,5)];
coordinates_solidworks6=
[X_mat_SG6040(:,6),Y_mat_SG6040(:,6),Z_mat_SG6040(:,6)];
coordinates_solidworks7=
[X_mat_SG6040(:,7),Y_mat_SG6040(:,7),Z_mat_SG6040(:,7)];
coordinates_solidworks8=
[X_mat_SG6040(:,8),Y_mat_SG6040(:,8),Z_mat_SG6040(:,8)];
coordinates_solidworks9=
[X_mat_SG6040(:,9),Y_mat_SG6040(:,9),Z_mat_SG6040(:,9)];
coordinates_solidworks10=
[X_mat_SG6040(:,10),Y_mat_SG6040(:,10),Z_mat_SG6040(:,10)];
coordinates_solidworks11=
[X_mat_SG6040(:,11),Y_mat_SG6040(:,11),Z_mat_SG6040(:,11)];
coordinates_solidworks12=
[X_mat_SG6040(:,12),Y_mat_SG6040(:,12),Z_mat_SG6040(:,12)];
coordinates_solidworks13=
[X_mat_SG6040(:,13),Y_mat_SG6040(:,13),Z_mat_SG6040(:,13)];
coordinates_solidworks14=
[X_mat_SG6040(:,14),Y_mat_SG6040(:,14),Z_mat_SG6040(:,14)];
coordinates_solidworks15=
[X_mat_SG6040(:,15),Y_mat_SG6040(:,15),Z_mat_SG6040(:,15)];
coordinates_solidworks16=
[X_mat_SG6040(:,16),Y_mat_SG6040(:,16),Z_mat_SG6040(:,16)];
coordinates_solidworks17=
[X_mat_SG6040(:,17),Y_mat_SG6040(:,17),Z_mat_SG6040(:,17)];
coordinates_solidworks18=
[X_mat_SG6040(:,18),Y_mat_SG6040(:,18),Z_mat_SG6040(:,18)];
coordinates_solidworks19=
[X_mat_SG6040(:,19),Y_mat_SG6040(:,19),Z_mat_SG6040(:,19)];
coordinates_solidworks20=
[X_mat_SG6040(:,20),Y_mat_SG6040(:,20),Z_mat_SG6040(:,20)];
coordinates_solidworks21=
[X_mat_SG6040(:,21),Y_mat_SG6040(:,21),Z_mat_SG6040(:,21)];
save blade_coordinates_solidworks1.txt coordinates_solidworks1 -ASCII -TABS
save blade_coordinates_solidworks2.txt coordinates_solidworks2 -ASCII -TABS
save blade_coordinates_solidworks3.txt coordinates_solidworks3 -ASCII -TABS
save blade_coordinates_solidworks4.txt coordinates_solidworks4 -ASCII -TABS
save blade_coordinates_solidworks5.txt coordinates_solidworks5 -ASCII -TABS
save blade_coordinates_solidworks6.txt coordinates_solidworks6 -ASCII -TABS
save blade_coordinates_solidworks7.txt coordinates_solidworks7 -ASCII -TABS
save blade_coordinates_solidworks8.txt coordinates_solidworks8 -ASCII -TABS
save blade_coordinates_solidworks9.txt coordinates_solidworks9 -ASCII -TABS
```

```
save blade_coordinates_solidworks10.txt coordinates_solidworks10 -ASCII -
TABS
save blade_coordinates_solidworks11.txt coordinates_solidworks11 -ASCII -
TABS
save blade_coordinates_solidworks12.txt coordinates_solidworks12 -ASCII -
TABS
save blade_coordinates_solidworks13.txt coordinates_solidworks13 -ASCII -
TABS
save blade_coordinates_solidworks14.txt coordinates_solidworks14 -ASCII -
TABS
save blade_coordinates_solidworks15.txt coordinates_solidworks15 -ASCII -
TABS
save blade_coordinates_solidworks16.txt coordinates_solidworks16 -ASCII -
TABS
save blade_coordinates_solidworks17.txt coordinates_solidworks17 -ASCII -
TABS
save blade_coordinates_solidworks18.txt coordinates_solidworks18 -ASCII -
TABS
save blade_coordinates_solidworks19.txt coordinates_solidworks19 -ASCII -
TABS
save blade_coordinates_solidworks20.txt coordinates_solidworks20 -ASCII -
TABS
save blade_coordinates_solidworks21.txt coordinates_solidworks21 -ASCII -
TABS
```

A.9 Main_OFFDESIGN.m

```
clear all
close all
clc
%% SCRIPT DESCRIPTION:
% This is the second of the 2 main scripts in this work. In it, the BEMT
algorithm explained in Section 6.2 is implemented. It reads the radial
position,chord and twist values at blade boundaries from .txt files and
interpolates them to get values at element nodes
% Then BEMT algorith is applyied over the blade nodes for a range of
velocities where the WT works in order to see the WT performance
curves(Power, Thrust,Torque,RFWBM). Finally, using the Weibull PDF and the
Power Curve, the AEP can be estimated.

%% SCRIPT START

%% SIMULATION TYPE
simulation='3D'; % Choose '3D' to use 3D corrected (by Snel et ali.) air-
foil data. Choose 2D to not consider 3D effects of the flow

%% EXTERNAL CYCLE I: CALCULATION ARE MADE FOR ALL THE AIRFOIL LOYOUTS
(S822,S823,S809 AND SG6040)
for f=44:4
     % Blade design and 2D data loading
      if f==1
        airfoil='S822';
        airfoil_data_2D=xlsread('Airfoil_Data_360_Extrapola-
ted_2D.xlsx','S822','B7:AE514'); % Airfoil 2D extrapolated data (Viterna)
(Just to compare, in Chapter 8)
        load Beta_S822.txt % Twist angle
        load Chord_S822.txt % Chord length
        load Radial_position_S822.txt
        chord= Chord_S822;
        beta= Beta_S822;
        rv(f,:)=Radial_position_S822'; % Blade segment boundary vector
        file_read='Airfoil_Data_S822_3D_360_extrapolated.xlsx'; % File to
read 3D corrected and extrapolated aerodynamic coefficients
        range='B7:AE514'; % Cells where 3D extrapolaed data are stored
     elseif f==2
        airfoil='S823';
        airfoil_data_2D=xlsread('Airfoil_Data_360_Extrapola-
ted_2D.xlsx','S823','B7:AE514');
        load Beta_S823.txt
        load Chord_S823.txt
        load Radial_position_S823.txt
        chord= Chord_S823;
        beta= Beta_S823;
        rv(f,:)=Radial_position_S823';
        file_read='Airfoil_Data_S823_3D_360_extrapolated.xlsx';
        range='B7:AE514';
      elseif f==3
        airfoil='S809';
        airfoil_data_2D=xlsread('Airfoil_Data_360_Extrapola-
ted_2D.xlsx','S809','B7:AE514');
        load Beta_S809.txt
        load Chord_S809.txt
        load Radial_position_S809.txt
        chord= Chord_S809;
```

```
        beta= Beta_S809;
        rv(f,:)=Radial_position_S809';
        file_read='Airfoil_Data_S809_3D_360_extrapolated.xlsx';
        range='B7:AE514';
      elseif f==4
        airfoil='SG6040';
        airfoil_data_2D=xlsread('Airfoil_Data_360_Extrapola-
ted_2D.xlsx','SG6040','B7:AE515');
        load Beta_SG6040.txt
        load Chord_SG6040.txt
        load Radial_position_SG6040.txt
        chord= Chord_SG6040;
        beta= Beta_SG6040;
        rv(f,:)=Radial_position_SG6040';
        file_read='Airfoil_Data_SG6040_3D_360_extrapolated.xlsx';
        range='B7:AE515';
     end
%% BLADE DESIGN PARAMETERS ( DESIGN VALUES, AIR PROPERTIES AND GEOMETRICAL
PARAMETERS) AND TURBINE CHARACTERISTICS
% Design parameters and air properties:
B=3; % Number of blades of the HAWT
lambda_design=6; %  quella dove il Cp di solito è massimo, diversa dalle
lambda calcolate ai diversi raggi
rho = 1.138; % Air density [kg/m3]
visc = 1.74e-5; % Air dynamic viscosity [Pa·s]
% Geometrical parameters:
D_hub = 0.3 ; % Diameter of the hub [m]
R_hub=D_hub/2;
R_tip=rv(f,end);
D_tip=2*R_tip; % Rotor dimeter [m]
Nb=21; % Number of blade boundaries. N° of elements = N-1
element_width=(rv(f,end)-rv(f,1))/(Nb-1); % Element width

% Definition of radial position, chord and twist angle in the element NODES
% (LINEAR INTERPOLATION)
for k=1:(Nb-1)
rv_nodes(k,1)=(rv(f,k)+rv(f,k+1))/2;
beta_nodes(k,1)=(beta(k)+beta(k+1))/2;
chord_nodes(k,1)=(chord(k)+chord(k+1))/2;
end
% Saving design data at element nodes
if f==1
    save radial_pos_nodes_S822.txt rv_nodes -ASCII -TABS
    save chord_nodes_S822.txt chord_nodes -ASCII -TABS
    save beta_nodes_S822.txt beta_nodes -ASCII -TABS
elseif f==2
    save radial_pos_nodes_S823.txt rv_nodes -ASCII -TABS
    save chord_nodes_S823.txt chord_nodes -ASCII -TABS
    save beta_nodes_S823.txt beta_nodes -ASCII -TABS
elseif f==3
    save radial_pos_nodes_S809.txt rv_nodes -ASCII -TABS
    save chord_nodes_S809.txt chord_nodes -ASCII -TABS
    save beta_nodes_S809.txt beta_nodes -ASCII -TABS
elseif f==4
    save radial_pos_nodes_SG6040.txt rv_nodes -ASCII -TABS
    save chord_nodes_SG6040.txt chord_nodes -ASCII -TABS
    save beta_nodes_SG6040.txt beta_nodes -ASCII -TABS
end
```

```
counter=0; % Control counter: last value has to be equal to Nb

% WIND TURBINE CHARACTERISTICS:
U_cut_in=3; % [m/s]
U_cut_out=20; % [m/s]
efficiency=0.85; % Drive train, generator and converter efficiency
delta_U=0.5; % [m/s], wind velocity step

%% EXTERNAL CICLE (II) TO OBTAIN PERFORMANCE FOR A RANGE OF WIND VELOCITIES
(DELTA=0.5 m/s)

U_vector=U_cut_in:delta_U:U_cut_out; % Wind velocity vector

for v=1:length(U_vector)
    estado=[v/length(U_vector)*100 f]
    U_inf=U_vector(v);
    if U_inf>= 3 && U_inf<=7.5 % The turbine will be in variable speed mode
    omega=lambda_design*U_inf/(D_tip/2);  % Variable speed control !
[rad/s] EQ.(6.1)
    else
    omega=lambda_design*7.5/(D_tip/2); % Omega=omega max=constant,
EQ.(6.28)
    end
    omega_mat(v,f)=omega; % Storage of rotational speed for each airfoil

%% LOOP FOR CALCULATING PARAMETERS MOVING SPAN-WISE THROUGH THE BOUNDARIES
(SPAN-WISE LOOP)

for n=1:length(rv_nodes) % for each of the blade nodes (20) do...

a0=0; % First guess of axial induction factor = 0
r=rv_nodes(n);
a_p0=0; % First guess of tangential induction factor = 0
c0=chord_nodes(n); % Loading chord length at the current blade node

lambda = omega*r/U_inf; % Local speed ratio
sigma_r = B*c0/(2*pi*r); % Chord solidity at the section, EQ.(4.17)
W = sqrt(U_inf^2*(1-a0)^2+omega^2*r^2*(1+a_p0)^2); % Relative velocity
[m/s], EQ.(4.7)
Re = W*c0*rho/visc; % Reynolds number, EQ.(3.4)
phi=atan(U_inf*(1-a0)/(omega*r*(1+a_p0))); % Inflow angle [rad], EQ.(4.22)
phi_deg=phi*180/pi; % Switched into degrees

% Calculation of angle of attack
alpha = phi_deg-beta_nodes(n);% in degrees
% Getting aerodynamic coefficients:
if n==1 || n==2 % If the node is in the cylindrical part...
C_l=0;
C_d=1.2;
elseif n==3 % If the node is in the cylindrical part...
C_l=0.7;
C_d=0.8;
elseif strcmp(simulation,'2D')
 airfoil_data=airfoil_data_2D; % Load airfoil 2D data
 [C_l,C_d] = CLift_and_CDrag(Re,alpha,airfoil_data);

elseif strcmp(simulation,'3D') % 3D corrected data is used
 airfoil_data=xlsread(file_read,num2str(n),range); % Loading the corrected
data for the current blade element
```

```
 [C_l,C_d] = CLift_and_CDrag(Re,alpha,airfoil_data); % Lift and drag coef-
ficients
end
% Decomposition of the lift and drag coefficients into normal and
% tangential axes
C_n=C_l*cos(phi)+C_d*sin(phi); % EQ.(4.13)
C_t=C_l*sin(phi)-C_d*cos(phi); % EQ.(4.14)

% CALCULATION OF LOSS CORRECTION FACTORS
f_tip=B*(R_tip-r)/(2*r*sin(phi));
F_tip=2/pi*acos(exp(-f_tip)); % Prandtl Tip-loss correction factor,
EQ.(4.23)

f_hub=B*(r-R_hub)/(2*R_hub*sin(phi));
F_hub=2/pi*acos(exp(-f_hub)); %Hub-loss correction factor, EQ.(4.24)

F=F_tip*F_hub; % Overall correction factor, EQ.(4.25)

% CALCULATION OF INDUCTION FACTORS (GLAUERT CORRECTION IF NEEDED)

C_T=sigma_r*((1-a0)^2)*C_n/sin(phi)^2; % Thrust coefficient calculation,
EQ.(4.33)
if C_T>0.96*F % The element is considered highly loaded
    a1=(18*F-20-3*sqrt(C_T*(50-36*F)+12*F*(3*F-4)))/(36*F-50); % Buhl's co-
rrection EQ.(4.32)
    C_T=8/9+(4*F-40/9)*a1+(50/9-4*F)*a1^2; % Buhl's correction EQ.(4.31)
else
a1=(1+4*F*sin(phi)^2/(sigma_r*C_n))^(-1); % Axial induction factor with
EQ.(4.28)
C_T=4*a1*(1-a1)*F; % EQ.(4.30)
end

a_p1=(4*F*sin(phi)*cos(phi)/(sigma_r*C_t)-1)^(-1); % Tangential induction
factor, EQ.(4.29)

delta_a=abs(a0-a1); % Difference between a and a old
delta_a_p=abs(a_p0-a_p1); % Difference between a' and a' old

epsilon=10^-5; % Tolerance for convergence
w_relax=0.2;

%% ITERATIVE PROCESS FOR EACH SINGLE RADIAL POSITION ( a and a' must con-
verge)
count=1; % Counter to see after how many iterations convergence is achieved
countmax=10^4; % Maximum number of iterations

while (max(delta_a,delta_a_p) > epsilon) && count<countmax % EQ.(6.15) con-
vergence criteria
    count=count+1;
    a_new=w_relax*a1+(1-w_relax)*a0; % EQ.(6.16)
    a_new_p=w_relax*a_p1+(1-w_relax)*a_p0; % EQ.(6.17)
    a0=a_new;
    a_p0=a_new_p;

W = sqrt(U_inf^2*(1-a0)^2+omega^2*r^2*(1+a_p0)^2);
Re = W*c0*rho/visc;
phi=atan(U_inf*(1-a0)/(omega*r*(1+a_p0)));
phi_deg=phi*180/pi; % switched into degrees
% Angle of attack calculation
```

```
alpha = phi_deg-beta_nodes(n);
% Evaluation of aerodynamic coefficients
if n==1 || n==2
C_l=0;
C_d=1.2;
elseif n==3 % If the node is in the cylindrical part...
C_l=0.7;
C_d=0.8;
else
[C_l,C_d] = CLift_and_CDrag(Re,alpha,airfoil_data);
end
sigma_r = B*c0/(2*pi*r);

C_n=C_l.*cos(phi)+C_d.*sin(phi);
C_t=C_l.*sin(phi)-C_d.*cos(phi);

% CALCULATION OF LOSS CORRECTION FACTORS
f_tip=B*(R_tip-r)/(2*r*sin(phi));
F_tip=2/pi*acos(exp(-f_tip)); %Tip-loss correction factor

f_hub=B*(r-R_hub)/(2*R_hub*sin(phi));
F_hub=2/pi*acos(exp(-f_hub)); %Hub-loss correction factor

F=F_tip*F_hub; %Overall correction factor

% CALCULATION OF INDUCTION FACTORS (GLAUERT CORRECTION IF NEEDED)

C_T=sigma_r*((1-a0)^2)*C_n/sin(phi)^2; % Thrust coefficient calculation
if C_T>0.96*F % The element is considered highly loaded
       a1=(18*F-20-3*sqrt(C_T*(50-36*F)+12*F*(3*F-4)))/(36*F-50); % Buhl's
correction
       C_T=8/9+(4*F-40/9)*a1+(50/9-4*F)*a1^2; % Buhl's correction
else
a1=(1+4*F*sin(phi)^2/(sigma_r*C_n))^(-1);
C_T=4*a1*(1-a1)*F;
end
a_p1=(4*F*sin(phi)*cos(phi)/(sigma_r*C_t)-1)^(-1);

delta_a=abs(a0-a1);
delta_a_p=abs(a_p0-a_p1);
end

%% CONVERGENCE ACHIEVED FOR A RADIAL POSITION: CALCULATION OF AERODYNAMIC
FORCES/MOMENTS AND DATA SAVING
% Axial direction
Elemental_Thrust=C_T*0.5*rho*U_inf^2*2*pi*r; % Elemental thrust in the tur-
bine annulus [N/m], EQ.(6.18)
Elemental_Axial_Load=Elemental_Thrust/3; % Axial load in each blade element
[N/m], EQ.(6.20)
Root_Flapwise_Bending_Moment=(Elemental_Axial_Load)*(r-R_hub); % Elemental
flapwise bending moment respect to the blade root [N.m/m], EQ.(6.23)

% Tangential direction
Elemental_Torque=4*F*pi*rho*U_inf*(omega*r)*a_p1*(1-a1)*r^2; % Elemental
torque in the turbine annulus [N.m/m], EQ.(6.19)
Elemental_Tangential_Load=Elemental_Torque/(B*r); % Tangential load in each
blade element [N/m], EQ.(6.21)
Elemental_Torque_1blade=Elemental_Torque/3; % Torque generated by each ele-
mental tangential load (1 blade) [N.m/m], EQ.(6.22)
```

```
% Data for each radial position is stored in 3D matrixes: (v rows,n columns
and f depth)

count_mat(v,n,f)=count;
glide_ratio(v,n,f)=C_l/C_d; % Glide ratio at blade setions
a(v,n,f)=a1; % Axial induction coefficient
a_p(v,n,f)=a_p1; % Tangential induction coefficient
W_mat(v,n,f)=W; % Relative velocity at section
alpha_mat(v,n,f)=alpha; % Angle of attack matrix
Re_mat(v,n,f)=Re; % Re number matrix
phi_mat(v,n,f)=phi*180/pi; %in gradi, non in radianti
F_mat(v,n,f)=F; % Overall loss factor

C_L_mat(v,n,f)=C_l; % Lift coefficient matrix
C_D_mat(v,n,f)=C_d; % Drag coefficient matrix
CT_mat(v,n,f)=C_T; % Thrust coefficient matrix

Thrust_mat(v,n,f)=Elemental_Thrust; % Elemental thrust matrix
Axial_Load_mat(v,n,f)=Elemental_Axial_Load; % Elemental axial load matrix
RFWBM_mat(v,n,f)=Root_Flapwise_Bending_Moment; % Element contribution to
Root Flap-wise Bending Moment matrix

Torque_mat(v,n,f)=Elemental_Torque; % Elemental torque matrix
Tangential_Load_mat(v,n,f)=Elemental_Tangential_Load; % Elemental tangen-
tial load matrix
Torque_1blade_mat(v,n,f)=Elemental_Torque_1blade; % Elemental torque by
each blade matrix

counter=counter+1;
end

%% END OF THE FOR LOOP THAT RUNS SPAN WISE
% CALCULATION OF TORQUE,THRUST,RFWBM,POWER AND POWER COEFFICIENT
% Numerical integration by rectangle method
TORQUE(v,f)=element_width*sum(Torque_mat(v,1:end,f)); % [N·m]
THRUST(v,f)=element_width*sum(Thrust_mat(v,1:end,f)); % [N]
RFWBM(v,f)=element_width*sum(RFWBM_mat(v,1:end,f)); % [N.m]

Power(v,f)=omega*TORQUE(v,f)*efficiency; % Generated electric power [W]
P_available=0.5*rho*U_inf^3*pi*R_tip^2;
Cp(v,f)= Power(v,f)/P_available;

% END OF THE LOOP FOR THE V-TH WIND VELOCITY
end

end
% Power,Torque,Thrust, Root bending moment, power coefficient and
 rotational speed for each airfoil

Power_S822=  Power(:,1)';
Power_S823= Power(:,2)';
Power_S809= Power(:,3)';
Power_SG6040= Power(:,4)';

Torque_S822= TORQUE(:,1)';
Torque_S823= TORQUE(:,2)';
Torque_S809= TORQUE(:,3)';
Torque_SG6040= TORQUE(:,4)';
```

```
Thrust_S822= THRUST(:,1);
Thrust_S823= THRUST(:,2);
Thrust_S809= THRUST(:,3);
Thrust_SG6040= THRUST(:,4);

Root_Flapwise_Bending_Moment_S822=RFWBM(:,1);
Root_Flapwise_Bending_Moment_S823=RFWBM(:,2);
Root_Flapwise_Bending_Moment_S809=RFWBM(:,3);
Root_Flapwise_Bending_Moment_SG6040=RFWBM(:,4);

Cp_S822=Cp(:,1);
Cp_S823=Cp(:,2);
Cp_S809=Cp(:,3);
Cp_SG6040=Cp(:,4);

omega_mat_rpm=omega_mat*30/pi; % Rotational speed in rpm

Omega_S822=omega_mat_rpm(:,1);
Omega_S823=omega_mat_rpm(:,2);
Omega_S809=omega_mat_rpm(:,3);
Omega_SG6040=omega_mat_rpm(:,4);

%% GENERATION OF WIND TURBINE CHARACTERISTIC PLOTS
U_vector_col=U_vector';
font='LM Roman 12';
figure('DefaultTextFontName', font, 'DefaultAxesFontName', font,'DefaultA-
xesFontSize',14,'DefaultTextFontSize',14);
% ELECTRIC POWER VS WIND VELOCITY
subplot(3,2,1)
plot (U_vector_col,Power_S822,'ro-');
hold on
plot (U_vector_col,Power_S823,'bs-');
hold on
plot (U_vector_col,Power_S809,'kd-');
hold on
plot (U_vector_col,Power_SG6040,'m^-');
hold on
grid on
xlabel('Wind velocity [m/s]');
ylabel({'Electric Power';'[W]'});
axis([2 20 0 10500])

% CP (WITH LOSSES) VS WIND VELOCITY
subplot(3,2,2)
plot (U_vector_col,Cp_S822,'ro-');
hold on
plot (U_vector_col,Cp_S823,'bs-');
hold on
plot (U_vector_col,Cp_S809,'kd-');
hold on
plot (U_vector_col,Cp_SG6040,'m^-');
hold on
grid on
xlabel('Wind velocity [m/s]');
ylabel({'Power coefficient'; '(with losses)'});
axis([2 20 0 0.45])

% TORQUE IN SHAFT VS WIND VELOCITY
```

```
subplot(3,2,3)
plot (U_vector_col,Torque_S822,'ro-');
hold on
plot (U_vector_col,Torque_S823,'bs-');
hold on
plot (U_vector_col,Torque_S809,'kd-');
hold on
plot (U_vector_col,Torque_SG6040,'m^-');
hold on
grid on
xlabel('Wind velocity [m/s]');
ylabel({'Shaft Torque'; '[N·m]'});

% TURBINE THRUST VS WIND VELOCITY
subplot(3,2,4)
plot (U_vector_col,Thrust_S822,'ro-');
hold on
plot (U_vector_col,Thrust_S823,'bs-');
hold on
plot (U_vector_col,Thrust_S809,'kd-');
hold on
plot (U_vector_col,Thrust_SG6040,'m^-');
hold on
grid on
xlabel('Wind velocity [m/s]');
ylabel({'Turbine Thrust','[N]'});
axis([2 20 0 2500])

% ROTATIONAL SPEED VS WIND VELOCITY
subplot(3,2,5)
plot (U_vector_col,Omega_S822,'ro-');
hold on
plot (U_vector_col,Omega_S823,'bs-');
hold on
plot (U_vector_col,Omega_S809,'kd-');
hold on
plot (U_vector_col,Omega_SG6040,'m^-');
hold on
grid on
lgd=legend('S822','S823','S809','SG6040','Location','SouthEast');
lgd.NumColumns=2;
lgd.Title.String='BLADE LAYOUT';
xlabel('Wind velocity [m/s]');
ylabel({'Rotational speed';'[rpm]'});
axis([2 20 40 150])

% ROOT FLAPWISE BENDING MOMENT VS WIND VELOCITY
subplot(3,2,6)
plot (U_vector_col,Root_Flapwise_Bending_Moment_S822,'ro-');
hold on
plot (U_vector_col,Root_Flapwise_Bending_Moment_S823,'bs-');
hold on
plot (U_vector_col,Root_Flapwise_Bending_Moment_S809,'kd-');
hold on
plot (U_vector_col,Root_Flapwise_Bending_Moment_SG6040,'m^-');
hold on
grid on
xlabel('Wind velocity [m/s]');
ylabel({'Root Flapwise';'Bending Moment';'[N.m]'});
```

```
axis([2 20 0 1200])

%% ANNUAL ENERGY PRODUCTION COMPUTATION
% WEIBULL CURVE DEFINITION
k=1.78;     % Shape parameter
c=4.9;        % Scale parameter
weibull = wblpdf(U_vector,c,k);
% Energy density curve
Power_S822_kW= Power_S822/1000;
Power_S823_kW= Power_S823/1000;
Power_S809_kW= Power_S809/1000;
Power_SG6040_kW= Power_SG6040/1000;

energy_density_S822=weibull.*Power_S822_kW;
energy_density_S823=weibull.*Power_S823_kW;
energy_density_S809=weibull.*Power_S809_kW;
energy_density_SG6040=weibull.*Power_SG6040_kW;
% Numerical integration of energy density
integral_edc_S822= trapz(U_vector,energy_density_S822);
integral_edc_S823= trapz(U_vector,energy_density_S823);
integral_edc_S809= trapz(U_vector,energy_density_S809);
integral_edc_SG6040= trapz(U_vector,energy_density_SG6040);

Annual_Energy_Production_S822=integral_edc_S822*8760   % in kWh
Annual_Energy_Production_S823=integral_edc_S823*8760
Annual_Energy_Production_S809=integral_edc_S809*8760
Annual_Energy_Production_SG6040=integral_edc_SG6040*8760

% Generation of a bar diagram for AEP
c = categorical({'S822','S823','S809','SG6040'});
AEP = [Annual_Energy_Production_S822 Annual_Energy_Production_S823 An-
nual_Energy_Production_S809 Annual_Energy_Production_SG6040];

figure('DefaultTextFontName', font, 'DefaultAxesFontName', font,'DefaultA-
xesFontSize',12,'DefaultTextFontSize',12);
bar(c,AEP,'FaceColor','0.30,0.75,0.93')
xlabel('Blade layout')
ylabel('Annual Energy Production [kWh_e]')

%% DIMENSIONATION OF THE BLADES: DIAMETER ITERATION (OBJETIVE: MAX=9950-
1050 W)
% Find maximum power and position
Max_Power_S822=max(Power_S822) % Maximum power for S822 blade layout
Max_Power_S823=max(Power_S823)
Max_Power_S809=max(Power_S809)
Max_Power_SG6040=max(Power_SG6040)

pos_S822=find(Power_S822==Max_Power_S822);
pos_S823=find(Power_S823==Max_Power_S823);
pos_S809=find(Power_S809==Max_Power_S809);
pos_SG6040=find(Power_SG6040==Max_Power_SG6040);

% % Wind velocity at which maximum power is achieved
Max_Power_S822_at=U_vector(pos_S822)
Max_Power_S823_at=U_vector(pos_S823)
Max_Power_S809_at=U_vector(pos_S809)
Max_Power_SG6040_at=U_vector(pos_SG6040)

% Power Coefficient at maximum power point:
```

```
Cp_Pmax_S822=Cp_S822(pos_S822)
Cp_Pmax_S823=Cp_S823(pos_S823)
Cp_Pmax_S809=Cp_S809(pos_S809)
Cp_Pmax_SG6040=Cp_SG6040(pos_SG6040)

% New diameter of the rotor for next iteration; from
P=0.5*rho*U^3*pi/4*Cp*D^2
New_Diameter_S822=round(sqrt(10000*8/(rho*Max_Po-
wer_S822_at^3*Cp_Pmax_S822*pi)),2)
New_Diameter_S823=round(sqrt(10000*8/(rho*Max_Po-
wer_S823_at^3*Cp_Pmax_S823*pi)),2)
New_Diameter_S809=round(sqrt(10000*8/(rho*Max_Po-
wer_S809_at^3*Cp_Pmax_S809*pi)),2)
New_Diameter_SG6040=round(sqrt(10000*8/(rho*Max_Po-
wer_SG6040_at^3*Cp_Pmax_SG6040*pi)),2)
```

APPENDIX B: WT_Perf INPUT/OUTPUT FILES

B.1 2D data simulation input file

----- WT_Perf Input File --
This line is for user comments.
----- Input Configuration --
False Echo: Echo input parameters to "<rootname>.ech"?
True DimenInp: Turbine parameters are dimensional?
True Metric: Turbine parameters are Metric (MKS vs FPS)?
----- Model Configuration --
1 NumSect: Number of circumferential sectors.
10000 MaxIter: Max number of iterations for induction factor.
1.0e-005 ATol: Error tolerance for induction iteration.
1.0e-006 SWTol: Error tolerance for skewed-wake iteration.
----- Algorithm Configuration --
True TipLoss: Use the Prandtl tip-loss model?
True HubLoss: Use the Prandtl hub-loss model?
True Swirl: Include Swirl effects?
False SkewWake: Apply skewed-wake correction?
True AdvBrake: Use the advanced brake-state model?
True IndProp: Use PROP-PC instead of PROPX induction algorithm?
True AIDrag: Use the drag term in the axial induction calculation?
True TIDrag: Use the drag term in the tangential induction calculation?
----- Turbine Data ---
3 NumBlade: Number of blades.
3.37 RotorRad: Rotor radius [length].
0.15 HubRad: Hub radius [length or div by radius].
0.0 PreCone: Precone angle. positive downstream [deg].
0.0 Tilt: Shaft tilt [deg].
0.0 Yaw: Yaw error [deg].
25 HubHt: Hub height [length or div by radius].
20 NumSeg: Number of blade segments (entire rotor radius).

RElm	Twist	Chord	AFfile	PrntElem
0.2305	0	0.2000	1	True
0.3915	0	0.2000	1	True
0.5525	10.3517	0.3735	2	True
0.7135	18.6356	0.5228	3	True

```
   0.8745   14.9781    0.4740   3        True
   1.0355   12.1495    0.4275   3        True
   1.1965    9.9260    0.3864   3        True
   1.3575    8.1437    0.3509   3        True
   1.5185    6.6890    0.3205   3        True
   1.6795    5.4820    0.2944   3        True
   1.8405    4.4655    0.2718   3        True
   2.0015    3.5972    0.2521   3        True
   2.1625    2.8449    0.2347   3        True
   2.3235    2.1823    0.2191   3        True
   2.4845    1.5857    0.2046   3        True
   2.6455    1.0296    0.1907   3        True
   2.8065    0.4806    0.1762   3        True
   2.9675   -0.1197    0.1592   3        True
   3.1285   -0.9892    0.1351   3        True
   3.2895   -2.7630    0.0894   3        True
----- Aerodynamic Data ------------------------------------------------------
1.13800             Rho:            Air density [mass/volume].
0.000015289         KinVisc:        Kinematic air viscosity
0.000               ShearExp:       Wind shear exponent (1/7 law = 0.143).
False               UseCm:          Are Cm data included in the airfoil tables?
3                   NumAF:          Number of airfoil files.
"Cylinder.dat"
"Transition_node.dat"
"SG6040_2D_POLARS.dat"
----- I/O Settings ----------------------------------------------------------
False           TabDel:         Make output tab-delimited (fixed-width otherwise).
False           KFact:          Output dimensional parameters in K (e.g.. kN instead on N)
True            WriteBED:       Write out blade element data to "<rootname>.bed"?
False           InputTSR:       Input speeds as TSRs?
"mps"           SpdUnits:       Wind-speed units (mps. fps. mph).
----- Combined-Case Analysis ------------------------------------------------
35              NumCases:       Number of cases to run.  Enter zero for parametric analysis.
WS or TSR   RotSpd   Pitch      Remove following block of lines if NumCases is zero.
   3.0000   51.0051    0
   3.5000   59.5060    0
   4.0000   68.0069    0
   4.5000   76.5077    0
   5.0000   85.0086    0
   5.5000   93.5094    0
   6.0000  102.0103    0
   6.5000  110.5111    0
   7.0000  119.0120    0
```

7.5000	127.5129	0
8.0000	127.5129	0
8.5000	127.5129	0
9.0000	127.5129	0
9.5000	127.5129	0
10.0000	127.5129	0
10.5000	127.5129	0
11.0000	127.5129	0
11.5000	127.5129	0
12.0000	127.5129	0
12.5000	127.5129	0
13.0000	127.5129	0
13.5000	127.5129	0
14.0000	127.5129	0
14.5000	127.5129	0
15.0000	127.5129	0
15.5000	127.5129	0
16.0000	127.5129	0
16.5000	127.5129	0
17.0000	127.5129	0
17.5000	127.5129	0
18.0000	127.5129	0
18.5000	127.5129	0
19.0000	127.5129	0
19.5000	127.5129	0
20.0000	127.5129	0

B.2 3D data simulation input file

----- WT_Perf Input File ----------
This line is for user comments.
----- Input Configuration ----------
False Echo: Echo input parameters to "<rootname>.ech"?
True DimenInp: Turbine parameters are dimensional?
True Metric: Turbine parameters are Metric (MKS vs FPS)?
----- Model Configuration ----------
1 NumSect: Number of circumferential sectors.
10000 MaxIter: Max number of iterations for induction factor.
1.0e-005 ATol: Error tolerance for induction iteration.
1.0e-006 SWTol: Error tolerance for skewed-wake iteration.
----- Algorithm Configuration ----------
True TipLoss: Use the Prandtl tip-loss model?
True HubLoss: Use the Prandtl hub-loss model?
True Swirl: Include Swirl effects?
False SkewWake: Apply skewed-wake correction?
True AdvBrake: Use the advanced brake-state model?
True IndProp: Use PROP-PC instead of PROPX induction algorithm?
True AIDrag: Use the drag term in the axial induction calculation?
True TIDrag: Use the drag term in the tangential induction calculation?
----- Turbine Data ----------
3 NumBlade: Number of blades.
3.37 RotorRad: Rotor radius [length].
0.15 HubRad: Hub radius [length or div by radius].
0.0 PreCone: Precone angle. positive downstream [deg].
0.0 Tilt: Shaft tilt [deg].
0.0 Yaw: Yaw error [deg].
25 HubHt: Hub height [length or div by radius].
20 NumSeg: Number of blade segments (entire rotor radius).

RElm	Twist	Chord	AFfile	PrntElem
0.2305	0	0.2000	1	True
0.3915	0	0.2000	1	True
0.5525	10.3517	0.3735	2	True
0.7135	18.6356	0.5228	3	True
0.8745	14.9781	0.4740	4	True
1.0355	12.1495	0.4275	5	True
1.1965	9.9260	0.3864	6	True
1.3575	8.1437	0.3509	7	True
1.5185	6.6890	0.3205	8	True
1.6795	5.4820	0.2944	9	True
1.8405	4.4655	0.2718	10	True

2.0015	3.5972	0.2521	11	True
2.1625	2.8449	0.2347	12	True
2.3235	2.1823	0.2191	13	True
2.4845	1.5857	0.2046	14	True
2.6455	1.0296	0.1907	15	True
2.8065	0.4806	0.1762	16	True
2.9675	-0.1197	0.1592	17	True
3.1285	-0.9892	0.1351	18	True
3.2895	-2.7630	0.0894	19	True

----- Aerodynamic Data ----------

1.13800	Rho:	Air density [mass/volume].
0.000015289	KinVisc:	Kinematic air viscosity
0.000	ShearExp:	Wind shear exponent (1/7 law = 0.143).
False	UseCm:	Are Cm data included in the airfoil tables?
19	NumAF:	Number of airfoil files.

"Cylinder.dat"
"Transition_node.dat"
"SG64040_Node_4.DAT"
"SG64040_Node_5.DAT"
"SG64040_Node_6.DAT"
"SG64040_Node_7.DAT"
"SG64040_Node_8.DAT"
"SG64040_Node_9.DAT"
"SG64040_Node_10.DAT"
"SG64040_Node_11.DAT"
"SG64040_Node_12.DAT"
"SG64040_Node_13.DAT"
"SG64040_Node_14.DAT"
"SG64040_Node_15.DAT"
"SG64040_Node_16.DAT"
"SG64040_Node_17.DAT"
"SG64040_Node_18.DAT"
"SG64040_Node_19.DAT"
"SG64040_Node_20.DAT"

----- I/O Settings ----------

False	TabDel:	Make output tab-delimited (fixed-width otherwise).
False	KFact:	Output dimensional parameters in K (e.g.. kN instead on N)
True	WriteBED:	Write out blade element data to "<rootname>.bed"?
False	InputTSR:	Input speeds as TSRs?
"mps"	SpdUnits:	Wind-speed units (mps. fps. mph).

----- Combined-Case Analysis ----------

35	NumCases:	Number of cases to run. Enter zero for parametric analysis.
WS or TSR	RotSpd	Pitch — Remove following block of lines if NumCases is zero.
3.0000	51.0051	0
3.5000	59.5060	0
4.0000	68.0069	0
4.5000	76.5077	0
5.0000	85.0086	0
5.5000	93.5094	0
6.0000	102.0103	0
6.5000	110.5111	0
7.0000	119.0120	0
7.5000	127.5129	0
8.0000	127.5129	0
8.5000	127.5129	0
9.0000	127.5129	0
9.5000	127.5129	0
10.0000	127.5129	0
10.5000	127.5129	0
11.0000	127.5129	0
11.5000	127.5129	0
12.0000	127.5129	0
12.5000	127.5129	0
13.0000	127.5129	0
13.5000	127.5129	0
14.0000	127.5129	0
14.5000	127.5129	0
15.0000	127.5129	0
15.5000	127.5129	0
16.0000	127.5129	0
16.5000	127.5129	0
17.0000	127.5129	0
17.5000	127.5129	0
18.0000	127.5129	0
18.5000	127.5129	0
19.0000	127.5129	0
19.5000	127.5129	0
20.0000	127.5129	0

B.3 2D data simulation output file

Results generated by WT_Perf (v3.10. 1-Jan-2009) for input file "Design_SG6040_2D.wtp".

WindSpeed	TSR	Rotor Speed	Pitch	Power	Torque	Thrust	FlapMoment	Cp	Cq
m/s	-	rpm	deg	W	N-m	N	N-m	-	-
3.0000	6.0000	51.0051	0.0000	247.5842	46.3533	144.1227	99.3437	0.4517	0.0753
3.5000	6.0000	59.5060	0.0000	398.9133	64.0161	197.2574	136.0022	0.4583	0.0764
4.0000	6.0000	68.0069	0.0000	599.2826	84.1492	258.8354	178.4961	0.4612	0.0769
4.5000	6.0000	76.5077	0.0000	857.4102	107.0175	329.1703	227.0322	0.4635	0.0772
5.0000	6.0000	85.0086	0.0000	1181.5182	132.7238	408.3143	281.6545	0.4656	0.0776
5.5000	6.0000	93.5094	0.0000	1577.4401	161.0900	495.2285	341.6803	0.4670	0.0778
6.0000	6.0000	102.0103	0.0000	2053.0764	192.1907	590.4208	407.4250	0.4682	0.0780
6.5000	6.0000	110.5111	0.0000	2615.5791	226.0128	693.9691	478.9059	0.4691	0.0782
7.0000	6.0000	119.0120	0.0000	3272.3833	262.5697	805.7964	556.1119	0.4699	0.0783
7.5000	6.0000	127.5129	0.0000	4030.8105	301.8627	925.9156	639.0460	0.4706	0.0784
8.0000	5.6250	127.5129	0.0000	4853.1626	363.4477	997.4636	683.8276	0.4669	0.0830
8.5000	5.2941	127.5129	0.0000	5647.4932	422.9343	1047.8131	714.3643	0.4530	0.0856
9.0000	5.0000	127.5129	0.0000	6398.3564	479.1654	1092.4807	740.4525	0.4323	0.0865
9.5000	4.7368	127.5129	0.0000	7122.1494	533.3694	1132.2743	763.9666	0.4092	0.0864
10.0000	4.5000	127.5129	0.0000	7802.5586	584.3245	1166.5856	784.5013	0.3843	0.0854

10.5000	4.2857	127.5129	0.0000	8412.8105	630.0256	1193.9460	801.3943	0.3580	0.0835
11.0000	4.0909	127.5129	0.0000	8965.6680	671.4285	1217.7078	815.4805	0.3318	0.0811
11.5000	3.9130	127.5129	0.0000	9259.5508	693.4369	1225.2990	824.4242	0.2999	0.0766
12.0000	3.7500	127.5129	0.0000	9509.8750	712.1835	1233.8098	830.5823	0.2711	0.0723
12.5000	3.6000	127.5129	0.0000	9700.1387	726.4321	1242.0345	836.7882	0.2446	0.0680
13.0000	3.4615	127.5129	0.0000	9800.7441	733.9663	1247.9878	840.6556	0.2197	0.0635
13.5000	3.3333	127.5129	0.0000	9831.7695	736.2898	1253.2504	843.2949	0.1968	0.0591
14.0000	3.2143	127.5129	0.0000	9781.9590	732.5594	1257.3635	844.1390	0.1756	0.0546
14.5000	3.1034	127.5129	0.0000	9624.3359	720.7554	1258.9340	841.7452	0.1555	0.0501
15.0000	3.0000	127.5129	0.0000	9378.5176	702.3462	1260.1384	838.2922	0.1369	0.0456
15.5000	2.9032	127.5129	0.0000	8925.9170	668.4515	1256.5850	828.8888	0.1181	0.0407
16.0000	2.8125	127.5129	0.0000	8690.9385	650.8542	1261.5737	825.3361	0.1045	0.0372
16.5000	2.7273	127.5129	0.0000	8120.4180	608.1285	1259.8422	815.1179	0.0890	0.0326
17.0000	2.6471	127.5129	0.0000	7423.2314	555.9171	1256.5989	802.7844	0.0744	0.0281
17.5000	2.5714	127.5129	0.0000	6636.1714	496.9751	1255.4500	791.2510	0.0610	0.0237
18.0000	2.5000	127.5129	0.0000	6003.3506	449.5840	1262.6451	786.6107	0.0507	0.0203
18.5000	2.4324	127.5129	0.0000	5580.4238	417.9114	1277.4640	788.1310	0.0434	0.0178
19.0000	2.3684	127.5129	0.0000	5677.7881	425.2029	1305.3053	801.7310	0.0408	0.0172
19.5000	2.3077	127.5129	0.0000	5760.6309	431.4069	1332.8513	814.6168	0.0383	0.0166
20.0000	2.2500	127.5129	0.0000	5806.6704	434.8548	1359.4814	826.0825	0.0358	0.0159

B.4 3D data simulation output file

Results generated by WT_Perf (v3.10. 1-Jan-2009) for input file "Design_SG6040_3D.wtp".

WindSpeed	TSR	RotorSpeed	Pitch	Power	Torque	Thrust	FlapMoment	Cp	Cq
m/s	-	rpm	deg	W	N-m	N	N-m	-	-
3.0000	6.0000	51.0051	0.0000	247.5839	46.3532	144.1238	99.3444	0.4517	0.0753
3.5000	6.0000	59.5060	0.0000	398.9099	64.0155	197.2576	136.0023	0.4583	0.0764
4.0000	6.0000	68.0069	0.0000	599.2795	84.1488	258.8363	178.4966	0.4612	0.0769
4.5000	6.0000	76.5077	0.0000	857.4073	107.0171	329.1707	227.0324	0.4635	0.0772
5.0000	6.0000	85.0086	0.0000	1181.5161	132.7236	408.3130	281.6540	0.4656	0.0776
5.5000	6.0000	93.5094	0.0000	1577.4270	161.0888	495.2264	341.6799	0.4670	0.0778
6.0000	6.0000	102.0103	0.0000	2053.0918	192.1921	590.4213	407.4257	0.4682	0.0780
6.5000	6.0000	110.5111	0.0000	2615.6006	226.0147	693.9698	478.9069	0.4691	0.0782
7.0000	6.0000	119.0120	0.0000	3272.3892	262.5702	805.7958	556.1127	0.4699	0.0783
7.5000	6.0000	127.5129	0.0000	4030.8550	301.8660	925.9195	639.0480	0.4706	0.0784
8.0000	5.6250	127.5129	0.0000	4854.4878	363.5469	998.9937	684.3307	0.4670	0.0830
8.5000	5.2941	127.5129	0.0000	5662.3809	424.0491	1055.0468	716.9036	0.4542	0.0858
9.0000	5.0000	127.5129	0.0000	6436.8975	482.0518	1106.4155	745.3429	0.4349	0.0870
9.5000	4.7368	127.5129	0.0000	7206.8115	539.7097	1155.6648	772.0460	0.4140	0.0874
10.0000	4.5000	127.5129	0.0000	7954.1318	595.6755	1200.9854	796.2609	0.3918	0.0871

10.5000	4.2857	127.5129	0.0000	8673.5039	649.5485	1243.1907	818.1041	0.3691	0.0861
11.0000	4.0909	127.5129	0.0000	9334.9414	699.0829	1280.1816	836.6499	0.3455	0.0844
11.5000	3.9130	127.5129	0.0000	9967.7949	746.4764	1315.7101	853.3818	0.3228	0.0825
12.0000	3.7500	127.5129	0.0000	10554.2363	790.3944	1348.5491	867.4185	0.3009	0.0802
12.5000	3.6000	127.5129	0.0000	11125.7500	833.1945	1381.5040	882.0768	0.2806	0.0779
13.0000	3.4615	127.5129	0.0000	11618.6582	870.1077	1410.1403	894.1628	0.2605	0.0753
13.5000	3.3333	127.5129	0.0000	11791.8379	883.0770	1428.4193	901.5146	0.2361	0.0708
14.0000	3.2143	127.5129	0.0000	11727.5918	878.2656	1442.5867	905.2458	0.2105	0.0655
14.5000	3.1034	127.5129	0.0000	11718.4287	877.5795	1454.2385	906.0757	0.1893	0.0610
15.0000	3.0000	127.5129	0.0000	11628.0137	870.8085	1460.9797	904.6630	0.1697	0.0566
15.5000	2.9032	127.5129	0.0000	11413.7539	854.7628	1466.3936	899.3281	0.1510	0.0520
16.0000	2.8125	127.5129	0.0000	11183.8369	837.5444	1473.9800	895.3146	0.1345	0.0478
16.5000	2.7273	127.5129	0.0000	10655.7207	797.9944	1476.4005	885.2329	0.1168	0.0428
17.0000	2.6471	127.5129	0.0000	9996.1553	748.6003	1467.0532	870.9607	0.1002	0.0379
17.5000	2.5714	127.5129	0.0000	9244.2764	692.2931	1462.6433	858.1459	0.0850	0.0330
18.0000	2.5000	127.5129	0.0000	8661.1299	648.6219	1468.5637	852.6923	0.0732	0.0293
18.5000	2.4324	127.5129	0.0000	8289.1025	620.7612	1483.0236	853.5176	0.0645	0.0265
19.0000	2.3684	127.5129	0.0000	8434.2383	631.6303	1511.0718	866.3798	0.0606	0.0256
19.5000	2.3077	127.5129	0.0000	8578.7510	642.4527	1539.6882	878.9135	0.0570	0.0247
20.0000	2.2500	127.5129	0.0000	8678.5098	649.9235	1567.7725	890.0311	0.0534	0.0237

REFERENCES

[1] Z. Du and M. Selig, "A 3-D stall-delay model for horizontal axis wind turbine performance prediction," *1998 ASME Wind Energy Symp.*, p. 8551262, 1998.

[2] "BP Statistical Review of World Energy About this review," no. June, 2016.

[3] G. M. J. Herbert, S. Iniyan, E. Sreevalsan, and S. Rajapandian, "A review of wind energy technologies," vol. 11, pp. 1117–1145, 2007.

[4] G. Pavesi, *Wind energy systems*. 2013.

[5] S. Gamesa, "SG 3.4 - 132 Datasheet," 2018.

[6] AEE, "EÓLICA 2018," 2018.

[7] REE, "EL SISTEMA ELÉCTRICO ESPAÑOL. Previsión de cierre 2018," 2018.

[8] M. Ósk Óskarsdóttir, "A General Description and Comparison of Horizontal Axis Wind Turbines and Vertical Axis Wind Turbines," 2014.

[9] Carbontrust, "Small-scale wind energy Policy insights and practical guidance." 2008.

[10] CanWEA, "Small Wind Turbine Purchasing Guide: Off-grid, Residential, Farm & Small Business Applications," *Can. Wind Energy Assoc.*

[11] "IEC 61400-2. Wind Turbines Part 2: Design requirements for small wind turbines," 2006.

[12] "Home - Bergey Wind PowerBergey Wind Power | Small Wind Turbines For Homes, Farms and Small Businesses." [Online]. Available: http://bergey.com/.

[13] C. Herrero Novoa, "Estudio de la velocidad del viento: análisis espacial y ajuste a una función de distribución," 2017.

[14] "IEC 61400-1. Wind Turbine—Part 1: Design Requirements," vol. 3, p. 60, 2005.

[15] AEMET, "Meteorología y climatología de Navarra." [Online]. Available: http://meteo.navarra.es/estaciones/mapadeestaciones.cfm. [Accessed: 11-Nov-2018].

[16] M. Meadors and D. Corbus, "Small Wind Research Turbine," 2005.

[17] Aemet, "Recurso eólico - Meteo Navarra. Mapas potencia de viento." [Online]. Available: http://meteo.navarra.es/energiasrenovables/mapaspotenciaviento.cfm. [Accessed: 01-Jan-2019].

[18] C. Moreno Figueredo, "¿Cómo medir la potencialidad del viento?" [Online]. Available: http://www.cubasolar.cu/biblioteca/energia/Energia55/HTML/articulo03.htm. [Accessed: 01-Jan-2019].

[19] et al D. Elliot, "Atlas de Recursos Eólicos del Estado de Oaxaca," 2008.

[20] J. F. Manwelll, J. G. McGowan, and A. L. Rogers, *Wind Energy Explained – Theory, Design and Application*, vol. 2. 2002.

[21] M. Khaled, "Aerodynamic Design and Blade Angle Analysis of a Small Horizontal–Axis Wind Turbine," *Am. J. Mod. Energy*, vol. 3, no. 2, p. 23, 2017.

[22] C. J. Bai, F. B. Hsiao, M. H. Li, G. Y. Huang, and Y. J. Chen, "Design of 10 kW horizontal-axis wind turbine (HAWT) blade and aerodynamic investigation using numerical simulation," *Procedia Eng.*, vol. 67, pp. 279–287, 2013.

[23] Y. I. N. Yin, H. Aung, M. M. Soe, and A. M. Thu, "Design and Performance Testing of a Small-Scaled Horizontal Axis Wind Turbine for Low Wind Speed," no. October, pp. 14–19, 2016.

[24] S. G. Gong, "Numerical analysis of the wake of a 10kW HAWT," *Fire Rescue Mag.*, vol. 25, no. 1, p. 6, 2017.

[25] P. Giguere and M. S. Selig, "Low Reynolds Number Airfoils for Small Horizontal Axis Wind Thrbines," *Wind Eng.*, vol. 21, no. 6, pp. 367–380, 1997.

[26] "Airfoil Tools." [Online]. Available: http://airfoiltools.com/. [Accessed: 01-Nov-2018].

[27] D. M. Somers, "The S822 and S823 Airfoils October 1992 — December 1993," *Natl. Renew. Energy Lab.*, no. January, 2005.

[28] C. J. Bai, P. W. Chen, and W. C. Wang, "Aerodynamic design and analysis of a 10 kW horizontal-axis wind turbine for Tainan, Taiwan," *Clean Technol. Environ. Policy*, vol. 18, no. 4, pp. 1151–1166, 2016.

[29] J. O. Mo, A. Choudhry, M. Arjomandi, and Y. H. Lee, "Large eddy simulation of the wind turbine wake characteristics in the numerical wind tunnel model," *J. Wind Eng. Ind. Aerodyn.*, vol. 112, no. March 2018, pp. 11–24, 2013.

[30] L. J. F. M.M. Hand, D.A. Simms, A. D.W. Jager, J.R. Cotrell, S. Schreck, and S. M. Larwood, "Unsteady Aerodynamics Experiment Phase VI: Wind Tunnel Test Configurations and Available Data Campaigns," *NREL/TP-50*, 2001.

[31] P. Giguère and M. S. Selig, "New Airfoils for Small Horizontal Axis Wind Turbines," *J. Sol. Energy Eng.*, vol. 120, no. 2, p. 108, 1998.

[32] C. Lindenburg, "Modelling of rotational augmentation based on engineering considerations and measurements," *Eur. Wind Energy Conf.*, no. November, pp. 22–25, 2004.

[33] H. Snel, R. Houwink, and J. Bosschers, "Sectional prediction of 3d effects for stalled flow on rotating blades and comparison with measurements," *ECN-C-93-052*, 1993.

[34] C. Bak, J. Johansen, and P. Andersen, "Three-dimensional corrections of airfoil characteristics based on pressure distributions," *Eur. Wind Energy Conf. Exhib.*, pp. 1–10, 2006.

[35] NREL's National Wind Technology Center, "NWTC Forum." [Online]. Available: https://wind.nrel.gov/forum/wind/index.php.

[36] L. A. Viterna and D. C. Janetzke, "Theoretical and Experimental Power From Large HorizontaI-Axis Wind Turbines," *Tech. Rep. N82-33830*, vol. 82944, 1982.

[37] F. Mahmuddin, "The Effect of Flat Plate Theory Assumption in Post-Stall Lift and Drag Coefficients Extrapolation with Viterna Method," *J. Subsea Offshore*, vol. 6, pp. 9–13, 2016.

[38] M. Jahangiri and A. A. Shamsabadi, "Designing a horizontal-axis wind turbine for South Khorasan province: A case study," *Int. J. Precis. Eng. Manuf.*, vol. 18, no. 10, pp. 1463–1473, 2017.

[39] X. Tang, X. Huang, R. Peng, and X. Liu, "A direct approach of design optimization for small horizontal axis wind turbine blades," *Procedia CIRP*, vol. 36, pp. 12–16, 2015.

[40] C. Thumthae, "Optimum Blade Profiles for a Variable-Speed Wind Turbine in Low Wind Area," *Energy Procedia*, vol. 75, pp. 651–657, 2015.

[41] C. Zhang and H.-P. Chen, "Aerodynamic performance assessment of wind turbine composite blades using corrected blade element momentum method," *2017 World Congr. Adv. Struct. Eng. Mech.*, 2017.

[42] B. Kim, W. Kim, S. Bae, J. Park, and M. Kim, "Aerodynamic design and performance analysis of multi-MW class wind turbine blade," *J. Mech. Sci. Technol.*, vol. 25, no. 8, pp. 1995–2002, 2011.

[43] D. Wood, *Small Wind Turbines: Analysis, Design and Application*. 2012.

[44] T. Burton, D. Sharpe, N. Jenkins, and E. Bossanyi, *WIND ENERGY HANDBOOK*. 2001.

[45] E. Brandlard, "Wind turbine tip-loss corrections," 2013.

[46] P. J. Moriarty and A. C. Hansen, "AeroDyn Theory Manual, NREL/TP-500-36881, Golden, Colorado: National Renewable Energy Laboratory," no. December, 2005.

[47] A. Matiz, "Development of a model for computing tip loss corrections in wind turbines," 2013.

[48] H. Glauert, "Airplane propellers, division 1," in *Aerodynamic Theory 4*, 1935, pp. 169–360.

[49] R. E. Wilson, B. S. Lissaman, and S. N. Walker, "Aerodynamic performance of wind turbines," p. 39, 1976.

[50] M. L. Buhl, "A New Empirical Relationship between Thrust Coefficient and Induction Factor for the Turbulent Windmill State," 2005.

[51] Enair, "E200L Wind Turbine DATA SHEET."

[52] J. Manwell, "Wind Turbine Noise Issues," no. July 2002, 2014.

[53] C. D. Botha, "Variable Speed and Torque Control of a Wind Turbine System with Assisted Reluctance Synchronous Generator Technology by," no. March, 2018.

[54] M. Ragheb and A. M. Ragheb, "The Betz equation and optimal rotor tip speed ratio," *Wind turbine theory*, vol. 1, no. 1, 2011.

[55] M. Buhl, "NWTC Design Codes (WT_Perf)," *Math. Comput. Model.*, vol. 33, no. 10–11, pp. 1099–1112, 2001.

[56] Bergey Wind Power, "Excel 10 Specification Sheet," 2013.

[57] "Residential Wind Energy Systems - Bergey Wind PowerBergey Wind Power." [Online]. Available: http://bergey.com/wind-school/residential-wind-energy-systems. [Accessed: 08-Feb-2019].

[58] J. Garcia de Jalón and J. I. Rodriguez, "Aprenda Matlab 7 . 0 como si estuviera en primero." Escuela Técnica Superior de Ingenieros Industriales. Universidad Politécnica de Madrid, 2005.

[59] D. Marten and J. Wendler, "QBlade Guidelines - v0.5," 2013.

[60] "Lucidchart Official Website. Online flowchart creation." [Online]. Available: https://www.lucidchart.com. [Accessed: 15-Jan-2019].

[61] Biblioteca de la Universidad Pública de Navarra. Oficina de Referencia. (2016). *Competencias informacionales: buscar. evaluar y utilizar la información. Trabajo de fin de grado en Ingenierías Industriales. Informática y de Telecomunicación*. Recuperado el 4 de noviembre de 2014 de https://miaulario.unavarra.es/portal/site/ci-para-ing-tfg

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

Lower case letters

a	Axial induction factor (from 4$^{\text{th}}$ Chapter on: factor local to the blade)
a_b	Axial induction factor local to the blade
$\bar{a}$	Azimuthally averaged axial induction factor
a'	Tangential induction factor
c	Chord length (also Weibull PDF scale parameter)
h_{hub}	Hub height
h_{ref}	Reference height of the station
k	Weibull PDF shape parameter
r	Radial position along the blade

Upper case letters

A	Area
B	Number of blades
C_d	Drag coefficient
C_l	Lift coefficient
C_P	Power coefficient
C_n	Normal component of the aerodynamic coefficients
C_t	Tangential component of the aerodynamic coefficients
C_T	Thrust coefficient
D	Drag force
D_{tip}	Rotor diameter
F	Tip-loss factor
L	Lift force
M	Pitch moment at blade section
P	Power
Q	Rotor torque
R	Rotor radius (*or* R_{tip})
T	Thrust force
U_d	Wind velocity at the rotor disc

U_w	Wind velocity in the far wake
U_∞	Wind stream velocity (upstream)
W	Relative velocity in the airfoil section

Lower case Greek letters

α	Angle of attack	
β	Twist angle	
η	Efficiency	
λ	Tip speed ratio	
λ_r	Local speed ratio	
μ	Dynamic viscosity	$[\mathrm{kg} \cdot \mathrm{m}^{-1} \cdot \mathrm{s}^{-1}]$
ν	Kinematic viscosity	$[\mathrm{m}^2 \cdot \mathrm{s}^{-1}]$
ρ	Air density	
σ'	Local blade solidity	
φ	Inflow angle	
ψ	Azimuthal coordinate of the rotor	
ω	Rotational speed of the rotor [rad/s]	

Acronyms

1D	One dimension
2D	Two dimensions
3D	Three dimensions
AEP	Annual Energy Production
AR	Aspect ratio of a wind/blade
BEM	Blade Element Momentum
BEMT	Blade Element Momentum Theory
BET	Blade Element Theory
CAD	Computer Assisted Design
CFD	Computational Fluid Dynamics
HAWT	Horizontal Axis Wind Turbine
IEC	International Electrotechnical Commission
MT	Momentum Theory
NREL	National Renewable Energy Laboratory (US)
PDF	Probability Density Function
RFWBM	Root Flap-Wise Bending Moment
TSR	Tip Speed Ratio
VAWT	Vertical Axis Wind Turbine
WT	Wind Turbine