

Democratic and Popular Republic of Algeria
Ministry of Higher Education and Scientific Research



المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

National Polytechnic School
Mechanical Engineering Department

Mechanical Engineering and Development Laboratory -LGMD-

مخبر الهندسة
الميكانيكية
والتطوير
LGMD

Final year project dissertation
For obtaining State Engineer degree in Mechanical Engineering

**An Artificial Intelligence Approach to Enhance Blade
Element Momentum Theory Performance for Horizontal
Axis Wind Turbine Application**

Presented by:

Ahmed LADJAL

Suggested and supervised by **Dr. Abdelhamid BOUHELAL**

Publicly presented and defended on Thursday, October 15th, 2020

Composition of the Jury:

President	Abdelhafid HELLAL	Prof	ENP
Supervisor	Abdelhamid BOUHELAL	Dr, MAB	ENP
Co-supervisor	Arezki SMAILI	Prof	ENP
Examiner	Mohammed Amokrane MAHDI	MCB	ENP
Guest	Nacer-eddine BOUMEZBEUR	Doctorant	ENP

Democratic and Popular Republic of Algeria
Ministry of Higher Education and Scientific Research



المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

National Polytechnic School

Mechanical Engineering Department

Mechanical Engineering and Development Laboratory -LGMD-

مخبر الهندسة
الميكانيكية
والتطوير
LGMD

Final year project dissertation

For obtaining State Engineer degree in Mechanical Engineering

**An Artificial Intelligence Approach to Enhance Blade
Element Momentum Theory Performance for Horizontal
Axis Wind Turbine Application**

Presented by:

Ahmed LADJAL

Suggested and supervised by **Dr. Abdelhamid BOUHELAL**

Publicly presented and defended on Thursday, October 15th, 2020

Composition of the Jury:

President	Abdelhafid HELLAL	Prof	ENP
Supervisor	Abdelhamid BOUHELAL	Dr, MAB	ENP
Co-supervisor	Arezki SMAILI	Prof	ENP
Examiner	Mohammed Amokrane MAHDI	MCB	ENP
Guest	Nacer-eddine BOUMEZBEUR	Doctorant	ENP

Dedication

I dedicate this work

For my dear parents who supported me and encouraged me during my years of studies and helped me to reach this state

For my sisters and my little brother

All my friends who encouraged me along the realization of this work

Acknowledgment

First of all, I thank **Allah** the Almighty for giving us courage, health, and support during the most difficult times.

I would like to sincerely thank my supervisor Mr. **Abdelhamid BOUHELAL**, a PhD of polytechnic school of Algiers for guiding me, helping me and trusting in my skills.

I would like to thank sincerely my Co. supervisor Mr. **Arezki SMAILI** for his guidance and his efforts teaching us for the past three years.

I would like also to thank all the jury members, Mr. **Abdelhafid HELLAL** for accepting to chair the jury, Mr. **Mohammed Amokrane MAHDI** for accepting to examine my work and to Mr. **Nacer-eddine BOUMEZBEUR** for attending my dissertation, and for making the project rich by their propositions.

I am very much thankful to my family for their care, support and prayers.

Finally, I would like to thank all the professors and the teachers of the mechanical engineering department for transmitting their knowledge to us and the efforts that they made during our studies.

حتى الآن، تظل نظرية زخم جزء الشفرة هي الطريقة الأكثر استخدامًا للتنبؤ بالأداء الديناميكي الهوائي لتربينات الرياح نظرًا لبساطتها. تعتمد نظرية زخم جزء الشفرة بشكل أساسي على بيانات الجناح لمجموعة واسعة من الظروف (شكل الجناح وزوايا المواجهة) عادة ما يتم جمع هذه البيانات في تجارب نفق الرياح للأجنحة الثابتة عند زوايا مواجهة منخفضة. ومع ذلك، فإن توربينات الرياح الدوارة لديها زوايا مواجهة أعلى. الدافع وراء هذا العمل هو تحسين طريقة زخم جزء الشفرة الكلاسيكية من خلال تحديد معاملات أداء الجناح حيث توجد بيانات تجريبية قليلة أو معدومة مثل زوايا مواجهة العالية وشكل الأجنحة الجديدة وأرقام رينولدز المنخفضة. لهذا الغرض، تم اقتراح نهج الذكاء الاصطناعي، باستعمال الشبكات العصبية الاصطناعية، للتنبؤ بمعاملات رفع الجناح والمقاومة. أولاً، العدد الأمثل للطبقات وكذلك العدد الأمثل للخلايا العصبية تم تحديده رقمياً لتدريب بيانات الشبكات العصبية الاصطناعية. بعد ذلك، تم مقارنة نتائج طريقة التي تدمج نظرية زخم جزء الشفرة والشبكات العصبية الاصطناعية المقترحة مع النتائج التجريبية المتاحة للتحقق من صحة الطريقة. توافق جيد تم الحصول عليه.

الكلمات الدالة: توربينات الرياح الأفقية، نظرية زخم جزء الشفرة، الشبكات العصبية الاصطناعية، زاوية مواجهة، عدد للخلايا العصبية.

Résumé

Jusqu'à présent, la théorie du moment d'élément de pale et quantité de mouvement (BEM) reste la méthode la plus largement utilisée dans la prédiction des performances aérodynamiques des éoliennes à axe horizontale (HAWTs) en raison de sa simplicité. La théorie de BEM est basée sur les données du profil de pale (la géométrie du profil, l'angle d'attaque (AOA)). Ces données sont généralement collectées dans des expériences en soufflerie pour des profils d'ailes stationnaires à des angles d'attaque faibles. Cependant une éolienne en rotation possède des angles d'attaque plus élevés. La motivation derrière ce travail est d'améliorer la méthode BEM classique en déterminant les coefficients de performance aérodynamique lorsqu'il existe peu ou pas de données expérimentales, comme des angles d'attaque élevés des nouveaux profils et pour de faible nombre de Reynolds. A cet effet, une approche d'intelligence artificielle appelée les réseaux de neurones (ANNs) a été proposée pour prédire les coefficients de portance et de traînée des profils. Tout d'abord, le nombre optimum des neurones et des couches discrètes ont été sélectionnés numériquement pour l'apprentissage des réseaux de neurones. Ensuite, les résultats de la méthode BEM-ANN proposée ont été comparés avec les résultats expérimentaux disponibles afin d'étudier sa validité. De bons accords ont été obtenus entre les prédictions numériques et l'expérimental.

Mots Clés : Eoliennes à axe horizontale, Théorie BEM, Réseaux de neurones artificiels (ANNs), Angle d'attaque (AOA), Nombre de neurones.

Summary

So far, the Blade Element Momentum (BEM) theory remains the most widely used method for predicting aerodynamic performance of horizontal axis wind turbines (HAWTs) owing to its simplicity. The BEM theory is mainly based on airfoils data for wide range of conditions (airfoil shape, angles of attack (AOAs)). These data are usually collected in wind tunnel experiments for stationary airfoils at low AOAs. However, a rotating wind turbine, has higher AOAs. The motivation behind this work is to improve the classical BEM method by determining the airfoil performance coefficients where little or no experimental data exists such as at high AOAs, new airfoils shape and for low Reynolds numbers. For this purpose, an artificial intelligence approach, namely Artificial Neural Networks (ANNs) is proposed for predicting the airfoils lift and drag coefficients. Firstly, the optimum number of layers as well as the optimum neurons number for training input-output data have been selected numerically. Afterwards, the results of the proposed BEM-ANN method have been compared with available experimental results in order to investigate its validity. Good agreements were obtained between numerical predictions and experimental results.

Keywords: Horizontal Axis Wind Turbines (HAWT), BEM theory, Artificial Neural Networks (ANNs), Angle of attack (AOA), Neurons number.

Table of content

Nomenclature

Figures list

Tables list

General introduction.....	12
Chapter 1: Aerodynamics of Wind turbines.....	15
1.1 Brief history of wind turbines.....	16
1.2 Types and classifications of wind turbines.....	16
1.2.1 Horizontal axis wind turbines HAWTs.....	17
1.2.2 Vertical axis wind turbines VAWTs.....	17
1.3 Aerodynamic methods for HAWTs.....	18
1.3.1 Vortex Methods.....	18
1.3.2 Generalized Actuator Disk Methods.....	19
1.3.3 Full rotor method.....	21
1.4 BEM corrections models.....	23
1.4.1 Tip loss factor models.....	23
1.4.2 Stall delay models.....	23
1.4.3 Dynamic stall models.....	25
1.5 Aerodynamics of HAWTs airfoil.....	26
1.5.1 Geometric characteristics.....	26
1.5.2 Classification of profiles.....	27
1.5.3 Airfoil performance.....	29
1.6 Literature review.....	31
1.6.1 Methods for Determining Lift and Drag.....	31
1.6.2 Previous works on BEM models.....	32
1.6.3 Available experiments and data.....	33
Chapter 2: Artificial neural networks.....	34
2.1 Introduction.....	35
2.2 Learning methods.....	36
2.2.1 Supervised training method.....	36
2.2.2 Unsupervised training method.....	38
2.3 Artificial neural network methods.....	39
2.4 ANN architecture.....	40
2.4.1 Hidden neurons.....	40
2.4.2 Hidden layers.....	40
2.4.3 Activation function.....	42

2.4.4	Training functions	46
2.4.5	Forward propagation	48
2.4.6	Cost function	49
2.4.7	Backpropagation.....	49
2.4.8	Cross validation.....	50
2.5	ANN algorithms	51
2.5.1	Initialization	52
2.5.2	Training and testing.....	53
Chapter 3: Mathematical formulation and numerical modeling.....		56
3.1	General momentum theory	57
3.1.1	Axial forces	57
3.1.2	Rotation annular stream tube.....	58
3.2	Blade element theory	59
3.3	Blade element momentum theory.....	60
3.4	Data Collection	62
3.5	Artificial neural network modeling	63
3.5.1	XY model	63
3.5.2	Airfoil characteristics model	64
3.5.3	The image model.....	64
3.6	Training algorithm	65
3.7	BEM-ANN integration	66
Chapter 4: Results and discussion		67
4.1	Models choice and validation of lift and drag coefficient	68
4.2	ANN performance	76
4.2.1	Effect of the number of neurons.....	76
4.2.2	Effect of the number of hidden layers	77
4.2.3	Effect of the training methods.....	77
4.3	BEM-ANN integration	78
General conclusion		82
References		84

Nomenclature

Abbreviations

ANN	Artificial neural network
BEM	Blade element momentum theory
HAWT	Horizontal axis wind turbine
VAWT	Vertical axis wind turbine
MSE	Mean squared error
MRE	Mean relative error
AOA	Angle of attack
ML	Machine learning
HL	Hidden layer

Symbols

T	Thrust of the wind turbine (N)
P	Power of the wind turbine (W)
\dot{m}	Flow rate (kg/s)
V_0	Inlet airflow velocity of the stream tube (m/s)
u_1	Outlet airflow velocity of the stream tube (m/s)
u	Airflow velocity in the actuator disk (m/s)
a	Axial interference factor
a'	Angular induction factor
c_p	Power coefficient
c_t	Thrust coefficient
ρ	Airflow density (kg/m ³)
A	Actuator disk area (m ²)
Δp	Pressure gradient (Pa)
Ω	Angular rotation (rad/s)
ω	Angular velocity of airflow (rad/s)
ϕ	Activation function
α	Angle of attack (degree)
θ_p	Initial blade twist (degree)
C_l	Lift coefficient
C_d	Drag coefficient
F_n	Normal force on the blade (N)
F_t	Tangential force on the blade (N)
Q	Axial force (N)
U_n	Normal velocity on the blade (m/s)
U_t	Tangential velocity on the blade (m/s)
U_{rel}	Relative velocity on the blade (m/s)
GNU	Open software license
GPL	Open software license
CFD	Computational fluid dynamics
w	Weights of the neural network
b	Bias of the neural network
x	Feature of the neural network

C	Cost function
p_2	Pressure in front of the actuator disk (Pa)
p_1	Pressure in the back of the actuator disk (Pa)
U_1	Inlet airflow velocity of the stream tube (m/s)
U_2	Velocity in front of the actuator disk (m/s)
U_3	Velocity in the back of the actuator disk (m/s)
U_4	Outlet airflow velocity of the stream tube (m/s)
L	Angular momentum (kg.m ² /s)
I	Moment of inertia (kg.m ²)
λ_r	Local tip speed ratio
c	Chord length (m)
B	Number of the blades
C_n	Normal force coefficient
C_t	Tangential force coefficient
F	Tip losses correction factor
R	Rotor radius (m)
Re	Reynolds number

Figures list

Figure 1-1: Wind turbines types.....	16
Figure 1-2: Horizontal axis wind turbine diagram.....	17
Figure 1-3: Vertical axis wind turbine types diagram.....	18
Figure 1-4: Velocity induced at point P by a vortex segment. [31]	18
Figure 1-5: Actuator disk stream tube. [32]	19
Figure 1-6: The distribution of the agnitude for an ideal wind turbine.[33]	20
Figure 1-7: Stream tube expansion, velocity and pressure through the actuator disk.[34]	20
Figure 1-8: Full rotor meshing [7].....	21
Figure 1-9: Blade subdivisions.....	22
Figure 1-10 : BEM iteration algorithm.[43].....	22
Figure 1-11: Helical wake pattren.....	23
Figure 1-12: Stall delay diagram.[35]	23
Figure 1-13: Velocity distribution on a boundary layer.[36]	24
Figure 1-14: Boundary layer characteristics.[37].....	24
Figure 1-15: Pressure gradient in a boundary layer.[38].....	25
Figure 1-16: Airfoil characteristics.....	26
Figure 1-17: Lift and drag Forces on airfoil.....	27
Figure 1-18: Velocity triangle on airfoil.[34].....	27
Figure 1-19: Symmetrical and non-symmetrical airfoils [39].....	28
Figure 1-20: 4 digits NACA airfoil.....	28
Figure 1-21: 5 digits NACA airfoil.....	29
Figure 1-22: 6 digits NACA airfoil.....	29
Figure 1-23: Pressure distribution on airfoil.[40].....	30
Figure 1-24: lift and drag coefficient of different camber line configurations [19].....	31
Figure 2-1: Artificial intelligence domain.....	35
Figure 2-2: salary vs. experience plot.....	37
Figure 2-3: E-mail classification.....	38
Figure 2-4: Human brain neuron.....	39
Figure 2-5: one neuron representation.....	39
Figure 2-6: Single neuron representation.....	40
Figure 2-7: Multiple hidden layer representation.....	41
Figure 2-8: one hidden layer representation.....	41
Figure 2-9: ANN representation.....	42
Figure 2-10: Activation function diagram.....	43
Figure 2-11: Threshold function graph.....	43
Figure 2-12: Linear function graph.....	44
Figure 2-14: Tanh function graph.....	45
Figure 2-15: ReLU function graph.....	45
Figure 2-16: Two hidden layers forward propagation.....	48
Figure 2-17: Three hidden layers forward propagation.....	48
Figure 2-18: Back propagation representation.....	50
Figure 2-19: Data splitting.....	51
Figure 2-20: Neural network start window [30].....	51
Figure 2-21: Neural network data manager [30].....	52
Figure 2-22: Importing data on NNTOOL [30].....	52
Figure 2-23: Creating the neural network [30].....	53
Figure 2-24: Training window [30].....	54

Figure 3-1: Stream tube representation.[34]	57
Figure 3-2: Annular stream tube representation.[41]	58
Figure 3-3: Airfoil's forces representation.[42].....	59
Figure 3-4: 1st model diagram.	63
Figure 3-5: 2nd model diagram.	64
Figure 3-6: 3rd model diagram.....	64
Figure 3-7: BEM-ANN integration diagram.....	66
Figure 4-1: NACA 0012 Cl prediction.....	70
Figure 4-2: NACA 0012 Cd prediction.....	70
Figure 4-3: S1012 Cl & Cd prediction.	71
Figure 4-4: E297 Cl & Cd prediction.....	71
Figure 4-5: Boeing 737b Cl & Cd prediction.....	72
Figure 4-6: E176 Cl & Cd prediction.....	72
Figure 4-7: GOE 443 Cl & Cd prediction	73
Figure 4-8 : LOCKHEED C-141 Cl & Cd prediction.....	73
Figure 4-9 : Boeing 106 Cl & Cd prediction.....	74
Figure 4-10 : E186 Cl & Cd prediction.....	74
Figure 4-11 : E1098 Cl & Cd prediction.....	75
Figure 4-12: STCYR 172 Cl & Cd prediction.	75
Figure 4-13: NREL phase IV performance.	79
Figure 4-14: NREL phase IV with NACA0012 blade in 3D	79
Figure 4-15: Extrapolated Cl data.	80
Figure 4-16: Extrapolated Cd data	80
Figure 4-17: Classical BEM vs BEM-ANN.....	80

Tables list

Table 1: Training functions characteristics [29].	47
Table 2: 2nd model data input template.	62
Table 3 : 2nd model data input 2 template.	63
Table 4: iterations of number of neurons inside the hidden layers.	65
Table 5: Performance of the XY model.	68
Table 6: Cl ANN performance.	69
Table 7: Cd ANN performance.	69
Table 8: Cl training information.	69
Table 9: Cd training information.	69
Table 10: NACA 0012 results.	70
Table 11: S1012 results.	71
Table 12: E297 results.	71
Table 13: Boeing 737b results.	72
Table 14: E176 results.	72
Table 15: GOE 443 results	73
Table 16 : LOCKHEED C-141 results.	73
Table 17 : Boeing 106 results.	74
Table 18 : E186 results.	74
Table 19 : E1098 results.	75
Table 20 : STCYR 172 results.	75
Table 21: Comparaison of the results of one hidden layer.	76
Table 22: Comparaison of the results of three hidden layers.	76
Table 23: Comparaison of the results of seven hidden layer.	77
Table 24: Comparaison of the results of diffrent combinaison of hidden layer.	77
Table 25: Comparaison of the results of diffrent traning methods.	78
Table 26: NACA0012 blade data	79

General introduction

General introduction

The wind energy is one of the essential green energy in the world nowadays, in the presence of global warming and the high amount of pollution. The wind energy is a good alternative source rather than oil and gaz. The increase consumption of the energy in the world, about 158839 TWh consumed in 2019 [1]. 33% produced by the oil and 25% produced by gaz. This number and percentages make us thinking about decreasing these two high percentages of non-green sources of energy and replace them with the eco-sources.

In 2019, Europe installed 15.4 GW of new wind power capacity. 27% higher than the capacity installed in 2018 and less than the record in 2017, and for now, it has a capacity of 205 GW of wind energy about 15% of the accounted electricity of the Europe consumed in 2019. An amount of three quarters of the wind energy installed in the last year in earth was onshore. New onshore installations of 2.2 GW of wind farms were installed in Spain, and new investments were done in Germany [2].

The exploitation of the wind turbines provides a clean and a prospective option for energy production. Therefore, a good understanding of how the wind turbines works and interacts with wind is very essential in order to improve their performance and reduce their cost.

Aerodynamics is the backbone and the foundation for the development and optimization of wind turbines. Wind turbine aerodynamics is generally divided into two main parts; rotor aerodynamics and wake aerodynamics. The first part aids in the design, development and optimization of wind turbine blades, whereas the other part predicts the capacity of wind farms and allows investigating the interaction of multiple wind turbines.

In order to study wind turbine rotor aerodynamics, one needs to be able to model the rotor at its operating conditions and interaction with the wind to a certain degree of accuracy. Several techniques exist by which this is traditionally done, e.g. Computational Fluid Dynamics (CFD) methods, vortex methods, and Blade Element Momentum (BEM) methods, each having their own advantages.

The Blade Element Momentum (BEM) remains the most widely used method for predicting aerodynamic performance of horizontal axis wind turbines (HAWTs) due to its simplicity. However, the BEM methods are based essentially on the introduction of the aerodynamic coefficients (drag and lift) as a function of the angle of attack as an input parameter. While these aerodynamics coefficients are not always available.

In the present study, an artificial intelligence approach based on the so-called artificial neural network (ANN) has been proposed, implemented and validated in order to predict the lift and

the drag coefficients. After that, this approach was coupled with the BEM theory in order to determine the performance of HAWTs in the case where the profile's data are not available.

This method can make the calculation of the performance in few minutes. In addition, the use of new airfoil profiles newly developed by researchers is possible since the ANN can predict the lift and drag coefficient, so it makes the determination of the wind turbine performance fast allowing the researchers to make many tests of the profile geometries in a very low time.

This report has been divided into four chapters organized as follows:

Chapter 1 presents some general and basic concepts commonly used in the analysis of the aerodynamics of wind turbines followed by the literature review.

Chapter 2 focuses on the theory behind the artificial neural network and its main features.

Chapter 3 presents the mathematical formulation and the numerical modeling of both artificial neural network and BEM method.

Chapter 4 presents and discusses the obtained results, which can be divided into two parts; drag and lift coefficients prediction and ANN-BEM validation.

Finally, the main conclusions found in this study and the main recommendations relevant to the future work will ultimately be addressed towards the end of this report.

Chapter 1: Aerodynamics of Wind turbines

Chapter 1: Aerodynamics of Wind turbines

1.1 Brief history of wind turbines

The wind turbines are an old technology appeared in seventh century in Europe, it was used that time to grind flour, corn in windmills and to pump water. They were composed of a rotor with multiple blades constructed in wood and a transmission system to accomplish the desired task.

The first rotor was fixed and oriented to one direction, and then an improved windmill had a movable part to face the incoming wind.

In United States, according to a Californian program encouraging the use of green energy, the first park of modern wind turbines was installed to produce electricity in 1980's. These machines generated 55 kW of power. Then the use of this green energy was highly explored in the world, new types of wind turbines and different power generating ranges were developed.

1.2 Types and classifications of wind turbines

There are two primary types of the wind turbine used to produce energy from wind. Horizontal axis wind turbines HAWTs and vertical axis wind turbines VAWTs. The horizontal axis wind turbine type is the most used type. It contains generally three blades.

A high solidity HAWT is used for pumping water on farms seen on the west of America, it has multiples of blades. Low solidity HAWT are used for electricity generation they have long blades and resemble to aircraft propellers.

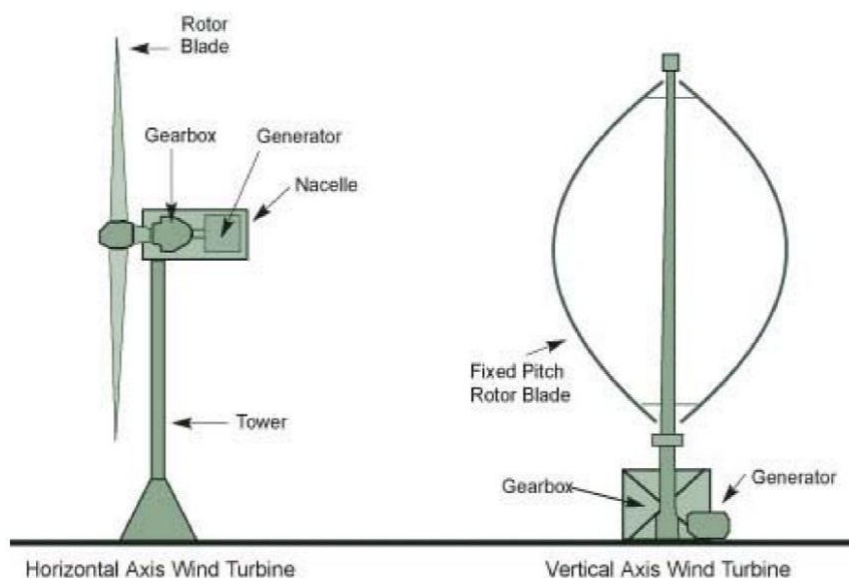


Figure 1-1: Wind turbines types.

1.2.1 Horizontal axis wind turbines HAWTs

They have at the top of the tower a main rotor shaft and an electric generator; they should be directed parallel to the wind direction with a simple wind vane placed with the rotor. The large wind turbines have sensor coupled with a servomotor to guide the turbine into the wind. Most of the wind turbines have a gearbox to increase the rotation of the generator in case of a slow rotation of the rotor.

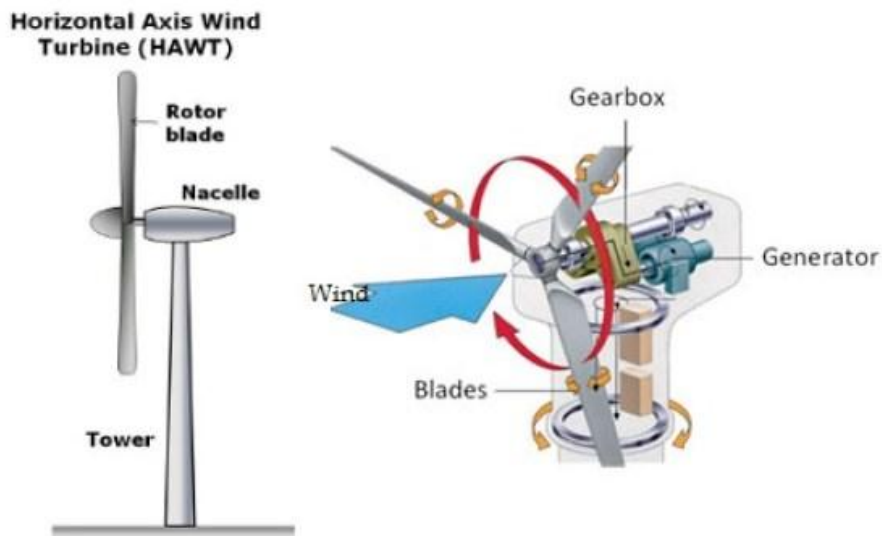


Figure 1-2: Horizontal axis wind turbine diagram.

The HAWT has some advantages such as a good access to strong wind thanks to the tall tower, every ten meters up increases the speed by 20% and the generated power by 34%. It has also a high efficiency since that the blades are perpendicular to the wind.

Whereas, the tall tower requires massive constructions to support the load of the blades and the generator. They disturb the appearance of the landscape since that they are visible across large areas. It is also necessary to make a braking system to stop the turbine from rotation with very high speed in order to avoid damaging or the destruction of its structure.

1.2.2 Vertical axis wind turbines VAWTs

They have a vertical rotor shaft with a vertical axis, this allows us to place the generator and the essential components near the ground, making their maintenance very easy comparing to HAWTs. The airflow near the ground is turbulent and can introduce vibrations and noise issues.

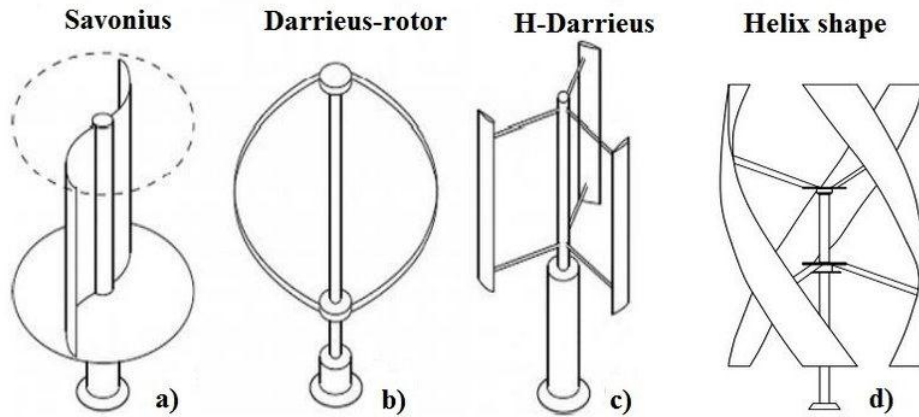


Figure 1-3: Vertical axis wind turbine types diagram.

The advantages of VAWTs are low maintenance costs, they can be installed in urban areas, they generate electricity in any wind direction and can be easily installed comparing to HAWTs.

Whereas, their efficiency is very low compared to HAWTs, they need initial push to start and they create noise and vibrations.

1.3 Aerodynamic methods for HAWTs

1.3.1 Vortex Methods

These types of models are based on potential flow theory. Using the superposition of elementary flow solutions, a linearized potential solution used to calculate the velocities and pressure. This model is based on the so-called Biot-Savart law. The velocity at any point is calculated by the formula

$$\vec{v} = \frac{\Gamma}{4\pi} \int \frac{d\vec{l} (\vec{r}_0 - \vec{r}_1)}{|\vec{r}_0 - \vec{r}_1|^3}$$

Where Γ is the circulation about the curved C in a velocity field \vec{v} .

$$\Gamma = \oint_C \vec{v} \cdot d\vec{l}$$

The blade is modeled with a lifting line presenting bound vorticity on the blade.

Lift can determinate the bound circulation on a blade section conforming to the Kutta-Joukowski theorem.

$$L' = \rho V \Gamma$$

Where L' is the lift span at a particular blade section, V is the incoming flow velocity.

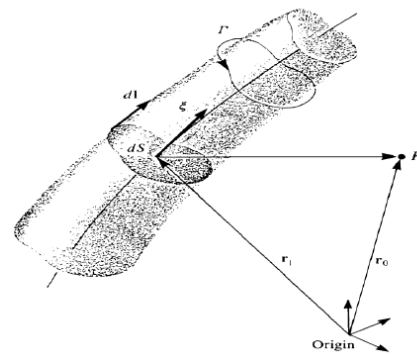


Figure 1-4: Velocity induced at point P by a vortex segment. [31]

The vortex model is divided into two types, the frozen wake model in which the geometry of the wake is prescribed on the rotor geometry and undisturbed flow; the free wake model is the second category, in which the wake self-induction and bound circulation on the blade is taken into account resulting in deformation wake.

1.3.2 Generalized Actuator Disk Methods

The idea of actuator disk principal in aerodynamic calculation is to replace the rotor of the wind turbine by a permeable disc of its equivalent surface, in which the forces are distributed in a circular disk. The balance between the changed flow field and the applied forces on the actuator disk is commanded by the mass conservation and the balance of momentum.

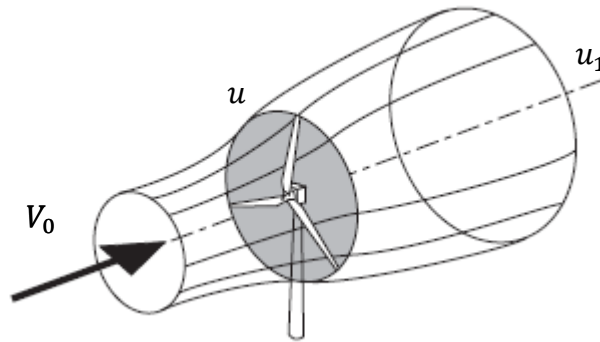


Figure 1-5: Actuator disk stream tube. [32]

The Rankine-Froude theory considers the balance of axial-momentum for a uniformly loaded disk actuator without rotation, the thrust and power are reduced to:

$$T = \dot{m}(V_0 - u_1) ; P = \frac{1}{2} \dot{m}(V_0^2 - u_1^2) \quad (1.1)$$

In non-dimensional form, the axial interference factor $a = 1 - \frac{u}{V_0}$ and $u = \frac{V_0 + u_1}{2}$

The power coefficient is given by:

$$C_p = \frac{P}{\frac{1}{2} \rho V_0^3 A} = 4a(1 - a)^2 \quad (1.2)$$

We can prove that the maximum theoretical power coefficient is equal to $C_{p,max} = \frac{16}{27}$ at $a = \frac{1}{3}$. This is known by Betz limit.

In addition, the thrust coefficient is given by:

$$C_T = \frac{T}{\frac{1}{2} \rho V_0^2 A} = 4a(1 - a) \quad (1.3)$$

The maximum thrust coefficient is equal to one when $a = \frac{1}{2}$.

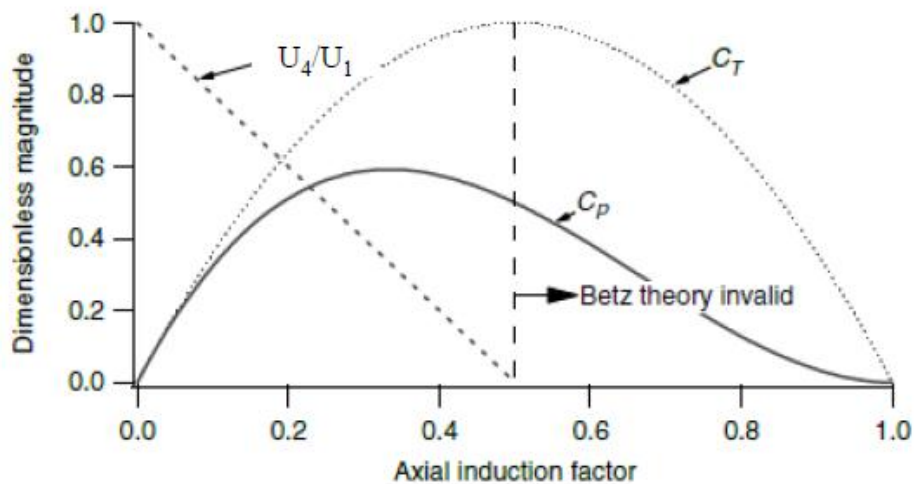


Figure1-6: The distribution of the agnitude for an ideal wind turbine.[33]

The distribution of the pressure and the velocity in the stream tube of the actuator disk is presented in the next figure.

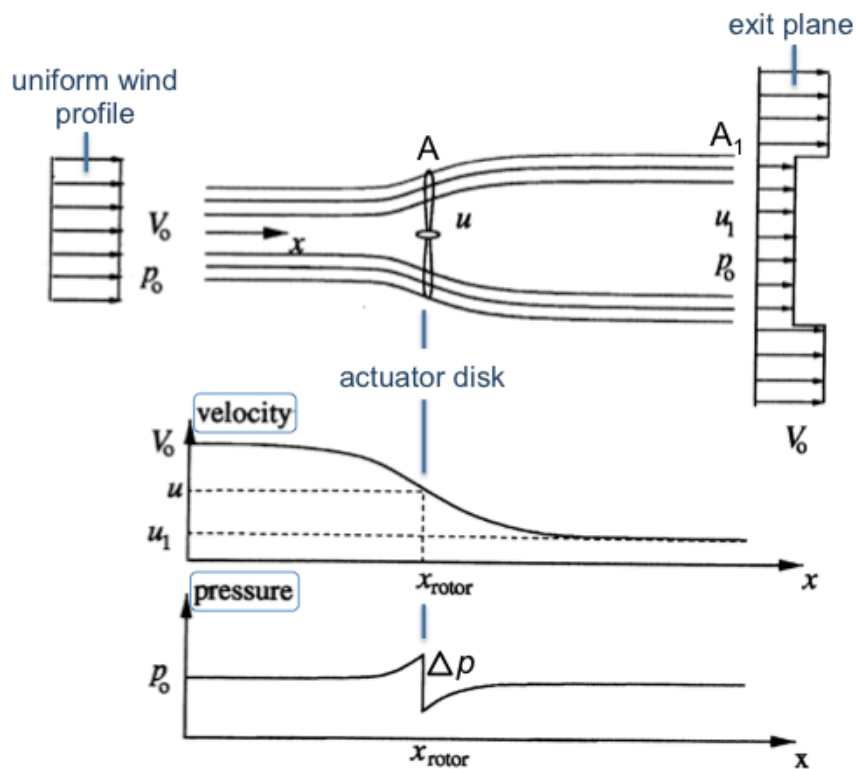


Figure 1-7: Stream tube expansion, velocity and pressure through the actuator disk.[34]

The incoming airflow has a magnitude of V_0 crosses the actuator disk, the velocity decreases continuously while the actuator extracts the energy from wind, and it is reduced to u_1 at the exit while the velocity far from the wind turbine remains constant. Whereas, the pressure distribution is not similar, it presents a discontinuity at the actuator disk. The static pressure around the wind turbine is equal to p_0 - the atmosphere pressure -, but the decreasing in the

velocity makes the static pressure rises according to Bernoulli's Law which is represented by a pressure gradient Δp .

The flow after the actuator disc is affected by the rotation of the blades, a tangential flow is represented by an angular induction factor a' .

$$a' = \frac{\omega}{2\Omega} \quad (1.4)$$

Where ω is the angular velocity imparted to the wake and Ω is the angular rotation of the rotor disk..

The generalized actuator disk method is based on solving the Navier-Stokes equations or Euler equation. Axisymmetric versions were solved by scientist such as Wu [4], Greenberg [5] and Conway [6].

1.3.3 Full rotor method

The full rotor method is taking on consideration the full geometry of the wind turbine in the determination of its performance; the rotor, the hub, the nacelle and the blades are designed in 3D using a CAO software.

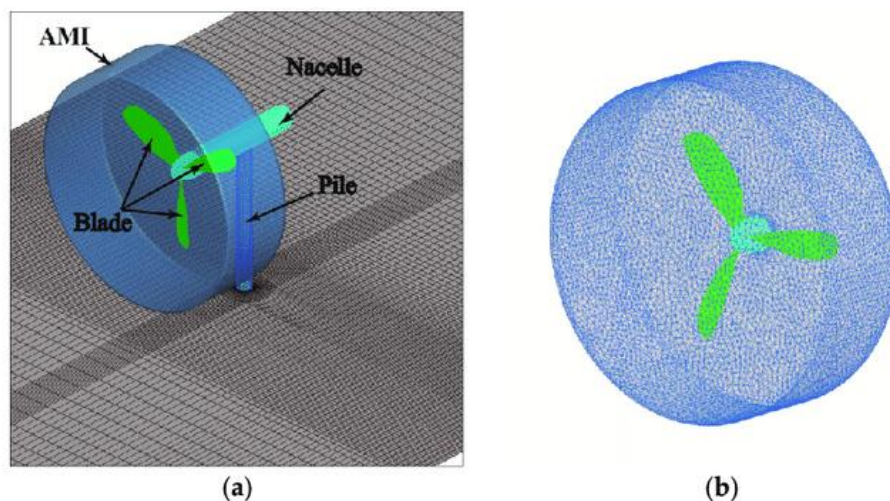


Figure 1-8: Full rotor meshing [7].

This method can simulate the real case of the operation of the wind turbine, and the real airflow condition, these are some of its advantages, but the disadvantage of this method is the long time and the huge hardware capacity required for calculations.

1.3.4 Blade Element Momentum Methods (BEM)

The Blade element momentum theory is a combination of the blade element theory and the momentum theory developed by Glauert and Betz. It allows us to make performance characteristics calculation of an annular section of the rotor.

The blade element theory mainly consists in subdividing the blade to N finite elements of profile sections. We suppose that there is no interaction between them in order to calculate the power and the trust of each element. Knowing that the blade can be twisted or may has more than one

airfoil type. Therefore, to take these in consideration, the BEM method remains the most used method to calculate the wind turbine performance.

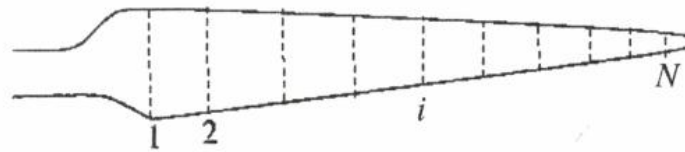


Figure 1-9: Blade subdivisions.

The momentum theory corrects the velocities by introducing the induction factor. It is divided into an axial momentum and tangential momentum, the axial momentum theory is based on the conservation laws applied to a stream tube of an actuator disk considering the axial forces and the axial inductors, and the tangential momentum results by a relation between the angular factor and the plane forces.

This method depends on the rotor blade design parameters such as:

- Rotor radius
- Blade chord
- Blade twist
- Number of blades
- Airfoil shape
- And the radial variation of all these parameters

The torque is equal to the aerodynamic force in the rotation plan times its distance from the rotation axis. The trust force is perpendicular to the plane of blade rotation in any section of the blade.

So, the BEM method is an iterative method, it iterates on the number of the subdivision of the blade. When we calculate the power generated by each element of the blade, we make a summarization to get the total power generated by the wind turbine.

The next diagram shows the steps to take in this iterative method.

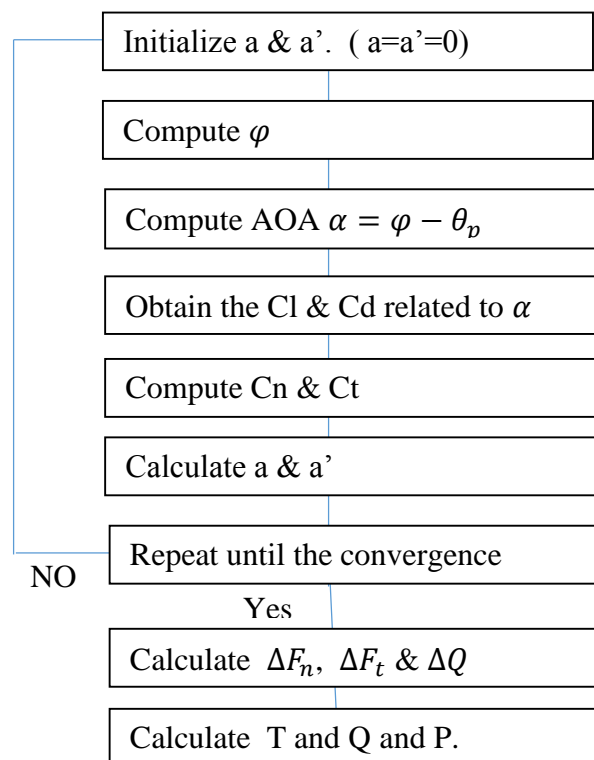


Figure 1-10 : BEM iteration algorithm.[43]

1.4 BEM corrections models

The classical BEM method does not take in consideration the losses phenomenon, which happens in the blade, like the stall delay and the tip losses. In order to make a realistic study and calculation with add some correction to the BEM method.

1.4.1 Tip loss factor models

The tip losses are caused by the tip vortices that create multiple helical structure in the wake as seen in the next figure.

This helical wake has an effect on the induced velocity near the tips of the blades. This area has the greatest influence in the power that the wind turbine produces.

To compensate these losses Prandtl developed a theory that simplify the wake of the turbine, this one is summarized by a correcting factor F.

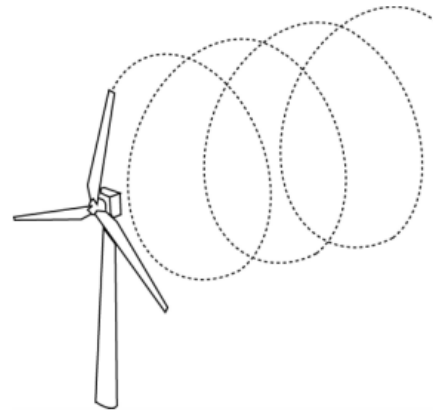


Figure 1-11: Helical wake patten.

Later on, an empirical relationship based on the Navier-Stokes solution of Xu and Sankar [8] for tip losses is considered as a correction of the Prandtl model, but this correction was made on a specific turbine design and may not be used for all the cases.

1.4.2 Stall delay models

The stall is an aerodynamic phenomena happens when the flow in the upper surface of the airfoil separates from the blade, it results of that a reduction of the generated lift. This phenomenon happens when the airfoil reaches a critical AOA.

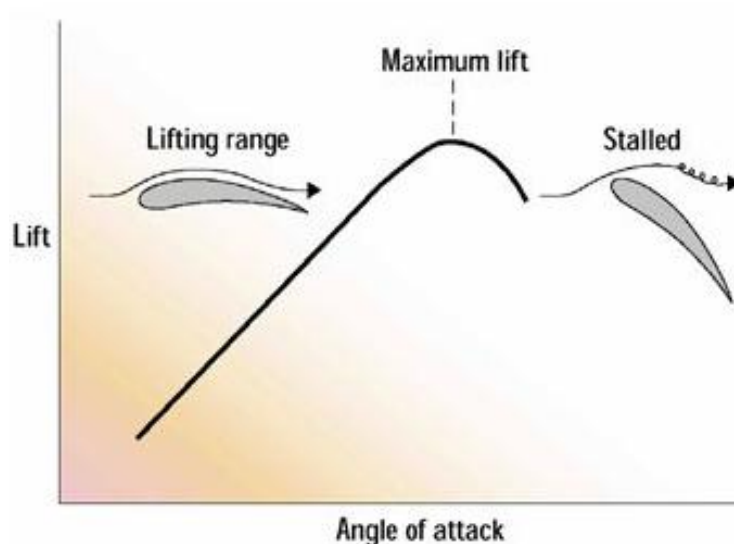


Figure 1-12: Stall delay diagram.[35]

To understand the stall phenomenon, we have to talk at first of the boundary layer in the blade. This approach consists on studying the effect of velocity in a thin region close to the blade surface. The stall phenomenon is directly linked to separation of the boundary layer from the blade surface.

A moving airflow adheres to the surface and relatively cannot move immediately at the surface. As a result, the distribution velocity normal to the surface starts from zero velocity in the surface and it increase with normal distance until it reaches free stream velocity as shown in the next figure.

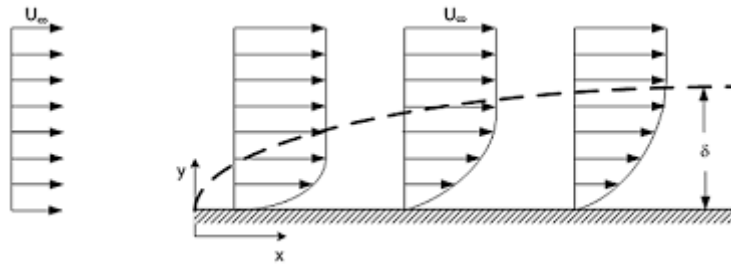


Figure 1-13: Velocity distribution on a boundary layer.[36]

The thickness of the boundary layer (denoted as δ) is the distance from the surface to the normal position that reaches 99% of the free stream velocity. It increases as the flow moves along the surface.

A drag force results from the effect of viscosity in the boundary layer; it is added to the total drag resulting from the pressure distribution.

The flow in the boundary layer on the blade surface if divided into two types. Laminar and turbulent flow; the laminar flow is located in a small distance travelled by the airflow for low number of Reynolds. After a certain distance, the flow becomes unstable and fluctuations on the motion of the flow start to appear, transiting from a laminar to a turbulent flow.

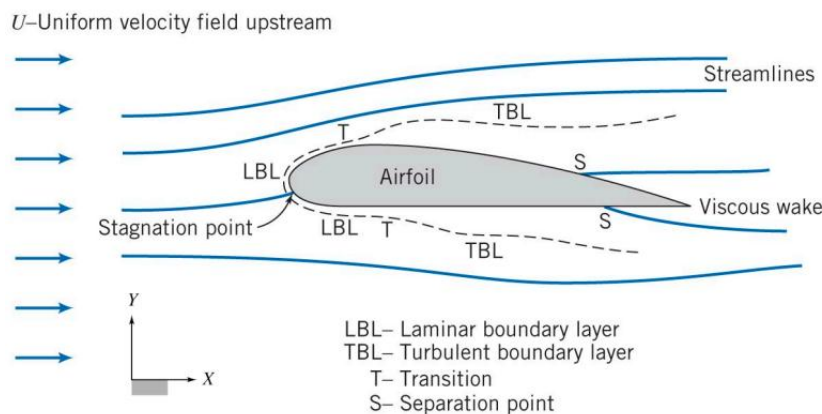


Figure 1-14: Boundary layer characteristics.[37]

The distribution of the pressure along the airfoil within the boundary layer is essential to predict the behavior of the flow for a given surface. This distribution is only function of the position along the surface, and it guides the airflow movement. A favorable gradient of the pressure $\frac{dp}{dx} < 0$ sucks the flow to the region, which has a low pressure – suction effect -, contrarily to the unfavorable gradient $\frac{dp}{dx} > 0$ that has an opposite effect that drives the airflow in the opposite of its direction.

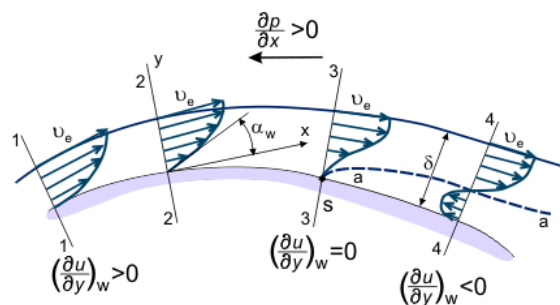


Figure 1-15: Pressure gradient in a boundary layer.[38]

Therefore, the stall phenomenon is the consequence of the separation of the flow from the surface in the boundary layer.

It has been noticed that rotation has a delaying effect on the separation of the flow observed on aircraft propeller blades by Himmelskamp. It has been also observed and studied on helicopter blades and later on wind turbine blades. It has been found that the stall delay is more present in the region close to the root of the blades.

Different correction models exist to take on consideration this effect when using the BEM method. The models of Snel et al [9], Chaviaropoulos and Hansen [10], and Raj [11] correct the C_l and C_d for 3D effects. The Du and Selig [12] model is based on the 3D integral boundary layer equations analysis on a rotating blade. Bak et al [13] proposed a model based on the analysis of the pressure distribution on a rotating and non-rotating blade.

1.4.3 Dynamic stall models

The dynamic loads from different sources are acting on the wind turbines; the air loads are unsteady because of the wind shear and turbulence. Yaw angle and upwind turbine wake make the inflow unsteady too. To achieve a 30 years lifetime of the wind turbine, we should take on consideration these loads in order to well design the blade, the hub and the support structure.

To regulate peak power and loads, the wind turbines operate in a steady stall. Buterfield et al discovered that during the normal operating conditions of the wind turbines the dynamic stall can exist on the blades.

The most of dynamic stall models were developed on helicopter aerodynamics. We cite as an example the Beddoes-Lieshmann model [14] and ONERA model [15]. Also there exists models for wind turbine application such as DTU Riso model [16], Larsen's model [17] and Snel's model [18].

- Beddoes-Lieshmann model

It was originally made for helicopter application, and then Gupta and Lieshmann adapted the model to make it usable in wind turbine applications and tested it on S809 airfoil. This model tries to simulate the physical mechanism commanding the dynamic stall phenomenon.

- Snel's model

This model takes into account the periodic vortex shedding effect in deep stall regime. It consists of two parts, a linear part and non-linear part modeled by a first order and second order differential equation respectively.

- ONERA model

This model was developed by Tran and Petot in 1980, based on simplifying assumption of linearization. It is very useful because it can be coupled with structural equations to make a complete dynamic analysis. It was developed by Petot to give dynamic lift, drag, and moment as well.

A part of the problem of the prediction of the dynamic loads is due to the lack of the knowledge of the rotary wing stall performance.

1.5 Aerodynamics of HAWTs airfoil

The airfoil of HAWTs has a very important rule in the performance of the wind turbine and the amount of the generated power. In this context, we will see the different characteristics, the classification and the performance of the airfoil.

1.5.1 Geometric characteristics

Let us look to a section profile of an airfoil; it has specific characteristics that should be presented to understand more its aerodynamics. The airfoil is defined by chord, thickness and the coordinates of its upper and lower surfaces. A mean line that separates the surface into two

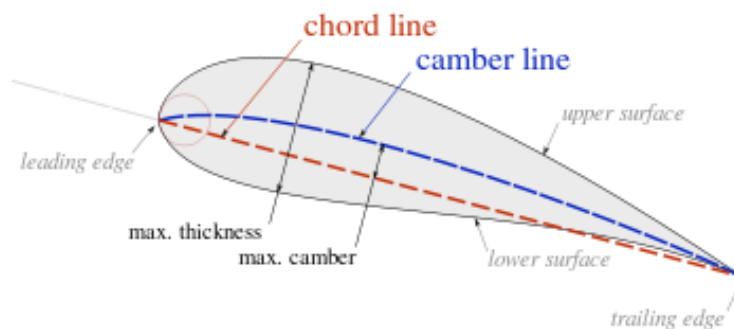


Figure 1-16: Airfoil characteristics.

equal parts is called the camber line. The leading edge is situated in the front and the trailing edge is situated in the back of the airfoil. The straight line from the leading edge to the trailing edge is called the chord line.

The lift and the drag are two type of forces that acts on the airfoil when the air flows around it. Their direction depends on the direction of the flowing air. The design of an airfoil has an objective to rise the lift and decrease the drag force in order to gain more power in lower wind speed.

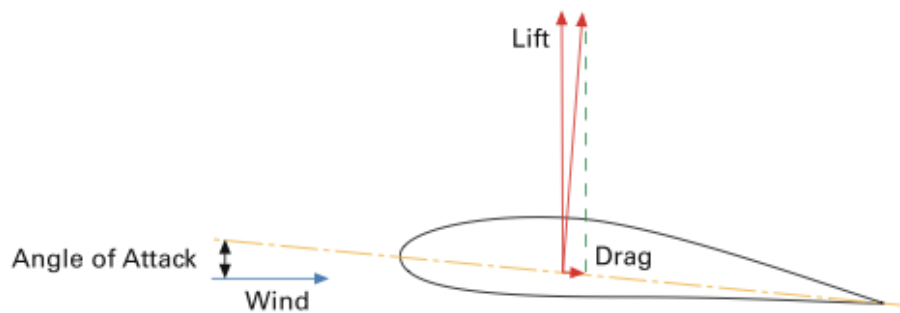


Figure 1-17: Lift and drag Forces on airfoil.

The design of the airfoil is basically done with two geometric rules:

- a) By the variation of the chord length and the thickness.
- b) By the variation of the mean camber line.

The incoming airflow can be divided into two composites, a relative and a normal airflow. The velocity triangle of an airfoil is presented in the next figure.

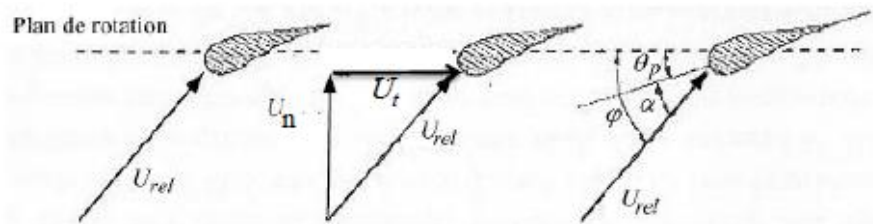


Figure 10-18: Velocity triangle on airfoil.[34]

The next relations are obtained from the velocity triangle:

$$U_n = U(1 - a) \quad (1.5)$$

$$U_t = (1 + a')\Omega r \quad (1.6)$$

$$\varphi = \arctan\left(\frac{U_n}{U_t}\right) \quad (1.7)$$

$$\varphi = \alpha + \theta_p \quad (1.8)$$

1.5.2 Classification of profiles

There are three classes of airfoils. The high lift, general purpose and high speed.

The high lift airfoils are used on aircrafts and sailplanes, they have a rounded leading edge and a high thickness chord ratio. The general-purpose airfoils have a sharp leading edge and a low thickness chord ratio. The high-speed airfoils have no camber and a lower thickness chord ratio; they are used in high-speed aircraft.

These airfoil profiles are divided into two categories, symmetrical and non-symmetrical.

The symmetrical airfoils have the same upper and lower surface; the mean camber line is upon the chord line. They produce no lift when the AOA is equal to zero.

The nonsymmetrical airfoils have different upper surface compared to the lower one with a curvature above the chord line. They produce a lift when the AOA is equal to zero.

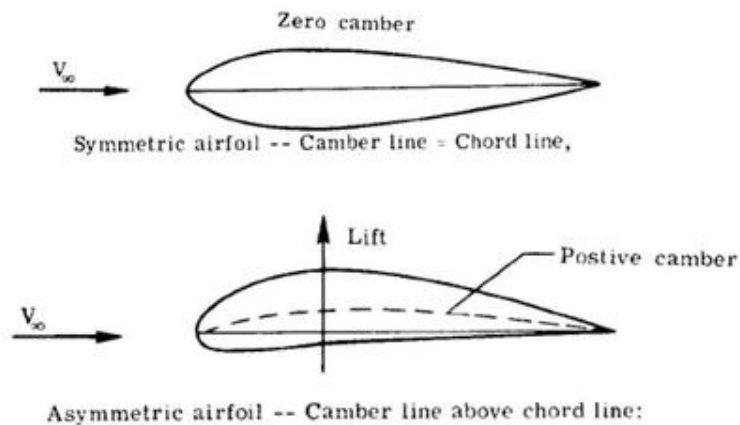


Figure 1-19: Symmetrical and non-symmetrical airfoils [39].

- NACA profiles

The National advisory committee for aeronautics (NACA) known now as ‘national aeronautics and space administration’ (NASA) designed different types of airfoil profiles. NACA 4 digits, NACA 5 digits and NACA 6 digits.

a) NACA 4 digits series:

They were designed based on the Clark-Y efficient profiles. We cite as an example the NACA 006, NACA 0008 ...etc.

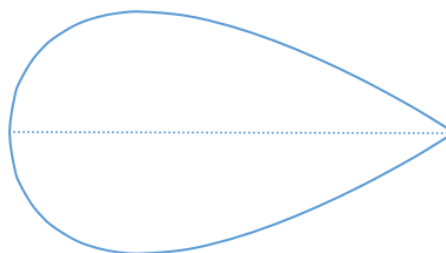


Figure 1-20: 4 digits NACA airfoil.

NACA XYZ₁Z₂: X refers to maximum camber in percentage of the chord.

Y refers to the position of maximum camber in tenth of the chord.

Z₁Z₂ refers to thickness in percentage of the chord.

b) NACA 5 digits series:

They have the same thickness variation over the chord of the NACA 4 digits series, but the camber line is curved to the top in order change the maximum camber position forward to rise the lift coefficient. We cite as an example the NACA 23015.

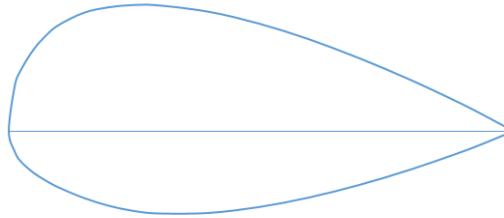


Figure 1-21: 5 digits NACA airfoil.

NACA $XY_1Y_2Z_1Z_2$: X refers to $20/3$ times the lift coefficient design.

Y_1Y_2 refers to two times the position of maximum camber in percentage of the chord.

Z_1Z_2 refers to thickness in percentage of the chord.

c) NACA 6 digits series:

They were developed to get a minimum of drag and to maintain a laminar flow over a portion of the profile. We cite as an example the NACA 63-210.

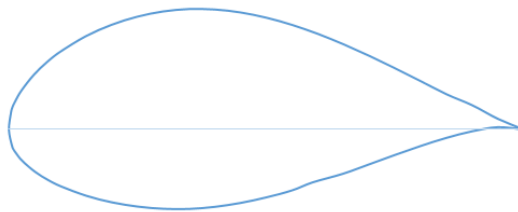


Figure 1-22: 6 digits NACA airfoil.

NACA $6XYZ_1Z_2$: 6 refers to series number 6.

X refers to position of the minimum pressure position to the tenth chord ratio.

Y refers to the tenth of the lift coefficient design.

Z_1Z_2 refers to thickness in the percentage of the chord.

1.5.3 Airfoil performance

The pressure around the airfoil in presence of an airflow determines the aerodynamics of the airfoil. The loads acting on the airfoil are the lift, which is perpendicular to the relative velocity and the drag, which is parallel to relative velocity.

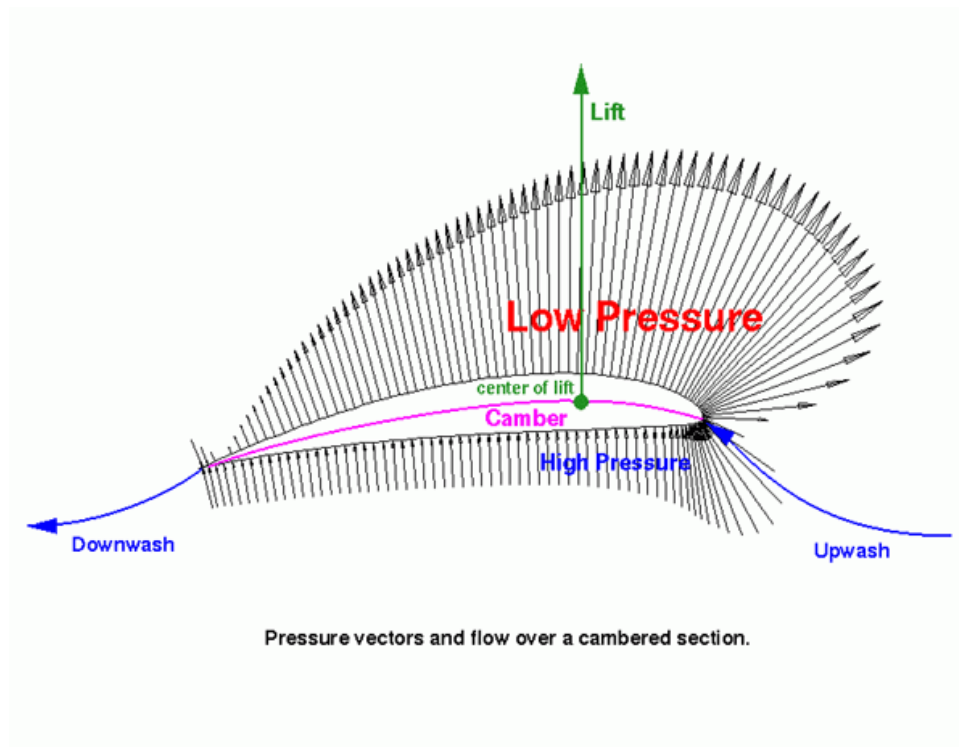


Figure 1-23: Pressure distribution on airfoil.[40]

When the air flows around the airfoil, it is divided into two parts, the upper part and the lower part. Very closely to the airfoil, the small quantity of air tends to move from the front to the back of the airfoil by maintaining the same time of travel. Which mean an increase of the velocity of the airflow of the upper part since that the distance is longer than the one of the lower part. This increasing of velocity leads to a decrease of pressure, which creates a pressure potential that allows the airfoil to move up.

The aerodynamic loads depend on the shape of the airfoil, the AOA, the Mach number and Reynolds number, any variation of these parameters leads to a change of the aerodynamic behavior of the airfoil.

The performances of an airfoil are measured by the lift coefficient and the drag coefficient. The more the lift coefficient is high and the drag coefficient is low the better is the profile of the airfoil in the wind-turbine use. For example, when we change the airfoil characteristics as the trailing edge thickness to rise the lift coefficient, the drag coefficient tends to rise too; so to have a good overview of the airfoil performance, we take both of these coefficients in consideration, by the creation of a ratio parameter, lift coefficient over drag coefficient $\frac{C_l}{C_d}$.

We compare in this case the variation of the $\frac{C_l}{C_d}$ parameter over AOA. This will help us in the optimization of the airfoil performance.

For example, let us see the comparison of the C_l and C_d of a fine plate that has different camber as shown in the next diagram.

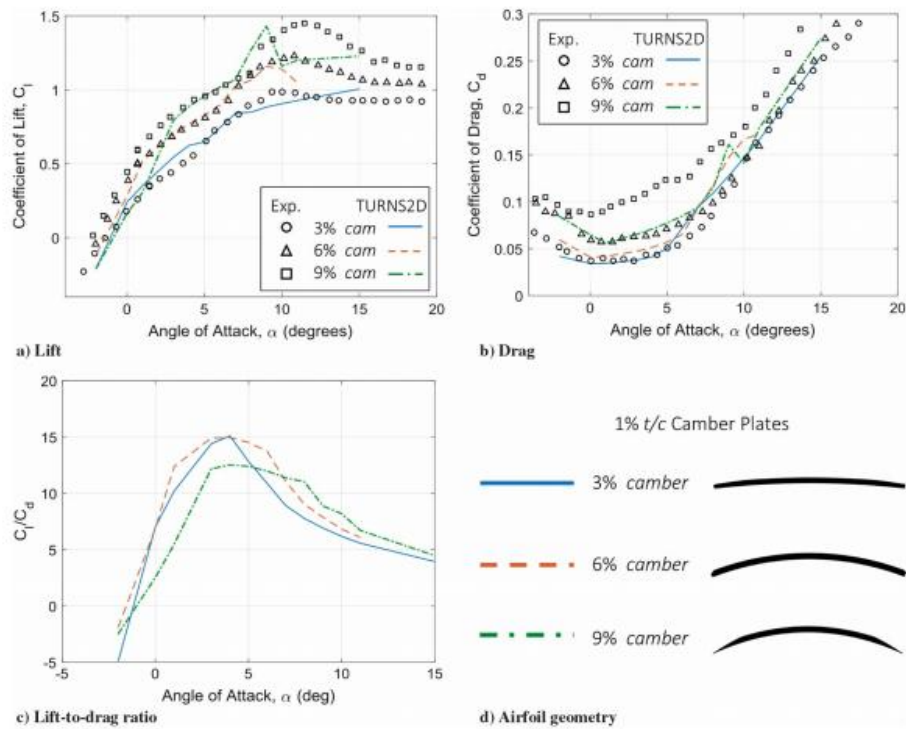


Figure 1-24: lift and drag coefficient of different camber line configurations [19].

The diagrams shows that the thin plate that has a 9% camber has the highest C_l then comes the 6% camber than 3% camber, even for the drag coefficient the three thin plate have the same order, so in this case we should look to the $\frac{C_l}{C_d}$ parameter to determine which thin plate is better in performance. From -2° to 6.5° of AOA, the 6% camber thin plate is the most performance between these three thin plates, then from 6.5° to 15° the 9% camber is performing better than the others.

1.6 Literature review

1.6.1 Methods for Determining Lift and Drag

There are few ways to determinate the lift and drag, theoretical and experimental ones.

These methods are:

A) CFD

The computational fluid dynamics simulations allow us to determine the lift and the drag coefficient by simulating a real case of the airflow and its conditions in the presence of the wind turbine model. This method demands a lot of technic and experience because the CFD is a huge field that can solve a lot of fluid problems.

The most known and used software is the ANSYS Fluent developed in 2006.

B) Xfoil

XFOIL is an interactive software used to analyses and design the airfoils in a subsonic state. It has many functions such as:

- Viscous analysis of an existing airfoil.

- Airfoil design and redesign by interactive modification of the surface speed distribution.
- Airfoil redesign by interactive modification of geometric parameters.
- Blending of airfoils.
- Writing and reading of airfoil coordinates and polar save file.
- Plotting of geometry, pressure distribution and multiple polars.

It has been written in FORTRAN by Mark Drela at MIT in 1980's and released in under GNU GPL. – free software license that guarantee to the end user the freedom of run, study, share and modify the software. –

C) JavaFoil

JavaFoil is a simple program that uses several traditional methods for airfoil analysis, the potential flow analysis and the boundary layer analysis. It has many functions such as:

- Creating the geometry
- Modify the essential element of the geometry
- Design and plot the pressure coefficient of the airfoil
- Plot the velocity distribution
- Plot the flow field
- Plot the boundary layers parameters
- Calculate the lift and the drag
- Analyze the lift and drag coefficient
- And other options

This program is written by Martin Hepperle in java.

D) Wind tunnel

We can determine the lift and the drag by using the wind tunnel. A printed model of the airfoil is required. Since that the external condition of the test are known such as wind speed, wind density, the pressure and the geometric characteristic of the airfoil, also the lift and the drag generated using mechanic tools, we can determine C_l and C_d easily.

1.6.2 Previous works on BEM models

A lot of researches and works were done on the BEM model, in order to improve its performance and make it more efficient. By adding some coefficients and formulations that corrects some physical phenomenon. Or by integrating some algorithms to the method to make it more powerful. In the next paragraphs, we will cite some of the works done on BEM method.

The blade element momentum theory introduced by H. Glauert in 1926, is a framework to model the aerodynamic interaction between a turbine and a fluid flow, it is used to estimate the efficiency of the turbine or to help in its design. This method is a good method to make the calculations but a lack of some physical interpretation limits its efficiency in the field of wind turbines.

The BEM is improved to quantify the miscalculated power at stalled flow regime, by correcting the aerodynamic characteristics taking into account the 3D flow physics such as stall delay and rotational effect [12].

Correcting the tip loss, skewed wake and rotational effect extends the use of this method to turbulent wake regime for the horizontal axis wind turbine [21].

A modeling approach based on the blade element momentum theory is coupled to the computational fluid dynamics to make a model to predict the performance of tidal stream in oceans environment [22].

The integration of new methods to the BEM method such as artificial intelligence to deal with the different sources of convergence issues using neural network [23].

An aerodynamic design and optimization program for wind turbine is developed by integrating the BEM method and genetic algorithm [24].

1.6.3 Available experiments and data

The airfoils data that we used in our study are available in the airfoiltools website [20], (i.e. the geometry, C_l and C_d). This data were used to train the ANN model. And for the BEM-ANN validation, we used the NREL phase IV with a NACA 0012 blade data.

Chapter 2: Artificial neural networks

Chapter 2: Artificial neural network

2.1 Introduction

The definition of machine learning in the literature is defined by Tom Mitchell (1998):

A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.

Machine learning is a data science field which and a technology of artificial intelligence. It allows to computers to learn without being explicitly programmed. To learn and to develop good model, computers need to analyze and to train with that data.

Machine learning is a modern science, which allows discovering patterns and making predictions based on statistics and mining data. First algorithms were created in the end of 1950, the most known is perception.

The basic analytics tools are not efficient to operate on big data, the large amount of data is so good for comprehensible analysis, making correlations that includes the maximum hypothesis, in a precise and fast way. For example, machine learning is able to detect a fraud transaction in milliseconds based on the transaction data (the amount, the localization, the stores...) and the history of transaction done by the person.

Therefore, the big data and machine learning are working pretty well with each other's to get a precise correlations and predictions.

The machine learning has wide applications like image recognition, automatic translation, self-driving cars, medical diagnostics, detection of malwares and frauds, intelligent robots and so on ...

Machine learning is used in many fields such as banks and industrial companies to prevent frauds, to identify opportunities... etc. health care sector to help doctors to analyze data of radiology, improve diagnostics and treatments. Marketing to analyze previous sales in order to determine other products that might interest the client. Transportation to determine the fast and the efficient itinerary for delivery people and public transportation.

Machine learning is a sub-part of Artificial intelligence

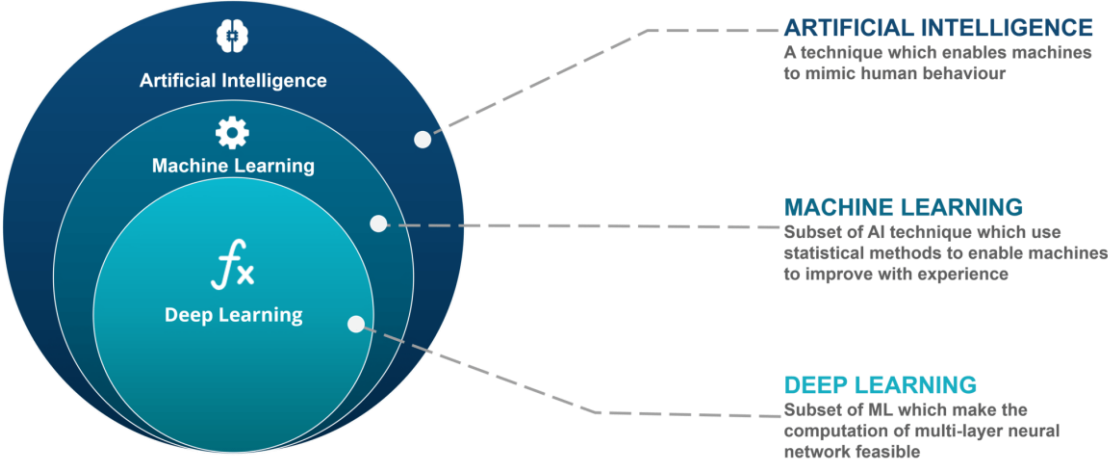


Figure 2-1: Artificial intelligence domain.

2.2 Learning methods

The learning methods of a machine learning depends on the problem that we are trying to solve and its model. Supervised learning, unsupervised learning and reinforcement learning are some of these methods, in the next paragraph we are going to talk about supervised and unsupervised method.

2.2.1 Supervised training method

This method is based on the approach of giving the computer labeled input data and the expected output results. During the training on data, the computer will search for correlations and patterns to figure out the relation between the inputs and the outputs in order to get good results when it comes to new data that it did not train on it, i.e. learning by examples. So, the objective of it, is to predict the correct output of a new input data.

The computer is fed with a huge amount of data, and it is told to obtain a specific output value for each input. The correlations that we get after the training is called “trained model». This trained model is tested with a test data set to verify its accuracy.

It can be expressed with a simple mathematic relationship:

$$Y = f(x)$$

Where Y is the output predicted by the trained model by a pre-made function using the input value x .

Supervised learning performs very well at regression and classification problems, such as predicting the price of a house based on its area, number of rooms, number of floors and location...etc. determining the category of a news article, classification of e-mail as spam or non-spam) ...etc.

A) regression

It is a predictive process where the designed model search for relationship between dependent and independent features and it has continuous output values.

A simple regression model with one feature can be expressed by this formula:

$$Y = wx + b$$

w is the slope, x is the feature and b is the y intercept.

For this model, a line that fits the most of the data presents a good model. For a model with two features, the plane that fits the most of data presents a good model.

In a general way, the formula that represent this type is:

$$y = w_0 * x_0 + w_1 * x_1 + w_2 * x_2 + \dots + w_n * x_n + b$$

w_i is called weight, x are the features and b is called the bias.

The computer initiates the weights and the bias randomly.

There are many regression algorithms, the most common are:

- linear regression
- logistic regression
- polynomial regression

Let us take an example of the salary according to the experience of the worker. A data of salary of 30 workers is fed to the computer according to years of their experience. After the training of the model, we got this result. A line that fits the most of the data, which can give us a good estimation of the salary of a new worker given his years of experiences.

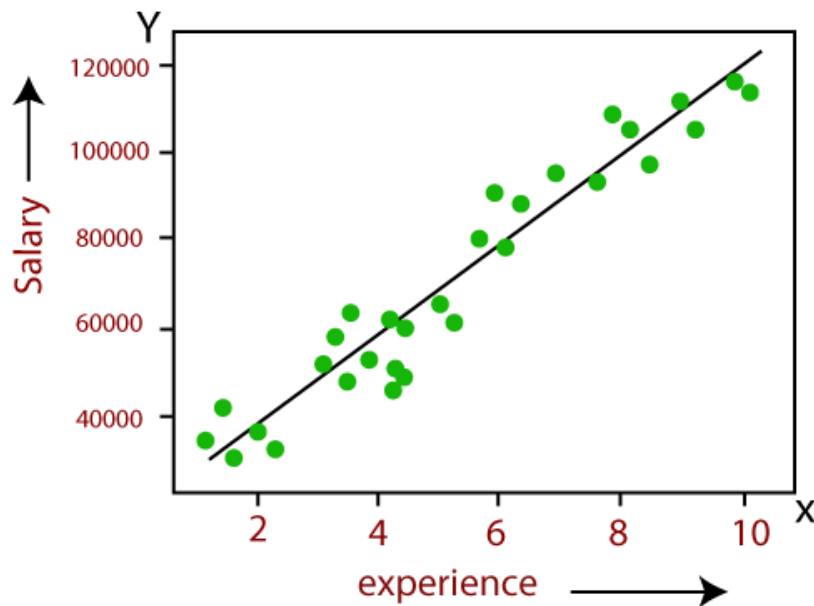


Figure 2-2: salary vs. experience plot.

B) classification

In this process, the computer is given data with assigned categories. The role of this classification algorithm is to assign the input value to an output category or class. The result of this model are discrete values.

The algorithm will figure out how to distinguish between classes and how to accord every input to his appropriate class using the mapping function:

$$Y = f(x)$$

We can use many classification algorithms. The choice of which to use depends on the problem that we are trying to solve. Here are some of them:

- Linear classification
- Support vector machine (SVM)
- Decision trees
- K-nearest Neighbor
- Random Forest

For example, the classification of an email whether is spam or not. Two classes are assigned “spam” and “non-spam” i.e. binary classification. According to number of mistaken words, number of words repeated in the mail, the number of external links ...etc.

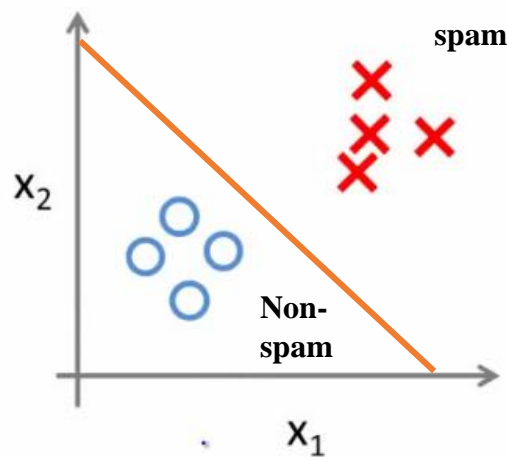


Figure 2-3: E-mail classification.

2.2.2 Unsupervised training method

It is a technique that do not need to be supervised, it is meant to work on its own and discover the patterns and the relationships that are hidden and undiscovered without guidance because it deals with data that is neither classified nor labeled.

The common unsupervised method is clustering, it finds hidden patterns or grouping in data according to similarities and differences without a prior training. It is used generally for in data analysis.

The unsupervised training method is divided into three categories:

- Clustering: discover the inherent grouping in a dataset
- Association: discover the rules that describe the portions of the dataset
- Anomaly detection: it detects the unusual values in the dataset

The pattern discovered with the unsupervised training method can be very useful when we tend to implement a supervised training method later. It helps to pre-organize the data and make pre-labels.

The most common clustering algorithms are:

- Hierarchical clustering
- K-means clustering
- Self-organizing map
- Gaussian mixture models
- Hidden Markov models

2.3 Artificial neural network methods

This method is inspired from the way the human brain analyzes and processes informations. It is made of a complex neuron network around 100 billion of neurons which are connected between them. They serve as transmitters of the information inside the brain. They receive signals from outside of the body and they transmit it to the appropriate place in the brain.

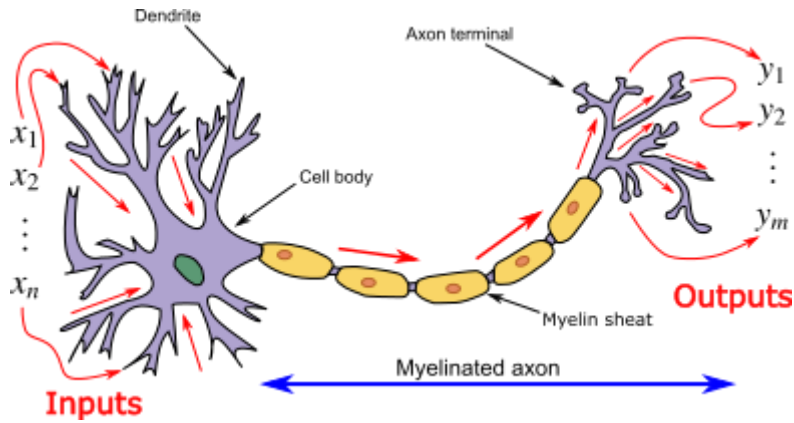


Figure 2-4: Human brain neuron.

The inputs are received in the Dendrite and transmitted through the Myelinated axon where we find the output in the axon terminal.

The same for the ANN, it has hundreds of processing units, input and output ones interconnected by nodes. The input units receive the information (the values) and based on an internal weighting system, it learns to produce the output desired.

The next diagram presents the general model of the artificial neural network (ANN) inspired from a biological neuron.

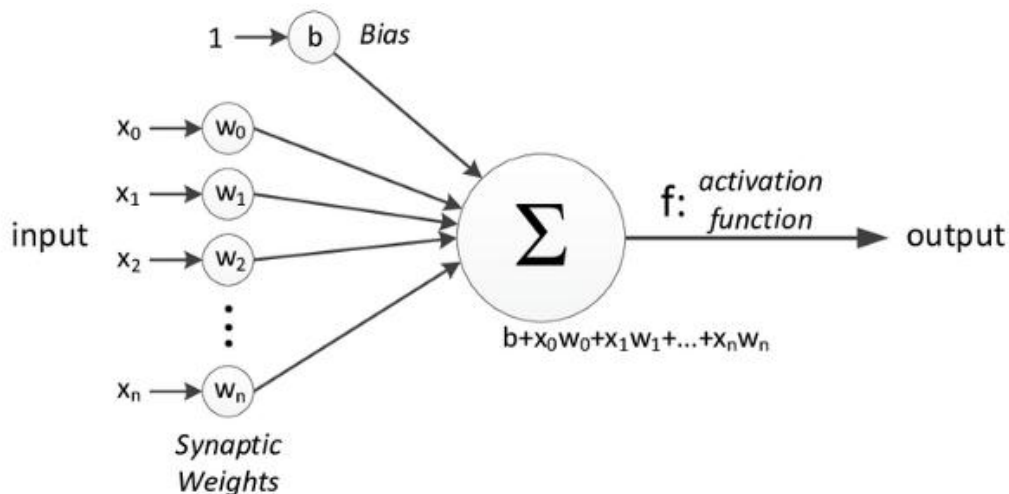


Figure 2-5: one neuron representation.

This diagram represents a single layer with one neuron. The x_i are the features, w_i are the weights and b is the bias.

Each feature x_i is multiplied to its connection weight, then summed with the bias and fed through the activation function (transfer function) to get at the end the output value.

2.4 ANN architecture

Like the human brain, the ANN has many important components, which are essential in the calculations such as hidden layers, hidden neurons and activation function or transfer function...etc.

2.4.1 Hidden neurons

As mentioned before, the neurons are the processing units that store the result of the previous layer calculated by $\phi(w_0^k * x_0 + w_1^k * x_1 + w_2^k * x_2 + \dots + w_n^k * x_n + b^k)$.

A simple neuron with a bias can be represented as follows:

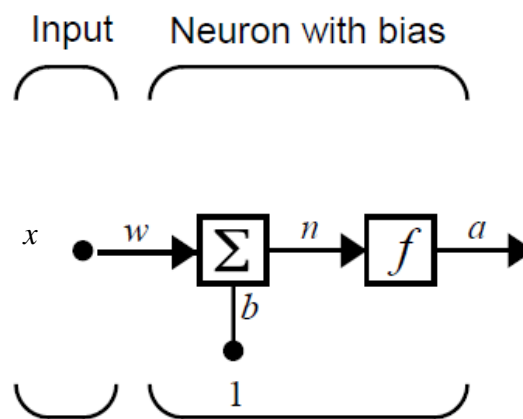


Figure 2-6: Single neuron representation.

The result is $a = f(w * x + b)$.

The number x is multiplied with w and summed with b and fed into the transfer function f which at the end gives us the output a .

2.4.2 Hidden layers

The word hidden means that they are not visible. The hidden layer contains amount of neurons that are separated between them and related to the next layer.

In general, one hidden layer is usually enough to solve the problem. The more the problem is complicated the more layers are added.

Therefore, the composition of the neural network is one input layer, followed by hidden layers and the output layer at the end. As the next diagram demonstrates.

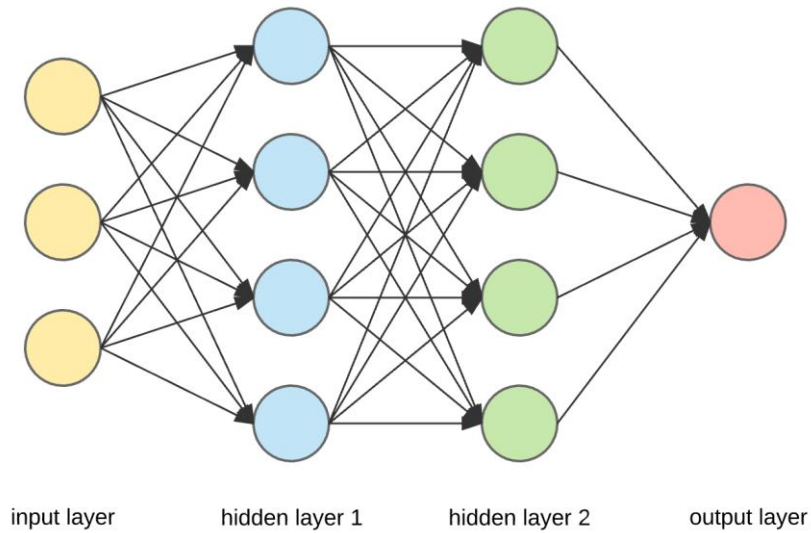


Figure 2-7: Multiple hidden layer representation.

The mathematical equation of one hidden layer is: $a = f(w * x + b)$

Where a, x, b are vectors and w is a matrix.

In another representation:

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ \vdots \\ a_n \end{pmatrix} = f \left\{ \begin{pmatrix} w_1 & w_2 & w_3 & w_4 & \dots & w_n \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_n \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ \vdots \\ b_n \end{pmatrix} \right\}$$

The schema of one hidden layer is given in the next diagram.

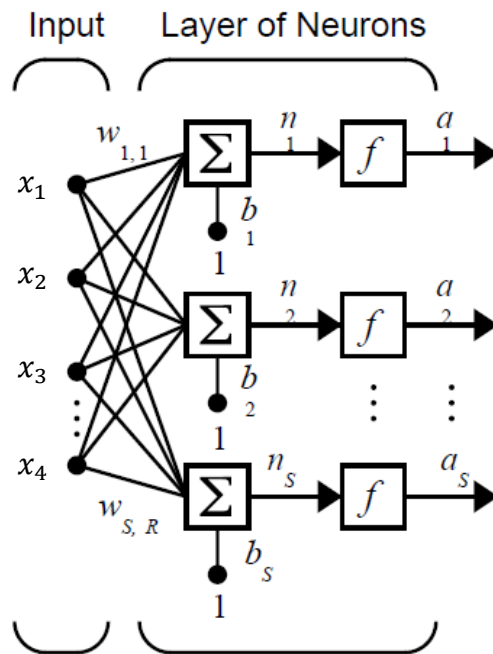


Figure 2-8: one hidden layer representation.

In a general case of k hidden layer, the mathematic equation is: $a^k = f(w^{(k)} * x + b^k)$

Where a, x, b are vectors and w is a matrix.

In another representation:

$$\begin{pmatrix} a_1^k \\ a_2^k \\ a_3^k \\ a_4^k \\ \vdots \\ a_n^k \end{pmatrix} = f \left\{ \begin{pmatrix} w_1^k & w_2^k & w_3^k & w_4^k & \dots & w_n^k \end{pmatrix} \begin{pmatrix} x_1^k \\ x_2^k \\ x_3^k \\ x_4^k \\ \vdots \\ x_n^k \end{pmatrix} + \begin{pmatrix} b_1^k \\ b_2^k \\ b_3^k \\ b_4^k \\ \vdots \\ b_n^k \end{pmatrix} \right\}$$

Where the index k represents the number of hidden layer and n the number of features.

The schema of three hidden layer is given in the next diagram.

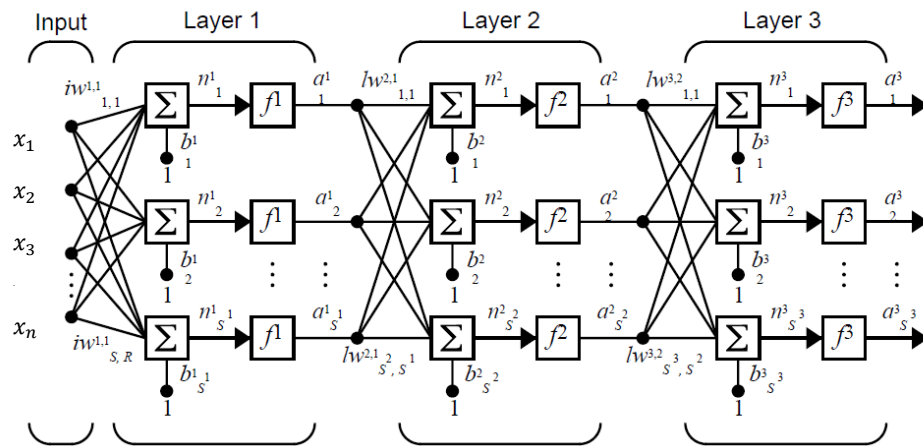


Figure 2-9: ANN representation.

2.4.3 Activation function

Called also transfer function, it is an essential part of the ANN. It makes sense of the complicated data. It relates the input of the layer into the output of the layer, which will be the input of the next layer.

It decides whether the neuron will be activated or not according to the sum of the product of weights by features to which we add the bias.

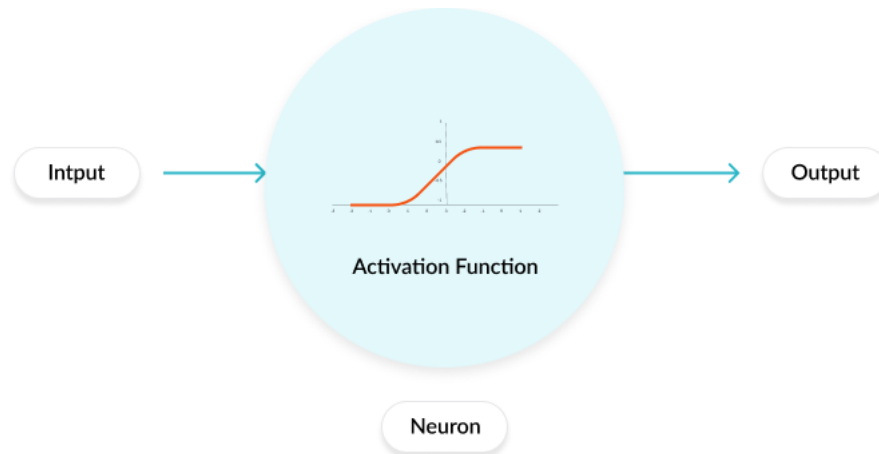


Figure 2-10: Activation function diagram.

The activation functions allow us to make non-linear models that maps the data. We should use the activation function to get significant results and make the model learn from a complicated data like images, audios and videos...etc.

There are many types of activation functions, such as

A) Threshold activation function (binary step function):

It returns the value of one if the input value is above the threshold, and the value of zero otherwise.

$$\text{For example, } f(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \quad (2.1)$$

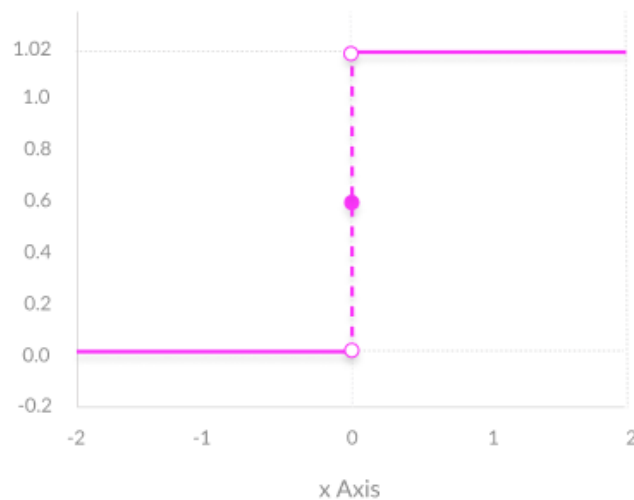


Figure 2-11: Threshold function graph.

This function is used when we have just two classes or two categories, basically a true or false statement.

B) Linear activation function

It multiplies the input by a weight in each neuron, and return a proportional output to the input. It is given by the formula:

$$f(x) = c * x \tag{2.2}$$

Where c is the weight and x is the feature.

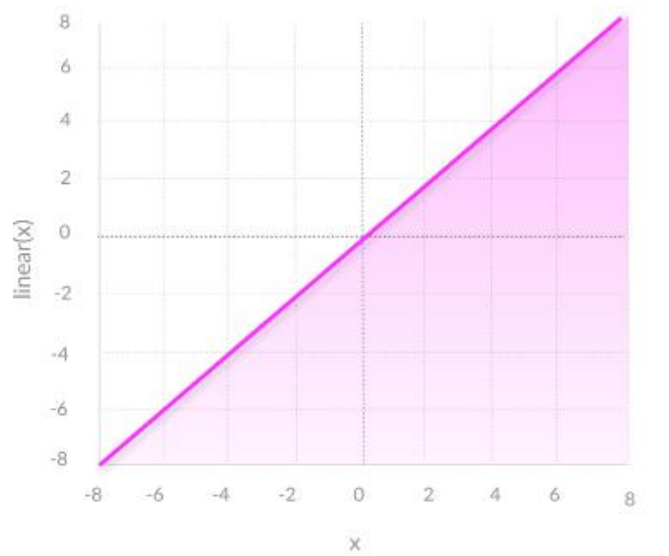


Figure 2-12: Linear function graph.

C) Sigmoid / logistic activation function

It returns a continuous value situated between zero and one. It is used in probabilities because of the output interval.

It is given by the formula:

$$f(x) = \frac{1}{1 + e^{-z}} \tag{2.3}$$

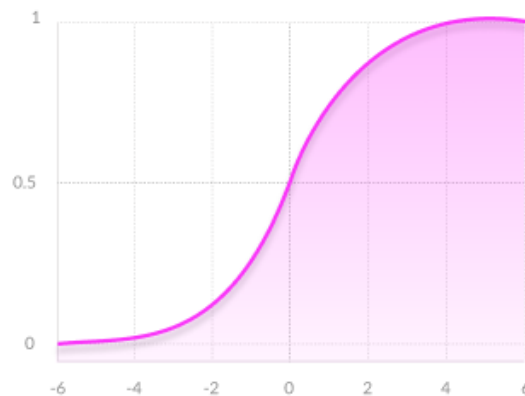


Figure 2-13: Sigmoid function graph

D) Hyperbolic tangent activation function (tanh)

It has the same principal with the sigmoid function; they are similar in the interval of output e.g. zero to one. However, it is better in performance; it maps easily strong negative or positive inputs.

It is given by the formula:

$$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (2.4)$$

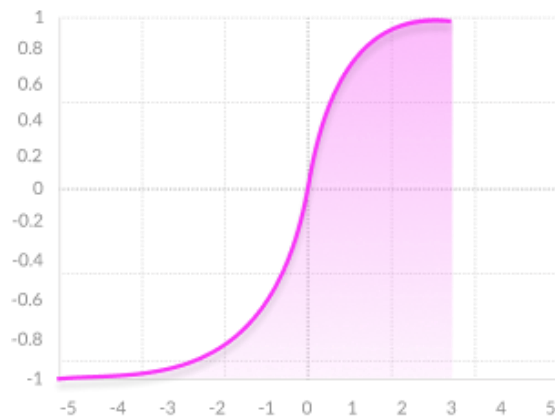


Figure 2-14: Tanh function graph.

E) Rectified linear unit (ReLU)

It returns the input value when it is positive and zero when it is negative. It is the most used activation function; it ranges the output from zero to infinity.

It is given by the formula:

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{otherwise} \end{cases} \quad (2.5)$$

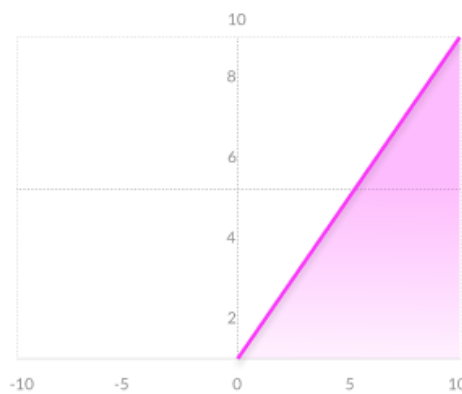


Figure 2-15: ReLU function graph.

Which activation function that should we use?

It depends on the problem that we are trying to solve; knowing the characteristics of all these functions helps us to use the right one on our calculations. Note that the choice of the activation function can affect the precision and the speed of calculations, so a good choice means good results.

2.4.4 Training functions

The ANN has several training functions that can be used; the main difference between them is the way of calculation.

In the next table, we will cite the different training function and their main characteristics.

Training function	Full function name	Characteristics
Trainlm	Levenberg-Marquardt backpropagation	<p>trainlm is a network training function that updates weight and bias values according to Levenberg-Marquardt optimization. trainlm is often the fastest backpropagation algorithm in the toolbox, and is highly recommended as a first-choice supervised algorithm, although it does require more memory than other algorithms.</p> <p>This function uses the Jacobian for calculations, which assumes that performance is a mean or sum of squared errors. Therefore, networks trained with this function must use either the mse or sse performance function.</p>
Trainbfg	BFGS quasi-Newton backpropagation	<p>trainbfg is a network training function that updates weight and bias values according to the BFGS quasi-Newton method.</p>
trainbr	Bayesian regularization backpropagation	<p>trainbr is a network training function that updates the weight and bias values according to Levenberg-Marquardt optimization. It minimizes a combination of squared errors and weights, and then determines the correct combination so as to produce a network that generalizes well. The process is called Bayesian regularization.</p> <p>This function uses the Jacobian for calculations, which assumes that performance is a mean or sum of squared errors. Therefore, networks trained with this function must use either the mse or sse performance function.</p>

traincgb	Conjugate gradient backpropagation with Powell-Beale restarts	traincgb is a network training function that updates weight and bias values according to the conjugate gradient backpropagation with Powell-Beale restarts.
Traincgf	Conjugate gradient backpropagation with Fletcher-Reeves updates	traincgf is a network training function that updates weight and bias values according to conjugate gradient backpropagation with Fletcher-Reeves updates.
traincgp	Conjugate gradient backpropagation with Polak-Ribière updates	traincgp is a network training function that updates weight and bias values according to conjugate gradient backpropagation with Polak-Ribière updates.
traingd	Gradient descent backpropagation	traingd is a network training function that updates weight and bias values according to gradient descent.
traingdm	Gradient descent with momentum backpropagation	traingdm is a network training function that updates weight and bias values according to gradient descent with momentum.
traingda	Gradient descent with adaptive learning rate backpropagation	traingda is a network training function that updates weight and bias values according to gradient descent with adaptive learning rate.
traingdx	Gradient descent with momentum and adaptive learning rate backpropagation	traingdx is a network training function that updates weight and bias values according to gradient descent momentum and an adaptive learning rate.
trainoss	One-step secant backpropagation	trainoss is a network training function that updates weight and bias values according to the one-step secant method.
trainr	Random order incremental training with learning functions	trainr trains a network with weight and bias learning rules with incremental updates after each presentation of an input. Inputs are presented in random order.
trainrp	Resilient backpropagation	trainrp is a network training function that updates weight and bias values according to the resilient backpropagation algorithm (Rprop).
trainscg	Scaled conjugate gradient backpropagation	trainscg is a network training function that updates weight and bias values according to the scaled conjugate gradient method.

Table 1: Training functions characteristics [29].

2.4.5 Forward propagation

Forward propagation is the passage from the input layer to the output layer passing through the hidden layers. A mathematic operation calculates the next neuron value. Taking an example of three neurons, two neurons situated in the first layer and the third one is in the second layer, the value of the third layer is calculated by the formula that follows as demonstrated in the diagram:

$$C = f(w_1 * A + w_2 * B) \tag{2.6}$$

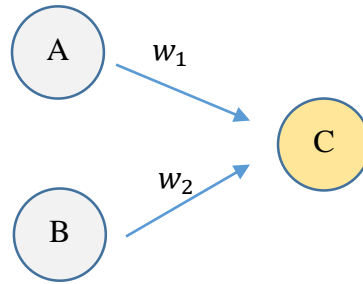


Figure 2-16: Two hidden layers forward propagation.

In general, it is represented as follows:

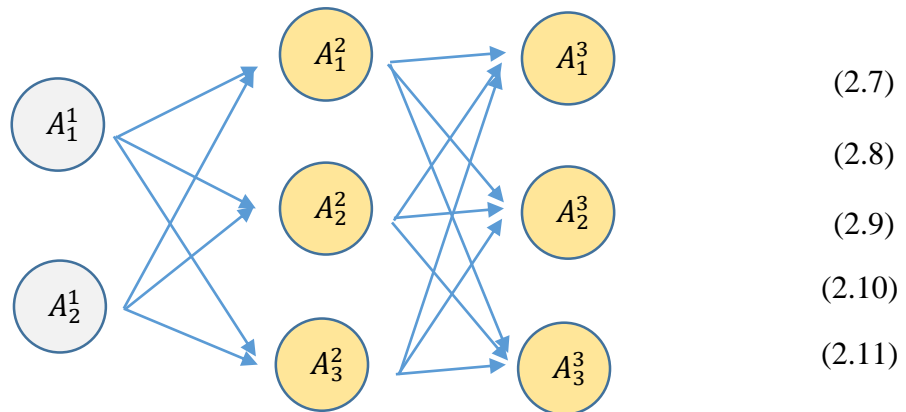


Figure 2-17: Three hidden layers forward propagation.

The mathematical operation of the feed forward is

$$A^{(1)} = X$$

$$Z^{(2)} = w^{(1)} * A^{(1)}$$

$$A^{(2)} = f(Z^{(2)})$$

$$Z^{(3)} = w^{(2)} * A^{(2)}$$

$$A^{(3)} = f(Z^{(3)})$$

2.4.6 Cost function

It is made to calculate a parameter that allows us to optimize the ANN learning, using backpropagation. The most used cost function is Mean Square Error (MSE). It is defined as:

$$C_{MSE} = \frac{1}{2} \sum_j (A_j^L - y_j)^2 \quad (2.12)$$

Where y_j is the desired output, A_j^L is the last output of the ANN and j is the number of features.

This function calculates the error between the desired output and the output of our ANN. It's a parameter that allows us to estimate the efficiency of the ANN.

2.4.7 Backpropagation

This algorithm aims to minimize the cost function by adjusting and updating the weights and bias of the network so that we get the desired output. However, it does not adjust the input layer because is the value initiated by the user. It calculates the cost function gradient; starting from the output layer and propagates backward to the input layer.

Gradient computation:

The gradient is computed by using a method called chain rule, let us take the example of the weight w_j^l

$$\frac{\partial C}{\partial w_{jk}^l} = \frac{\partial C}{\partial z_j^l} \frac{\partial z_j^l}{\partial w_{jk}^l} \quad (2.13)$$

And
$$z_j^k = \sum_{k=1}^m w_j^k a_k^{l-1} + b_j^k \quad (2.14)$$

So,
$$\frac{\partial z_j^l}{\partial w_{jk}^l} = a_k^{l-1} \quad (2.15)$$

It results from that:

$$\frac{\partial C}{\partial w_{jk}^l} = \frac{\partial C}{\partial z_j^l} a_k^{l-1} \quad (2.16)$$

We will do the same operation with the bias b_j^l

$$\frac{\partial C}{\partial b_j^l} = \frac{\partial C}{\partial z_j^l} \frac{\partial z_j^l}{\partial b_j^l} \quad (2.17)$$

And
$$z_j^k = \sum_{k=1}^m w_j^k a_k^{l-1} + b_j^k \quad (2.18)$$

So,
$$\frac{\partial z_j^l}{\partial b_j^l} = 1 \quad (2.19)$$

It results from that:

$$\frac{\partial C}{\partial b_j^l} = \frac{\partial C}{\partial z_j^l} \quad (2.20)$$

The partial derivative $\frac{\partial C}{\partial z_j^l}$ is called local gradient δ_j^l .

Therefore, to optimize the ANN we update the value of the weights and bias of all the ANN one by one, to obtain the desired result, using this formulation:

$$w := w - \epsilon \frac{\partial C}{\partial w} \quad (2.21)$$

$$b := b - \epsilon \frac{\partial C}{\partial b} \quad (2.22)$$

Where ϵ is the learning rate

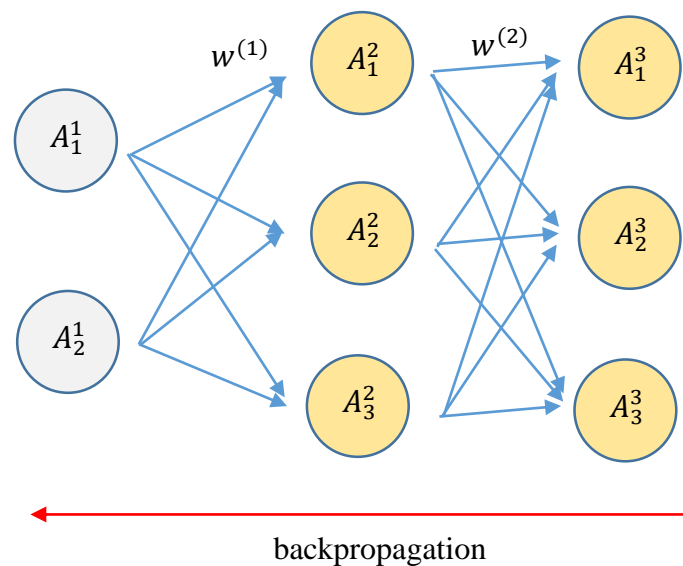


Figure 2-18: Back propagation representation.

2.4.8 Cross validation

It is a technique that measures the effectiveness of the machine-learning model. This technique splits randomly the data into two portions, train and test one.

When the ANN is trained with the train data, the test data helps to evaluate the accuracy of its results.



Figure 2-19: Data splitting.

2.5 ANN algorithms

In this project, we will use the MATLAB software to create our ANN. MATLAB offers a good tool to work on ANN called NNTOOL – Neural network tool -.

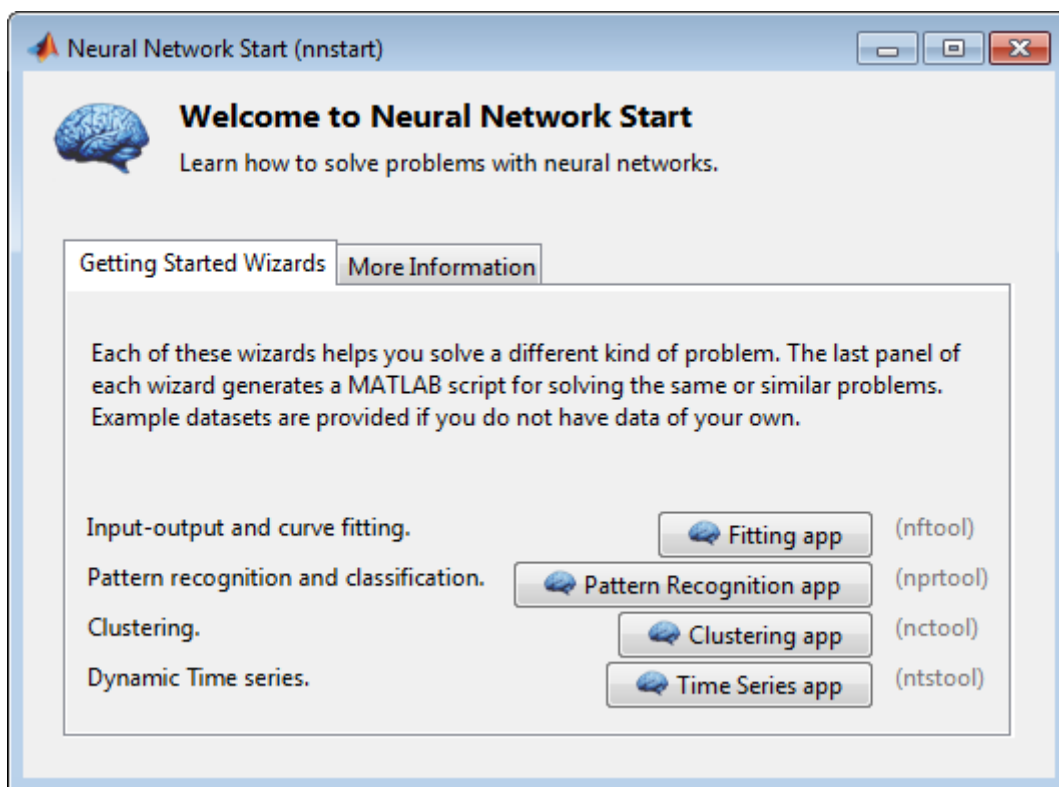


Figure 2-20: Neural network start window [30].

This is a welcoming window of NNTOOL, it allows us to work on different types of machine learning categories e.g. times series, clustering, fitting and ANN.

To launch ANN program, we should write NNTOOL in the command bar and we will get this window:

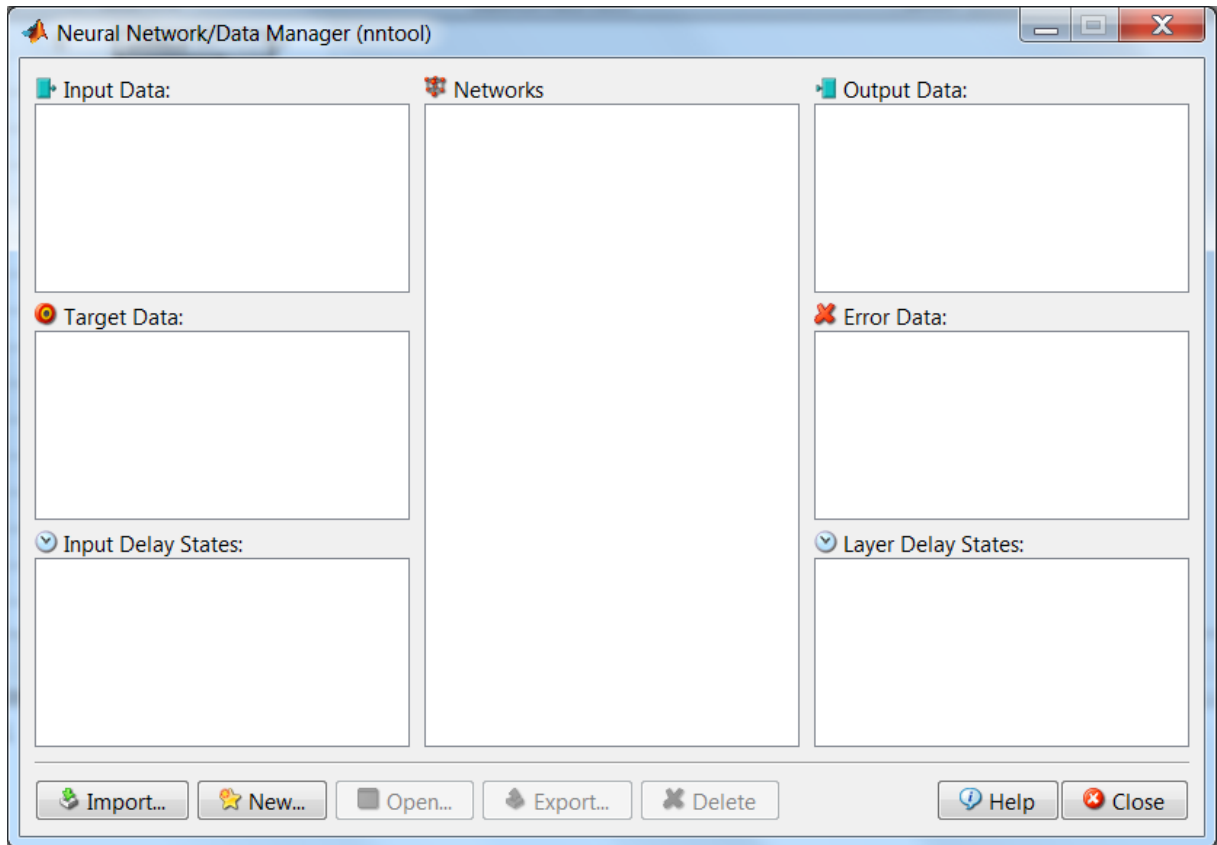


Figure 2-21: Neural network data manager [30].

2.5.1 Initialization

First, we should start with the data arranging the data. Make it in a vector or a matrix and separate the input ones from the outputs. Then importing the data.

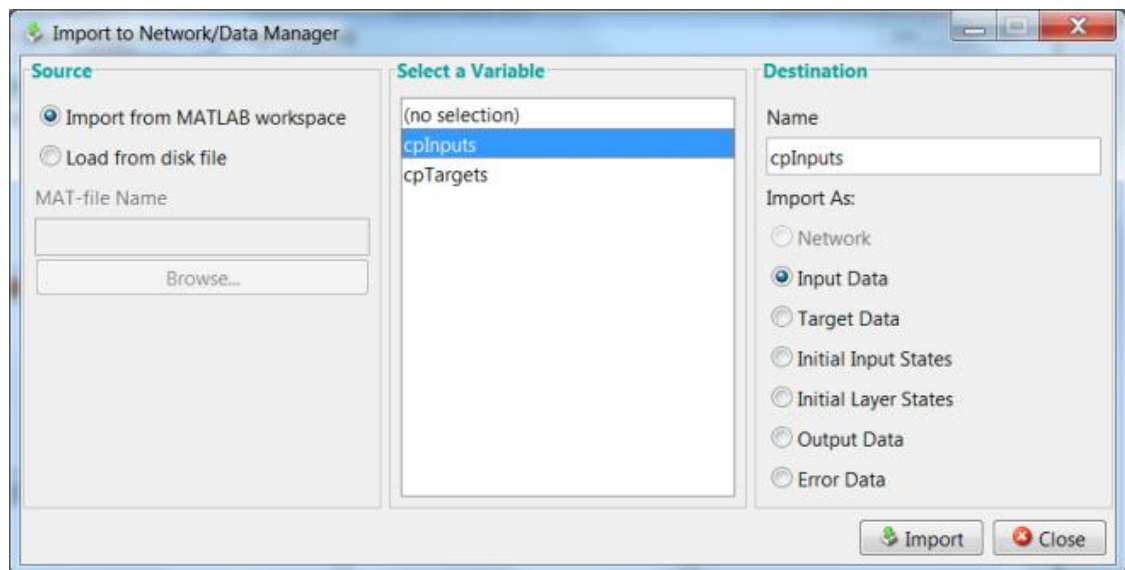


Figure 2-22: Importing data on NNTOOL [30].

Then creating the network by choosing the algorithm, the training function, the number of hidden layers and the number of neurons in each layer and the cost function.

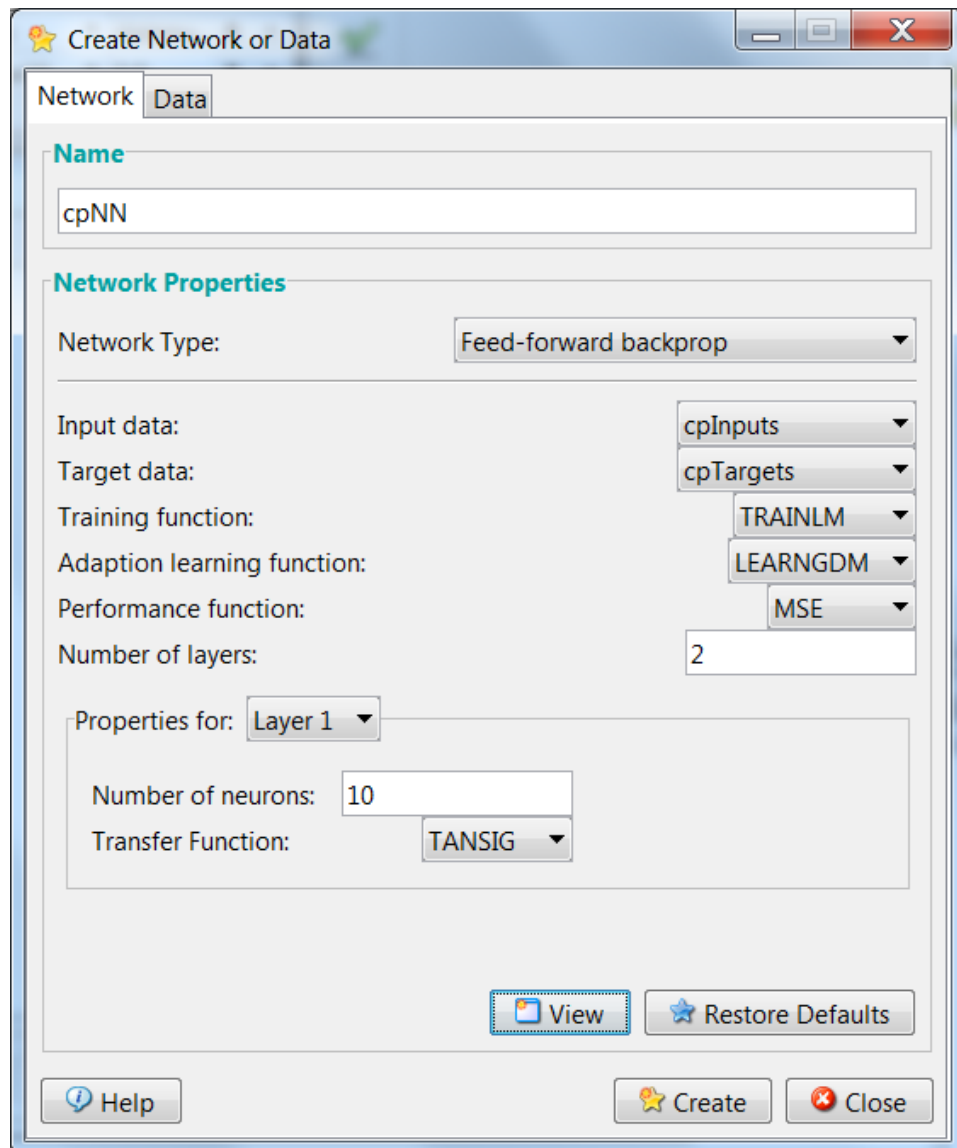


Figure 2-23: Creating the neural network [30].

Finally, launching the calculations.

2.5.2 Training and testing

The algorithm starts the training and learn from the imported data. A window appears to show us the evolution of the training. Indicating the MSE and the epochs – iterations-. The NNTOOL calculates itself the performance, the gradient, the validation check and the epochs of the ANN.

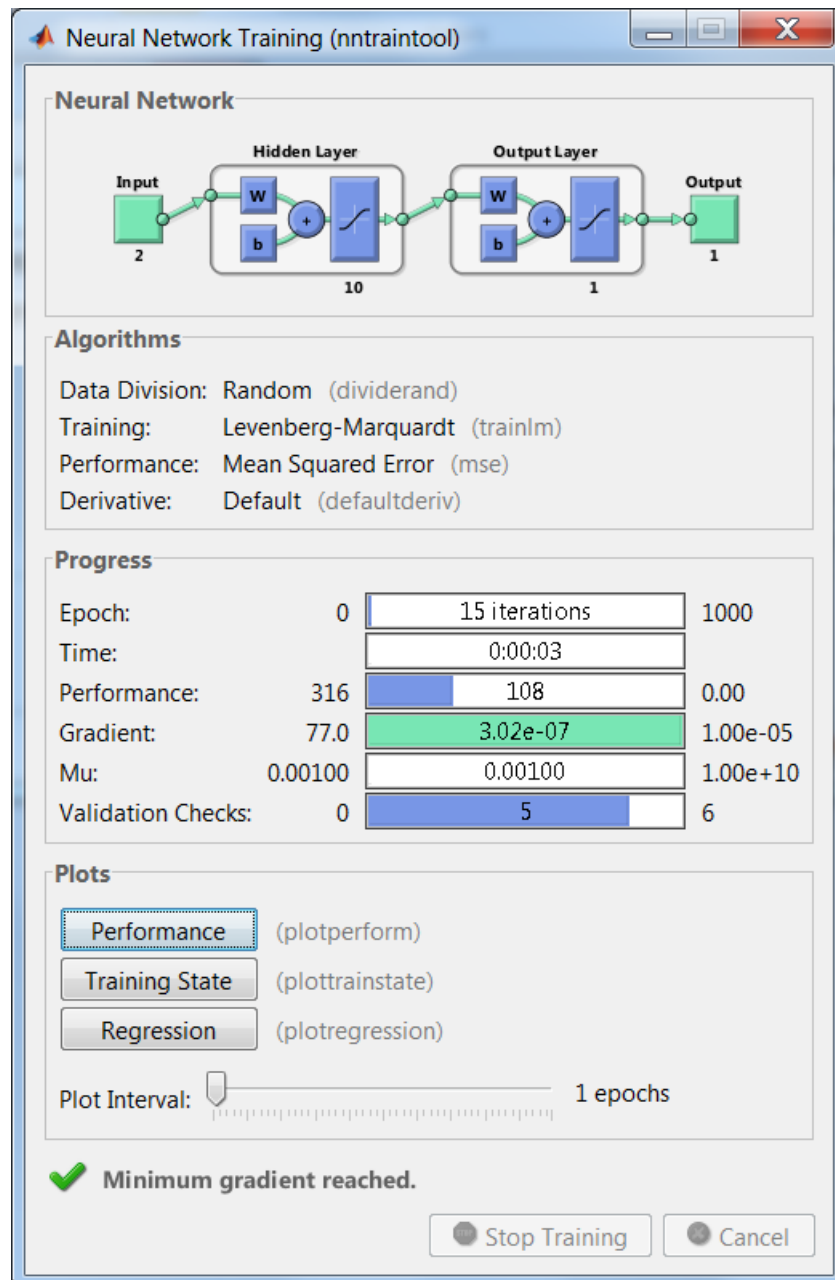


Figure 2-24: Training window [30].

After the training, we test the program with a new set of data to see how it performs.

Then if it works pretty well, we can save the ANN and use it directly or integrate it in another program.

NOTE:

We must go through a period of trial and error in the design decisions before coming up with a satisfactory design. The design issues in neural networks are complex and are the major concerns of system developers.

Designing a neural network consist of:

- Arranging neurons in various layers.
- Deciding the type of connections among neurons for different layers.
- Deciding which activation function to use between layers.
- Deciding which training function to use.
- Determining the strength of connection within the network by allowing the network learn the appropriate values of connection weights by using a training data set.

The process of designing a neural network is an iterative process, and it takes a lot of time, whether in the training (calculations) or in testing all the available possibilities to get the good configuration of the ANN.

Chapter 3: Mathematical formulation and numerical modeling

Chapter 3: Mathematical formulation and numerical modeling

In this chapter, we will present the main mathematical formulation and the numerical modeling used in our study.

3.1 General momentum theory

3.1.1 Axial forces

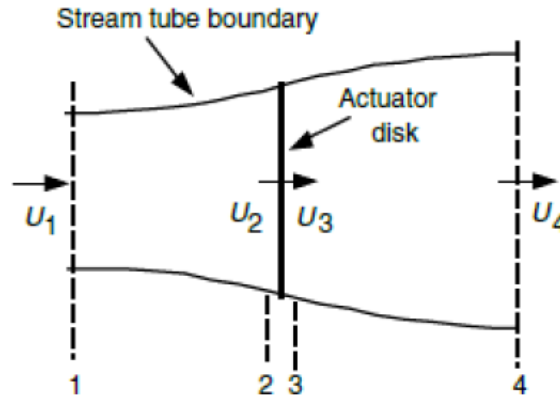


Figure 3-1: Stream tube representation.[34]

Four states are considered around the wind turbine, the wind energy is extracted between two and three causing a change in the pressure. Assuming that $p_1 = p_4$, $U_2 = U_3$ and the flow is frictionless between state one and state two, also between state three and state four. By applying Bernoulli's law, we get:

$$p_2 - p_3 = \frac{1}{2}\rho(U_1^2 - U_4^2) \quad (3.1)$$

In addition, the pressure force is given by:

$$dF_x = (p_2 - p_3)dA \quad (3.2)$$

Combining the two equations (3.1) and (3.2):

$$dF_x = \frac{1}{2}\rho(U_1^2 - U_4^2)dA \quad (3.3)$$

The axial induction factor as:

$$a = \frac{U_1 - U_2}{U_1} \quad (3.4)$$

So, from (3.4) we get:

$$U_2 = U_1(1 - a) \quad (3.5)$$

$$U_4 = U_1(1 - 2a) \quad (3.6)$$

Finally, the pressure force is:

$$dF_x = \frac{1}{2}\rho U_1^2 [4a(1 - 2a)] 2\pi r dr \quad (3.7)$$

3.1.2 Rotation annular stream tube

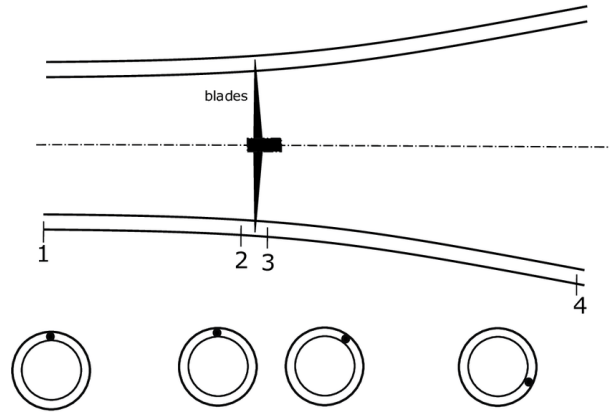


Figure 3-2: Annular stream tube representation.[41]

Considering four states in the rotating annular stream tube, exactly like above. Between the state two and three, the rotation of the rotor blades impacts a rotation into the blade wake.

The torque is given by the next formula:

$$T = \frac{dL}{dt} \quad (3.8)$$

Where L is the angular momentum

$$L = I\omega \quad (3.9)$$

I is the moment of inertia of an annulus, ω is the angular velocity of the blade wake.

$$I = mr^2 \quad (3.10)$$

From these equations, we get:

$$T = \frac{dI\omega}{dt} = \frac{d(mr^2\omega)}{dt} = \frac{dm}{dt}r^2\omega \quad (3.11)$$

For a small element

$$dT = d\dot{m}\omega r^2 \quad (3.12)$$

And $d\dot{m} = \rho AU_2$

$$d\dot{m} = \rho 2\pi r AU_2 dr \quad (3.13)$$

$$dT = \rho U_2 \omega r^2 2\pi r dr \quad (3.14)$$

The induction factor a' :

$$a' = \frac{\omega}{2\Omega} \quad (3.15)$$

$$\text{Hence, } U_2 = U_1(1 - a) \quad (3.16)$$

$$dT = 4a'(1 - a) \rho U_1 \Omega \pi r^3 dr \quad (3.17)$$

The local tip speed ratio λ_r or TSR relative speed is defined as

$$\lambda_r = \frac{\Omega r}{U_1} \quad (3.17)$$

3.2 Blade element theory

The blade is now subdivided into N elements. This allows us to determinate the forces acting on a single element which as following:

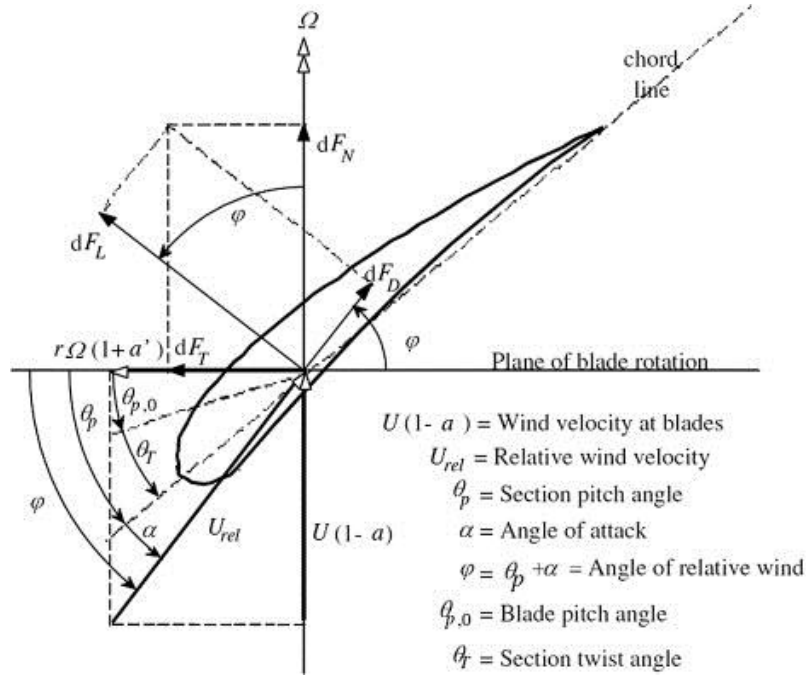


Figure 3-3: Airfoil's forces representation.[42]

The tangential elementary force dF_t is given by:

$$dF_t = dF_l \sin\varphi - dF_d \cos\varphi \quad (3.18)$$

The normal elementary force dF_n is given by:

$$dF_n = dF_l \cos\varphi - dF_d \sin\varphi \quad (3.19)$$

And from the speed triangle's relations we get:

$$\tan\varphi = \frac{1-a}{\lambda_r(1+a')} \quad (3.20)$$

The elementary drag force and lift force are given by:

$$dF_l = \frac{1}{2} C_l \rho U_{rel}^2 c dr \quad (3.21)$$

$$dF_d = \frac{1}{2} C_d \rho U_{rel}^2 c dr \quad (3.22)$$

By replacing the equations (3.21) and (3.22) in the equations (3.18) and (3.19) we get:

$$dF_t = \frac{1}{2} B \rho U_{rel}^2 (C_l \sin\varphi - C_d \cos\varphi) c dr \quad (3.23)$$

$$dF_n = \frac{1}{2} B \rho U_{rel}^2 (C_l \cos\varphi - C_d \sin\varphi) c dr \quad (3.24)$$

Where B is the number of the blades.

The normal force coefficient is defined by: $C_n = (C_l \cos\varphi - C_d \sin\varphi)$ (3.25)

The tangential force coefficient is defined by: $C_t = (C_l \sin\varphi - C_d \cos\varphi)$ (3.26)

The axial load is given by:

$$dQ = dF_t \cdot r = \frac{1}{2} B \rho u_{rel}^2 (C_l \sin\varphi - C_d \cos\varphi) c dr \quad (3.27)$$

3.3 Blade element momentum theory

By combining the blade element theory and the momentum equation, we get:

$$a' = \frac{1}{\frac{4 \sin\varphi \cos\varphi}{\sigma' C_t} - 1} \quad (3.28)$$

$$a = \frac{1}{\frac{4 \sin^2\varphi}{\sigma' C_n} + 1} \quad (3.29)$$

With

$$\sigma' = \frac{B c}{2 \pi r} \quad (3.30)$$

$$C_t = (C_l \sin\varphi - C_d \cos\varphi) \quad (3.31)$$

$$C_n = (C_l \cos\varphi - C_d \sin\varphi) \quad (3.32)$$

This is valid while $C_T < 0.96$ with

$$C_T = \frac{\sigma' (1 - a)^2 (C_l \cos\varphi + C_d \sin\varphi)}{\sin^2\varphi} \quad (3.33)$$

If $C_T > 0.96$, the tip loss correction factor correlation is used:

$$a = \left(\frac{1}{F}\right) [0.143 + \sqrt{0.0203 - 0.6427(0.889 - C_T)}] \quad (3.34)$$

Where F is the tip losses correction factor.

- The correction of the tip losses

We introduce a Prandtl's approach for correcting the top losses, resumed in F number given by the relation:

$$F = \frac{2}{\pi} \cos^{-1} \left[\exp \left(- \frac{\left(\frac{B}{Z} \right) \left(1 - \left(\frac{r}{R} \right) \right)}{\left(\frac{r}{R} \right) \sin \varphi} \right) \right] \quad (3.35)$$

With $0 < F < 1$

And

$$\alpha' = \frac{1}{\frac{4 F \sin \varphi \cos \varphi}{\sigma' C_t} - 1} \quad (3.36)$$

$$a = \frac{1}{\frac{4 F \sin^2 \varphi}{\sigma' C_n} + 1} \quad (3.37)$$

- The correction of the static stall delay

We introduce the correction method proposed by Du & Selig, this approach take on consideration the 3D effect of the stall by adding a correcting coefficient.

$$Cl_{3D} = Cl_{2D} + \Delta Cl_{2D} = Cl_{2D} + f_L (Cl_p - Cl_{2D}) \quad (3.38)$$

$$Cd_{3D} = Cd_{2D} + \Delta Cd_{2D} = Cd_{2D} + f_D (Cd_{2D} - Cd_0) \quad (3.39)$$

With

$$Cl_p = 2\pi(\alpha - \alpha_0) \quad (3.40)$$

$$f_L = \frac{1}{2\pi} \left[\frac{1.6 \left(\frac{c}{r} \right)}{0.1267} \left(\frac{a - \left(\frac{c}{r} \right)^{\frac{dR}{\Lambda r}}}{b + \left(\frac{c}{r} \right)^{\frac{dR}{\Lambda r}}} \right) - 1 \right] \quad (3.41)$$

$$f_D = \frac{1}{2\pi} \left[\frac{1.6 \left(\frac{c}{r} \right)}{0.1267} \left(\frac{a - \left(\frac{c}{r} \right)^{\frac{dR}{2\Lambda r}}}{b + \left(\frac{c}{r} \right)^{\frac{dR}{2\Lambda r}}} \right) - 1 \right] \quad (3.42)$$

And $\Lambda = \frac{\Omega R}{\sqrt{U_0^2 + (\Omega R)^2}}$; a, b and c are the correcting factors considered equal to 1.

There is a relation established by Eggars to obtain ΔCd which is as follows:

$$\Delta Cd = \left(\frac{\sin(\varphi) - X \cos(\varphi)}{X \sin(\varphi) + \cos(\varphi)} \right) \Delta Cl \quad (3.43)$$

With $X = \frac{0.12 \cos(\theta_p) + \sin(\theta_p)}{\cos(\theta_p) - 0.12 \sin(\theta_p)}$

- Drag coefficient in the cylindrical part of the blade

The lift in the blade near the hub is considered equal to zero, because the blade has a cylindrical shape. Moreover, the drag is supposed equal to 1.2. Therefore the Cd changes with the Re. for a cylinder of infinite length:

$$(3.44)$$

$$C_d^{cyl,\infty} = \begin{cases} 1.8 + \frac{6.8}{Re^{0.89}} + \frac{1.96}{\sqrt{Re}} - \frac{0.0004 Re}{1 + 3.64 \times 10^{-7} Re^2} & Re < 2 \times 10^5 \\ 1.6 - 3.04348 \times 10^{-6} Re & 2 \times 10^5 < Re < 4.6 \times 10^5 \\ 0.14321 + 0.1234568 \times 10^{-6} & 4.6 \times 10^5 < Re < 4.5 \times 10^6 \\ 0.7 & Re > 4.5 \times 10^6 \end{cases}$$

Since that the blade is not infinite, so we should correct the Cd

$$C_d^{cyl,AR} = (0.624 + 0.0077 AR) C_d^{cyl,\infty} \quad (3.45)$$

With $AR = \frac{R}{d}$, R is the total radius and d is the diameter of the cylinder.

3.4 Data Collection

We used the Airfoil tools website to download the geometry of the wind-turbine airfoil profile. The website contains one thousand six hundreds of different types of the airfoils.

For the second model which will be presented in the next section, we use all the data available in the website without restriction on the type of the airfoil, whether it is symmetrical or not. We used 328 airfoils of all NACA profiles, all Wortmann, all Clark and all RAE. Except the NACA 0012 and the modified airfoils of NACA 0012.

In this study, we used another input model, which contains the characteristics of the airfoil. The structure of the geometry matrix is as follows:

- The first row contains the maximum thickness of the airfoil
- The second row contains the location of the maximum thickness
- The third row contains the maximum camber of the airfoil
- The fourth row contains the location of the maximum camber

Example of the characteristic input template:

	Airfoil 1	Airfoil 2	Airfoil 3	Airfoil 4	Airfoil 5
1	12	15	15	6	10
2	30	30	29	30	30
3	0	0	1,6	0	24
4	0	0	0	0	10

Table 2: 2nd model data input template.

The structure of the Cl and Cd matrix is as follows:

- First row contains the number of Cl and Cd AOA.
- The other rows contain the Cl and Cd values.
- The last three columns are for the test geometry.

Example of the Cl and Cd input template:

	Airfoil 1			Airfoil 2		
	AOA	Cl	Cd	AOA	Cl	Cd
1	9	9	9	11	11	11
2	-20	0.165	0.01	-20	0.165	0.01
3	-15	0.174	0.013	-18	0.174	0.013
4	-10	0.183	0.016	-12	0.183	0.016
5	-5	0.192	0.019	-9	0.192	0.019
6	0	0.201	0.022	-4	0.201	0.022
7	5	0.21	0.025	0	0.21	0.025
8	10	0.219	0.028	4	0.219	0.028
9	15	0.228	0.031	9	0.228	0.031
10	20	0.237	0.034	12	0.237	0.034
11				18	0.246	0.037
12				20	0.255	0.04

Table 3 : 2nd model data input 2 template.

3.5 Artificial neural network modeling

In this study, we considered three approaches in order to solve the problem.

3.5.1 XY model

The idea of this first model is simply giving the geometry as an input data and Cl and Cd as a target data. This first intuition of using the coordinates of the geometry directly was a good start of our study.

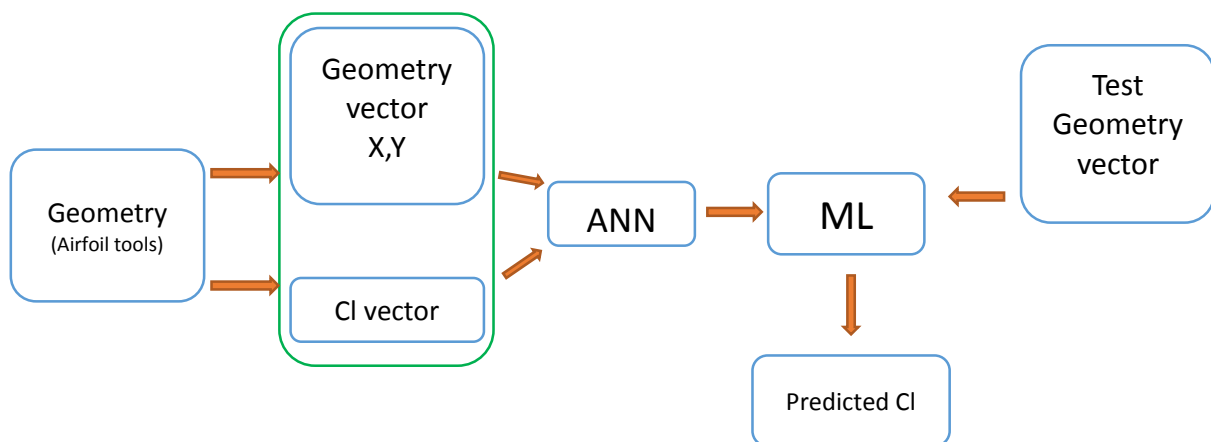


Figure 3-4: 1st model diagram.

The data was not in the same distribution, so we had to clean the data and make it all the same in term of distribution.

The geometries were in a dimensionless scale, varying from zero to one but with different steps. We had to make the steps unique for every geometry.

The solution was to interpolate each geometry from zero to one, then to define the unique step. This step is a variable, it depends of our choice. We programmed it in MATLAB in that way.

Also, for the Cl and Cd data, the same idea has been considered, the interpolation of the data in order to heighten the number of data. This may help on the learning of our ANN.

3.5.2 Airfoil characteristics model

The idea of this second model is using the max thickness value and its position and the max camber value and its position of the airfoil as input data - each airfoil has characteristic parameters of its geometry - and the AOA, Cl as target data.

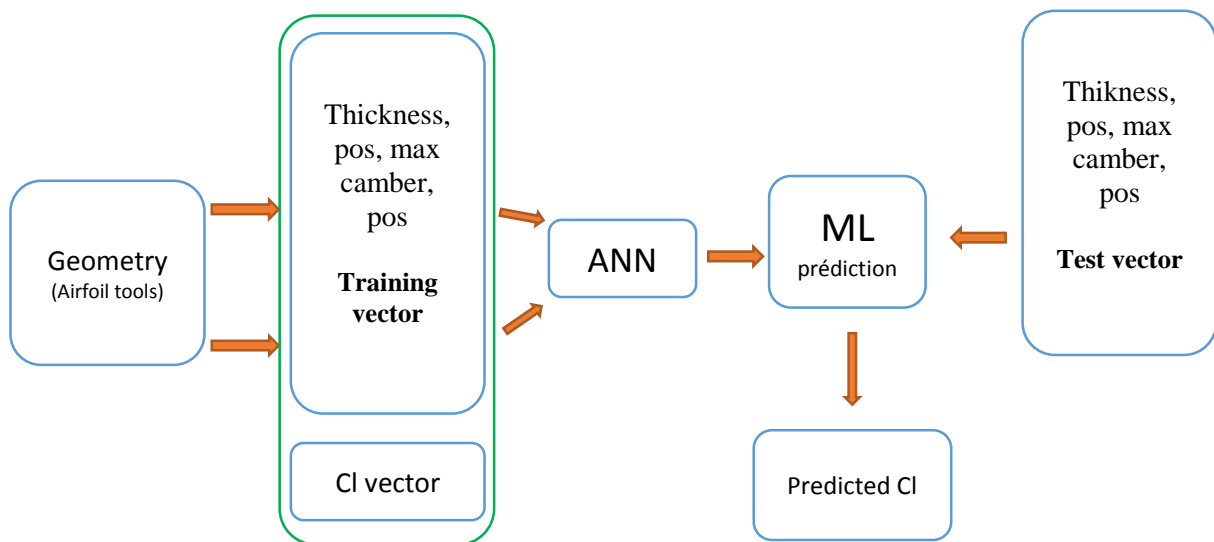


Figure 3-5: 2nd model diagram.

3.5.3 The image model

The idea of this third model is to present the geometry of the airfoil in a binary matrix; this matrix shows the shape of the airfoil. Knowing that the machine learning performs very well when the data is huge, so giving all the detail of the airfoil will help the ANN to get better results.

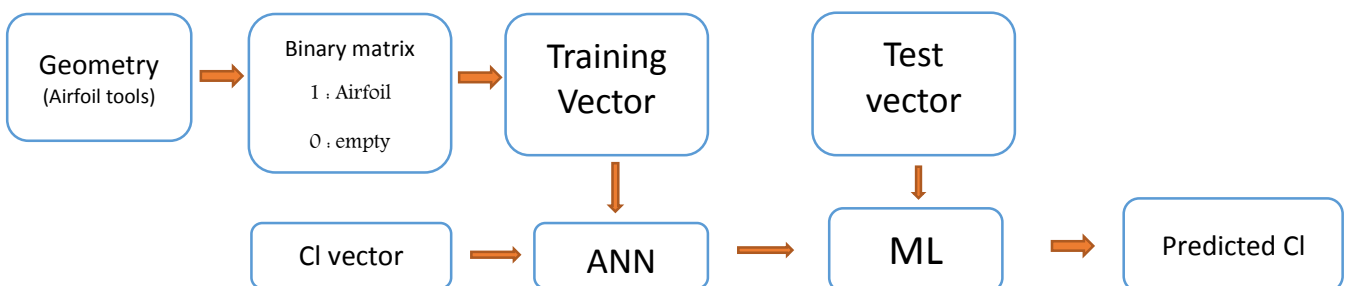


Figure 3-6: 3rd model diagram.

3.6 Training algorithm

One of the hard parts of the machine learning is the design of the ANN, the programmer should make a lot of tests and combinations to find the right design.

The ANN has many variables to change and it takes a lot of time to test all the probabilities.

There is not a way to get directly the right design. The variables are:

- Number of hidden layers
- Number of neurons in each hidden layer
- The learning function
- The activation function of each hidden layer
- The interpolation step of the data

So, we thought to make the process of changing the variables automatic as much as we can.

For a fixed number of hidden layers, we make iterations of the number of neurons in each layer.

Noting that machine learning does not have a rule to get the adequate number of neurons, number of hidden layers, and even the learning method or the activation function to use; this is why we should test all the possibilities in order to get the right configuration.

For example, using the TrainLM learning method and the linear activation function for all the hidden layers. In the case of two hidden layers, we start with one neuron in each layer, then we increase the number of neurons to test all the possibilities.

Number of neurons in 1 st H L	Number of neurons in 2 nd H L
1	1
1	2
1	3
1	4
⋮	⋮
1	50
2	1
2	2
2	3
2	4
2	5
⋮	⋮
2	50
3	1
3	2
⋮	⋮
3	50
⋮	⋮
50	50

Table 4: iterations of number of neurons inside the hidden layers.

In every iteration, we calculate the predicted Cl values of the test airfoils and we compare them with the Cl calculated by the Xfoil software. We save the errors in a vector to make a global performance evaluation of the ANN later.

3.7 BEM-ANN integration

The BEM method uses Cl and Cd values in the computations, there are generally considered as inputs to the method. A data table of Cl and Cd of the specific airfoil shape is used. However, in our case we are using a new airfoil that we do not know its Cl and Cd value. Therefore, in order to get that information, we will use our ANN to predict these values. Moreover, since the BEM method is an iterative method, in each iteration, we should get the Cl and Cd value; we will integrate the ANN to the BEM so that it will be efficient.

In the step of the computation of Cl and Cd value, we will use the ANN. Like it is mentioned in the next figure.

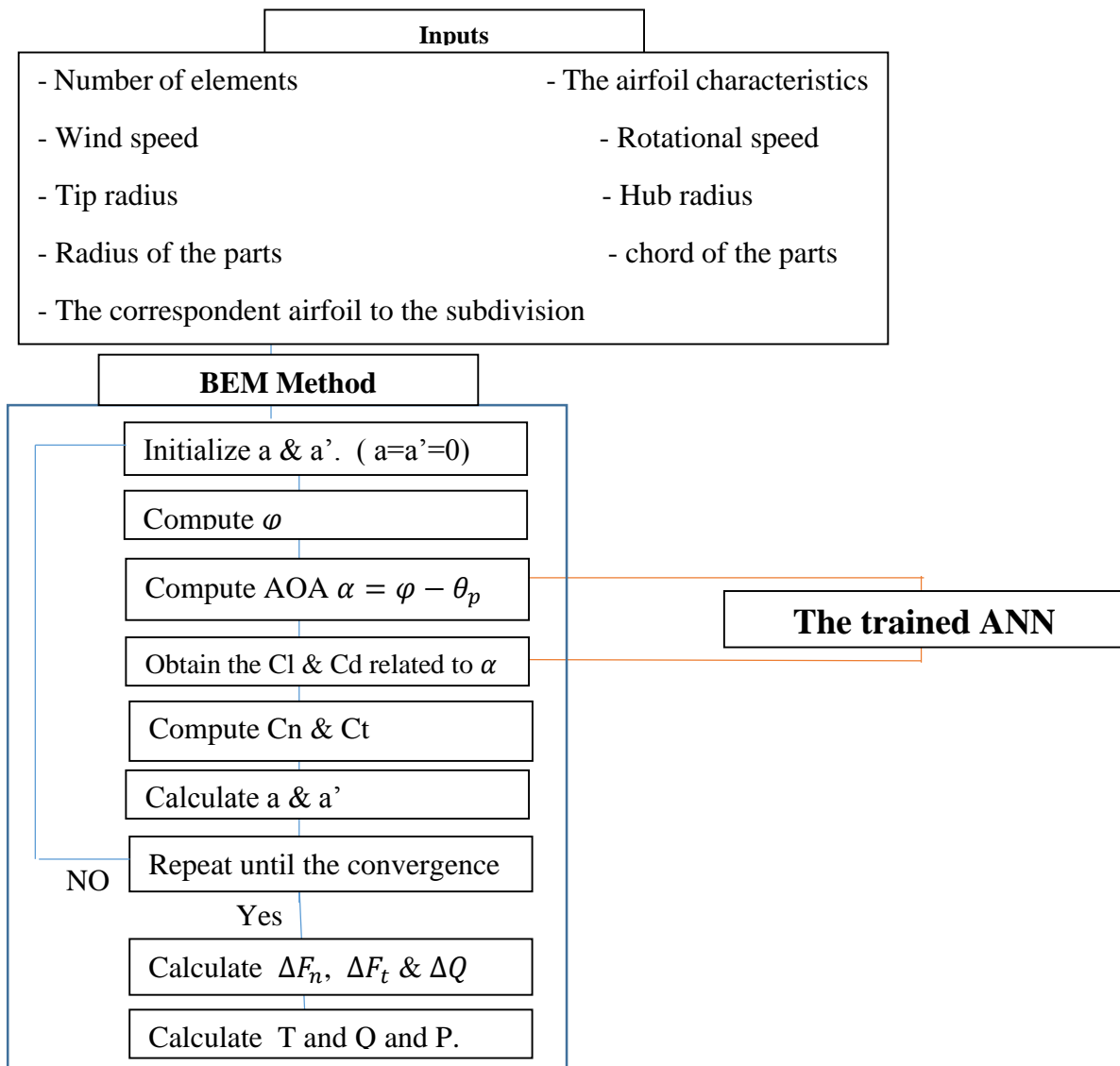


Figure 3-7: BEM-ANN integration diagram.

Chapter 4: Results and discussion

CHAPTER 4: Results and Discussion

4.1 Models choice and validation of lift and drag coefficient

In our study, we studied three models because of some restrictions in the hardware capacity. We were obliged to change the model every time that we could not go so far with results.

In this chapter, we will present the results of each model, and then we will continue the discussion with the good model.

The criteria of validation is the mean square error and the mean relative error. The MSE should has the lowest value and the MRE value should be less than 10^{-2} .

The MRE is calculated by:

$$MRE = \frac{1}{n} \sum_{i=1}^n |Cl_{XFoil,i} - Cl_{ANN,i}| \quad (4.1)$$

Where Cl_{XFoil} is the target result and Cl_{ANN} is the predicted result, n is the number of AOA.

- XY model:

This was our first model, we made the study on the range of -2° to 2° in order to get familiar with the ANN notions, and a database of 6 geometries. Here is the best MRE result for different neurons in two hidden layers presented in the next table for the NACA 0015 geometry.

hidden layer number	Neurons in HL	MRE
2	2_20	0.1660
2	5_20	0.1796
2	5_1	0.1833
2	1_5	0.2019

Table 5: Performance of the XY model.

These values are not good; we need to achieve an error of 10^{-3} at least. Knowing that the database in this case is very small, we tried to use more geometries as input data, but the poor hardware performance of the PC was a barrier for us to extend this model. A lot of data and a huge number of neurons in each layer requires a lot of memory. Therefore, we could not make the calculations, so we changed the model to make it less RAM consumer.

- Image model:

The image model seemed to be a good model since that the geometry is presented in a binary matrix that does not really consume the RAM, and it would give us more space to use more geometries.

Firstly, the model was doing well with a data set of 30 geometries, but it did not show high accuracy. This pushed us to use more geometries, but the same thing with this model after expanding the data, the computer could not make the calculation and it showed error messages about excessive memory use.

- Airfoil characteristics model:

For 328 geometries and a maximum range of AOA [-20°, 20°], this model has good results for the next ANN configuration:

For lift coefficient:

hidden layer number	Neurons in HL	MRE	MSE
7	200 – 200 – 100 – 50 – 50 – 50 – 50	0.0485	0.00448

Table 6: Cl ANN performance.

For drag coefficient:

hidden layer number	Neurons in HL	MRE	MSE
7	200 – 200 – 100 – 50 – 50 – 50 – 50	0.0031	6.7028e-5

Table 7: Cd ANN performance.

Some of the information relative to the model are represented in the next tables:

Training informations for Cl

training function	trainscg
activation function	tansig
number of inputs	5 x 162201 (5 x 328 x AOA)
target data	1 x 162201 (1 x 328 x AOA)
number of outputs	1
number of epochs	5438
times of calculations (hh:mm:ss)	11:33:57
best validation performance at epochs	5432

Table 8: Cl training information.

Training informations for Cd

training function	trainscg
activation function	tansig
number of inputs	5 x 162201 (5 x 328 x AOA)
target data	1 x 162201 (1 x 328 x AOA)
number of outputs	1
number of epochs	1624
times of calculations (hh:mm:ss)	06:44:37
best validation performance at epochs	1618

Table 9: Cd training information.

The graph of the predicted Cl and the Cd vs. AOA of the principal airfoil that we will use in the section of BEM-ANN validation “NACA0012” is presented in the next page:

Airfoil	Coeff.	AOA range	mean error	max error	min error
NACA0012	Cl	-18.5° / 18.5°	0.0485	0.1547	4.94e-04
	Cd	-18.5° / 18.5°	0.0031	0.0213	5.48e-07

Characteristics

12,30,0,0

Table 10: NACA 0012 results.

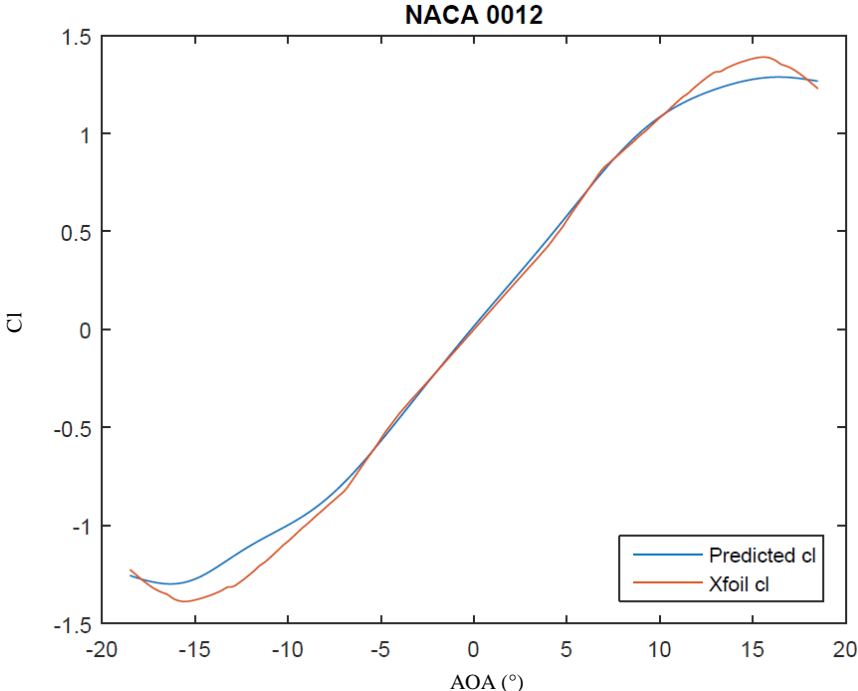


Figure 4-1: NACA 0012 Cl prediction.

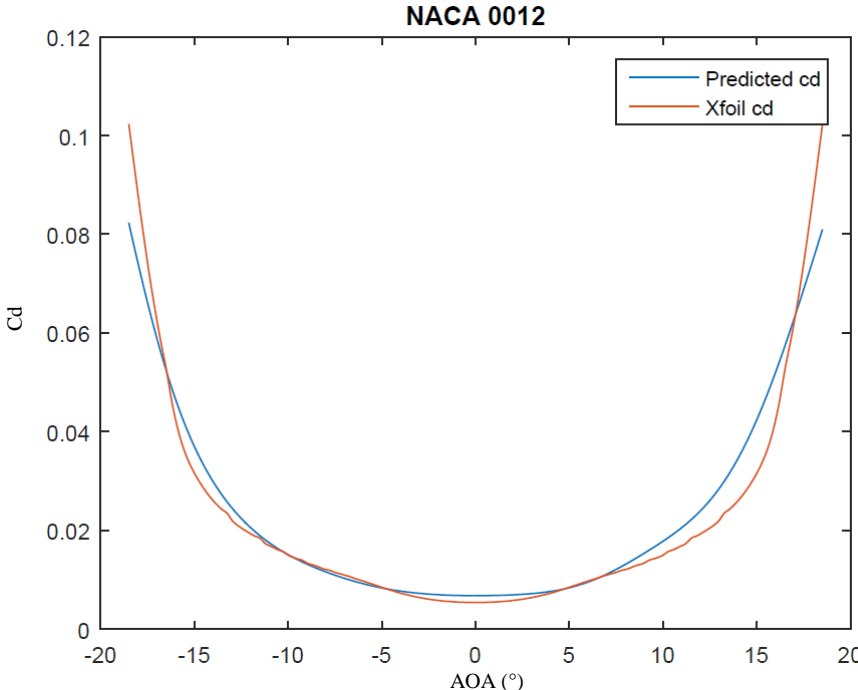


Figure 4-2: NACA 0012 Cd prediction.

The results of prediction for other airfoils:

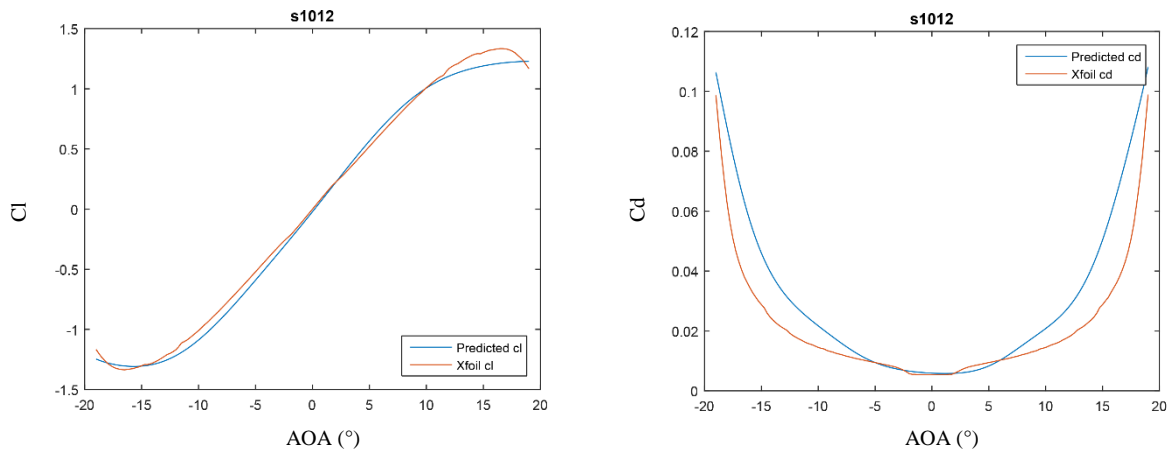


Figure 4-3: S1012 Cl & Cd prediction.

Airfoil	Coeff.	AOA range	mean error	max error	min error
S1012	Cl	-19° / 19°	0.0511	0.123	0.0027
	Cd	-19° / 19°	0.0094	0.0336	4.817e-06

Characteristics

12,37.7,0,0

Table 11: S1012 results.

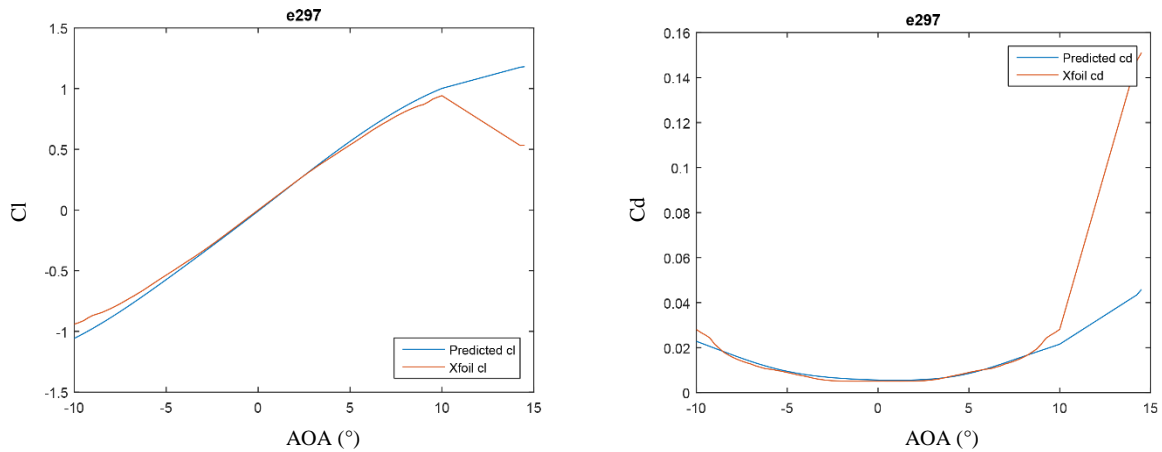


Figure 4-4: E297 Cl & Cd prediction.

Airfoil	Coeff.	AOA range	mean error	max error	min error
E297	Cl	-10° / 14,5°	0.0514	0.6486	0.0005217
	Cd	-10° / 14,5°	0.0038	0.1052	3.825e-05

Characteristics

11.4,37.7,0,0

Table 12: E297 results.

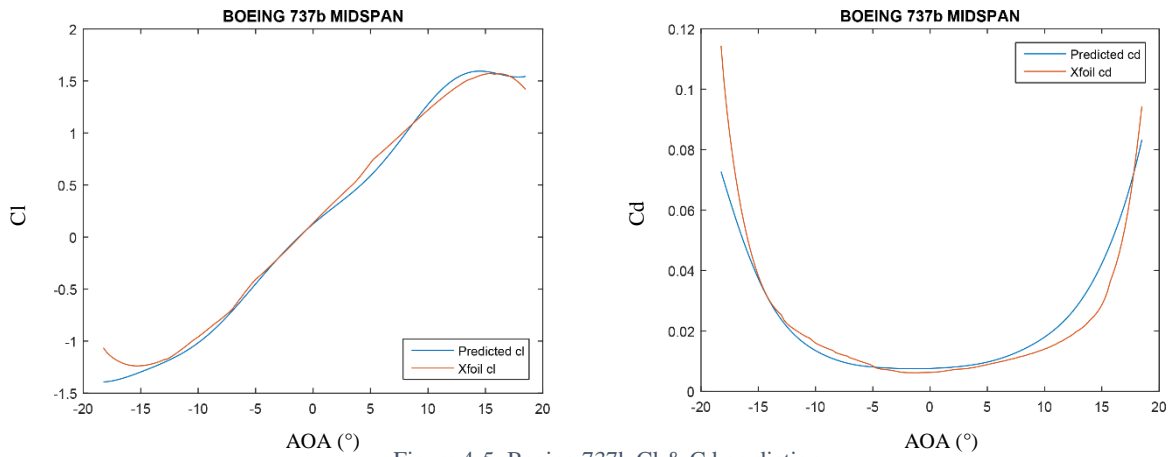


Figure 4-5: Boeing 737b CI & Cd prediction.

Airfoil	Coeff.	AOA range	mean error	max error	min error
boeing 737b midspan cl	Cl	-18.25° / 18.5°	0.0571	0.3282	0.0006746
	Cd	-18.25° / 18.5°	0.0044	0.0417	0.0001229
Characteristics					
12.5, 29.7, 0.8, 10					

Table 13: Boeing 737b results.

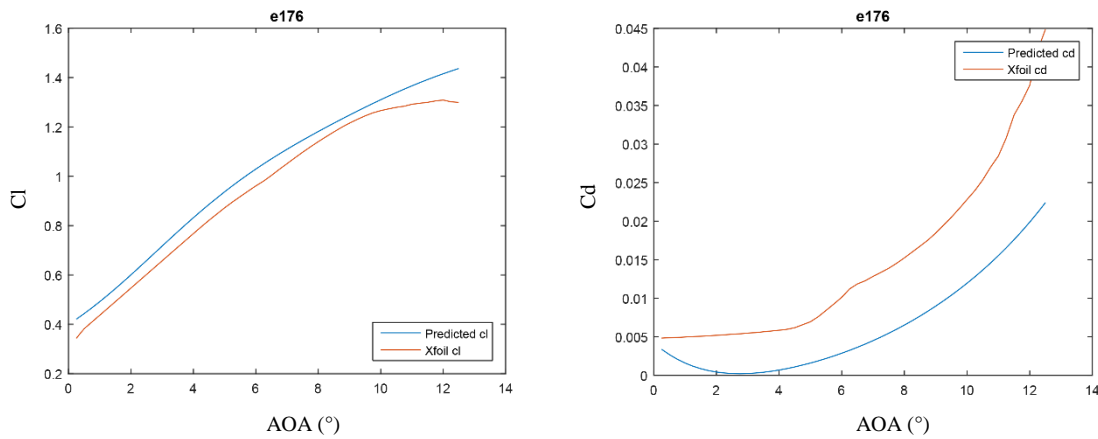


Figure 4-6: E176 CI & Cd prediction

Airfoil	Coeff.	AOA range	mean error	max error	min error
e176	Cl	-0.25° / 12.5°	0.0621	0.1373	0.0326
	Cd	-0.25° / 12.5°	0.0084	0.0225	0.0014
Characteristics					
8.8 ,30.5,2.9,44.2					

Table 14: E176 results.

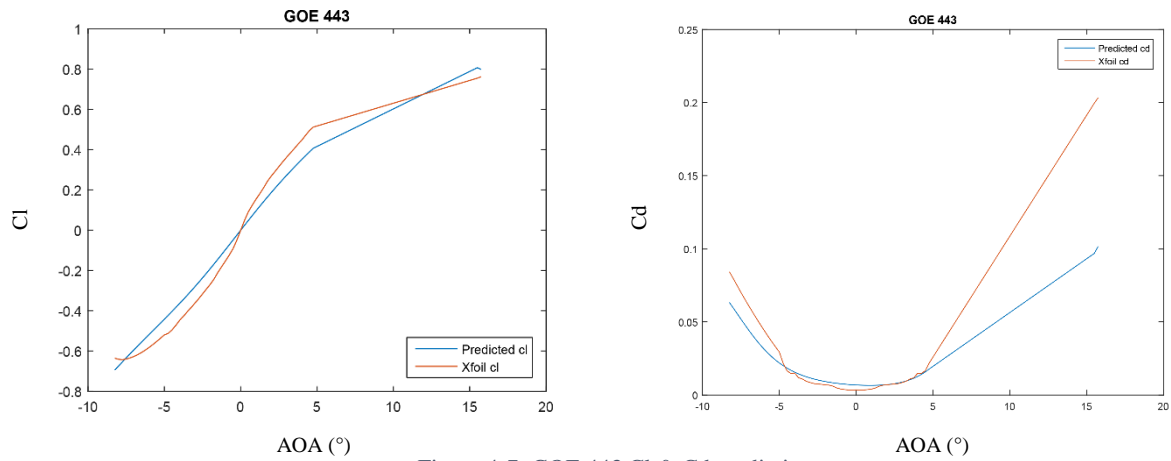


Figure 4-7: GOE 443 Cl & Cd prediction

Airfoil	Coeff.	AOA range	mean error	max error	min error
GOE 443	Cl	-8.25° / 15.75°	0.0662	0.1059	0.000267
	Cd	-8.25° / 15.75°	0.0089	0.1027	8.724e-05

Characteristics

5,30 ,0 ,0

Table 15: GOE 443 results

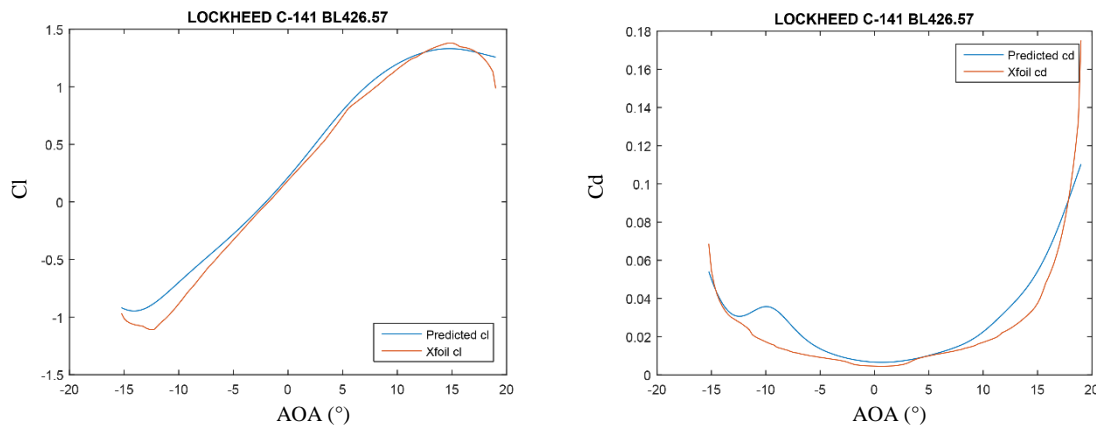


Figure 4-8 : LOCKHEED C-141 Cl & Cd prediction.

Airfoil	Coeff.	AOA range	mean error	max error	min error
LOCKHEED C-141 BL426.57	Cl	-15.25° / 19°	0.0725	0.2726	0.0006794
	Cd	-15.25° / 19°	0.073	0.0647	0.0001235

Characteristics

11,40.2,1.3,50

Table 16 : LOCKHEED C-141 results.

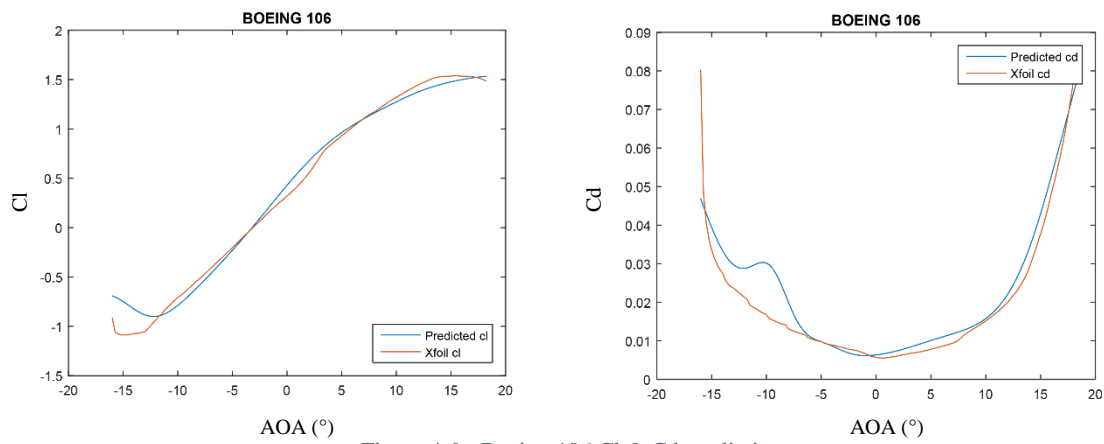


Figure 4-9 : Boeing 106 Cl & Cd prediction

Airfoil	Coeff.	AOA range	mean error	max error	min error
boeing 106	Cl	-16° / 18.25°	0.0741	0.3591	0.0012
	Cd	-16° / 18.25°	0.0037	0.0334	2.428e-05

Characteristics

13.1, 30, 3.3, 40

Table 17 : Boeing 106 results.

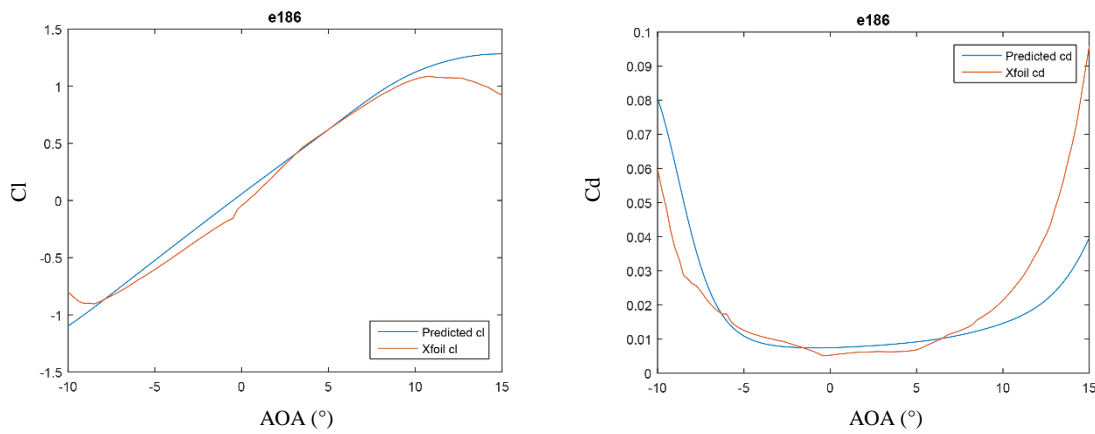


Figure 4-10 : E186 Cl & Cd prediction.

Airfoil	Coeff.	AOA range	mean error	max error	min error
e186	Cl	-10° / 15°	0.0897	0.3599	0.0003725
	Cd	-10° / 15°	0.0084	0.0561	1.753e-05

Characteristics

10.2,29,0.8,29

Table 18 : E186 results.

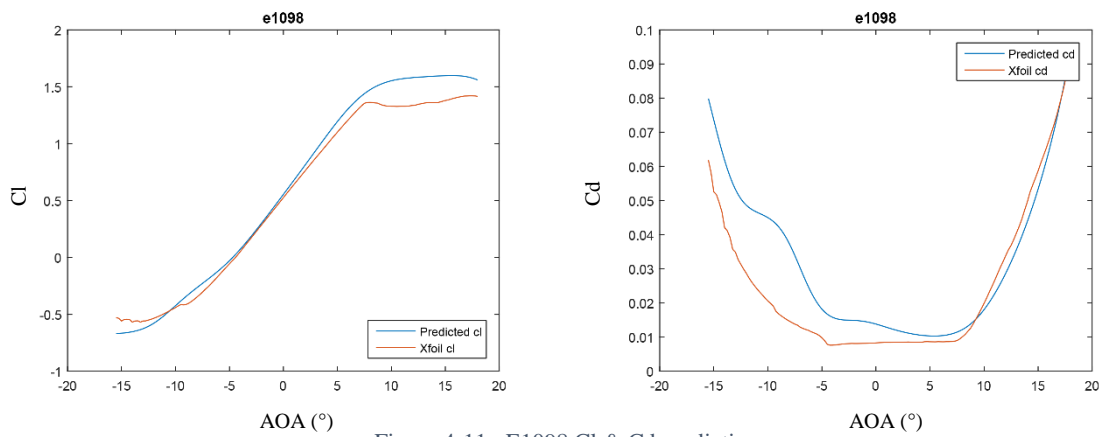


Figure 4-11 : E1098 Cl & Cd prediction.

Airfoil	Coeff.	AOA range	mean error	max error	min error
e1098	Cl	-15.5° / 18°	0.1011	0.2467	0.0002504
	Cd	-15.5° / 18°	0.0088	0.0256	4.454e-05

Characteristics

18.9, 36.6, 3.7, 54.5

Table 19 : E1098 results.

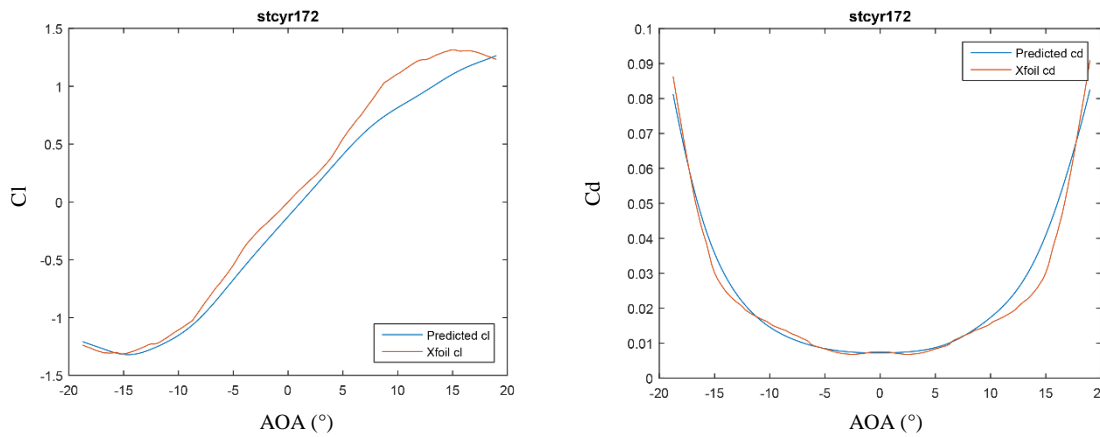


Figure 4-12: STCYR 172 Cl & Cd prediction.

Airfoil	Coeff.	AOA range	mean error	max error	min error
STCYR 172	Cl	-18.75° / 19°	0.1267	0.3013	0.0023
	Cd	-18.75° / 19°	0.0023	0.0108	5.375E-06

Characteristics

13.6,30,0,0

Table 20 : STCYR 172 results.

Discussion of the results:

The principal profile NACA0012 presented a very good prediction result of the Cl and Cd values. Also, for the S1012, Boeing 737b, Boeing 106, Lockheed C141 and GOE 443 presented very good results of the prediction of Cl. STCYR, Boeing 737b and S1012 presented a very good result of the prediction of Cd. Some other airfoils did not present very good results in some range of AOA, it depends on the characteristics value of the airfoil. Therefore, this is due to the training data that is not enough to take all the consideration of airfoil types and all the range of AOA.

The training data should be more expanded and includes a lot of airfoils, more than one thousand to get the best results. This means also having a lot of hardware capacities – huge memory -.

4.2 ANN performance

There is a lot of possibilities to design the ANN, a lot of combinations are possible, the criteria of the selection or the validation of the ANN is the efficiency of its output results.

The performance listed in the tables below are for the Cl number ANN.

4.2.1 Effect of the number of neurons

In this subsection, we will discuss the effect of the number of neurons in the precision and in the efficiency of the model. The increase of number of neurons makes the calculations longer and more expensive. The tables below show the training parameters of one, three and seven hidden layers configuration.

Neurons	MRE	Time	MSE	Epochs
5	0.1728	00:00:22	0.0441	138
10	0.1571	00:00:44	0.0426	225
20	0.1702	00:01:19	0.0396	309
30	0.1608	00:01:45	0.0399	289
40	0.1745	00:02:39	0.377	366
50	0.1306	00:15:50	0.026	1908
70	0.1376	00:30:03	0.0232	2549

Table 21: Comparison of the results of one hidden layer.

Neurons	MRE	Time	MSE	Epochs
10-10-5	0.1685	00:01:42	0.0391	224
15-15-5	0.1301	00:24:09	0.0223	2578
20-10-5	0.17	00:04:56	0.031	577
20-15-5	0.1719	00:02:38	0.0399	265
30-15-5	0.1267	00:49:01	0.0172	4408
30-30-5	0.0928	00:54:29	0.0144	3821

Table 22: Comparison of the results of three hidden layers.

Neurons	MRE	Time	MSE	Epochs
100-100-100-50-50-50-50	0.0494	12:08:41	0.00307	5169
200-100-100-50-50-50-50	0.0502	11:10:05	0.00536	3863
200-200-100-50-50-50-50	0.048	11:33:57	0.00444	5438

Table 23: Comparaision of the results of seven hidden layer.

From these tables, we notice that the variation of the number of the neurons in each layer makes a slight difference in the results for good models as shown in the third table, and a huge difference in the other cases where the models has a low precise result. The MSE decreases with the increase of the number of neurons, but the prediction precision has a random changes. However, as we notice from all these tables that the increase in the number of hidden layers will not necessarily improve the precision. It depends in the complexity and the data distribution of the chosen model.

4.2.2 Effect of the number of hidden layers

In this subsection, we will discuss the effect of the number of hidden layers in the precision and the efficiency of the model. The next table shows the good result of each hidden layer configuration: one, two, three, four and seven hidden layers.

The increase of hidden layers allows us to treat complex and non-linear problems, but the disadvantages is that the calculations become longer and more expensive.

Hidden layers	MRE	Time	MSE	Epochs
200-200-100-50-50-50-50	0.048	11:33:57	0.00444	5438
20-20-20-5	0.0751	01:21:16	0.0103	6229
30-30-5	0.0928	00:54:29	0.0144	3821
30-20	0.1038	00:13:27	0.0226	1376
50	0.1306	00:15:50	0.026	1908

Table 24: Comparaision of the results of diffrent combinaison of hidden layer.

The MSE parameter decreases with the increase of hidden layer, the same note for the MRE, this shows us that the increase of the number of hidden layers makes the model more efficient. The time of calculations and the number of iterations increases which is very logical.

Generally, one or two hidden layers can solve the simple problem, but it is required to use more than two hidden layers for complex ones. The number of hidden layers depends on complexity of the model and the problem. The increase of hidden layers allows us to treat complex and non-linear problems, but the disadvantages is that the calculations become longer and more expensive.

4.2.3 Effect of the training methods

The existence of many training functions makes us wonder which one to us. In this part we will see the effect of the training function on the efficiency and the precision of the model.

The results are shown in the next table for one hidden layer configuration and three training function.

neurons	Trainscg			Trainbfg			Trainlm		
	MRE	Time	MSE	MRE	Time	MSE	MRE	Time	MSE
5	0.173	00:00:22	0.044	0.154	00:01:03	0.042	0.165	00:00:50	0.042
10	0.157	00:00:44	0.043	0.154	00:02:08	0.04	0.323	00:00:11	0.073
20	0.17	00:01:19	0.04	0.162	00:05:17	0.029	0.164	00:00:17	0.062
30	0.161	00:01:45	0.04	0.143	00:13:06	0.024	0.167	00:00:44	0.035
40	0.175	00:02:39	0.377	0.144	00:16:38	0.021	0.894	00:00:14	0.453
50	0.131	00:15:50	0.026	0.128	00:21:24	0.023	/	/	/
70	0.138	00:30:03	0.023	0.139	00:15:38	0.022	/	/	/

Table 25: Comparison of the results of different training methods.

In our case, we used for the training the function Trainscg because it does not use a lot of memory and it is fast in calculations. The other functions did not perform well and we could not use Trainbfg and Trainlm because they require a lot of memory.

In the manual of MATLAB, it has been mentioned that the Trainscg uses less memory and suitable in low memory situations. Trainlm is the fastest but requires a lot of memory. Trainbfg may take longer time than Trainlm.

The Pc that we used has Intel® Core™ i7-5500U CPU @ 2.40 GHz, 16 Go of RAM and Intel® HD Graphics 5500.

The choice of the training function depends on the type of the problem, for example regression training functions does not perform very well in the classification. So, it is required to test all the training functions and choose then the adequate one. In addition, the use of these training methods depends on the hardware material available for the user. Because some training function requires a lot of memory “RAM”.

Conclusion

It is never known with precision the best number of hidden layer and the number of hidden neurons or even the training function to use for a given model for the first insight. Even in literature, some rule for the determination of the number of hidden layers and the number of neurons are made but they still not efficient for every models. So, the good number of hidden layers or neurons and the training function are determined by testing all the possibilities available until getting the efficient results.

4.3 BEM-ANN integration

In the first step, in order to validate the BEM-ANN method, we started with the validation of the Classical BEM using the NREL phase IV with the S809 airfoil. We could not make the comparison with the experimental data using the NACA0012 airfoil because we did not find a reliable source of its performance.

The result of the classical BEM compared to the experimental data are presented in the next figure.

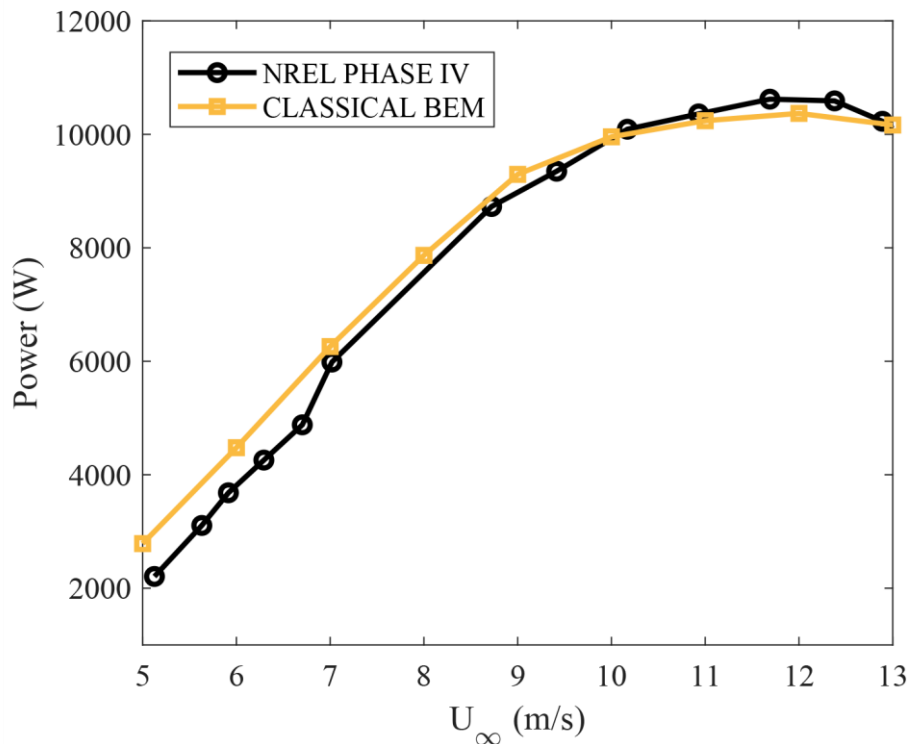


Figure 4-13: NREL phase IV performance.

The figure shows that the classical BEM method is giving good results comparing it to the experimental values; the slight errors are due to the physical phenomena that are not taken in consideration by the classical method. Therefore, we can say that the program of classical BEM is valid and working pretty well.

In the second step, we moved to validate the BEM-ANN method, we made a comparison between the classical BEM method and the BEM-ANN method using the NREL data with a NACA0012 airfoil.

The data of the blade is as follows:

number of blades	2
hub radius (m)	0.883
rotor radius (m)	5.029
Rotational speed (RPM)	71.63

Table 26: NACA0012 blade data

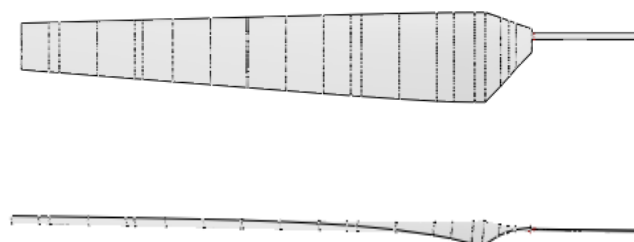


Figure 4-14: NREL phase IV with NACA0012 blade in 3D

The values of C_l and C_d are shown in the next figures:

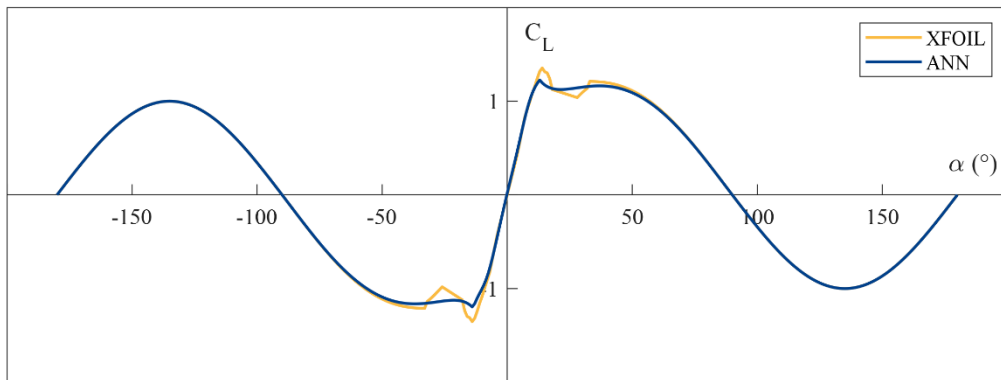


Figure 4-15: Extrapolated C_l data.

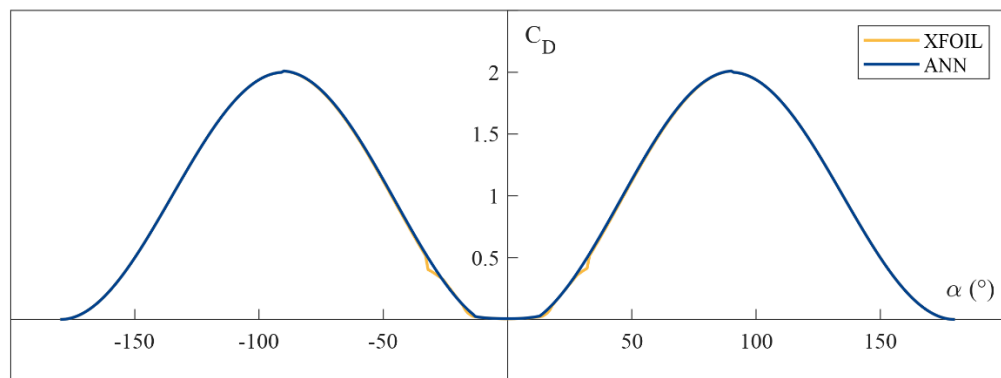


Figure 4-16: Extrapolated C_d data

Note: the values of C_l and C_d are extrapolated to 360° using the Viterna method.

The results of the comparison is presented in the next figure:

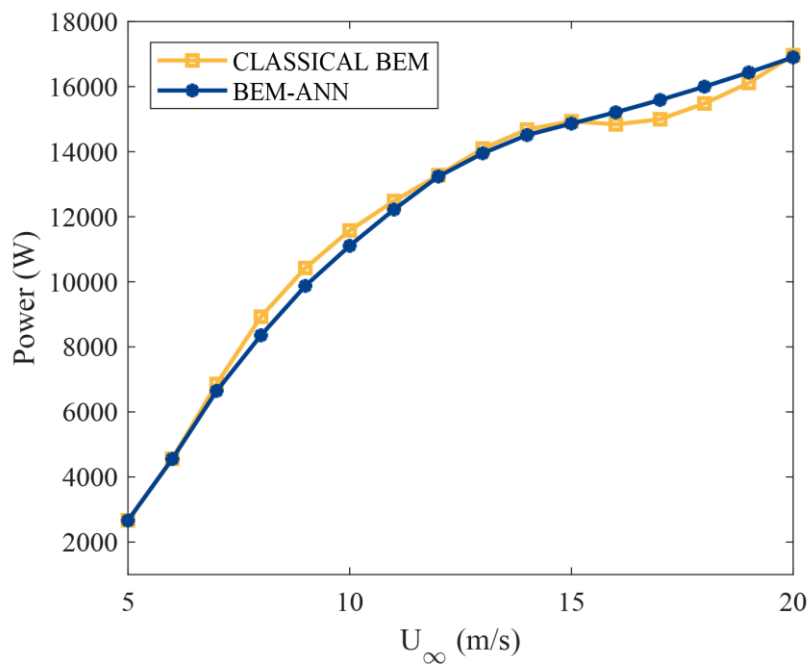


Figure 4-17: Classical BEM vs BEM-ANN.

The figure shows the power graphs drawn with both classical BEM and BEM-ANN. The results are so close for different values of wind speed. The slight difference is due to the stall AOA values, because the values of $C_{l_{ANN}}$ $C_{d_{ANN}}$ are slightly different from the $C_{l_{XFOIL}}$ $C_{d_{XFOIL}}$ values respectively in stall interval.

The maximum error is about 3%, this error is acceptable in the field of mechanics. So this method can be used to predict the performance of a horizontal axis wind turbine for a new airfoil profile, it has a great accuracy and it does the calculations in a lower time.

General conclusion

General conclusion

In this study, an artificial neural network (ANN) approach has been proposed, implemented and validated to predict lift and drag coefficients for horizontal axis wind turbine (HAWTs) applications. The ANN approach can increase the accuracy of the BEM theory through accurate prediction of the lift and drag coefficients. Firstly, a parametric study takes into account the variation of several parameters which influence a lot on the performances of the ANN has been carried out. This parametric study includes the effect of number of neurons, number of hidden layers, learning algorithm, training function, activation function and input method on the performance of the ANN and its accuracy. Then, the optimized ANN architecture was coupled with the BEM theory to predict the performance of HAWTs.

From the work presented in this study, the following conclusions can be drawn:

- The increase of the neurons number in the hidden layers does not ensure the increase in the accuracy of the results and increases the computational time
- Increasing the number of hidden layers increases the accuracy of the results but requires a longer computation time and sometimes leads to convergence problems
- The optimum number of hidden layers was found to be 7HL with [200 – 200 – 100 – 50 – 50 – 50 – 50] hidden neurons
- The `trainscg` learning function which is based on scaled conjugate gradient backpropagation gives accurate results with an acceptable computational time
- The proposed airfoil characteristics input model gave satisfactory results and did not require a large capacity of memory compared to other models (XY model and Image model)
- The proposed ANN-BEM approach is considered as a promising hybrid method that can be used for simulation and optimization of the HAWTs performance.

As perspectives related to this study, on the one hand, it is recommended to use supercomputers and high-performance computing in order to be able to test more the effect of the parameters influencing the performance of the ANN. On the other hand, it is recommended to collect as much data as possible and it is preferable that these data be experimental, or at least from CFD.

References

- [1] Global Change Data Lab. Energy - Our world in data [Online].[consulted on 08/01/2020] available on www.ourworldindata.org/energy
- [2] Wind Europe asbl/vzw. Data & analysis | windEurope. [Online].[consulted on 08/01/2020] available on www.windeurope.org/data-and-analysis
- [3] National agency of investment development. Renewable energy. [Online].[consulted on 08/01/2020] available on www.andi.dz/index.php/en/les-energies-renouvelables
- [4] Wu TY. Flow through a heavily loaded actuator disc. Schiffstechnik 1962. [consulted on 06/03/2020]
- [5] Greenberg MD. Nonlinear actuator disc theory. NASA. 1970. [Online]. [consulted on 06/03/2020] Available on <https://core.ac.uk/download/pdf/85237211.pdf>
- [6] Conway JT. Analytical solutions for the actuator disc with variable radial distribution of load. [Online] .[consulted on 06/03/2020] available on www.cambridge.org/core/journals/journal-of-fluid-mechanics/article/analytical-solutions-for-the-actuator-disk-with-variable-radial-distribution-of-load/DD0A601AF1DF77542B094AF6DDCD99EB
- [7] Xiangfeng Lin, Ji-sheng Zhang , Yuquan Zhang, Jing Zhang. Comparison of Actuator Line Method and Full Rotor Geometry Simulations of the Wake Field of a Tidal Stream Turbine. [Online]. [consulted on 06/03/2020]. available on www.researchgate.net/publication/331851708_Comparison_of_Actuator_Line_Method_and_Full_Rotor_Geometry_Simulations_of_the_Wake_Field_of_a_Tidal_Stream_Turbine
- [8] L.N. Sankar. Computational studies of horizontal axis wind turbines - annual status report 2000-2001. Technical report. [Online] [consulted on 06/03/2020] available on citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.461.5475&rep=rep1&type=pdf
- [9] Snel H, Houwink B, Bosschers J, Piers WJ, van Bussel GJW, Bruining A. Sectional prediction of 3-D effects for stalledflow on rotating blades and comparison with measurements.
- [10] Chaviaropoulos.P.K & Hansen.M.O.L 2000, investigating three-dimensional and rotational effects on wind turbine blades by means of a quasi-3D navier stokes solver.
- [11] Raj, NV, An Improved Semi-Empirical Model for 3-D Post-Stall Effects in Horizontal Axis Wind Turbines, Master of Science Thesis in Aeronautical and Astronautical Engineering, University of Illinois at Urhana-Champaign: Urhana, IL,2000
- [12] Du.Z & Selig.M.A 1998, "A 3-D stall-delay model for horizontal axis wind turbine performance prediction. [Online] [consulted on 08/03/2020] available on experts.illinois.edu/en/publications/a-3-d-stall-delay-model-for-horizontal-axis-wind-turbine-performa-2

-
- [13] Bak.C, Johansen.J, & Andersen.P.B 2006, "Three-dimensional corrections of aerofoil characteristics based on pressure distributions", Proceedings of the European Wind Energy Conference.
- [14] J G Leishman and T S Beddoes. A Semi-Empirical Model for Dynamic Stall. [consulted on 08/03/2020] Available on www.semanticscholar.org/paper/A-Semi-Empirical-Model-for-Dynamic-Stall-Leishman-Beddoes/cd957c91f6d35c6ae85a7ed06ddf640c0e8f7bfe
- [15] C T Tran and D Petot. Semi-empirical model for the dynamic stall of airfoils in view to the application to the calculation of responses of a helicopter blade in forward flight.
- [16] Morten Hartvig Hansen, Mac Gaunaa, and Helge Aagaard Madsen. A Beddoes-Leishman type dynamic stall model in state-space and indicial formulations. [Online] [consulted on 08/03/2020] available on www.researchgate.net/publication/229058188_A_Beddoes-Leishman_Type_Dynamic_Stall_Model_in_State-space_and_indicial_formulations
- [17] J. W. Larsen, S. R K Nielsen, and S. Krenk. Dynamic stall model for wind turbine airfoils. [Online] [consulted on 08/03/2020] available on www.sciencedirect.com/science/article/abs/pii/S0889974607000291
- [18] H Snel. Heuristic modelling of dynamic stall characteristics.
- [19] Okamoto. M., Yasuda. K., and Azuma. A. Aerodynamic Characteristics of the Wings and Body of a Dragonfly. [online] [consulted on 09/03/2020] available on jeb.biologists.org/content/199/2/281
- [20] NASA, Airfoil Tools. [Consulted on 03/01/2020] available on airfoiltools.com
- [21] Su Liu, Isam Janajreh. Development and application of an improved blade element momentum method on horizontal axis wind turbines. [Online] [consulted on 09/03/2020] available on https://www.researchgate.net/publication/257885705_Development_and_application_of_an_improved_blade_element_momentum_method_model_on_horizontal_axis_wind_turbines
- [22] R, Malik, A.J. Williams, T.N. Croft, M. Togneri, I. Masters. A coupled blade element momentum – Computational fluid dynamics model for evaluating tidal stream turbines performance. [Online] [consulted on 10/03/2020] available on www.sciencedirect.com/science/article/pii/S0307904X12004374
- [23] Abdulaziz Abutunis, Rafid Hussein, K. Chandrashekhara. A neural network approach to enhance blade element momentum theory performance for horizontal axis wind turbine application. [Online] [consulted on 01/03/2020] available on www.sciencedirect.com/science/article/abs/pii/S0960148118311807
- [24] OZLEM CEYHAN, Aerodynamic design and optimization of horizontal axis wind turbine by using BEM theory and genetic algorithm. [Online] [consulted on 10/03/2020] available on etd.lib.metu.edu.tr/upload/12610024/index.pdf
- [25] Coursera, machine learning. [Online] [Consulted on 03/01/2020] available on fr.coursera.org/learn/machine-learning

-
- [26] P.A.Costa Rocha, H.H.Barbarosa Rocha, F.O.Moura Carneio, M.E.Vieira da Silva, C.Freitas de Andrade. A case study on the calibration of the k-w SST (shear stress transport) turbulence model for small scale wind turbines designed with cambered and symmetrical airfoils. [Online] [consulted on 25/03/2020] available on www.sciencedirect.com/science/article/abs/pii/S0360544215017260?via%3Dihub
- [27] Paulo Alexandre Costa Rocha, Daniel Albiero, Marcos Paulo Gomes Fernandes, Jefferson Lemos Fernandes. Aerodynamic performance of NACA four-digit profiles in small-scale wind turbines for small farms. [Online] [consulted on 25/03/2020] available on www.researchgate.net/publication/279914708_Aerodynamic_performance_of_NACA_four-digit_profiles_in_small-scale_wind_turbines_for_small_farms
- [28] M.O.L. Hansen, J.N. Sørensen, S. Voutsinas, N. Sørensen, H.Aa. Madsen. State of the art in wind turbine aerodynamics and aeroelasticity. [Online] [consulted on 29/03/2020] available on www.sciencedirect.com/science/article/abs/pii/S0376042106000649
- [29] Matlab help documentation, training functions
- [30] Matlab figures
- [31] Joseph Katz and Allen Plotkin. Low Speed Aerodynamics. Cambridge University Press, Cambridge, New York, Melbourne, Madrid, Cape Town, Singapore, São Paulo, Delhi, Dubai, Tokyo, Mexico City, second edition, 2010. ISBN 0-521-66219-2.
- [32] electrical engineering portal. Aerodynamic of horizontal axis wind turbines. [Online] [consulted on 28/03/2020] available on <https://electrical-engineering-portal.com/aerodynamics-of-horizontal-axis-wind-turbines>
- [33] A. SMAILI. Notes of renewable energy course. Chapter 3 Aerodynamics of wind turbines. Elearning.gm.enp.edu.dz .[consulted on 28/03/2020]
- [34] PennState. The Actuator Disc Model | AERSP 583 Wind Turbine Aerodynamics. [Online] [consulted on 06/03/2020] available on <https://www.e-education.psu.edu/aersp583/node/470>
- [35] Aero12. Angle of attack. [Online] [consulted on 08/03/2020] available on http://www.geocities.ws/pegasusair_tr/week241.htm
- [36] Kay Gemba, Measurement of boundary layer on flat plate. [Online] available on <http://web.iitd.ac.in/~pmvs/courses/mel705/boundarylayer.pdf>
- [37] Dr. Y.S. Muzychka. ENGR 5961 Fluid mechanics 1. [Online] [consulted on 02/05/2020] available on <http://www.engr.mun.ca/muzychka/Fluids-Section-5.pdf>
- [38] Thermopedia. Boundary layer. [Online] available on <http://www.thermopedia.com/content/595/>
- [39] Prof. Galal Bahgat Salem. Chapter 5 Airfoils. [Online] available on <https://www.slideshare.net/anashalim/aer-101-chapter-5>
- [40] Tailwinds. Pressure distribution. [Online] available on <http://runs2xs.blogspot.com/2006/03/pressure-distribution.html>

-
- [41] Galih Bangga. Enhancing BEM simulations of a stalled wind turbine using a 3D correction model. [Online] available on https://www.researchgate.net/figure/Rotating-annular-stream-tube-at-various-streamwise-positions_fig1_323924102
- [42] Adam Chehouri, Rafic Younes, Adrian Ilinca, Jean Perron. Review of performance optimization techniques applied to wind turbines. [Online] available on <https://ars.els-cdn.com/content/image/1-s2.0-S0306261914013002-gr3.jpg>
- [43] BEM theory by Glauert (1926).