

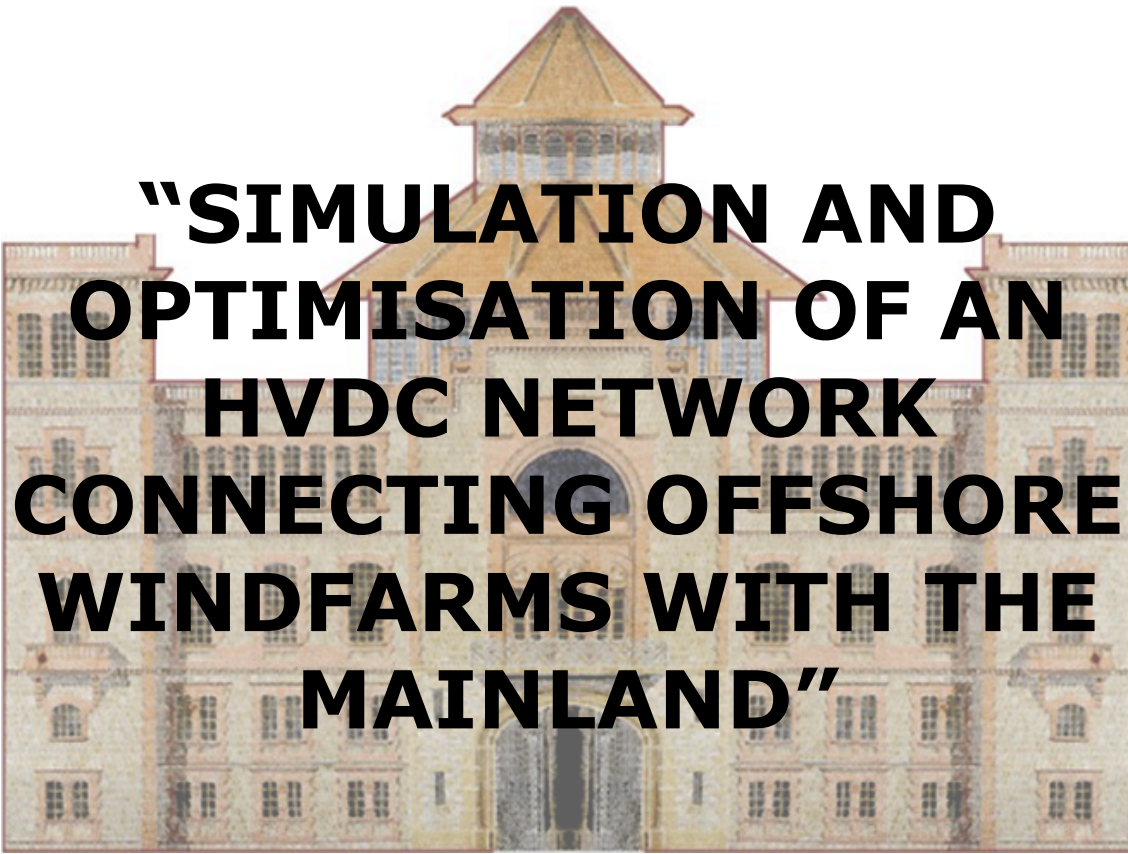


Escola Universitària d'Enginyeria
Tècnica Industrial de Barcelona
Consorci Escola Industrial de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Project report - Budget

TREBALL FINAL DE GRAU



"SIMULATION AND OPTIMISATION OF AN HVDC NETWORK CONNECTING OFFSHORE WINDFARMS WITH THE MAINLAND"

TFG presentat per obtenir el títol de GRAU en
ENGINYERIA ELÈCTRICA

Per **Nil Falgueras Farrerons**

Barcelona, 08 de Juny de 2016

Director: Arnau Dòria Cerezo
Departament d'Enginyeria Elèctrica (EE - 709)
Universitat Politècnica de Catalunya (UPC)

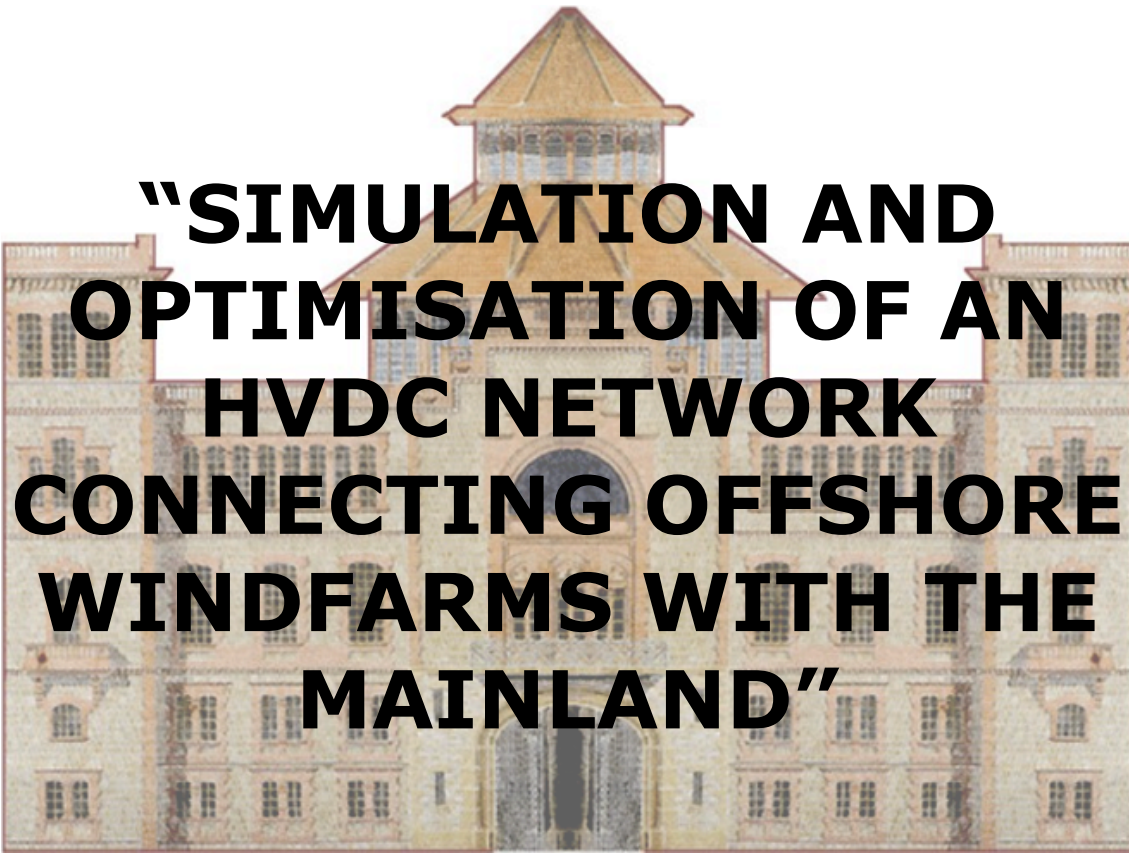


Escola Universitària d'Enginyeria
Tècnica Industrial de Barcelona
Consorci Escola Industrial de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Project report

TREBALL FINAL DE GRAU



"SIMULATION AND OPTIMISATION OF AN HVDC NETWORK CONNECTING OFFSHORE WINDFARMS WITH THE MAINLAND"

TFG presentat per obtenir el títol de GRAU en
ENGINYERIA ELÈCTRICA

Per **Nil Falgueras Farrerons**

Barcelona, 08 de Juny de 2016

Director: Arnau Dòria Cerezo
Departament d'Enginyeria Elèctrica (EE - 709)
Universitat Politècnica de Catalunya (UPC)

TABLE OF CONTENTS

List of figures.....	6
List of tables.....	6
RESUMEN	7
RESUMEN	7
ABSTRACT	7
CHAPTER 1: INTRODUCTION	9
1.1 Aims and goals of this project.....	9
1.2 Context and motivation	9
1.3 Wind power and offshore wind farms	10
1.4 HVDC transmission	12
CHAPTER 2: MODEL OF AN HVDC NETWORK CONNECTING OFFSHORE FARMS WITH THE MAINLAND	15
2.1 HVDC network mathematical model.....	15
2.2 Wind energy mathematical model.....	18
CHAPTER 3: Control and supervision of HVDC power grids	21
3.1 Control scheme	21
3.2 Decentralized voltage controller.....	22
3.3 Secondary control: Optimal power flow	23
CHAPTER 4: Simulations	25
4.1 General description	25
4.2 Code	27
4.2.1 Excel file.....	27
4.2.2 Linking the different simulations	30
4.2.3 Initial conditions.....	31
4.2.4 Optimal Power Flow.....	32
4.3 Simulations results.....	35
CHAPTER 5: CONCLUSIONS AND FUTURE WORK.....	41
CHAPTER 6: BIBLIOGRAPHY	43
6.1 Bibliographic references.....	43
6.2 Complementary bibliography	44
ANNEX: MATLAB CODE.....	47
Sim.m file	47
Ode4CG.m file.....	53
OPFnlc.m file	55
OPFof.m file.....	55
plots.m file.....	55
Budget.....	61

LIST OF FIGURES AND TABLES

List of figures

FIGURE 1.1. <i>WORLWIDE NEW INSTALLED CAPACITY PER YEAR SINCE 2000 TO 2015.</i> (GWEC, 2015)	10
FIGURE 1.2. <i>WORLWID INSTALLED CAPACITY SINCE 2000 TO 2015. (GWEC, 2015).</i>	10
FIGURE 1.3. <i>ANNUAL ONSHORE AND OFFSHORE INSTALLATIONS. (EWEA, 2016)</i>	11
FIGURE 1.4. <i>COMPARISON OF AC AND DC TRANSMISSION SYSTEM COSTS. (SOOD, 2007).</i>	13
FIGURE 1.5. <i>BASIC SCHEME OF AN INTERCONNECTION BETWEEN AC TERMINALS USING HVDC TRANSMISSION. (FELTZ, 2016).</i>	14
FIGURE 2.1. <i>EQUIVALENT CIRCUIT OF A VSC.</i>	16
FIGURE 2.2. <i>EQUIVALENT CIRCUIT OF A TRANSMISSION LINE.</i>	16
FIGURE 2.3. <i>CUMULATIVE RELATIVE WIND SPEED (BLUE) AND ITS WEIBULL APPROXIMATION (RED).</i>	18
FIGURE 2.4. <i>WIND TURBINE POWER CURVE. IMAGE OBTAINED FROM HTTP://WWW.WIND-POWER-PROGRAM.COM/POPUPS/POWERCURVE.HTM.</i>	20
FIGURE 3.1. <i>HIERARCHICAL CONTROL SCHEME.</i>	21
FIGURE 3.2. <i>ADMISSIBILITY REGION: POWER INJECTION (BLUE); POWER CONSUMPTION (RED). (DÒRIA-CEREZO ET AL. 2015)</i>	22
FIGURE 3.3. <i>DROOP CONTROL CURVES FOR A WIND FARM CONVERTER (BLUE) AND GRID STATION (RED).</i>	23
FIGURE 4.1. <i>SCHEME OF THE ALGORITHM APPLIED IN (GRAU, 2015).</i>	26
FIGURE 4.2. <i>IDEA OF THE PROCESS APPLIED IN THE SIMULATION PART.</i>	26
FIGURE 4.3. <i>APPROXIMATION OF THE INTERCONNEXION OF THE STUDY CASE.</i>	35
FIGURE 4.4. <i>EVOLUTION OF VSC VOLTAGE IN EACH NODE.</i>	37
FIGURE 4.5. <i>EVOLUTION OF THE VSC CURRENT IN EACH NODE.</i>	37
FIGURE 4.6. <i>EVOLUTION OF VSC POWER IN EACH NODE.</i>	38
FIGURE 4.7. <i>EVOLUTION OF THE CURRENT IN EACH LINE OF THE NETWORK.</i>	38
FIGURE 4.8. <i>SPLINE INTERPOLATED WINDS VALUES, IN THIS CASE THREE DIFFERENT POINTS WERE USED TO DEFINE THE SPLINE.</i>	39
FIGURE 4.9. <i>GENERATED POWER OF THE WIND FARMS OF THE NETWORK.</i>	39

List of tables

TABLE 4.1. <i>TRANSMISSION LINES IN CASE STUDY.</i>	36
TABLE 4.2. <i>NODES IN CASE STUDY.</i>	36
TABLE 4.3. <i>PARAMETERS OF THE SECONDARY CONTROL (OPF).</i>	36

RESUMEN

En aquest projecte s'ha creat un algoritme capaç de simular una xarxa HVDC connectant parcs eòlics marins amb punts consumidors. Aquesta simulació té en compte el model analític en que es fonamenten les xarxes HVDC. La simulació també té en compte el control que s'aplica en aquest tipus de casos. A més, el mateix control del sistema permet calcular el punt òptim de treball per tal de que la xarxa optimitzi els resultats de transferència d'energia.

RESUMEN

En este proyecto se ha creado un algoritmo capaz de simular una red HVDC conectando parques eólicos marinos con puntos consumidores. Esta simulación tiene en cuenta el modelo analítico en que se fundamentan las redes HVDC. La simulación también tiene en cuenta el control que se aplica en este tipo de casos. Además, el mismo control del sistema permite calcular el punto óptimo de trabajo para que la red optimice los resultados de transferencia de energía.

ABSTRACT

In this project an algorithm capable of simulating a network connecting offshore wind farms HVDC points with consumers has been created. This simulation takes into account the analytical model that HVDC network is based on. The simulation also takes into account the control that is applied in such cases. In addition, the control of the system calculates the optimum working point to optimise the network energy transfer.

CHAPTER 1:

INTRODUCTION

1.1 Aims and goals of this project

There is not a unique goal in this project. The first aim of this document is to develop an algorithm capable of modelling and simulating an HVDC (High Voltage Direct Current) network interconnecting offshore wind farms with the mainland. At the same time it will be studied, and implemented in the simulation, the possibility to optimise the behaviour of the network. While developing the algorithm, an educational goal of learning and improving the knowledge in the mathematical model which the project is based on.

1.2 Context and motivation

Contemporary societies have developed a high dependency on energies. New societies have been developing at a high speed in the recent centuries, and have achieved high comfortable life. The energy plays a vital role in the economic growth, progress and development of any society.

Until now the generation of electricity comes from fossil fuel and non-renewable resources, reaching around 80%. However this type of resources present some big problems, like the extinction of the fossil fuels are the big impact on the CO₂ emissions. Nowadays, 60% of the global emissions of CO₂ come from energy. Moreover, the levels of CO₂ are increasing

dangerously day by day, reaching levels that never have been achieved. As it is known, these emissions are thought to be the principal cause of the climate change.

During the past years, the production of energy has been focused on developing sustainable energies, and those are becoming more important nowadays. Until now, the renewable energies accounted for 77% of new power plant installation in 2015, with 22,3GW of a total of 29 GW. This means quite a huge growing of the sector, and there is clear evidence that this development will not stop.

1.3 Wind power and offshore wind farms

Wind power generation stands out from the other renewable energies, as it has had an enormous development in the last years. In year 2015 wind power installed more power plants than in 2014, a 5,4% more. Moreover, wind power installed more than any other form of power generation, accounting for 44,2% of all power capacity installations.

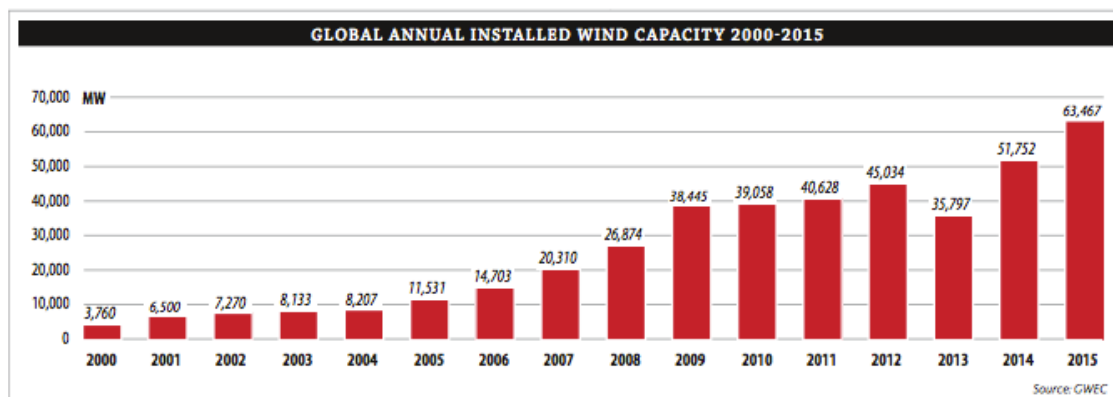


Figure 1.1. Worldwide new installed capacity per year since 2000 to 2015. (GWEC, 2015)

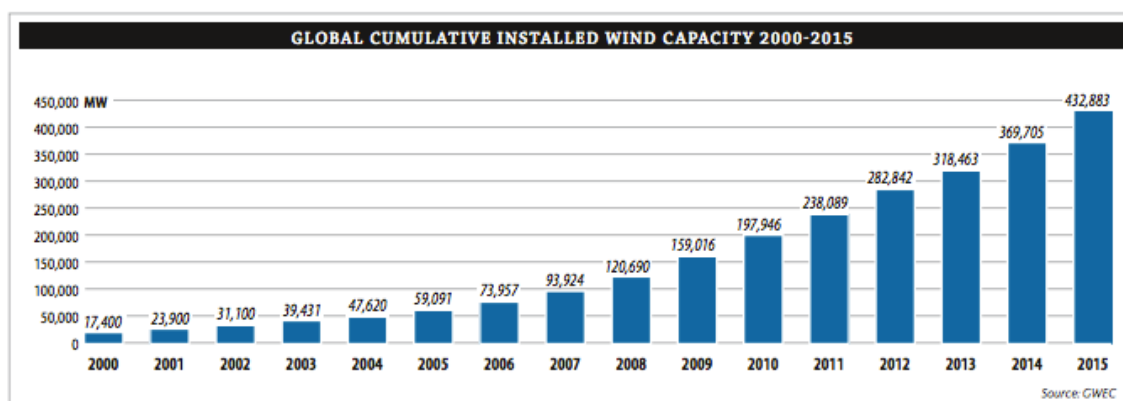


Figure 1.2. Worldwide installed capacity since 2000 to 2015. (GWEC, 2015)

At the end of 2015, the European Union had 142 GW of installed wind power plants, 131 GW of them was onshore and 11 GW was offshore. However, this tendency is changing, as the onshore market is decreasing,

7,8% in 2015, and offshore is increasing. In 2015 new offshore wind installation more than doubled the installations in 2014 (GWEC, 2015), with 3034,5 MW of new gross capacity grid-connected. Offshore installations represented 24% of the annual EU wind energy market, up from 13% in 2014 (Ewea, 2016).

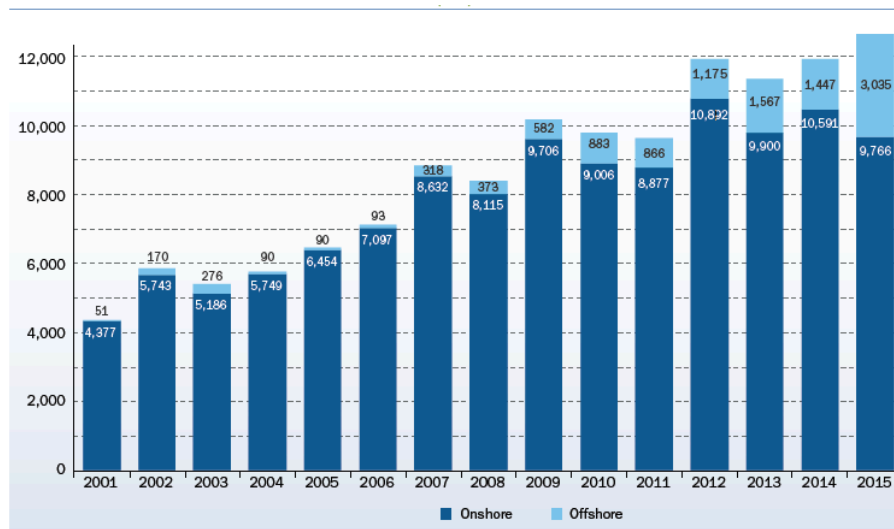


Figure 1.3. Annual onshore and offshore installations. (Ewea, 2016)

The reason of this changing tendency is that offshore wind farms present some benefits compared to onshore wind farms. The availability of higher wind speeds, as well the direction and speed are more constant due to the non-presence of obstacle decreasing the turbulence of the wind. The size of the onshore wind farms is limited by the ability to transport the large structure of the turbines. On offshore wind farm this problem is avoided, and wind turbine already exceed 5 MW and may eventually exceed 10 MW in the future (Snyder & Kaiser, 2009). Finally, the acoustic, environmental and visual impacts are reduced.

Although these positive aspects, there are also some negative ones. Firstly, the construction and maintenance of the wind turbine is more expensive than onshore. The distance to shore influences a lot in this aspect. During construction ships will have to make several trips between the wind turbine and the land. Every travel is costly and therefore the closer an offshore site is to the land, the more economical will be. The support structures or foundations needed for offshore wind farms are very expensive and, in some cases, not viable for deep seas.

In relation of this last handicap for offshore wind farms, a growing tendency of installing wind turbines in shallow seas has appeared recently. A very convenient one is the Baltic Sea, especially the North Sea where many countries like Belgium, Denmark, Germany and the United Kingdom have installed offshore wind farms.

1.4 HVDC transmission

The High voltage direct current (HVDC) technology first made its mark in the early undersea 96 km cable interconnection between Gotland (1954) and Sardinia (1967). Then, in long distances transmission with the Pacific Intertie (1970) and Nelson River (1973) schemes using mercury arc valves. Many mercury arc based transmission systems were to follow all around the world. The evolution of power electronics and semiconductors caused, at the late 1960s, the introduction of thyristor based valve technology. The installation, which marked this technology and replaced the mercury arc valves, was the interconnection at Eel River between Quebec and New Brunswick in 1972. Nowadays the majority of the HVDC transmissions systems in the world use the thyristor based line commutated current source converter (CSC) technology. An example is one of the largest transmissions, the Three Gorges – Shanghai with a rating of 3000 MW and ± 500 kV.

The reasons for choosing HVDC systems are generally economic not technical. Other reasons may be power system stability improvements and environmental circumstances.

The energy resources are generally located in places far from where the energy is consumed. In the case of offshore wind farms, is obvious that the distance is significant in the majority of the cases. This fact is important for the HVDC transmission systems; the longer the lines are, the more convenient the HVDC transmission line is compared to the AC systems.

A very important aspect of HVDC transmission lines is the cost of the infrastructure and operational requirements such as transmission losses. If it is assumed that the insulation requirements for peak voltage level for both AC and DC lines are similar, a DC line can carry as much power as an AC line. Obviously, the DC line having fewer conductors than the AC line, two conductors for the DC and three conductors for the AC. Hence, a DC line requires simpler and cheaper infrastructure.

It is worth considering the minimization of power losses in DC, as there are fewer conductors. In the case of two conductors in the DC line and three conductors in the AC line, the transmission losses are reduced to about two-thirds compared to the AC system (Sood, 2007).

The absence of skin effect in DC is also beneficial as the power losses and dielectric losses are reduced. Also, the Corona effect tends to be less significant on DC conductors.

When designing an AC transmission line over long distances it has the inconvenient of the significant inductance the line will have. The effects of the inductance have to be compensated along the line and this adds costs for long distances. For the DC lines there is no inductance as the frequency is zero. This makes the DC line to be less costly per kilometre. However, the HVDC converter stations at both end of the transmission line are more

expensive than the equivalent AC terminals. The following figure shows the total cost for AC and DC, and the influence on the transmission distance (Setreus & Bertling, 2008).

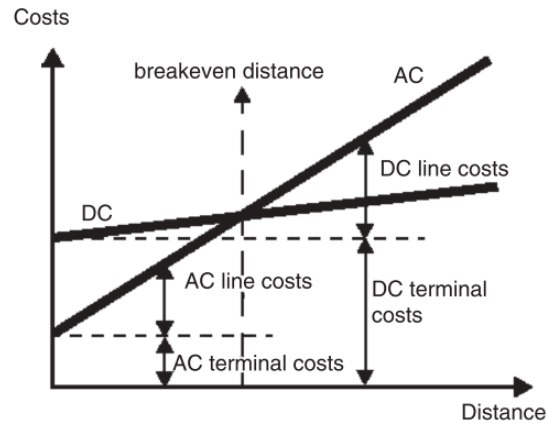


Figure 1.4. Comparison of AC and DC transmission system costs. (Sood, 2007)

It is obvious, looking at the previous figure that for distances less than the "breakeven distance" is more convenient, in economic terms, to use DC. This is due to the combination of the terminal equipment costs and line costs of the two types of transmission. The breakeven distances can vary from 400 to 700 km in overhead lines and from 25 to 50 km for a cable system.

The basic concept of the transmission HVDC connections can be seen in Figure 1.5. The objective is to connect two AC terminals with a DC line. The first terminal converts the alternating current into direct current and injects it into the transmission line. When the second terminal receives the direct current does the opposite action, inverts the direct current into alternating current.

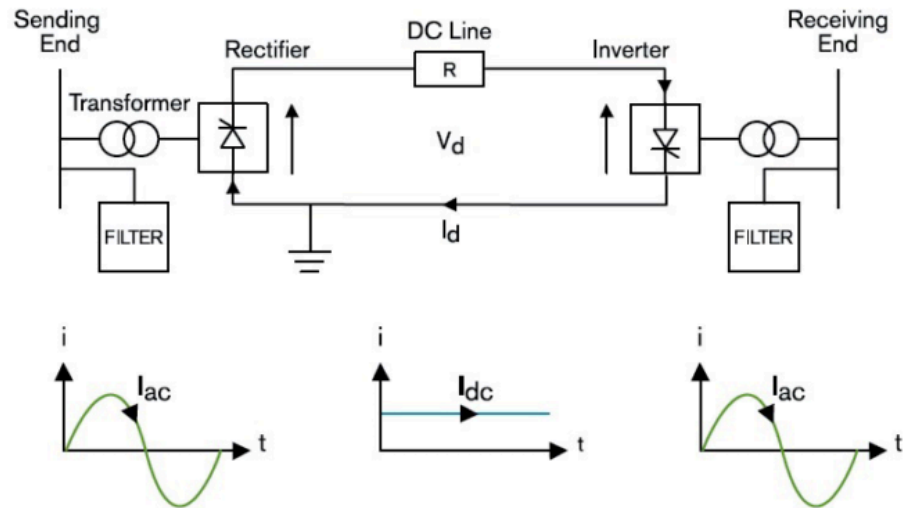


Figure 1.5. Basic scheme of an interconnection between AC terminals using HVDC transmission. (Feltz, 2016)

CHAPTER 2: MODEL OF AN HVDC NETWORK CONNECTING OFFSHORE FARMS WITH THE MAINLAND

2.1 HVDC network mathematical model

A HVDC network can be represented by an electrical circuit, in which the VSC converters are modelled as a current source in parallel with a capacitor, see Figure 2.1. The transmission line can be considered as having resistive, capacitive and inductive elements as can be seen in Figure 2.2. The nodes injecting current to the DC grid have positive values and the ones demanding power have negative values.

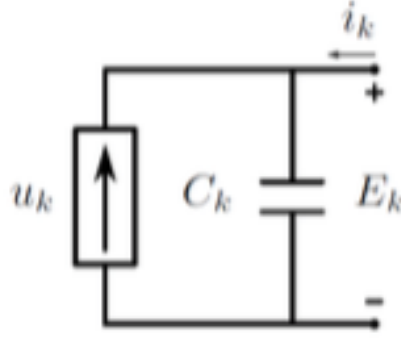


Figure 2.1. Equivalent circuit of a VSC.

The dynamics of both VSCs and DC meshed transmission lines are obtained from Kirchhoff Current Law (KCL) and Kirchhoff Voltage Law (KVL) (Dòria-Cerezo et al. 2015). The equation that models a VSC is given by

$$C_k = \frac{dE_k}{dt} = u_k(E_k) + i_k \quad (1)$$

where E_k is the voltage across the capacitor C_k , and $u_k(E_k)$ is the current injected (or consumed) by the power converter.

It is important to focus on the capacitance at each node, C_k , because it can be treated as two different capacitors with a reactance value half of the line capacitance, located at both ends of the transmission line, see Figure 2.2. Then, the line capacitance is considered as part of the node capacitance, adding half of the value to each node.

$$C_k = C_k^{VSC} + \sum_{j=1}^j \frac{1}{2} C_{kl} \quad (2)$$

for j , number of lines.

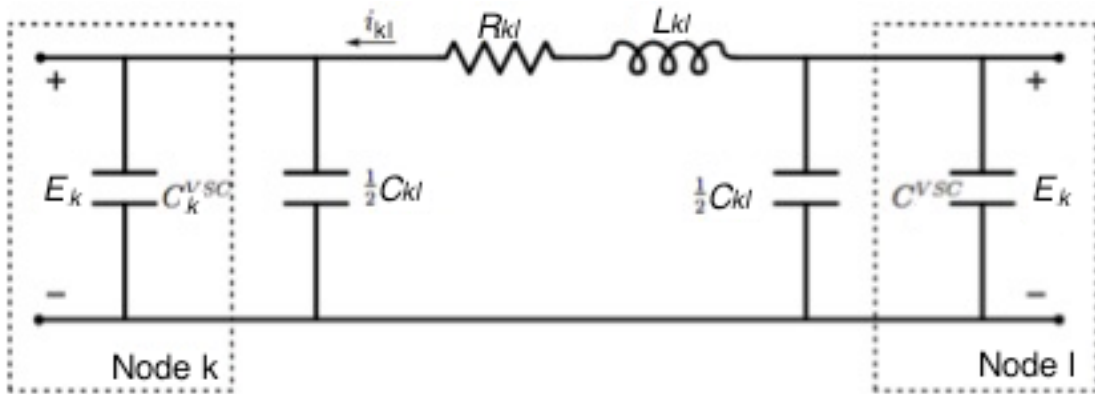


Figure 2.2. Equivalent circuit of a transmission line.

The incoming current into the capacitor k , i_k , can be described as the sum of all the currents flowing into the node k from the interconnected nodes

$$i_k = C_k^{VSC} + \sum_{l=1}^n b_{kl} i_{kl} C_{kl} \quad (3)$$

where i_{kl} is the current flowing from node l to node k , n is the number of nodes of the line, and b_{kl} are the elements of the incidence matrix B which indicates the connections of the network between nodes.

$$b_{kl} \begin{cases} 1 & \text{if } l \text{ and } k \text{ are connected,} \\ 0 & \text{if } l \text{ and } k \text{ are not connected.} \end{cases} \quad (4)$$

As it is considered that the capacitance is part of the node, the behaviour of the lines is simplified. Applying the KVL the dynamics of a transmission line connecting nodes k and l are given by

$$E_l = E_k + R_{kl} i_{kl} + L_{kl} \frac{di_{kl}}{dt} \quad (5)$$

where R_k and L_{kl} are the resistance and inductance of each line, respectively. Notice that $i_{kl} = -i_{lk}$.

Letting $j \in \mathbb{N}$ stand for the total number of transmission lines, the line current vector is defined as

$$i = i_{kl} \in \mathbb{R}^j \quad (6)$$

where $k, l = 1, \dots, n$ with $k < l$ and $b_{kl} \neq 0$. From where it is obvious that $j = \frac{n(n-1)}{2}$. Then the VSC dynamics can be expressed in matrix form as

$$C \frac{dE}{dt} = -B i + u(E) \quad (7)$$

where $E = (E_i) \in \mathbb{R}^n$, $u = (u_i) \in \mathbb{R}^n$, $C = \text{diag}(C_i) \in \mathbb{R}^{n \times n}$, and $B = (B_k) \in \mathbb{R}^n$ is the incidence matrix

$$B_{kj} \begin{cases} 1 & \text{if line } j \text{ connects from node } k, \\ -1 & \text{if line } j \text{ connects to node } k, \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

Similarly, it is possible to present a matrix description of the overall transmission lines dynamics. Defining the $\mathbb{R}^{j \times j}$ inductance and resistance matrices $L = \text{diag}(L_{kl})$, $R = \text{diag}(R_{kl})$, respectively, the matrix description is represented as

$$L \frac{di}{dt} = -R i + B^T E \quad (9)$$

Hence, the overall HVDC system dynamics is given by

$$\frac{d}{dt} \begin{pmatrix} E \\ i \end{pmatrix} = \begin{pmatrix} \mathbb{O}_n & -\mathbf{C}^{-1}\mathbf{B} \\ -\mathbf{L}^{-1}\mathbf{B}^\top & -\mathbf{L}^{-1}\mathbf{R} \end{pmatrix} \begin{pmatrix} E \\ i \end{pmatrix} + \begin{pmatrix} \mathbf{C}^{-1} \\ \mathbb{O}_{j \times n} \end{pmatrix} u, \quad (10)$$

with \mathbb{O}_n and $\mathbb{O}_{m \times n}$ denoting $n \times n$ and $j \times j$ null matrices, respectively.

2.2 Wind energy mathematical model

When modelling multiterminal HVDC transmission scheme for offshore wind farms it is essential to take into account the characteristics of the wind. These include speed and direction, which are affected by several factors; like geographic location, height above ground, surface topography and climate characteristics (Bianchi, et al. 2007). All these circumstances lead to wind turbulences, which are the responsible of variations in the speed and direction of the wind.

These turbulences are characterized by chaotic random property changes, which deviate randomly from the mean wind speed. By definition, turbulence includes all wind speed fluctuations with frequencies above the spectral gaps. Therefore, in order to model the wind with accuracy, it is not enough to know the mean annual wind speed for a location.

Thus, it is needed to study experimentally the wind speed by placing anemometers to recollect data, usually over a year. The wind distribution experimentally obtained can be approximated by a Weibull distribution.

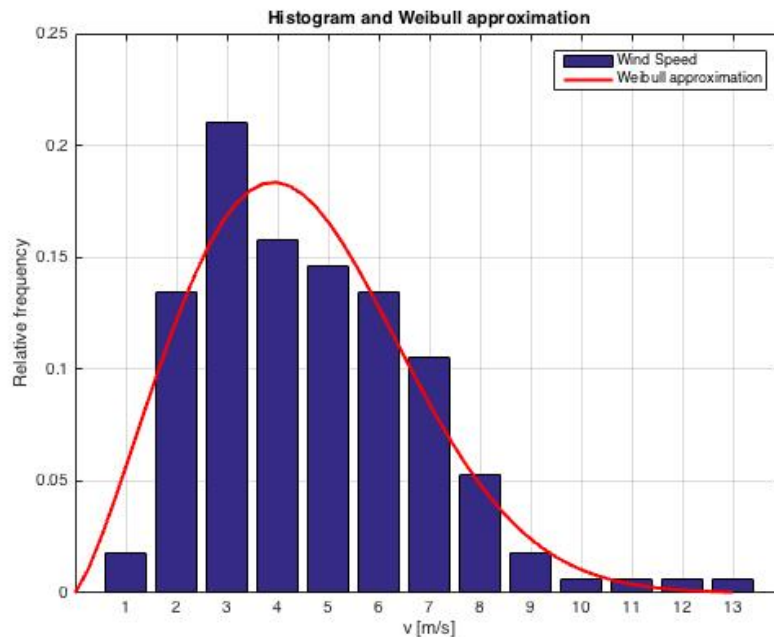


Figure 2.3. Cumulative relative wind speed (blue) and its Weibull approximation (red).

As it can be seen in Figure 2.3 this Weibull distribution shows if, the wind is measured throughout a year, the moderate wind speeds are more frequent whereas large wind speeds rarely occur.

The cumulative Weibull distribution function is defined as

$$\phi = 1 - e^{-\left(\frac{v}{\alpha}\right)^k} \quad (11)$$

where ϕ is the Weibull distribution function, α defines the scaling function, v the wind speed and finally k defines the form parameter (for most wind distributions it ranges from $1.5 \leq k \leq 2.5$).

The inverse of this accumulative function, which allows obtaining the wind speed, is

$$v = \alpha \ln\left(-\frac{1}{\phi_c - 1}\right)^{\frac{1}{k}} \quad (12)$$

The power in the wind passing through an area A with a speed v and the air density ρ , is defined by

$$P_v = \frac{1}{2} \rho A v^3 \quad (13)$$

This can be derived from the kinetic energy ($E_k = \frac{1}{2} \rho v^2$) multiplied by the flow rate (Av). This is the available power in the wind, however it does not mean that a wind turbine can extract all this energy from the wind.

In order to obtain the total extracted power from the wind turbine it is needed to consider a coefficient, known as the Betz's power coefficient. It is a dimensionless measure of the efficiency of a wind turbine in extracting the energy content of a wind stream. The Betz limit represents the most optimal case with a value of $16/27$, which is the maximum power any turbine can extract.

Therefore, the total extracted power is given by

$$P_v = \frac{1}{2} C_p(\lambda, \beta) \cdot \rho A v^3 \quad (14)$$

As it can be seen the C_p value, which theoretically corresponds to the Betz coefficient, any real turbine cannot achieve this value. Therefore, when applying it in the model it depends on the pitch angle (β) and the tip-speed-ratio (λ). This speed ratio can be defined as

$$\lambda = \frac{\omega R}{v} \quad (15)$$

where ω defines the angular speed of the turbine and R the rotor radius. This tip-speed-ratio is extremely important because, together with the pitch angle, determines the operating condition of a wind turbine.

Offshore wind farms, which use large wind turbines, are equipped with control mechanisms that allow controlling these parameters. By using

control strategies over the rotation speed and the pitch angle, the power extracted from the wind in the zone between the cut-in v^{in} and cut-off wind speeds v^{off} , can be maximized, see Figure 2.4.

This area defined by the cut-in and the cut-off wind speeds is defined for the minimum speed and maximum speed to generate some power. The cut-in speed is the minimum wind velocity needed for the turbine to turn on; beyond it the available wind energy is too low to compensate the operation costs and losses. Below this speed the turbine is turned off until the wind speed reaches this value. By the other hand, when the cut-off wind speed is reached, the turbine is forced to turn off, due to a too high wind speed that could cause the generator and various mechanical components to be overloaded or cause damage in the turbine.

The ideal power curve for a typical wind turbine is delimited by this two parameters (v^{in}, v^{off}). Nevertheless, there is the need to establish another limitation, the rated power (v^{max}). As the wind speed rises above the cut-in speed, the electrical power generated rises rapidly, reaching the limit that the electrical power generator is capable of. At higher wind speeds, the turbine is arranged, by adjusting the blade angles, to limit the power to this maximum level, keeping it constant.

The effect of the varying power coefficient can be expressed using the piecewise function (Hossain & Mahmud, 2014):

$$f(x) = \begin{cases} 0 & \text{if } v \leq v^{in} \\ \frac{1}{2} C_p^{max} \cdot \rho A v^3 & \text{if } v^{in} < v < v^{max} \\ P^{max} & \text{if } v^{max} < v < v^{off} \\ 0 & \text{if } v \geq v^{off} \end{cases} \quad (16)$$

The representation of this function is shown in the following figure.

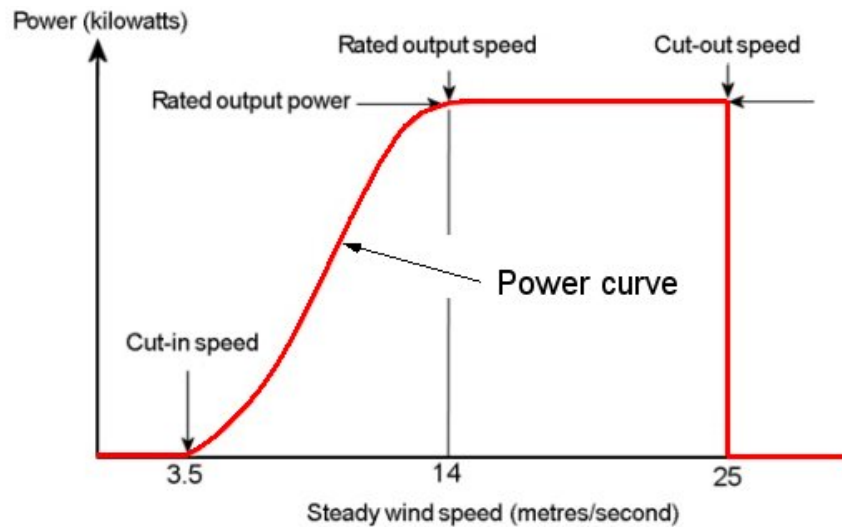


Figure 2.4. Wind turbine power curve. Image obtained from <http://www.wind-power-program.com/popups/powercurve.htm>

CHAPTER 3: CONTROL AND SUPERVISION OF HVDC POWER GRIDS

3.1 Control scheme

The operation and control of HVDC transmission systems can be described as three control levels, see Figure 3.1. The hierarchical scheme shown in the figure consist of a secondary control, a supervisor algorithm that sets the required voltage in all the VSC, E_k^* , a primary control that regulates the voltages of each VSC capacitor, and a current controller that provides a switching policy to inject or extract the required current, u_k , to/from the capacitor, C_k . This current is obtained from the primary control. In the current work, the control loop on the current is considered as ideal. Note that the controller is based on both a centralized (the secondary control) and a decentralized control (the primary control).

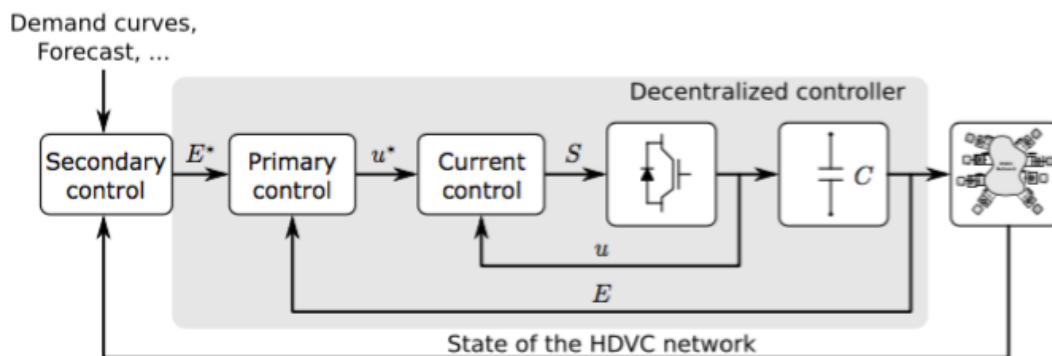


Figure 3.1. Hierarchical control scheme.

The purpose of the secondary control is to establish and communicate the required voltages of the network to achieve an optimal power flow. This optimal power flow is based on the losses of the system, and the optimal will be achieved when the losses are minimized. This part will be explained more in detail in Section 3.3.

On the other hand, the primary control regulates the voltage in each node using decentralized control. This type of control has to guarantee the stability of the system without communication.

3.2 Decentralized voltage controller

When a decentralized control is implemented, one of the most frequently used strategies for the VSC in an HVDC network is the droop voltage control. It is based on a static relationship between voltage in the DC capacitor, E_k , and the current injected in the VSC, u_k . Basically, it maintains maximum power for a certain range of E_k , and switches to proportional gain when required.

During the operation the resulting set point must remain within the admissibility region. The area within VSC must be working is the one bounded by the current, power and voltage limitation, as seen in Figure 3.2.

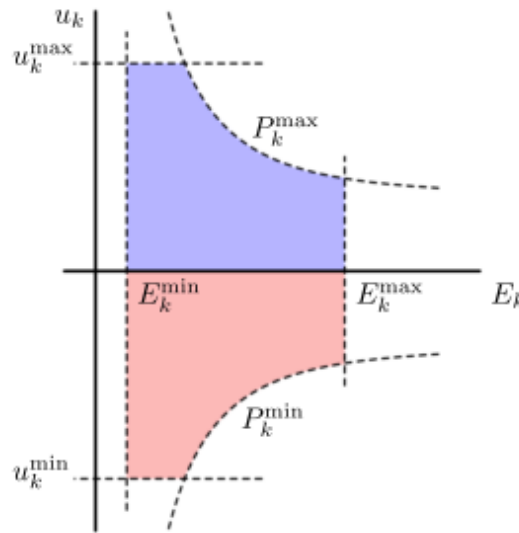


Figure 3.2. Admissibility region: power injection (blue); power consumption (red).
(Dòria-Cerezo et al. 2015)

The droop control action in each agent can be generalised as (Dòria 2014)

$$u_k = \begin{cases} u_k^{max} & \text{if } E_k \leq P_{kl} u_k^{max} \\ \frac{E_k}{P_{kl}} & \text{if } P_{kl} u_k^{max} < E_k < E_k^l \\ -m_k^d (E_k - E_k^*) & \text{if } E_k^l < E_k < E_k^h \\ \frac{E_k}{P_{kc}} & \text{if } E_k^l < E_k \end{cases} \quad (17)$$

where $P_{kl} > 0$ is the power generated by a WF and $P_{kc} < 0$ is the maximum power that can be consumed by a GS node. The slope of the droop control is denoted by m_k^d , and E_k^* is the voltage for zero injected current. The low and high values of the droop region, E_k^l and E_k^h , can be obtained as

$$E_k^{l,h} = \frac{1}{2} \left(\left| E_k^{*2} - \frac{4P_{kl,kc}}{m_k} \right| \right) \quad (18)$$

The curve (17) is characterised in Figure 3.3. Note that for the case of WF power plants the consumed power is zero. Therefore, in WF converters $P_{kc} = 0$ which, considering (18), implies $E_k^h = E_k^*$.

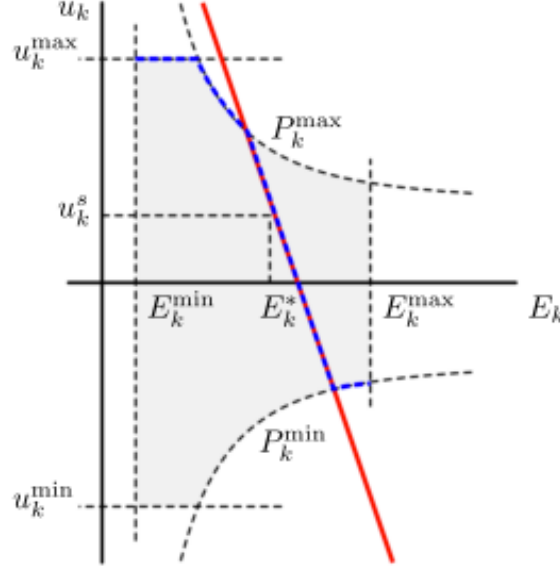


Figure 3.3. Droop control curves for a wind farm converter (blue) and grid station (red).

3.3 Secondary control: Optimal power flow

In the present case, it is proposed an operating point calculator based on an optimal power flow algorithm that will try to always bring the network to the minimal transmission losses (Gavrilita, 2014). As it is known, power losses on transmissions lines in DC are just produced by resistive effect. There are

two main ways to minimize these losses, transporting the power over relatively shorter distances or at high voltage. The reason is due to the Joule effect. The power in DC can be expressed as $P=E \cdot u$ and the Ohm law as $E=u \cdot R$, it is possible to obtain the losses in terms of currents and resistance.

$$P_{loss} = R \cdot u^2 \quad (19)$$

When implementing for the totality of the nodes, the expression of this cost function is given as

$$P_{loss} = \sum_{k=1}^n \sum_{l=1}^n g_{kl} \cdot (E_k - E_l)^2 \quad (20)$$

where E_k and E_l are the voltages in node k and l , respectively, and g_{kl} is the conductance between the nodes k and l . The same relationship can be expressed in matrix form as shown above

$$P_{loss} = E^T \cdot \mathbf{G} \cdot E \quad (21)$$

where $E = [E_1, E_2, \dots, E_n]^T$ and \mathbf{G} is the conductance matrix. This cost function needs to be minimized considering the equality constraints. From the dynamics of the HVDC network, which are describe in Section 2.1, it is possible to find the equilibrium points of the network given by (Benedito et al. 2016)

$$0 = -\mathbf{B}i + uE \quad (22)$$

$$0 = -\mathbf{B}^T E - Ri \quad (23)$$

which yields the following relationship

$$u = \mathbf{B}R^{-1}\mathbf{B}^T E = \mathbf{G}E \quad (24)$$

The objective function is subject to inequality constraints. As mentioned in the previous section, the nodes of the grid should be working on the admissibility area, Figure 3.1. This zone is defined by the minimum and maximum current, power and voltage in each node.

$$E_{min,k} \leq E_k \leq E_{max,k} \quad (25)$$

$$u_{min,k} \leq u_k \leq u_{max,k} \quad (26)$$

$$P_{min,k} \leq P_k \leq P_{max,k} \quad (27)$$

It is possible to use these inequality constraints as equality constraints by setting the minimum and maximum bound equal to each other. An example would be that a node needs to work in a specific voltage and the minimum and maximum bound have the same value, $E_{min,k} = E_{max,k}$.

CHAPTER 4: SIMULATIONS

4.1 General description

In the previous chapters (chapters 2 and 3), the model of an HVDC network was presented, as well as the control and supervision of it. Once they are explained, they are implemented on MATLAB in order to simulate and study the behaviour of the system. To obtain the desired MATLAB program that allow obtaining the simulations to achieve the aim of this project, two different previous works have been used.

Firstly, the work presented in (Grau, 2015), which consist of a tool capable of simulating an HVDC network, as well as to study the different numerical methods and optimisations that can be applied to it. From the point of view of the present work, the simulation achieved in (Grau, 2015) represent a unique simulation where the OPF is not integrated.

As defined in the document the final program could be divided in different steps. The first one is to read all the necessary parameters that are needed to simulate the model of the HVDC network: number of nodes, connectivity of the transmission lines, resistances, inductances, capacitances, Weibull parameters, nominal voltages, etc. All these parameters are presented in 4.2. Secondly, it is necessary to execute some pre-simulation operations, such as the computation of the necessary matrices for the simulation, validate the data previously obtained or calculate the wind profile. After that, solve the system of Ordinary Differential Equations (ODEs) to obtain the results of the simulation. Finally, plot the variables of interest and calculate some other variables of interest on the previous step and represent them.



Figure 4.1. Scheme of the algorithm applied in (Grau, 2015).

The other work of interest is presented in (Paulet, 2016). It implements several algorithms that optimise the simulation results of an HVDC network control algorithm and makes it an automatic process. The part used in the present work is the Optimal Power Flow algorithm. As explained in the previous chapter, it consists of a minimization of the transmission losses produced by the Joule effect.

Once explained what the program is based on, it is time to explain the idea of the current work. The main idea of the project is to create an algorithm capable of linking two or more simulations. Each of these simulations is independent from the others, which means that every one has to solve its own system of ODEs. The difference between them is the conditions applied in every simulation.

The schematic representation shown above is the main idea of the process followed in the simulation part of the program.

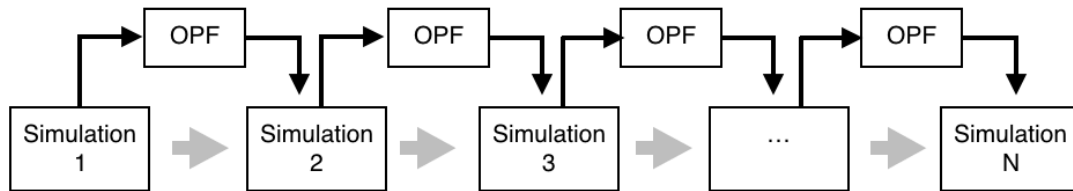


Figure 4.2. Idea of the process applied in the simulation part.

First of all, it is needed to gather all the necessary information and parameters that the program uses to simulate the HVDC network. This step is basic in the program as all the data has to be correctly introduced. To make sure this step is properly completed an Excel file is provided to enter the parameters used in the program.

Once this first step is completed, the program can proceed to compute the necessary matrices to begin with the simulating part and solving the ODE system. By the other hand, it creates the wind profile and the power that can be extracted from it. The implementation on Matlab is based on parameters α , k and the *rated wind speed*. The wind energy model is explained in Chapter 2.2 of this document.

It is important to take into account that the wind profile needed to perform all the different simulations has to be a continuous profile during all the simulation.

After these two first steps the simulations can be started. As it was said some paragraphs above, the idea is to link two or more consecutive simulations. In order to achieve this, it is necessary to repeat one simulation as many times as it is desired. However it is not that simple, some changes have to be applied in order to perform properly. The most important one is to introduce the correct initial conditions of each simulation to the system of ODEs. Once computed the first simulation, the following ones have to take the final conditions of the previous simulation and introduce them as the initial conditions for the following one.

It can be seen in Figure 4.2 that in every simulation it is calculated the Optimal Power Flow (OPF), which considers the conditions of the wind and the restrictions of each node. In each simulation the wind conditions and the voltage to optimise change, however, the other bounds for the admissibility area are the same during all the simulations. The admissibility zone is defined by the minimum and maximum bounds of current, power and voltage. The power bounds can be defined by this area where the optimiser can work or by the available power in the wind, depending on which has the absolute minimum. In this case, the minimum of these two powers is the one used to work as the admissibility region, since if the power available in the wind is lower than the bound fixed by the admissibility area, it will not work.

4.2 Code

4.2.1 Excel file

All the information required by the simulation of the network will be presented in this section. As previously commented, an Excel file is provided to cover all the necessary parameters. This file is called "*HVDC_data.xls*". It is divided in three different sheets, which are called *Lines*, *Nodes* and *Other*. It is essential not to modify the position of the columns, since each one define a given parameter. If it is wanted to modify the parameters of the simulation, it is only allowed to modify the parameters that each column or row contain, but not the position.

The first sheet contains the parameters of the different transmission power lines; each row of the file is a different power line. It can be seen above the meaning of all these parameters.

- N_s . Node where the line starts.
- N_e . Node where the line ends.
- $Length [km]$. Length of the transmission power line.
- $R [\Omega/km]$. Resistance per km.

- $R [\Omega]$. Total resistance of the transmission power line.
- $L [\text{mH/km}]$. Inductance per km.
- $L [\text{mH}]$. Total inductance of the transmission power line.
- $C [\mu\text{F/km}]$. Capacitance per km.
- $C [\mu\text{F/km}]$. Total capacitance of the transmission power line.
- $iniL [\text{kA}]$. Initial conditions for the transmission power line current.

The second excel sheet is the one called *Nodes*, it contains all the information related to the nodes. In this case, each row represents a different node, except the header row, where the names of the parameters are defined.

- N . Number of nodes.
- $iniC [\text{kV}]$. Initial voltage on the capacitors
- $C [\mu\text{F}]$. Capacitance of the node.
- $I_{max} [\text{kA}]$. Maximum current that the node can provide.
- $E_0 [\text{kV}]$. Nominal voltage for each node; value in which the droop control is focused. This value only works for the first simulation, from the second simulation until the end, this value is given by the optimiser (Optimal Power Flow).
- $md [\text{A/V}]$. Slope of the drop control.
- $Pg [\text{MW}]$. Maximum generated power for each node.
- $\tau [\text{s}]$. Time constants that describe the closed-loop dynamics of the current conversion.
- $\alpha [\text{m/s}]$. Scaling factor of the Weibull distribution.
- k . Parameter of the Weibull distribution.
- *Wind speed* $[\text{m/s}]$. Minimum wind speed the wind farm can generate power.

Note that the parameters referring to the wind model (α , k and *Wind speed*) only need to be filled for wind farm nodes.

The following parameters refer to the optimiser of the Optimal Power Flow. The first five parameters (columns) define the bounds in which the optimiser can work. The other two are coefficients for the determination of the wind power bound. These parameters are also included in the second sheet of the Excel file.

- P_{Max} [MW]. Maximum power admitted by the node, if the node is a wind farm, so generates power, the value is positive; otherwise, if it consumes power, it is negative.
- P_{Min} [MW]. Minimum power admitted by the node, as in the above parameter, if the node is a wind farm the value is positive, otherwise is negative. It is important to take into account that following relation, $P_{Min} < P_{Max}$, it is always needed to be fulfilled.
- I_{Min} [kA]. Minimum bound current of the node, if the node is a wind farm the value $I_{Min} \geq 0$; otherwise, as it consumes power, it is negative, $I_{Min} < 0$.
- E_{Max} [MW]. Maximum voltage bound of the node.
- E_{Min} [MW]. Minimum voltage bound of the node. Note that the relation $E_{Min} < E_{Max}$ needs to be fulfilled.
- g_{Max} . Coefficient used to decide the maximum wind power bound. It will be explained in Section 4.2.4
- g_{Min} . Coefficient used to decide the minimum wind power bound. It will be explained in Section 4.2.4

Finally, the *Other* sheet contains the information and the parameters needed for the simulation itself.

- *Wind Sample Time (s)*. Represents the length of the interval in which the wind speed changes.
- *Wind changes*. Defines whether the wind energy conversion works or not. If the value of the cell is zero, the model works as if every wind farm generates constant power with the value defined by P_g .
- *Simulation Start Time (s)*. Beginning of the simulation and the wind profile. It is suggested to start with negative values to avoid the "bad-performance" of the model, in order to ensure that at time $t=0s$ the system will have been properly stabilised.
- *Wind End Time (s)*. End of the wind profile.
- *Step Size*. Step size for a fixed-step ODE solver.
- *Absolut and Relative Tolerance*. Required accuracy of the solution for adaptive step size.
- *Number of Simulations*. Number of iterations for the total of the simulation.
- *Time of the simulations (s)*. It represents the duration of each simulation.

4.2.2 Linking the different simulations

Linking the different simulations is an essential point in order that the simulation works properly. Therefore, in this section it is going to be presented the way to link the simulations and how to modify the parameters of the Excel file to simulate a desired case. As an example, a case in which three iterations of the simulation part is included. The parameters that control this part of the simulation are the ones defined in the Other sheet of the Excel file. The parameters needed are the following ones:

- *Time of the simulations (s).*
- *Simulation Start Time (s).*
- *Wind End Time (s).*
- *Number of Simulations.*

The parameter *Simulation Start Time* defines the beginning of the simulation, as well as the beginning of the wind profile. *Wind End Time* defines the ending time of the wind profile, as said in the previous section, it is important the starting time begins before $t = 0$ s. The *Number of simulations* represents the number of iterations of the simulation part. Finally, the duration of every iteration is defined with the parameter *Time of the simulation*, since all the iterations have the same duration.

To clarify the use of these parameters an example is presented. Three simulations are taken, with duration of 2 seconds each. The values of the parameters should be the ones shown above. However, it is not the only option to perform correctly.

- *Time of the simulations (s) = 2*
- *Number of Simulations = 3*
- *Simulation Start Time (s) = -1*
- *Wind End Time (s) = ≤ 5*

The first two parameters are obvious, since it is what was defined for the example, three iterations of 2 seconds each. The starting time is a negative value, $t = -1$ s, as it was said this is recommendable in order to give the simulation enough time to stabilize the system, though it could be another one such as $t = -2$ s. Note that if the time of one simulation is 2 seconds, when the first simulation arrives at $t = 0$ s, it will be in the middle of its duration. Then the first simulation will end at $t = 1$ s. In the case that it is desired to simulate one entire simulation beginning at $t = 0$ s, it will be needed to simulate a whole iteration in the negative part of the time.

Finally, the wind profile ending time has to be equal to the sum of the total time of the simulation. In this case it is equal to 6 seconds, which is the number of simulations multiplied by the time of each simulation. If the wind profile ending time is bigger than the duration of the total simulation the program also works properly, but never be smaller.

4.2.3 Initial conditions

In order to start with the simulation, the following initial conditions are needed: E_0 , $iniC$, $iniL$ and icu . The variable E_0 is the nominal voltage for each node and is defined as a column vector with same length as the number of nodes. As it was done with all the other parameters, it is introduced to the program through the Excel file. The three remaining variables are defined in Matlab inside a vector called $x0$.

```
x0=[icC; icL; icu];
```

The variable icC and icL represent the initial current of the transmission power line and the initial voltage on the capacitor, respectively. The icu represents the initial VSC current values, the dimension of this vector is the length of the number of lines. The dimensions of icC and icu is the length of the number of nodes for both.

The first two variables – icC and icL – are conditions introduced directly to the Excel file; the other one – icu – has been obtained from the following expressions.

```
satP=min(PgA(:,1), max(Pc, (-md.*(icC-E0)).*icC));  
icu=min(umax, max(umin, (satP)./icC));
```

The saturated power – $satP$ on the implementation in Matlab – is needed to calculate the variable icu . This variable compares the generated power available in the wind with the power that each node can absorb or the power of the node calculated with the droop control action depending on which one is bigger. As this value is an initial condition, it is used at the first instant of the generated power from the wind.

All these initial conditions are used for the first simulation, and they are required in order to execute the ODE solver. However, for the following iterations other conditions are going to be needed; in the consecutive simulations the initial conditions will be defined by the final conditions of the previous simulation. So these conditions are the ones obtained from the last point calculated by the ODE solver. This ODE solver returns a matrix called x with all the points calculated in one iteration, and this includes the final conditions of the simulation, which correspond to the new initial conditions

of the following simulation. This new initial conditions is the last column of the matrix. The length of it is the one corresponding to the number of points in which it is calculated.

The nominal voltage for each node in the new conditions, E_0 for the initial conditions, will be given by the optimiser (Optimal Power Flow), which calculates the optimal voltage for each node. This optimisation is explained in Section 3.3 and the implementation on Matlab is explained in next section of this chapter.

4.2.4 Optimal Power Flow

This section consists on explaining the implementation of the optimiser in the program. It is important to keep in mind that the main idea of the Optimal Power Flow is to minimize the losses on the transmission lines.

As seen in the Diagram 4.2 the Optimal Power Flow is optimised in every iteration of the simulation. So the conditions considered in each optimisation are different. In the current case these changing conditions are represented by the available power in the wind, as the other bounds are equal in every simulation.

The constraint that should meet the final solution is the admissibility zone bounded by the current, power and voltage. All these parameters are defined in the Excel file. Note that the power is defined both by the wind and the constraint.

In order to achieve the minimization of the objective function the *fmincon* Matlab function was used. This function finds a minimum of a constrained nonlinear multivariable function.

$$\min f(x) \text{ such that } \begin{cases} c(x) \leq 0 \\ ceq(x) = 0 \\ A \cdot x \leq b \\ Aeq \cdot x = beq \\ lb \leq x \leq ub, \end{cases} \quad (27)$$

In order to achieve a correct performance of the main function it is needed to define a vector containing all the required initial conditions. By the other hand, it is also needed to define the constraint functions, which are represented by $c(x)$ and $ceq(X)$, as well as the lower and upper bounds, defined as lb and ub , respectively. Finally, we define the objective function.

The objective function is defined by the equation (21). The V variable, representing the nominal voltage for each node, is the one that is going to be optimised. At the end of each optimisation process, the optimal E variable is stored in a vector with the size as the totality of nodes. Finally, it is also needed to define the conductance matrix, which will be defined as $AiRA$ in the experimental model.

The constraints that the *fmincon* has to tolerate are the maximum and minimum power and current bounds. The lower and upper bounds are the minimum and maximum voltage.

The *fmincon* function is introduced in Matlab using the following expression

```
[V,FVAL,EXITFLAG,OUTPUT] = fmincon(@OPFof,V0,[],[],[],[],lb,ub,
'OPFnlc',options1);
```

The objective function defined in the Matlab program corresponds to the file *@OPFof*

```
function Ploss = OPFof(V)
global AiRA
Ploss=V'*AiRA*V;
```

where *V* is the variable to be optimised and stores the results of the optimisation process. Each iteration of the simulation this variable adopts different values.

The initial conditions of the optimisation process are the ones defined in the vector *V0*. Note that in every new iteration this vector is the same as the previous optimised vector *V*. The lower and upper bounds of the function are the minimum and maximum voltage. In the program they are represented with a vector with size of the totality of nodes.

The current and power constraints are introduced into the *fmincon* function through function *c(x)*. This function represents inequality functions, equal or lower than zero. The *ceq(x)* function is not used because it refers to an equality, equal to zero, and this is not the aim of the current and power bounds. The purpose of these two constraints is to limit a value of the variables but not to fix it.

For the power constraints

$$(P_1, \dots, P_n) - (P_{max_1}, \dots, P_{max_n}) \leq 0 \quad (28)$$

$$-(P_1, \dots, P_n) + (P_{min_1}, \dots, P_{min_n}) \leq 0 \quad (29)$$

With

$$(P_1, \dots, P_n) = (V_1, \dots, V_n) \cdot (I_1, \dots, I_n) \quad (30)$$

For the current constraints

$$(I_1, \dots, I_n) - (I_{max_1}, \dots, I_{max_n}) \leq 0 \quad (31)$$

$$-(I_1, \dots, I_n) + (I_{min_1}, \dots, I_{min_n}) \leq 0 \quad (32)$$

With (I_1, \dots, I_n) as defined in (24)

$$(I_1, \dots, I_n) = G_{n \times l}(V_1, \dots, V_n) \quad (30)$$

As it was previously mentioned the power constraint can be defined in two different ways, by the node admissibility bound or by the power available in the wind. Obviously, this only affects the nodes that are considered wind farms.

For the choice of the maximum power bound it is needed to make a comparison between the power available in the wind and the maximum power bound of the admissibility zone of the node. The chosen value is the minimum of these two. The wind values chosen to make this comparison are the ones corresponding to the last calculated point of the current iteration, in other words, the last instant of the iteration. The implementation of this comparison is applied in Matlab as the following expression.

```
Pmax = min(PmaxInitial, PgA(:, (length(tspan)-1)*i).*gmax);
```

where *PmaxInitial* is the maximum bound of the admissibility zone and *PgA* is the wind generated power.

The idea of the minimum power bound is the same, however, the implementation on the program is quite different. To be able to choose the correct power bounds the following expressions were used. The difference from the maximum power bound is that the minimum bound for the wind farm nodes take the maximum value of the comparison. The consumer nodes make the same comparison as the maximum power bound, taking the minimum value of the comparison.

```
for k=1:nN
    if isWF(k)==1
        pmin1(k) = max(PminInicial(k), PgA(k, (length(tspan)-1)*i).*gmin(k));
    else
        pmin0(k) = min(PminInicial(k), PgA(k, (length(tspan)-1)*i).*gmin(k));
    end
```

```
end
```

```
Pmin=pmin1+pmin0;
```

As it can be seen in both cases, for the minimum and maximum power bounds, there is a coefficient multiplying the generated power from the wind. In the minimum power bound is called *gmin* and in the maximum power bound *gmax*. They represent the power bounds of the generated wind power and the values that they can take range from 0 to 1, being always $gmin \leq gmax$.

4.3 Simulations results

To analyse the behaviour of the model presented in Sections 2 and 3 a HVDC network is simulated with Matlab. The scheme shown above is an approximation of the distribution of the network.

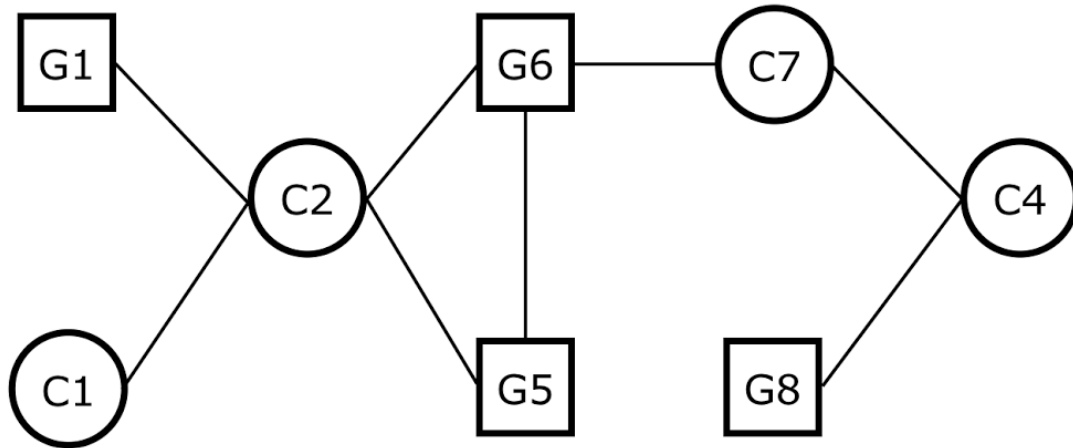


Figure 4.3. Approximation of the interconnexion of the study case.

The following tables show the data that define the different parameters of the network. The first table shows the parameters of the different transmission. The second one shows the values of the different nodes. Note that the wind farm nodes are the only ones who have parameters related to the wind model conversion, which are α , k and *Wind speed*. Finally, a table defining the admissibility zone in which the optimiser, the Optimal Power Flow, should work.

Table 4.1. Transmission lines in case study.

Line	Ns	Ne	R [Ohm]	L [mH]	Cl (μ F/km)	iniL [kA]
1	1	2	8	764	8,8	0,5
2	2	3	6	573	6,6	0,6
3	2	5	20	1910	22	-0,4
4	2	6	60	5730	66	-0,1
5	4	7	35	3342,5	38,5	-0,1
6	4	8	30	2865	33	-0,3
7	5	6	10	955	11	0,1
8	6	7	6	573	6,6	0,6

Table 4.2. Nodes in case study.

Node	iniC [kV]	C [μ F]	I _{Max} [kA]	E ₀ [kV]	md	Pg [MW]	τ [ms]	α	k	Wind speed [m/s]
1	400	75	1	400	1,9	500	1	6,4	2,1	9
2	400	75	1,5	417	1,8	500	1	-	-	-
3	400	75	1,3	409	1,6	500	1	-	-	-
4	400	75	1,5	413	1,7	500	1	-	-	-
5	400	75	1,2	400	1,7	500	1	6,8	2,2	10
6	400	75	1,4	405	1,8	500	1	6,4	2,3	9,4
7	400	75	1,5	400	1,9	500	1	-	-	-
8	400	75	1	410	1,8	500	1	6,3	2	9,6

Table 4.3. Parameters of the secondary control (OPF).

P _{maxN} [MW]	P _{minN} [MW]	I _{min} [kA]	E _{maxN} [kV]	E _{minN} [kV]	g _{max}	g _{min}
500	0	0	420	400	0,95	0,75
-100	-300	-1,5	420	400	0,92	0,79
-200	-300	-1,3	420	400	0,87	0,8
-200	-400	-1,5	420	400	0,91	0,79
350	0	0	420	400	0,93	0,77
500	0	0	420	400	0,96	0,8
-200	-600	-1,5	420	400	0,94	0,81
600	0	0	420	400	0,97	0,78

The simulation consists of four iterations with 10 seconds each one. The simulation start time is defined at $t = -10$ seconds, this means that the first iteration is not on the positive values of the time. Hence, the first simulation is not shown on the graphics of interest. Note that the Wind Sample Time is also 10 seconds. The following graphics show the results of the three remaining iterations.

The first graphic of interest is the voltage in every node of the network. It can be seen the different iterations as the voltage changes when the optimiser finds the new optimal voltage value. Note that the voltages of the nodes are maintained within the limits imposed, from 400 kV to 420 kV.

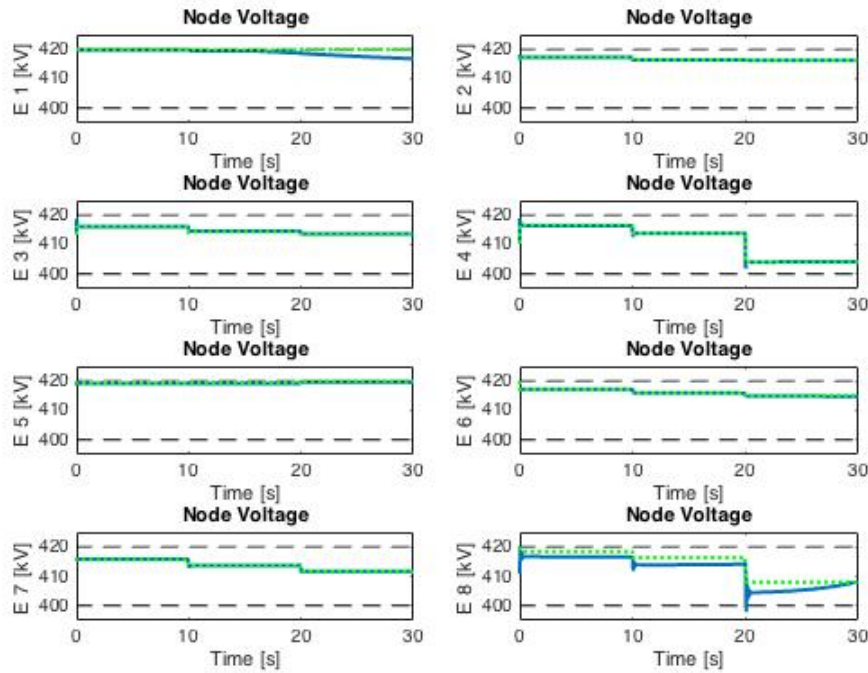


Figure 4.4. Evolution of VSC voltage in each node.

The following plot shows the current in every node of the network.

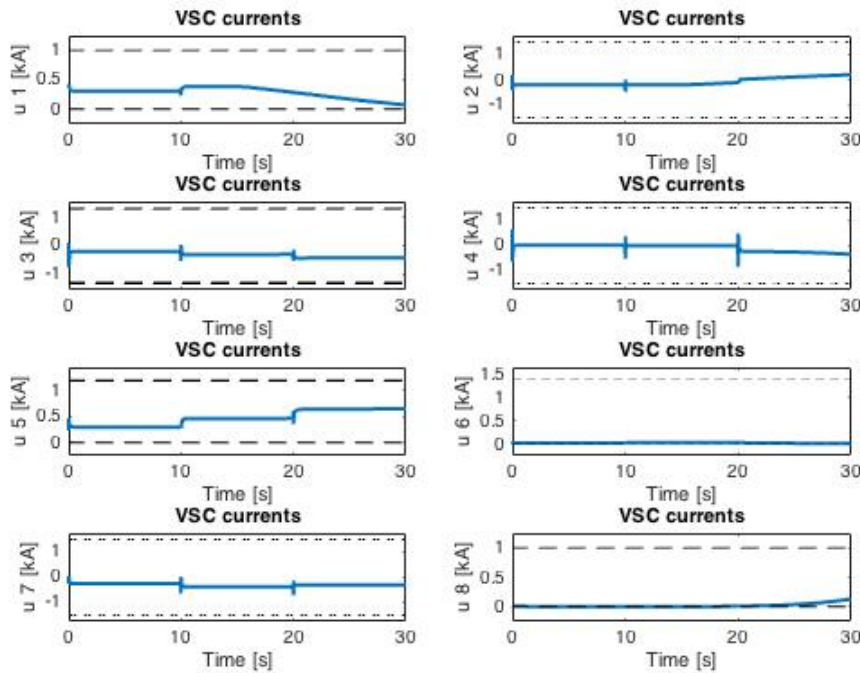


Figure 4.5. Evolution of the VSC current in each node.

The next graphic is the power in every node. It is calculated multiplying the current by the voltage in each node, $P=u \cdot E$.

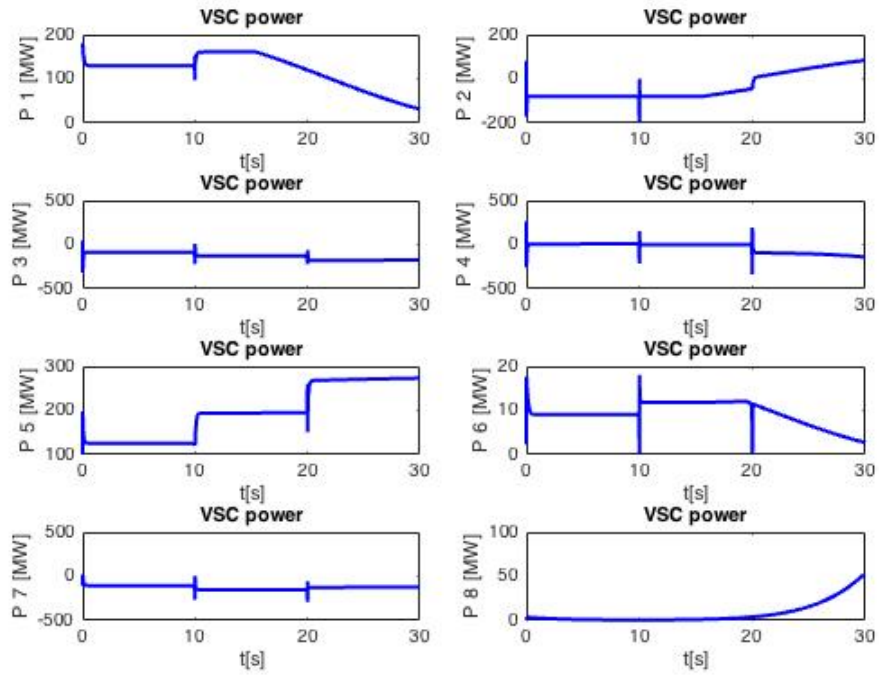


Figure 4.6. Evolution of VSC power in each node.

The following graphic represents the current in each line of the network.

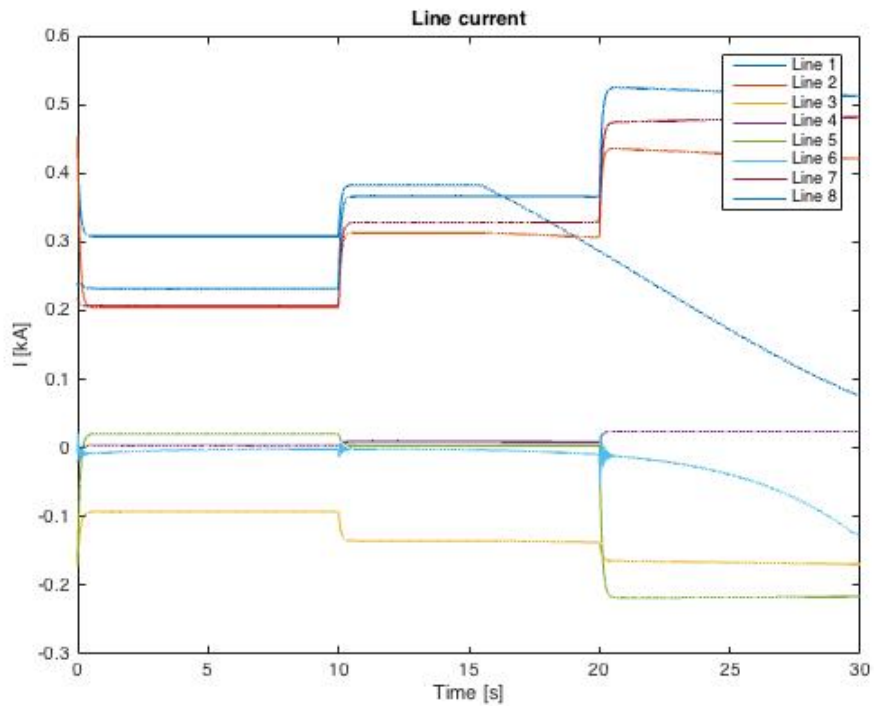


Figure 4.7. Evolution of the current in each line of the network.

Other plots of interest are the ones related to the wind profile. The first one is the representation of the spline-interpolated wind values. The second one is the actual generated power in the wind farm nodes. In both graphics only the wind farms nodes are represented, as the other ones are not of interest.

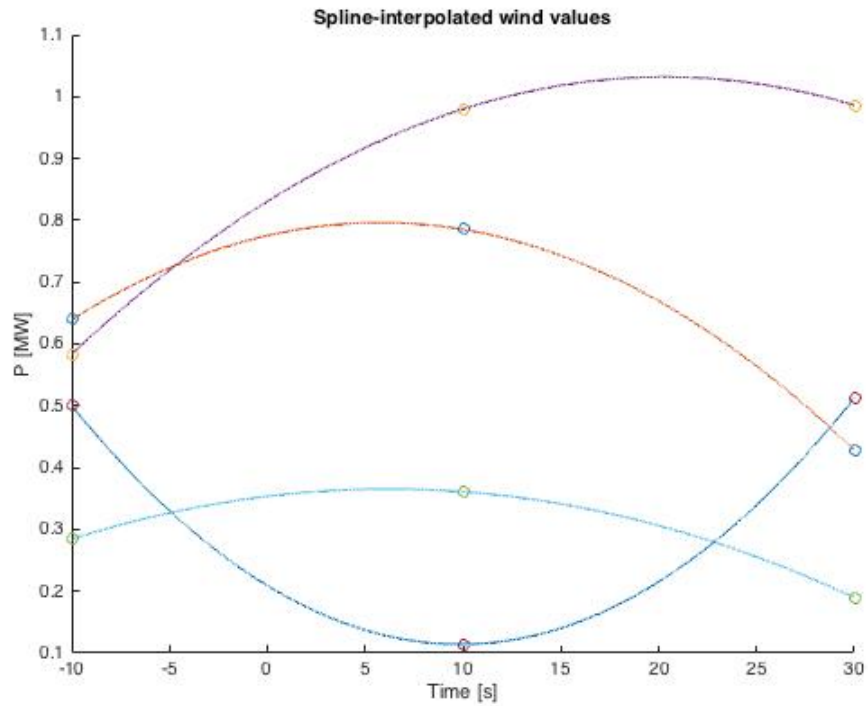


Figure 4.8. Spline interpolated winds values, in this case three different points were used to define the spline.

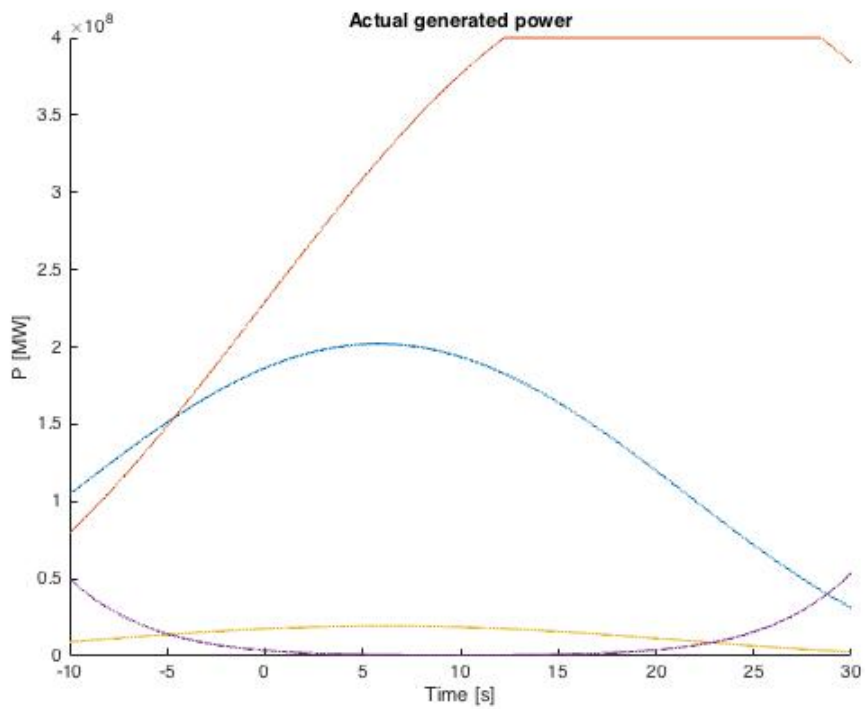


Figure 4.9. Generated power of the wind farms of the network.

CHAPTER 5: CONCLUSIONS AND FUTURE WORK

As it was exposed in the beginning of this document, the aim was to develop an algorithm capable of modelling and simulating an HVDC network interconnecting offshore wind farms with the mainland. Once implemented all the steps to achieve the proper simulation it can be concluded that the main goal of the project is accomplished.

Once the algorithm has been implemented, in which different simulations are linked and the secondary control calculates the OPF, some conclusions can be extracted.

Firstly, it is an interesting point to simulate different iterations to be able to introduce different conditions. In this case the conditions are given by the optimiser, which calculate the OPF for the given conditions at the end of each iteration. The output of the secondary control is the voltage in which the network will give a better performance minimizing the power losses on the transmission lines.

The implementation of the secondary control in the model represents a more realistic case. In real life there is a demand of the power needed in every consumption point. This is the case of the presented model in which every node has some restrictions that characterise it.

Another point of interest is the wind profile used in the simulation. In the current work an approximation of the wind available in every node was calculated using the Weibull distribution. The parameters used are the typical ones that can be found in the areas where the offshore wind farms are located. However, the result obtained from the wind model might not be a realistic case as it have pronounced variations in short times. These variations can be seen in Figure 4.8.

While developing the project some ideas and possible implementations came out, however they have not been implemented in this project. Some of the ideas would be of interest in case it is wanted to investigate in the future.

In the real transmission system it is known that sometimes the networks fail and a power plant or a transmission can not be used, or used but not on its totality. An interesting implementation would be to simulate the failure of a node. The idea could be to make a random algorithm that chooses the node that will fail and decide whether it can be used partially or it cannot be used.

Finally, some constraints could be applied to the transmission lines to make more realistic case. As it is known, the conductors in real life have a maximum current admissibility, and if this maximum is exceeded the conductor gets burnt and the fluctuation of power is stopped. This consideration could be considered of study and try to implemented on the algorithm.

CHAPTER 6: BIBLIOGRAPHY

6.1 Bibliographic references

Benedito, E., Puerto-flores, D., Arnau, D., & Feltz, O. Van Der. (2016). Strictly convex losses functions for port-Hamiltonian based optimal algorithm for MTDC networks, Submitted to the 55th IEEE Conference on Decision and Control

Bianchi, F., H. D. Battista, and R. Mantz (2007). Wind turbine control systems: Principles, modelling and gain scheduling design. Springer.
<http://doi.org/10.1007/1-84628-493-7>

Dòria-Cerezo, A., Olm, J.M., & Scherpen, J.M.A., Passivity-based control of multi-terminal HVDC systems under control saturation constraints, Proc. 5th IFAC Workshop on Lagrangian and Hamiltonian Methods for Non Linear Control, Lyon (France), July 4-7, 2015. <http://doi.org/10.1016/j.ifacol.2015.10.227>

Ewea. (2016). Wind in power - 2015 European statistics. 2015 European Statistics., 2016(February 13), The European Wind Energy Association.

Feltz, O. Van Der. (2016). Control of multi-terminal HVDC transmission systems. M.Sc. Thesis, University of Groningen.

Gavriluta, C. (2014). Control and Operation of Multi-Terminal VSC-DC Networks. PhD Thesis Universitat Politècnica de Catalunya

GWEC. (2015). Global wind report; annual market update 2014. Technical report, Global Wind Energy Council.

Hossain, J., & Mahmud, A. (2014). *Large Scale Renewable Power*

Generation, Advances in Technologies for generation, transmission and storage.

- Paulet, G. (2016). Design and simulation of a supervisor algorithm for a HVDC network. BSc Thesis, Universitat Politècnica de Catalunya
- Pons, J. (2015). Modeling and simulation of an HVDC network for offshore wind farms. BSc Thesis, Universitat Politècnica de Catalunya
- Setreus, J., & Bertling, L. (2008). Introduction to HVDC Technology for Reliable Electrical Power Systems. *Proceedings of the 10th International Conference on Probabilistic Methods Applied to Power Systems*, (June 2016), 1–8.
- Snyder, B., & Kaiser, M. J. (2009). Ecological and economic cost-benefit analysis of offshore wind energy. *Renewable Energy*, 34(6), 1567–1578. <http://doi.org/10.1016/j.renene.2008.11.015>
- Sood, V. K. (2007). *Power Electronics Handbook*. Power Electronics Handbook. Elsevier. <http://doi.org/10.1016/B978-012088479-7/50048-1>

6.2 Complementary bibliography

- Aragüés-Peñalba, Mònica, Agustí Egea-Àlvarez, Oriol Gomis-Bellmunt, and Andreas. Sumper. "Optimum voltage control for loss minimization in HVDC multi-terminal transmission systems for large offshore wind farms." *Electric Power Systems Research*, no. 89 (2012): 54-63.
- Dòria-Cerezo, A., J. Olm, M. di Bernardo, M. Quaglia, and E. Nuño (2014). Bounded synchronization in resistive multiterminal VSC-HVDC transmission systems. In *Proc. 53th IEEE Conf. on Decision and Control*.
- Doria-Cerezo, Arnau, Josep M. Olm, Mario di Bernardo, and Emmanuel. Nuno. "Modelling and control for bounded synchronization in multi-terminal VSC-HVDC transmission networks." *IEEE Transactions on Circuits and Systems I: Regular Papers*. 2016.
- Egea-Alvarez, A., F. Bianchi, A. Junyent-Ferré, G. Gross, and O. Gomis-Bellmunt (2013). Voltage control of multiterminal VSC-HVDC

transmission systems for offshore wind power plants: design and implementation in a scaled platform. IEEE Trans. on Industrial Electronics 60(6), 2381–2391.

Gomis-Bellmunt, O., J. Liang, J. Ekanayake, and N. Jenkins (2011). Voltage-current characteristics of multiterminal HVDC-VSC for offshore wind farms. Electric Power Systems Research 81(2), 440–450.

Hau, E. (2006). Wind turbines. Fundamentals, technologies, application, economics (2nd ed.). Springer.

Pinto, R. (2014). Multi-Terminal DC Networks System Integration, Dynamics and Control. Ph. D. thesis, Delft University of Technology.

ANNEX: MATLAB CODE

Sim.m file

```
clearvars;
clear all;
close all;
clc;

tic1=tic; % Start all-simulation long stopwatch

global Pmax Pmin AiRA nN Emax Emin Imax Imin V0
%% Parameters

% Read Excel file

file='HVDC_data';

lines=xlsread(file,'Lines'); % 1st sheet contains each line's
parameters
vertex=xlsread(file,'Nodes'); % 2nd sheet with node parameters
vertex(isnan(vertex))=0; % Swap NaN values (empty cells) for zeros
simdata=xlsread(file,'Other'); % 3rd sheet with simulation options

% Simulation/Wind parameters
tSample=simdata(1); % Wind sample time
WindCh=simdata(2); % Wind changes, 1 = YES 0 = NO
t0=simdata(3); % Simulation start time
tf=simdata(4); % Wind end time
StepSize=simdata(5); % Max timestep ode / timestep (if fixed)
RelTol=simdata(6); % Relative tolerance for the ode solver
AbsTol=simdata(7); % Absolute tolerance for the ode solver
nSim=simdata(8); % Number of simulations/iterations
tSim=simdata(9); % Time of each simulation/iteration

% Network parameters
LisLin=lines(:,1:2); % Nodes that define the network lines
```

```

LisN=vertex(:,1); % List of nodes
nLin=size(lines,1); % Number of lines
nVSC=nnz(vertex(:,3)); % Number VSC nodes
nN=max(max(LisLin)); % Number of total nodes

% Incidence Matrix
A=zeros(nN,nLin); % Empty matrix
for i=1:nLin % Check every line
    A(LisLin(i,1),i)=1; % A(i,j)=1 if line j connects from node i
    A(LisLin(i,2),i)=-1; % A(i,j)=-1 if line j connects to node i
end

% G Matrix
G=zeros(nN,nN); % Empty matrix. G used to unscramble node matrices
for i=1:nN
    G(LisN(i),i)=1;G(LisN(i),i)=1; % G(k,j)=1 if node k is in position
    j in the Excel sheet
end

% Resistance, Inductance
R=diag(lines(:,5)); % R values are in mOhm in the spreadsheet
L=diag(lines(:,7))*1e-3; % L values are in mH in the sheet

% Conductance matrix
AiRA=A*inv(R)*A'; % Calculation of the conductance matrix G

% Capacitance
C=zeros(nN,1); % Empty capacitance matrix
for i=1:nN % Fill matrix with the capacitance values of each VSC
    C(LisN(i))=vertex(i,3);
end
for i=1:nLin % Add to matrix the capacitance of the lines connecting
the nodes
    for j=1:2 % If line connects to or from node j
        C(LisLin(i,j))=C(LisLin(i,j))+lines(i,9)/2; % add C_line/2 to
the total capacitance
    end
end
iC=inv(diag(C*1e-6)); % uF values to F; diagonal and inverse

% Initial Conditions
icL=lines(:,10)*1e3; % Initial line currents in kA
icC=G*vertex(:,2)*1e3; % Initial line voltages in kV

% Windfarms/GS (VSCs)
iTau=vertex(:,8).^(-1); % Time constants of the VSCs (inverted)
iTau(isinf(iTau))=0;
iTau=diag(G*iTau);
alpha=G*vertex(:,9); % Scaling factor of the Weibull distribution
(alpha)
k=G*vertex(:,10); % Weibull form parameter (k)
RatSpd=G*vertex(:,11); % Rated wind speed for wind turbines of each WF
isWF=sign(alpha); % isWF(k)=1 if node k is a wind farm, 0 otherwise
nWF=nnz(isWF); % Number of wind farms
nGS=nVSC-nWF; % Number of grid stations

```



```

% Droop control parameters
umax=G*vertex(:,4)*1e3; % Maximum current each node can generate
umin=umax.*(isWF-1); % Current each VSC can absorb (0 for wind farms)
E0=G*vertex(:,5)*1e3; % Voltage nominal value for each node
md=G*vertex(:,6); % Slope proportional section droop control
Pg=G*vertex(:,16)*1e6; % Maximum power each node can generate
Pc=Pg.*(isWF-1); % Power each node can consume

% Line periods, warning if StepSize is not small enough
[omega,zeta]=deal(zeros(nLin,1)); % Empty matrices
for i=1:nLin % For each line
    omega(i)=sqrt(iC(LisLin(i,2),LisLin(i,2))/L(i,i)...
        +iC(LisLin(i,1),LisLin(i,1))/L(i,i));
    zeta(i)=R(i,i)/(L(i,i)*2*omega(i));
end % Calculates values relating to the lines transfer functions
T=(2*pi)./(omega.*sqrt(1-zeta.^2)); % Periods of the lines dynamics
if (min(T)/StepSize) < 39.9 % Issue a warning if StepSize is
considered too "coarse"
    warning off backtrace
    warning('WarnTests:convertTest',...
        ['Your chosen step size of ', num2str(StepSize,'%10.2e\n'),...
        ' may not be small enough.\n Please consider using a step size
of around ',...
        num2str(round(min(T)/40, 3, 'significant'), '%10.2e\n'), ' or
smaller.\n']);
end

%% Wind profiles

tic % Start wind profiles stopwatch timer
fprintf ('Generating wind profiles, please wait...\n');

% Time vectors
if tSample>tf % If simulation time is smaller than wind sample time
    tSample=tf; % simulate wind profiles just for simulation time
end
Samples=t0:tSample:tf; % Intervals when wind changes
tWind=t0:StepSize/2:tf; % Times the wind values are needed (Ode4 uses
the midpoint)

% Inverse cumulative Weibull distribution function
invW=@(alpha,k,x) alpha*log(-1/(x-1))^(1/k);

% Wind values, Generated Power
rWind=zeros(nN,length(Samples)); % Empty matrix with "real" wind
speeds
Wind=zeros(nN,length(tWind)); % Matrix with simulated (interpolated)
wind values
PgA=diag(Pg)*ones(nN,length(Wind)); % Actual generated power
initialized as nominal power

rng(3) % Controls random number generated sequence with 'rand' below

```

```

    for i=1:nN % For each node
        if WindCh==1 && isWF(i)==1 % If wind changes and node is a
wind farm
            for j=1:length(Samples) % For every interval in which wind
speed changes
                rWind(i,j)=invW(alpha(i),k(i),rand(1))./RatSpd(i); %
"Real" wind speed
            end % Wind speed is "normalized" in relation to the rated
speed of the turbine
            Wind(i,:)=spline(Samples,rWind(i,:),tWind); % Spline-
interpolated wind values
            for j=1:length(tWind) % For each simulated wind value
                PgA(i,j)=max(min(Pg(i),Pg(i)*Wind(i,j).^3),-Pc(i)); %
Actual generated power
            end
        end
    end

toc % Read latest stopwatch timer

%% Simulation

% Matrices needed for calculations
On=zeros(nN); % Empty matrix no.nodes-by-no.nodes
Oln=zeros(nLin,nN); % Empty matrix no.lines-by-no.nodes
Onl=zeros(nN,nLin); % Empty matrix no.nodes-by-no.lines
B1=[On,-iC*A,iC
    L\A',-L\R,Oln
    On,Onl,-iTau]; % Matrix with ode coefficients multiplying 'x'
B2=[On; Oln; iTau]; % Matrix multiplying 'ud' (desired currents VSCs)

cellsims = cell(1,nSim); % Initialization of the cell sims
celltime = cell(1,nSim); % Initialization of the cell time
cellE=cell(1,nSim); % Initialization of the cell voltage
celltspan = cell(1,nSim); % Initialization of the cell voltage
vect=[]; % Initialization of the vector time
vecx=[]; % Initialization of the array results of the ODE
vecE=[]; % Initialization of the array which stock the voltage in each
simulation
vecEoptim=[]; % Initialization of the array optimal voltage

% ODE options, times, initial values
satP=min(PgA(:,1), max(Pc,(-md.*(icC-E0)).*icC)); % Power saturation
on initial E values
icu=min(umax, max(umin, (satP)./icC)); % VSC currents initial values
x0=[icC; icL; icu]; % Initial values for simulation

tStart=t0;
tStop=tStart+tSim;

options1 = optimoptions('fmincon','Algorithm','sqp'); % Options and
specifications for the optimization process

Pmax = vertex(:,12)*1e6; % Array of the maximus power bounds for all
the nodes
Pmin = vertex(:,13)*1e6; % Array of the minimus power bounds for all
the nodes

```

```
Imax = umax; % Array of the maximus current bounds for
all the nodes
Imin = vertex(:,14)*1e3; % Array of the minimus current bounds for
all the nodes
Emax = vertex(:,15)*1e3; % Array of the maximus voltage bounds for
all the nodes
Emin = vertex(:,16)*1e3; % Array of the minimus voltage bounds for
all the nodes
gmax = vertex(:,17); % Coefficient for the maximum limit of
generated power for the wind
gmin = vertex(:,18); % Coefficient for the maximum limit of
generated power for the wind

PmaxInicial=Pmax;
PminInicial=Pmin;

V0=E0;

for i=1:nSim

tspan=tStart:StepSize:tStop;

lb=[Emin]; % Lower bounds for the optimizing algorithm
ub=[Emax]; % Upper bounds for the optimizing algorithm

try % Tries to generate C code of the file Ode4CG with '>codegen'
% Code generation
tic % Start codegen stopwatch timer
fprintf ('\nGenerating C code for faster simulation, please
wait...\n');
codegen Ode4CG -args {tspan-
t0,x0,nN,umax,umin,E0,md,Pc,PgA,StepSize,B1,B2}
toc % Read latest stopwatch timer

catch % If the computer/MATLAB version doesn't allow code generation
(or there is
% an error), it procedes with the simulation with Ode4CG.m in MATLAB
(slower) code
fprintf ('\nCould not generate C code for faster simulation. ');
fprintf ('\nSlower simulation now starting, please be patient...\n');
tic % Start "slow" simulation stopwatch timer
[x] = Ode4CG(tspan-t0,x0,nN,umax,umin,E0,md,Pc,PgA,StepSize,B1,B2)

toc % Read latest stopwatch timer

end

% MEX-Simulation
tic % Start "fast" simulation stopwatch timer
fprintf ('\nNow simulating, please wait...\n');

[V,FVAL,EXITFLAG,OUTPUT] =
fmincon(@OPFof,V0,[],[],[],[],lb,ub,'OPFnlc',options1); % Optimizing
algorithm function

EXITFLAG

V0=V;
E0=V;
```

```

% Calculation of the power bounds comparing PgA and bounds of the
nodes

pmin1=zeros(nN,1); % Initialization of the minimum power for WF
vector
pmin0=pmin1;      % Initialization of the minimum power for GS
vector

Pmax = min(PmaxInicial,PgA(:,(length(tspan)-1)*i).*gmax); % Array of
the maximus power bounds for all the nodes

% Calculation of the minimum bounds
for k=1:nN
    if isWF(k)==1
        pmin1(k) = max(PminInicial(k),PgA(k,(length(tspan)-
1)*i).*gmin(k));%Canviar PminInitial !!!!!!!!!!!!!
    else
        pmin0(k) = min(PminInicial(k),PgA(k,(length(tspan)-
1)*i).*gmin(k));% !!!!!!!!!!!!!!!!!!!!!
    end
end

Pmin=pmin1+pmin0; % Array of the minimum power bounds for all the
nodes

vecPmax=diag(Pmax)*ones(size(Pmax,1),size(tspan,2)); % Array of the
minimum power bounds
vecPmin=diag(Pmin)*ones(size(Pmin,1),size(tspan,2)); % Array of the
maximum power bounds

[x] = Ode4CG_mex(tspan-
t0,x0,nN,umax,umin,E0,md,Pc,PgA,StepSize,B1,B2);

cellE{i}=[V]; % Stock all the results of the voltage
cellsims{i} = [x]; % Stock all the results of the ODE solver
celltime{i}=[tspan]; % Stock all the points of the
simulation/iteration

vecE=[vecE,cellE{i}]; % Creation of the voltage array
vecx=[vecx,cellsims{i}]; % Creation of the results array
vect=[vect,celltime{i}]; % Creation of the time vector

if i==1
    x_initial=zeros(size(x,1),nSim);
    lengthsim=length(vect);
end
for w=1:lengthsim % For every point of the simulation/iteration

    vecEoptim=[vecEoptim,V]; % Creation of the optimal voltage
array
end

% New initial conditions for the next simulation/iteration

x_initial(:,i)=x(:,(size(x,2)));
x0=x_initial(:,i); %

```

```
% Definig new interval of time for the next simulation/iteration
tStart=tStop;
tStop=tStop+tSim;

toc % Read latest stopwatch timer
end

%%
fprintf ('\nSimulation finished.\n');
toc1=toc(tic1); % Display total time elapsed for the simulation
fprintf(['Total elapsed time is ',num2str(toc1),' seconds.\n\n']);

% % No postprocessing takes place in this file; use the Plots
"section"
```

Ode4CG.m file

```
function Y = Ode4CG(tspan,y0,...
    nN,umax,umin,E0,md,Pc,PgA,StepSize,B1,B2) %#codegen

% % Ode4: Solve differential equations with a non-adaptive method of
% % order 4.
% %
% % Y = ODE4(TSPAN,Y0, FUNMODEL) with TSPAN = [T1, T2, T3, ... TN]
% % integrates the
% % system of differential equations  $y' = f(t,y)$  by stepping from T0
% % to T1 to TN.
% % Function FUNMODEL (T,Y) must return f(t,y) in a column vector.
% % The vector Y0
% % is the initial conditions at T0. Each row in the solution array
% % Y corresponds
% % to a time specified in TSPAN.
% %
% % Y = ODE4(TSPAN,Y0,P1,P2...) passes the additional parameters
% % P1,P2... to
% % the derivative function as FUNMODEL (T,Y,P1,P2...).
% %
% % This is a non-adaptive solver. The step sequence is determined
% % by TSPAN but the
% % derivative function ODEFUN is evaluated multiple times per step.
% % The solver
% % implements the classical Runge-Kutta method of order 4.
% %
% % Example
% %     tspan = 0:0.1:20;
% %     y = ode4(tspan,[2 0],@vdp);
% %     plot(tspan,y(:,1));
% % solves the system  $y' = vdp1(t,y)$  with a constant step size of
% % 0.1, and plots
% % the first component of the solution.
```

```

if ~isnumeric(tspan) % Make sure that tspan contains only numbers
    error('TSPAN should be a vector of integration steps.');
```

end

```

if ~isnumeric(y0) % Make sure that y0 contains only numbers
    error('Y0 should be a vector of initial conditions.');
```

end

```

h = diff(tspan); % Vector with all the step-sizes
if any(sign(h(1))*h <= 0) % Make sure that there are no 'negative'
step-sizes
    error('Entries of TSPAN are not in order.')
```

end

```

neq = length(y0); % Number of unknowns/equations to solve
N = length(tspan); % Number of timesteps
Y = zeros(neq,N); % Empty solution matrix: no.unknowns-by-no,
timesteps
K = zeros(neq,4); % For each timestep RK4 calculates 4 values of the
approx. slope
```

```

Y(:,1) = y0; % The solution for the first timestep are the initial
conditions
for i = 2:N % For the 2nd timestep and onwards
    ti = tspan(i-1); % Initial time for the step
    hi = h(i-1); % Step-size for the timestep
    yi = Y(:,i-1); % The initial sol. for the timestep is the final from
the previous one
    K(:,1) = funmodel(ti,yi,nN,...
        umax,umin,E0,md,Pc,PgA,StepSize,B1,B2); % K1 = f(ti, yi)
    K(:,2) = funmodel(ti+0.5*hi,yi+0.5*hi*K(:,1),nN,...
        umax,umin,E0,md,Pc,PgA,StepSize,B1,B2); % K2 = f(ti+h/2,
yi+K1*h/2)
    K(:,3) = funmodel(ti+0.5*hi,yi+0.5*hi*K(:,2),nN,...
        umax,umin,E0,md,Pc,PgA,StepSize,B1,B2); % K3 = f(ti+h/2,
yi+K2*h/2)
    K(:,4) = funmodel(tspan(i),yi+hi*K(:,3),nN,...
        umax,umin,E0,md,Pc,PgA,StepSize,B1,B2); % K4 = f(ti+h, yi+K3*h)
    Y(:,i) = yi + (hi/6)*(K(:,1) + 2*K(:,2) + 2*K(:,3) + K(:,4));
    % y[n+1]=y[n]+(h/6)*(K1+2*K2+2*K3*K4)
end
```

```

function dxdt=funmodel(t,x,nN,umax,umin,E0,md,Pc,PgA,StepSize,B1,B2)

% % FUNMODEL solves the right-hand side of the system of differential
equations

E=x(1:nN); % Node voltage values, the first nN values in the solution
matrix
satP=min(PgA(:,single((2*t/StepSize)+1)),...
    max(Pc,(-md.*(E-E0)).*E)); % Power saturation in VSC nodes
ud=min(umax, max(umin, (satP)./E)); % Current saturation; VSCs
currents

dxdt=(B1*x+B2*ud);
```

OPFnlc.m file

```
function [c,ceq]=OPFnlc(V)
global Pmax Pmin AiRA nN Emax Emin Imax Imin

%Limit of maximum and minimum power for all vertex
for i=1:nN
    y(i) = V(i) * (AiRA(i,:) * V) - Pmax(i);
end

for i=1:nN
    z(i) = -V(i) * (AiRA(i,:) * V) + Pmin(i);
end

%Limit of maximum and minimum current for all vertex
for i=1:nN
    q(i) = (AiRA(i,:) * V) - Imax(i);
end

for i=1:nN
    w(i) = -(AiRA(i,:) * V) + Imin(i);
end

c=[y;z;q;w];
ceq=[];
```

OPFof.m file

```
function Ploss = OPFof(V)
global AiRA
Ploss=V'*AiRA*V;
```

plots.m file

```
close all

%% Arrays of the maximum and minimum bounds

% Matriu de plot per els l imits
diagV=diag(V);
vecplotEmax=diag(Emax)*ones(size(Emax,1),size(vect,2)); % Maximum
voltage
vecplotEmin=diag(Emin)*ones(size(Emin,1),size(vect,2)); % Minimum
voltage
vecplotImax=diag(Imax)*ones(size(Imax,1),size(vect,2)); % Maximum
current
vecplotImin=diag(Imin)*ones(size(Imin,1),size(vect,2)); % Minimum
current
```

```

%% Array of the results
vecplotE=vecx(1:nN,:); % VSC voltage
vecploti=vecx(nN+1:nN+nLin,:); % Line currents
vecplotu=vecx(nN+nLin+1:nN+nLin+nVSC,:); % VSC currents
%% Wind generation plots

for i=1:nN
    if isWF(i)==1
        hold on
        plot(tWind,PgA(i,:))
        title('Actual generated power')
        xlabel('Time [s]')
        ylabel('P [MW]')
    else
        end
end

figure
for i=1:nN
    if isWF(i)==1
        hold on
        plot(Samples,rWind(i,:), 'o', tWind, Wind(i,:))
        title('Spline-interpolated wind values')
        xlabel('Time [s]')
        ylabel('P [MW]')
    else
        end
end

%% Node voltage plot

figure
for k=1:nN % For every node

    subplot(ceil(nN/2),2,k)

    plot(vect,vecplotE(k,:)*1e-3,'Linewidth',2)
    hold on
    plot(vect,vecplotEmax(k,:)*1e-3,'--
', 'Linewidth',1, 'color','Black') % Graphic of the maximum levels
    plot(vect,vecplotEmin(k,:)*1e-3,'--
', 'Linewidth',1, 'color','Black') % Graphic of the minimum levels
    plot(vect,vecEoptim(k,:)*1e-3,':g', 'Linewidth',2) % Graphic of
the optimal voltage
    axis([0 vect(end) ((Emin(k)*1e-3)-5) ((Emax(k)*1e-3)+5)])
    title('Node Voltage')
    xlabel('Time [s]')
    ylabel(['E {' num2str(k) '} [kV]'])

    end

%% Current line plots
figure
for j=nN+1:nN+nLin

    plot(vect,vecx(j,:)*1e-3)
    hold on
    title('Line current')
    xlabel('Time [s]')

```



```
ylabel('I [kA]')
xlim([0 vect(end)])
legendInfo{j-nN}=['Line ' num2str(j-nN)];

end
legend(legendInfo)
%% VSC current plots
figure
for j=1:nLin

subplot(ceil(nLin/2),2,j)

plot(vect,vecplotu(j,:)*1e-3, 'Linewidth', 2)
hold on
plot(vect,vecplotImax(j,:)*1e-3,'--','Linewidth',1,'color','Black')
% Graphic of the maximum levels for
plot(vect,vecplotImin(j,:)*1e-3,'--','Linewidth',1,'color','Black')
% Graphic of the minimum levels for
axis([0 vect(end) ((Imin(j)*1e-3)-0.25) ((Imax(j)*1e-3)+0.25)])
title('VSC currents')
xlabel('Time [s]')
ylabel(['u {' num2str(j) '} [kA]'])

end
%% Power plots
vecP = vecplotE.*vecplotu; % Calculation of the power vector for all
nodes
figure;
for i=1:nN
    subplot(ceil(nN/2),2,i)
    plot(vect,vecP(i,:)*1e-6,'b','Linewidth',2) % Graphic of the
power
    hold on
    title('VSC power'),xlabel('Time [s]'),ylabel('P [MW]')
    xlabel(['t[s]'])
    ylabel(['P {' num2str(i) '} [MW]'])
    xlim ([0 vect(end)])

end
```

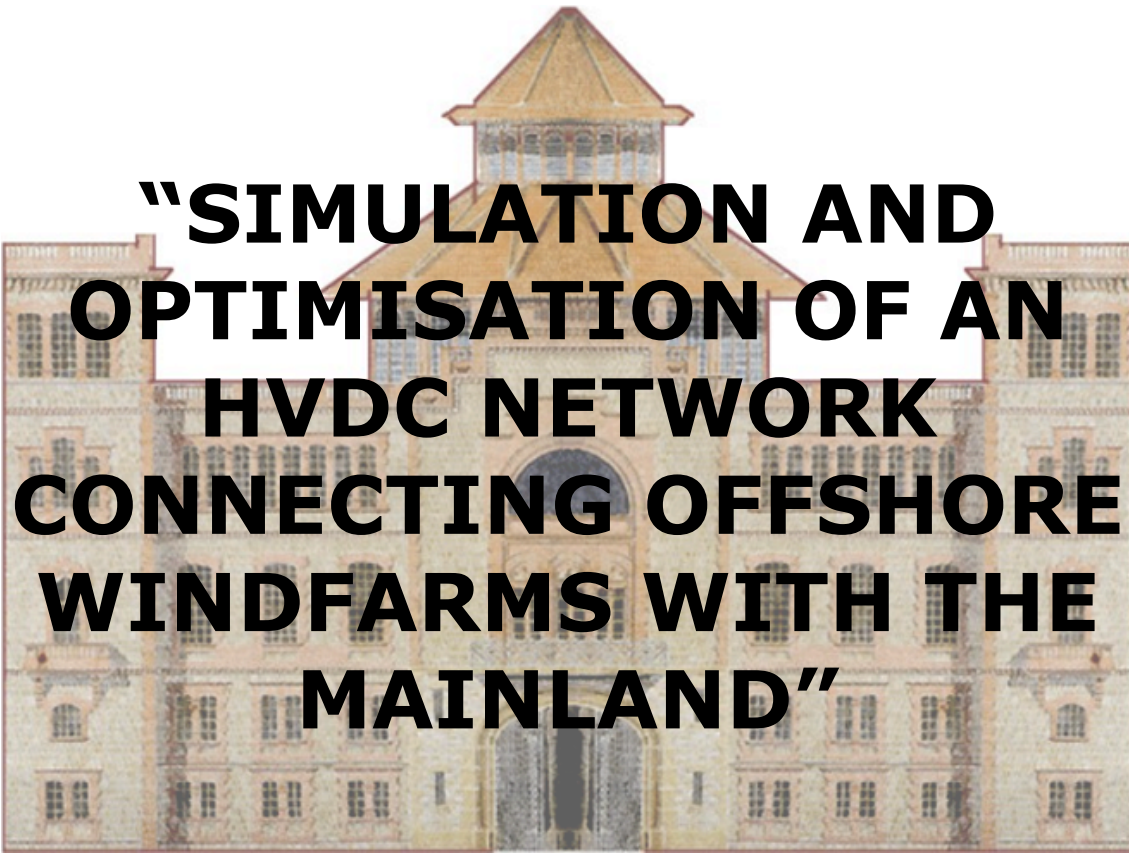



Escola Universitària d'Enginyeria
Tècnica Industrial de Barcelona
Consorci Escola Industrial de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Budget

TREBALL FINAL DE GRAU



"SIMULATION AND OPTIMISATION OF AN HVDC NETWORK CONNECTING OFFSHORE WINDFARMS WITH THE MAINLAND"

TFG presentat per obtenir el títol de GRAU en
ENGINYERIA ELÈCTRICA

Per **Nil Falgueras Farrerons**

Barcelona, 08 de Juny de 2016

Director: Arnau Dòria Cerezo
Departament d'Enginyeria Elèctrica (EE - 709)
Universitat Politècnica de Catalunya (UPC)

BUDGET

Concept	Unit cost	Units	Total
Engineering costs			
Initial research and contextualization	20 €/h	125 h	2.500 €
Programming	20 €/h	350 h	7.000 €
Redaction	20 €/h	125 h	2.500 €
Technological resources			
Matlab Student license	69 €/h	1	69 €
Amortization of computer hardware	200€/h	1	200 €
Subtotal			12.269 €
IVA (21%)			2.576,49 €
Total			14.845,49 €

Nil Falgueras Farrerons

Barcelona, 08 June 2016