Politecnico di Torino

Politecnico di Torino

Corso di Laurea in Ingegneria Meccanica

A.A. 2022/2023

Sessione di Laurea Marzo/Aprile 2023

Tesi di Laurea Magistrale

Implementation of a Model Predictive Controller (MPC) for Floating Offshore Wind Turbines (FOWTs)

Relatore :

Giovanni Bracco

Correlatore :

Massimo Sirigu

Candidato :

Davide Issoglio

Abstract

In recent times, renewable resources have been increasingly exploited, with wind energy being second only to hydro and solar power. Over the last decade, thanks to technological advances, the average cost of energy has come down considerably, making this sector increasingly competitive. Although onshore wind and fixed offshore are more widespread due to their greater development and fewer technological challenges, floating offshore is also starting to progress, also thanks to the possibility of exploiting areas with good wind resources but with a deep seabed and to the possibility of reaching large sizes without visual pollution or environmental impact problems.

With the development of floating offshore wind turbines (FOWTs), it therefore becomes necessary to adapt control strategies, starting from conventional controllers used in the non-floating counterpart, or developing new types of control, such as those based on PID logic, or unconventional ones. The aim of this thesis is the realization of a controller that falls into the latter category that is based on MPC (Model Predictive Control) logic. Controllers of this type exploit the knowledge of the mathematical model of the controlled system to find the optimal values of the control actions with the aim of maximizing its performances.

After a brief introduction on the current situation of wind energy and the description of the main components of a wind turbine, the case study adopted for the implementation of the controller will be described. After that, the mathematical model of the latter will be obtained, based on a pre-existing model, and validated through it. In a second step, the control strategies for wind turbines will be analysed and a linear MPC controller will be studied from a theoretical point of view. Finally, exploiting the knowledge of the system's model and of the theoretical foundations above mentioned, an MPC controller will be implemented, based on a set of LTI (Linear Time Invariant) systems, each of which is obtained by linearizing the starting non-linear system in different operating conditions. It will be exploited for the realization of the various controller modules, together with optimization algorithms for the search of the control actions and algorithms for the estimation and forecasting of some unknown quantities for the control loop closure. Once implemented, the controller will be tested by comparing its performance with other widely used conventional controllers, in order to draw some conclusions on its effectiveness and trying to understand how its unconventional logic can be exploited for further developments.

Contents

List of Figures

List of Tables

1 Introduction

In this section wind energy current situation is briefly shown, whit a look on market status of 2021 and market outlook until 2031, according with information shared by GWEC, the Global Wind Energy Council. Most of data which will be shown are from last reports compiled by GWEC: "Global Wind Report 2022" [1], "Global Offshore Wind Report 2022" [2] and "Floating Offshore Wind - A Global Opportunity" [3]. In addition to discussing the economic point of view, on which the first part of this introduction focuses, a brief explanation of the various components that constitute a wind turbine will be given in the second part, in which technology innovation in this field will be discussed. Since the subject of this thesis is focused on floating offshore wind turbines, particular attention will be paid to the platforms on which these types of turbines are mounted, giving an overview of the different types of solutions adopted and trying to understand their advantages and disadvantages.

1.1 Wind Energy Overview

The wind industry has made good progress recently, with 2020 and 2021 as the years of highest annual growth ever. Regarding 2022, this trend was slightly worse than precedent years due to energy crisis, but it has led to positive changes in term of green energy goals, so there are pretty good chances development of this sector will rapidly increase.

According to the Global Wind Energy Council's (GWEC) 'Global Wind Report 2022', almost 94 GW of power will be installed in 2021, with a 12% year-on-year (YoY) increase, despite the ongoing COVID-19 pandemic. The onshore wind market added 1,8% less capacity than the previous year due to a slowdown in China and the US, the world's two largest wind markets. However, there was record growth in Europe, Latin America and Africa and the Middle East. The offshore wind market recorded its best year ever in 2021, three times added capacity more than the previous year. China contributed 80% of this offshore growth, the fourth year in which China led the ranking of new installations. Europe is the only other region to record new offshore wind installations, thanks to the UK's record year.

Figure 1.1: Wind turbines' photos

A total of 557 GW of capacity is expected to be added globally in the period 2022-2026. Furthermore, these market forecasts could be revised upwards due to ongoing changes in energy policies and the impact of the Ukrainian crisis in 2022. However, this growth rate may not be fast enough to reach the targets set by the Paris Agreement or to achieve zero net emissions by 2050. Policy changes and faster authorisations for wind power projects are needed to accelerate the deployment of renewable energy.

1.1.1 Market Status up to 2022

In 2021, 72.5 GW of new onshore wind capacity was recorded globally, bringing cumulative onshore wind capacity to 780 GW. China had a 39% drop in new installations compared with 2020, following the end of the Feed-in-Tariff (FiT) and the transition to a "grid parity" system, where onshore wind is paid based on the regulated price for coal power. In the US, growth slowed down in the second half of the year due to supply chain issues and other disruptions caused by COVID-19. In addition to China (30.7 GW) and the US (12.7 GW), the top five onshore wind markets were Brazil (3.8 GW), Vietnam (2.7 GW) and Sweden (2.1 GW).

In 2021, the world saw a record-breaking 21.1 GW of offshore wind become grid connected, bringing the total global offshore wind capacity to 57.2 GW. China led the way in annual offshore wind installations for the fourth consecutive year, with nearly 17 GW of new capacity, bringing its cumulative offshore wind installations to 27.7 GW. Europe accounted for most of the remaining new capacity, with 3.3 GW of offshore wind capacity added in 2021. The UK installed 2.3 GW of new offshore wind last year, making it the largest European offshore wind market in 2021, followed by Denmark (608 MW) and the Netherlands (392 MW). There was only one small offshore wind project under construction in Germany during 2021, and no offshore wind turbines were installed, primarily due to previously unfavourable market conditions and a low level of short-term offshore wind projects.

Figure 1.2: Onshore and offshore new installation and total installation, percentages related to different countries [1].

1.1.2 Market Outlook

The global wind industry is expected to see even more growth in the coming years, with a CAGR of 6.6% for the next five years. Despite the second highest level of installed capacity in history in 2021, GWEC Market Intelligence predicts that 557 GW of new capacity will be added in the next five years under current policies, which equates to more than 110 GW of new installations each year until 2026.

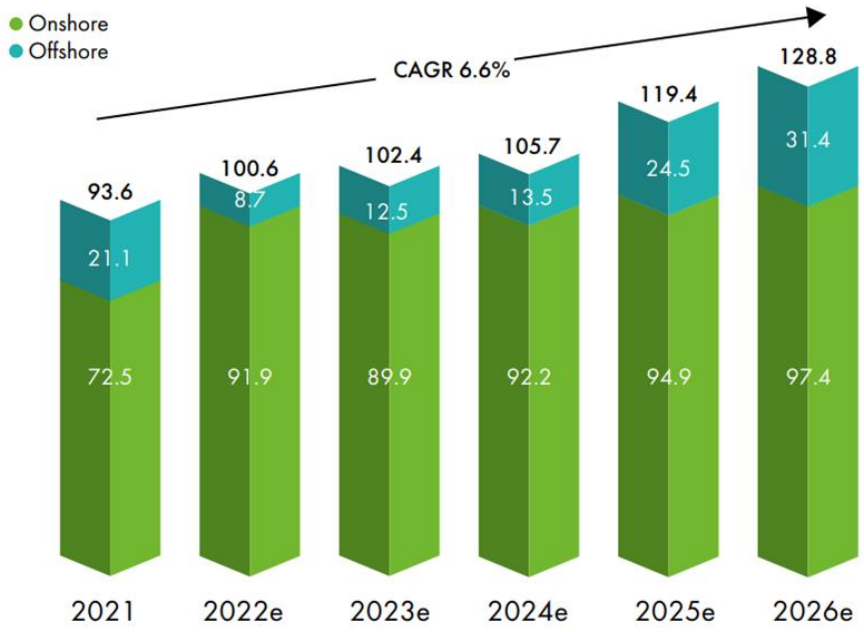

Figure 1.3: New wind power installation outlook 2022-2026. CAGR is the compound annual growth rate [1].

The global onshore wind industry is expected to see a compound annual growth rate (CAGR) of 6.1% in the next five years. The average annual installation is expected to be 93.3 GW, with a total of 466 GW likely to be built between 2022 and 2026.

The global offshore wind market outlook looks extremely promising for the medium and long-term. With an expected compound average annual growth rate of 6.3% until 2026 and 13.9% up to the beginning of next decade. GWEC Market Intelligence expects that over 315 GW of new offshore wind capacity will be added over the next decade (2022-2031), bringing the total offshore wind capacity to 370 GW by the end of 2031. 29% of this new volume will be installed in the first half of the decade (2022-2026). The volume of annual offshore wind installations is expected to more than double from 21.1 GW in 2021 to 54.9 GW in 2031, bringing offshore share of global new installations from 23% in 2021 to 32% by 2031.

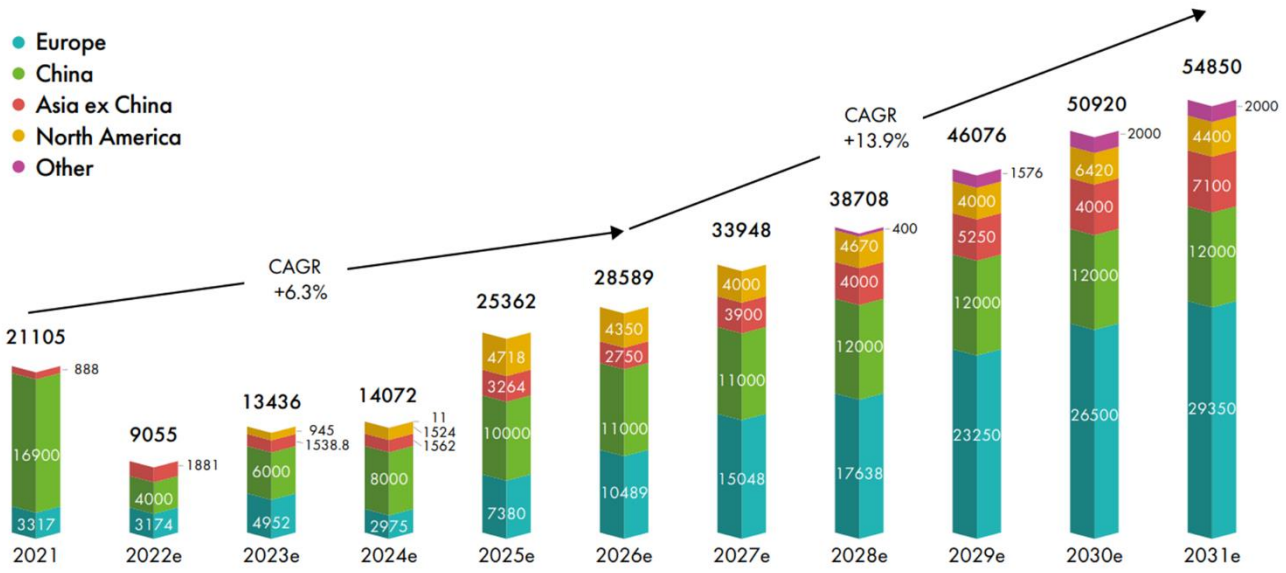

Figure 1.4: Outlook for new offshore installations (MW) up to 2031, divided by Countries [2].

Although global offshore wind market has seen significant growth in recent years, with Europe currently leading in terms of total installed capacity, floating offshore wind technology has lagged behind fixed-bottom installations, with only 121.4 MW of net capacity in operation worldwide as of the end of 2021. Despite this, significant progress has been made in the development of floating wind, with a global pipeline of 120 GW now in place. The UK, Portugal, Japan, Norway, and China currently lead in terms of total gross installations.

GWEC Market Intelligence predicts that 18.9 GW of floating wind capacity will be built globally by 2030. The majority of new installations are expected to come online in 2027-2031, with Europe contributing 59.2% of total installations, followed by Asia at 29.4% and North America at 11.4%. By 2031, a total of 28.8 GW of floating wind is expected to be installed worldwide, bringing its contribution to total offshore wind installations to 7.8%.

1.1.2.1 Europe Market Outlook

The European continent remains the leader in the offshore wind industry, with a mature supply chain established in countries surrounding the North Sea and Baltic Sea. Europe is still the largest regional market in terms of total offshore wind installations as of the end of 2021. However, Asia has taken the lead in terms of new installations in 2020. Market growth in Europe is expected to be slow in the near-term (2022-2024) due to lower activity in established markets, but it is expected to accelerate from 2025 onwards. The total capacity to be added in the next ten years (140.8 GW), 79% will be built in the second half of the decade (2027-2031).

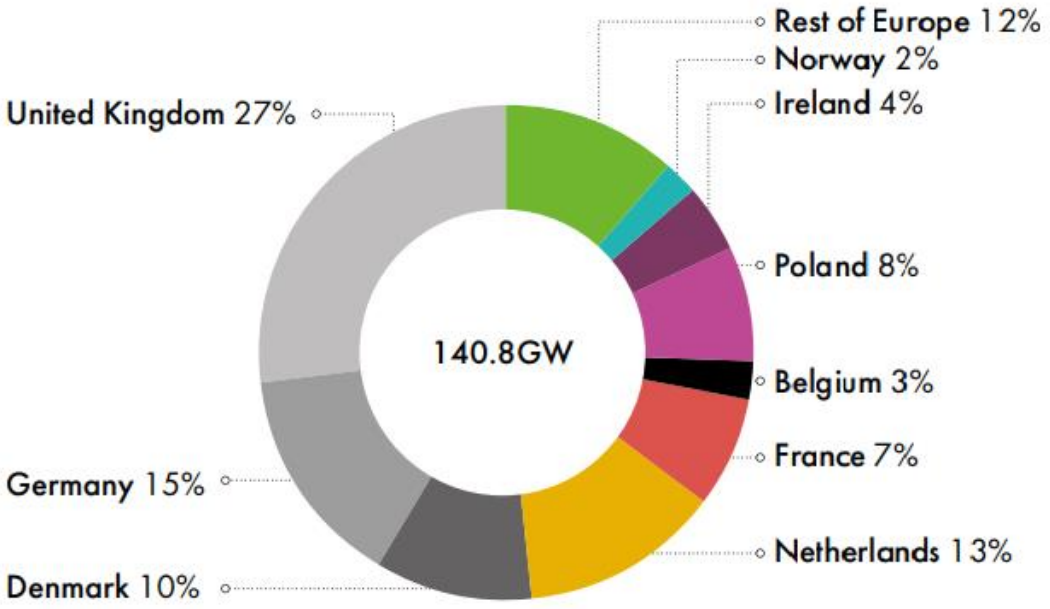

Figure 1.5: Europe's offshore total capacity added between 2022 and 2031, divided by countries [2].

1.2 Wind Turbine Technology

Wind turbine technology is a complex field involving several technical disciplines such as aerodynamics, mechanics, meteorology, and electrical engineering. It deals with the generation, transmission, and integration of wind energy into power systems. This technology has made considerable progress over the years and is now considered the most reliable and promising renewable energy technology. The size and design of wind turbines have evolved considerably from the small kilowatt turbines of the 80s to the larger multimegawatt turbines of today. The design has also moved from being based on conventions to being optimised for the specific operating conditions and market environment.

There are different types of wind turbines, with a wide range of sizes and power output. The main subdivision is between Horizontal Axis Wind Turbines (HAWT) and Vertical Axis Wind Turbines (VHWT), which differ in the orientation of the axis of rotation of the blades. The focus here will be on HAWTs, given their greater prevalence and the fact that floating wind turbines (the subject of this thesis) only exist today with horizontal axis.

In the following sections, main components of this type of turbine will be listed, briefly explaining their function.

1.2.1 Wind Turbine Components

The main components of a wind turbine are the rotor, nacelle, tower and foundation (or the floating platform in the case of floating offshore wind turbines). The rotor consists of the blades and the hub, through which connection to the nacelle is made. The nacelle contains various components that contribute, with their specific function, to the wind energy conversion process into electricity.

1.2.1.1 Rotor

The wind turbine's aerodynamic rotor converts the wind's power into kinetic energy by means blades, which are attached to the hub via mechanical joints. Modern wind turbines typically have two or three blades (because it leads to optimal performance/cost combination) made of fiberglass-reinforced polyester or fiberglass reinforced with carbon fiber, through which the highest possible strength-to-weight ratio is sought. The blades in older turbines were fixed to the hub, while in newer ones they can rotate around their longitudinal axes to optimize performance and prevent damage in high wind conditions.

The design of the blades in a wind turbine is crucial for reducing noise, surviving high winds, and producing high energy output. The main concept behind this is the use of two types of aerodynamic forces generated by airflow over a surface: drag forces in the direction of the airflow and lift forces perpendicular to the wind. To harness lift, the blades have a specially curved airfoil shape, similar to those on airplane wings.

The curve causes the wind to move faster along one side of the blade, creating a pressure difference that results in a net force perpendicular to the wind. This causes the blades to turn. The orientation of the blades, called the rotor pitch, must be precisely adjusted to maintain optimal performance as wind speeds change.

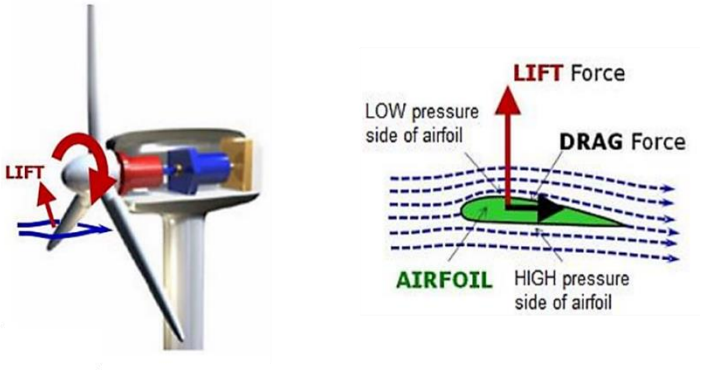

Figure 1.6: Working principle of a wind turbine.

1.2.1.2 Nacelle

The nacelle is a protective covering that encloses the turbine's electrical and mechanical components. Made of materials such as fiberglass, it houses the main drive shaft, gearbox, and generator. Additionally, it includes the blade pitch control, a hydraulic system that adjusts the angle of the blades, and the yaw drive, which regulates the turbine's position in relation to the wind.

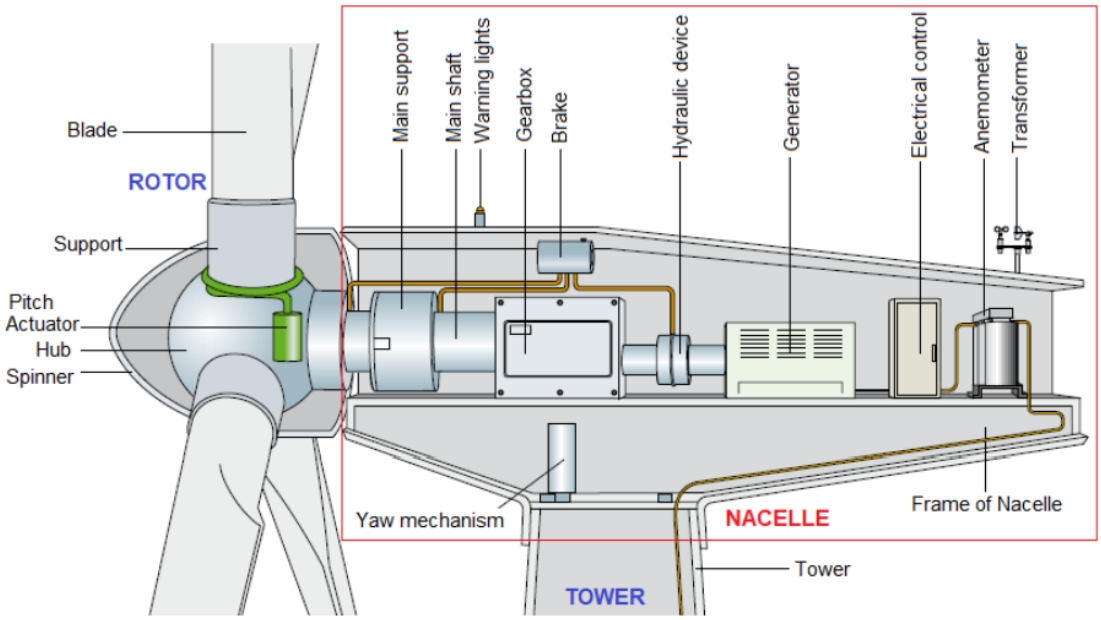

Figure 1.7: Section of a Nacelle, in which its main components are visible.

Main nacelle's sub-components are the drivetrain, the generator, the braking unit, the yaw mechanism and the control unit. All of them will be briefly explained below.

Drivetrain

There are mainly two types of turbines in terms of drivetrain, in the most common there is a transmission of mechanical power from the rotor to the generator via two shafts connected with a gearbox. This is used to switch between the low angular speed of the rotor and the higher speed required for proper operation of the electric generator. The gearbox is a bulky, heavy element and may cause problems from the point of view of the system's service life, so lately, direct-drive wind turbines in which the gearbox is not present are becoming increasingly common. In this type of turbine, generator's rotor is integral with turbine's rotor, while the generator stator is fixed with nacelle. The generator has numerous poles in order to feed power into the grid at the correct frequency despite the typically low angular speed of the rotors, which is around 6-9 rpm. Figure below shows the two solutions mentioned, where it can be seen that in the direct drivetrain case the components inside the nacelle are fewer, thus being able to reduce its size.

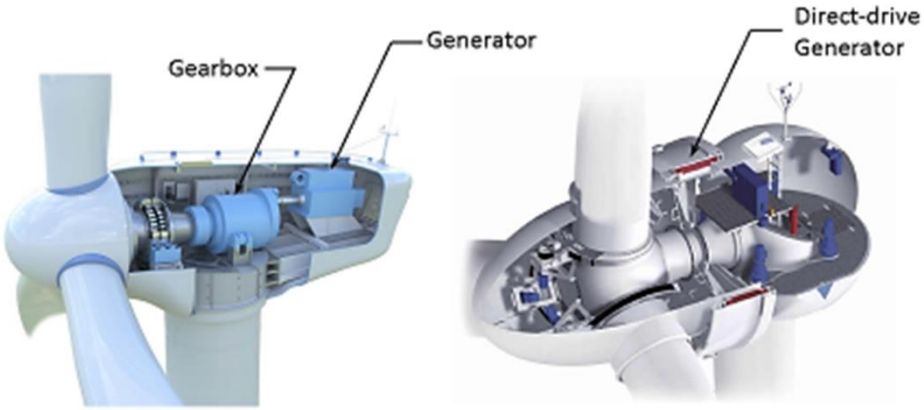

Figure 1.8: Wind turbine nacelle in two different configurations. On the left there is the case with gearbox, while on the right there is the case with direct transmission.

Generator

The electric generator is the component through which the mechanical energy of the rotor is converted into electrical energy to be fed into the electrical grid. The most commonly used generator types in wind turbines are the fixed speed squirrel-cage induction machine, the wound-rotor induction machines with variable resistance (through which it is possible to decouple the speed and torque of the generator, so as to have turbines with variable-speed rotor while maintaining the correct electrical frequency), the doubly fed induction machines (DFIM) with power electronics (through which the above-mentioned decoupling can reach wider ranges than in the previous case) and, finally, the permanent magnet generators (PMG) or other synchronous generators.

Braking Unit

The braking unit, in most cases, consists of a mechanical brake made up of a perforated disc and clamps. It is not used to slow down the rotor in normal operations, for which large breaking torques would be needed, leading to overheating and wear of the unit. In normal operations, an aerodynamic brake is used, which consists of rotating the blades around their own axis in order to manage aerodynamic torque and thus rotor speed. The mechanical brake, on the other hand, is mainly used to stop the rotor after it has been aerodynamically braked previously (for example during maintenance operations), or can be used in the event of a failure in the blade rotation system.

Yaw Mechanism

The yaw mechanism ensures that the turbine always faces into the wind. Since the wind direction is constantly changing, the rotor must follow the wind and adapt its orientation to obtain maximum power. The yaw mechanism helps to change the direction of the nacelle with respect to the wind direction. Usually, a servo-mechanism driven by a wind direction sensor controls the yaw motor, keeping the turbine correctly oriented. However, many turbine designs are limited in their yaw movement because the cables carrying the power and/or control signals from the bottom to the top of the tower are usually coiled together and can be twisted during the rotation of the nacelle. If these cables are twisted too much, they can be wrenched from their anchorages resulting in extreme damage. A limit switch is used to notify the controller when the twist limit is reached.

Control Unit

The control of a wind turbine is a challenging task due to the unpredictable nature of wind. Rapid fluctuations in wind speed can occur at any time, requiring constant adjustments to the turbine's power output. To overcome this, modern wind turbine generators require advanced and reliable controllers. These controllers monitor various parameters such as wind speed, direction, shaft speed, torque, and temperature to ensure optimal performance.

The control system is responsible for several functions, including starting and stopping the turbine, adjusting the rotor's position to face the wind, regulating the power output by adjusting the pitch of the blades, and implementing safety measures. An anemometer is used to measure the wind speed and intensity, which can range from simple instruments to complex computer-aided systems. These measurements are used by the turbine's electronic controller to start the turbine when wind speed reaches a certain threshold and to stop it if wind speeds become too high to protect the turbine and its surroundings.

A weathervane or wind vane is used to determine the wind direction and communicate with the yaw drive to orient the turbine accordingly. The yaw drive, powered by a motor, keeps the turbine facing the wind in upwind turbines, whereas it is not necessary for downwind turbines as the wind naturally blows the rotor away. A pitch system is used to control rotor speed to optimize the extracted power or for security purposes. Finally, the controller manages the generator torque to optimize the power too. The electronic controller also performs safety functions, such as automatically shutting down the turbine in case of overheating or other malfunctions and alerting the turbine operator via a telephone modem link.

1.2.1.3 Tower

The wind turbine tower supports the rotor and nacelle, its height is important because taller towers allow turbines to capture more energy and generate more electricity, as wind speed increases with height. The tower also raises the turbine above any air turbulence near the ground caused by obstacles such as hills, buildings, and trees. On average, the tower is around 50 meters high, with the tallest reaching over 200 meters. The height is determined by the location, rotor diameter, and wind speed conditions. In general, the tower height is slightly taller than the rotor diameter for medium and large turbines, and taller than the rotor diameter for small turbines to avoid poor wind speeds from being too close to the ground. Increasing the tower height can greatly increase power output for a small investment. The average weight of a wind turbine tower is over 40 tonnes, and the tower can account for more than 10% of the total cost.

The main design considerations for the tower include minimizing mass, maximizing the stiffness-to-mass ratio and minimizing vibration. It is important to design the tower so that its natural frequencies are different from the excitation frequency due to the loads from the blades. Furthermore, when designing the tower, the maximum possible loads, e.g. in the event of strong wind gusts, must be taken into account, as well as fatigue loads and thus its service life.

Three most common types of wind turbine tower are lattice tower, guyed pole tower and tubular tower, briefly explained below.

Lattice Tower

Lattice towers are constructed using welded steel profiles and are made up of steel rods that are connected to form a lattice-like structure.

They have a similar appearance to traditional communications towers, are less expensive as they require less material than a free-standing tubular tower of comparable stiffness.

These towers are also easy to transport and install in sections. However, their appearance is often considered a downside, and they are not as commonly used in contemporary wind turbines.

Figure 1.9: Example of lattice tower wind turbine.

Guyed Pole Tower

This type of tower is only used for small turbines, they are held in place by four guy ropes (at 45°), with one rope that can be released to lower the tower for maintenance. These towers can be tilt-up or fixed and are sturdy and cost-effective when correctly erected. However, they require more space for guy wires surrounding the tower. The radius of the guy wires should be one-half to three-quarters of the tower height.

Tubular Tower

Tubular towers, made of rolled steel plate or concrete, are the most common type of tower used in large wind turbines. They are fabricated in sections of 20-30 meters, with flanges at the end that are bolted together on site.

The towers are typically coated with a zinc-based finish and epoxy and urethane layers for corrosion resistance. Most modern turbines have towers with a circular tubular steel design, with a diameter of 3 to 5 meters and a height of 75 to 110 meters.

These towers allow for safe access for repair and maintenance but are more expensive than lattice towers.

Figure 1.10: Example of tubular tower wind turbine.

1.2.1.4 Foundation - Platform

The components analysed above are similar for all types of turbines, i.e., onshore, fixed offshore and floating offshore turbines, but when it comes to the element on which the turbine tower is mounted, the differences are evident in the three types mentioned above. In general, we speak of foundations for the first two types, and of a floating platform for the third. Below are the main characteristics of foundations and platforms.

Onshore Wind Turbine Foundations

Wind turbines are large structures that must be able to withstand high wind speeds and are anchored to the ground using deep, excavated holes. The foundations of these turbines are made of concrete with steel reinforcements, and the design of the foundation is determined by the soil conditions and wind strength.

When the soil has good mechanical properties, a spread foundation is used, that consist of a big plate through which the loads are transferred to the ground.

When the soil properties aren't sufficient for a spread foundation, a god solution can be to install piles to conduct load at a greater depth in the ground.

Figure 1.11: Onshore wind turbines' spread and piled foundations [37].

Fixed Offshore Wind Turbine Foundations

Fixed offshore wind turbines are turbines located in the shallow sea and fixed to the seabed by means of different types of foundations, but in all cases, it is necessary that the seabed does not exceed a certain depth due to technical and economic aspects. The figure below shows the main types of foundations currently in use: monopile, gravity base, jacket, tripod and tripile.

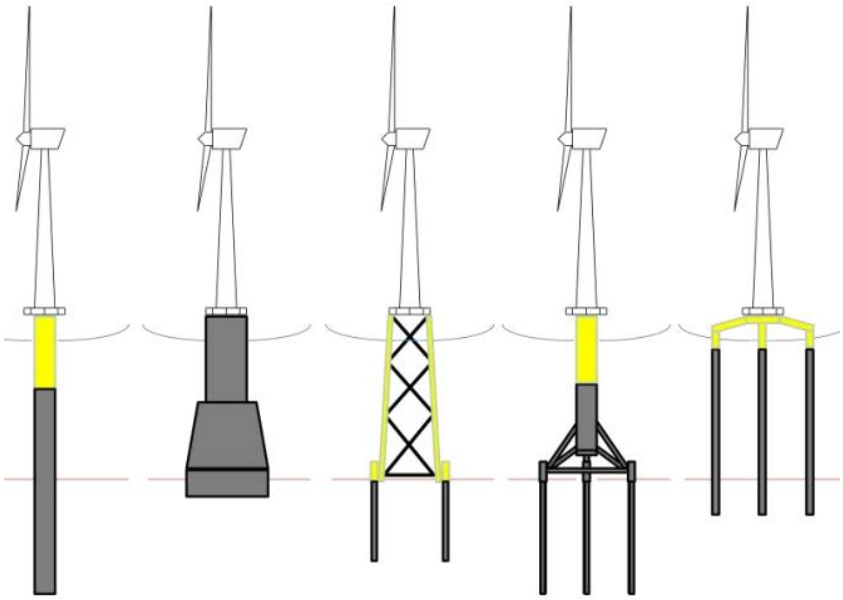

Figure 1.12: Types of support structure concepts. From left to right: monopile, gravity base, jacket, tripod, tripile, spar and semisubmersible platform [38].

The above-mentioned foundations are described below.

Monopiles

The most typical offshore wind support construction is a monopile. On this support structure solution, more than half of all projects are constructed. The North Sea oil and gas industry, where monopiles have been employed for many years, is where this technology's origins can be found. This has made it possible for them to be implemented in the offshore wind industry with little risk and a variety of already-in-use procedures. The monopile's architecture is quite straightforward and consists of a foundation pile that is cylindrical and resembles the turbine tower. The erosion that water currents cause on the pile at seafloor level is a crucial factor to take into account during the monopile installation procedure. The length of the pile exposed to hydrodynamic loads increases as a result of the tendency of water flow to lower the layer of soil covering the pile. To solve this problem, stones are used to cover the foundation and prevent soil erosion. In general, monopiles are the most often utilised support structure because to the comparatively straightforward production and installation procedure. However, they have some water depth restrictions; in Europe, monopile support structures are not practical for waters deeper than 40 metres.

Gravity base

The tower and turbine are secured in place by gravity base support structures, which rely on their own weight and ballast. The tower is positioned on base constructions that are composed of steel-reinforced concrete. Regarding the base structure, there are several designs. The gravity base has the benefit that no drilling or hammering is necessary during the installation process. However, the gravity-based support structure requires a horizontal seafloor that has been covered in crushed stone, which ultimately raises the overall cost. Additionally, the overall cost typically varies with water depth, with levels deeper than 20 metres making it impossible to achieve an economically viable level. Ballast is filled on the gravity foundation once it has been put on the seabed. One benefit of this support structure is that it has a lifespan of more than 100 years with minimal maintenance after the installation is complete.

Jackets

The offshore oil and gas business first used jackets as structures. A structure with three or four legs joined by thin bracing is called a jacket. Since they are all tubular, welding is used to connect them all. Similar to monopiles, jackets require a support structure to hold up the wind turbine tower. One of the crucial processes in the manufacture of the jacket is melting the transition piece with the substructure. In the near future, it may be possible to observe variants or new concepts of jackets, such as those in which the weight of the transition piece is lowered, the jacket and the tower are combined into a single element, or twisted jackets are employed in place of the traditional approach.

Tripod

A tripod is a three-legged steel tube structure, these legs transition to the wind turbine tower by being joined to the main tube in the middle of the structure. Through sleeves at each leg's end, foundation piles are used to drive the substructure into the seabed. As part of the installation process, multiple tripods are loaded onto a barge and towed to the installation site, where a crane lifts and guides the support structures into place. At this point, each sleeve is penetrated by a pile, and the spaces between the piles and sleeves are filled with concrete or grouting. There is no need to prepare the seabed.

Tripile

The tripile is a modernised version of the conventional monopile. It is composed of three separate tubular steel piles, with a three-legged transition piece sitting on top of each one and linking it to the turbine tower. This construction has the benefit of being easily adjustable to account for changes in water depth. The transition piece proportions may be kept the same, but the pile dimensions may be changed to accommodate the site. The piles are first pushed into the seafloor during installation. The transition piece is then positioned on top with each leg end pointed into a pile as the piles' tops rise above the water. Making sure that the transition piece fits inside the piles at this stage of installation demands precision.

Floating Offshore Wind Turbine Platforms

As a result of this industry's propensity to move into deeper waters, floating support structures have become more common in the offshore wind market. Offshore wind parks have gradually entered deeper and deeper oceans since the first gravity base was built, whether by monopiles, jackets, or tripods. The wind's superior quality at such sites is the primary driver of this transformation. However, some nations must dive deeper due to the topology of their bottom in addition to the need for a greater wind resource. There are now three major categories of floating foundations, each of which is based on a distinct approach, the spar buoy monopile, the semi-submersible platform and the tension leg platform. Figure below shows mentioned concepts.

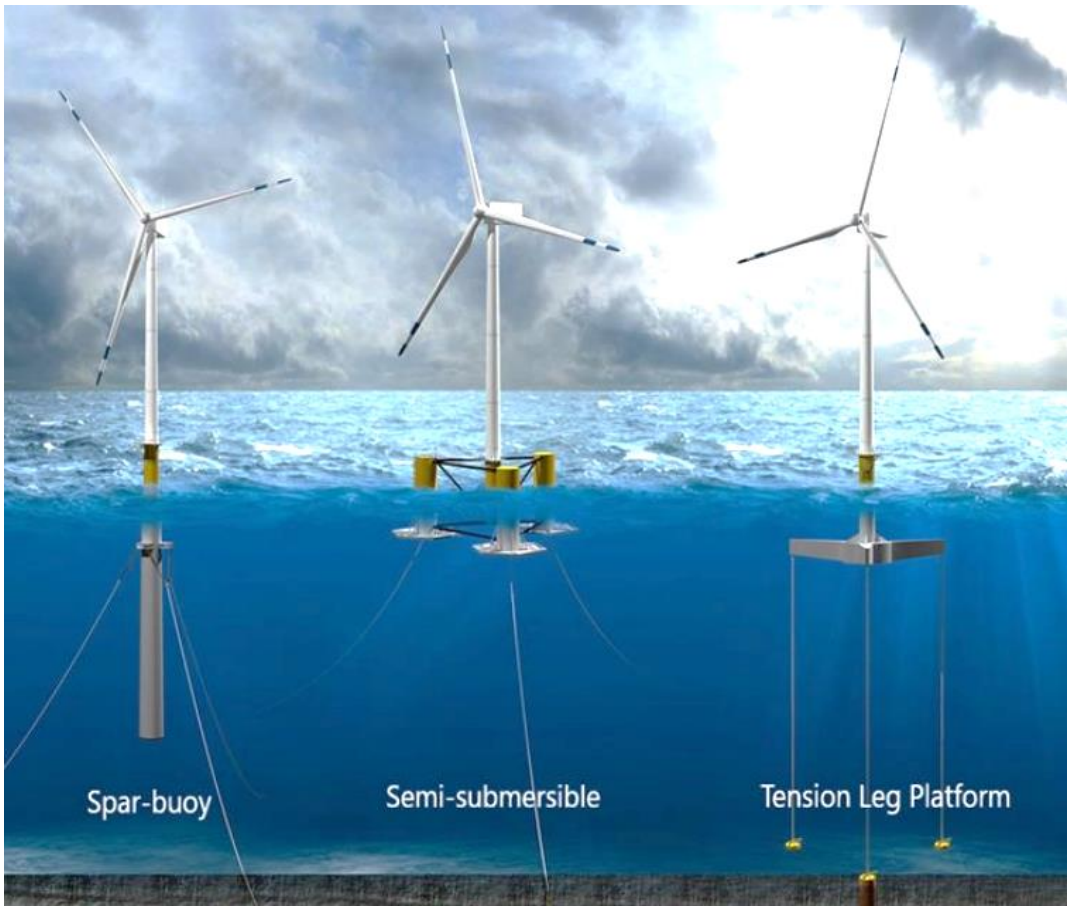

Figure 1.13: Three main types of floating wind turbine platforms. From left to right: Spar-buoy, Semi-submergible, Tension Leg Platform.

In the next page there is a briefly explanation of these platform concepts, with their advantages and disadvantages, especially in terms of production and installation costs.

Spar-buoy

The Spar-buoy design is built on a cylindrical body made of steel and concrete. The construction is ballast-stabilized, and the stability is obtained by having the center of gravity lower in the water than the center of buoyancy. These floating foundations are ballasted with concrete and seawater and are typically distinguished by low water plane areas. These constructions also have a substantial draught, which maximizes stability and reduces heave motions. This technique draws its inspiration from the offshore oil business, where spar-buoy-based platforms are employed in 1000 m of water.

The catenary, which is often composed of steel chains and/or wires, is the mooring system that is typically used with spar-buoy structures. The catenary's weight and curved shape hold the floating platform in place. The considerable draught of the structure additionally necessitates exact installation and shipping procedures. In fact, because the port's depth is smaller than the draught, it is essential to tow the structure to the site in a horizontal position, where a specialized vessel will complete the installation in deep water. The installation expenses of the structure may rise if specialized boats, such as heavy lift vessels, are used for placing.

On the other hand, because the construction doesn't have any complicated pieces, the manufacturing expenses are not high. Additionally, since the metal only needs to have high structural stability and corrosion resistance, the cost of materials, and specifically the metal, is not prohibitive.

The spar-buoy idea has the following benefits:

- Low installed mooring cost;
- Simple design that makes this design acceptable for serial fabrication process;
- Tendency for lower critical wave-induced vibrations, which overall gives high stability.

In contrast, the following are the key drawbacks of these concepts:

- Needs deeper water (>100 meters) than other concepts;
- Offshore operations require heavy-lift vessels and can currently only be done in relatively sheltered, deep water;
- The installation of the wind turbine on the foundation cannot be done in port but must be done offshore.

Semi-submersible

The semi-submersible structures are composed of three big columns joined to one another by connecting braces or submerged pontoons. The wind turbine may be fixed to one of the columns, or it may be placed in the geometric center of the column tubes and supported by lateral bracing components. With the platform tilted, the leeward portion of the platform has a bigger submerged volume, and the windward portion of the platform has a lower submerged volume than in the equilibrium state, which helps the semi-submersible design achieve the desired static stability. This indicates that the buoyancy force is greater in the leeward area. As a result, the restoring moment that is required to offset the wind inclining moment is created.

Semi-submersible concepts are kept in place by mooring lines to prevent the platform from drifting away due to the impact of the wind, waves, and marine current forces. Semi-submersible offshore wind turbines can be installed considerably more easily, and they don't need specialized vessels because those are just needed for towing. These attributes result in a greater fabrication cost than with previous floating designs, but with reduced shipping and installation expenses.

The semi-submersible concept has the following benefits:

- Can be built on land or in a dry dock;
- Fully equipped platforms, including turbines, can float during transport with draughts under 10 meters;
- Can be transported to the site using conventional tugs, resulting in low transport costs;
- Can be used in water depths up to about 40 meters, providing high flexibility in the range of suitable locations related to sea depth;
- Lower installed mooring costs;

On the other hand, these notions' primary drawbacks are:

- Tendency for higher critical wave-induced motions;
- Tendency to employ more material and larger structures;
- Complex fabrication when compared to other concepts, especially spar buoy.

Tension Leg Platform

Tension leg platform (TLP) designs have a high degree of buoyancy and consist of a central column and radial arms attached to tensioned tendons that anchor the structure to pile or suction anchors. In reaction to wind and wave loads, the increased tension in the windward leg as opposed to the leeward leg provides a restoring moment that balances the inclining moment caused by the wind turbine aerodynamic thrust. It has a shallow draught that is larger than a semi-submersible design but smaller than a spar. As they can be put in moderately shallow to extremely deep seas, TLPs have significant water depth flexibility.

The TLP concept has the following benefits:

- Low mass;
- Tendency for less severe wave-induced vibrations;
- Assembly on land or in a dry dock.

The primary disadvantages of these principles, however, are:

- More expensive installed mooring compared to other floating structure designs;
- Depending on the design, a specific purpose vessel may be needed.
- Harder to keep stable during shipping and installation.

2 Case Study Description

The objective of this thesis is the realisation of an MPC (Model Predictive Control) controller for Floating Offshore Wind Turbines (FOWT), which will be implemented in such a way that it can be adapted to different types of turbines and platforms by simply varying the characteristics of the physical system under consideration. In this case, it was decided to use, for the development of the system model and therefore of the controller (which is based on the latter), a reference turbine (including floating platform) whose design was developed by the National Renewable Energy Laboratory (NREL), the Technical University of Denmark (DTU) and by the University of Maine (UMaine).

Reference turbine means the simplified design of a turbine that does not necessarily have to reflect a specific real-world model, but is used, for example, for preliminary studies of such a system. For this reason, the design provided is simplified, i.e. only the main features are included and not, for example, details concerning structural aspects or details of the various secondary components such as the braking unit, cooling unit, etc. Reference wind turbines play a variety of functions within the wind industry; they are used as baselines for studies that investigate new technologies or design processes as open benchmarks that are set using publicly available design criteria. Reference wind turbines also facilitate collaboration between industry and outside researchers because of their open architecture. Industry can safeguard its intellectual property while still collaborating with outsiders on the development of sophisticated technology by adopting a reference turbine. Finally, they provide a starting point and educational platform for people new to wind energy.

The NREL 5-MW turbine was the first widely used reference turbine and is still employed by many studies today. However, to keep up with the technological evolution of floating wind turbines, which in the coming years will have a nominal power of between 10 and 20 MW, a reference turbine with a nominal power of 15 MW, called "*IEA (International Energy Agency) Wind 15-Megawatt Offshore Reference Wind Turbine*", has been developed by the aforementioned institutions, and such a turbine will be used in this thesis. Specifically, the National Renewable Energy Laboratory (NREL) and the Technical University of Denmark (DTU) have developed the turbine, which can be conceived as part of a fixed offshore device (coupled with a suitable platform) or can be coupled to a floating platform to be able to process FOWTs. In this case, since we will be dealing with a floating turbine, reference will be made to the VolturnUS-S platform, developed by the University of Maine (UMaine).

In the following pages the main characteristics of the chosen turbine and platform will be listed and briefly explained. More information regarding the IEA Wind 15-Megawatt Offshore Reference Wind Turbine or the VolturnUS-S platform can be found here [4], [5].

Figure 2.1 shows the case of study system, composed by the abovementioned IEA 15 MW Offshore Wind Turbine mounted on the VolturnUS-S platform.

Figure 2.1: Illustration of the case of study system, i.e., the IEA 15 MW Offshore Wind Turbine mounted on the VolturnUS-S platform [5].

The IEA 15 MW wind turbine is a direct-drive machine, with a rotor diameter of 240 m and a hub height of 150 m, while the VolturnUS-S platform is a four-column, three-radial and one central, steel semisubmersible platform kept in place through a three-line catenary mooring array that is deployed in 200 meters (m) of water. Figure 2.2 shows orthogonal projections of the system and its main measures.

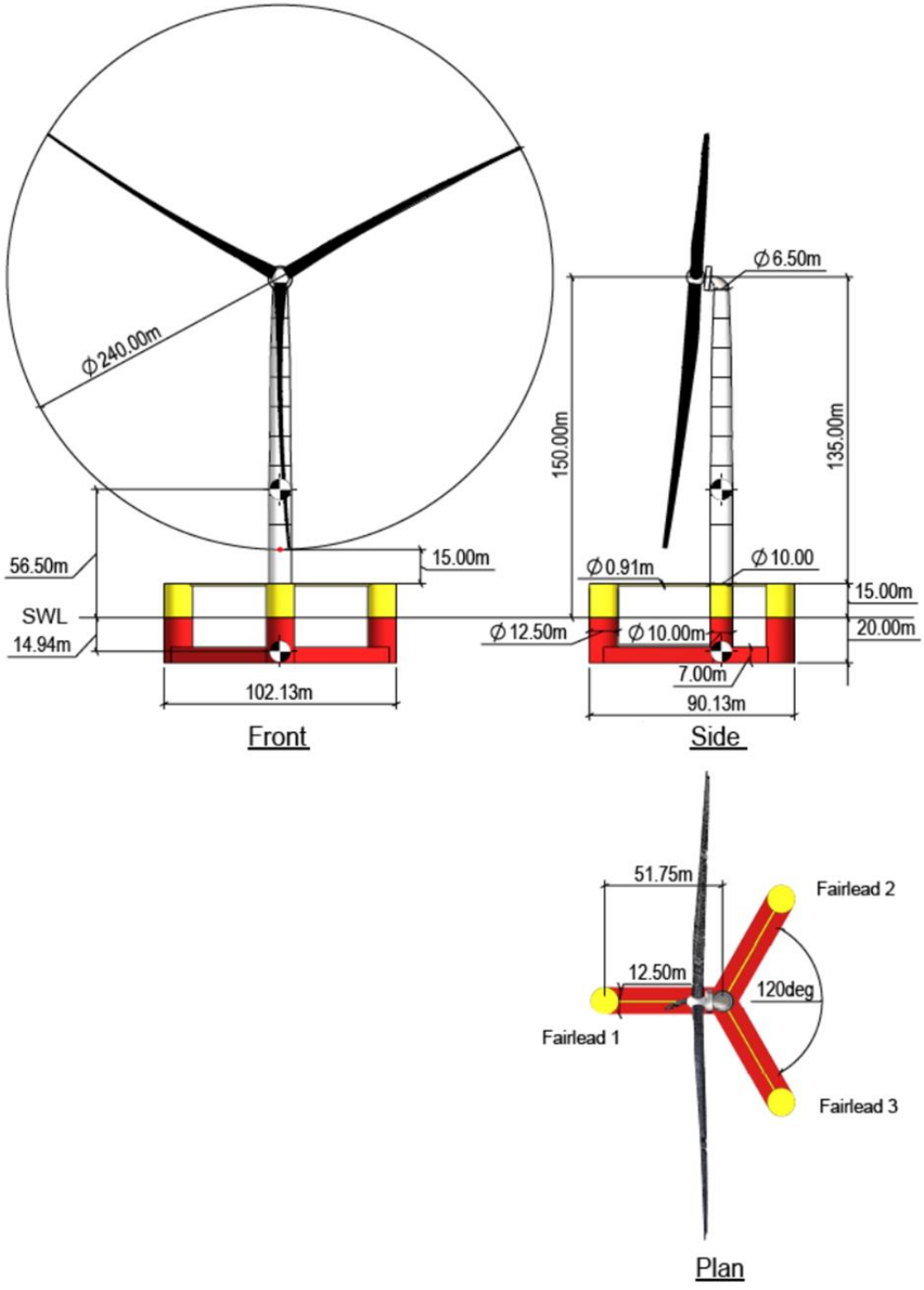

Figure 2.2: Orthogonal projections of the IEA 15 MW Offshore Wind Turbine mounted on the VolturnUS-S floating platform and their main measures [5].

Figure 2.3 shows the floating offshore wind turbine reference coordinate system used from now on; SWL means till water level; surge, sway and heave are the displacement of the origin of the system respectively along x, y and z directions; finally, roll, pitch and yaw are rotations respectively around x, y and z axes.

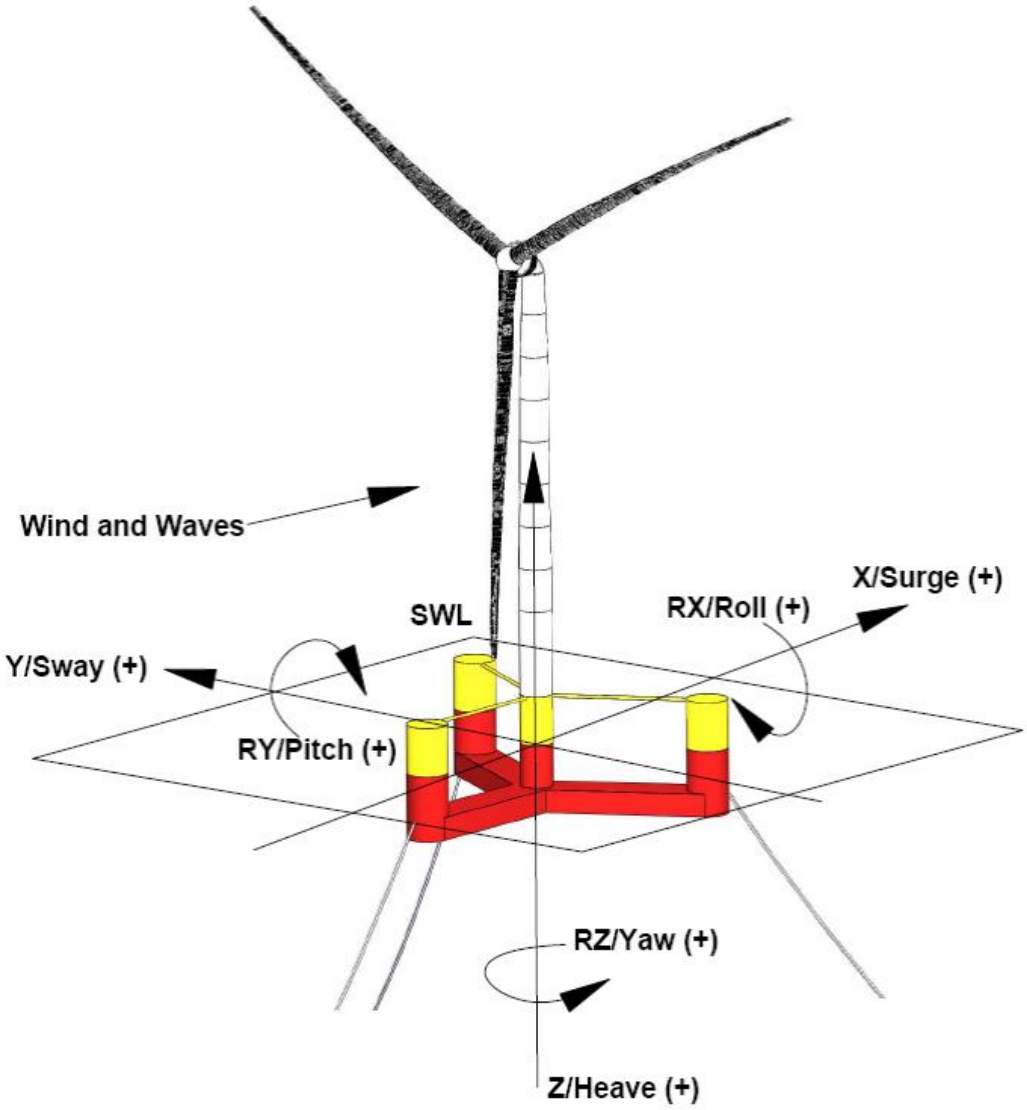

Figure 2.3: Floating offshore wind turbine reference coordinate system [5].

In the next sections, the characteristics of the system will be presented, first focusing on the turbine, and then analysing the floating platform.

2.1 IEA 15 MW Wind Turbine

The turbine considered is a direct drive turbine with a rated power of 15 MW and variable rotor speed. The rotor speed and extracted power are adjusted by varying the torque exerted by the generator and the collective blade pitch angle. The main characteristics of the turbine are shown in Table 2.1. Cut-in and cut-out speeds are defined as the wind speeds below which the turbine is inactive and above which the turbine enters a safe position to limit the loads on the blades. Rated wind speed, on the other hand, is the nominal wind speed from which rated power begins to be produced.

Parameter	Units	Value
Power rating	MW	15
Rotor orientation	-	Upwind
Number of blades	-	3
Control	-	Variable speed, variable collective pitch
Cut-in wind speed	m/s	3
Rated wind speed		10.59
Cut-out wind speed	m/s	25
Rotor diameter	m	240
Hub height	m	150
Hub diameter	m	7.94
Hub overhang	m	11.35
Drivetrain	-	Direct drive
Design tip-speed ratio	-	9
Minimum rotor speed	rpm	5
Maximum rotor speed	rpm	7.56
Maximum tip speed	rpm	95
Tilt angle	deg	6
Precone angle	deg	-4
Blade mass	t	65
Rotor nacelle assembly mass	t	1017
Tower mass	t	860

Table 2.1: Main characteristics of IEA 15 MW offshore wind turbine.

2.1.1 Blade Properties

This 15 MW IEA Wind reference turbine has a blade length of 117 m, a root diameter of 5.2 m, and a maximum chord of 5.77 m at around 20% of the span. The blade's overall mass is around 65 tonnes, and it is intended to have a power coefficient, or C_P, of 0.489. The power coefficient is the ratio of produced to available power.

The blades were created using airfoils from the DTU FFA-W3 series. These airfoils are well-known and openly accessible. In order to offer more room at the tip, the blade was constructed with a substantial pre-curvature away from the tower, with a 4 m distance between the chord line at the tip and the root.

The blade's structural design is quite conventional, with reinforcements along the leading and trailing edges and two primary load-bearing spars positioned in a straight line linking the root and tip. These spars are arranged with one on the suction side and the other on the pressure side of the profile. The carbon fiber used in the spars ensures maximum stiffness while being lightweight. Two shear straps linking the pressure and suction sides of the blade are fastened to the main spars, which span 10% to 95% of the aperture. Uniaxial glass fibre reinforcements were also added to the leading and trailing edges to give further edge stiffness.

The table below provides the main features of the turbine blades.

Parameter	Units	Value
Blade length	m	117
Root diameter	m	5.2
Max chord	m	5.77
Max chord position	m	27.2
Tip prebend	m	4
Precone	deg	-4
Blade mass	t	65
Blade center of mass	m	26.8
First flapwise natural frequency	Hz	0.555
First edgewise natural frequency	Hz	0.642
Design C_P	-	0.489
Design C_T	-	0.799

Table 2.2: Main features of IEA 15 MW offshore wind turbine's blades.

2.1.2 Tower Properties

The tower is made of an isotropic steel tube, and a large portion of its design was bound by the fact that its initial tower mode falls between the ranges of frequencies at which blades pass in all wind speeds (1P and 3P). With 120-m blades, the tower height was designed so that the hub height reaches 150 m, providing 30 m of ground (water surface) clearance.

2.1.3 Nacelle, Drivetrain, and Hub Properties

The IEA Wind 15-MW reference wind turbine uses a direct-drive layout with an outer-rotor permanent-magnet generator. The assembly consists of a hub shaft supporting the turbine and generator rotors on two main bearings housed on a stationary turret that is cantilevered from the bedplate. The bedplate transmits the hub loads and weight of the rotor, generator, hub, shaft, and turret to the tower.

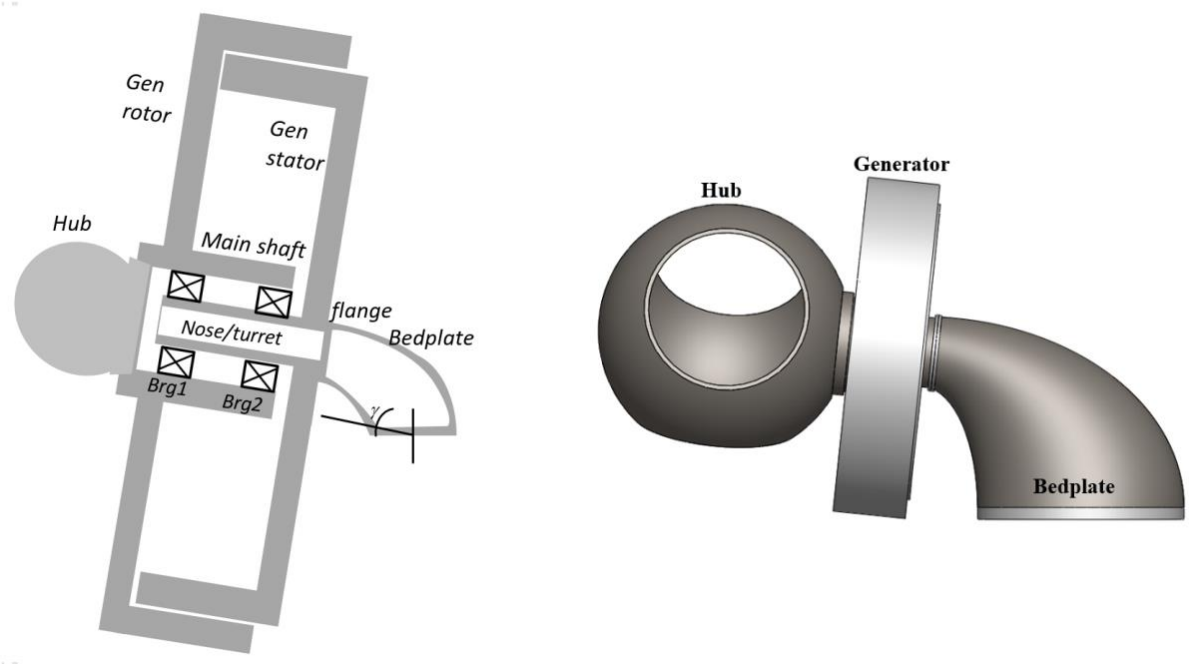

Figure 2.4: Nacelle layout of the 15 MW offshore wind turbine.

The hub's structure is based on a straightforward concept of a spherical hub shell with cutouts for the blades and flange. The hub shell's thickness is intended to withstand the forces produced during an emergency shutdown event.

The main shaft features a 6 degrees tilt and a hollow cylindrical cross section with a constant wall thickness. Two major bearings hold the main shaft and the rotor in place. These two main bearings both feature fixed inner raceways and spinning outer raceways. A turret supported by the bedplate can contain the outer raceways and the bearing housing. It is assumed that the main shaft transmits the entire thrust load to the upwind main bearing. Additionally, the turret has a hollow cylindrical cross section with walls that are cantilevered from the bedplate and have a constant thickness. A 2 m inner diameter was chosen to allow for technician access.

The bedplate is a cantilever beam that is hollow, elliptically bent, and has circular cross sections. A smaller cross portion of the bedplate connects to the bedplate flange, and a bigger cross section connects to the yaw bearing at the top of the tower.

The yaw system bearings are based on double-row, angular contact ball bearings that may be found in the SKF catalogue. The tower top and bedplate base are joined by the yaw system at a 6.5 m diameter.

2.1.4 Generator Properties

The generator structure comprises a radial flux topology machine with a surface-mounted permanent magnet on the outer rotor. This outer rotor design offers a robust and simple structure, efficient production, shorter terminal windings and better heat transfer than the inner rotor configuration. During optimization, only the radial components of the air gap flux were taken into account and no dispersion effects were considered. The optimized design includes 100 pole pairs in an air gap diameter of 10.53 m, a height of 2.17 m and an efficiency of 96.5%. The total weight of the generator is 372 tonnes, including 227 tonnes for the stator and 145 tonnes for the rotor. At least half of the weight is attributed to the structural support.

2.2 VolturnUS-S Platform

Table 2.3 lists the general characteristics of the four-column, steel VolturnUS-S platform, including system masses, dimensions, centres of gravity and buoyancy, and inertias. Three pontoons and three radial struts attached to the bottom and top of the buoyant columns, respectively, connect the fourth central column (the interface between the platform and the tower) to the outer columns. The platform weighs a total of 17854 t when it is on station, of which 3914 t are made up of structural steel, 2540 t are fixed iron-ore-concrete ballast that is distributed evenly and placed at the base of the three radial columns, 11300 t are seawater ballast that floods the majority of the three submerged pontoons.

Parameter	Units	Value
Hull displacement	m^3	20206
Hull Steel Mass	t	3914
Ballast mass	t	13840
Draft	m	20
Vertical center of gravity from SWL	m	-14.94
Vertical center of buoyancy from SWL	m	-13.63
Roll Inertia about center of gravity	kg m^2	1.25E+10
Pitch Inertia about center of gravity	kg m^2	1.25E+10
Yaw Inertia about center of gravity	kg m^2	2.37E+10

Table 2.3: Main features of VolturnUS-S.

The mooring system is made up of three 850-meter-long chain catenary lines; at a depth of 14 m below the SWL, each line is joined at the fairlead to one of the platform's three outer columns. The lines extend radially to anchors at a depth of 200 m and spaced radially 837.60 m from the centerline of the tower, which are evenly spaced at 120 degrees in the surge-sway plane. A studless chain with a nominal diameter of 185 millimeters is used on all lines. The chain size was chosen to keep the system's peak surge-sway offset under 25 m during normal operational conditions.

Parameter	Units	Value
Mooring system type	-	Chain catenary
Line breaking strength	kN	22286
Number of lines	-	3
Anchor depth	m	200
Fairlead depth	m	14
Nominal chain diameter	mm	185
Dry line linear density	kg/m	685
Line unstretched length	m	850

Table 2.4: Main features of VolturnUS-S mooring system.

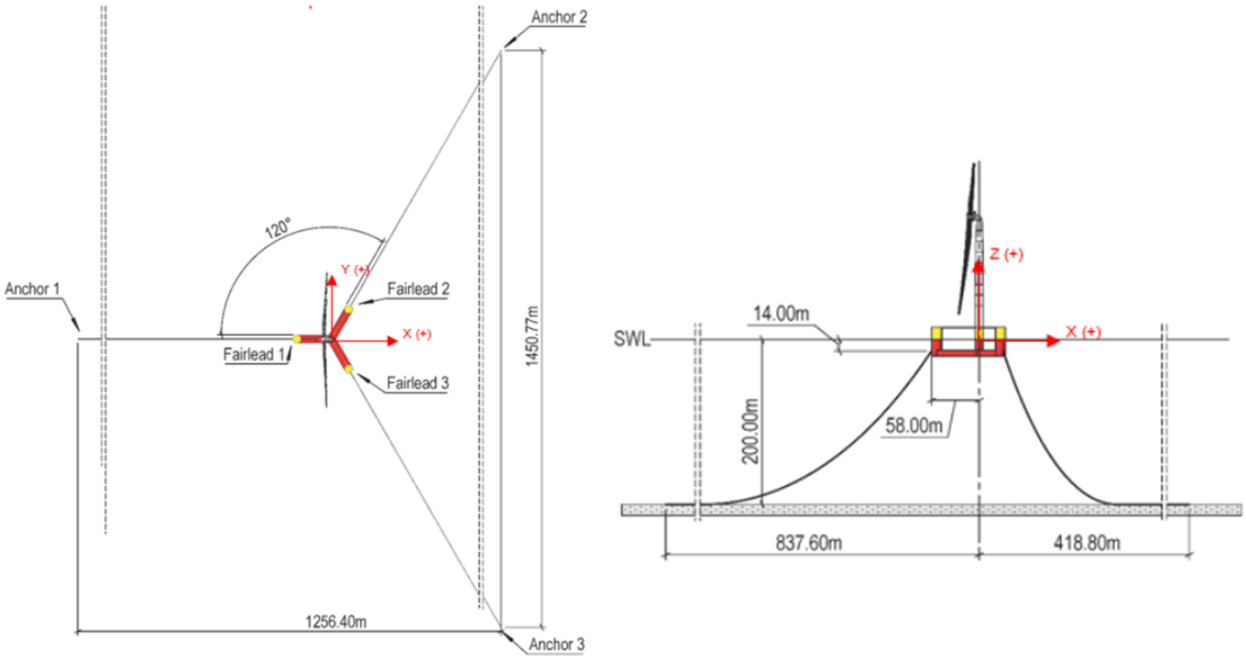

Figure 2.5: Mooring system layout, top and side view [5].

3 Model of the system

In this work, the model adopted for the system is a 4 d.o.f. model, in which the firsts 3 are part of the 6 d.o.f. of the structure and the fourth is the rotational joint between rotor and nacelle. The three d.o.f. of the structure considered are surge, heave, and pitch, according with figure 2.3. The model's inputs are the free flow wind speed field, the excitation forces and torques caused by incident waves, generator torque and blade pitch angle. In this section the equations describing the above-mentioned nonlinear system will be obtained. First, we will find the motion equations of the structure and those related to the rotation equilibrium of the rotor, then the external inputs of the system will be defined. Finally, starting from system's model obtained, a simplified model will be defined, which will be useful for MPC controller's implementation.

3.1 Model equations

3.1.1 Structure equation

From momentum and angular momentum equations $\left(\frac{d\boldsymbol{P}}{dt} = \sum \boldsymbol{F}_{ext} \;,\; \frac{d\boldsymbol{L}}{dt} = \sum \boldsymbol{M}_{ext}\right)$ we can obtain the equations of motion for rigid bodies (considering all six degrees of freedom).

$$\mathrm{m}\left[\boldsymbol{a}_{wt} + \dot{\boldsymbol{\omega}} \wedge \boldsymbol{r}_{\mathrm{cg,wt}} + \boldsymbol{\omega} \wedge (\boldsymbol{\omega} \wedge \boldsymbol{r}_{\mathrm{cg,wt}})\right] = \sum \boldsymbol{F}_e \quad (3.1)$$

$$\boldsymbol{r}_{cg,wt} \wedge m\,\boldsymbol{a}_{wt} + \mathbf{I}\,\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \wedge \mathbf{I}\,\boldsymbol{\omega} = \sum \boldsymbol{M}_{e,wt} \quad (3.2)$$

- $\boldsymbol{r}_{\mathrm{cg},\omega t} = \boldsymbol{r}_{cg} - \boldsymbol{r}_{\omega t}$;
- $\boldsymbol{a}_{wt} = \{\ddot{surge}, \ddot{sway}, \ddot{heave}\}^{\mathsf{T}}$;
- $\dot{\boldsymbol{\omega}} = \{\ddot{roll}, \ddot{pitch}, \ddot{yaw}\}^{\mathsf{T}}$;
- $\mathbf{I} = \boldsymbol{R}^{0-wt} \cdot \mathbf{I}^{wt} \cdot \boldsymbol{R}^{0-wt\,\mathsf{T}}$.

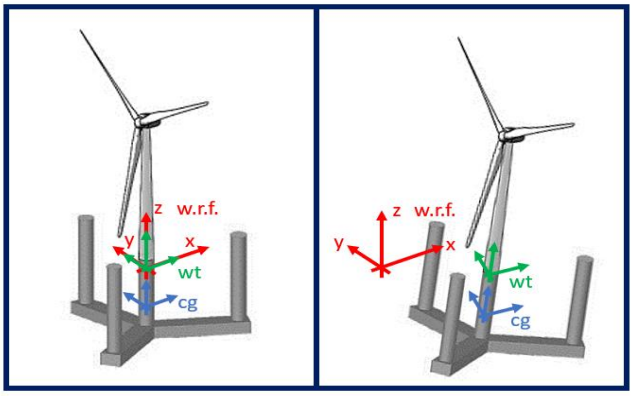

Figure 3.1: Scheme of a 6 d.o.f. floating wind turbine.

All quantities without superscript are expressed in the world reference frame (see Figure 3.1), m is the whole structure mass and $\mathbf{I}^{wt}$ is the structure inertia matrix referred to its center of gravity and expressed in wind turbine (wt) reference frame. $\boldsymbol{R}^{0-wt}$ is the rotation matrix between world reference frame and wind turbine one. In particular, its columns are the base versor of wt-reference expressed in w.r.f. coordinates.

Assuming three d.o.f. model, some simplifications can be made:

- $\boldsymbol{R}^{0-wt} = \boldsymbol{R}_y(p) = \begin{bmatrix} \cos p & 0 & \sin p \\ 0 & 1 & 0 \\ -\sin p & 0 & \cos p \end{bmatrix} \rightarrow \boldsymbol{I}^0_{yy} = \boldsymbol{I}^{wt}_{yy}$;
- $(\boldsymbol{\omega} \wedge \mathbf{I}\,\boldsymbol{\omega}) \cdot \hat{\boldsymbol{j}} = \mathbf{0}$;
- Low speeds → $\boldsymbol{\omega} \wedge (\boldsymbol{\omega} \wedge \boldsymbol{r}_{cg,wt})$ neglected;
- Small-amplitude motion → $\boldsymbol{r}_{cg,wt} \approx \boldsymbol{r}_{cg,wt}{}^{wt}$;

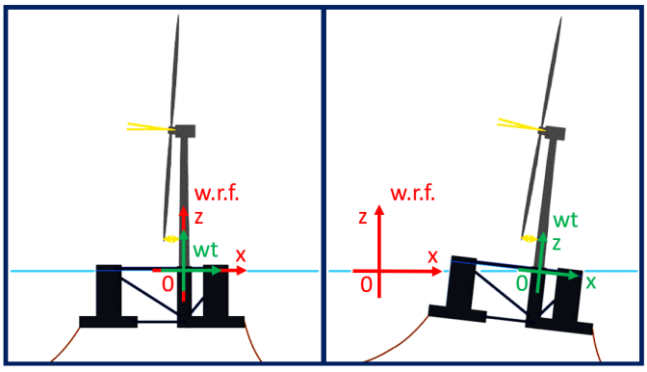

Figure 3.2: Scheme of a 3 d.o.f. floating wind turbine.

Then we arrive to:

$$\begin{bmatrix} m & 0 & m\,z^{wt}_{cg} \\ 0 & m & -m\,x^{wt}_{cg} \\ m\,z^{wt}_{cg} & -m\,x^{wt}_{cg} & I^{wt}_{pp} \end{bmatrix} \cdot \begin{Bmatrix} \ddot{s} \\ \ddot{h} \\ \ddot{p} \end{Bmatrix} = \begin{Bmatrix} \sum F_{x_e} \\ \sum F_{z_e} \\ \sum M_{y_e} \end{Bmatrix} \quad \rightarrow \quad \boldsymbol{M}\,\ddot{\boldsymbol{x}} = \boldsymbol{F}^{\boldsymbol{tot}} \qquad (3.3)$$

$\boldsymbol{F}^{\boldsymbol{tot}}$ in expression (3.3) is the sum of all forces and moments, grouped as:

$$\boldsymbol{F}^{\boldsymbol{tot}} = \boldsymbol{F}_{\boldsymbol{hydro}} + \boldsymbol{F}_{\boldsymbol{g}} + \boldsymbol{F}_{\boldsymbol{aero}} + \boldsymbol{F}_{\boldsymbol{mooring}} \qquad (3.4)$$

Where:

$\boldsymbol{F}_{\boldsymbol{hydro}}$: Hydrostatic and Hydrodynamic forces (and torques);

$\boldsymbol{F}_{\boldsymbol{g}}$: Gravity force and torque;

$\boldsymbol{F}_{\boldsymbol{aero}}$: Aerodynamics forces and torques produced by the airflow on the blades;

$\boldsymbol{F}_{\boldsymbol{mooring}}$: Forces and torques due to mooring lines.

3.1.1.1 F_{hydro}

Hydrostatic and hydrodinamic forces/torques are:

$$\boldsymbol{F_{hydro}} = \boldsymbol{F_{Rest}} + \boldsymbol{F_{FK}} + \boldsymbol{F_D} + \boldsymbol{F_{Rad}} + \boldsymbol{F_{Drag}} \qquad (3.5)$$

Where:

$\boldsymbol{F_{Rest}}$: Hydrostatic restoring forces (and torques);

$\boldsymbol{F_{FK}}$: Froude-Krylov forces, due to incident waves;

$\boldsymbol{F_D}$: Diffraction forces, caused by interaction between incoming waves and the structure;

$\boldsymbol{F_{Rad}}$: Forces caused by motion of the structure in the fluid;

$\boldsymbol{F_{Drag}}$: Friction forces proportional to velocities squares.

F_{Rest}

Restoring forces/torques can be expressed as:

$$\boldsymbol{F_{Rest}} = \begin{Bmatrix} F_{\text{x, Rest}} \\ F_{\text{z, Rest}} \\ M_{\text{y, Rest}} \end{Bmatrix} = -\boldsymbol{C} \cdot \boldsymbol{X} + \begin{Bmatrix} 0 \\ \gamma V_w \\ \gamma V_w \cdot \mathrm{x}_B^{wt} \cdot \cos(\mathrm{p}) \end{Bmatrix} \qquad (3.6)$$

- V_w: displaced fluid volume;
- x_B^{wt}: position of buoyancy center in wt-coordinates;
- $\boldsymbol{X} = \begin{Bmatrix} s \\ h \\ p \end{Bmatrix}$: surge, heave, pitch;
- $\cos(\mathrm{p}) \approx 1$, due to small amplitude motion hypothesis;
- $\boldsymbol{C} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \boldsymbol{C}_{22} & 0 \\ 0 & 0 & \boldsymbol{C}_{33} \end{bmatrix}$: stiffness matrix, defined as:
  - $\boldsymbol{C}_{22} = \frac{-\Delta F_z}{h} = \int_{S_b} \rho g h \cdot \mathbf{n} \cdot \hat{k} \cdot dS \cdot \frac{1}{h} = \rho g A$
  - $\boldsymbol{C}_{33} = \frac{-\Delta M_y}{p} = \int_{S_b} \boldsymbol{r} \wedge P\,\mathbf{n} \cdot dS \cdot \hat{\jmath} \cdot \frac{1}{\mathrm{p}} = \rho g V_w z_B + \rho g \int_{s_b} x^2\, n_3\, dS$

F_{FK}, F_D and F_{Rad}

Froude-Krylov, Diffraction and Radiation forces (and moments) are consequences, respectively, of fluid velocity fields associated with incident waves, diffracted waves (caused by interaction of incident waves with the structure) and radiated waves (due to structure motion). They can be calculated resolving a boundary value problem (BVP) based on potential flow theory, for which following hypotheses hold:

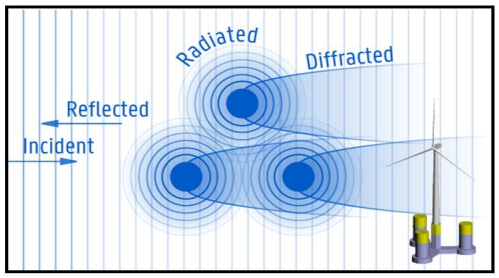

Figure 3.3: Different types of waves.

$$\rho = cost \quad \Rightarrow \quad \nabla \cdot \boldsymbol{u} = 0 \qquad (3.7)$$

$$\mu = 0 \quad \Rightarrow \quad \nabla \times \boldsymbol{u} = 0 \qquad (3.8)$$

With: $-$ $\boldsymbol{u}(\boldsymbol{x}, t)$: speed field (m/s);

$-$ μ: viscosity $(N\ s/\ m^2)$;

Thanks to (3.8) velocity field can be expressed as gradient of a function, in this case called ϕ, and due to (3.7):

$$\boldsymbol{u} = \nabla\phi$$

$$\nabla \cdot (\nabla\phi) = 0 \quad \Rightarrow \quad \nabla^2\phi = 0 \quad (3.9)$$

Moreover, with these condition, Bernoulli's equation for non-stationary systems, expressed in term of the potential ϕ, is valid:

$$\frac{\partial\phi(\boldsymbol{x},t)}{\partial t} + \frac{1}{2}|\nabla\phi(\boldsymbol{x},t)|^2 + \frac{p-p_a}{\rho} + gz = 0, \quad \text{with } z = 0 \ \ at\ s.w.l \qquad (3.10)$$

Then we arrive at the following system:

$$\begin{aligned}
&\nabla^2\phi(\boldsymbol{x},t) = 0, && \boldsymbol{x} \in V && \\
&\frac{\partial\phi}{\partial\boldsymbol{n}} = \mathbf{v}\cdot\mathbf{n}, && \boldsymbol{x} \in S_b, && \mathbf{v}: body\ local\ velocity, \quad S_b: body\ surface; \\
&\frac{\partial\phi}{\partial\boldsymbol{n}} = 0, && \boldsymbol{x} \in S_z, && S_s: sea\ bed\ surface; \\
&\boldsymbol{u}(\boldsymbol{x},t) \to 0, && |\boldsymbol{x}| \to \infty && \qquad (3.11) \\
&\frac{\partial\eta}{\partial t} + \nabla\eta\cdot\nabla\phi = \frac{\partial\phi}{\partial z}, && \boldsymbol{x} \in S_f, && S_f: free\ surface; \\
&\frac{\partial\phi}{\partial t} + \frac{1}{2}|\nabla\phi(\boldsymbol{x},t)|^2 + gz = 0, && \boldsymbol{x} \in S_f, && S_f: free\ surface;
\end{aligned}$$

Assuming H << λ (where H and λ are waves height and wavelength) and a small-amplitude motion of the structure, non-linear terms depending on velocities can be neglected and free surface conditions can be simplified. The resulting linear BVP is:

$$
\begin{aligned}
&\nabla^2\phi(\boldsymbol{x},t) = 0, && \boldsymbol{x} \in V \\
&\frac{\partial\phi}{\partial\boldsymbol{n}} = \mathbf{v}\cdot\mathbf{n}, && \boldsymbol{x} \in S_b, \\
&\boldsymbol{u}(\boldsymbol{x},t) \to 0, && |\boldsymbol{x}| \to \infty \qquad (3.12) \\
&\frac{\partial^2\phi}{\partial t^2} + g\frac{\partial\phi}{\partial z} = 0, && \boldsymbol{x} \in S_f, \\
&\frac{\partial\phi}{\partial\boldsymbol{n}} = 0, && \boldsymbol{x} \in S_z,
\end{aligned}
$$

$-\ \mathbf{v}$: $body\ local\ speed$ $\quad -\ S_b$: $body\ surface$

$-\ S_f$: $free\ surface$ $\quad -\ S_z$: $sea\ bed\ surface$

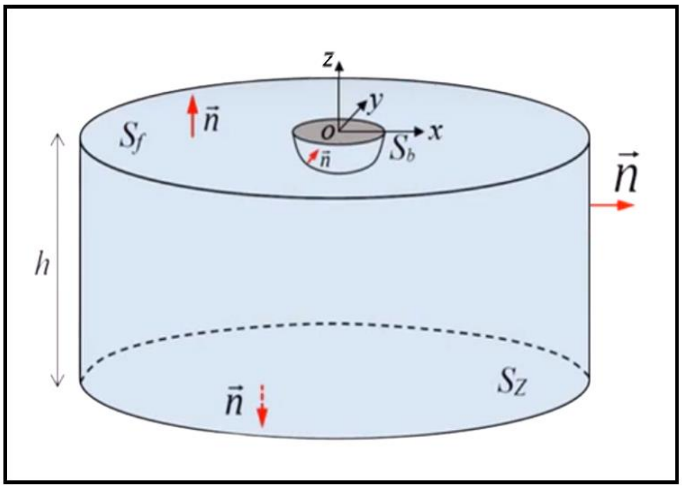

Figure 3.4: Boundary of the boundary value problem.

Solution of linear BVP-Frequency Domain

In frequency domain analysis we assume motion in the form:

$$\boldsymbol{X}(t) = X_0 \cdot \cos(\omega t + \boldsymbol{\varphi}) = \mathrm{Re}\left\{\boldsymbol{X}\cdot e^{j\omega t}\right\},\ \boldsymbol{X} = X_0 e^{j\varphi} \qquad (3.13)$$

Considering linearity of the BVP, velocity potential (solution of the BVP) can be assumed in the form:

$$\phi(\boldsymbol{x},t) = \varphi(\boldsymbol{x})\cdot e^{j\omega t} \qquad (3.14)$$

$$\phi(\boldsymbol{x},t) = \phi_I(\boldsymbol{x},t) + \phi_D(\boldsymbol{x}),t + \phi_{Rad}(\boldsymbol{x},t) \qquad (3.15)$$

Where $\varphi_I(\boldsymbol{x}) + \varphi_D(\boldsymbol{x}) + \varphi_{Rad}(\boldsymbol{x})$ are, respectively, incident (Froude-Krylov), diffraction and radiation potentials. They will be analyzed separately to calculate related forces and moments. Moreover, expression of surface elevation $\eta(\boldsymbol{x},t)$ is chosen based on the hypothesis of regular unidirectional planar waves:

$$\eta(\boldsymbol{x},t) = \eta_0\cos(\omega t - kx + \varphi) = \mathrm{Re}\left\{\boldsymbol{\eta}\cdot e^{j(\omega t - kx)}\right\} \qquad (3.16)$$

$$\boldsymbol{\eta} = \eta_0 e^{j\varphi} \qquad (3.17)$$

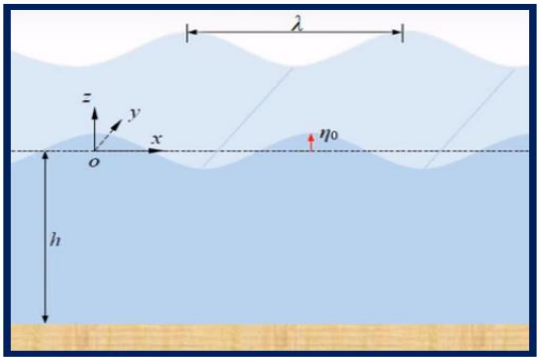

Figure 3.5: Rapresentation of regular waves.

Quantities in relations 3.16 and 3.17 are defined as:

$$- \quad T = \frac{2\pi}{\omega} : \ wave\ period; \qquad - \quad \eta_0 : wave\ amplitude;$$
$$- \quad \lambda = \frac{2\pi}{k} : \ wavelength; \qquad - \quad c = \frac{\lambda}{T} = \frac{\omega}{k} : phase\ velocity.$$

$\mathbf{F_{ext}}$

Froude-Krylov and diffraction forces are present even in the absence of movement of the structure. For the purpose of calculating the forces, due to the hypothesis of small displacements, the structure is considered in the position of rest. They will be considered together defining $\boldsymbol{\phi}_{ext}(\boldsymbol{x}, \mathrm{j}\omega) = \boldsymbol{\phi}_I(\boldsymbol{x}, \mathrm{j}\omega) + \boldsymbol{\phi}_D(\boldsymbol{x}, \mathrm{j}\omega)$, where $\boldsymbol{\phi}_{ext}(\boldsymbol{x}, \mathrm{j}\omega)$ is Fourier transform of $\phi_{ext}(\boldsymbol{x}, t)$, the excitation potential, which is related with $\mathbf{F}_{\text{ext}}$, excitation forces.

From Airy theory, excitation potential is:

$$\boldsymbol{\phi}_{\text{ext}}(\boldsymbol{x}, \mathrm{j}\omega) = j \cdot \frac{g \cdot \cosh(k(z+h))}{\omega \cdot \cosh(kh)} \cdot \boldsymbol{\eta} \cdot e^{-jkx} \qquad (3.18)$$

Where the dispersion relation hold: $k \tanh(kh) = \frac{\omega^2}{g}$.

Considering deep water case, (3.18) and dispersion relation become:

$$\boldsymbol{\phi}_{\text{ext}}(\boldsymbol{x}, \mathrm{j}\omega) = j \cdot \frac{g}{\omega} \cdot e^{kz} \cdot \boldsymbol{\eta} \cdot e^{-jkx} \qquad \text{with: } \omega = \sqrt{gk} \qquad (3.19)$$

Remembering Bernoulli's equation (3.10) we can calculate excitation forces and moments as consequences of dynamic pressure related to excitation potential:

$$\boldsymbol{F}_{ext_i} = -\int_{S_b} p_{ext} \cdot \tilde{n}_i \, dS = \int_{S_b} \rho \frac{\partial \phi_{\text{ext}}(\boldsymbol{x}, \mathrm{t})}{\partial t} \tilde{n}_i \, dS \qquad (3.20)$$

- $\boldsymbol{F}_{ext} = \begin{Bmatrix} F_{\text{ext}_x} \\ F_{\text{ext}_z} \\ M_{\text{ext}_y} \end{Bmatrix}$, $\tilde{n} = \begin{Bmatrix} \mathbf{n} \cdot \hat{\imath} \\ \mathbf{n} \cdot \hat{k} \\ (\boldsymbol{r} \wedge \boldsymbol{n}) \cdot \hat{\jmath} \end{Bmatrix}$;
- $\boldsymbol{r}$: distance between generic wet body surface point and center of wt reference frame.

Then $\boldsymbol{F}_{\text{ext}}(\mathrm{j}\omega)$, Fourier transform of $\boldsymbol{F}_{\text{ext}}(\mathrm{t})$, is:

$$\boldsymbol{F}_{ext}(\mathrm{j}\omega) = \boldsymbol{\eta} \cdot \widetilde{\boldsymbol{F}}_{ext}(j\omega) \qquad (3.21)$$

Where: $\widetilde{\boldsymbol{F}}_{ext_i}(j\omega) = -\int_{S_b} \rho \cdot g \cdot e^{\frac{\omega^2}{g}(z - jx)} \cdot \tilde{n}_i \, dS$

F_{Rad}

F_{Rad} depends on radiation potential, it is defined as a sum of n terms, each term is related to motion of one degree of freedom and proportional to its amplitude. So, Fourier transform of potential $\boldsymbol{\phi}_{\mathrm{Rad}}\ (\boldsymbol{x}, \mathrm{t})$ is:

$$\boldsymbol{\phi}_{\mathrm{Rad}}\ (\boldsymbol{x}, \mathrm{j}\omega)\ = \sum \boldsymbol{\phi}_{\mathrm{Rad,i}}\ (\boldsymbol{x}, \mathrm{j}\omega) \qquad (3.22)$$

$$\boldsymbol{\phi}_{\mathrm{Rad,i}}\ (\boldsymbol{x}, \mathrm{j}\omega) = \ \boldsymbol{\phi}'_{\mathrm{Rad,i}}\ (\boldsymbol{x}, \mathrm{j}\omega) \cdot \boldsymbol{X}_i(j\omega) \qquad (3.23)$$

Remembering Bernoulli's equation (3.10) we can calculate radiation forces and torques as consequences of dynamic pressure related to radiation potentials. Focusing into one of the three component of generalized forces (forces along surge and heave direction and torque around y axis) we have:

$$\boldsymbol{F}_{Rad_i}(t) = -\int_{S_b} p_{Rad}\ \cdot \tilde{n}_i\ dS = \int_{S_b} \rho \frac{\partial \phi_{\mathrm{Rad}}\ (\boldsymbol{x}, t)}{\partial t} \tilde{n}_i\ dS \qquad (3.24)$$

$$\text{With:}\quad \tilde{n} = \begin{Bmatrix} \mathbf{n} \cdot \hat{\imath} \\ \mathbf{n} \cdot \hat{k} \\ (\boldsymbol{r} \wedge \boldsymbol{n}) \cdot \hat{\jmath} \end{Bmatrix}$$

And given (3.13) and (3.16):

$$\boldsymbol{F}_{Rad_i}(j\omega)\ =\ j\omega \cdot \sum_j\ \boldsymbol{X}_j \cdot \int_{S_b} \rho \cdot \phi'_{\mathrm{Rad,j}} \frac{\partial \phi'_{\mathrm{Rad,i}}}{\partial n} \cdot dS =\ j\omega \cdot \sum_j\ \widetilde{\boldsymbol{F}}_{Rad_{i,j}}(j\omega)\ \cdot\ \boldsymbol{X}_j \qquad (3.25)$$

$$\text{With:}\quad \widetilde{\boldsymbol{F}}_{Rad_{i,j}}(j\omega) = \int_{S_b} \rho \cdot \phi'_{\mathrm{Rad,j}} \frac{\partial \phi'_{\mathrm{Rad,i}}}{\partial n} \cdot dS$$

$$\boldsymbol{F}_{Rad_i}(j\omega)\ =\ -\ \sum_j \frac{1}{\omega} \cdot \mathrm{Im}\left[\widetilde{\boldsymbol{F}}_{Rad_{i,j}}\right] \cdot \omega^2\ \boldsymbol{X}_j\ +\ \sum_j\ \mathrm{Re}\left[\widetilde{\boldsymbol{F}}_{Rad_{i,j}}\right] \cdot j\ \omega\ \boldsymbol{X}_j \qquad (3.26)$$

Finally, generalized forces vector is:

$$\boldsymbol{F}_{Rad}(j\omega)\ = \omega^2\ \mathbf{A}(\omega) \cdot \boldsymbol{X}\ -\ j\ \omega \cdot \mathbf{B}(\omega) \cdot\ \boldsymbol{X}_j \qquad (3.27)$$

Once defined added mass matrix **A** and radiation damping matrix **B**:

$$\mathbf{A}_{i,j}(\omega) = -\frac{1}{\omega} \cdot \mathrm{Im}\ \left[\widetilde{\boldsymbol{F}}_{Rad_{i,j}}(j\omega)\right] \qquad (3.28)$$

$$\mathbf{B}_{i,j}(\omega) = -\ \mathrm{Re}\ \left[\widetilde{\boldsymbol{F}}_{Rad_{i,j}}(j\omega)\right] \qquad (3.29)$$

In this figure there is an example of **A** and **B** matrix components with respect to wave frequency.

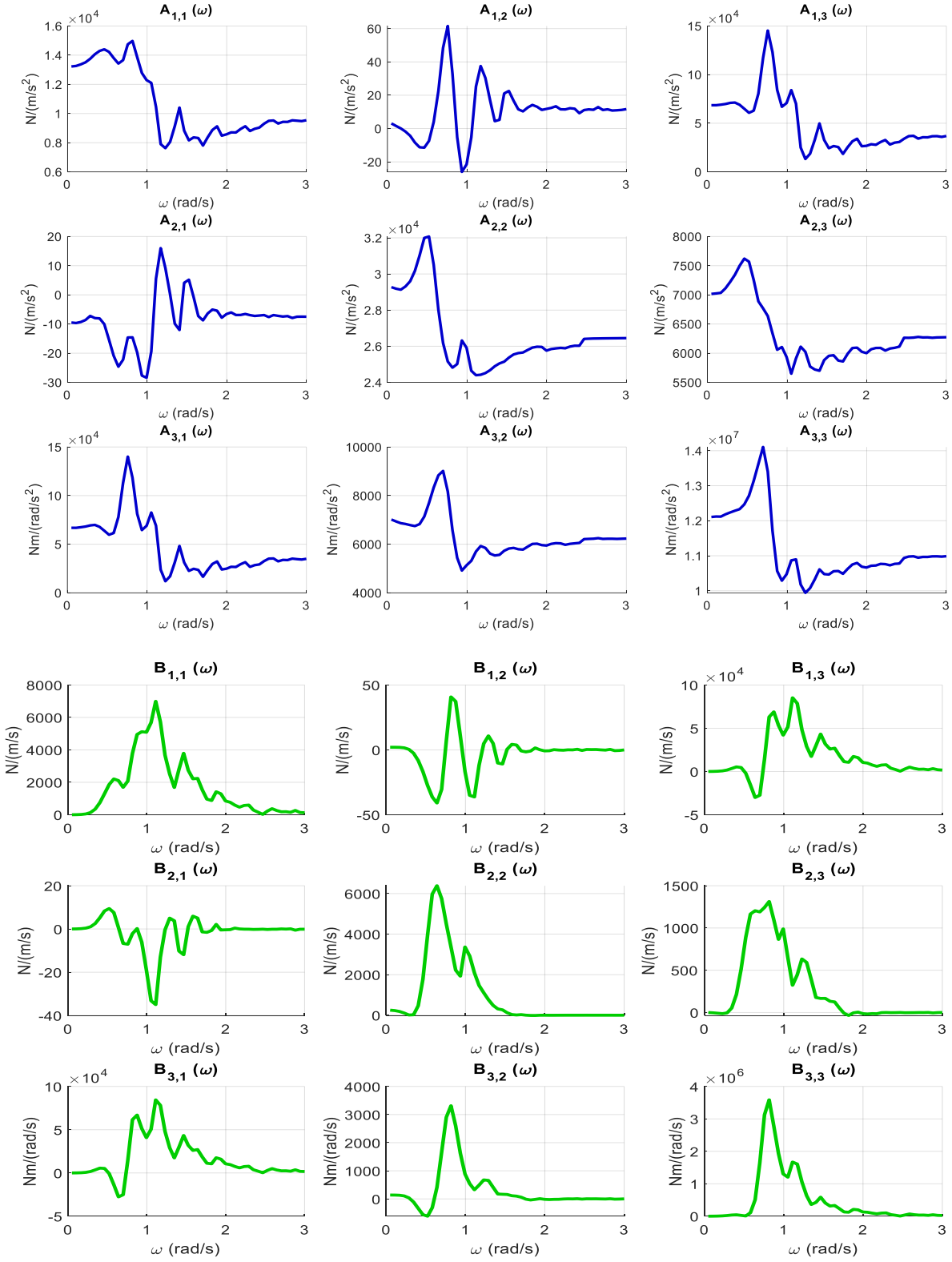

Figure 3.6: Added mass A and radiation damping B matrix coefficients with respect to frequency (rad/s)

Solution of linear BVP-Time Domain

As for the frequency analysis, thanks to linearity of BVP, we can threat potential as a sum of incident, diffracted and radiated ones and calculate the response of the system as sum of response due to different effects (superposition effect).

$$\phi(\boldsymbol{x},t) = \phi_I(\boldsymbol{x},t) + \phi_D(\boldsymbol{x}) + \phi_{Rad}(\boldsymbol{x},t) \qquad (3.30)$$

Once each potential is computed, correlated forces can be obtained through Bernoulli's equation (3.10).

$\mathbf{F_{ext}}$

We treat, as in the frequency analysis, the incident and diffracted potential together. Their sum, the exitation potential, is defined as follows:

$$\phi_{ext}(\boldsymbol{x},t) = \phi_I(\boldsymbol{x},t) + \phi_D(\boldsymbol{x},t) \qquad (3.31)$$

In time domain analyses we assume possibility of any form of surface elevation $\boldsymbol{\eta}$(t), instead of regular waves used in frequency domain. To do so, $\boldsymbol{\eta}$(t) can be expressed as the sum of its harmonics, everyone in the form: $\eta(t) = \eta_0 \cos(\omega t - kx + \varphi)$.

$$\eta(t) = \sum_i \eta_{0_i} \cdot \cos(\omega_i t - k_i x + \varphi_i) = Re \sum_1^N \eta_{0_i} e^{j(\omega_i t - k_i x + \varphi_i)} \qquad (3.32)$$

So, remembering (3.21), obtained in previous analysis, and assuming N components:

$$\boldsymbol{F}_{ext}(t) = Re\left[\sum_1^N \eta_{0,i} \cdot \widetilde{\boldsymbol{F}}_{\text{ext}}(\omega_i) \cdot e^{j(\omega_i t + \varphi_i)}\right] \qquad (3.33)$$

$$\boldsymbol{F}_{ext}(\text{j}\omega) = \boldsymbol{\eta} \cdot \widetilde{\boldsymbol{F}}_{ext}(j\omega) \qquad (3.34)$$

With: $\widetilde{\boldsymbol{F}}_{ext_i}(j\omega_i) = -\int_{S_b} \rho \cdot g \cdot e^{\frac{\omega_i^2}{g}(z - jx)} \cdot \tilde{n}_i \, dS$

F_{Rad}

F_{Rad} depend on radiation potential, it is defined as a sum of n terms, each term is related to motion of one degree of freedom and proportional to its amplitude. Cummins [6] proposed a solution of BVP in which radiation potential is expressed as:

$$\phi_{\text{Rad}}(\boldsymbol{x},t) = \boldsymbol{\Psi}(\boldsymbol{x}) \cdot \dot{\boldsymbol{X}}(t) + \int_0^t \boldsymbol{\chi}(\boldsymbol{x}, t-\tau) \cdot \dot{\boldsymbol{X}}(\tau)\, d\tau \qquad (3.35)$$

$$\text{With:} \quad \dot{\boldsymbol{X}}(t) = \{\dot{s}(t), \dot{h}(t), \dot{pitch}(t)\}^T, \quad \boldsymbol{\Psi}, \boldsymbol{\chi} \in \mathbb{R}^{1\times 3}$$

Remembering Bernoulli's equation (3.10) we can calculate radiation forces and moments as consequences of dynamic pressure related to radiation potentials. Focusing into one of the three component of generalized forces (forces along surge and heave direction and torque around y axis) we have:

$$\boldsymbol{F}_{Rad_i}(t) = \int_{S_b} \rho \frac{\partial \phi_{\text{Rad}}(\boldsymbol{x},t)}{\partial t} \tilde{n}_i \, dS \qquad (3.36)$$

$$\boldsymbol{F}_{Rad_i}(t) = -\left[\int_{\dot{S}b} - \rho \cdot \boldsymbol{\Psi}(\boldsymbol{x}) \cdot \tilde{n}_i dS\right] \cdot \ddot{\boldsymbol{X}}(t) - \int_0^t \left[\int_{\text{Sb}} - \rho \frac{\partial \boldsymbol{\chi}(\boldsymbol{x}, t-\tau)}{\partial t} \cdot \tilde{n}_i \cdot dS\right] \cdot \dot{\boldsymbol{X}}(\tau) d\tau \quad (3.37)$$

$$\text{With:} \ \tilde{n} = \begin{Bmatrix} \mathbf{n} \cdot \hat{\imath} \\ \mathbf{n} \cdot \hat{k} \\ (\boldsymbol{r} \wedge \boldsymbol{n}) \cdot \hat{\jmath} \end{Bmatrix}$$

Finally, generalized forces vector is:

$$\boldsymbol{F}_{Rad}(t) = -\boldsymbol{A}_\infty \cdot \ddot{\boldsymbol{X}}(t) - \int_0^t \boldsymbol{K}_R(t-\tau) \cdot \dot{\boldsymbol{X}}(\tau)\, d\tau \qquad (3.38)$$

Once defined:

$$\mathbf{A}_{\infty,ij} = -\rho \int_{\dot{S}b} \Psi_{\boldsymbol{j}}(\boldsymbol{x}) \cdot \tilde{n}_i \, dS \qquad (3.39)$$

$$\mathbf{K}_{R,ij}(t) = -\rho \int_{\dot{S}b} \frac{\partial \chi_j(\boldsymbol{x},t)}{\partial t} \cdot \tilde{n}_i \, dS \qquad (3.40)$$

On the next page there is the representation of $\mathbf{K}_{R,ij}(t)$, (Figure 3.7), in which we notice that some extra-diagonal terms are much smaller than the relative diagonal terms. For this reason, these components will be neglected. Furthermore, since the convolution integral in (3.38) requires a lot of computation time, it was decided to approximate the latter with the output of a state space system whose inputs are structure's speeds $\dot{\boldsymbol{X}}(t)$. Due to approximations mentioned above the convolution integral becomes:

$$\int_0^t \boldsymbol{K}_R(t-\tau) \cdot \dot{\boldsymbol{X}}(\tau) d\tau = \int_0^t \begin{bmatrix} K_{R_{11}} & 0 & K_{R_{13}} \\ 0 & K_{R_{22}} & 0 \\ K_{R_{31}} & 0 & K_{R_{33}} \end{bmatrix} \cdot \dot{\boldsymbol{X}}(\tau) d\tau = \begin{Bmatrix} \sum_{j=1,3} \int_0^t K_{R_{1j}}(t-\tau) \cdot \dot{X}_j \, d\tau \\ \int_0^t K_{R_{22}}(t-\tau) \cdot \dot{X}_2 d\tau \\ \sum_{j=1,3} \int_0^t K_{R_{3j}}(t-\tau) \cdot \dot{X}_j \, d\tau \end{Bmatrix} \quad (3.41)$$

Then, each scalar convolution integral is approximated with a linear SISO state space whose input is the related velocity component and output is the contribution of the related component of radiation forces vector.

$$\int_0^t K_{R_{ij}}(t-\tau)\cdot \dot{X}_j\, d\tau \approx y_{R_{ij}} \leftrightarrow \begin{cases} \dot{\boldsymbol{q}}_{R_{ij}} = \boldsymbol{A}_{R_{ij}}\cdot \boldsymbol{q}_{R_{ij}} + \boldsymbol{B}_{R_{ij}}\cdot \dot{X}_j \\ y_{R_{ij}} = \boldsymbol{C}_{R_{ij}}\cdot \boldsymbol{q}_{R_{ij}} \end{cases} \qquad (3.42)$$

$\boldsymbol{A}_{R_{ij}}, \boldsymbol{B}_{R_{ij}}$ *and* $\boldsymbol{C}_{R_{ij}}$ can be calculated as that who lead to have a system whose impulse response is $K_{R_{ij}}(t-\tau)$. Putting SISO state spaces together we obtain a MIMO state space with velocity vector $\dot{\boldsymbol{X}}$ as input and convolution integral as output:

$$\int_0^t \boldsymbol{K}_R(t-\tau)\cdot \dot{\mathbf{X}}(\tau)\, d\tau \approx \boldsymbol{y}_R \leftrightarrow \begin{cases} \dot{\boldsymbol{q}}_R = \boldsymbol{A}_R\cdot \boldsymbol{q}_R + \boldsymbol{B}_R\cdot \dot{\mathbf{X}} \\ \boldsymbol{y}_R = \boldsymbol{C}_R\cdot \boldsymbol{q}_R \end{cases} \qquad (3.43)$$

More information about approximation of added damping integral by a linear state space can be find here [7].

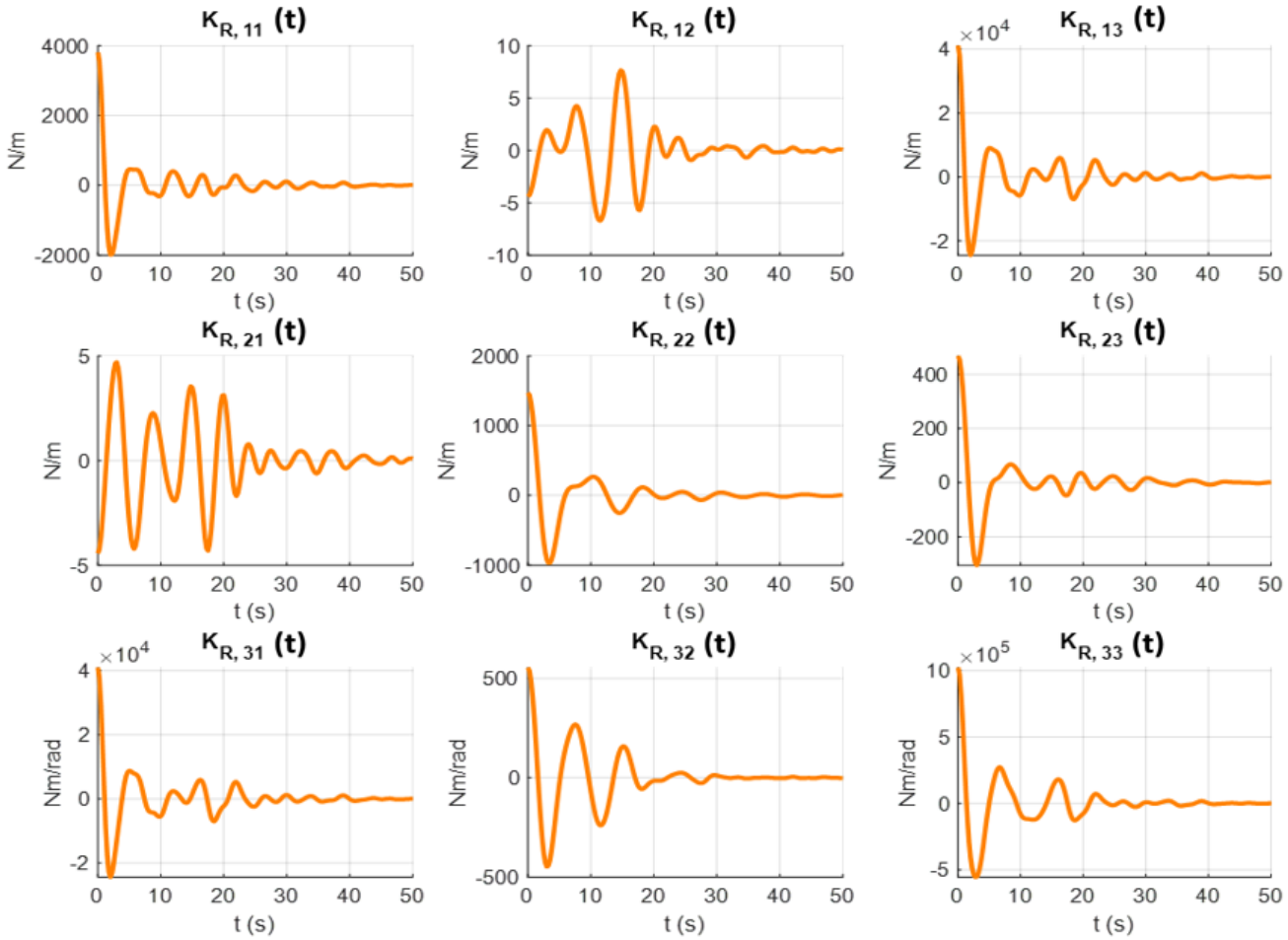

Figure 3.7: Components of radiaton damping impulse rensponse function $\boldsymbol{K}_R(t)$.

Relation Between Frequency and Time domain

In this work we will use the time approach, so data needed for simulation are $\mathbf{K}_R$ and $\mathbf{A}_\infty$. They are obtained once **A**(ω) and **B**(ω) are known by solving the BVP problem in the frequency domain with Nemoh [8], an opensource software which solve it with panel method. $\mathbf{K}_R$ and $\mathbf{A}_\infty$ are then calculated utilizing **A**(ω) and **B**(ω). Alternatively, if state space representation of radiation convolution integral is used, we can directly estimate them from **A**(ω) and **B**(ω) matrices.

To calculate $\mathbf{K}_R$ and $\mathbf{A}_\infty$ from **A**(ω) and **B**(ω), we can express equation (3.38) in frequency domain form taking Fourier transform of $\boldsymbol{F}_{Rad}(t)$, once assumed regular waves and motion in the form $\boldsymbol{X}(t) = \boldsymbol{X}_0 \cdot \cos(\omega t + \boldsymbol{\varphi})$.

$$\mathcal{F}[\boldsymbol{F}_{Rad}(t)] \equiv \boldsymbol{F}_{Rad}(j\omega) = \mathcal{F}\left[-\boldsymbol{A}_\infty \cdot \ddot{\boldsymbol{X}}(t) - \int_0^t \boldsymbol{K}_R(t-\tau) \cdot \dot{\boldsymbol{X}}(\tau)\, d\tau\right] \tag{3.44}$$

$$\boldsymbol{F}_{Rad}(j\omega) = (\omega^2 \boldsymbol{A}_\infty - \mathcal{F}[\boldsymbol{K}_R(t)] \cdot j\omega) \cdot \boldsymbol{X}(j\omega)) \tag{3.45}$$

$$\boldsymbol{F}_{Rad}(j\omega) = \omega^2 (\boldsymbol{A}_\infty + \frac{\mathbf{1}}{\omega} Im[\boldsymbol{K}_R(j\omega)]) \cdot \boldsymbol{X}(j\omega) - j\omega \cdot Re[\boldsymbol{K}_R(j\omega)] \cdot \boldsymbol{X}(j\omega) \tag{3.46}$$

With: $\mathcal{F}[\boldsymbol{K}_R(t)] = \boldsymbol{K}_R(j\omega) = Re[\boldsymbol{K}_R(j\omega)] + j\, Im[\boldsymbol{K}_R(j\omega)]$

$$Re[\boldsymbol{K}_R(j\omega)] = \int_0^\infty \boldsymbol{K}_R(t)\cos(\omega t)\, dt$$

$$Im[\boldsymbol{K}_R(j\omega)] = -\int_0^\infty \boldsymbol{K}_R(t)\sin(\omega t)\, dt$$

Comparing (3.46) with (3.27): $\boldsymbol{F}_{Rad}(j\omega) = \omega^2\, \mathbf{A}(\omega) \cdot \boldsymbol{X} - j\omega \cdot \mathbf{B}(\omega) \cdot \boldsymbol{X}_j$, relations 3.47-3.49, proposed by Ogilvie [9], can be written:

$$\boldsymbol{A}(\omega) = \boldsymbol{A}_\infty - \frac{1}{\omega}\int_0^\infty \boldsymbol{K}_R(t)\sin(\omega t)\, dt \tag{3.47}$$

$$\boldsymbol{B}(\omega) = \int_0^\infty \boldsymbol{K}_R(t)\cos(\omega t)\, dt \tag{3.48}$$

$$\boldsymbol{K}_R(j\omega) = \boldsymbol{B}(\omega) - j\omega(\boldsymbol{A}_\infty - \boldsymbol{A}(\omega)) \tag{3.49}$$

F_{Drag}

Drag load considers the effects of damping due to the viscosity of the fluid dissipated energy in vortices. Since linear models don't consider these phenomena because of their basic assumptions, extra damping it is necessary to accurately model the damping in a real system. These non-linearities can be supplemented by a quadratic drag force model. In this case drag forces vector, whose components are surge and heave forces and pitch moment, is defined as:

$$\boldsymbol{F}_{Drag}(t) = \boldsymbol{D} \cdot \begin{Bmatrix} \dot{s} \cdot |\dot{s}| \\ \dot{h} \cdot |\dot{h}| \\ \dot{p} \cdot |\dot{p}| \end{Bmatrix} \qquad (3.50)$$

With: $\mathbf{D} = \begin{bmatrix} d_{11} & 0 & d_{13} \\ 0 & d_{22} & 0 \\ d_{31} & 0 & d_{33} \end{bmatrix}$, $D_{ij} \propto \rho A_i$

3.1.1.2 F_g

Gravity forces vector is defined below, m is the mass of entire structure, x_{cg}^{wt} and z_{cg}^{wt} are non-zero component of gravity center (cg) position with respect to wt center, expressed in wt-reference frame. The third term considers the fact that cg and wt, the point around which the moments are calculated, do not coincide.

$$\boldsymbol{F}_g(t) = \begin{Bmatrix} 0 \\ -mg \\ mg \cdot \left(x_{cg}^{\mathrm{wt}} \cos(p) + z_{cg}^{wt} \sin(p)\right) \end{Bmatrix} \qquad (3.51)$$

3.1.1.3 F_{aero}

The aerodynamic loads acting on the rotor blades are calculated through BEM (Blade Element Momentum) theory, a union between two different methods: the first, provides a global approach, in which thrust and torque are calculated starting from momentum and angular momentum balances. In this case, various simplifications apply, such as an infinite number of blades and frictionless disk is considered. Moreover, the flow is stationary, incompressible and frictionless. The second method (Blade Element) starts from the calculation of the local forces acting on a blade element thanks to the knowledge of profile shape of the latter.

Momentum – Angular momentum

A simple turbine model, attributed to Betz [10], can be used to approximately compute extracted power of an ideal turbine and thrust force of the wind on the rotor. This model is based on momentum equation ($\frac{d\boldsymbol{P}}{dt} = \sum \boldsymbol{F}_{ext}$) computed on an ideal rotor disk rotating in a control volume, with two cross-sections of the streamtube and its surface as boundaries (see Figure 3.8). Following hypothesis are made:

- Absence of flow through lateral surface;
- Rotor with infinite blades (disk), which represent a pressure discontinuity;
- Uniform thrust force over the rotor;
- Static pressure far upstream and far downstream of the rotor is equal to the undisturbed ambient static pressure;
- Non-rotating wake;
- Air density ρ=cost.

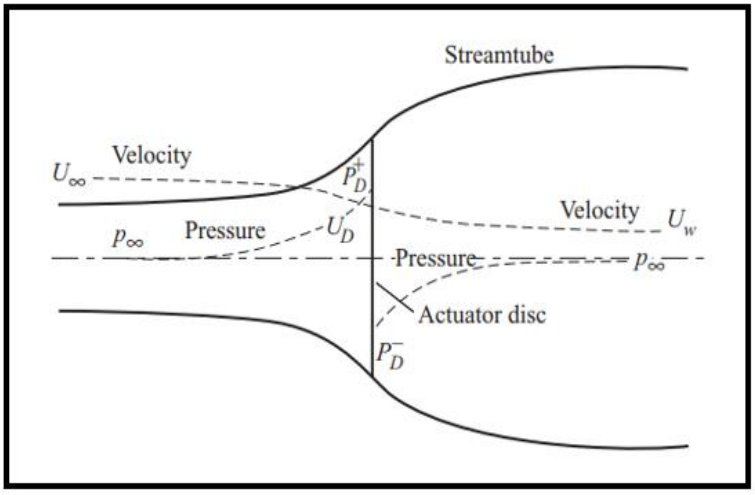

Figure 3.8: Betz ideal disk rotor model

Since we assume constant air density and since mass flow rate must be the same everywhere, following equations can be write:

$$\dot{m} = \rho A_\infty U_\infty = \rho A_D U_D = \rho A_W U_W \qquad (3.52)$$

Where ∞ refers to conditions far upstream, D refers to conditions at the disc and W refers to conditions in the far wake. Disk induces a velocity variation which is superimposed on the free stream velocity. The stream-wise component of this induced flow at the disc is given by $-aU_\infty$, where a is called the axial flow induction factor, or the inflow factor, defined as:

$$a = \frac{U_\infty - U_D}{U_\infty} \qquad (3.53)$$

At the disc, therefore, the net velocity is:

$$U_D = U_\infty(1-a) \qquad (3.54)$$

Rate of change of momentum, since an air mass flow changes its speed from inlet to outlet of stream-tube, is:

$$\frac{dP}{dt} = (U_\infty - U_W)\,\rho\, A_D\, U_D \qquad (3.55)$$

Rate of change of momentum equals force acting on air (which come entirely from pressure drop through the disk), which, for dynamic third principle, is equal and opposite to thrust force T acting on rotor disk.

$$T = (p_D^+ - p_D^-)\, A_D = (U_\infty - U_W)\,\rho\, A_D\, U_D \qquad (3.56)$$

$(p_D^+ - p_D^-)$ is obtained from stationary Bernoulli's equation ($\frac{1}{2}\rho U^2 + p + \rho g h =$ const) applied from upstream to disk section and from downstream to disk cross section. These equations, considering constant density and height and considering no work is produced, lead to:

$$\frac{1}{2}\rho U_\infty^2 + p_\infty = \frac{1}{2}\rho U_D^2 + p_D^+ \qquad (3.57)$$

$$\frac{1}{2}\rho U_W^2 + p_\infty = \frac{1}{2}\rho U_D^2 + p_D^- \qquad (3.58)$$

Thus, also remembering equation 3.54:

$$T = \frac{1}{2}\rho(U_\infty^2 - U_W^2)A_D = (U_\infty - U_W)\,\rho\, A_D\, U_\infty(1-a) \qquad (3.59)$$

$$U_W = (1-2a)\,U_\infty \qquad (3.60)$$

Finally, expression of thrust force and thrust force per area unit (with $dA = 2\pi r dr$ and considering uniform thrust, so constant axial induction factor along blade length) are:

$$T = \frac{1}{2}\,\rho\, A_D {U_\infty}^2\, C_T \qquad (3.61)$$

$$dF_N = \rho\, U_\infty^2\, C_T\, \pi r dr \qquad (3.62)$$

Where C_T is the thrust coefficient, defined as thrust over maximum ideal thrust $\frac{1}{2}\,\rho\, A_D {U_\infty}^2$ (axial force that would occur if the downstream wind speed was zero, i.e., all momentum was transferred to the rotor):

$$C_T = \frac{T}{\frac{1}{2}\rho\, A_D {U_\infty}^2} = 4a(1-a) \qquad (3.63)$$

Through thrust expression, ideal extracted power can be defined as:

$$\mathrm{P} = \mathrm{T}\, U_{\mathrm{D}} = \frac{1}{2}\, \rho\, A_D {U_\infty}^3\ C_P \qquad (3.64)$$

C_P is the power output over the available wind power, $\frac{1}{2}\, \rho\, A_D {U_\infty}^3$, called power coefficient:

$$C_P = \frac{P}{\frac{1}{2} \rho\, A_D {U_\infty}^3} = 4a(1-a)^2 \qquad (3.65)$$

Actually, due to interaction with rotor, downstream air speed has an additional component opposite to rotor local velocity (wake effect). To consider the latter we can define the angular induction factor a':

$$a' = \frac{\omega_3}{2\Omega} \qquad (3.66)$$

ω_3 , depending on position along blade length, is induced angular speed at the final cross section of rotor (point 3 in Figure 3.11). It is defined as induced velocity over radius at which a' is computed. Furthermore, for convention, induced angular speed at a section corresponding to ¼ of chord length (point 2 in Figure 3.11) is $\omega_2 = \omega_3/2$.

From angular momentum law ($\frac{d\boldsymbol{L}}{dt} = \sum \boldsymbol{M}_{ext}$), torque per area unit can be expressed, at each radius, in terms of mass flow rate and angular induced speed:

$$dQ = \dot{m}\, \omega_2\, r^2 \quad (3.67)$$

Finally, considering 3.52 and definition of a and a', dQ in term of axial and angular induction factor, undisturbed wind velocity, air density and rotor angular speed becomes [11], [12]:

$$dQ = \rho\, U_\infty^2 \cdot [4a'\,(1-a)] \cdot \lambda_r \pi r^2 dr \quad (3.68)$$

With:

- $$\lambda_r = \frac{\Omega r}{U_\infty} \quad \left(\lambda = \frac{\Omega R}{U_\infty}\right) \qquad (3.69)$$

Where λ_r and λ are local and global tip speed ratios.

Blade element

Blades are divided into small elements of thickness *dr* represented by 2D airfoils which are only subject to local physical events. For each blade, the forces contribution from all elements are summed along the span of the blade to calculate the total loads on the rotor. It is possible, once the geometry of airfoils is chosen, to calculate lift and drag coefficients (C_L and C_D), through which lift and drag forces per unit length can be find. These coefficients depend on angle of attack α, that is the angle between the chord line and the direction of relative wind speed V_{rel} (see Figure 3.9). Lift and drag coefficients are defined as:

$$C_L(\alpha) = \frac{L/l}{\frac{1}{2}\rho\, c\, V_{rel}^2} \qquad (3.70)$$

$$C_D(\alpha) = \frac{D/l}{\frac{1}{2}\rho\, c\, V_{rel}^2} \qquad (3.71)$$

$-\ c: chord;\ \ -\ D: Drag\ force;\ \ -\ L: Lift\ force;$

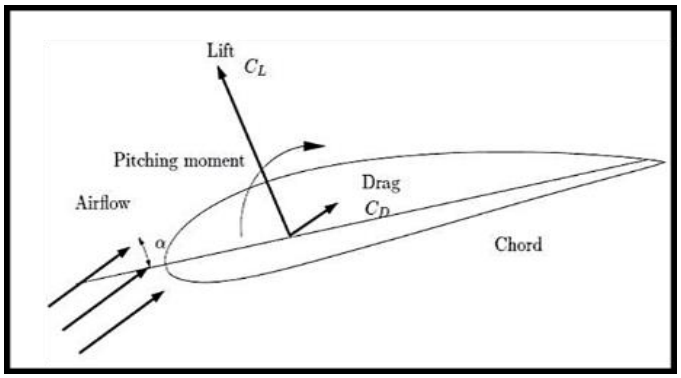

Figure 3.9: Blade profile, Lift force, Drag force and Pitching moment produced due to airflow inclined of α, angle of attack, from chord line.

In the figure below there is an example of C_L and C_D with respect to angle of attack α:

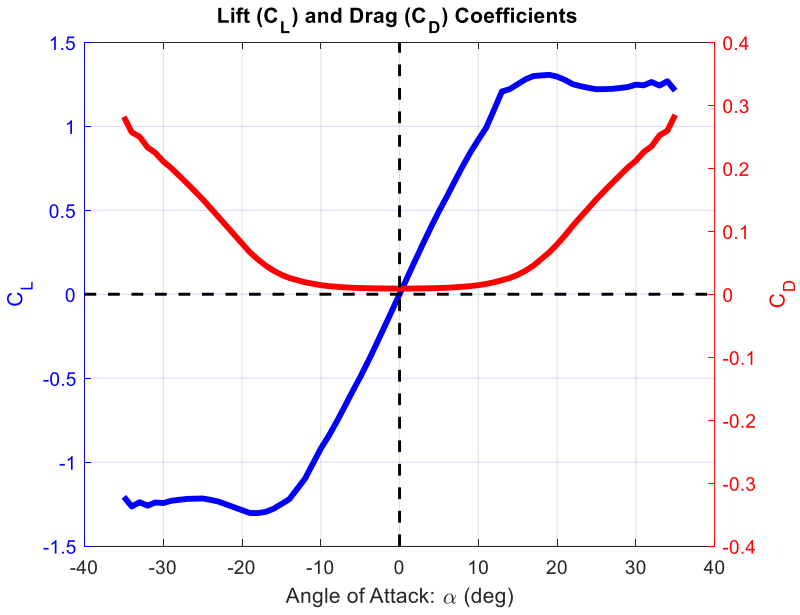

Figure 3.10: Lift and Drag coefficent with respect to angle of attack α

From definitions (3.70-3.71), and looking pictures below, we can find expression of infinitesimal normal and tangential forces for each blade element. V_{rel} is relative velocity between air and point 2 (moving with blade), which is along chord line and far c/4 from point 1, ϕ is the incidence angle, α is attack angle, c is the chord and Ω is the rotor angular velocity.

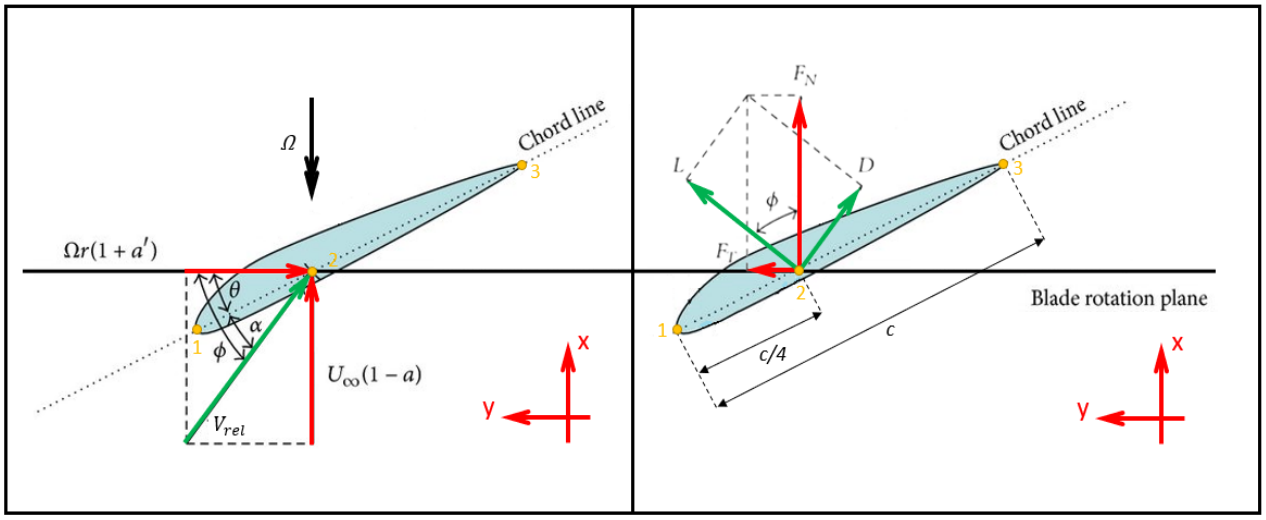

Figure 3.11: Speeds (left) and forces per unit length (right) involved in aerodynamic forces and moment computation.

$$dF_{n_i} = \frac{1}{2}\rho c V_{\text{rel}}^2 \cdot [C_L(\alpha)\cos\phi + C_D(\alpha)\sin\phi]\, dr \qquad (3.72)$$

$$dF_{t_i} = \frac{1}{2}\rho c V_{rel}^2 \cdot [C_L(\alpha)\,\text{sen}\phi - C_{\text{D}}(\alpha)\cos\phi] \cdot dr \qquad (3.73)$$

$$\text{tg}\,\phi = \frac{U_\infty(1-a)}{\Omega \cdot r \cdot (1+a')} = \frac{(1-a)}{\lambda_r(1+a')} \qquad (3.74)$$

Incident angle ϕ is the sum of attack angle and θ, which is sum of blade pitch angle and twist angle:

$$\phi(r) = \theta(r) + \alpha(r) \qquad (3.75)$$

$$\theta(r) = \theta_{\text{bp}} + \theta_{tw}\,(r) \qquad (3.76)$$

So, total infinitesimal forces (related to an annulus of thickness dr) become:

$$dF_{\text{n}} = \frac{1}{2}\rho\sigma U_\infty^2 \cdot \left(\frac{1-a}{\sin\phi}\right)^2 \cdot [C_L(\phi - \theta)\cos\phi + C_D(\phi - \theta)\sin\phi]2\pi r dr \qquad (3.77)$$

$$dF_{\text{t}} = \frac{1}{2}\rho\sigma U_\infty^2 \cdot \left(\frac{1-a}{\sin\phi}\right)^2 \cdot [C_L(\phi - \theta)\sin\phi - C_D(\phi - \theta)\cos\phi]2\pi r dr \qquad (3.78)$$

With: $\sigma = \frac{Bc}{2\pi r}$

B: number of blades

Rearranging equations from momentum theory (3.62 and 3.68) ad from blade element one (3.74, 3.77 and 3.78), we can find the system below, which will be iteratively solved to calculate **a** and **a'**. Once we find them, we can obtain dF_{n_i} and dF_{t_i} and, by integration, aerodynamic forces and moment acting on each blade (at blade root).

$$\begin{cases} \frac{\sigma}{4\text{sen}^2\ \phi}[C_L\cos\ \phi + c_\text{D}\sin\ \phi] = \frac{a}{1-a} \\ \frac{\sigma}{4\text{sen}^2\ \phi}[C_L\sin\ \phi - c_\text{D}\cos\ \phi] = \frac{a'}{1-a} \\ \text{tg}\ \phi = \frac{1-a}{\lambda_r(1-a')} \end{cases} \qquad (3.79)$$

Corrections of BEM model

The BEM model is simplified model in which many approximations are made. In order to consider some neglected effects (those which have more effect), the following factors are defined:

- Prandtl's tip and hub loss factors;
- Glauert's factor;
- Actual wind velocity factor.

Prandtl's tip and hub loss factors

Normally, in the middle of the blade, air flows straight across the wing, parallel to the airfoil, while in the hub and tip zones this isn't true, the airflow is no longer parallel, and vortex are generated. So, to consider the consequent losses and to not overestimate aerodynamic forces, equations 3.62 and 3.68 are modified with $F_{Pr}(r)$ factor, product of tip and hub contributes:

$$dF_n = 4a(1-a)F_{Pr}U_\infty^2\,\rho\pi r dr \qquad (3.80)$$

$$dF_\text{t} = 4a'(1-a)F_{Pr}U_\infty^2\,\rho\pi r\Omega dr \qquad (3.81)$$

With:

$$F_{Pr}(r) = F_{Pr,tip}(r)\cdot F_{Pr,hub}(r) \qquad (3.82)$$

$$F_{Pr,tip}(r) = \frac{2}{\pi}\cos^{-1}\ e^{-\frac{B}{2}\frac{R-r}{r\sin\phi}} \qquad (3.83)$$

$$F_{Pr,hub}(r) = \frac{2}{\pi}\cos^{-1}\ e^{-\frac{B}{2}\frac{r-R_h}{r\sin\phi}} \qquad (3.84)$$

The figure below shows Prandtl's factor along normalized radius. It can be seen that hub factor is influent in a narrow range, whereas tip factor has values far from 1 starting from roughly 80% of the radius.

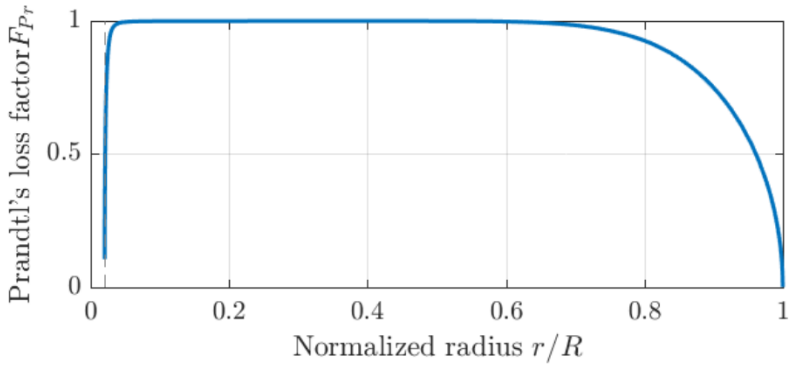

Figure 3.12: Prandtl's loss factor with respect to normalized radius.

Glauert's factor

For induction factors **a** larger than about 0.4, a turbulent wake usually appears, not considered in momentum theory. Glauert firstly, and then many other authors, proposed a modify in the expression of thrust (3.62) when **a** is greater than a given threshold $\mathbf{a_c}$. In this work correction proposed by Spera and Wilson [13] is used and 3.62 becomes (also Prandtl's factor is included in this formula):

$$dF_{\mathrm{n}} = C_T\, F_{Pr}\, U_{\infty}^2 \rho \pi r dr \qquad (3.85)$$

With: $\quad C_T = 4(a(1-a) + (a-a_c)_+^2)$

$$(a-a_c)_+ := max\{0, a-a_c\}, \quad a_c = 1/3$$

In this figure we can see expression of C_T , local thrust coefficient, with respect to induction factor for momentum theory ($C_T = 4a(1-a)$), with correction we made and from CFD analysis.

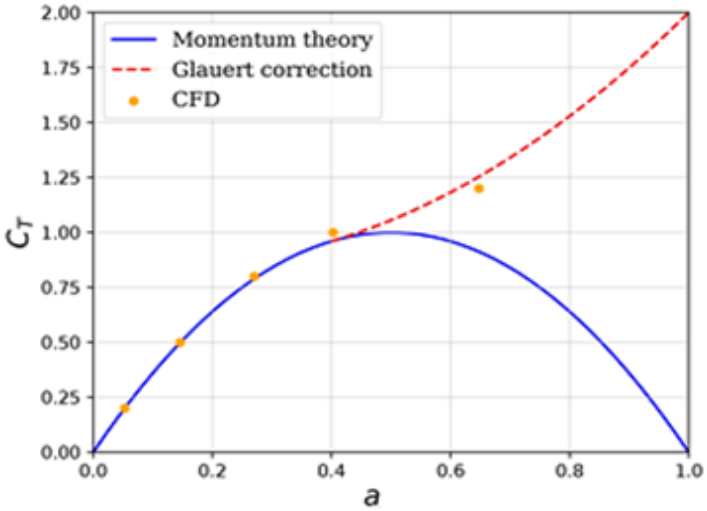

Figure 3.13: C_T , local thrust coefficient, from Momenthum theory, Glauert's correction and CFD analysis.

Actual wind velocity factor

Until now, we have considered the undisturbed wind velocity (U_∞) as the wind speed blowing on the blades in x direction of blade reference frame (bl in Figure 3.14). Actually, some clarifications have to be made.

Firstly, the structure is moving, so relative velocity change; moreover, the rotor rotates around an axis which is inclined with respect to horizontal one by the tilt angle and the blades are inclined about rotation axis by the precone angle (see Figure 3.14). Furthermore, wind velocity isn't constant in space, so different elements are subject to different flow.

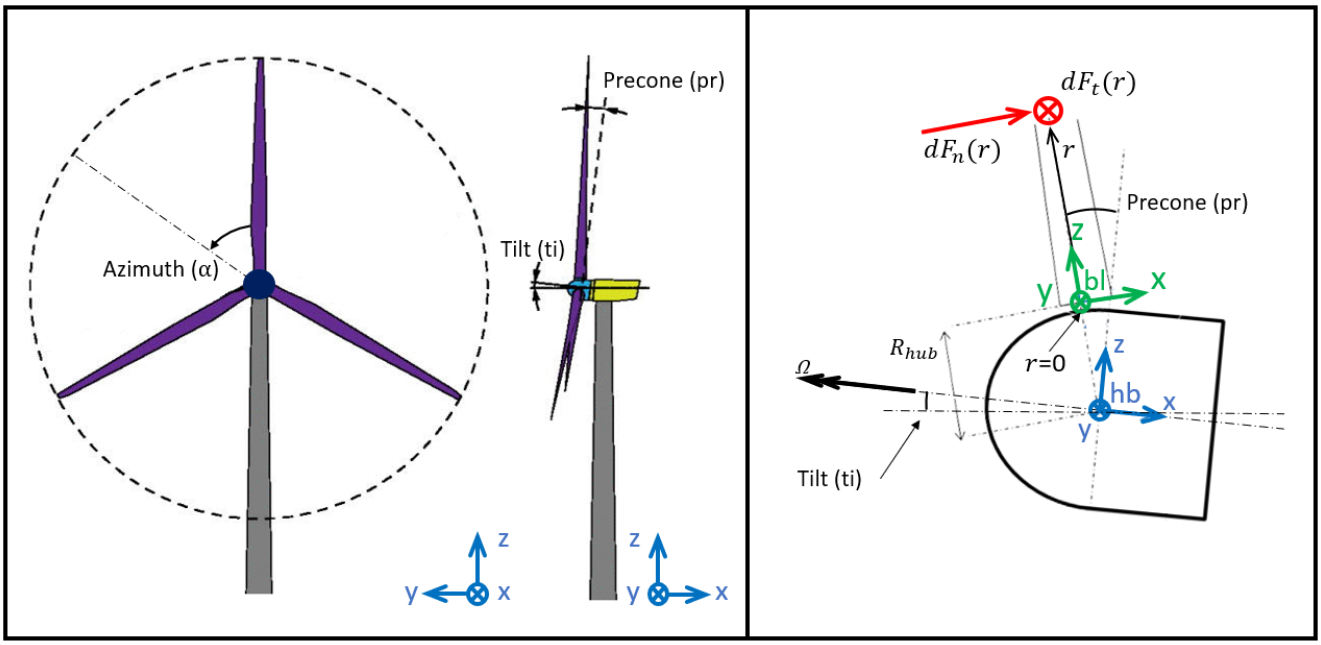

Figure 3.14: Rapresentation of azimuth, tilt and precone angles (left), and rapresentation of forces per unit length acting at radius r of blade (bl) reference frame (right).

To consider these facts, we define v_{w_i} as relative wind velocity along x direction of blade i reference frame calculated on point ct, center of thrust, and $U^0_{\infty,m}$ as the undisturbed wind mean velocity (assumed along x direction of world reference frame w.r.t.).

$U^0_{\infty,m}$ is the mean of undisturbed wind velocities calculated on N points along blade length, whose distribution considers that elements at higher radii are more important in term of total aerodynamic torque. Similar considerations can be made about center of thrust. Figure (3.15) shows an example of distribution with N=4 points.

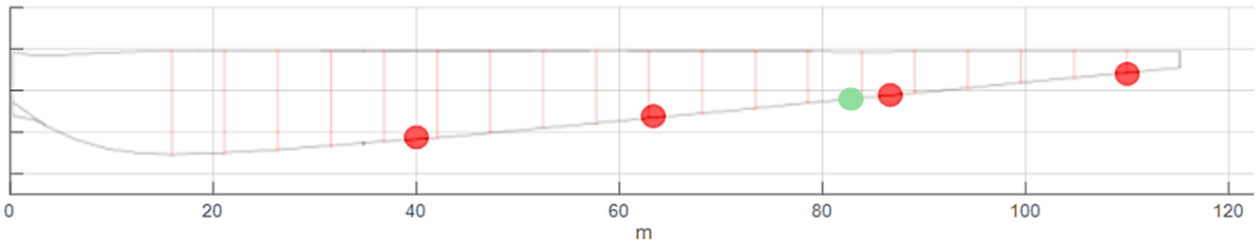

Figure 3.15: Points at which wind speeds are computed (red) and center of thrust (green).

Here there are the expression of v_{w_i}, where 0 superscript indicates quantities are in w.r.t. coordinates. v^0_{ct,bl_i} is center of thrust relative velocity of blade i, n^0_{x,bl_i} is a versor parallel to x-direction of blade i reference frame and Ω is rotor angular velocity. R_{hub} and R_{ct,bl_i} are hub radius and distance between blade root and blade center of thrust in z-direction of blade i reference frame.

$$v_{w_i} = \left(U^0_{\infty,m} - v^0_{ct,bl_i}\right) \cdot n^0_{x,bl_i} \qquad (3.86)$$

$$v^0_{ct,bl_i} = \begin{Bmatrix} \dot{s} \\ 0 \\ \dot{h} \end{Bmatrix} + R_y(p) \cdot \left(\begin{Bmatrix} 0 \\ \dot{p} \\ 0 \end{Bmatrix} \wedge r^{wt}_{hub} \right) + R_y(p + t_i) \cdot \left(\begin{Bmatrix} \Omega \\ \dot{p} \\ 0 \end{Bmatrix} \wedge R_y(pr) \cdot \begin{Bmatrix} 0 \\ 0 \\ R_{hub} + R_{ct,bl_i} \end{Bmatrix} \right) \qquad (3.87)$$

With: $$R_y(*) = \begin{bmatrix} \cos * & 0 & \sin * \\ 0 & 1 & 0 \\ -\sin * & 0 & \cos * \end{bmatrix}$$

Aerodynamic forces computation

Once system (3.79) is improved with mentioned corrections, it can be iteratively solved to calculate, through integration along radius r (see Figure 3.14), forces and moment at blades root starting from infinitesimal normal and tangential forces. Forces and moment at blades root depend on actual undisturbed wind velocity, rotor angular speed and blade pitch. Below the expressions for a singular blade and an image showing an example of torque $M^{bl}_x(v_{w_i}, \Omega, \theta_{bl})$ for some values of v_{w_i}, Ω and θ_{bl}.

$$F^{bl}_x(v_{w_i}, \Omega, \theta_{bl}) = \int_0^R dF_n(r) \qquad (3.88)$$

$$F^{bl}_y(v_{w_i}, \Omega, \theta_{bl}) = \int_0^R dF_t(r) \qquad (3.89)$$

$$M^{bl}_y(v_{w_i}, \Omega, \theta_{bl}) = \int_0^R r \cdot dF_n(r) \qquad (3.90)$$

$$M^{bl}_x(v_{w_i}, \Omega, \theta_{bl}) = \int_0^R -r \cdot dF_t(r) \qquad (3.91)$$

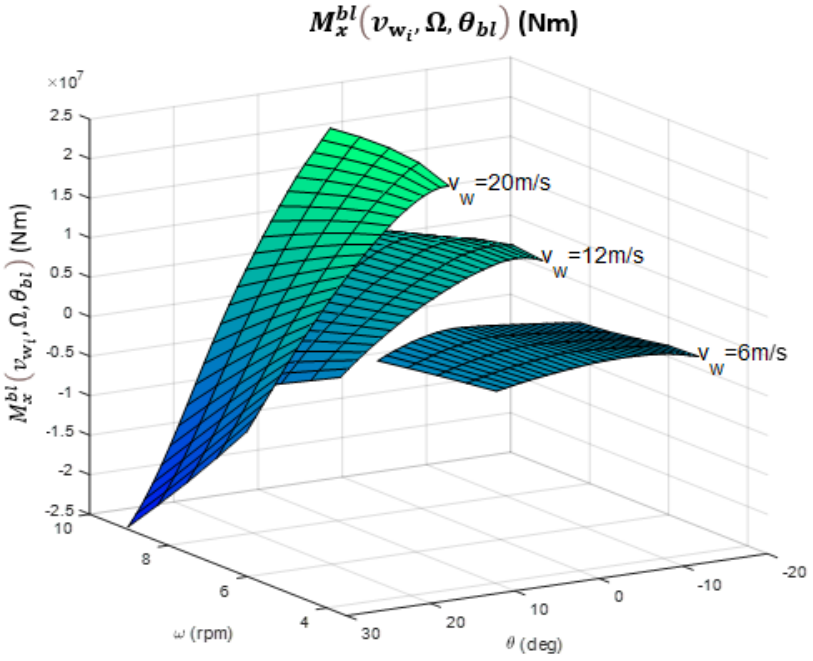

Figure 3.16: Example of torque $M^{bl}_x(v_{w_i}, \Omega, \theta_{bl})$ for some values of v_{w_i}, Ω and θ_{bl}

Once forces and torques at blades roots for each blade are known, total torque applied on the hub and total forces and torques applied on the entire structure can be calculated. The first step is finding, for each blade, aerodynamic forces and torques expressed in the hub reference frame:

$$F_i^{hu} = R^{hu-bl} \cdot \begin{Bmatrix} F_x^{bl_i} \\ F_y^{bl_i} \\ 0 \end{Bmatrix} \tag{3.92}$$

$$M_i^{hu} = R^{hu-bl} \cdot \left(\begin{Bmatrix} M_x^{bl_i} \\ M_y^{bl_i} \\ 0 \end{Bmatrix} + \begin{Bmatrix} 0 \\ 0 \\ R_{hub} \end{Bmatrix} \wedge \begin{Bmatrix} F_x^{bl_i} \\ F_y^{bl_i} \\ 0 \end{Bmatrix} \right) \tag{3.93}$$

With:
- $R^{hu-bl} = R_x(\alpha_i) \cdot R_y(pr)$
- $\alpha_i = \alpha_1 + \frac{2}{3}\pi(i-1)\ with\ \ i = 1, \cdots, B$
- $R^{0-bl_i} = R_y(p + ti) \cdot R_x(\alpha_i) \cdot R_y(pr)$

So, total aerodynamic torque around hub x-direction is:

$$C_{\text{aero}} = \sum_i M_{i,x}^{\text{hu}} \tag{3.94}$$

While forces and torques around wt point, expressed in w.r.f. coordinates are:

$$F^0 = R^{0-hu} \cdot \left(\sum_i F_i^{hu} \right) \tag{3.95}$$
$$M^0 = R^{0-hu} \cdot \left(\sum_i M_i^{hu} \right) + r_{hub}^0 \wedge F^0 \tag{3.96}$$

With: $R^{o\text{-}hu} = R_y(p + t_i), \quad r_{hub}^0 = R^{o\text{-}wt} \cdot r_{hub}^{wt}, \quad R^{o\text{-}wt} = R_y(p)$

$$R_x(*) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos * & -\sin * \\ 0 & \sin * & \cos * \end{bmatrix}, \qquad R_y(*) = \begin{bmatrix} \cos * & 0 & \sin * \\ 0 & 1 & 0 \\ -\sin * & 0 & \cos * \end{bmatrix}$$

Finally, aerodynamic forces vector present in structure equation of motion is:

$$F_{\text{aero}} = \begin{Bmatrix} F_x^0 \\ F_z^0 \\ M^0{}_y \end{Bmatrix} \tag{3.97}$$

3.1.1.4 $F_{Mooring}$

$F_{Mooring}$ in the generalized force vector containing forces and moment due to mooring lines action. During simulation of the system, they are computed thanks to pre-calculated results from an external software, called MAP (Mooring Analysis Program), designed to model static and dynamic forces of mooring systems ([14], [15]). It was developed by Marco Masciola with the National Renewable Energy Laboratory (NREL).

In this paper, the implementation of a multisegmented, quasi-static (MSQS) mooring model is used. MSQS model is developed as an extension of a single-line element theory combining many individual catenary cables at common connection points. Once combined, static equilibrium is achieved when the connection point forces sum is zero. Figure below shows an example of mooring system with a line composed by 3 catenary elements and 4 nodes (node 1 is linked with seabed, while 3 and 4 with the floating platform).

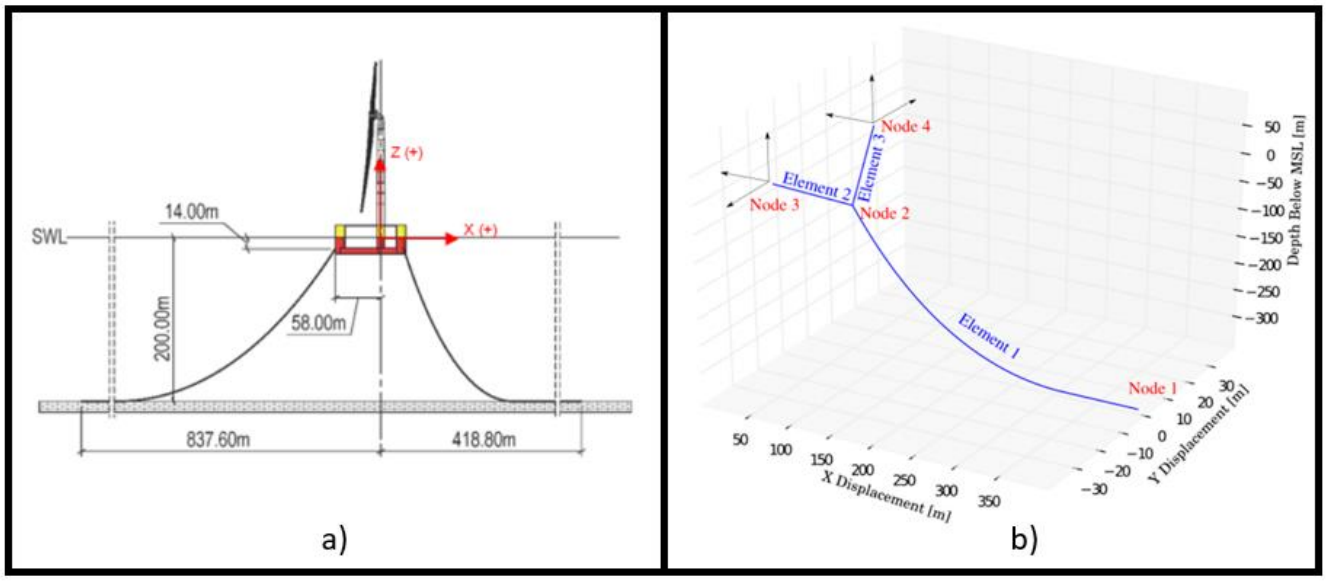

Figure 3.17: a) Example of mooring lines layout. b) Example of a multisegment mooring model of a line with 3 elemnts and 4 nodes.

This model requires two different sets of equations to be solved. The first is the continuous catenary algebraic equations, while the second set checks if equilibrium is reached by compute the sum of forces at the connection nodes.

First equations set are solutions to the common closed-form analytical equation for a single-line cable element hanging between two fixed points. Firsts of them are valid for a hanging cable that is not in contact with the seabed (case A in Figure 3.18). The horizontal l and vertical h fairlead displacement can be calculated knowing cable characteristics and nodes forces H and V:

$$l = \frac{H}{W}\left[\sinh^{-1}\left(\frac{V}{H}\right) - \sinh^{-1}\left(\frac{V-WL}{H}\right)\right] + \frac{HL}{EA} \qquad (3.98)$$

$$h = \frac{H}{W}\left[\sqrt{1+\left(\frac{V}{H}\right)^2} - \sqrt{1+\left(\frac{V-WL}{H}\right)^2}\right] + \frac{1}{EA}\left(VL - \frac{WL^2}{2}\right) \qquad (3.99)$$

Lasts of them are valid for a cable in contact with the seabed (case B in Figure 3.18). The horizontal l and vertical h fairlead displacement can be calculated knowing cable and seabed characteristics and nodes forces H and V:

$$l = L_B + \frac{H}{W}\sinh^{-1}\left(\frac{V}{H}\right) + \frac{HL}{EA} + \frac{C_B W}{2EA}\left[\mu\left(L - \frac{V}{W} - \frac{H}{C_B W}\right) - \left(L - \frac{V}{W}\right)^2\right] \quad (3.100)$$

$$h = \frac{H}{W}\left[\sqrt{1 + \left(\frac{V}{H}\right)^2} - 1\right] + \frac{V^2}{2EAW} \quad (3.101)$$

$$\mu = \begin{cases} L - \frac{V}{W} - \frac{H}{C_B W} & \text{if } \left(L - \frac{V}{W} - \frac{H}{C_B W}\right) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (3.102)$$

Where:

- H\H_A: Horizontal fairlead\anchor force;
- V\V_A: Vertical fairlead\anchor force;
- A: Cable cross-section line;
- E: Young's modulus;
- g: Gravity acceleration;
- ρ_{cable}: Cable density;
- ρ: Fluid density;
- L: Unstretched line length;
- L_B: Line length resting on the seabed;
- C_B: Seabed friction coefficient;
- x_0: Horizontal force transition point;
- W: cable mass per unit length.

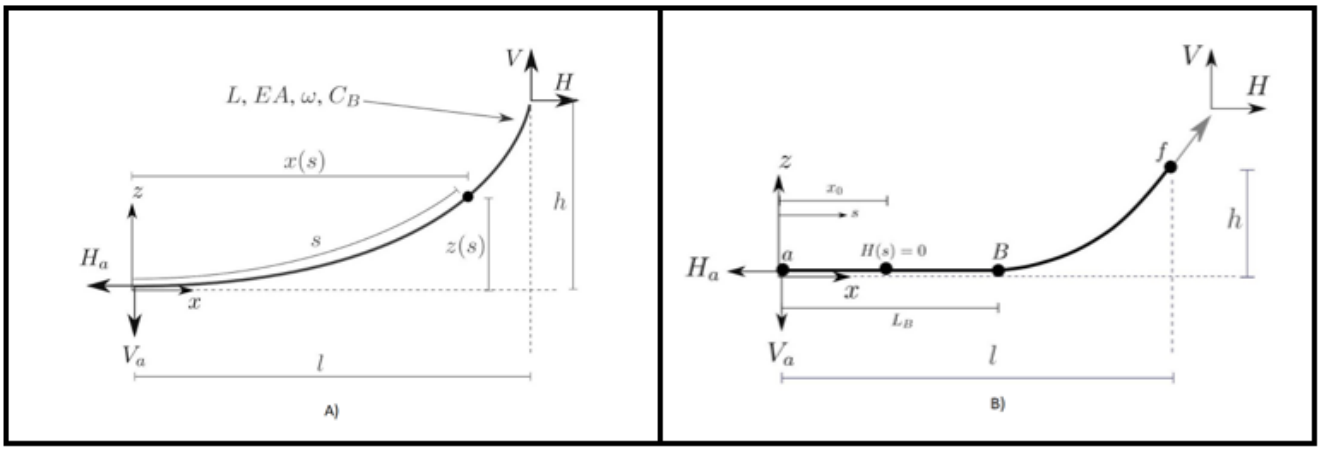

Figure 3.18: A) Hanging cable. B) Cable in contact with seabed.

Second set of equations checks if each node is in an equilibrium condition; for this to happen, the sum of the tension forces of the elements connected to a node must equal the sum of the external forces applied to it. Equations for j-node are:

$$F_x^j = \sum_{i=1}^{N} H_i \cos\ \alpha_i - Fx_j^{ext} < \varepsilon \qquad (3.103)$$

$$F_y^j = \sum_{i=1}^{N} H_i \sin\ \alpha_i - Fy_j^{ext} < \varepsilon \qquad (3.104)$$

$$F_z^j = \sum_{i=1}^{N} V_i - Fz_j^{ext} + M_j g - gB_j \rho < \varepsilon \qquad (3.105)$$

Where:

- N: Number of elements linked with node j;
- M_j : Point mass applied to the j-node;
- B_j : Displaced volume applied to j-node;
- F ext: external force applied to j-node;
- α_i: Angle between i-element and global X direction.

Once these equations are obtained, model of each mooring line can be solved iteratively for any structure position. The solution process begins by guess position of those nodes which isn't linked with structure or seabed and then by evaluating the two continuous analytical catenary equations for each element based on l and h values obtained through node displacement relationships. An element is defined as the component connecting two adjacent nodes together.

Once the element fairlead (H, V) and anchor (H_A, V_A) forces are solved at the element level, the forces are transformed from the local element frame into the global coordinate system. The force contribution at each element's anchor and fairlead is added to the corresponding node it attaches to. Then thee force-balance equation is evaluated for each node, where ε is the convergence tolerance limit. Based on the error of Eq. (3.103 – 3.104), the node position is updated. As an outcome, the element forces must be recalculated, and the process begins again. When the solution is reached for all lines, forces of nodes linked with the structure are manipulated to obtain total force and moment acting on the structure, computed at a chosen point.

For this work, remembering the 3 d.o.f. assumptions, a certain number of combinations of surge, heave and pitch are chosen to obtain $F_{x,\,\boldsymbol{Mooring}}$, $F_{z,\,\boldsymbol{Mooring}}$ and $M_{y,\,\boldsymbol{Mooring}}$ acting at wt-reference frame center, then these result will be exploit, by interpolation, during the simulation to calculate mooring actions for any structure position $\boldsymbol{X}$.

$$\boldsymbol{F_{Mooring}} = \begin{Bmatrix} F_{x,\boldsymbol{Mooring}}\,(\boldsymbol{X}) \\ F_{z,\boldsymbol{Mooring}}\,(\boldsymbol{X}) \\ M_{y,\boldsymbol{Mooring}}\,(\boldsymbol{X}) \end{Bmatrix},\ \boldsymbol{X} = \begin{Bmatrix} surge \\ heave \\ pitch \end{Bmatrix} \qquad (3.106)$$

Figure below shows mooring forces and moment with respect to structure displacements. Every component of interest is represented with respect to a motion mode keeping other displacements at their rest value.

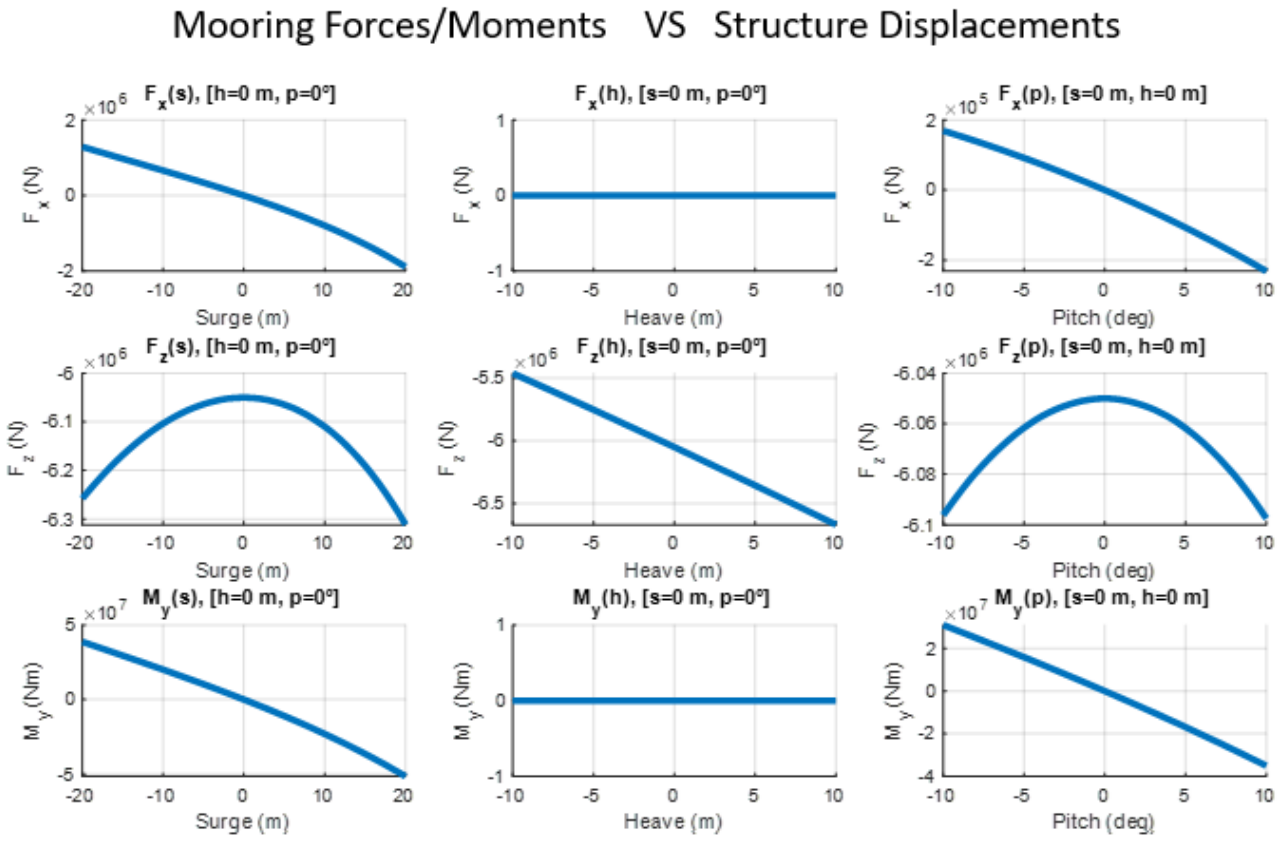

Figure 3.19: Mooring forces and moment with respect structure displacements. Every component of interest is represented with respect to a motion mode keeping other displacements at their rest value (surge=0 m, heave=0 m, pitch=0°).

3.1.2 Rotor equation

In this section equation for fourth degree of freedom of the model is found. It is the rotational equilibrium equation around x axis of hub reference frame. In this work we consider an ideal gearbox between generator and rotor shafts (unitary efficiency factor) and rigid and massless shafts. From figure (3.20) following relation can be write:

$$C_{\text{aero}} - C_{gen,,rot} = I_{\text{rot}}\,\dot{\Omega}_{rot} \qquad (3.107)$$

$$C_{rot,gen} - C_{gen} = I_{gen}\dot{\Omega}_{gen} \qquad (3.108)$$

$$I_{eq} = I_{rot} + N^2 \cdot I_{gen} \qquad (3.109)$$

With:

- C_{aero} : Aerodynamic torque around x- direction of hub reference frame (equation 3.82);
- $C_{gen,,rot}$: Internal torque from gearbox low speed shaft to rotor (same as $C_{rot,gen}$);
- I_{rot} : Rotor moment of inertia (sum of hub's and blades' ones);
- I_{gen}: Generator rotor moment of inertia;
- N: gearbox ratio, defined as $N = \Omega_{gen}/\Omega_{rot} = C_{gen,rot}/C_{rot,gen}$.

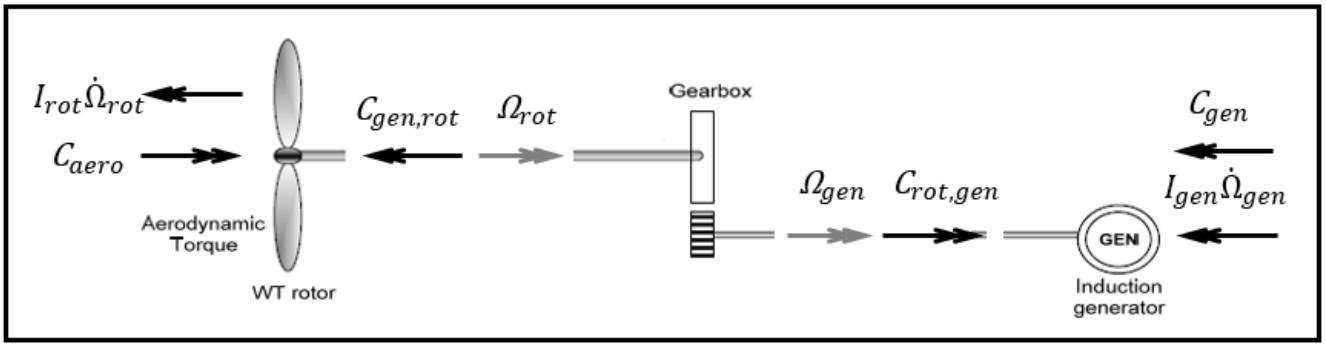

Figure 3.20: Free body diagram of rotor and generator shafts.

Then, from equations (3.107–3.109), rotor angular acceleration can be expressed in term of I_{eq} , C_{aero} (which depend on undisturbed wind speed field, rotor angular velocity, blade pitch angle and structure motion) and C'_{gen}, defined as $C'_{gen} = N \cdot C_{gen}$.

$$\dot{\Omega}_{\text{rot}} = \frac{1}{I_{\text{eq}}}\left(C_{\text{aero}} - C'_{gen}\right) \qquad (3.110)$$

3.2 External Inputs

In this section external inputs of the model are defined; they are wind speed field and excitation forces/moments due to incident waves.

3.2.1 Wind

System model receives as wind input the spatial distribution of wind undisturbed velocities along a spatial vertical grid whose origin is located at hub centre (see Figure 3.21). This grid is uniformly discretized with n points along z and y axis to obtain N=n x n total input velocities, which are obtained with TurbSim [16], a stochastic, full-field, turbulent wind generator. It uses statistical models to calculate time series of a 3D wind field. In this work it is assumed presence of only speed component along surge axis (X in Figure 3.21), that is only U component is not null (which corresponds to undisturbed wind speed U_∞ introduced in section (3.1.1.3), that, for simplicity, will no longer have subscript.

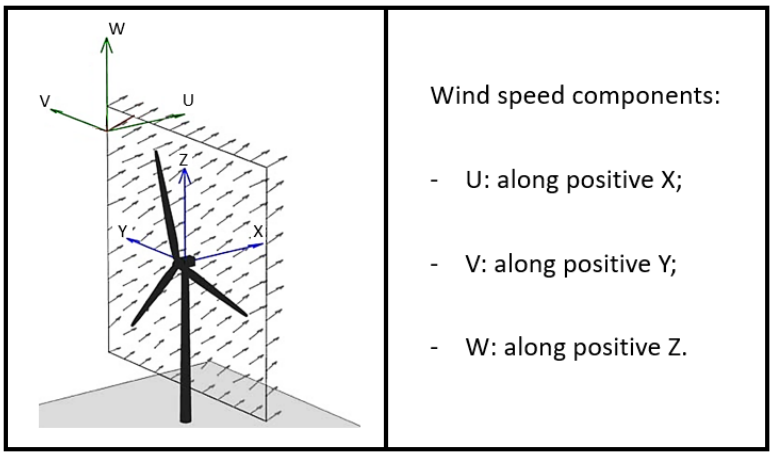

Figure 3.21: Example of 3D wind speed field on a grid lying on rotor plane.

U field depend on time, Y and Z and it is defined as the sum of a mean component U_m, function of only height Z, and a turbulent one, U_t:

$$\boldsymbol{U}(t,Y,Z) = \boldsymbol{U_m}\,(Z) + \boldsymbol{U_t}\,(t,Y,Z) \qquad (3.111)$$

Mean speed component profile is calculated considering surface roughness and other factors, at swl it is null, then it increases with height until far-field value. It can be expressed, as suggested by Jonkman [17], by a power law, as a function of wind mean velocity at a reference height H and an exponent α:

$$\boldsymbol{U_m}(Z) \;=\; \boldsymbol{U_m}(h_{hub}) \cdot \left(\frac{z}{h_{hub}}\right)^{\alpha} \qquad (3.112)$$

In this case exponent α=0.14, according to IEC 61400-3 [18] and assuming offshore wind turbine and hub height h_{hub} is chosen as reference height.

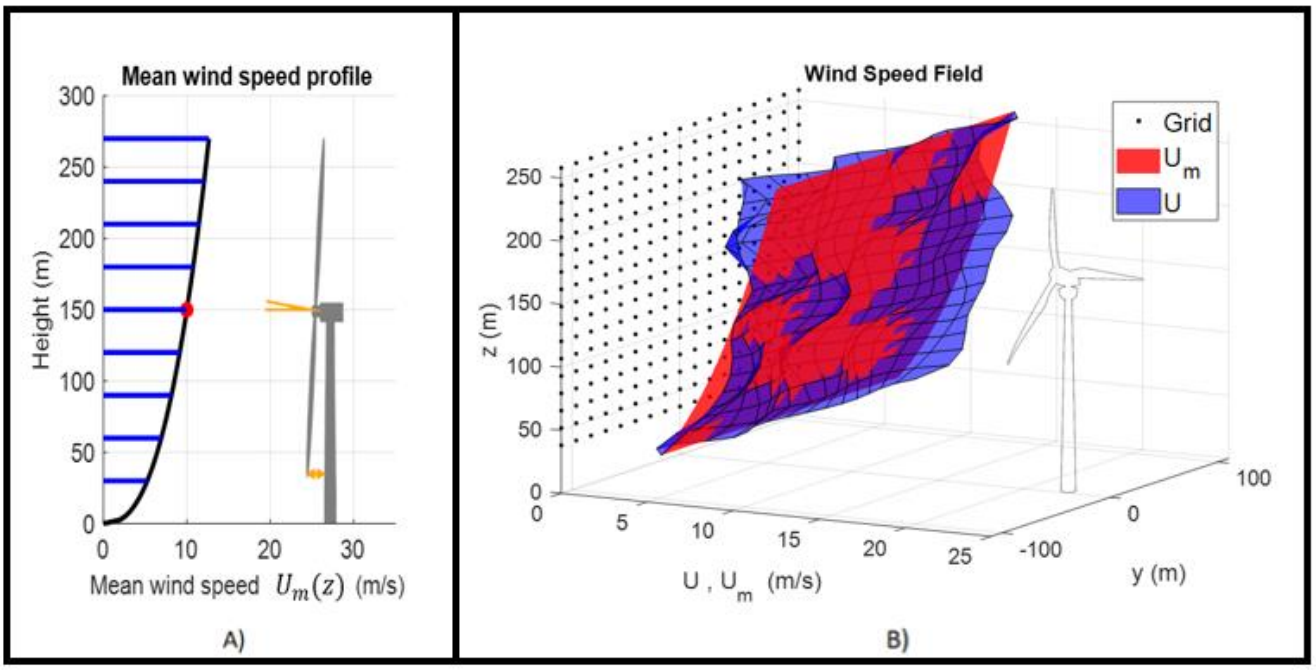

Figure 3.22: A) Mean wind speed profile with respect to height. B) Example of wind speed field, Um is mean wind speed (red), while U is total wind speed,along x direction (blue). Black dots are locations of points at which velocitites are computed.

Turbulent component $\boldsymbol{U_t}\ (t, Y, Z)$ is generated as a random signal in which its turbulence intensity TI is specified. Turbulence intensity is defined as $TI = \sigma / U_{m,Hr}$, where σ is signal's standard deviation (assumed equal for all points) and $U_{m,Hr}$ is wind mean speed at a reference height H_r. Signal generation starts from this information and from choice of a spectrum type (for instance, Kaimal spectrum can be choose [19]), which determines the form of power spectral density (PSD) of the signal, related to signal power content at each frequency. Once PSD is obtained, Fourier transform of the signal is computed, also considering coherence spatial needs. As an example, two neighbouring points have more similar velocities than two points with high distance between each other. Moreover, other factors influence signal computation, for instance, low frequencies speed components are more correlated than high frequencies ones. Detailed information about turbulent wind speed computation can be find in TurbSim User's Guide [16]. Figure 3.22 shows an example of wind field at a certain instant.

Input of the treated model will be, at any time, a vector $\mathbf{U_{wind}}$ containing speed values of all grid points and their coordinates.

3.2.2 Waves

External inputs related to waves action are forces and moment due to excitation velocity potential introduced in section 3.1.1.1, which considers incident and diffracted waves together. In this work irregular waves along surge direction are considered, that is all waves travel along x direction and their free surface elevation can take any form. In this case it is constructed from a linear superposition of N regular waves:

$$\eta(t) = \sum_{i=1}^{N} \eta_{0_i} \cdot \cos\left(\omega_i t - k_i x + \varphi_i\right) \quad (3.113)$$

- ω : $wave\ frequency\ (rad/s)$;
- η_0 : $wave\ amplitude\ (m)$;
- k : $wave\ number\ (m^{-1})$;
- φ : $phase\ angle\ (rad)$.

According to Equation 3.33, the forces along x and z direction (surge and heave) and the torque around y direction (pitch) are defined as follows:

$$\boldsymbol{F}_{ext}(t) = \begin{Bmatrix} F_{\text{ext}_x}(t) \\ F_{\text{ext}_z}(t) \\ M_{\text{ext}_y}(t) \end{Bmatrix} = Re\left[\sum_1^N \eta_{0,i} \cdot \widetilde{\boldsymbol{F}}_{\text{ext}}(\omega_i) \cdot e^{j(\omega_i t + \varphi_i)}\right] \quad (3.114)$$

Where $\widetilde{\boldsymbol{F}}_{\text{ext}}(\omega_i)$ is frequency dependent complex wave-excitation amplitude vector, which, as mentioned in section 3.1.1.1, is computed resolving the boundary value problem 3.12 with BEM solver NEMOH,[8].

Wave amplitude of each component is obtained from an energy spectrum which represent the energy content at different frequencies. If we define $\bar{E}$ (J/m^2) as the wave energy per unit of surface area, omnidirectional energy spectrum S(f) ($m^2 s$) can be defined according to:

$$\frac{\bar{E}}{\rho g} = \sum_i \bar{E}_i / \rho g = \sum_i S(f_i) \cdot \Delta f_i \quad (3.115)$$

Where f_i and Δf_i are the i-frequency and its frequency range, while $\bar{E}_i$ is the energy per unit area associated to a single component. From Airy wave theory, $\bar{E}_i$ is:

$$\bar{E}_i = \frac{1}{2} \rho\, g\, {\eta_0}^2 \quad (3.116)$$

So, wave amplitude of i-component can be expressed in term of energy spectrum:

$$\eta_{0_i} = \sqrt{2 \cdot S(f_i) \cdot \Delta f_i} \quad (3.117)$$

As mentioned above, we consider omnidirectional spectrum, which represent full energy content associated to unidirectional waves (in this case along surge direction). Moreover, to generate a no angular dependency (Gaussian) sea, phase angle of each component is chosen as a random value between 0 and 2π.

Many spectra can be used to compute wave elevation of an irregular sea condition, for example Pierson and Moskowitz one [20], who considers a fully-developed sea, that is waves reaches equilibrium with the wind after it has been blowing for a long time and over a wide area. To obtain the spectrum, they use measurements from accelerometers on British weather ships in the North Atlantic. Another spectrum, the one considered in this case, is the JONSWAP spectrum [21]. It assumes that sea is never fully developed, but it continues to develop through wave-wave interactions ever for long time and distance. JONSWAP spectrum S(f) is defined as:

$$S(f) = \alpha_\gamma \cdot \frac{5}{16} \cdot H_s^2 \cdot \frac{(2\pi f_p)^4}{(2\pi f)^5} \cdot e^{-\frac{5}{4}\cdot\left(\frac{f}{f_p}\right)^{-4}} \cdot \gamma^{e^{-\frac{(f-f_p)^2}{2\sigma^2 f_p^2}}} \qquad (3.118)$$

With: $-\ \alpha_\gamma = 1 - 0{,}287 \cdot log\,(\gamma);$ $-\ \sigma = \begin{cases} 0{,}01 & if \quad f \le f_p \\ 0{,}09 & if \quad f > f_p \end{cases};$ $-\ \gamma = 3.3;$

$-\ H_s = 4 \cdot \sqrt{m_0};$ $-\ m_k = \sum_{i=1}^{N} f_i^{\,k} S(f_i) \cdot \Delta f_i;$ $-\ T_p = 1/f_p$

T_p is the peak period, at which energy density is the maximum, H_s is the significant height, which is wave height a regular wave would have if all sea energy was associated to that wave. m_k are the k-moments of the spectrum, for instance, m_0 is the total energy content per unit of area, density, and gravity acceleration.

Figure below shows an example of JONSWAP spectrum with different values of peak period (T_p) and significant height (H_s).

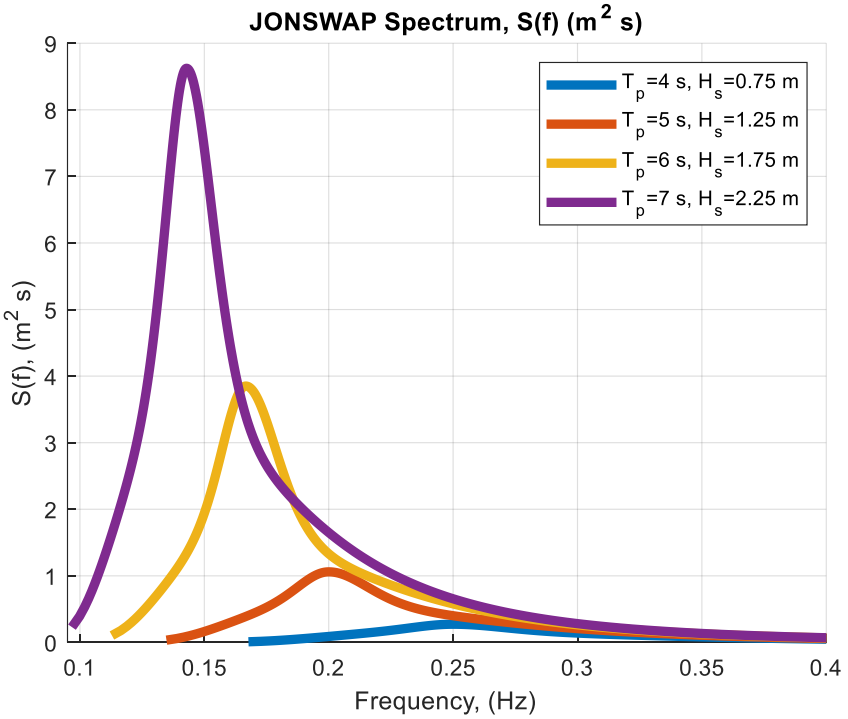

Figure 3.23: JONSWAP spectrum for some values of peak period (T_p) and significant height (H_s).

3.3 State space form of the model

Once equations describing the floating wind turbine are obtained and once external inputs are defined, we can express the non-linear model equations in a compact form which will be useful in the next sections. It is a MIMO state space form, defined as:

$$\begin{cases} \dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{v}(t)) \\ \boldsymbol{y}(t) = g(\boldsymbol{x}(t), \boldsymbol{u}(t)) \end{cases} \quad (3.119)$$

With:

$$\boldsymbol{x} = \{s, h, p, \dot{s}, \dot{h}, \dot{p}, \alpha, \Omega, \ \boldsymbol{q}^T{}_R\}^\top$$

$$\boldsymbol{u} = \left\{C'_{\text{gen}}, \theta_{bl}\right\}^\top$$

$$\boldsymbol{v} = \{\boldsymbol{U}^T_{\text{wind}}, \boldsymbol{F}^T_{\text{ext}}\}^\top$$

$$\boldsymbol{y} = \{s, h, p, \dot{s}, \dot{h}, \dot{p}, \alpha, \Omega, P\}^\top$$

$\boldsymbol{x}$ is the states vector, composed by the four d.o.f. considered (surge, heave, pitch, and azimuth angle of the rotor), their derivatives, and $\mathbf{q}_r$ (equation 3.43), which are the state vector of linear state space representation of convolution integral related to radiation damping term, part of hydrodynamics' forces (section 3.1.1.1). **u** and **v** are inputs vectors. Two different vectors are defined to separate external inputs (**v**) and controlled input (**u**), which will be the inputs of the system defined by control unit.

The controlled inputs are the torque electric generator exerts on the rotor, C'_{gen} , (defined in section 3.1.2), and the collective pitch angle of the blades θ_{bl}. The external inputs are the undisturbed wind velocity vector $\mathbf{U}_{\mathbf{wind}}$ (defined in section 3.2.1), and excitation forces vector $\mathbf{F}_{\text{ext}}$ (defined in section 3.2.2).

Finally, output vector **y** is composed by elements of states vector and power output, computed as: $P = C'_{\text{gen}} \cdot \Omega$.

Considering above definitions, function f is a vectorial function composed by $8+n_{\mathbf{q}R}$ scalar equations ($n_{\mathbf{q}R}$ is radiation states' number utilized for radiation damping convolution integral approximation). Inputs number is $n_x+n_u+n_v$, with $n_x=8$, $n_u=2$ and $n_v=Ngrid+3$, where N_{grid} is the number of grid points at which wind speed data are provided). Equations 1,2,3 and 7 are simply in the form: $\dot{x} = d\,x(t)/d\,t$, while equations 4,5,6 and 8 are structure and rotor relations found in the previous sections. Particularly, firsts three are vectorial structure equation 3.3 and last is rotor equilibrium 3.110. Vectorial equation g is composed by 9 scalar equations, with $n_x + n_u$ inputs.

3.4 Simplified model

In this section a simplified version of the previously obtained model is defined. As mentioned during the introductory part, in order to achieve the set objectives, it will be necessary to linearize the system model to obtain a set of LTI (linear Time Invariant) systems, each of which valid for certain parameter values specified by the choice of operational points around which the starting system is linearized. The simplifications that will be introduced are such as to obtain a model, again non-linear, but through which the computation and use of above mentioned LTI set is simplified.

Simplifications made are:

- **Wind speed input**: instead of having, as input, the wind speed for each point in a spatial grid, which would lead to a MIMO system with too many inputs, it was decided to consider only one speed data, e.g., the average speed (averaged over the grid points) or the speed relative to a certain point, e.g. at the hub. Once the speed has been chosen, it is considered to be constant along the entire grid. In system 3.120 it is named $U_{wind,m}$ since, in this case, mean speed is considered;

- **Azimuth state**: Considering above approximation and considering rotor pitch inertia do not depend on azimuth angle, the latter only influences calculation of absolute speed of a certain point along blade, so relative speed between wind and the latter. As can be seen by comparing Figures 3.33-3.35, considering or not the angular position of the rotor does not lead to great differences. Elimination of this state will lead to a simplification when linearizing the model, because it avoids having to consider numerous operating points at different azimuth values (since the sine and cosine functions, which appear in the model equations, are strongly non-linear, it would be necessary to divide one revolution into many angular segments). Furthermore, when the linear system used must be changed according to the current parameters, the elimination of this state leads to a reduction in the rate of change, making a certain linear model effective for a longer time;

- **Radiation damping**: in order to reduce states' number, so LTI set size and computation time, radiation damping forces and moment are neglected. This choice is made due to those actions are much smaller than other ones, i.e., excitation forces, since radiation damping is proportional to structure speed, which in this case, is limited.

State space representation of simplified model is:

$$\begin{cases} \dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{v}(t)) \\ \boldsymbol{y}(t) = g(\boldsymbol{x}(t), \boldsymbol{u}(t)) \end{cases} \quad \text{With:} \quad \begin{aligned} \boldsymbol{x} &= \{s, h, p, \dot{s}, \dot{h}, \dot{p}, \Omega\}^{\top} \\ \boldsymbol{u} &= \left\{C'_{\text{gen}}, \theta_{bl}\right\}^{\top} \\ \boldsymbol{v} &= \left\{U_{wind,m}, \boldsymbol{F}^{T}_{\text{ext}}\right\}^{\top} \\ \boldsymbol{y} &= \{s, h, p, \dot{s}, \dot{h}, \dot{p}, \Omega, P\}^{\top} \end{aligned} \tag{3.120}$$

3.5 Models validation

In this section, a mostly qualitative validation of the previously obtained models will be carried out. To perform this, the behavior of the systems will be compared with a reference system. The latter is a non-linear, time domain numerical model for the simulation of offshore floating wind turbines, named MOST [23]. The model can evaluate the movement of the platform in six degrees of freedom, the power production and the loads acting on the blades. It is implemented in Matlab-Simulink environment using Simscape Multibody. The aerodynamics is modelled with BEM (Blade Element Momentum) theory and the hydrodynamics is modelled using WEC-Sim, a Simscape library developed by NREL and SANDIA [24].

In other to compare the three model the same wind turbine and floating platform are considered, moreover, all inputs are equal, and the same control algorithm is adopted. Component characteristics are those mentioned in Section 2, whereas the controller used for validation is the Baseline controller described in Section 4. We do not focus here on the definition of the control algorithm as the main objective now is the comparison of the various systems, the important thing then being only that they all receive the same controlled input under the same conditions.

MOST and not simplified non-linear model (named NL) receive as wind input velocities of each point of the spatial grid obtained from TurbSim (specifying mean velocity at hub position and turbulence intensity), as specified in section 3.2.1, while non-linear simplified model (named NLS) receives spatial mean of those speeds as unique wind data. Wave inputs are excitation forces/moments, equal for all model simulated, obtained, according with section 3.2.2, once spectrum type, peak period and significant height are specified.

Figures below show time series of external inputs of simulations.

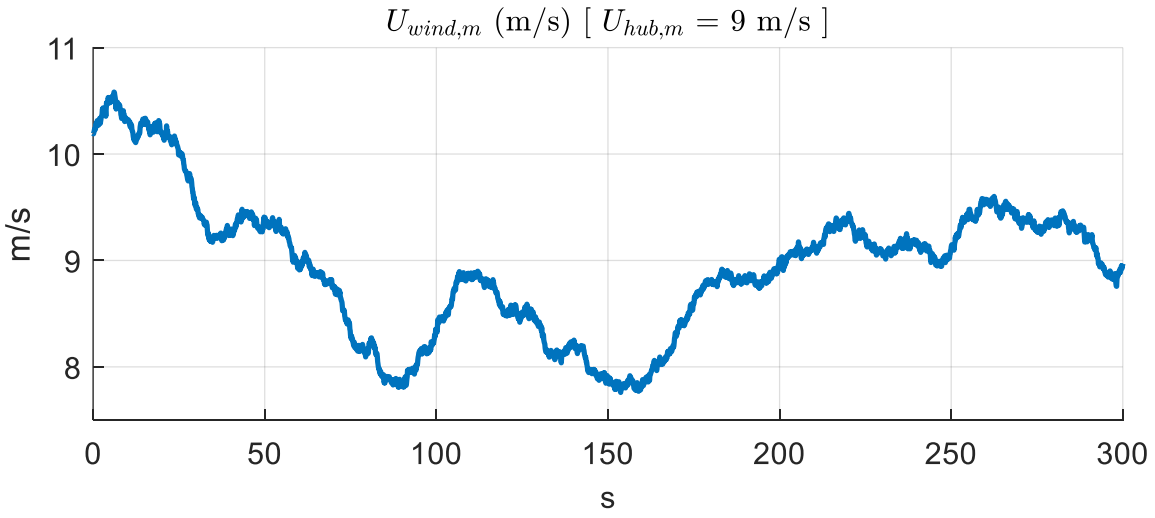

Figure 3.24: Time series of $U_{wind,m}$, spatial mean wind speed, obtained from TurbSim with hub temporal mean speed, $U_{hub,m} = 9\ m/s$.

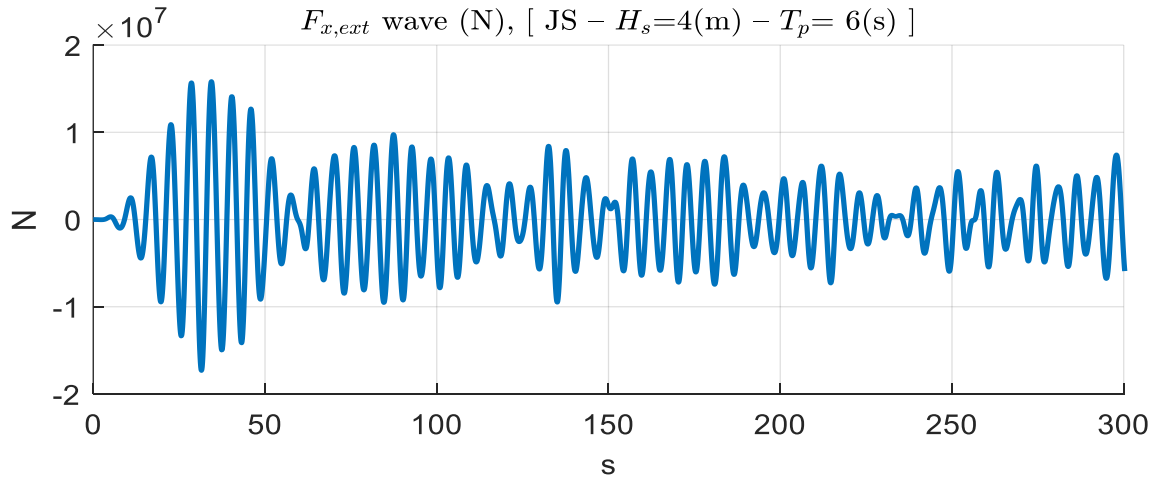

Figure 3.25: Time series of Fx,ext (wave excitation force along x direction), computed with a JONSWAP (JS) spectrum, significant height Hs=4 m and peak period Tp=6 s.

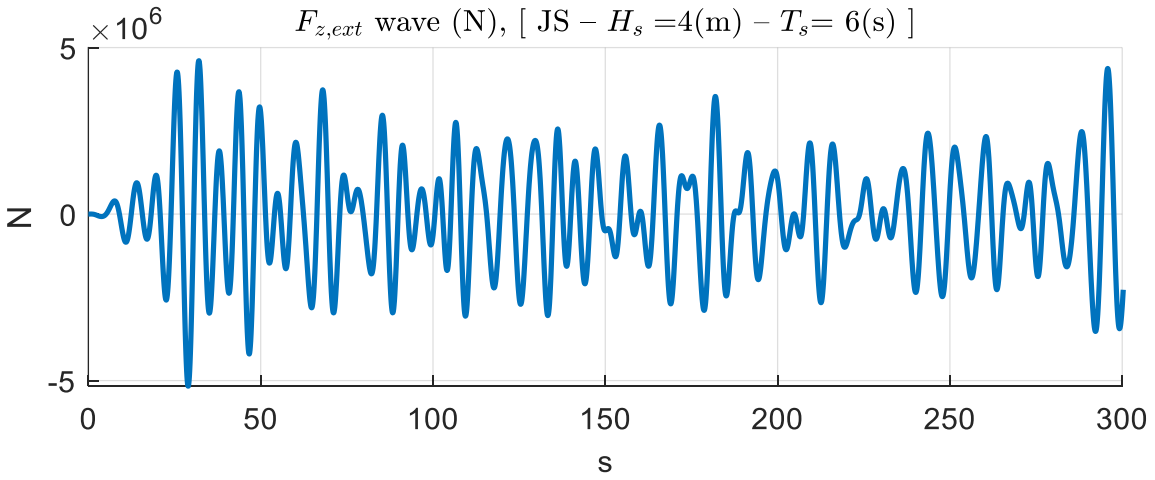

Figure 3.26: Time series of Fz,ext (wave excitation force along z direction), computed with a JONSWAP (JS) spectrum, significant height Hs=4 m and peak period Tp=6 s.

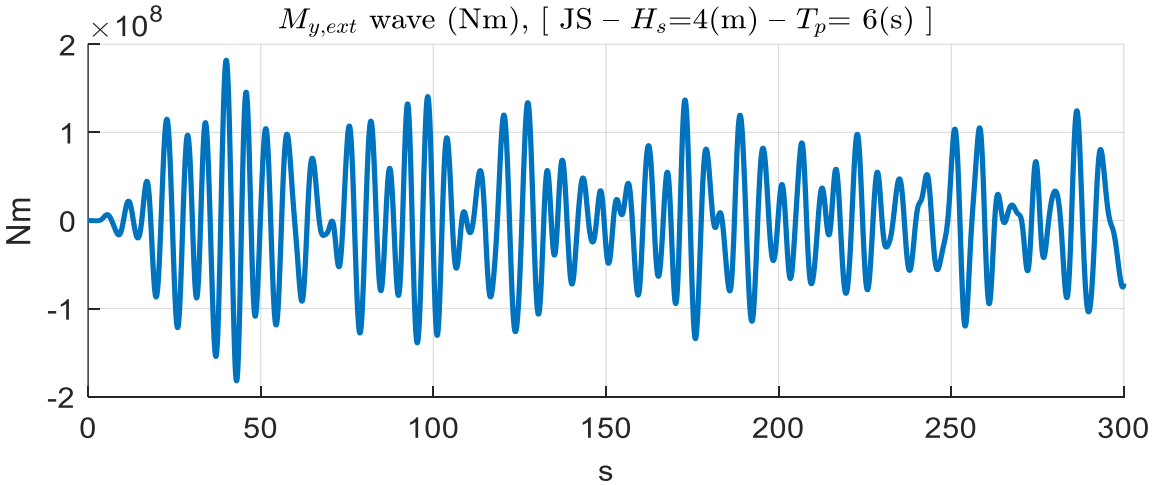

Figure 3.27: Time series of My,ext (wave excitation torque around y axis), computed with a JONSWAP (JS) spectrum, significant height Hs=4 m and peak period Tp=6 s.

Figures below show time some simulation result, that is time series of some of the output of the system described in section 3.3. Reported results are surge, heave, pitch, rotor angular speed and power output.

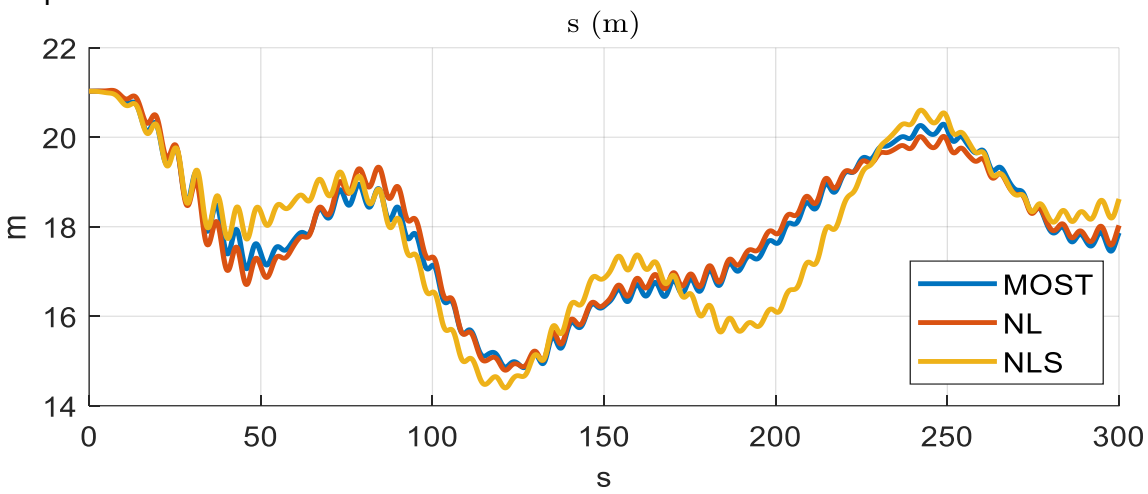

Figure 3.28: Comparison between NL, NLS and MOST models, surge (m).

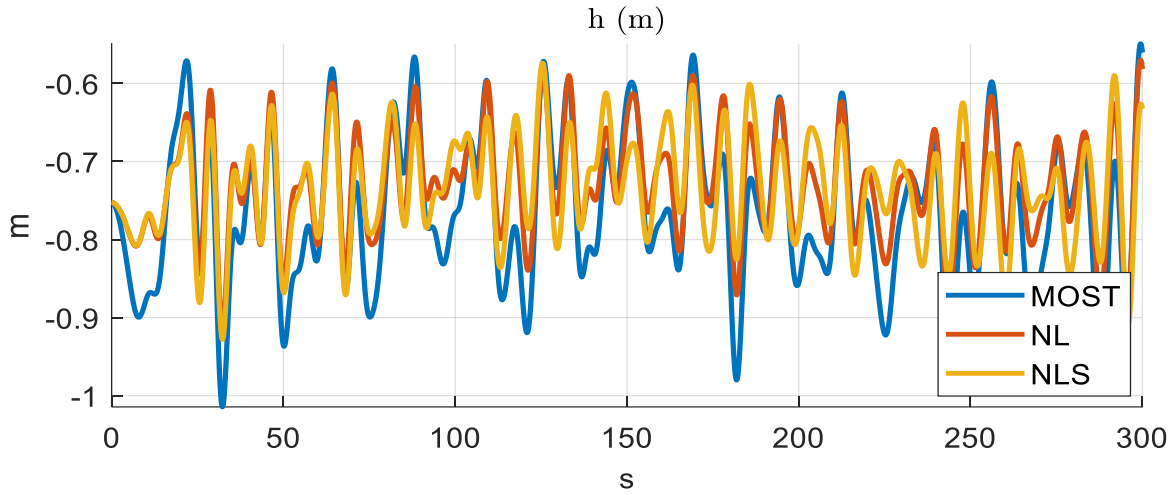

Figure 3.29: Comparison between NL, NLS and MOST models, heave (m).

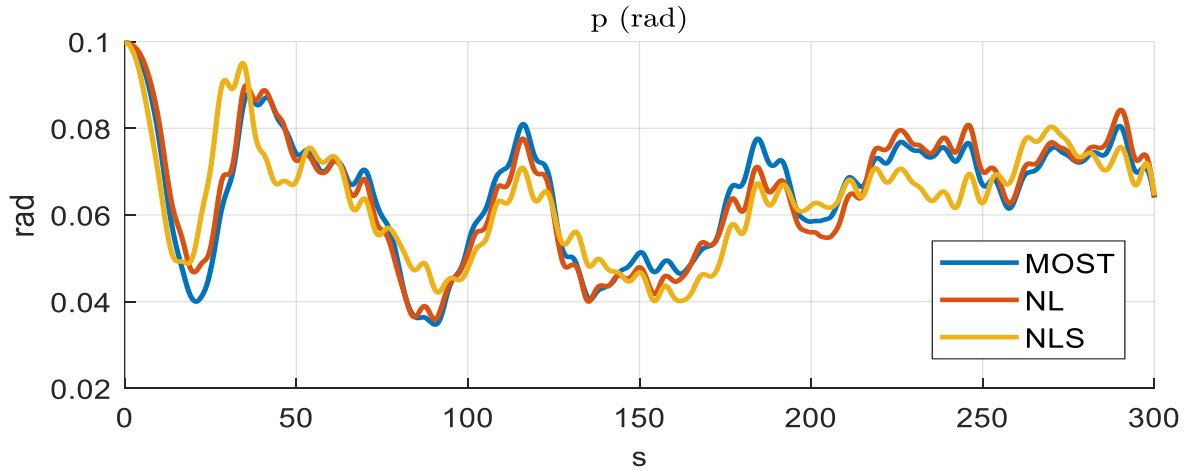

Figure 3.30: Comparison between NL, NLS and MOST models, pitch (rad).

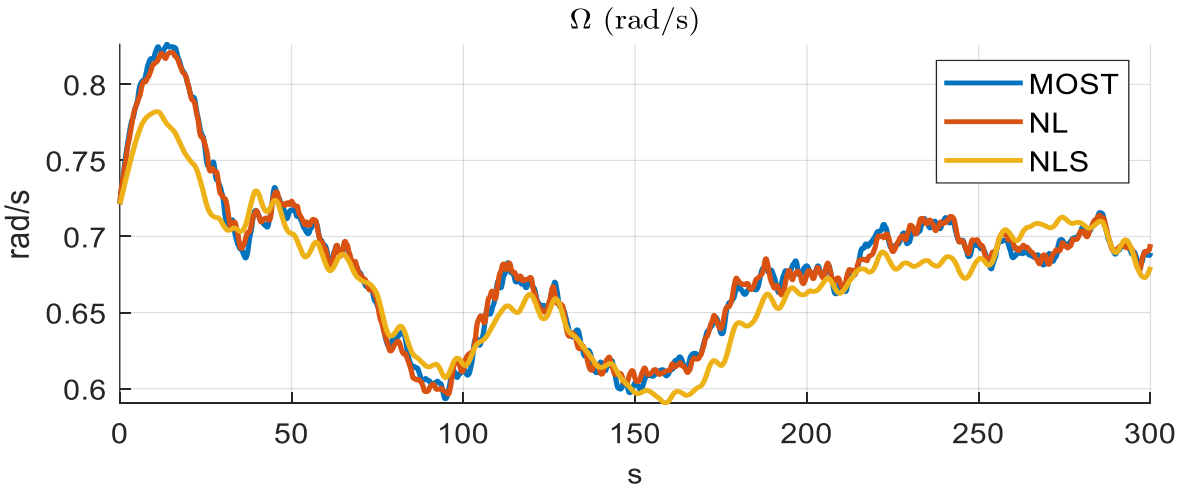

Figure 3.31: Comparison between NL, NLS and MOST models, rotor angular speed Ω (rad/s).

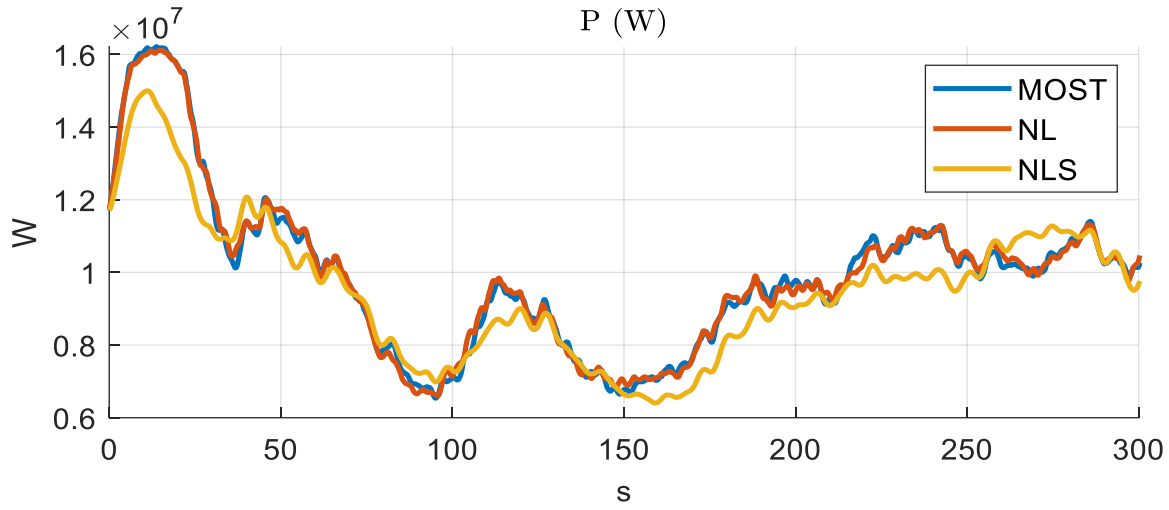

Figure 3.32: Comparison between NL, NLS and MOST models, output power (W).

As can be seen from the comparative graphs, the full model, NL, and the MOST model fit quite well, larger differences, however, are found in the simplified model, although the trends can still be considered satisfactory for the objectives set. Among the approximations made to obtain the simplified model described in section 3.4, the one that most influences the results is the assumption of having an equal wind speed at every point in space. To demonstrate this, the following page shows the simulation output time series of pitch, rotor angular speed and power, having this time chosen a constant speed (in space) as wind input, for all three simulated models. With this assumption, both NL and NLS systems' results fit well with MOST's ones.

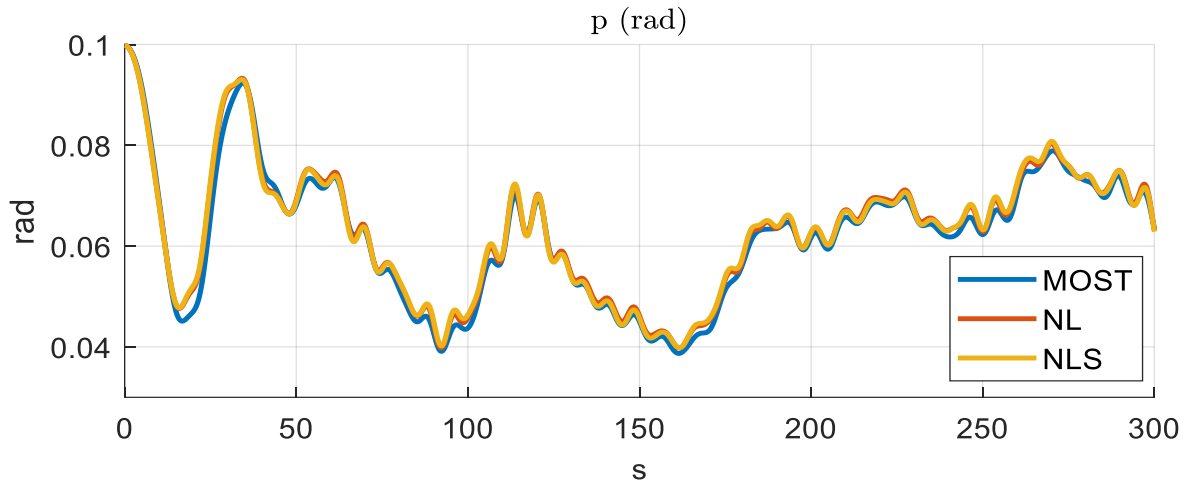

Figure 3.33 Comparison between NL, NLS and MOST models with spatial constant wind speed, pitch (rad).

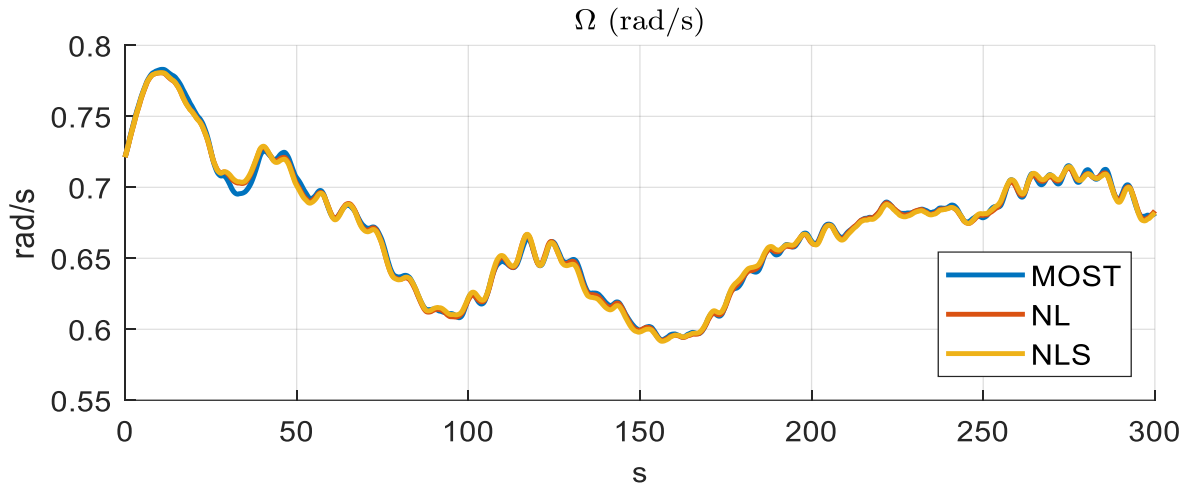

Figure 3.34: Comparison between NL, NLS and MOST models with spatial constant wind speed, angular speed Ω (rad/s).

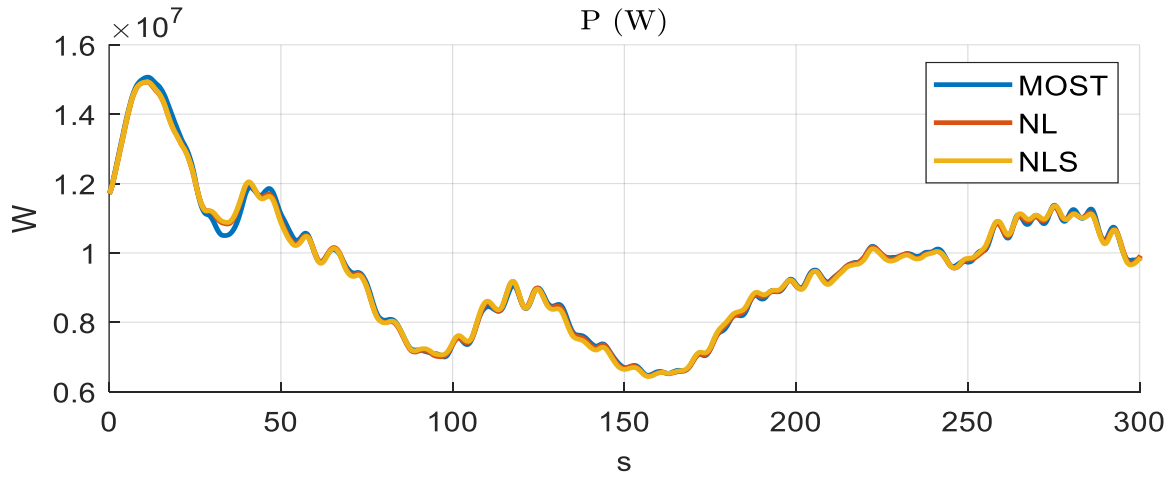

Figure 3.35: Comparison between NL, NLS and MOST models with spatial constant wind speed, power P (W).

4 Wind Turbine Control

In this section, the control of wind turbines will be discussed. First, the main objectives of the control unit will be identified, and then some of commonly used algorithms will be listed [25]. Next, some theoretical foundations will be taken up in order to study the main control strategies that can be used. Finally, two different control algorithms will be presented, which will then be compared with the MPC controller implemented in this work.

4.1 Overview of Wind Turbine Controllers

The control unit of a wind turbine has several tasks to perform, such as switching the system on or off according to wind conditions, safety functions or performance optimization. Regarding the first task mentioned, depending on the size and design of the turbine, there is a wind speed range outside which the system is turned off. If the wind speed is below the lower limit (cut-in speed), the system is stopped because the available power is too low and energy transformation becomes unprofitable. In the opposite case, when the wind speed is over the so-called cut-out speed, which would result in excessive loads, as an example, in most cases the blades are rotated to reduce the loads on the blades (basically, a very low angle of attack and thus low lift and drag forces are achieved. This position is called feathering). With regard to performance optimization, which is valid when the wind speed is in the useful range, various actuations can be used, such as varying the blade pitch angle (either individually or collectively), the torque of the electric generator or the yaw angle.
Once the turbine and the generator have been designed, there is a rated power, beyond which it is not advisable to overtake because of generator overheating problems, as well as a maximum angular speed beyond which it would be dangerous to go because too high loads could occur. This rated power production, under design condition, is reached when wind blows at so called rated speed. When wind speed is smaller than it, principal goal of control unit is to maximize the power capture, while wind speed becomes grater, according with above, main task is to maintain power production and rotor angular speed at their rated values (more information about control strategies can be found in the next section). Moreover, other control objectives may exist, for instance offshore floating wind turbine should keep structure oscillations as low as possible.

Most of installed wind turbine use classical Proportional-Integral-Derivative (PID) controller with constant gains or gain scheduled PID, which are controllers in which gains are made to vary according to system conditions. This type of control is useful when dealing with highly non-linear systems, as it is possible to adopt gain values adapted to the characteristics of the system when it is under certain conditions to always obtain an optimal control law. Despite their simplicity and robustness, these kinds of model have their limits, i.e., they are not suitable for MIMO systems, so they don’t work well with systems with many actuations.

In addition to classical controllers, it is possible to rely on many innovative control strategies, such as algorithms based on optimization or prediction techniques. These are, for example, Linear Quadratic Gaussian (LQG) methods, which rely on knowledge of the system model to calculate an optimal control action once the system states have been estimated. Another example, similar to previous ones, are MPC controllers, which rely on knowledge of the model and the estimation of external inputs in order to find the optimal control trajectory within a certain time horizon.

Other methods are becoming more widely available in wind turbine controls such as Neural Network (NN) and Fuzzy Logic (FL), or hybrid methods based on them, called Adaptive Neuro-Fuzzy Inference Systems (ANFIS).

4.2 WT Design: Basic Concepts

As already mentioned, maximizing the extracted power is the main objective of a wind turbine, as long as it remains within the prescribed limits with regard to, for example, loads on the structure or generator maximum permitted power. During turbine design, and in particular blades one, many decisions are made to achieve these objectives; quantities such as number of blades, their geometry or the twist angle are usually chosen at the design stage in order to obtain a good compromise between the various desired characteristics. To make some considerations on the subject, as well as to deal with the control of wind turbines, it is important to recall the expressions of the thrust and power coefficients and tip speed ratio TSR, introduced in section 3.1.13, where the aerodynamic forces on the blades are discussed.

$$- \; C_T = \frac{T}{\frac{1}{2}\rho \, A_D {U_\infty}^2} \qquad - \; C_P = \frac{P}{\frac{1}{2}\rho \, A_D {U_\infty}^3} \qquad - \; \lambda = TSR = \frac{\Omega R}{U_\infty} \qquad (4.1)$$

C_T is thrust force T over maximum ideal one $\frac{1}{2}\rho \, A_D {U_\infty}^2$ (thrust that would occur if the downstream wind speed were zero, i.e. all momentum was transferred to the rotor), C_P is the output power P over the available wind power $\frac{1}{2}\rho \, A_D {U_\infty}^3$. U_∞ is the undisturbed wind speed, A_D is the rotor area and ρ is the air density.

From momentum theory [10], power coefficient ideal limit, called Betz limit, is $C_{P,id} \cong 0.59$. Introducing rotating wake effect (but still assuming other simplifications) it can be seen that C_P tends to this limit for high values of TSR. Including other effect, some of them introduced in section 3.1.13, C_P values decrease, as can be seen in Figure 4.1. It shows ideal C_P and power coefficient values, with respect to design tip speed ratio, that can be reached in real turbines (λ_d is the tip speed ratio at optimal working condition).

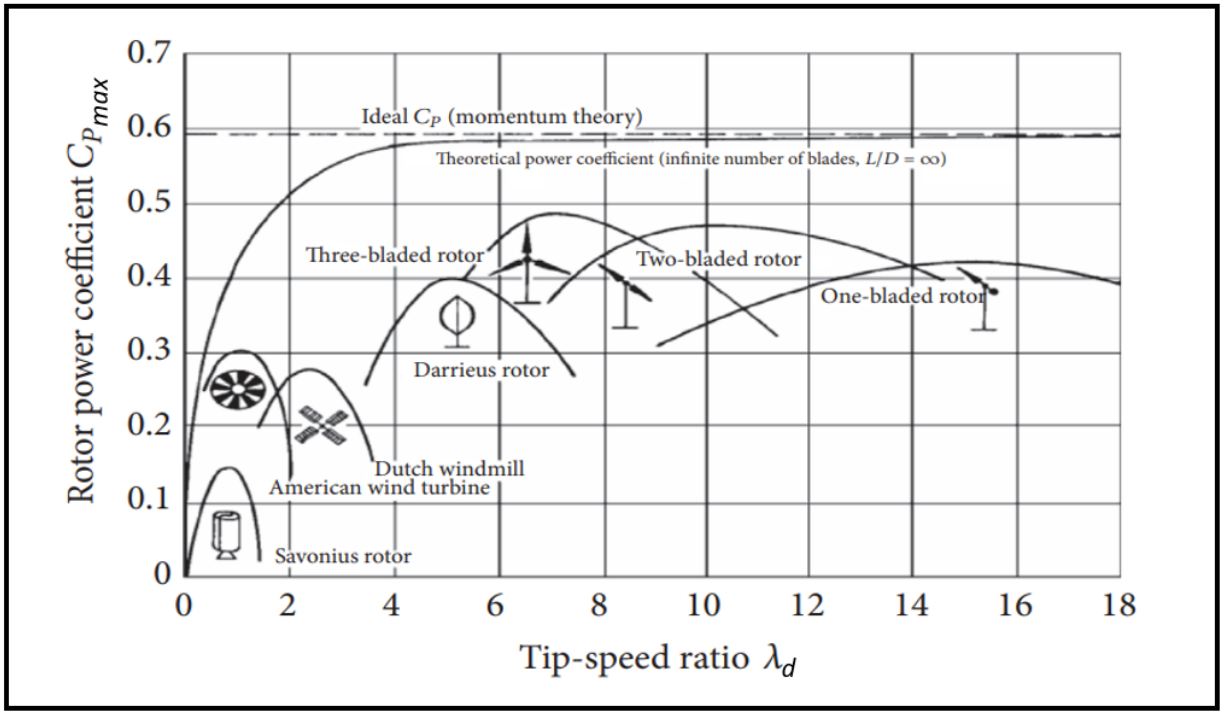

Figure 4.1: Maximum C_P coefficient with respect to design tip speed ratio λ_d for horizontal axis wind turbine with different number of blades and for some other types of wind turbines.

Most of HAWTs (horizontal axis wind turbines) have 3 blades. As can be seen in figure 4.1, they present higher maximum power coefficient than turbine with less blades, moreover, TSR range with C_P higher than a certain percentage of its maximum becomes smaller with higher number of blades. Another important factor is cost, for which there are often no more than three blades, despite some performance benefits. Regarding design tip speed ratio, an optimal value is chosen to maximize power as well as considering other factors, such as maximum tangential velocity. Very high tip speed can lead to erosion due to impact on dust or sand in the air, or to excessively high centrifugal loads. Furthermore, high speeds lead to high noise and vibrations, so typical range of design TSR for a three blades wind turbine is $\lambda_d = 6 - 8$.

Once optimal TSR (design TSR) and blades' number is chosen, from BEM theory (section 3.1.1.3), optimal blade features along its length (usually chord length and twist angle) can be computed to maximize performances when design TSR holds. Shortly, the aim is to have a constant angle of attack, which one leads to maximum value of C_L/C_D (with C_L and C_D lift and drag coefficient, see 3.70 and 3.71), but actual blade shape is also influenced by structural and design purposes, such as connection with hub.

Once turbine design is defined, actual power and thrust coefficients can be computed, through BEM theory. They are function of tip speed ratio (λ) and blade pitch angle (θ_{bl}, we assume the same angle for each blade: collective pitch). Maximum value of power coefficient, which is the optimal/design value, is reached at design condition, that is design tip speed ratio and null blade pitch angle.

An example of thrust and power coefficient, with respect to TSR and blade pitch angle, are shown in Figure 4.2; they refer to the wind turbine case of study presented in section 2.

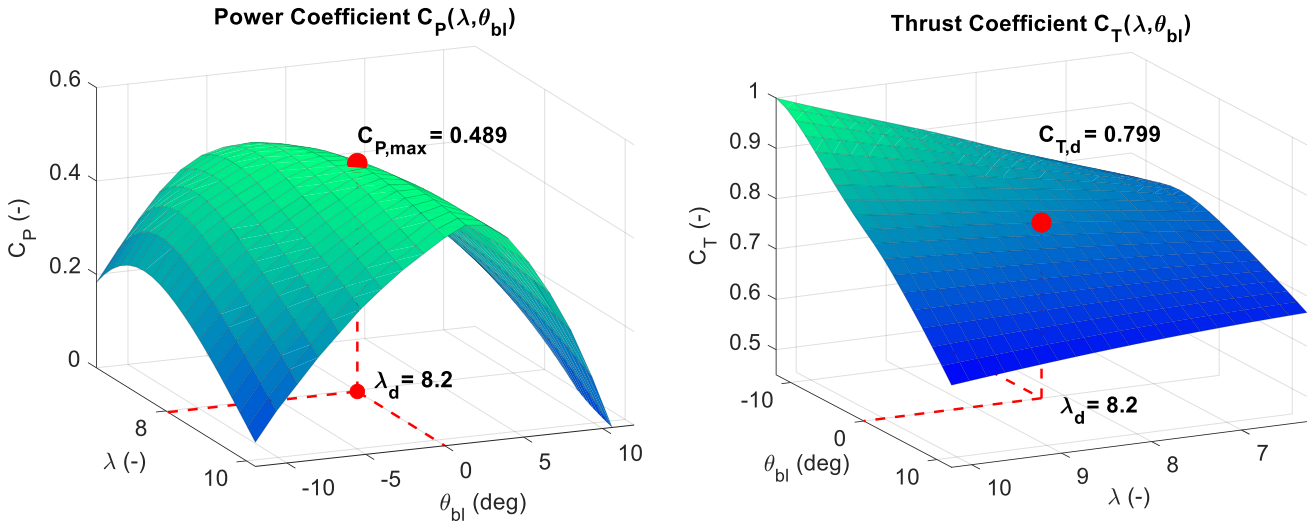

Figure 4.2: Example of real Cp and C_T coefficient at different tip speed ratio (TSR) and collective blade pitch angle (θ_{bl}).

4.3 Wind Turbine Control Strategies

In this section some of the main control strategies used in wind turbine controls are introduced, all of them usually consist of yaw control module too, which will not be treated here, as well as emergency, cut-in/cut-off, or rotor braking ones. So, following information regard performance control, that is power optimization below rated conditions and rated power tracking above it; main strategies are:

- **Fixed-speed fixed-pitch**: it is the simplest configuration, where it is impossible to improve performance during operations with active controls. In these turbines the generator is directly coupled with power grid since rotor speed is kept constant. Due to that, if wind speed varies, so does tip speed ratio and power coefficient. This means maximum power coefficient can be reached only at design wind speed. Regarding above rated wind speed condition, the so-called "stall passive control" occurs, through which power output is limited: with a wind speed increase, since rotor speed is constant, angle of attack increase too until stall is reached, with a decrease of lift and torque and an increase of thrust. This type of control is rarely used, particularly for big size wind turbines and not at all for offshore floating ones.

- **Variable-speed fixed-pitch**: this configuration consists of a variable speed rotor/generator, so a frequency converter is needed to connect generator with power grid (it means more costs). Variable speed permits to keep the tip speed ratio constant (at its design value) to maximize power extraction in below rated wind conditions. Figure 4.3 shows power with respect to rotor angular speed, Ω, at different wind speed U_{wind}. Colored points are steady state conditions that can be reached thanks to rotor speed variation, while blue curve is optimal curve, representing convolution of these optimal working points. Then, to maximize power under nominal conditions, the generator torque is adjusted to reach the optimum speed. Above nominal conditions, on the other hand, the rotor speed is kept constant (thus having a similar behavior to the previous case) since, to have constant power equal to the nominal value, unacceptable rotor speeds would have to be reached.

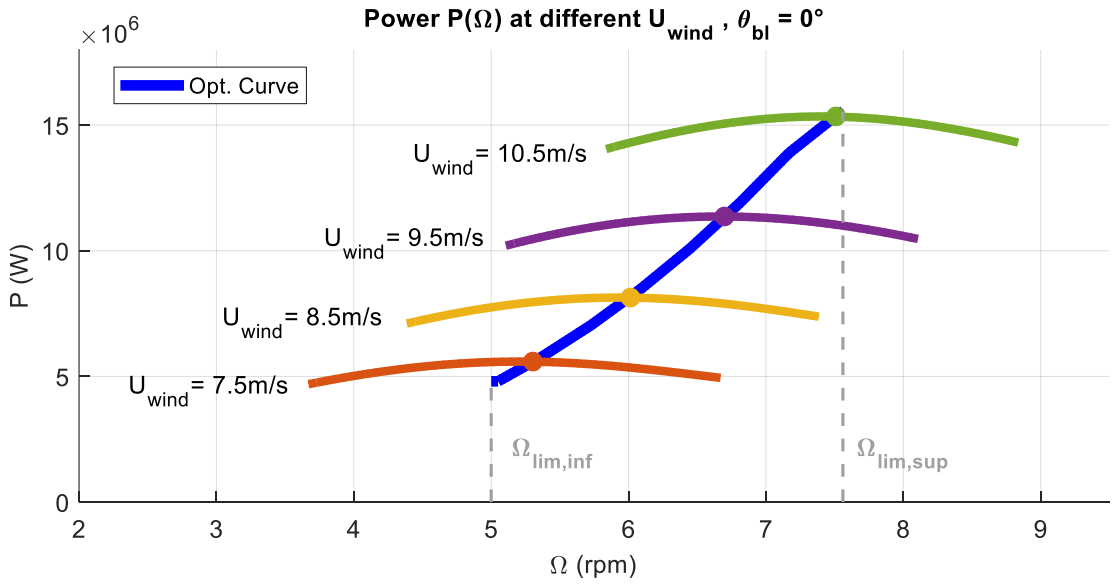

Figure 4.3: Power production with respect to rotor angular speed Ω at different values of wind speed U_{wind}. Optimal curve represent target working condition below rated wind speed for variable speed wind turbines. $\Omega_{lim,inf}$ and $\Omega_{lim,sup}$ are cut-in and rated rotor angular speed.

- **Variable-speed variable-collective pitch:** in this case control actions are generator torque and collective blade pitch angle (equal angle for each blade). Control strategies at below wind speed conditions are the same of previous category, since blade are kept at its rest position (design working point at null blade pitch). Above rated condition blade pitch is adjusted to track power production at its rated value. It ca be moved, to reduce power coefficient if wind speed increases, in both direction, increasing (stall) or decreasing (feather) angle of attack. In the first case sensitiveness is higher, so smaller angles are needed, but drag increases. On the other hand, in the second case, drag decreases with lift, but sensitiveness is smaller.

  In most cases, especially for traditional controller, generator torque is varied to track desired speed at below rated condition keeping constant the blade pitch angle (=0°), while blade pitch angle is controlled at above rated condition, keeping rotor speed constant at its rated value. With these actions it is theoretically possible to reach optimal trajectory, that is obtaining power maximization at below rated conditions and tracking rated one at above rated conditions (Opt. curve in Figure 4.4).

  Figure 4.4 shows two plots with output power with respect to wind speed; in particular, first one shows three curves at different values of rotor speed (all with null blade pitch). In can be seen as wind speed increases, rotor speed for which power maximization occurs increases too. Red curve represents optimal one, obtained by reaching optimal rotor speed at each value of wind velocity. Second plot shows some curves at different values of blade pitch (all with rated rotor speed), through which it can be seen as wind speed increases (above rated one) blade pitch can be regulated to keep output power at its rated value.

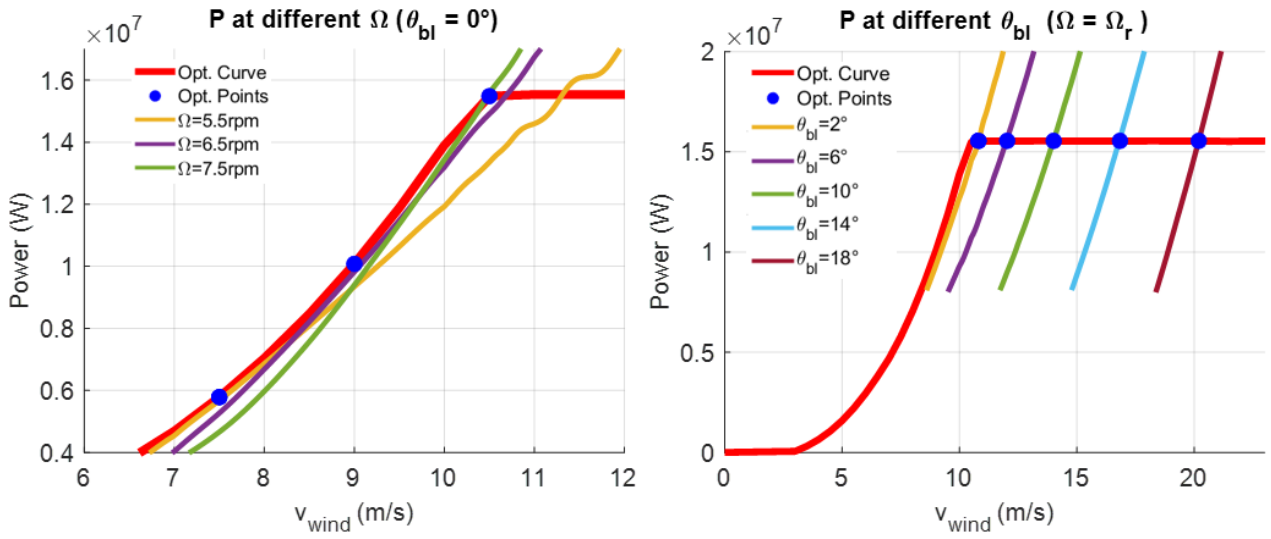

Figure 4.4: Output power at different values of rotor speed (left) and at different values of blade pitch (right). These graphs show control strategies adopted in variable-speed variable-pitch wind turbines to reach, at any wind speed condition, optimal power production, that is maximum one below rated wind and target one above rated wind speed (red curve).

- **Variable-speed variable-individual pitch:** this is an extension of the previous category. Despite optimal working condition can be theoretically reached also with collective blade pitch, the possibility of adjusting the various blades independently offers advantages in terms of reducing the loads on the blades and concerning the stability and oscillations of floating offshore wind turbines. Given the large number of controllable inputs, traditional controllers, often a combination of several SISO controllers, do not work well, which is why this strategy is more likely to be accompanied by more advanced controllers, which are designed to handle MIMO systems, such as model predictive controllers (MPC) or LQG ones.

Figure below shows an example of three curves that represent normalized power with respect to wind speed of different control strategies. Particularly, control strategies showed are fixed-speed fixed pitch (FS-FP), variable-speed fixed pitch (VS-FP) and variable-speed variable pitch (collective pitch) (VS-VP). In addition to being able to observe what has been explained above, it can be seen in cases with a fixed pitch, it is preferred to have a slightly lower power in the rated condition (in this case this is at wind speeds of around 15 m/s) in order not to exceed the maximum power when the wind speed is higher than the rated one.

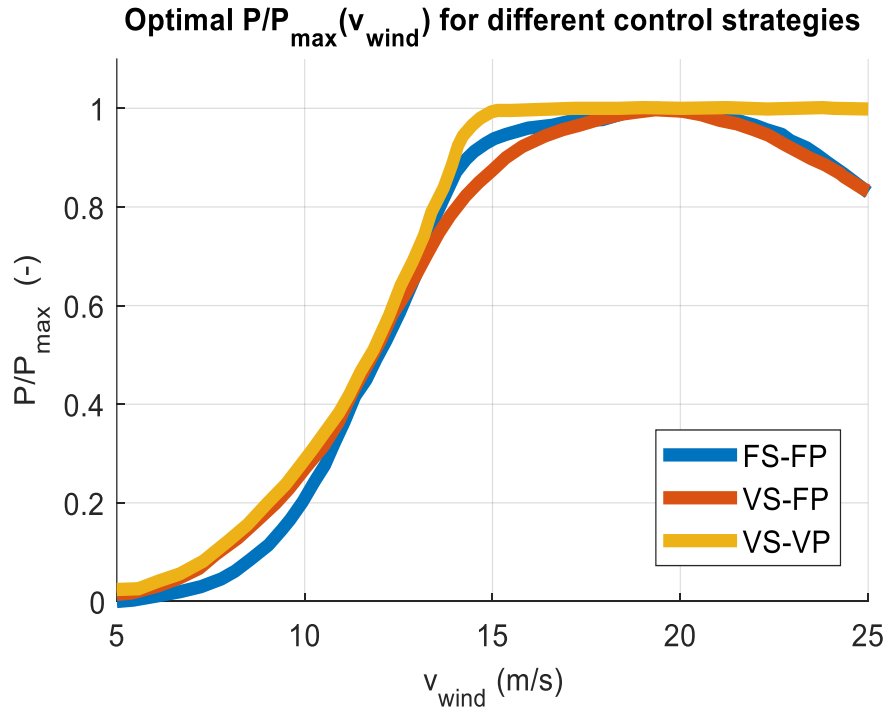

Figure 4.5: Relation between wind speed and output power for different control strategies: fixed-speed fixed pitch (FS-FP), variable-speed fixed pitch (VS-FP) and variable-speed variable pitch (collective pitch) (VS-VP).

4.4 Examples of Conventional Controllers

In this section two different conventional controllers are presented; conventional controllers are those based on PID logic, possibly with some additional modules to improve performances. Specifically, the chosen ones are the Baseline controller, a well-known controller widely used in wind turbines, and the ROSCO controller (Reference Open-Source COntroller for fixed and floating offshore wind turbines), developed by researchers at the Delft University of Technology and presented in Mulders and van Wingerden (2018).

4.4.1 Baseline Controller

Baseline is a conventional variable-speed variable collective pitch controller [26], [27], which is made up of two independent systems:

- A generator torque controller designed to maximize power extraction below nominal wind speed;
- A blades collective pitch controller designed to regulate rotor and generator speed above nominal wind speed.

This work does not include any further control for start-up, shutdown, safety and protection sequences or any nacelle-yaw regulation. Table 4.1 and Figure 4.6 show principal case of study wind turbine characteristics (IEA 15 MW, see Section 2 for more information) and a possible baseline control logic.

Parameter	Value	Unit
Cut-in wind speed	3	m/s
Rated wind speed	10.5	m/s
Cut-out wind speed	25	m/s
Rated rotor speed	7.56	rpm
Drivetrain	Direct drive	-
Rated torque	19.6	MNm
Rated mechanical power	15.5	MW

Table 4.1: Main features of IEA 15 MW wind turbine.

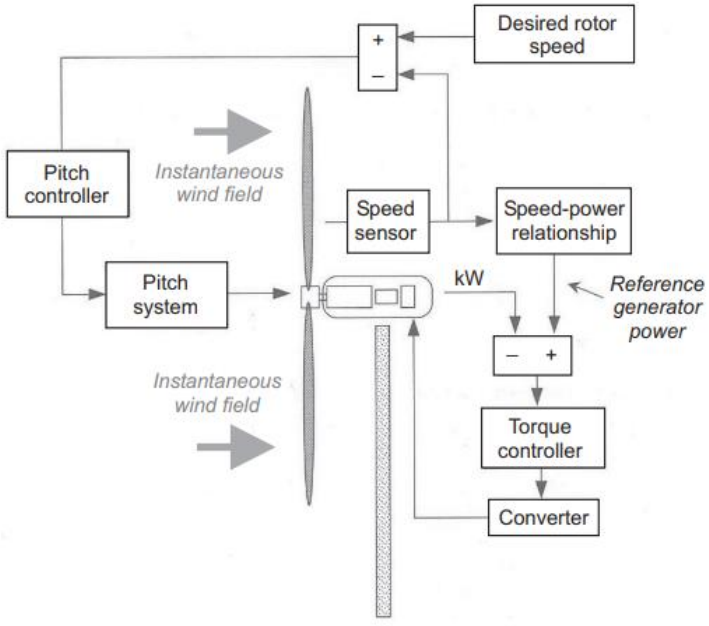

Figure 4.6: Baseline control scheme.

4.4.1.1 Generator-Torque Controller

The generator-torque control law is designed to have three main regions and two transition ones between them. Aerodynamic torque acts as an accelerating load, the generator torque, converting mechanical energy to electrical energy, acts as a braking load. The generator torque is computed as a tabulated function of the filtered generator speed, incorporating 5 control regions: 1, 1.5, 2, 2.5 and 3.

- **Region 1**: control region before cut-in wind speed, where the generator is detached from the rotor to allow the wind to accelerate the rotor for start-up. In this region, the generator torque is zero and no power is extracted from the wind;

- **Region 2**: control region where extracted power is maximized. Here, to maintain the tip speed ratio constant at its optimal value, the generator torque is proportional to the square of the filtered generator speed. From definition of power coefficient (equation 4.1) aerodynamic torque can be expressed as:

$$C_{aero} = \frac{1}{2}\,\rho\,\pi\frac{R^5}{\lambda^3}C_P\,(\lambda,\theta_{bl})\cdot\,\Omega^2\;=\;k_{opt}\cdot\Omega^2 \qquad (4.2)$$

  Where k_{opt} is obtained with TSR and blade pitch values that lead to maximum power coefficient ($\lambda = \lambda_{opt}\ ,\ \theta_{bl} = 0°$);

- **Region 3**: above rated condition region, where the generator torque is kept constant at its rated value. In this region pitch control is active to maintain rotor speed at its rated value.

Regions 1.5 and 2.5 are transition region, the first one is called start-up region and permits a smooth transition between null torque and optimal one. Figure 4.7 shows control law of generator torque controller.

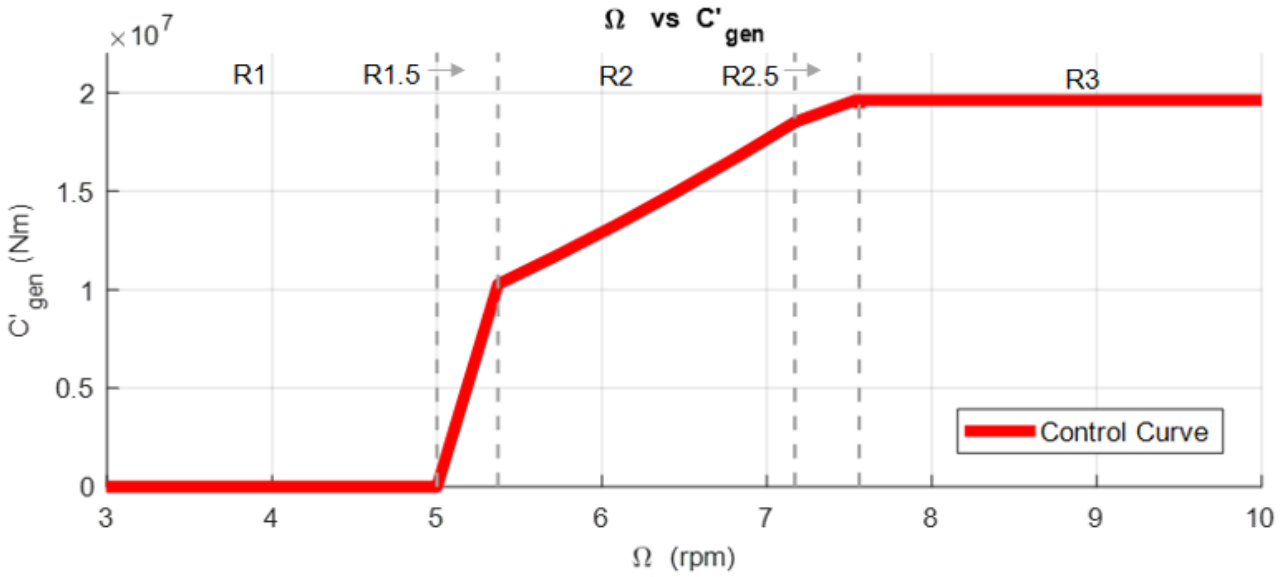

Figure 4.7: Control law of generator torque controller. Ω and C'_{gen} are, respectively, rotor angular speed and generator torque. Apex indicates torque on rotor shaft is considered ($C'_{gen} = N\cdot C_{gen}$, see equation 3.110, but in this case, since a direct drivetrain is considered, N=1).

4.4.1.2 Blade-Pitch Controller

As anticipated, the rotor-collective blade-pitch-angle regulates the generator speed in region 3 (where wind speed exceeds its rated value) to maintain it at its nominal value through a scheduled proportional-integral control (PI). The output of this PI-controller is used as reference pitch signal θ_{ref} to the pitch system. The reference pitch angle θ_{ref} is compared to the actual pitch angle θ and then the error $\Delta\theta$ is corrected by the servomechanism. Scheduling is implemented to compensate the non-linear aerodynamic characteristics.

Rearranging equation 3.10, the equilibrium equation of the rotor around its axis of rotation obtained in section 3.2.1, we can write:

$$C_{\text{aero}} - C'_{gen} = I_{\text{eq}}\,\dot{\Omega} \quad (4.3)$$

With $C'_{gen} = N \cdot C_{gen}$ and $N = \Omega_{gen}/\Omega_{rot}$. The case study we refer to considers a direct drivetrain, so N=1. Due to this, we will omit apex.

In region 3 generator torque is kept constant at its rated value, defined as $C_{gen,r} = P_r / \Omega_r$. Aerodynamic torque C_{aero} depends on wind speed, rotor speed and blade pitch (section 3.1.1.3), but assuming in this region rotor speed maintains its rated value Ω_r (this assumption can be made since the control objective is to track that value) and assuming power to wind speed sensitivity can be neglected, following expression holds:

$$C_{\text{aero}} \approx C_{\text{aero}}\left(U_{wind,0}, \Omega_r, \theta_{bl,0}\right) + \left.\frac{dC_{\text{aero}}\,(U_{wind},\ \Omega,\ \theta_{bl})}{d\theta_{bl}}\right|_{\substack{\theta_{bl}=\theta_{bl,0} \\ U_{wind}=U_{wind,0} \\ \Omega=\Omega_r}} \left(\theta_{bl} - \theta_{bl,0}\right) =$$

$$= \frac{P\left(U_{wind,0},\ \Omega_r,\ \theta_{bl,0}\right)}{\Omega_r} + \frac{1}{\Omega_r}\left.\frac{dP(U_{wind},\ \Omega,\ \theta_{bl})}{d\theta_{bl}}\right|_{\substack{\theta_{bl}=\theta_{bl,0} \\ U_{wind}=U_{wind,0} \\ \Omega=\Omega_r}} \left(\theta_{bl} - \theta_{bl,0}\right) \quad (4.4)$$

$U_{wind,0}$ and $\theta_{bl,0}$ are wind speed and blade pitch at which linearization is made. Once first is chosen, $\theta_{bl,0}$ is which one leads to a steady state condition with extracted power equal to the rated one. So, aerodynamic torque expression becomes:

$$C_{\text{aero}} \approx \frac{P_r}{\Omega_r} + \frac{1}{\Omega_r}\frac{dP}{d\theta_{bl}}\Delta\theta_{bl} \quad (4.5)$$

Where $\Delta\theta_{bl}$ represents a small perturbation of the blade pitch angle about its linearization point $\theta_{bl,0}$ (zero in this case).

By expressing the blade-pitch regulation starting from the speed perturbation with a proportional-integrative control law (PI), it is possible to write:

$$\Delta\theta_{bl} = K_P\,\Delta\Omega + K_I \int_0^t \Delta\Omega\, dt \qquad (4.6)$$

Where K_P is the proportional gain and K_I the integrative gain; $\Delta\Omega$ represents a small perturbation of rotor speed about its rated value: $\Delta\Omega = (\Omega - \Omega_r)$.

By inserting equations (4.5), (4.6) and $C_{gen,r} = P_r / \Omega_r$ into the equation (4.3), it is possible to obtain, once defined $\Delta\Omega = \dot{\delta}$, the following relation:

$$\frac{P_r}{\Omega_r} + \frac{1}{\Omega_r}\frac{dP}{d\theta_{bl}}\left(K_P\dot{\delta} + K_I\delta\right) - \frac{P_r}{\Omega_r} = I_{eq}\ddot{\delta} \qquad (4.7)$$

Which can be rearranged as:

$$I_{eq}\ddot{\delta} + \left[-\frac{dP}{d\theta_{bl}}\frac{K_P}{\Omega_r}\right]\dot{\delta} + \left[-\frac{dP}{d\theta_{bl}}\frac{K_I}{\Omega_r}\right]\delta = 0 \qquad (4.8)$$

That in the canonical form becomes:

$$M\ddot{\delta} + C\dot{\delta} + K\delta = 0 \qquad (4.9)$$

With: $M = I_{eq}$, $C = \left[-\frac{dP}{d\theta_{bl}}\frac{K_P}{\Omega_r}\right]$, $K = \left[-\frac{dP}{d\theta_{bl}}\frac{K_I}{\Omega_r}\right]$

Now it is possible to choose proportional and integral gains in order to obtain desired characteristics of second order system 4.9. Its characteristics directly depend on natural frequency and damping ratio:

$$\omega_n = \sqrt{\frac{M}{K}}\ , \quad \zeta = \frac{C}{2M\omega_n} \qquad (4.10)$$

Once defined ω_n and ζ, expressions of proportional and integral gains become:

$$K_P = \frac{2\,I_{eq}\,\omega_n\,\zeta\,\Omega_r}{-\frac{dP}{d\theta_{bl}}} \qquad (4.11)$$

$$K_I = \frac{I_{eq}\,\omega_n^2\,\Omega_r}{-\frac{dP}{d\theta_{bl}}} \qquad (4.12)$$

The term $\frac{dP}{d\theta_{bl}}$ is power to pitch sensitivity, which, as can be seen in equation 4.4, depends on wind speed and blade pitch (related each other as previously mentioned) adopted during linearization. So, to always have the same system characteristic (ω_n and ζ), proportional and integral gains must vary with a variation of blade pitch and so of wind speed.

Figure 4.8 shows power to pitch sensitivity with respect to blade pitch; as can be seen there, it can be well approximated with a quadratic regression, through which quadratic form that minimize sum of square error is computed. Thanks to this regression, power to pitch sensitivity expression becomes of the form:

$$\frac{dP}{d\theta_{bl}} \approx c_1 {\theta_{bl}}^2 + c_2 \theta_{bl} + c_3 \quad (4.13)$$

$\frac{dP}{d\theta_{bl}}$ is the power to pitch sensitivity and c1 (W/deg^3), c2 (W/deg^2) and c3 (W/deg) are the coefficients of its quadratic regression.

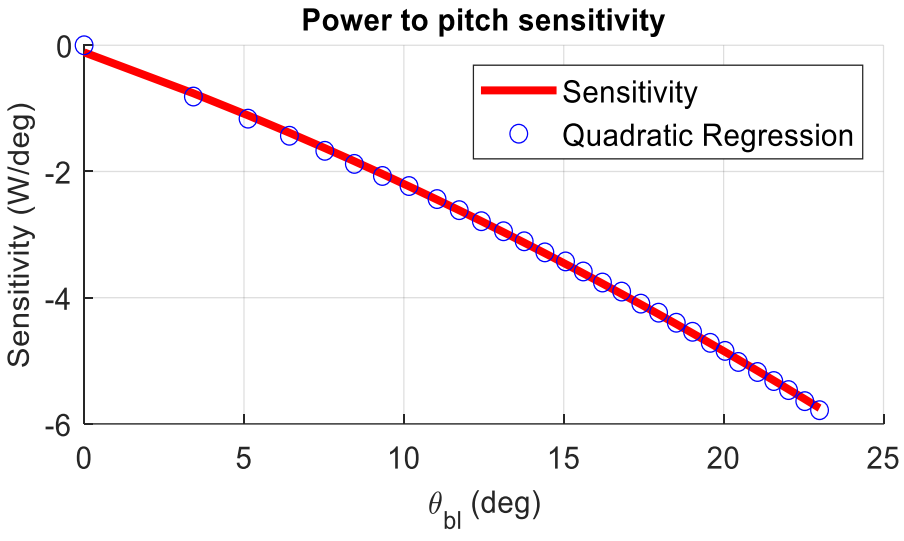

Figure 4.8: Power to pitch sensitivity $\frac{dP}{d\theta_{bl}}$ with respect to blade pitch θ_{bl}. Blue points are true values at several θ_{bl} (and U_{wind}), while red curve is their quadratic regression.

4.4.2 ROSCO Controller

In this section main features of ROSCO controller are introduced; ROSCO controller (Reference Open-Source COntroller for fixed and floating offshore wind turbines) was developed by researchers at the Delft University of Technology [28] to provide a modular reference wind turbines controller that represent industry standards and performs comparably or better than existing reference controllers, such as baseline, discussed in previous section. Reference controllers commonly cited in literature have been developed to work with specific turbines, so they are difficult to modify for use on other turbines, while this controller has been developed to provide a reference from which also non-control engineers can deal with, starting from few information (substantially main geometric characteristics and power coefficient curve). The primary functions of the controller are still to maximize power in below-rated operations and to regulate rotor speed in above-rated ones, moreover, it also provides additional modules which can improve control performances. Figure 4.9 shows a block diagram of ROSCO logic where flow of input and output signals through its various modules can be seen.

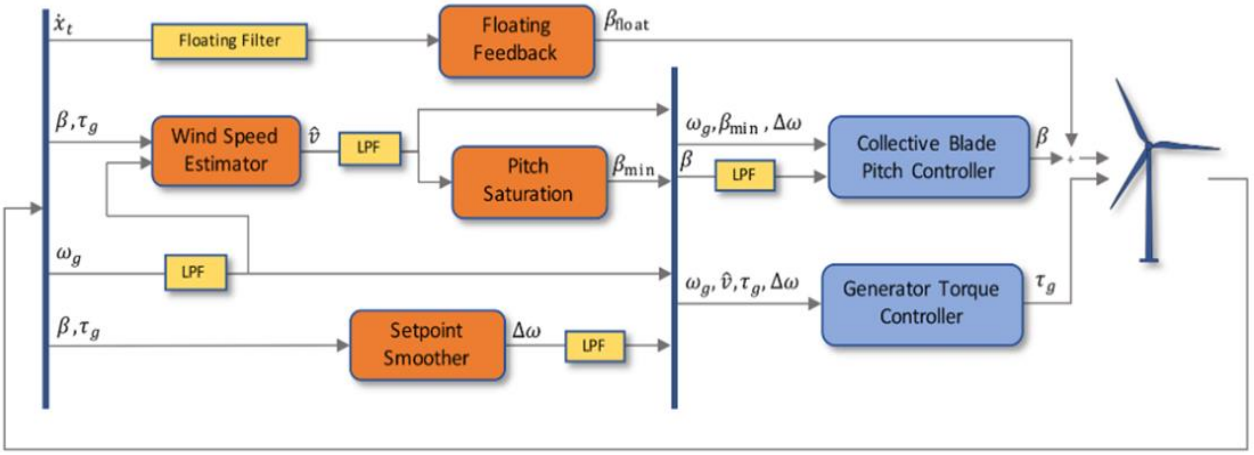

Figure 4.9: Block diagram showing ROSCO logic: blue squares represent generator torque and blade pitch controllers, while orange ones represent various additional (and optional) modules. Yellow squares are filters (LPF means Low-Pass filter), whose primary function is smoothing input signals. Input and output signals of different modules are: $\dot{x}_t$ (tower-top fore–aft velocity), β (collective blade pitch angle), ω_g (generator speed), τ_g (generator torque), β_{float} (floating controller's contribution to the blade pitch angle), β_{min} (minimum blade pitch angle), $\hat{v}$ (estimated wind speed) and $\triangle\omega$ (controller reference set point shifting term from the set point smoother).

4.4.2.1 Control Regions

ROSCO controller, as well as Baseline and most of other conventional ones, consists of two methods of actuation: generator torque and collective blade pitch. Strategies of actuation are commonly separated into four main regions, with transition logic between them. Regions 1 and 4 correspond to below cut-in and above cut-out wind speed conditions, these regions are generally out of interest for standard control purposes (performances optimization) and so they will not be further discussed below. In region 1 generator torque is set to zero to allow the wind to accelerate the rotor for start-up. In this region, no power is extracted. In region 4 blades are pitched to reduce thrust force to zero (feathering position).

Control strategies for regions 1.5, 2, 2.5 and 3 are highly like those ones adopted in Baseline control. Region 2 is when wind speed is below rated condition, here main goal is power extraction maximization. To do so, two methods can be used, a quadratic low (as in Baseline controller) of generator torque with respect to rotor angular speed or a tip speed ratio (TSR) tracking to maintain the latter at its optimal value (in this case a wind speed estimator is needed). Region 3 is when wind speed is above rated condition, blade pitch is regulated to maintain rotor speed at its rated value and to stabilize platform (for offshore floating wind turbines, through floating feedback module), while generator torque is kept constant at its rated value. Region 1.5 is a transition region from cut-in wind speed and region 2. Here generator torque is regulated to maintain a defined minimum rotor speed and blades are pitched to compensate resulting high values of TSR to improve power extraction. Region 2.5 is simply a linear transition between regions 2 and 3, reason why it is not depicted in Figure 4.10, which represents the values of inputs and some important outputs with respect to wind speed (steady state values) that summarize the strategies just explained.

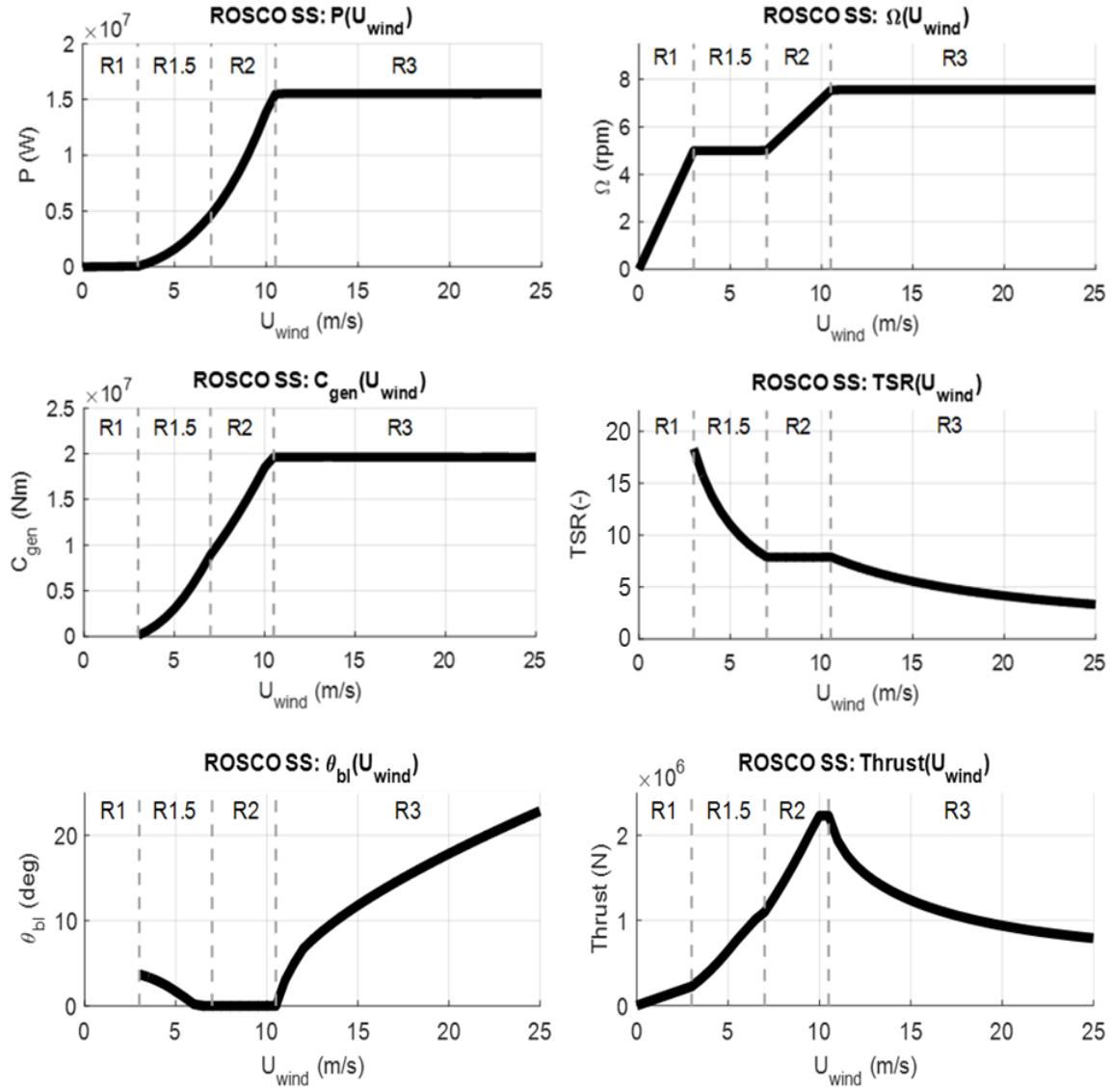

Figure 4.10: Steady state values of main input/output ROSCO controller signals with respect wind speed. Showed quantities are: P (extracted power), Ω (rotor speed), C_{gen} (generator torque), TSR (tip speed ratio), θ_{bl} (blade pitch angle) and thrust force.

4.4.2.2 ROSCO Implementation

Controller implementation starts from aerodynamic torque (C_{aero}) expression and rotor equilibrium equation, as in Baseline controller. Rearranging 4.2 and 4.3 following equations can be written:

$$C_{aero} = \frac{1}{2}\,\rho\, A_D\, C_P\,(\lambda, \theta_{bl})\,\frac{{U_\infty}^3}{\Omega} \qquad (4.14)$$

$$\dot{\Omega} = \frac{C_{\text{aero}} - C'_{gen}}{I_{\text{eq}}} \qquad (4.15)$$

With $C'_{gen} = N \cdot C_{gen}$ and $N = \Omega_{gen}/\Omega_{rot}$. The case study we refer to considers a direct drivetrain, so N=1. Due to this, we will omit apex. $\dot{\Omega}$ is the rotor angular acceleration, I_{eq} is the rotor inertia, ρ is the air density, A_D is the rotor area, Cp is the power coefficient (defined in eq. 4.1) and U_∞ is the wind undisturbed wind speed.

The first-order linearization of eq 4.14 at some nominal steady-state operational point is:

$$\Delta C_{aero} = \Gamma_\Omega\big|_{op}\,\Delta\Omega + \Gamma_{\theta_{bl}}\big|_{op}\,\Delta\theta_{bl} + \Gamma_U\big|_{op}\,\Delta U \qquad (4.16)$$

With: $\Gamma_\Omega\big|_{op} = \partial C_{aero}/\,\partial\Omega\,\big|_{op}, \quad \Gamma_{\theta_{bl}}\big|_{op} = \partial C_{aero}/\,\partial\theta_{bl}\,\big|_{op}, \quad \Gamma_U\big|_{op} = \partial C_{aero}/\,\partial U\,\big|_{op}$ (4.17)

"op" denotes the steady-state operational point at which linearization is made. Equation 4.15 can then be rewritten as (Δ denotes the perturbation from steady state value "op" and $\boldsymbol{X}_{\text{op}} = \left\{\lambda_{\text{op}}, \theta_{bl\,\text{op}}\right\}$):

$$\Delta\dot{\Omega} = A\big(\boldsymbol{X}_{\text{op}}\big)\Delta\Omega + B_{C_{gen}}\Delta C_{gen} + B_{\theta_{bl}}\big(\boldsymbol{X}_{\text{op}}\big)\Delta\theta_{bl} + B_U\big(\boldsymbol{X}_{\text{op}}\big)\Delta U \qquad (4.18)$$

With:

$$A\big(\boldsymbol{X}_{\text{op}}\big) = \frac{1}{I_{\text{eq}}}\frac{\partial C_{aero}}{\partial\lambda}\frac{\partial\lambda}{\partial\Omega} \qquad (4.19)$$

$$\frac{\partial C_{aero}}{\partial\lambda} = \frac{1}{2}\rho A_{\text{D}} R {U_{\text{op}}}^2 \frac{1}{\lambda_{\text{op}}^2}\left(\frac{\partial C_{\text{p}}}{\partial\lambda}\lambda_{\text{op}} - C_{\text{p,op}}\right) \qquad (4.20)$$

$$\frac{\partial\lambda}{\partial\Omega} = \frac{R}{U_{\text{op}}}, \;\; (\lambda = \frac{\Omega R}{U}) \qquad (4.21)$$

$$B_{C_{gen}} = -\frac{1}{I_{\text{eq}}} \qquad (4.22)$$

$$B_{\theta_{bl}}\big(\boldsymbol{X}_{\text{op}}\big) = \frac{1}{2\,I_{\text{eq}}}\rho A_{\text{D}} R {U_{\text{op}}}^2 \frac{1}{\lambda_{\text{op}}^2}\left(\frac{\partial C_{\text{p}}}{\partial\theta_{bl}}\,\lambda_{\text{op}}\right) \qquad (4.23)$$

All derivatives are calculated at "op" values; ΔU, difference between actual wind speed and wind speed at linearization point, is considered equal to zero during control tuning, that is computation of control gains.

Both generator torque and blade pitch controllers are PI controllers, generically defined as:

$$y = K_P u + K_I \int_0^T u \, \mathrm{d}t \qquad (4.24)$$

Where u represents the input and y the output, while K_P and K_I are respectively the proportional and integral gains. Generator torque controller has input and output:

$$u = -\delta\Omega\,, \;\; y = \Delta C_{gen} \qquad (4.25)$$

Blade pitch controller has input and output:

$$u = -\delta\Omega, \;\; y = \Delta\theta_{bl} \qquad (4.26)$$

$\delta\Omega$ is defined as a perturbation from the reference speed:

$$\Omega(t) = \Omega_{\text{ref}} + \delta\Omega \longrightarrow -\delta\Omega = \Omega_{\text{ref}} - \Omega(t) \qquad (4.27)$$

While ΔC_{gen} and $\Delta\theta_{bl}$ are perturbations from steady state values:

$$\theta_{bl}(t) = \theta_{bl_{\text{op}}} + \Delta\theta_{bl}, \quad C_{gen}(t) = C_{gen_{\text{op}}} + \Delta C_{gen} \qquad (4.28)$$

Now, defining $\Delta\Omega_{\text{ref}} = \Omega_{\text{ref}} - \Omega_{op}$ (assumed =0, since "op" point is chosen at a steady state condition with $\Omega_{op} = \Omega_{\text{ref}}$), we can combine equation 4.18 with above definitions to obtain a differential equation that relates $\Delta\Omega = \Omega - \Omega_{op}$ and $\Delta\Omega_{\text{ref}}$. Then, it we take the Laplace transform of this equation we arrive to two closed-loop transfer functions (one for the generator torque module and the other for the blade pitch one) in the form:

$$H(s) = \frac{\Delta\Omega(s)}{\Delta\Omega_{\text{ref}}(s)} = \frac{B\left(K_P(\boldsymbol{X}_{\text{op}})s + K_I(\boldsymbol{X}_{\text{op}})\right)}{s^2 + \left(B\,K_P(\boldsymbol{X}_{\text{op}}) - A(\boldsymbol{X}_{\text{op}})\right)s + BK_I(\boldsymbol{X}_{\text{op}})} \qquad (4.29)$$

Where B is $B_{C_{gen}}$ or $B_{\theta_{bl}}$,depending on which module is considered, since when generator torque loop is considered, $\Delta\theta_{bl}$ is set to zero and, when blade pitch loop is considered, ΔC_{gen} can be equal to zero or $B_{C_{gen}}$ can be englobed in A. Moreover, in both cases we consider $\Delta U = 0$.

$H(s)$ is a simple second order system whose characteristics are strictly related to natural frequency and damping ratio of its canonical form. They can be defined, in order to reach desired performance, choosing values of proportional and integral gains. If we call ω_n the natural frequency and ζ the damping ratio, K_P and K_I expressions (varying with operational steady state point) are:

$$K_P = \frac{1}{B(\boldsymbol{X}_{\text{op}})}\left(2\zeta\omega_n + A(\boldsymbol{X}_{\text{op}})\right) \qquad (4.30)$$

$$K_I = \frac{\omega_{\text{n}}^2}{B(\boldsymbol{X}_{\text{op}})} \qquad (4.31)$$

Once transfer function of generator torque and blade pitch closed loop has been defined, and once way through which PI controllers' gains are computed has been explored, we can focus, specifically, on the two different modules to investigate the reference speed signals adopted and how the scheduling of gains is performed, varying according to the conditions in which the system is.

4.4.2.3 Generator Torque Controller

Four different generator torque controllers are available in ROSCO, they are the possible combination between two methods for below wind speed operations and two methods for above wind speed conditions.

Regarding below rated operations, to maximize extracted power at each wind condition, a quadratic low of generator torque with respect to rotor angular speed can be adopted. In this section we omit exploitation of this method since is the same adopted in Baseline controller (see equation 4.2). Alternatively, a tip speed ratio tracking to maintain TSR at its optimal value can be adopted. If the wind speed can be measured or estimated accurately, a generator torque controller can be designed to maintain the λ_{opt} and maximize power capture, so reference rotor angular speed becomes:

$$\Omega_{\mathrm{ref}_\tau} = \frac{\lambda_{\mathrm{opt}}\,\hat{\mathrm{U}}}{R} \qquad (4.32)$$

Where subscript τ indicates the reference speed of torque controller and $\widehat{U}$ is the estimated wind speed provided by wind estimator (one of the additional ROSCO modules). From equations 4.19, 4.22, 4.30 and 4.31, it can be seen than integral gain K_I of generator torque controller is constant, whereas A, so proportional gain K_P, are both dependent on U (wind speed). However, it was found that fixing K_P = K_P (U = U_{rated}) does not negatively affect power production.

Regarding the two existing methods for above rated conditions, first of them considers a constant generator torque, defined as:

$$C_{gen,ar}(t) = C_{rated} = \frac{P_{\mathrm{rated}}}{\Omega_{\mathrm{rated}}} \qquad (4.33)$$

Where subscript "ar" means "above rated".

On the other hand, second strategy considers a constant extracted power equal to its rated value, so generator torque is defined as:

$$C_{gen,ar}(t) = \frac{P_{\mathrm{rated}}}{\Omega} \qquad (4.34)$$

4.4.2.4 Blade Pitch Controller

Main goal of blade pitch controller is keeping rotor angular speed at its rated value, so reference speed is (both in below rated and above rated conditions):

$$\Omega_{\text{ref},\theta_{bl}} = \Omega_{\text{rated}} \quad (4.35)$$

Where subscript θ_{bl} means we refer to blade pitch controller. In below rated conditions, generator speed is lower than rated value, so $-\delta\Omega = \Omega_{\text{ref}} - \Omega > 0$ and, since gains are normally negative, θ_{bl} is saturated at its minimum value, defined by an additional module of ROSCO controller which will be discussed later. According to equations 4.31 and 4.32, to find controllers gain values, $B_{\theta_{bl}}(\boldsymbol{X}_{\text{op}})$ and $A(\boldsymbol{X}_{\text{op}})$ should be computed. They change for any operation point at which system is linearized, so they are function of $\boldsymbol{X}_{\text{op}} = \{\lambda_{\text{op}}, \theta_{bl_{\text{op}}}\}$. Linearization point can be the optimal steady state values chosen during strategy definition (i.e. those showed in Figure 4.10), for which there is a unique relationship between λ_{op} and $\theta_{bl_{\text{op}}}$. For this reason, $B_{\theta_{bl}}$ and A can be expressed with respect to $\theta_{bl_{\text{op}}}$, so gains' values are scheduled with θ_{bl} as parameter.

4.4.2.5 Additional Control Modules

In this section principal additional modules are briefly discussed to understand their functions and how they modify control output; for more information it is possible to consult [28]. They are:

- **Wind speed estimator:** This module estimates wind sped used for TSR tracking in the generator torque controller. Employed algorithm is based on a continuous-discrete Kalman filter, which exploits system model (equations 4.4 and 4.5), a wind auto regressive model and other information, like covariance matrices based on the expected wind field or measure's confidence of rotor speed to estimate a mean wind speed across rotor area at each time.

- **Set Point Smoothing:** Generator torque and blade pitch controllers will normally conflict with each other in near-rated operation due to incompatible reference rotor speed. To avoid this, a set point smoother can be employed; it shifts the speed reference signal of the inactive controller while the active one works. As an example, at above rated condition torque controller is the inactive one and vice versa. If TSR tracking were to be adopted for the torque generator, then the reference speed at high wind speeds would be higher than the one actually wanted (rated one), so the smoother brings the reference towards the rated speed and the resulting torque approaches the rated one, the one actually intended by adopting a constant torque strategy under above conditions.

- **Minimum pitch Saturation:** This module defines a minimum value of blade pitch angle which will be used as a saturation limit during control operations. It mainly modifies expected blade pitch values in region 1.5 and near rated conditions and leads to two effects:

    - **Peak shaving:** Near rated condition thrust value reaches the highest values, since below rated wind speed is lower and above rated condition blade pitching reduces that force. So, to limit loads, minimum pitch module imposes not null pitch angles also below rated wind speed, near that value.

- **Power maximization in low wind:** In region 1.5, as mentioned in control region section, a minimum value of rotor speed is imposed, so at low wind speeds TSR deviates far from its optimum value. To compensate this fact and to increase power coefficient value in this condition, blade pitch is led to be greater.

- **Floating offshore wind turbine feedback**: this module is though for FOWTs (Floating Offshore Wind Turbines) and introduces a new term in the PI blade pitch controller, which becomes:

$$\Delta\theta_{bl} = -k_{\mathrm{p}}\,\delta\Omega - k_{\mathrm{i}}\int_0^T \delta\Omega\ \mathrm{d}t + k_{\theta_{bl,float}}\,\dot{x}_t \qquad (4.36)$$

Additional term is tower-top velocity $\dot{x}_t$ multiplied by $k_{\theta_{bl,float}}$ gain. The latter is chosen from a manipulation of rotor equilibrium equation and structure pitch equation, in which expression of thrust and power coefficients compare. The aim is to find gains' value that reduces rotor angular acceleration to tower speed sensitivity in order to mitigate structure pitch effect on rotor aerodynamic torque. This expedient increases the average extracted power and stabilizes the structure.

5 Model Predictive Control

In this section an advanced method of control which has had a great impact and development over the thirty lasts years. It has so far been applied manly in chemical and petrochemical industry, but it is currently being used in several other sectors. This control family is named MPC (Model Predictive Control) and comes from optimal control theory; its operation is based on the knowledge of the mathematical model of the system to be controlled and the calculation of the control actions that allow the system to evolve in such a way as to achieve certain objectives. Typically, an MPC controller uses the model to calculate the controlled inputs which allow a certain cost function (depending on them) to be minimized, also taking into account the possibility of having limits on the control actions or system outputs provided by the mathematical model.

In the next sections, an overview of the main features of MPC controllers will be presented, with a brief discussion of the advantages and disadvantages compared to other types of control, then the main MPC algorithms will be listed, trying to understand when the use of one rather than other types is recommended and how they work. Finally, a particular case of MPC control will be discussed in detail (linear time invariant MPC controller), so that the next chapter, in which an MPC controller for controlling a wind turbine will be implemented and tested, can be addressed.

5.1 MPC: an overview

The main ingredients of an MPC algorithm are:

- **Process model**: MPC controllers exploit knowledge of the model of the system to be controlled to be able to predict its performance and, based on this, to obtain a control law that best achieves the set goals. The model of the system can be obtained in different ways, e.g., it is possible to obtain it from knowledge of the physical laws governing the phenomena under consideration, or it is possible to obtain a black-box model from experimental data.

- **Input, output, and state constraints**: MPC controllers can deal with constrained control problems, and it is therefore possible to foresee, when calculating optimal control actions, limits both in the control actions and in the states and outputs of the system to be controlled.

- **Cost function**: In order to quantify the goodness of the actions imposed by the controller, a cost function is normally defined, dependent on them. It must then be minimized to find the optimal values of the control actions (controller output and controlled system input).

- **Prediction and Control horizon**: MPC control is based on prediction of system behaviour to compute control action which optimizes it. In particular a forecast within a certain time frame, called prediction horizon, is taken into account, moreover, an MPC control can generate, at every instant, a control strategy that covers a certain time frame, called control horizon. This means that it is possible to predict the control action for that time such as to optimize the response of the system within the prediction horizon. The control horizon can be equal to the prediction one or less (assuming constant control actions in the remaining time).

In principle, it is therefore possible to carry out the numerical optimization at time intervals equal to the prediction horizon. Despite this, the so-called receding horizon algorithm is often adopted, in which the optimization is carried out at each time step and only the first result of the calculated sequence are imposed as control actions. This technique, although leading to a greater use of calculation capacity, often allows a more successfully targets tracking because it mitigates possible discrepancies between the system outputs predicted by the model and the actual ones.

- **Optimisation algorithm**: Once the cost function and constraints (if these exist) have been defined, the optimisation problem is fully formulated. It must then be solved using a specific algorithm, chosen according to the type of problem being handled. There are several algorithms that are useful for this purpose and some of them will be presented below.

Figure 5.1 shows some described features of an MPC control algorithm with a SISO (Single-Input Single-Output) system; we now pay attention to a certain time instant *k*, the red line represents the reference trajectory of the single output considered. It is defined for the entire prediction horizon and is the target to be aimed at. The control signals applied in the instants preceding k and the values measured in the past of the system output are shown in blue and orange, respectively. Instead, the light-blue and brown curves are, respectively, the value of the control quantity (defined for the whole control horizon, but constant from control horizon end to predicted horizon one) resulting from the optimization carried out at instant k. The result of the optimization is a finite number of values of the control action, each of which is constant for a certain time period, called sample time, which is the time step with which the input data are updated and with which the subsequent optimizations are interspersed in the receding horizon algorithm case. This discrete subdivision is mainly due to the need to have a finite number of unknowns (control actions) and due to this nature of operation of MPC controllers in most cases they deal with discrete-time models.

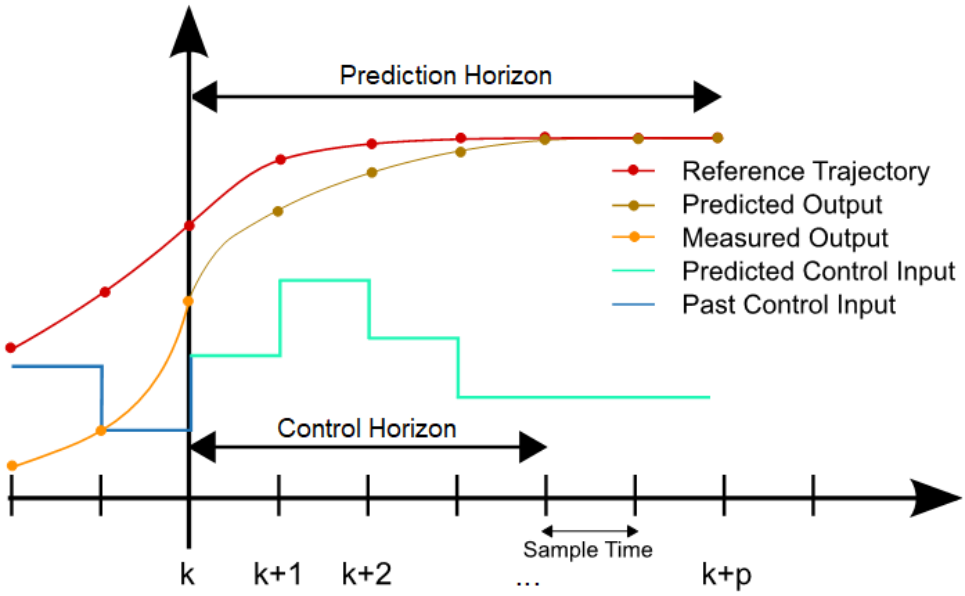

Figure 5.1: A basic working principle scheme of a discrete-time Model Predictive Control.

MPC controllers have had a great development and are chosen on several occasions because they have some characteristics that make them very competitive when compared to other types of control, such as traditional PID. Now that working principle has been understood, the main advantages and disadvantages compared to traditional controllers can be listed and quickly commented on. The main advantages can be summarized in:

- Intuitive basic idea behind MPC algorithms (basically main goal is simply track plant output to a desired value finding optimal control action thanks to model system knowledge) and simple tuning.

- An optimal control strategy can be calculated to achieve some desired plant output values, which is a feature that conventional control such as PID ones doesn't have. As we have seen in section 4, PID controllers can be tuned to have a desired closed-loop control dynamic, possibly by scheduling gains, but in any case, it is not possible to impose specific targets on the outputs of the plant to be controlled.

- Possibility to deal with black-box system models, which permits an MPC algorithm implementation also without knowing physics equation that describe the plant.

- Output constraints in the formulation of the control problem. For example, this option is not possible with other simpler controllers that are based on optimal control (for example LQR, Linear Quadratic Regulator, or LQG, Linear Quadratic Gaussian, controllers), as well as it is not possible in traditional controllers such as PID ones. Furthermore, considering the control actions constrains during optimization problem resolution eliminates problems linked with wind-up effect, since saturations of control signal are not ideally needed.

- Presence of constrains in the optimization problem permits to work near limit conditions. If the desired control action is outside the imposed limits, the actual action will most likely be close to these limits, causing system performance to be often maximized, usually better than what could be achieved with classic controllers.

- MPC is more convenient to use for Multiple-Input Multiple-Output (MIMO) systems than PID controllers because it is easily compatible with MIMO plants unlike PIDs where a lot of effort is needed to design flows where certain outputs of the system influence certain inputs. Actually, they are designed precisely to deal with this type of system, if then the system considered is SISO (Single-Input Single-Output) it will only be a matter of simplification of a MIMO.

- Robust MPC controller can be implemented. During real operation, plant or measurements disturbances can exist, so model prevision and output measurement can be subjected to errors; MPC controller can be implemented considering this fact in order to obtain a so-called robust control, thank to statistical information about above mentioned disturbances. Moreover, MPC can exploit prevision of plant disturbances to improve control actions; to do so estimation and forecasting tool are needed.

Although there are many advantages to using MPC controllers rather than other types, there are also some disadvantages, which will be briefly described below. They are:

- Problems related to the complexity in solving the constrained optimisation problem and thus to the high computational demand. This is the main problem usually encountered when dealing with this type of algorithm. However, this problem can be mitigated through certain measures, for example, it is possible to reduce the complexity of the model or to reduce the number of constraints. Another possibility is to reduce the control and prediction horizons, or to place a maximum limit on the number of iterations used during the optimisation procedure.

  If after trying these measures an effective compromise has not been found, a good idea would be to use a special type of MPC controller, called explicit MPC. Using the latter, it is not necessary to solve the optimisation problem online, but it is solved offline for a number of situations (dictated by the system's operating conditions) by obtaining the corresponding control strategies. During the system's operation, then, depending on the operating conditions, the results calculated offline will be applied when the system was in conditions similar to these. This technique allows a reduction in the computing power required during online operations, at the expense of more memory required to store offline results.

- As seen above, MPC controllers are based on the model of the system, in most cases this model is expressed in the state space form, and to solve the optimisation problem, knowledge of the states of the system at each instant is required. In general, it is possible that the information regarding certain states is not available due to the lack of instruments to measure these states, which could be a disadvantage when using this type of controller. Usually, to overcome this drawback, estimators are used that exploit knowledge of the system's model and available measurements to evaluate quantities not known a priori. A frequently used estimation algorithm for this purpose is the Kalman filter, which will be discussed in detail (one of its many versions) later.

Once basic concept of MPC controls and their main advantages and disadvantages are presented, we can make a simple classification of main types of MPC controller, also to understand the vastity of possible variation from the same idea. Since, as we have just seen, computational time is an important issue, the simplest version of MPC algorithm is usually used, in particular, it considers a linear state space model of the system, with linear constraint (both equality and inequality constraints can be implemented) and a simple form of cost function, i.e. a linear or a quadratic form. However, several times highly non-linear plant are involved, so it is difficult to deal with a linear model to control the system because that model may not represent well the plant. Moreover, sometimes more complicated constrains or cost function are needed, so different model predictive control algorithm should be used.

A few types of MPC controllers will be described below, but it should be noted that there are many different variants not mentioned, not least because it is difficult to make a proper classification of this type of controller, since what they have in common is only their basic operating principle (technically, any control algorithm that uses the system's model to predict its behaviour in order to achieve optimal control can be placed in this category). The prominent types are as follows:

- **Linear-Time-Invariant (LTI) MPC**: This is an MPC used to control linear plant dynamical systems with linear constraints and quadratic (or linear) cost functions. This is the simplest and widely used type, it is often used also with plant with non-linearities, as long as they are not too accentuated and in this case it is possible to linearize the non-linear model of the system in a suitable point. Discrete-time linear systems are often used, but it is also possible to implement MPC controllers that are based on continuous systems, even though the controller output continues to have a discrete nature.

- **Adaptive MPC:** This MPC is used for highly non-linear plant models with linear constraints and linear or quadratic cost functions. The plant model is linearized at each point of interest in an MPC cycle so that a LTI MPC can be used to control this linearized model. This LTI model only works well at that particular operating point, so linearization of the system must be done at each time step (this considerably increase online computational time). The structure of the optimization problem, i.e. the number of states and constraints, remain the same at each operating point.

- **Gain-scheduled MPC:** With this method, the non-linear system model is linearized offline to obtain several LTI MPC controller, each one valid in a particular condition. In this case each model may have its own unique number of states and constraints. When running, a switching algorithm is used to select the appropriate MPC at each time step, this consumes a lot more memory than previous MPC types since multiple MPCs are stored in memory, but it may be computationally less expensive since several operations are performed offline.

- **Non-Linear MPC:** This MPC is used for nonlinear plant models with nonlinear cost functions and nonlinear constraints. As such, optimization is usually non-convex with many local optima and a global minimum difficult to find. Solving this optimization problem requires advanced optimization algorithms, which require high computational efforts, without the certainty of finding an optimal solution that leads to a stable and effective control of the system.

As above mentioned, several other variants of MPC algorithm exist; as an example, in the next chapter an hybrid form between adaptive and gain-scheduled MPC will be implemented.

In the next section simplest MPC version (LTI MPC) will be widely described to obtain an MPC problem solution, also describing some optimization algorithms, which can be exploited for the wind turbine MPC control implementation, main objective of this work.

5.2 Linear Time Invariant (LTI) Model Predictive Control

In this section the simplest form of MPC, based on linear time invariant plant model, linear constrains and quadratic cost function, will be described in order to understand how a model predictive control works and how it can be implemented. Methods and solutions obtained here will be then used to implement an MPC control for an FOWT (Floating Offshore Wind Turbine) in the next chapter of this thesis.

LTI MPC control exploits a linear model of the system, so it works well with systems whose differential equations governing it are linear or have slight non-linearities, such that its behaviour can be effectively approximated by a linear system obtained by linearising the starting non-linear model. In this section Multiple-Input Multiple-Output (MIMO) systems expressed in a state space form are considered, given that this mode of representation works well with MIMO systems, moreover, the MPC controller studied will be based on discrete systems, given the simplicity of treatment that follows and the good compatibility with MPC logic, which is discrete by nature.

At first, the system model and cost function will be defined, and then the solution will be found if no constraints of any kind are present. In a second step, the case with constraints will be dealt with, in which some methods used to solve the constrained optimisation problem will be briefly described.

5.2.1 Plant Model

In this section plant's discrete state space model which is used in the MPC algorithm will be derived. As starting point, we define the linear continuous state space model:

$$\begin{cases} \dot{\boldsymbol{x}}_{\boldsymbol{m}}(t) = \boldsymbol{A}_{m,c}\,\boldsymbol{x}_{\boldsymbol{m}}(t) + \boldsymbol{B}_{m,c}\,\boldsymbol{u}_{\boldsymbol{m}}(t) \\ \boldsymbol{y}_{\boldsymbol{m}}(t) = \boldsymbol{C}_{m,c}\,\boldsymbol{x}_{\boldsymbol{m}}(t) + \boldsymbol{D}_{m,c}\,\boldsymbol{u}_{\boldsymbol{m}}(t) \end{cases} \qquad (5.1)$$

Where $\boldsymbol{x}(t)$, $\boldsymbol{u}(t)$ and $\boldsymbol{y}(t)$ are, respectively, the state, input, and output vectors. Assuming $\boldsymbol{x} \in \mathbb{R}^{\mathrm{n\,x\,1}}$, $\boldsymbol{u} \in \mathbb{R}^{\mathrm{m\,x\,1}}$ and $\boldsymbol{y} \in \mathbb{R}^{\mathrm{q\,x\,1}}$, then $\boldsymbol{A}_{\mathrm{m,c}} \in \mathbb{R}^{\mathrm{n\,x\,n}}$, $\boldsymbol{B}_{\mathrm{m,c}} \in \mathbb{R}^{\mathrm{n\,x\,m}}$, $\boldsymbol{C}_{\mathrm{m,c}} \in \mathbb{R}^{\mathrm{q\,x\,n}}$ and $\boldsymbol{D}_{\mathrm{m,c}} \in \mathbb{R}^{\mathrm{q\,x\,m}}$. Subscripts m and c indicates we are referring to *model* of the system in the *continuous* form. Moreover, due to the principle of receding horizon control, where a current information of the plant is required for prediction and control, we have implicitly assumed that the input cannot affect the output at the same time. Thus, $\boldsymbol{D}_{\mathrm{m,c}}$ = 0 in the plant model and solution of the system is:

$$\begin{cases} \boldsymbol{x}_{\boldsymbol{m}}(t) = e^{\boldsymbol{A}_{m,c}(t-t_0)}\,\boldsymbol{x}_{\boldsymbol{m}}(t_0) + \int_{t_0}^{t} e^{\boldsymbol{A}_{m,c}[t-\tau]}\,\boldsymbol{B}_{m,c}\,\boldsymbol{u}_{\boldsymbol{m}}(\tau)d\tau \\ \boldsymbol{y}_{\boldsymbol{m}}(t) = \boldsymbol{C}_{m,c}\,\boldsymbol{x}_{\boldsymbol{m}}(t) \end{cases} \qquad (5.2)$$

Where $e^{\boldsymbol{A}_{m,c}\,t}$ is defined as:

$$e^{\boldsymbol{A}_{m,c}\,t} = I + At + \frac{1}{2!}A^2t^2 + \frac{1}{3!}A^3t^3 + \cdots \qquad (5.3)$$

To obtain the discrete form of model state space, we define $t_0 = kT$ and $t = (k+1)T$, where T is the sample time such that: $\boldsymbol{x_m}^k = \boldsymbol{x_m}(kT)$, $\boldsymbol{u_m}^k = \boldsymbol{u_m}(kT)$ and $\boldsymbol{y_m}^k = \boldsymbol{y_m}(kT)$. With these definitions we can write:

$$\boldsymbol{x_m}((k+1)T) = e^{\boldsymbol{A}_{m,c}T}\,\boldsymbol{x_m}(kT) + \int_{kT}^{(k+1)T} e^{\boldsymbol{A}_{m,c}[(k+1)T-\tau]}\,\boldsymbol{B}_{m,c}\,\boldsymbol{u_m}(\tau)d\tau \quad (5.4)$$

Here we assume $\boldsymbol{u}(t)$ is constant along the sample time T, the discretization coming from this assumption is called ZOH (zero order hold). For information, there are other types of discretization, such as the one proposed by Euler, bilinear discretization or first order hold discretization, which assumes a linear variation of the input along time T. With this assumption and defining $\lambda = \tau - kT$, equation 5.4 becomes:

$$\boldsymbol{x_m}^{k+1} = e^{\boldsymbol{A}_{m,c}T}\,\boldsymbol{x_m}^k + \int_0^{\mathrm{T}} e^{\boldsymbol{A}_{m,c}[T-\lambda]}\,\boldsymbol{B}_{m,c}\,d\lambda \cdot\, \boldsymbol{u_m}^k \quad (5.5)$$

Changing variables again to $\tau = T - \lambda$ yields finally:

$$\boldsymbol{x_m}^{k+1} = e^{\boldsymbol{A}_{m,c}T}\,\boldsymbol{x_m}^k + \int_0^{\mathrm{T}} e^{\boldsymbol{A}_{m,c}\tau}\,\boldsymbol{B}_{m,c}\,d\tau \cdot\, \boldsymbol{u_m}^k \qquad (5.6)$$

Now we can define discrete form of matrices $\boldsymbol{A}_{m,c}$ and $\boldsymbol{B}_{m,c}$:

$$\boldsymbol{A}_{m,d} = e^{\boldsymbol{A}_{m,c}T} \qquad (5.7)$$

$$\boldsymbol{B}_{m,d} = \int_0^{\mathrm{T}} e^{\boldsymbol{A}_{m,c}\tau}\,\boldsymbol{B}_{m,c}\,d\tau \qquad (5.8)$$

Finally, by these definitions we can write the discrete state space form of plant model, with discretization time T such that $\boldsymbol{x_m}^k = \boldsymbol{x_m}(kT)$, $\boldsymbol{u_m}^k = \boldsymbol{u_m}(kT)$ and $\boldsymbol{y_m}^k = \boldsymbol{y_m}(kT)$:

$$\begin{cases} \boldsymbol{x_m}^{k+1} = \boldsymbol{A}_{m,d}\,\boldsymbol{x_m}^k + \boldsymbol{B}_{m,d}\,\boldsymbol{u}^k \\ \boldsymbol{y_m}^k = \boldsymbol{C}_{m,d}\,\boldsymbol{x_m}^k \end{cases} \qquad (5.9)$$

Whit $\boldsymbol{C}_{m,d} = \boldsymbol{C}_{m,c}$, since output equation is a non-dynamic equation.

Once discrete state space model is obtained, it can be augmented to arrive at the so-called augmented system, which has an embedded integrator, useful, for control purposes, for those systems that do not already have this feature within them. From equation 5.9, since it is valid at each time, following expressions can be write:

$$\boldsymbol{x_m}^{k} = \boldsymbol{A}_{m,d}\, \boldsymbol{x_m}^{k-1} + \boldsymbol{B}_{m,d}\, \boldsymbol{u_m}^{k-1} \quad (5.10)$$

Now, defining:

$$\Delta \boldsymbol{x_m}^{k+1} = \boldsymbol{x_m}^{k+1} - \boldsymbol{x_m}^{k} \quad (5.11)$$

$$\Delta \boldsymbol{u_m}^{k+1} = \boldsymbol{u_m}^{k+1} - \boldsymbol{u_m}^{k} \quad (5.12)$$

$$\Delta \boldsymbol{y_m}^{k+1} = \boldsymbol{y_m}^{k+1} - \boldsymbol{y_m}^{k} \quad (5.13)$$

And subtracting 5.9 and 5.10, we arrive to:

$$\Delta \boldsymbol{x_m}^{k+1} = \boldsymbol{x_m}^{k+1} - \boldsymbol{x_m}^{k} = \boldsymbol{A}_{m,d} \left(\boldsymbol{x_m}^{k} - \boldsymbol{x_m}^{k-1}\right) + \boldsymbol{B}_{m,d} \left(\boldsymbol{u_m}^{k} - \boldsymbol{u_m}^{k-1}\right) \quad (5.14)$$

$$\Delta \boldsymbol{x_m}^{k+1} = \boldsymbol{A}_{m,d}\, \Delta \boldsymbol{x_m}^{k} + \boldsymbol{B}_{m,d}\, \Delta \boldsymbol{u_m}^{k+1} \quad (5.15)$$

In order to relate the output $\boldsymbol{y_m}^{k+1}$ to the state variable $\Delta \boldsymbol{x_m}^{k}$ and $\boldsymbol{y_m}^{k}$, we deduce that:

$$\boldsymbol{y_m}^{k+1} = \boldsymbol{y_m}^{k} + \Delta \boldsymbol{y_m}^{k+1} = \boldsymbol{y_m}^{k} + \boldsymbol{C}_{m,d}\, \Delta \boldsymbol{x_m}^{k+1} = \boldsymbol{y_m}^{k} + \boldsymbol{C}_{m,d}\, \boldsymbol{A}_{m,d}\, \Delta \boldsymbol{x_m}^{k} + \boldsymbol{C}_{m,d}\, \boldsymbol{B}_{m,d}\, \Delta \boldsymbol{u_m}^{k} \quad (5.16)$$

Finally, we define state and output vectors of the augmented system ($\boldsymbol{x}$ and $\boldsymbol{y}$) as:

$$\boldsymbol{x}^{k} = \left[\Delta \boldsymbol{x_m}^{k^T} \;\; \boldsymbol{y_m}^{k^T}\right]^{T} \quad (5.17)$$

$$\boldsymbol{y}^{k} = \boldsymbol{y_m}^{k} \quad (5.18)$$

From equations obtained above, following relations can be written:

$$\boldsymbol{x}^{k+1} = \begin{bmatrix} \Delta \boldsymbol{x_m}^{k+1} \\ \boldsymbol{y}^{k+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_{m,d} & \boldsymbol{Z} \\ \boldsymbol{C}_{m,d}\, \boldsymbol{A}_{m,d} & \boldsymbol{I} \end{bmatrix} \cdot \begin{bmatrix} \Delta \boldsymbol{x_m}^{k} \\ \boldsymbol{y}^{k} \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}_{m,d} \\ \boldsymbol{C}_{m,d}\, \boldsymbol{B}_{m,d} \end{bmatrix} \cdot \Delta \boldsymbol{u_m}^{k} \quad (5.19)$$

$$\boldsymbol{y}^{k} = \begin{bmatrix} \boldsymbol{Z} & \boldsymbol{I} \end{bmatrix} \cdot \begin{bmatrix} \Delta \boldsymbol{x_m}^{k} \\ \boldsymbol{y}^{k} \end{bmatrix} \quad (5.20)$$

Where $\boldsymbol{I}$ is the identity matrix with dimensions q×q, which is the number of outputs; and $\boldsymbol{Z}$ is a n×q zero matrix in equation 5.19 and a q×n zero matrix in equation 5.20.

Rewriting equations 5.19 and 5.20 in a more compact form, augmented system is:

$$\begin{cases} \boldsymbol{x}^{k+1} = \boldsymbol{A}\,\boldsymbol{x}^{k} + \boldsymbol{B}\,\Delta\boldsymbol{u}^{k} \\ \boldsymbol{y}^{k} = \boldsymbol{C}\,\boldsymbol{x}^{k} \end{cases} \qquad (5.21)$$

Where following definitions hold:

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_{m,d} & \mathbf{Z} \\ \boldsymbol{C}_{m,d}\,\boldsymbol{A}_{m,d} & \boldsymbol{I} \end{bmatrix},\ \boldsymbol{A} \in \mathbb{R}^{(\mathrm{n+q})\ \mathrm{x}\ (\mathrm{n+q})} \qquad (5.22)$$

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{B}_{m,d} \\ \boldsymbol{C}_{m,d}\,\boldsymbol{B}_{m,d} \end{bmatrix},\ \boldsymbol{B} \in \mathbb{R}^{(\mathrm{n+q})\ \mathrm{x}\ \mathrm{m}} \qquad (5.23)$$

$$\boldsymbol{C} = [\boldsymbol{Z} \quad \boldsymbol{I}],\ \boldsymbol{C} \in \mathbb{R}^{\mathrm{q}\ \mathrm{x}\ (\mathrm{n+q})} \qquad (5.24)$$

$$\boldsymbol{x}^{k} = \left[\Delta\boldsymbol{x_m}^{k^T}\ \ \boldsymbol{y_m}^{k^T}\right]^T,\ \boldsymbol{x}^{k} \in \mathbb{R}^{(\mathrm{n+q})\ \mathrm{x}\ 1} \qquad (5.25)$$

$$\Delta\boldsymbol{u}^{k} = \Delta\boldsymbol{u_m}^{k},\ \Delta\boldsymbol{u}^{k} \in \mathbb{R}^{\mathrm{m}\ \mathrm{x}\ 1} \qquad (5.26)$$

$$\boldsymbol{y}^{k} = \boldsymbol{y_m}^{k},\ \boldsymbol{y}^{k} \in \mathbb{R}^{\mathrm{q}\ \mathrm{x}\ 1} \qquad (5.27)$$

5.2.2 Prediction of the Outputs

Once mathematical model of the system is formulated, the next step in designing an MPC is to calculate the predicted output, with the control signals as variables, within the prediction horizon. It is defined as $H_p = T \cdot N_p$, where T is the discretization time (corresponding to sample time used to obtain discrete model from continuous one) and N_p is the size of prediction window. On the other hand, control horizon is the horizon within control variables are defined; it is defined as $H_c = T \cdot N_c$, where N_c is the size of control window (equal or smaller than N_p). Furthermore, we assume that the control variables remain constant over the time span between the control and prediction horizons, so, remembering relation 5.26, $\Delta\boldsymbol{u}$ is set to zero over this time.

Remembering relations 5.26 and 5.27, we define future control trajectory vector $\Delta\boldsymbol{U}$ and future output trajectory vector $\boldsymbol{Y}$ as:

$$\Delta\boldsymbol{U} = \left[\Delta\boldsymbol{u}^{k^T}, \Delta\boldsymbol{u}^{k+1^T}, \dots, \Delta\boldsymbol{u}^{k+N_c-1^T}\right]^T,\ \Delta\boldsymbol{U} \in \mathbb{R}^{(N_c \cdot \mathrm{m})\ \mathrm{x}\ 1} \qquad (5.28)$$

$$\boldsymbol{Y} = \left[\boldsymbol{y}^{k+1^T}, \boldsymbol{y}^{k+2^T}, \dots, \boldsymbol{y}^{k+N_p{}^T}\right]^T,\ \boldsymbol{Y} \in \mathbb{R}^{(N_p \cdot \mathrm{q})\ \mathrm{x}\ 1} \qquad (5.29)$$

Where *k* indicates the instant at which the optimization is done; here we assume the state vector $\boldsymbol{x}^{k}$, which describes the plant information, is available through measurements. The more general situation where the state is not directly measured will be discussed later.

From state space form of the augmented system defined in 5.21, the future state variables can be calculated as follow:

$$\begin{aligned}
&\boldsymbol{x}^{k+1} = \boldsymbol{A}\,\boldsymbol{x}^{k} + \boldsymbol{B}\,\Delta\boldsymbol{u}^{k} \\
&\boldsymbol{x}^{k+2} = \boldsymbol{A}\,\boldsymbol{x}^{k+1} + \boldsymbol{B}\,\Delta\boldsymbol{u}^{k+1} = \boldsymbol{A}^{2}\boldsymbol{x}^{k} + \boldsymbol{AB}\,\Delta\boldsymbol{u}^{k} + \boldsymbol{B}\,\Delta\boldsymbol{u}^{k+1} \\
&\vdots \\
&\boldsymbol{x}^{k+N_p} = \boldsymbol{A}^{N_p}\boldsymbol{x}^{k} + \boldsymbol{A}^{N_p-1}\boldsymbol{B}\,\Delta\boldsymbol{u}^{k} + \boldsymbol{A}^{N_p-2}\boldsymbol{B}\,\Delta\boldsymbol{u}^{k+1} + \cdots + \boldsymbol{A}^{N_p-N_c}\boldsymbol{B}\,\Delta\boldsymbol{u}^{k+N_c-1}
\end{aligned} \tag{5.30}$$

From the predicted state variables, the predicted output variables are:

$$\begin{aligned}
&\boldsymbol{y}^{k+1} = \boldsymbol{CA}\,\boldsymbol{x}^{k} + \boldsymbol{CB}\,\Delta\boldsymbol{u}^{k} \\
&\boldsymbol{y}^{k+2} = \boldsymbol{CA}\,\boldsymbol{x}^{k+1} + \boldsymbol{CB}\,\Delta\boldsymbol{u}^{k+1} = \boldsymbol{CA}^{2}\boldsymbol{x}^{k} + \boldsymbol{CAB}\,\Delta\boldsymbol{u}^{k} + \boldsymbol{CB}\,\Delta\boldsymbol{u}^{k+1} \\
&\boldsymbol{y}^{k+3} = \boldsymbol{CA}^{3}\boldsymbol{x}^{k} + \boldsymbol{CA}^{2}\boldsymbol{B}\,\Delta\boldsymbol{u}^{k} + \boldsymbol{CAB}\,\Delta\boldsymbol{u}^{k+1} \\
&\vdots \\
&\boldsymbol{y}^{k+N_p} = \boldsymbol{CA}^{N_p}\boldsymbol{x}^{k} + \boldsymbol{CA}^{N_p-1}\boldsymbol{B}\,\Delta\boldsymbol{u}^{k} + \boldsymbol{CA}^{N_p-2}\boldsymbol{B}\,\Delta\boldsymbol{u}^{k+1} + \cdots + \boldsymbol{CA}^{N_p-N_c}\boldsymbol{B}\,\Delta\boldsymbol{u}^{k+N_c-1}
\end{aligned} \tag{5.31}$$

All predicted variables are formulated in terms of current state variable information $\boldsymbol{x}^{k}$ and the future control movement $\Delta\boldsymbol{U}$; collecting 5.28, 5.29 and 5.31 we can write the output vector as:

$$\boldsymbol{Y} = \boldsymbol{F}\cdot\boldsymbol{x}^{k} + \boldsymbol{\Phi}\cdot\Delta\boldsymbol{U} \tag{5.32}$$

Where matrices $\boldsymbol{F}$ and $\boldsymbol{\Phi}$ are defined as:

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{CA} \\ \boldsymbol{CA^{2}} \\ \boldsymbol{CA^{3}} \\ \vdots \\ \boldsymbol{CA^{N_P}} \end{bmatrix} \tag{5.33}$$

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{CB} & 0 & 0 & \dots & 0 \\ \boldsymbol{CAB} & \boldsymbol{CB} & 0 & \dots & 0 \\ \boldsymbol{CA^{2}B} & \boldsymbol{CAB} & \boldsymbol{CB} & \dots & 0 \\ \vdots & & & & \\ \boldsymbol{CA^{N_P-1}B} & \boldsymbol{CA^{N_P-2}B} & \boldsymbol{CA^{N_P-3}B} & \dots & \boldsymbol{CA^{N_P-N_c}B} \end{bmatrix} \tag{5.34}$$

Matrix $\boldsymbol{F}$ has dimension $qN_p \times n$ and $\boldsymbol{\Phi}$ has dimension $qN_p \times mN_c$.

5.2.3 Cost Function

The main objective of an MPC control is to find components of the control vector $\mathbf{U}$ such that a cost function is minimised. The cost function is usually defined in such a way as to consider both the output target and the control input target, i.e. the output signals must be as close as possible to the target and the rate of change of the control action must be as small as possible, in order to reduce energy requirements and actuator wear.

A possible cost function may be defined as:

$$J = \frac{1}{2}\left[(\boldsymbol{T_Y} - \boldsymbol{Y})^T \cdot \boldsymbol{W_Y} \cdot (\boldsymbol{T_Y} - \boldsymbol{Y}) + \Delta \boldsymbol{U}^T \cdot \boldsymbol{W_{\Delta U}} \cdot \Delta \boldsymbol{U}\right] \qquad (5.35)$$

With:

$$\Delta \boldsymbol{U} = \left[\Delta \boldsymbol{u}^{k^T}, \Delta \boldsymbol{u}^{k+1^T}, \dots, \Delta \boldsymbol{u}^{k+N_c-1^T}\right]^T,\ \Delta \boldsymbol{U} \in \mathbb{R}^{(N_c \cdot \mathrm{m})\ \mathrm{x}\ 1} \qquad (5.36)$$

$$\boldsymbol{Y} = \left[\boldsymbol{y}^{k+1^T}, \boldsymbol{y}^{k+2^T}, \dots, \boldsymbol{y}^{k+N_p{}^T}\right]^T,\ \boldsymbol{Y} \in \mathbb{R}^{(N_p \cdot \mathrm{q})\ \mathrm{x}\ 1} \qquad (5.37)$$

$$\boldsymbol{T_Y} = \left[\boldsymbol{t}^{k+1^T}, \boldsymbol{t}^{k+2^T}, \dots, \boldsymbol{t}^{k+N_p{}^T}\right]^T,\ \boldsymbol{T_Y} \in \mathbb{R}^{(N_p \cdot \mathrm{q})\ \mathrm{x}\ 1} \qquad (5.38)$$

Vectors $\boldsymbol{t}$ are target vectors ($\in \mathbb{R}^{q\ \mathrm{x}\ 1}$) for each time step within prediction horizon, while matrices $\boldsymbol{W_Y}$ ($\in \mathbb{R}^{(N_p \cdot \mathrm{q})\ \mathrm{x}\ (N_p \cdot \mathrm{q})}$) and $\boldsymbol{W_{\Delta U}}$ ($\in \mathbb{R}^{(N_c \cdot \mathrm{m})\ \mathrm{x}\ (N_c \cdot \mathrm{m})}$) are diagonal weight matrices through which different importance can be given to different targets or inputs rate of change.

5.2.4 Solution Without Constraints

Once a cost function is defined and the prediction of system outputs is computed, we can obtain the solution of the optimization problem, that is the control action vector $\Delta \boldsymbol{U}$ which minimize the cost function. We will initially focus on solving the relatively uncommon unconstrained problem, in which there are no conditions to be fulfilled in terms of output or control actions, and then concentrate in detail on the problem with constraints.

By making some substitutions in relation 5.35, in particular, by inserting definitions 5.32, 5.33 and 5.34, the cost function can be expressed as follows:

$$J = \frac{1}{2}\left[\left(\boldsymbol{T_Y} - \boldsymbol{F} \cdot \boldsymbol{x}^k - \boldsymbol{\Phi} \cdot \Delta \boldsymbol{U}\right)^T \cdot \boldsymbol{W_Y} \cdot \left(\boldsymbol{T_Y} - \boldsymbol{F} \cdot \boldsymbol{x}^k - \boldsymbol{\Phi} \cdot \Delta \boldsymbol{U}\right) + \Delta \boldsymbol{U}^T \cdot \boldsymbol{W_{\Delta U}} \cdot \Delta \boldsymbol{U}\right] \qquad (5.39)$$

$$J = \frac{1}{2}\left(\boldsymbol{T_Y} - \boldsymbol{F} \cdot \boldsymbol{x}^k\right)^T \cdot \boldsymbol{W_Y} \cdot \left(\boldsymbol{T_Y} - \boldsymbol{F} \cdot \boldsymbol{x}^k\right) - \Delta \boldsymbol{U}^T \cdot \boldsymbol{\Phi}^T \cdot \boldsymbol{W_Y} \cdot \left(\boldsymbol{T_Y} - \boldsymbol{F} \cdot \boldsymbol{x}^k\right) +$$

$$+ \frac{1}{2}\Delta \boldsymbol{U}^T \cdot \boldsymbol{\Phi}^T \cdot \boldsymbol{W_Y} \cdot \boldsymbol{\Phi} \cdot \Delta \boldsymbol{U} + \frac{1}{2}\Delta \boldsymbol{U}^T \cdot \boldsymbol{W_{\Delta U}} \cdot \Delta \boldsymbol{U} \qquad (5.40)$$

Now, noting that the first term of 5.40 does not depend on $\Delta \boldsymbol{U}$, the cost function became a canonical quadratic function and can be rewritten as:

$$J = \frac{1}{2}\Delta \boldsymbol{U}^T \cdot (\boldsymbol{\Phi}^T \cdot \boldsymbol{W_Y} \cdot \boldsymbol{\Phi} + \boldsymbol{W}_{\Delta U}) \cdot \Delta \boldsymbol{U} - \Delta \boldsymbol{U}^T \cdot \boldsymbol{\Phi}^T \cdot \boldsymbol{W_Y} \cdot \left(\boldsymbol{T_Y} - \boldsymbol{F} \cdot \boldsymbol{x}^k\right) \quad (5.41)$$

Finally, by imposing $\frac{\partial J}{\partial \Delta U} = 0$, we can compute optimal value of Δ**U**, which minimize cost function:

$$\frac{\partial J}{\partial \boldsymbol{U}} = (\boldsymbol{\Phi}^T \cdot \boldsymbol{W_Y} \cdot \boldsymbol{\Phi} + \boldsymbol{W}_{\Delta U}) \cdot \Delta \boldsymbol{U} - \boldsymbol{\Phi}^T \cdot \boldsymbol{W_Y} \cdot \left(\boldsymbol{T_Y} - \boldsymbol{F} \cdot \boldsymbol{x}^k\right) = 0 \quad (5.42)$$

$$\Delta \boldsymbol{U} = (\boldsymbol{\Phi}^T \cdot \boldsymbol{W_Y} \cdot \boldsymbol{\Phi} + \boldsymbol{W}_{\Delta U})^{-1} \cdot \left(\boldsymbol{\Phi}^T \cdot \boldsymbol{W_Y} \cdot \boldsymbol{T_Y} - \boldsymbol{\Phi}^T \cdot \boldsymbol{W_Y} \cdot \boldsymbol{F} \cdot \boldsymbol{x}^k\right) \quad (5.43)$$

Matrix $(\boldsymbol{\Phi}^T \cdot \boldsymbol{W_Y} \cdot \boldsymbol{\Phi} + \boldsymbol{W}_{\Delta U})$ has dimension $mN_c \times mN_c$, $\boldsymbol{\Phi}^T \cdot \boldsymbol{W_Y} \cdot \boldsymbol{F}$ has dimension $mN_c \times n$ and $\boldsymbol{\Phi}^T \cdot \boldsymbol{W_Y}$ has dimension $mN_c \times qN_p$.

Considering a receding horizon control strategy, through which only firsts m elements of $\Delta \boldsymbol{U}$ are considered and repeating the optimization procedure at each time step, the optimization procedure result is, according with definition 5.36:

$$\Delta \boldsymbol{u}^k = [\boldsymbol{I_m} \quad \boldsymbol{Z_m} \quad \dots \quad \boldsymbol{Z_m}] \cdot (\boldsymbol{\Phi}^T \cdot \boldsymbol{W_Y} \cdot \boldsymbol{\Phi} + \boldsymbol{W}_{\Delta U})^{-1} \cdot \left(\boldsymbol{\Phi}^T \cdot \boldsymbol{W_Y} \cdot \boldsymbol{T_Y} - \boldsymbol{\Phi}^T \cdot \boldsymbol{W_Y} \cdot \boldsymbol{F} \cdot \boldsymbol{x}^k\right) \quad (5.44)$$

Where $\boldsymbol{I_m}$ and $\boldsymbol{Z_m}$ are, respectively, the identity and zero matrix with dimension m × m. Equation 5.44 can be rewrite in a more compact form:

$$\Delta \boldsymbol{u}^k = \mathbf{K_y} \cdot \boldsymbol{T_Y} - \mathbf{K_{mpc}} \cdot \boldsymbol{x}^k \quad (5.45)$$

With:
$$\mathbf{K_y} = [\boldsymbol{I_m} \quad \boldsymbol{Z_m} \quad \dots \quad \boldsymbol{Z_m}] \cdot (\boldsymbol{\Phi}^T \cdot \boldsymbol{W_Y} \cdot \boldsymbol{\Phi} + \boldsymbol{W}_{\Delta U})^{-1} \cdot \boldsymbol{\Phi}^T \cdot \boldsymbol{W_Y} \quad (5.46)$$

$$\mathbf{K_{mpc}} = [\boldsymbol{I_m} \quad \boldsymbol{Z_m} \quad \dots \quad \boldsymbol{Z_m}] \cdot (\boldsymbol{\Phi}^T \cdot \boldsymbol{W_Y} \cdot \boldsymbol{\Phi} + \boldsymbol{W}_{\Delta U})^{-1} \cdot (-\boldsymbol{\Phi}^T \cdot \boldsymbol{W_Y} \cdot \boldsymbol{F}) \quad (5.47)$$

Finally, we can write the closed-loop system obtained through receding horizon MPC control substituting equation 5.45 into the augmented system equation 5.21:

$$\begin{cases} \boldsymbol{x}^{k+1} = (\boldsymbol{A} - \boldsymbol{B} \cdot \mathbf{K_{mpc}}) \cdot \boldsymbol{x}^k + \boldsymbol{B} \cdot \mathbf{K_y} \cdot \boldsymbol{T_Y} \\ \boldsymbol{y}^k = \boldsymbol{C}\, \boldsymbol{x}^k \end{cases} \quad (5.48)$$

5.2.5 Solution With Constraints

We have seen, in section 5.2.4, how future control action vector can be computed to minimize cost function. This vector, $\Delta \boldsymbol{U}$, has been obtained not considering rate of change of control actions can have a limit, imposed by actuators characteristics. This fact may lead to a deterioration of system performances, since actual control action can be saturated due to physical limits and, as a result, outputs may be pretty different than those predicted by ideal solution. Taking into account actuators limits, on the other hand, can improve performances, also with the same physical limit, because during optimization problem computation a sub-optimal solution, which respect limits, is obtained, and in the most of cases it is better than the optimal solution a posteriori saturated.

Furthermore, in addition to the limits on the rate of change of control actions, the limits of actions in an absolute sense should often also be considered, in order to avoid finding impossible solutions and falling into the same problem. This section will define how these constraints can be taken into account when solving the optimization problem, which becomes a constrained optimization problem. Here, for the sake of simplicity, only linear constraints will be considered, but it is also possible to deal with more complex non-linear ones. Finally, to complete the discussion, constraints on outputs will also be dealt with, also in the form of linear inequality. This type of constraint is very effective from the point of view of the final performance of the control and is usually impossible to apply in traditional controllers.

Next, the way of expressing the above-mentioned constraints will be defined, after which some methods for solving constrained optimizations will be quickly presented, and finally, the method used in the controller implemented in this work will be discussed in more detail.

5.2.5.1 Constraints Definition

In these section the three types of constraints above mentioned will be defined, first one considers the limit on incremental variation of control variables, second one considers their absolute value, whereas third type considers output constraints.

Constraint on Control Variable Rate of Change

Considering a Multiple-Input system, with *m* control inputs, and according with definition 5.26, upper and lower limit on their rate of change can be defined as:

$$\Delta \boldsymbol{u}^{max} = [\Delta u_1^{max}, \Delta u_2^{max}, \dots, \Delta u_m^{max}]^T \quad (5.49)$$

$$\Delta \boldsymbol{u}^{min} = \left[\Delta u_1^{min}, \Delta u_2^{min}, \dots, \Delta u_m^{min}\right]^T \quad (5.50)$$

Assuming, moreover, that these limits are constant throughout the time span defined by the control horizon H_c, we can write, remembering equation 5.28:

$$\Delta \boldsymbol{U} = \left[\Delta \boldsymbol{u}^{k^T}, \Delta \boldsymbol{u}^{k+1^T}, \dots, \Delta \boldsymbol{u}^{k+N_c-1^T}\right]^T \leq \Delta \boldsymbol{U}^{max} \qquad (5.51)$$

$$-\Delta \boldsymbol{U} = -\left[\Delta \boldsymbol{u}^{k^T}, \Delta \boldsymbol{u}^{k+1^T}, \dots, \Delta \boldsymbol{u}^{k+N_c-1^T}\right]^T \leq -\Delta \boldsymbol{U}^{min} \qquad (5.52)$$

Where $\Delta \boldsymbol{U}^{max}$ and $\Delta \boldsymbol{U}^{min}$ are defined as:

$$\Delta \boldsymbol{U}^{max} = \begin{bmatrix} I_m \\ \vdots \\ I_m \end{bmatrix} \cdot \Delta \boldsymbol{u}^{max}, \quad \Delta \boldsymbol{U}^{min} = \begin{bmatrix} I_m \\ \vdots \\ I_m \end{bmatrix} \cdot \Delta \boldsymbol{u}^{min} \qquad (5.53)$$

Where I_m is the identity matrix with dimension *m* x *m.* Finally, 5.51 and 5.52 can be rewrite in a more compact form as follow:

$$\begin{bmatrix} -I_{m \cdot N_c} \\ I_{m \cdot N_c} \end{bmatrix} \Delta \boldsymbol{U} \leq \begin{bmatrix} -\Delta \boldsymbol{U}^{min} \\ \Delta \boldsymbol{U}^{max} \end{bmatrix} \qquad (5.54)$$

Constraints on the Amplitude of the Control Variable

These are the most encountered constraints among all constraint types, which are the physical hard constraints on the control actions. Considering a Multiple-Input system, with *m* control inputs, upper and lower limit on the amplitude of the control inputs can be defined as:

$$\boldsymbol{u}^{max} = [u_1^{max}, u_2^{max}, \dots, u_m^{max}]^T \quad (5.55)$$

$$\boldsymbol{u}^{min} = \left[u_1^{min}, u_2^{min}, \dots, u_m^{min}\right]^T \quad (5.56)$$

Assuming, that these limits are constant throughout the time span defined by the control horizon H_c, we can write:

$$\boldsymbol{U} = \left[\boldsymbol{u}^{k^T}, \boldsymbol{u}^{k+1^T}, \dots, \boldsymbol{u}^{k+N_c-1^T}\right]^T \leq \boldsymbol{U}^{max} \qquad (5.57)$$

$$-\boldsymbol{U} = -\left[\boldsymbol{u}^{k^T}, \boldsymbol{u}^{k+1^T}, \dots, \boldsymbol{u}^{k+N_c-1^T}\right]^T \leq -\boldsymbol{U}^{min} \qquad (5.58)$$

Where $\boldsymbol{U}^{max}$ and $\boldsymbol{U}^{min}$ are defined as:

$$\boldsymbol{U}^{max} = \begin{bmatrix} I_m \\ \vdots \\ I_m \end{bmatrix} \cdot \boldsymbol{u}^{max}, \quad \boldsymbol{U}^{min} = \begin{bmatrix} I_m \\ \vdots \\ I_m \end{bmatrix} \cdot \boldsymbol{u}^{min} \qquad (5.59)$$

Now we have to express equations 5.57 and 5.58 in terms of vector $\Delta \boldsymbol{U}$, which is the variable vector of the optimization problem. To do so, equation 5.12 can be rewritten as:

$$\Delta \boldsymbol{u}^k = \boldsymbol{u}^k - \boldsymbol{u}^{k-1} \quad (5.60)$$

Rearranging 5.60, following equations hold:

$$\begin{aligned} &\boldsymbol{u}^k = \boldsymbol{u}^{k-1} + \Delta \boldsymbol{u}^k \\ &\boldsymbol{u}^{k+1} = \boldsymbol{u}^{k-1} + \Delta \boldsymbol{u}^k + \Delta \boldsymbol{u}^{k+1} \\ &\vdots \\ &\boldsymbol{u}^{k+N_c-1} = \boldsymbol{u}^{k-1} + \Delta \boldsymbol{u}^k + \Delta \boldsymbol{u}^{k+1} + \cdots + \Delta \boldsymbol{u}^{k+N_c-1} \end{aligned} \quad (5.61)$$

Equations 5.61 in a more compact form becomes:

$$\boldsymbol{U} = \begin{bmatrix} \boldsymbol{u}^k \\ \boldsymbol{u}^{k+1} \\ \boldsymbol{u}^{k+2} \\ \vdots \\ \boldsymbol{u}^{k+N_c-1} \end{bmatrix} = \begin{bmatrix} I_m \\ I_m \\ I_m \\ \vdots \\ I_m \end{bmatrix} \cdot \boldsymbol{u}^{k-1} + \begin{bmatrix} I_m & 0 & 0 & \dots & 0 \\ I_m & I_m & 0 & \dots & 0 \\ I_m & I_m & I_m & \dots & 0 \\ \vdots & & & & \\ I_m & I_m & \dots & I_m & I_m \end{bmatrix} \cdot \begin{bmatrix} \Delta \boldsymbol{u}^k \\ \Delta \boldsymbol{u}^{k+1} \\ \Delta \boldsymbol{u}^{k+2} \\ \vdots \\ \Delta \boldsymbol{u}^{k+N_c-1} \end{bmatrix} \quad (5.62)$$

Re-writing 5.62 in a compact matrix form and repeating for the lower limits, with $\boldsymbol{C_1}$ and $\boldsymbol{C_2}$ corresponding to the appropriate matrices, constraints for the control actions are:

$$-\boldsymbol{C_2} \cdot \Delta \boldsymbol{U} \leq -\boldsymbol{U}^{min} + \boldsymbol{C_1} \cdot \boldsymbol{u}^{k-1} \quad (5.63)$$

$$\boldsymbol{C_2} \cdot \Delta \boldsymbol{U} \leq \boldsymbol{U}^{max} - \boldsymbol{C_1} \cdot \boldsymbol{u}^{k-1} \quad (5.64)$$

Output Constraints

Last type of constrain are those imposed on the output variables. As we have seen in Section 5.2.2, output variables across the prediction horizon can be formulated in terms of current state variable information $\boldsymbol{x}^k$ and the future control inputs $\Delta \boldsymbol{U}$; remembering definition 5.29 and equation 5.32 we can write:

$$\boldsymbol{Y} = \left[\boldsymbol{y}^{k+1^T}, \boldsymbol{y}^{k+2^T}, \dots, \boldsymbol{y}^{k+N_p{}^T}\right]^T, \ \boldsymbol{Y} \in \mathbb{R}^{(N_p \cdot \mathrm{q}) \times 1} \quad (5.65)$$

$$\boldsymbol{Y} = \boldsymbol{F} \cdot \boldsymbol{x}^k + \boldsymbol{\Phi} \cdot \Delta \boldsymbol{U} \quad (5.66)$$

As was assumed for the other constraint types, upper and lower bound values are assumed to be constant throughout the prediction horizon.

Output limits are defined as:

$$\boldsymbol{Y}^{max} = \begin{bmatrix} I_q \\ \vdots \\ I_q \end{bmatrix} \cdot \boldsymbol{y}^{max}, \quad \boldsymbol{Y}^{min} = \begin{bmatrix} I_q \\ \vdots \\ I_q \end{bmatrix} \cdot \boldsymbol{y}^{min} \qquad (5.67)$$

Where, according with dimension of the augmented model defined in section 5.2.1, matrix I_q is the identity matrix of dimension *q*, number of different outputs of the MIMO augmented model. Output constraints are often implemented as 'soft' constraints in the way that a slack variable $\mathbf{s_v} > 0$ is added to them. Main reason why we use a slack variable is because without it unsolvable optimization problem can occur, when any control action cannot produce output results within their limits. It is important to emphasize that when introducing a slack variable, the cost function must be modified by adding the latter in an attempt to minimize it. Outputs constrains are defined as follow:

$$\boldsymbol{Y}^{min} - \mathbf{s_v} \leq \boldsymbol{F}\,\boldsymbol{x}^k + \boldsymbol{\Phi}\,\Delta\boldsymbol{U} \leq \boldsymbol{Y}^{max} + \mathbf{s_v} \qquad (5.68)$$

Rearranging 5.68 to express inequality in terms of control variable $\Delta\boldsymbol{U}$, it becomes:

$$\begin{bmatrix} -\boldsymbol{\Phi} \\ \boldsymbol{\Phi} \end{bmatrix} \cdot \Delta\boldsymbol{U} \leq \begin{bmatrix} -\boldsymbol{Y}^{min} + \boldsymbol{F} \cdot \boldsymbol{x}^k + \mathbf{s_v} \\ \boldsymbol{Y}^{max} - \boldsymbol{F} \cdot \boldsymbol{x}^k + \mathbf{s_v} \end{bmatrix} \qquad (5.69)$$

Finally, putting all constrain equation previously obtained together, and expressing them in a compact form with control vector $\Delta\boldsymbol{U}$ as variable, we can write following relation:

$$\boldsymbol{M} \cdot \Delta\boldsymbol{U} \leq \boldsymbol{\gamma} \quad (5.70)$$

With:

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{M_1} \\ \boldsymbol{M_2} \\ \boldsymbol{M_3} \end{bmatrix}; \quad \boldsymbol{\gamma} = \begin{bmatrix} \boldsymbol{N_1} \\ \boldsymbol{N_2} \\ \boldsymbol{N_3} \end{bmatrix};$$

$$\boldsymbol{M_1} = \begin{bmatrix} -I_{m \cdot N_c} \\ I_{m \cdot N_c} \end{bmatrix}; \quad \boldsymbol{M_2} = \begin{bmatrix} -\boldsymbol{C_2} \\ \boldsymbol{C_2} \end{bmatrix}; \quad \boldsymbol{M_3} = \begin{bmatrix} -\boldsymbol{\Phi} \\ \boldsymbol{\Phi} \end{bmatrix};$$

$$\boldsymbol{N_1} = \begin{bmatrix} -\Delta\boldsymbol{U}^{min} \\ \Delta\boldsymbol{U}^{max} \end{bmatrix}; \quad \boldsymbol{N_2} = \begin{bmatrix} -\boldsymbol{U}^{min} + \boldsymbol{C_1} \cdot \boldsymbol{u}^{k-1} \\ \boldsymbol{U}^{max} - \boldsymbol{C_1} \cdot \boldsymbol{u}^{k-1} \end{bmatrix}; \quad \boldsymbol{N_3} = \begin{bmatrix} -\boldsymbol{Y}^{min} + \boldsymbol{F} \cdot \boldsymbol{x}^k + \mathbf{s_v} \\ \boldsymbol{Y}^{max} - \boldsymbol{F} \cdot \boldsymbol{x}^k + \mathbf{s_v} \end{bmatrix}$$

When the constraints are fully imposed, the number of constraints is equal to $4 \times m \times N_c + 2 \times q \times N_p$, where *m* is the number of inputs and *q* is the number of outputs. N_c and N_p are number of samples related to control and prediction horizons.

5.2.5.2 Numerical Solutions Using Quadratic Programming

In this section, once cost function and constrain have been defined, numerical solution of a constrained optimization problem will be discussed. Since cost function was defined as a quadratic function and constrains one was defined as linear, the problem which must be solved is a standard quadratic programming with linear constrains problem, which has been extensively studied in the literature because of its key role in obtaining the optimal solution of the constrained control problems.

To address the desired numerical solution, the cost function and the constraint relation (5.41 and 5.70), the two components that make up the constrained optimisation problem, are recalled:

$$J = \frac{1}{2}\Delta \boldsymbol{U}^T \cdot (\boldsymbol{\Phi}^T \cdot \boldsymbol{W_Y} \cdot \boldsymbol{\Phi} + \boldsymbol{W_U}) \cdot \Delta \boldsymbol{U} - \Delta \boldsymbol{U}^T \cdot \boldsymbol{\Phi}^T \cdot \boldsymbol{W_Y} \cdot \left(\boldsymbol{T_Y} - \boldsymbol{F} \cdot \boldsymbol{x}^k\right) \qquad (5.71)$$

$$\boldsymbol{M} \cdot \Delta \boldsymbol{U} \leq \boldsymbol{\gamma} \quad (5.72)$$

For sake of simplicity, the following definitions are made:

$$\boldsymbol{z} = \Delta \boldsymbol{U} \qquad (5.73)$$

$$\boldsymbol{E} = (\boldsymbol{\Phi}^T \cdot \boldsymbol{W_Y} \cdot \boldsymbol{\Phi} + \boldsymbol{W_U}) \qquad (5.74)$$

$$\boldsymbol{f} = -\boldsymbol{\Phi}^T \cdot \boldsymbol{W_Y} \cdot \left(\boldsymbol{T_Y} - \boldsymbol{F} \cdot \boldsymbol{x}^k\right) \qquad (5.75)$$

Where, without loss of generality, $\boldsymbol{E}$ is assumed to be symmetric and positive definite. From these definitions, the objective function J and the constraints are expressed as:

$$J(\boldsymbol{z}) = \frac{1}{2}\, \boldsymbol{z}^T \cdot \boldsymbol{E} \cdot \boldsymbol{z} + \boldsymbol{z}^T \cdot \boldsymbol{f} \qquad (5.76)$$

$$\boldsymbol{g}(\boldsymbol{z}) = \boldsymbol{M} \cdot \boldsymbol{z} - \boldsymbol{\gamma} \leq 0 \qquad (5.77)$$

Minimization with only Equality Constraints

In order to tackle the constrained optimisation problem with both equality and inequality constraints, we start by solving the simpler problem in which only the first type of mentioned constraints appear. In particular, we are going to find the minimum of a positive definite quadratic function with linear equality constraints.

Positive definite quadratic functions have their level surfaces as hyper-ellipsoids, while linear equality constrains define hyperplanes. The constrained minimum is located at the point of tangency between constrain boundaries and hyper-ellipsoids.

In other words, considering a single constraint $g(\mathbf{z})$, the gradient of $g(\boldsymbol{z})$, $\nabla g(\boldsymbol{z})$, is the direction perpendicular to the constraint, i.e. the direction in which one cannot move. If, at a certain point $\boldsymbol{z}^*$, this direction corresponds to the direction $-\nabla J(\boldsymbol{z})$, i.e., the direction one would like to travel in order to go towards the minimum of J, then $\boldsymbol{z}^*$ is a constrained minimum point. In mathematical terms, and considering the possibility of having several constraints ($\boldsymbol{g}(\mathbf{z})$ is a vectorial function), the conditions that must be fulfilled to have a constrained minimum point are:

$$-\nabla J(\boldsymbol{z}) = \sum_{k=1}^{C} \lambda_k \nabla g_k(\boldsymbol{z}) \quad \Leftrightarrow \quad \nabla J(\boldsymbol{z}) + \sum_{k=1}^{C} \lambda_k \nabla g_k(\boldsymbol{z}) = 0 \qquad (5.78)$$

Where λ_k are arbitrary scalar values. Substituting equations 5.76 and 5.77 in 5.78 we obtain:

$$\boldsymbol{E}\boldsymbol{z} + \boldsymbol{f} + \boldsymbol{M}^T\boldsymbol{\lambda} = 0 \qquad (5.79)$$

Where $\boldsymbol{\lambda}$ is a vector containing C scalar values. Equation 5.79, with constraint equation $\boldsymbol{M} \cdot \boldsymbol{z} - \boldsymbol{\gamma} = 0$, represent two necessary equations to solve the problem. To deal with this problem, it is convenient to introduce the method of Lagrange multipliers, where the basic idea is to convert the constrained problem defined by these two relations to an unconstrained one in which the so called Lagrangian function must be minimized. It is defined as:

$$\mathcal{L}(\boldsymbol{z}, \boldsymbol{\lambda}) = \frac{1}{2}\, \boldsymbol{z}^T \boldsymbol{E}\, \boldsymbol{z} + \boldsymbol{z}^T \cdot \boldsymbol{f} + \boldsymbol{\lambda}^T (\boldsymbol{M}\, \boldsymbol{z} - \boldsymbol{\gamma}) \qquad (5.80)$$

Now the coefficients λ_k, components of $\boldsymbol{\lambda}$**,** are called Lagrange multipliers. By imposing the gradient of Lagrangian function equal to zero, ($\nabla \mathcal{L}(\boldsymbol{z}, \boldsymbol{\lambda}) = 0$), we arrive at the same equations as above:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{z}} = \boldsymbol{E}\boldsymbol{z} + \boldsymbol{f} + \boldsymbol{M}^T\boldsymbol{\lambda} = 0 \quad (5.81)$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}} = \boldsymbol{M}\, \boldsymbol{z} - \boldsymbol{\gamma} = 0 \qquad (5.82)$$

Values of $\boldsymbol{z}$ and $\boldsymbol{\lambda}$ that minimize the Lagrangian function (so, those which resolve linear equations set composed by 5.81 and 5.82) are:

$$\boldsymbol{\lambda} = -\left(\boldsymbol{M}\boldsymbol{E}^{-1}\boldsymbol{M}^{\boldsymbol{T}}\right)^{-1}\left(\boldsymbol{\gamma} + \boldsymbol{M}\boldsymbol{E}^{-1}\boldsymbol{f}\right) \quad (5.83)$$

$$\boldsymbol{z} = -\boldsymbol{E}^{-1}\left(\boldsymbol{M}^{\boldsymbol{T}}\boldsymbol{\lambda} + \boldsymbol{f}\right) \qquad (5.84)$$

The number of equality constraints must be less than or equal to the number of decision variables (dimension of $\boldsymbol{z}$). If the number of equality constraints equals the number of decision variables, the only feasible solution is the one that satisfies the constraints and there is no additional variable in $\boldsymbol{z}$ that can be used to optimize the original objective function. Finally, if the number of constrain is greater than the dimension of $\boldsymbol{z}$ there is no feasible solution.

Minimization with Inequality Constraints

Inequality constraints $\boldsymbol{M \cdot z \leq \gamma}$ can include active constraints and inactive constraints. An inequality constrain $\boldsymbol{M_i \cdot z \leq \gamma_i}$ (where $\boldsymbol{M_i}$ is the i-row of $\boldsymbol{M}$ and γ_i is the i-element of $\boldsymbol{\gamma}$) is an active one if $\boldsymbol{M_i \cdot z = \gamma_i}$, whereas it is said inactive if $\boldsymbol{M_i \cdot z < \gamma_i}$. To deal with an optimization problem with both inequality and equality constrains (or with only inequality ones) the so-called Karush-Kuhn-Tucker (KKT) conditions are used. They are a generalization of Lagrange multipliers, and they give a set of necessary conditions which solve this type of optimization problem. KKT conditions are:

$$\boldsymbol{Ez + f + M^T \lambda = 0} \qquad (5.85)$$

$$\boldsymbol{M\,z\, - \,\gamma \leq\, 0} \qquad (5.86)$$

$$\boldsymbol{\lambda^T (M\,z\, - \,\gamma) = 0} \qquad (5.87)$$

$$\boldsymbol{\lambda \geq 0} \qquad (5.88)$$

Where the vector **λ** contains the Lagrange multipliers. A constraint is active when the unconstrained minimum of cost function is in the not permitted region related to that constrain, in this case related component of **λ** is a positive number. On the other hand, a constrain is inactive when global minimum is in the permitted region, in this case related component of **λ** is zero. As can be seen in 5.85-5.88 in both cases KKT are met.

These conditions can be expressed in a simpler form in terms of the set of active constraints. Let S_{act} indicates the index set of active constraints. Then the KKT conditions become:

$$\boldsymbol{Ez + f + \sum_{k \in S_{act}} \lambda_k M_k^{\,T} = 0} \qquad (5.89)$$

$$\boldsymbol{M_i\,z\, - \,\gamma_i = 0} \quad if\ k \in S_{act} \qquad (5.90)$$

$$\boldsymbol{M_i\,z\, - \,\gamma_i < 0} \quad if\ k \notin S_{act} \qquad (5.91)$$

$$\lambda_k > 0 \quad if\ k \in S_{act} \qquad (5.92)$$

$$\lambda_k = 0 \quad if\ k \notin S_{act} \qquad (5.93)$$

If the active set were known, the original problem could be replaced by the corresponding problem having related equality constraints only, since inactive constrains do not affect the search for the minimum.

The algorithms involved in solving this type of optimisation problem have the task of working out which constraints are active, so that inactive constraints can be neglected, and the constrained minimum point obtained as in the case of constrained optimisation where only equality constraints appear. On the next page, two of the most common algorithms used within MPC controllers will be presented, after which a commonly used procedure exploiting one of the two algorithms will be described. It will also be used in the wind turbine MPC controller that will be implemented in the next chapter.

Active Set Method

The main idea of active set methods is to define, at each iteration of an algorithm, a set of constraints that is treated as the active set. The active set is chosen to be a subset of the constraints that are active at the previous step (at the first iteration all constrain are considered active). The algorithm then proceeds to move on the surface defined by the working set of constraints to find a constrained minimum, so at each step of the active set method, an equality constrained problem is solved.

If all the active set's Lagrange multipliers $\lambda_{act} \geq 0$, then the point is a local solution to the original problem. If, on the other hand, there exists a $\lambda_i < 0$, then the objective function value can be decreased by relaxing the constraint i (i.e., deleting it from the constraint equation). During minimization, it is necessary to monitor the values of the other constraints to be sure that they are not violated, since all points defined by the algorithm must be feasible. It often happens that while moving on the working surface, a new constraint boundary is encountered. It is necessary to add this constraint to the working set, then proceed to the re-defined working surface.

The algorithm proceeds until the actual active constrains are found, checking that all inequalities of inactive set are verified. This type of algorithm does not work well if there are a lot of constraints, in which case calculation time can become very long, reason why it may not be suitable for real-time calculations, a typical situation of MPC controllers.

Primal-Dual Method

Primal-dual method derives from Lagrangian duality theory, through which an optimization problem can be solved. The basic idea is to solve a problem, called primal problem, by looking at a different optimization problem, the dual problem. More specifically, the solution of the dual problem can provide the same optimal solution as the primal problem, which is useful if the primal problem is harder to solve than the dual one.

In this case, the primal problem is the minimization of cost the function such that some constrains are satisfied:

$$min \quad J(\boldsymbol{z}) = \frac{1}{2}\, \boldsymbol{z}^T \cdot \boldsymbol{E} \cdot \boldsymbol{z} + \boldsymbol{z}^T \cdot \boldsymbol{f}$$

$$s.t. \quad \boldsymbol{g}(\boldsymbol{z}) = \boldsymbol{M} \cdot \boldsymbol{z} - \boldsymbol{\gamma} \leq 0 \qquad (5.94)$$

Let the optimum of this problem be $\boldsymbol{z}^*$. As seen previously, the Lagrangian function is defined as:

$$\mathcal{L}(\boldsymbol{z}, \boldsymbol{\lambda}) = J(\boldsymbol{z}) + \boldsymbol{\lambda}^T \boldsymbol{g}(\boldsymbol{z}) \qquad (5.95)$$

The Lagrange dual function is defined to be:

$$q(\boldsymbol{\lambda}) = \min_{\boldsymbol{z}} \mathcal{L}(\boldsymbol{z}, \boldsymbol{\lambda}) \qquad (5.96)$$

Where minimization 5.96 is unconstrained.

For $\boldsymbol{\lambda} \geq 0$ and for any $\boldsymbol{z}$ that obeys the constrain $\boldsymbol{g}(\boldsymbol{z}) \leq 0$, generally called $\boldsymbol{z'}$, following relation holds:

$$\mathcal{L}(\boldsymbol{z}', \boldsymbol{\lambda}) \leq J(\boldsymbol{z}') \quad (5.97)$$

Thus, in particular, if $\boldsymbol{z^*}$ is the optimum of the original problem:

$$q(\boldsymbol{\lambda}) = \min_{\boldsymbol{z}} \mathcal{L}(\boldsymbol{z}, \boldsymbol{\lambda}) \leq \mathcal{L}(\boldsymbol{z}', \boldsymbol{\lambda}) \leq J(\boldsymbol{z}^*) \quad (5.98)$$

So, it is natural to look for the highest value of $q(\boldsymbol{\lambda})$, along $\boldsymbol{\lambda} \geq 0$. This is the Lagrange dual problem:

$$\begin{aligned} &max \quad q(\boldsymbol{\lambda}) \\ &s.t. \quad \boldsymbol{\lambda} \geq 0 \end{aligned} \quad (5.99)$$

Dual problem can also be written as a minimization problem changing sign of argument:

$$\begin{aligned} &min \quad -q(\boldsymbol{\lambda}) \\ &s.t. \quad \boldsymbol{\lambda} \geq 0 \end{aligned} \quad (5.100)$$

Remembering that:

$$q(\boldsymbol{\lambda}) = \min_{\boldsymbol{z}} \left[\mathcal{L}(\boldsymbol{z}, \boldsymbol{\lambda}) = \tfrac{1}{2}\, \boldsymbol{z}^T \boldsymbol{E}\, \boldsymbol{z} + \boldsymbol{z}^T \cdot \boldsymbol{f} + \boldsymbol{\lambda}^T (\boldsymbol{M}\, \boldsymbol{z} - \boldsymbol{\gamma}) \right] \quad (5.101)$$

And that this minimization is unconstrained, we can find values of $\boldsymbol{z}$, depending on $\boldsymbol{\lambda}$, that minimize the Lagrangian function $\mathcal{L}$:

$$\boldsymbol{z} = -\boldsymbol{E^{-1}} (\boldsymbol{M^T \lambda} + \boldsymbol{f}) \quad (5.102)$$

Substituting 5.102 in 5.101, the dual problem 5.100 becomes:

$$\begin{aligned} &min \quad \frac{1}{2}\, \boldsymbol{\lambda}^T\, \boldsymbol{H}\, \boldsymbol{\lambda} + \boldsymbol{\lambda}^T \boldsymbol{K} + \frac{1}{2}\, \boldsymbol{\gamma}^T\, \boldsymbol{E^{-1}}\, \boldsymbol{\gamma} \\ &s.t. \quad \boldsymbol{\lambda} \geq 0 \end{aligned} \quad (5.103)$$

Where matrices $\boldsymbol{H}$ and $\boldsymbol{K}$ are given by:

$$\boldsymbol{H} = \boldsymbol{M E^{-1} M^T} \quad (5.104)$$

$$\boldsymbol{K} = \boldsymbol{\gamma} + \boldsymbol{M E^{-1} f} \quad (5.105)$$

Resolving 5.103 values of $\boldsymbol{\lambda}$ can be found, those which are negative are set to zero and those which are positive become the active ones, through which, from equation 5.102, optimization problem can be solved.

Hildreth's Quadratic Procedure

In this section a simple procedure to solve the primal-dual problem, described above, can be solved. This algorithm is called Hildreth's quadratic procedure [29], [30]. In this algorithm vector of Lagrangian multipliers is computed with an iterative procedure, finding it element by element through the primal-dual method. Thanks to an element-by-element search it does not require any matrix inversion.

Looking at the dual problem 5.103, we see that the last term does not depend on the variable $\boldsymbol{\lambda}$, so it reduces to the following minimization:

$$\min_{\boldsymbol{\lambda}\geq 0} \ \frac{1}{2}\,\boldsymbol{\lambda}^T\,\boldsymbol{H}\,\boldsymbol{\lambda} + \boldsymbol{\lambda}^T\boldsymbol{K} \qquad (5.106)$$

Whose solution is:

$$\boldsymbol{H}\,\boldsymbol{\lambda} + \boldsymbol{K} = 0 \quad \Leftrightarrow \quad \boldsymbol{\lambda} = -\boldsymbol{H}^{-1}\boldsymbol{K} \qquad (5.107)$$

A generic i-row of 5.107 can be write as follows:

$$\sum_{j=1}^{C} H_{ij}\cdot\lambda_j + K_i = 0 \quad \Leftrightarrow \quad \sum_{\substack{j=1\\ j\neq k}}^{C} H_{ij}\cdot\lambda_j + H_{ik}\cdot\lambda_k + K_i = 0 \qquad (5.108)$$

Where C is the number of constraints, that is the dimension of $\boldsymbol{\lambda}$**.** From 5.108, by using the k-row to calculate λ_k, we can write:

$$\lambda_k = -\frac{1}{H_{kk}}\left[\sum_{\substack{j=1\\ j\neq k}}^{C} H_{kj}\cdot\lambda_j + K_k\right] \qquad (5.109)$$

Finally, Hildreth's procedure can be started from choosing a first attempt $\boldsymbol{\lambda}$ vector, with all null elements, then an iterative procedure starts, when at each iteration *l* all element of vector $\boldsymbol{\lambda}^{l+1}$ are calculated from vector computed at previous step, $\boldsymbol{\lambda}^{l}$, through following expression (if computed value is negative, it is substituted with 0, since it indicates that corresponding constrain is inactive):

$$\lambda_k^{l+1} = max\left(0\,, -\frac{1}{H_{kk}}\left[\sum_{j=1}^{k-1} H_{kj}\lambda_j^{l+1} + \sum_{j=k+1}^{C} H_{kj}\lambda_j^{l} + K_k\right]\right) \qquad (5.110)$$

The iterative procedure continues until a converged vector $\boldsymbol{\lambda}^*$ is obtained, with null component for inactive constrain. Finally, $\boldsymbol{\lambda}^*$ is used to compute optimum $\boldsymbol{z}$, which corresponds to the control variable vector that leads to a constrained minimization of cost function ***J(z)***. To do so, following equation is exploited:

$$\boldsymbol{z} = -\boldsymbol{E}^{-1}\left(\boldsymbol{f} + \boldsymbol{M}^{\boldsymbol{T}}\boldsymbol{\lambda}^*\right) \qquad (5.111)$$

5.2.6 State Estimation

MPC controllers are based on knowledge of the system states and inputs at each instant. This is only possible in ideal cases in which all inputs are known, and the system model perfectly describes the real behaviour of the system, or in cases in which every state of the system can be measured using error-free measuring instruments. In most cases, however, some states of the system cannot be directly measurable and those that are measured, corresponding to the outputs of the system, are affected by measurement errors; in addition, some input may be unknown.

In order to cope with all these problems, it is therefore often necessary to resort to algorithms by means of which it is possible to estimate the value of the states of a system. They work thanks to its mathematical model, the measurements taken by sensors and the knowledge of certain information regarding the measurements of the outputs and the disturbances acting on the system.

These types of algorithms are called observers, and, in this section, we will discuss one in particular, probably the most commonly used and most famous, called Kalman Filter. It can also be used to obtain an estimate of a certain signal measured by non-ideal sensors by exploiting the model of the system. This is why the term filter is used, while Kalman derives from the name of the author who proposed it. There are different types of Kalman filter, which can deal with linear or non-linear, discrete or continuous ones. In the following we will study the simplest of them, a discrete Kalman filter based on a linear time-invariant system model.

It is important to specify that the system model used by the observer may be different from the system model used by the MPC controller for numerical optimisation, the one introduced in section 5.2.1. To give one example, if an input of the controlled plant is unknown, an observer can be used to estimate it. The observer will exploit the model of a system in which this quantity is an unknown state, which will be then estimated.

5.2.6.1 Discrete Kalman Filter

Model

In this section, the algorithm for a discrete linear Kalman filter will be derived. As mentioned above, it utilises the linear model of a certain system whose states are to be known; it is expressed in the discrete state space form, and it is defined as follows:

$$\begin{cases} \boldsymbol{x_{k+1}} = \boldsymbol{A}\,\boldsymbol{x_k} + \boldsymbol{B}\,\boldsymbol{u_k} + \boldsymbol{G}\,\boldsymbol{w_k} \\ \boldsymbol{y_k} = \boldsymbol{C}\,\boldsymbol{x_k} + \boldsymbol{v_k} \end{cases} \qquad (5.112)$$

Where:

- $\boldsymbol{x}$**:** states vector**,** $\boldsymbol{x} \in R^{n\,x\,1}$**;**
- $\boldsymbol{u}$**:** inputs vector**,** $\boldsymbol{u} \in R^{m\,x\,1}$**,** known by assumption;
- $\boldsymbol{w}$**:** process noise vector**,** $\boldsymbol{w} \in R^{d\,x\,1}$. It is a vector whose component are inputs of the system whose value is normally unknown. In the Kalman filter algorithm, process noise is assumed to be a random process, that is a white noise with zero mean and a certain value of variance;
- $\boldsymbol{y}$**:** outputs vector**,** $\boldsymbol{y} \in R^{q\,x\,1}$. The outputs are measured quantities, whose measure is composed by a deterministic part ($\boldsymbol{C}\,\boldsymbol{x}$) and stochastic one due to sensor noise;

- $\boldsymbol{v}$: measurement noise vector, $\boldsymbol{v} \in R^{q\,x\,1}$. It is the stochastic part of the outputs' measurement. As process noise, in the Kalman filter algorithm measurements noise is assumed to be a white noise with zero mean and a certain value of variance;

- $\boldsymbol{A}$: state matrix, $\boldsymbol{A} \in R^{n\,x\,n}$;

- $\boldsymbol{B}$: input matrix, $\boldsymbol{B} \in R^{n\,x\,m}$;

- $\boldsymbol{G}$: process noise matrix, $\boldsymbol{G} \in R^{n\,x\,d}$;

- $\boldsymbol{C}$: output matrix, $\boldsymbol{C} \in R^{q\,x\,n}$;

- T: sample time of discrete model, that is, as an example $\boldsymbol{x_k} = \boldsymbol{x}(kT)$.

Process and Measurement Noises characteristics

The other information required for the execution of the algorithm is statistical information regarding process noise and measurements noise. Since, by definition, they are white noise, their expected value, at each instant of time, is zero:

$$E[\boldsymbol{w}_k] = 0, \quad E[\boldsymbol{v}_k] = 0 \qquad (5.113)$$

Where $E[\boldsymbol{w}_k]$ indicates the expected value of vector $\boldsymbol{w}_k$. Each i-th component of $E[\boldsymbol{w}_k]$ is defined as: $E[w_{k_i}] = \sum_n p_n\, w_{k_{i_n}}$, with p_n the probability correlated with $w_{k_{i_n}}$, that is the probability that at time k, the i-th component of $\boldsymbol{w}_k$ is $w_{k_{i_n}}$. The same is valid for $\boldsymbol{v}_k$.

Given two vectors, for example $\boldsymbol{w}_k$ and $\boldsymbol{w}_s$, the covariance matrix is defined as the matrix $\boldsymbol{Co}(\boldsymbol{w}_k,\boldsymbol{w}_s)$, whose *ij-th* component is:

$$\boldsymbol{Co}(\boldsymbol{w}_k,\boldsymbol{w}_s)_{\,ij} = E\left[(w_{k_i} - E[w_{k_i}]) \cdot (w_{s_j} - E\left[w_{s_j}\right])\right] \qquad (5.114)$$

$\boldsymbol{Co}(\boldsymbol{w}_k,\boldsymbol{w}_s)_{\,ij}$ is the covariance between w_{k_i} and w_{s_j}, which is strictly linked with the correlation between them. If the correlation is zero, it means that the variation of one value is not influenced by the other and vice versa, while a positive correlation means that the probability of having a positive variation of one of the two quantities is greater if there is a positive variation of the other, vice versa for negative values of covariance.

In matrix form, covariance matrix $\boldsymbol{Co}(\boldsymbol{w}_k,\boldsymbol{w}_s)$ is defined as:

$$\boldsymbol{Co}(\boldsymbol{w}_k,\boldsymbol{w}_s) = E[(\boldsymbol{w}_k - E[\boldsymbol{w}_k]) \cdot (\boldsymbol{w}_s - E[\boldsymbol{w}_s])^T] \qquad (5.115)$$

Subscripts *k* and *s* indicate two different time instants. If two vector involved are coincident, the matrix $\boldsymbol{Co}$ is called auto-covariance matrix.

We assume there is no correlation between process noise and measurements one, also considering the same instant, since they are two white noises. From these considerations we can write:

$$\boldsymbol{Co}(\boldsymbol{w}_k, \boldsymbol{v}_s) = 0, \quad \forall\, k, s \qquad (5.116)$$

Moreover, by definition of white noise, signals at different time are uncorrelated, hence:

$$\boldsymbol{Co}(\boldsymbol{w}_k, \boldsymbol{w}_s) = 0, \quad if \quad k \neq s \qquad (5.117)$$

$$\boldsymbol{Co}(\boldsymbol{v}_k, \boldsymbol{v}_s) = 0, \quad if \quad k \neq s \qquad (5.118)$$

If, on the other hand, $k = s$, that is we are considering the same instant, non-zero covariances may exist. In this regard, we define:

$$\boldsymbol{R}_k = \boldsymbol{Co}(\boldsymbol{v}_k, \boldsymbol{v}_k), \quad \boldsymbol{R}_k \in R^{q\,x\,q} \qquad (5.119)$$

$$\boldsymbol{Q}_k = \boldsymbol{Co}(\boldsymbol{w}_k, \boldsymbol{w}_k), \quad \boldsymbol{Q}_k \in R^{d\,x\,d} \qquad (5.120)$$

$\boldsymbol{R}_k$ is the covariance matrix of $\boldsymbol{v}_k$, its diagonal contains variances of $\boldsymbol{v}_k$, while extra-diagonal terms are the covariances between different components of that vector. In Kalman filter matrices $\boldsymbol{R}_k$ and $\boldsymbol{Q}_k$ are considered time-invariant, for this reason they will be indicated simply by $\boldsymbol{R}$ and $\boldsymbol{Q}$.

Definitions

Before deriving the algorithm, it is necessary to make a few definitions; they generally apply to all quantities. As an example, for the state variable $\boldsymbol{x}$, the definitions are:

- $\boldsymbol{x}_{k|k-1}$: Subscript "$k \mid k-1$" indicates a variable at time k computed with information available at time $k-1$;

- $\boldsymbol{x}_k$: Subscript "k" indicates a variable at time k computed with information available at time k;

- $\hat{\boldsymbol{x}}$ **:** The hat indicates the expected value of a certain variable: $\hat{\boldsymbol{x}} = E[\boldsymbol{x}]$;

Then two errors are defined, both of which are the difference between the expected value of the vector x at a certain instant k and its actual value. The first error, called *'a priori'* error, $\boldsymbol{\varepsilon_k}$, considers the expected value calculated with the information available at time $k-1$, while the second, called *'a posteriori'* error, $\boldsymbol{e_k}$, uses the information up to time k. According with above definition, we can write:

$$\boldsymbol{\varepsilon_k} = \hat{\boldsymbol{x}}_{\boldsymbol{k|k-1}} - \boldsymbol{x_k} \qquad (5.121)$$

$$\boldsymbol{e_k} = \hat{\boldsymbol{x}}_{\boldsymbol{k}} - \boldsymbol{x_k} \qquad (5.122)$$

Kalman Filter Derivation

Basic idea of Kalman filter is to obtain the estimation of state vector at time k, $\hat{\boldsymbol{x}}_k$, through a correction of the value predicted by the model of the filter, $\hat{\boldsymbol{x}}_{k|k-1}$, which is made thanks to the measurement $\boldsymbol{y_k}$. Estimated value also depends on matrices $\boldsymbol{R}$ and $\boldsymbol{Q}$ and information at time $k-1$. A common form of correction is:

$$\hat{\boldsymbol{x}}_{\boldsymbol{k}} = \hat{\boldsymbol{x}}_{\boldsymbol{k|k-1}} + \boldsymbol{K_k}\,(\boldsymbol{y_k} - \hat{\boldsymbol{y}}_{\boldsymbol{k}}) \quad (5.123)$$

Where, by definition $\hat{\boldsymbol{y}}_{\boldsymbol{k|k-1}} = \boldsymbol{C}\,\hat{\boldsymbol{x}}_{\boldsymbol{k|k-1}}$ and $\hat{\boldsymbol{x}}_{\boldsymbol{k|k-1}} = \boldsymbol{E}[\boldsymbol{A}\,\boldsymbol{x_{k-1}} + \boldsymbol{B}\,\boldsymbol{u_{k-1}} + \boldsymbol{G}\,\boldsymbol{w_{k-1}}] = \boldsymbol{A}\,\hat{\boldsymbol{x}}_{\boldsymbol{k-1}} + \boldsymbol{B}\,\boldsymbol{u_{k-1}}$.

$\boldsymbol{K_k}$ is the Kalman gain matrix which is needed to estimate $\hat{\boldsymbol{x}}_{\boldsymbol{k}}$. It is calculated as the value that minimize the square sum expected value of *'a posteriori'* error components, called $J = \boldsymbol{E}[\boldsymbol{e_k^T e_k}]$. The *'a posteriori'* error, according with 5.121 and 5.122, is:

$$\begin{aligned} \boldsymbol{e_k} &= \hat{\boldsymbol{x}}_{\boldsymbol{k}} - \boldsymbol{x_k} = \hat{\boldsymbol{x}}_{\boldsymbol{k|k-1}} + \boldsymbol{K_k}(\boldsymbol{y_k} - \boldsymbol{C}\,\hat{\boldsymbol{x}}_{\boldsymbol{k|k-1}}) - \boldsymbol{x_k} \\ &= \hat{\boldsymbol{x}}_{\boldsymbol{k|k-1}} + \boldsymbol{K_k}(\boldsymbol{C}\,\boldsymbol{x_k} + \boldsymbol{v_k} - \boldsymbol{C}\,\hat{\boldsymbol{x}}_{\boldsymbol{k|k-1}}) - \boldsymbol{x_k} \\ &= (\boldsymbol{I} - \boldsymbol{K_k C})\boldsymbol{\varepsilon_k} + \boldsymbol{K_k v_k} \end{aligned} \quad (5.124)$$

Thus:

$$\begin{aligned} \boldsymbol{e_k^T e_k} &= [\boldsymbol{\varepsilon_k} - \boldsymbol{K_k C \varepsilon_k} + \boldsymbol{K_k v_k}]^T [\boldsymbol{\varepsilon_k} - \boldsymbol{K_k C \varepsilon_k} + \boldsymbol{K_k v_k}] \\ &= \boldsymbol{\varepsilon_k^T \varepsilon_k} + \boldsymbol{\varepsilon_k}^T \boldsymbol{C}^T \boldsymbol{K_k}^T \boldsymbol{K_k C \varepsilon_k} - \boldsymbol{2\varepsilon_k}^T \boldsymbol{K_k C \varepsilon_k} + \boldsymbol{2\varepsilon_k}^T \boldsymbol{K_k v_k} \\ &\quad \boldsymbol{-2v_k}^T \boldsymbol{K_k}^T \boldsymbol{K_k C \varepsilon_k} + \boldsymbol{v_k}^T \boldsymbol{K_k}^T \boldsymbol{K_k v_k} \end{aligned} \quad (5.125)$$

Taking the derivative of $\boldsymbol{e_k^T e_k}$ with respect to gain $\boldsymbol{K_k}$, and omitting subscript k for simplicity (from this point, an omitted subscript means there should be a *k*):

$$\begin{aligned} \frac{d}{d\boldsymbol{K}}(\boldsymbol{e^T e}) &= \frac{d}{\boldsymbol{dK}}[\boldsymbol{\varepsilon^T C^T K^T K\, C\varepsilon} - 2\boldsymbol{v^T K^T K C \varepsilon} + \boldsymbol{v^T K^T K v}] + 2\frac{d}{\boldsymbol{dK}}[\boldsymbol{\varepsilon^T K v} - \boldsymbol{\varepsilon^T K C \varepsilon}] \\ &= 2\boldsymbol{K C \varepsilon \varepsilon^T C^T} - 2[\boldsymbol{K C \varepsilon v^T} + \boldsymbol{K v \varepsilon^T C^T}] + 2\boldsymbol{K v v^T} + 2[\boldsymbol{\varepsilon v^T} - \boldsymbol{\varepsilon \varepsilon^T C^T}] \end{aligned} \quad (5.126)$$

Since $\frac{d}{d\boldsymbol{K_k}}(\boldsymbol{E}[\boldsymbol{e_k^T e_k}]) = \boldsymbol{E}[\frac{d}{d\boldsymbol{K_k}}(\boldsymbol{e_k^T e_k})]$ and $J = \boldsymbol{E}[\boldsymbol{e_k^T e_k}]$ we can write:

$$\frac{dJ}{d\boldsymbol{K_k}} = \boldsymbol{E}\left[\, 2\boldsymbol{K C \varepsilon \varepsilon^T C^T} - 2[\boldsymbol{K C \varepsilon v^T} + \boldsymbol{K v \varepsilon^T C^T}] + 2\boldsymbol{K v v^T} + 2[\boldsymbol{\varepsilon v^T} - \boldsymbol{\varepsilon \varepsilon^T C^T}]\right] \quad (5.127)$$

Then, $\boldsymbol{K_k}$ such that $\frac{dJ}{d\boldsymbol{K_k}} = 0$ is the optimal gain we are looking for because it minimizes norm of *'a posteriori'* error.

Thus, the following equation must be solved:

$$\boldsymbol{E}[\,\boldsymbol{KC\varepsilon\varepsilon^T C^T - KC\varepsilon v^T - Kv\varepsilon^T C^T + Kvv^T + \varepsilon v^T - \varepsilon\varepsilon^T C^T}] = 0 \qquad (5.128)$$

Thanks to linearity of expectation value operator:

$$\boldsymbol{KC\,E[\varepsilon\varepsilon^T]C^T - KC\,E[\varepsilon v^T] - K\,E[v\varepsilon^T]C^T + K\,E[vv^T] + E[\varepsilon v^T] - E[\varepsilon\varepsilon^T]C^T} = 0 \qquad (5.129)$$

We note that following relations hold:

$$\boldsymbol{E[v\varepsilon^T] = E[\varepsilon\, v^T] = E[(\hat{x}_{k|k-1} - x)\, v^T] = E[\hat{x}_{t|t-1}\, v^T] - E[x\, v^T]} = 0 \qquad (5.130)$$

Now we define the matrix $\boldsymbol{P_{k|k-1}}$ as the auto-covariance matrix of the *'a priori'* error at time *k*, $\boldsymbol{\varepsilon}$:

$$\boldsymbol{P_{k|k-1} = E[\varepsilon\varepsilon^T]} \qquad (5.131)$$

With 5.119, 5.130 and 5.131, the equation 5.129 becomes:

$$\boldsymbol{KCP_{k|k-1}C^T + KR - P_{k|k-1}C^T = 0} \qquad (5.132)$$

From 5.132, we obtain Kalman gain (at time k) expression**:**

$$\boldsymbol{K = P_{k|k-1}C^T\left[C\,P_{k|k-1}C^T + R\right]^{-1}} \qquad (5.133)$$

Only unknown quantity in equation 5.133 is the covariance matrix of *'a priori'* error, $\boldsymbol{P_{k|k-1}}$. To compute it we define $\boldsymbol{P_k}$, the covariance matrix of *'a posteriori error'*, as:

$$\boldsymbol{P_k = E[e\, e^T] = E[(\hat{x}_k - x_k)(\hat{x}_k - x_k)^T]} \qquad (5.134)$$

Substituting 5.124 in 5.134 and remembering 5.130, we have:

$$\begin{aligned}
\boldsymbol{P_k} &= \boldsymbol{E}[\,((\boldsymbol{I - KC})\boldsymbol{\varepsilon} + \boldsymbol{Kv})((\boldsymbol{I - KC})\boldsymbol{\varepsilon} + \boldsymbol{Kv})^T\,] \\
&= \boldsymbol{E}[\,(\boldsymbol{I - KH})\boldsymbol{\varepsilon\varepsilon^T}(\boldsymbol{I - KH})^T] + \boldsymbol{E}[(\boldsymbol{I - KH})\boldsymbol{\varepsilon v^T K^T}\,] + \boldsymbol{E}[\boldsymbol{Kv\varepsilon^T}(\boldsymbol{I - KH})^T] + \boldsymbol{E}[\boldsymbol{Kvv^T K^T}] \qquad (5.135) \\
&= (\boldsymbol{I - KH})\boldsymbol{E}[\boldsymbol{\varepsilon\varepsilon^T}](\boldsymbol{I - KH})^T + \boldsymbol{KE}[\boldsymbol{vv^T}]\boldsymbol{K^T} \\
&= (\boldsymbol{I - KC})\boldsymbol{P_{k|k-1}}(\boldsymbol{I - KC})^T + \boldsymbol{KRK^T}
\end{aligned}$$

Finally, putting 5.133 in 1.135, following relation can be written:

$$\boldsymbol{P_k = (I - KC)P_{k|k-1}} \qquad (5.136)$$

In order to compute $\boldsymbol{P_{k|k-1}}$, we note that $\boldsymbol{P_{k|k-1}}$ at time k is the same as $\boldsymbol{P_{k+1|k}}$ calculated at previous step. To do so, we note that:

$$\begin{aligned}\boldsymbol{\varepsilon_{k+1}} &= \hat{\boldsymbol{x}}_{k+1} - \boldsymbol{x}_{k+1} \\ &= \boldsymbol{A}\,\hat{\boldsymbol{x}}_k + \mathbf{B}\,\boldsymbol{u}_k - (\mathbf{A}\,\boldsymbol{x}_k + \mathbf{B}\,\boldsymbol{u}_k + \mathbf{G}\boldsymbol{w}_k) = \boldsymbol{A}\,\mathbf{e}_k - \mathbf{G}\,\boldsymbol{w}_k \qquad (5.137)\end{aligned}$$

$$\begin{aligned}\boldsymbol{P_{k+1|k}} &= \boldsymbol{E}\left[\boldsymbol{\varepsilon_{k+1}}\boldsymbol{\varepsilon_{k+1}^T}\right] = \boldsymbol{E}[(\boldsymbol{A}\,\mathbf{e}_k - \mathbf{G}\,\boldsymbol{w}_k)(\boldsymbol{A}\,\mathbf{e}_k - \mathbf{G}\,\boldsymbol{w}_k)^T] = \\ &= \boldsymbol{AE}\left[\boldsymbol{e_k e_k^T}\right]\boldsymbol{A^T} - \mathbf{G}\boldsymbol{E}\left[\boldsymbol{w_k e_k^T}\right]\boldsymbol{A^T} - \mathbf{A}\boldsymbol{E}\left[\boldsymbol{e_k w_k^T}\right]\boldsymbol{G^T} + \mathbf{G}\boldsymbol{E}\left[\boldsymbol{w_k w_k^T}\right]\boldsymbol{G^T} \qquad (5.138)\end{aligned}$$

Finally, since $\boldsymbol{E}\left[\boldsymbol{e_k w_k^T}\right] = \boldsymbol{E}\left[\boldsymbol{w_k e_k^T}\right] = \mathbf{0}$, $\boldsymbol{E}\left[\boldsymbol{e_k e_k^T}\right] = \boldsymbol{P_k}$ and $\boldsymbol{E}\left[\boldsymbol{w_k w_k^T}\right] = \boldsymbol{Q}$, $\boldsymbol{P_{k+1|k}}$ is:

$$\boldsymbol{P_{k+1|k}} = \boldsymbol{A}\,\boldsymbol{P_k}\,\boldsymbol{A^T} + \mathbf{G}\boldsymbol{Q}\boldsymbol{G^T} \qquad (5.139)$$

Kalman Filter Algorithm

Main relations derived in previous section can be put together to get the discrete Kalman filter algorithm, whose inputs are measurements $\boldsymbol{y_k}$ (at instant *k*), input of the model at time $k-1$, $\boldsymbol{u_{k-1}}$, and initial values of covariance matrix of the *'a priori'* error, $\boldsymbol{P_0}$, and the initial estimated state, $\hat{\boldsymbol{x}}_{\boldsymbol{0}}$. The output of the observer is the state vector's estimation at instant k, $\hat{\boldsymbol{x}}_{\boldsymbol{k}}$.

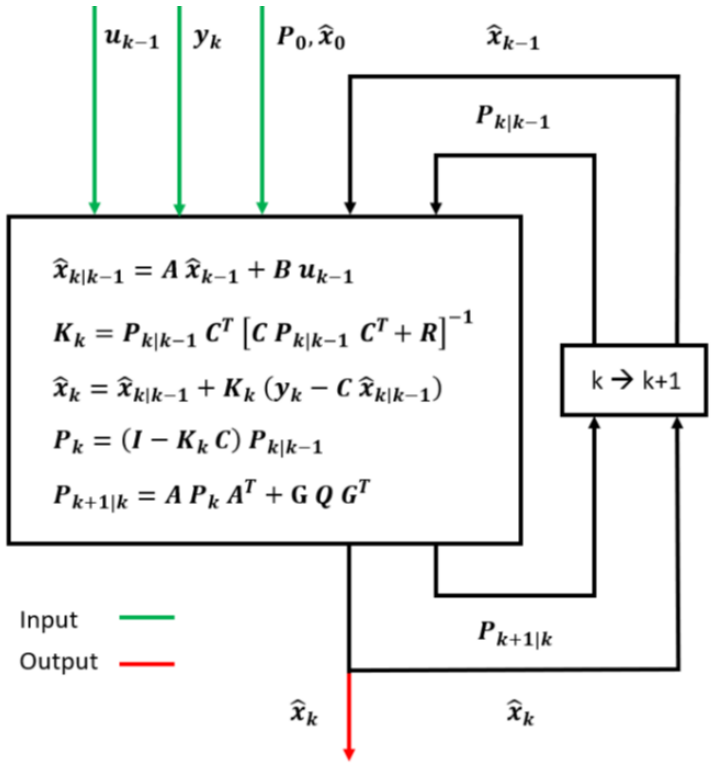

Figure 5.2: Scheme of the algorithm used in the Kalman filter, with its inputs, outputs and needed equations.

6 Implementation of an MPC for Floating Offshore Wind Turbines

In this chapter, a Model Predictive Control (MPC) for Floating Offshore Wind Turbines (FOWTs) will be implemented. The theoretical bases on MPCs described in Chapter 5 will be applied and adapted to the case under consideration. In particular, the notions seen for linear MPCs will be exploited for the design of a controller which is able to handle non-linear systems. Several linear systems are used to create MPC models, each consisting of the same states, inputs, and outputs, and obtained offline by linearising a non-linear starting system under certain operating conditions.

Initially a LTI (Linear Time Invariant) systems set is computed, it is a collection of LTI systems in state space form obtained by linearising a non-linear system at different points of a grid, which are identified by a set of parameters. Then they will be used to create different models for the MPC optimizations. The LTI systems set is calculated from the non-linear floating wind turbine model presented in Chapter 3, specifically the simplified system called NLS and described in Section 3.4.

Then the modules which make up the implemented controller will be described, most of which exploit the LTI systems set mentioned above. The main modules are:

- **Estimation:** In this module, some unknown quantities will be estimated, which are required to perform the numerical optimisation by which the control actions are calculated. To do this, linear Kalman filters such as the one described in Chapter 5 will be used.

- **Forecasting:** In this module, to complete the information required for numerical optimisation, the future values of the model's external inputs will be estimated within the prediction horizon of the MPC. To reach these results, an auto-regressive model will be used whose coefficients will be calculated online via a recursive least square method (RLS).

- **Optimization**: It is the main module, in which all information from other ones are used to perform the numeric optimization through which control output values are computed. An optimization will be performed each time step and only the first of a certain number of resultant future control actions will be considered (receding horizon method). Regarding optimization main ingredients (system model, cost function and constraints), at each time different version of them will be used to better describe real system and to correctly implement the strategies adopted in wind turbine control, described in Chapter 4.

6.1 LTI Systems Set

As above mentioned, the MPC controller implemented in this chapter is based on an set of LTI (Linear Time Invariant) systems, that is a set of linear systems in the state space form obtained from the linearisation of a certain non-linear system carried out at certain operation points to which are associated certain parameters whose time-varying values determine which of the linear systems must be used to appropriately describe the dynamics of the real system. In order to better understand how it is obtained and how it works, let us start with the non-linear model of the system obtained in Chapter 3. The model used for the construction of this set is the simplified system called NLS (Simplified Non-Linear) described in section 3.4:

$$\begin{cases} \dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{v}(t)) \\ \boldsymbol{y}(t) = g(\boldsymbol{x}(t), \boldsymbol{u}(t)) \end{cases} \tag{6.1}$$

With:

$$\boldsymbol{x} = \{s, h, p, \dot{s}, \dot{h}, \dot{p}, \Omega\}^{\top}$$

$$\boldsymbol{u} = \left\{C'_{\text{gen}}, \theta_{bl}\right\}^{\top}$$

$$\boldsymbol{v} = \left\{U_{wind,m}, \boldsymbol{F}^{T}_{\text{ext}}\right\}^{\top}$$

$$\boldsymbol{y} = \{s, h, p, \dot{s}, \dot{h}, \dot{p}, \Omega, P\}^{\top}$$

We recall that $\boldsymbol{x}$ is the states vector, whose component are surge (s), heave (h), pitch (p), their time derivative ($\dot{s}, \dot{h}, \dot{p}$) and rotor angular speed (Ω). Firsts three quantities are the 3 degrees of freedom of the floating structure considered, while rotor speed is the time-derivative of the fourth d.o.f., that is the azimuth angle.

Vector $\boldsymbol{u}$ is the controlled inputs vector, composed by quantities whose values are chosen by the control law, i.e., the generator torque (C'_{gen}) and the collective blade pitch angle (θ_{bl}), since we are going to deal with a variable-speed variable-collective-pitch wind turbine (see section 4.3).

Vector $\boldsymbol{v}$ is the external inputs vector, whose components are the uncontrollable quantities that act on the system and influence its behaviour. They are wind speed ($U_{wind,m}$) and wave excitation forces ($\boldsymbol{F}_{ext}$). It should be noted that in the simplified model, as specified in section 3.4, the entire velocity field distributed along the rotor area is not taken into account, but a single velocity value is considered, obtained as the average of a number of equally spaced values belonging to a grid covering the aforementioned area. Furthermore, let us remember that it was decided to separate the controlled inputs from the external ones because it will be useful in the implementation of the MPC controller and because it makes sense on a conceptual level; however, formally they are the same thing and for now we could consider a single vector grouping the quantities of $\boldsymbol{u}$ and $\boldsymbol{v}$. The main reason for making this distinction even now is to have a notation compatible with the following sections.

Finally, vector $\boldsymbol{y}$ is the outputs vector, whose chosen component are all the model states and the extracted power, that is the product of the generator torque and its angular speed.

The goal, in this section, is to linearize the simplified floating wind turbine nonlinear model and obtain the associated LTI systems set. To do this, it is necessary to choose in an appropriate way the points around which to carry out the linearization, called linearization points. A linearization point is uniquely defined by the value of the states and inputs of the system, and the resulting linear system will behave similarly to the nonlinear when it is under similar conditions, i.e., when the value of states and inputs do not differ much from those adopted during the linearization.

To understand what the appropriate points may be, we need to think about the control strategy adopted for the wind turbine and therefore, what will be the regime values of states and controlled inputs as the external inputs vary, considering the latter constant over time. From now on, we will call 'Operation Points' those points (characterized by determined values of the states, inputs and outputs) required by the control strategy. Recalling what is written in section 4.4.2.1, for example, according to the region in which one is located, i.e. according to the wind speed, different control strategies are undertaken and it is possible to assign the desired values of the controlled inputs in order to have the desired outputs, in particular, the power and rotor speed values .

It is assumed that if the control of the system occurs correctly, with a certain value of the external inputs and imposing the corresponding controlled inputs desired by the strategy, then the system will behave in a similar way to the desired one. For example, with a certain wind speed and resulting controlled inputs, the system conditions will most likely not deviate much from the nominal ones, so using a system linearized around that condition might be a good way. In the next sub-section, we will explain how the 'Operating Points' are chosen and how the corresponding states and controlled inputs of the system are calculated. Once they have been obtained, starting from them the linearization points will be defined and how to obtain the LTI systems set will be explained.

6.1.1 Operation Points

As mentioned earlier, in order to know which operation points are chosen, it is necessary to define the control strategy to be adopted. Since in the following sections we will compare the traditional controllers presented in section 4 with the MPC controller implemented in this chapter, we will consider the same control strategy adopted in the ROSCO controller, the one described in section 4.4.2.1, in which the different control regions were discussed, and in which the stationary values of the controlled inputs, rotor speed and extracted power as the average wind speed varies were defined for each region (see Figure 4.10). From now on, we will call SS these steady state points provided by control strategy, while we will call OP the operation point obtained from the model.

Specifically, OP points are computed through a Simulink tool which can calculate trim points from the non-linear model and from some specification. A trim point is a point whose values of states, their derivative, inputs, and outputs of the system are compatible with specification made. Simulink tool searches, by means of a sequential quadratic algorithm, the system condition that comes closest to the desired nominal condition set in the specification. Stationary points are often sought, so the derivatives of the system states are often set to zero, but it is also possible to impose no constraint on these quantities or to impose values other than zero.

To each imposed quantity, it is also possible to indicate minimum and maximum values beyond which the search algorithm tries not to go, as well as it is possible to impose starting values of unknown quantities and some algorithm options, such as maximum number of iteration it can make and tolerances value, that is the maximum error value (in the sense of the magnitude of departure from the limits) allowed.

In this case, for each OP computed, following specifications are made:

- **Inputs:**
  - External inputs $\boldsymbol{v}$: As mentioned above, main quantity on which the control strategy is based is wind speed. For each OP a certain value of the mean wind speed across rotor area, $U_{wind,m}$, is chosen. As far as wave excitation forces are concerned, they are specified as zero. All external inputs are imposed to be know, that is during search their value cannot vary.

  - Controlled Inputs $\boldsymbol{u}$: Generator torque, C'_{gen} , is specified to be unknown so that the most suitable value can be found, even though the value predicted by the control strategy is known. This is done so as not to over-constrain the search, but nevertheless tight bounds are adopted so that the value found by the algorithm does not deviate much from that predicted by the SS points. As for the blade pitch angle, θ_{bl}, however, it is considered to be known and its value is that predicted by the control strategy.

- **States:** The states surge (s), heave (h) and pitch (p) are specified to be unknown. Their initial values are calculated starting from thrust force provided by steady state condition at wind speed considered. Their derivative, on the other hand, are imposed to zero to try to obtain an OP that is also a steady state value. Rotor speed Ω is imposed to be unknown and limited by tight bounds as done with generator torque.

- **State derivatives:** The state derivatives are, in this case, all imposed to be zero.

- **Outputs:** The only one of the outputs for which no specification has yet been made, since the others correspond to the states of the system, is the power extracted from the electric generator. This is considered unknown, and the value expected by the control strategy for the wind speed considered is specified.

Three examples of results from the operation point search are shown on the following page. It was decided to also depict the results obtained from the MOST model and the non-simplified non-linear model, NL, in order to understand whether the simplified model (NLS) gives results that were compatible with the models most faithful to the real system. The black lines and dots refer to the nominal conditions, i.e., the stationary points predicted by the control strategy (see Figure 4.10), while the red, blue and green dots show the operation points obtained through the search explained above. The red dotted line represents the constant power curve whose value corresponds to the nominal one, to see how close the points found come to this objective. It can be seen from the graphs that the results obtained are quite accurate and compatible with each other.

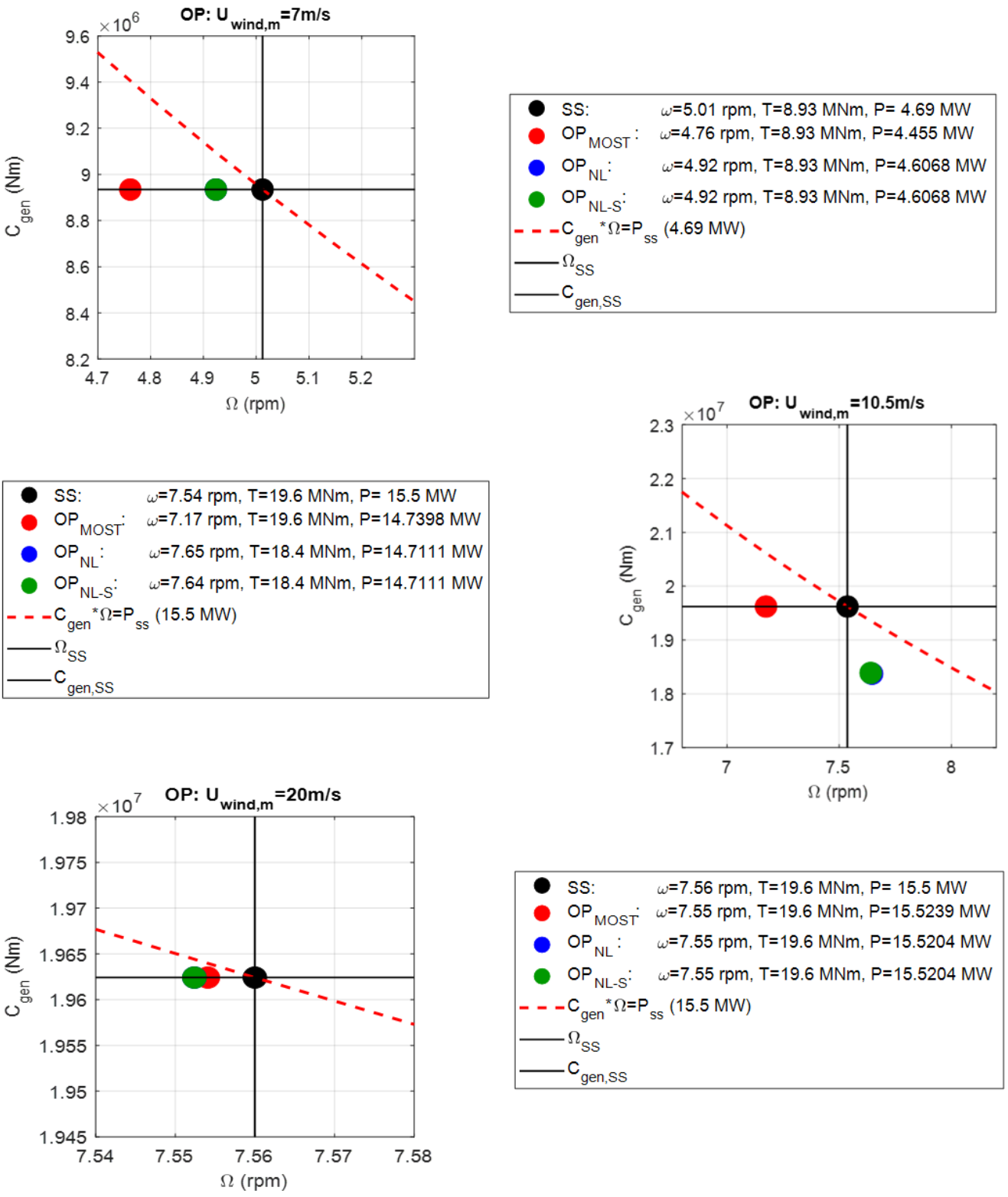

Figure 6.1: Example of Operation Point at three different mean wind speed. Each plot shows comparison between steady state values suggested by control strategy (SS) and operation points (OP) computed through trimming algorithm for MOST, NL and NLS models.

Figure 6.2 shows the operation points found for the simplified model (NLS) for all considered wind speeds. The OP points obtained are compared with the nominal steady state points SS, predicted by the control strategy, and it can be seen that the obtained results are overall very close to the nominal values. Furthermore, it is interesting to note that for wind speed values a little higher than the nominal one, slightly lower blade pitching values occur, to compensate for the fact that the model achieves lower powers than the theoretical ones, mainly due to the non-zero angle between wind and rotor caused by the structure being rotated by a certain pitch angle (not taken into account when calculating the SS values).

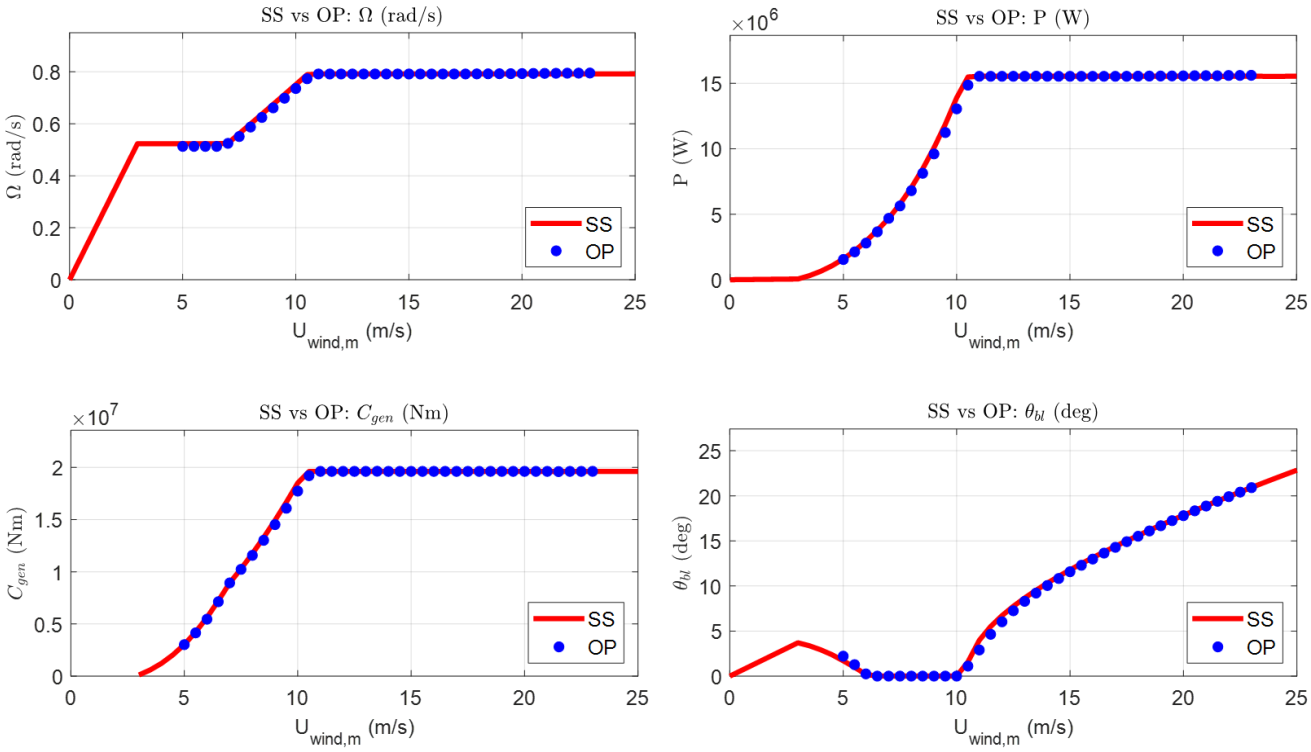

Figure 6.2: Comparison between nominal steady state point provided by control strategy (SS) and operation point (OP) obtained by trimming search for the simplified model NLS. Quantities represented are the two controlled inputs (C_{gen} and θ_{bl}), rotor angular speed Ω and extracted power P.

Finally, in Figure 6.2, as in 6.1, only some of the quantities in play are represented, but it is important to emphasise that for each search, the values of all states, inputs and outputs are found.

6.1.2 Linearization

In the previous section, the operating points, i.e. the steady state points (denoted by their values of states, inputs and outputs), varying with average wind speeds were identified. It was also mentioned that the system model, given its non-linearities, can only be appropriately approximated by a linear system if the operating conditions do not deviate much from those under which the linearisation is carried out. It is now necessary to understand what are the points around which to linearise the system in order to obtain a set of linear systems which, together, are able to approximate the simplified nonlinear system well. A linearisation point is uniquely defined once the states and inputs of the system have been defined, so it is understood that if one wanted to have a set of linear systems covering all possibilities it would require a lot of computing time to obtain it and a lot of memory to use it. It is therefore necessary to make choices to obtain a set of linear systems that is not too numerous. Linearization points will correspond to any possible combination of these parameters. For this reason, the set of systems which is considered is called "parameter varying system ".

Given what has been explained in the previous section and in the section dedicated to the explanation of the control strategies, the first parameter that has been chosen to use is the average wind speed, on which depends, in addition to the behaviour of the system, the control strategy. Due to its importance, it was decided to have a dense discretization of this parameter within the considered range. Furthermore, for the LTI systems set that will be built, it was decided to adopt two further parameters, in order to have a total of three parameters, which we have seen to be a good compromise between the performance offered and the memory occupied, as well as calculation time to obtain it. The two further parameters chosen are the rotor speed and the collective blade pitch.

The system presents evident non-linearities also in the mooring forces and on the viscous friction forces, proportional to the square of the speed, which lead to greater discrepancies between the non-linear system and the linearized system when the surge value and its derivative deviate from those adopted during linearization (as far as heave and pitch are concerned, the effect is less as they tend to have smaller ranges of variation). However, it was decided not to consider the surge and the surge speed as parameters, in order to reduce the size of the LTI systems set, because the errors that arise are in terms of these two quantities, which are of low importance for the turbine control , in the sense that it is not necessary to predict the value of these quantities with certainty because it is not necessary to trace them precisely.

The last important concept to underline is that, as we have seen previously, as the wind speed varies, the operation points obtained also have different values of the states, therefore it is advisable to take this into account during the linearization. On the other hands, as far as the other two parameters vary, only the corresponding state (or input) varies. In this case it was decided to vary it uniformly, with the central value corresponding to that in the operation point obtained with a given wind speed. In other words, for each wind speed an OP is obtained from which different linearization points are obtained by uniformly varying the remaining two parameters.

The set of linearization points chosen is called linearization grid and, in accordance with what has been said up to now, the image below shows an example of a linearization grid with the three parameters previously described. In this image, for graphic reasons, there is a reduced number of points, but actually, to obtain a systems set that well approximates the non-linear system, a greater number of points will be adopted, especially as regards the wind speed.

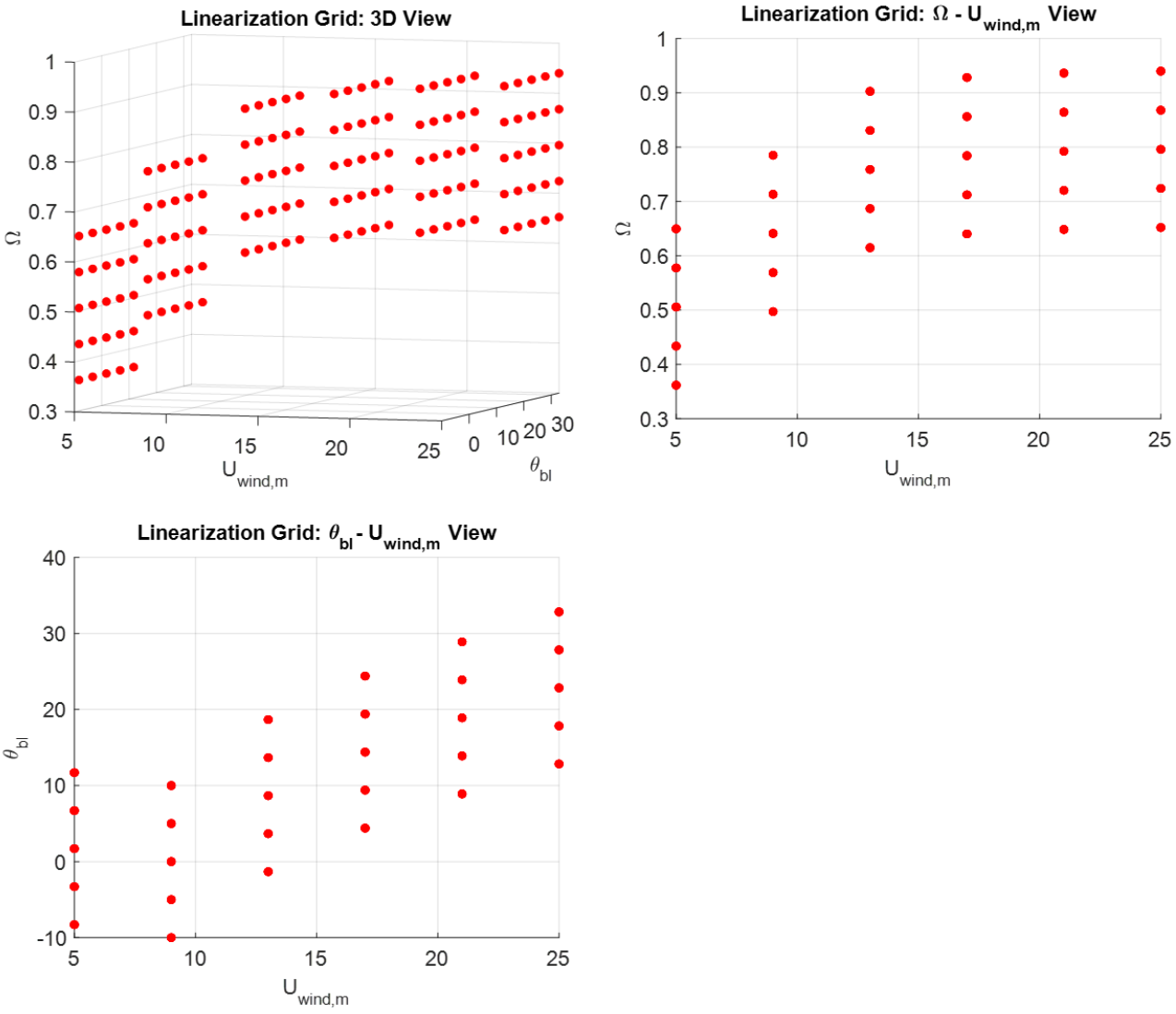

Figure 6.3: Example of linearization grid; here the adopted parameter are mean wind speed across rotor area ($U_{wind,m}$), rotor angular speed (Ω) and collective blade-pitch angle (θ_{bl}). The image is divided into three parts, where in the second two a projection onto $\Omega - U_{wind,m}$ plane and onto $\theta_{bl} - U_{wind,m}$ plane are depicted, to clearly show that Ω and θ_{bl} central grid points depend on $U_{wind,m}$, while other ones are evenly spaced around the latter.

Once we have understood how to obtain the linearization points, we proceed to explain how the linearization is carried out. In this case, a centered finite difference method is used, implemented via a Matlab function. The starting point is the simplified non-linear system (NLS):

$$\begin{cases} \dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}) \\ \boldsymbol{y}(t) = g(\boldsymbol{x}, \boldsymbol{u}) \end{cases} \quad (6.2)$$

A linearization point is denoted with subscript 0, and particularly with its states, controlled inputs and external inputs, respectively vectors $\boldsymbol{x}_0$, $\boldsymbol{u}_0$ and $\boldsymbol{v}_0$. Using first order Taylor approximation of states derivatives and outputs, we can write:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}) \approx f(\boldsymbol{x}_0, \boldsymbol{u}_0, \boldsymbol{v}_0) + \frac{\partial f}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}_0, \boldsymbol{u}_0, \boldsymbol{v}_0} \cdot (\boldsymbol{x} - \boldsymbol{x}_0) + \frac{\partial f}{\partial \boldsymbol{u}}\Big|_{\boldsymbol{x}_0, \boldsymbol{u}_0, \boldsymbol{v}_0} \cdot (\boldsymbol{u} - \boldsymbol{u}_0) + \frac{\partial f}{\partial \boldsymbol{v}}\Big|_{\boldsymbol{x}_0, \boldsymbol{u}_0, \boldsymbol{v}_0} \cdot (\boldsymbol{v} - \boldsymbol{v}_0) \quad (6.3)$$

$$\boldsymbol{y} = g(\boldsymbol{x}, \boldsymbol{u}) \approx g(\boldsymbol{x}_0, \boldsymbol{u}_0) + \frac{\partial g}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}_0, \boldsymbol{u}_0} \cdot (\boldsymbol{x} - \boldsymbol{x}_0) + \frac{\partial g}{\partial \boldsymbol{u}}\Big|_{\boldsymbol{x}_0, \boldsymbol{u}_0} \cdot (\boldsymbol{u} - \boldsymbol{u}_0) \quad (6.4)$$

Now, some definitions can be made:

$$\frac{\partial f}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}_0, \boldsymbol{u}_0, \boldsymbol{v}_0} = \boldsymbol{A}, \quad \frac{\partial f}{\partial \boldsymbol{u}}\Big|_{\boldsymbol{x}_0, \boldsymbol{u}_0, \boldsymbol{v}_0} = \boldsymbol{B_u}, \quad \frac{\partial f}{\partial \boldsymbol{v}}\Big|_{\boldsymbol{x}_0, \boldsymbol{u}_0, \boldsymbol{v}_0} = \boldsymbol{B_v}$$
$$\frac{\partial g}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}_0, \boldsymbol{u}_0, \boldsymbol{v}_0} = \boldsymbol{C}, \quad \frac{\partial g}{\partial \boldsymbol{u}}\Big|_{\boldsymbol{x}_0, \boldsymbol{u}_0, \boldsymbol{v}_0} = \boldsymbol{D} \quad (6.5)$$

All these quantities, valid for one linearization point $(\boldsymbol{x}_0, \boldsymbol{u}_0, \boldsymbol{v}_0)$, can be computed, as in this case, through a centred finite difference. Specifically, taking matrix $\boldsymbol{A}$ as an example, it is calculated as follows:

$$\boldsymbol{A}_{ij} = \frac{\partial f_i}{\partial \boldsymbol{x}_j}\Big|_{\boldsymbol{x}_0, \boldsymbol{u}_0, \boldsymbol{v}_0} \approx \frac{f_i(\boldsymbol{x}_0 + \boldsymbol{\varepsilon}_{x,j}, \boldsymbol{u}_0, \boldsymbol{v}_0) - f_i(\boldsymbol{x}_0 - \boldsymbol{\varepsilon}_{x,j}, \boldsymbol{u}_0, \boldsymbol{v}_0)}{2 \cdot \epsilon_{x,j}} \quad (6.6)$$

Where $\boldsymbol{\varepsilon}_{x,j}$ is a vector of the same dimension of $\boldsymbol{x}$ with all zero component except the j-th one, which is $\epsilon_{x,j}$. Using the same technique, the components of the other matrices can be calculated. Generally, the values of ϵ can be different between states, inputs, and outputs and between the various components of them, but in this case, it was decided to adopt a single value, equal to 0.01% of the value of the component at the linearisation point (or 10^{-6} if the latter is zero).

From equations 6.3 and 6.4, we can define:

$$
\begin{aligned}
(\dot{\boldsymbol{x}})_0 &= \boldsymbol{f}(\boldsymbol{x}_0, \boldsymbol{u}_0, \boldsymbol{v}_0), \\
\boldsymbol{y}_0 &= \boldsymbol{g}(\boldsymbol{x}_0, \boldsymbol{u}_0) \\
\delta\boldsymbol{x} &= \boldsymbol{x} - \boldsymbol{x}_0 \\
\delta\boldsymbol{u} &= \boldsymbol{u} - \boldsymbol{u}_0 \\
\delta\boldsymbol{v} &= \boldsymbol{v} - \boldsymbol{v}_0 \\
\delta\boldsymbol{y} &= \boldsymbol{y} - \boldsymbol{y}_0 \\
\dot{\delta\boldsymbol{x}} &= \dot{\boldsymbol{x}}
\end{aligned}
\qquad (6.7)
$$

Finally, we can write the obtained linear system in state space form as:

$$
\begin{cases}
\dot{\delta\boldsymbol{x}} = (\dot{\boldsymbol{x}})_0 + \boldsymbol{A} \cdot \delta\boldsymbol{x} + \boldsymbol{B_u} \cdot \delta\boldsymbol{u} + \boldsymbol{B_v} \cdot \delta\boldsymbol{v} \\
\delta\boldsymbol{y} = \boldsymbol{C} \cdot \delta\boldsymbol{x} + \boldsymbol{D} \cdot \delta\boldsymbol{u}
\end{cases}
\qquad (6.8)
$$

System 6.8 is different for each combination of parameters, more precisely, there are different offsets and matrices for any combination of them. Offsets are the quantities $\boldsymbol{x}_0, \boldsymbol{u}_0, \boldsymbol{v}_0, \boldsymbol{y}_0$ and $(\dot{\boldsymbol{x}})_0$, while the matrices are $\boldsymbol{A}, \boldsymbol{B_u}, \boldsymbol{B_v}, \boldsymbol{C}$ and $\boldsymbol{D}$. Moreover, in this case, a nearest-point interpolation is performed, i.e. once the parameters have been measured, the linear system adopted is that obtained from the grid point that has the closest parameter values to those measured. The search for the nearest grid point is carried out at each simulation step, so each different linear system is active for one step.

6.1.3 LTI Systems Set Validation

In this section, a mostly qualitative validation of the previously obtained LTI systems set will be carried out. To perform this, the behavior of the systems will be compared with the non-linear system from which it is computed, that is the simplified model called NLS (Non-Linear-Simplified), described in section 3.4. To compare these two models, simulations with the same inputs and control algorithm are carry out.

External inputs are the wave excitation forces/torques and a simplified form of wind speed, that is the average speed across rotor area, instead of wind speeds at several grid points distributed along it, as it occurs with NL (Non-Linear) model described in section 3. Alternatively, wind speed at grid point corresponding with hub center can be considered. In this case, the wind speed data used are obtained from TurbSim by setting an average speed (over time) at the center of the hub of 9 m/s (see section 3.2.1). While, regarding the waves, a JONSWAP spectrum with significant height H_s=4 m and peak period T_p=6 s is used (see section 3.2.2).

Since the MPC controller implemented in this chapter has not yet been described in detail, a Baseline controller will be used for these simulations, which has already been described in Section 4.

Figures below show time series of external inputs and some simulation result, that is time series of some of the output of the system (see equation 6.1). Reported results are surge, heave, pitch, rotor angular speed and power output.

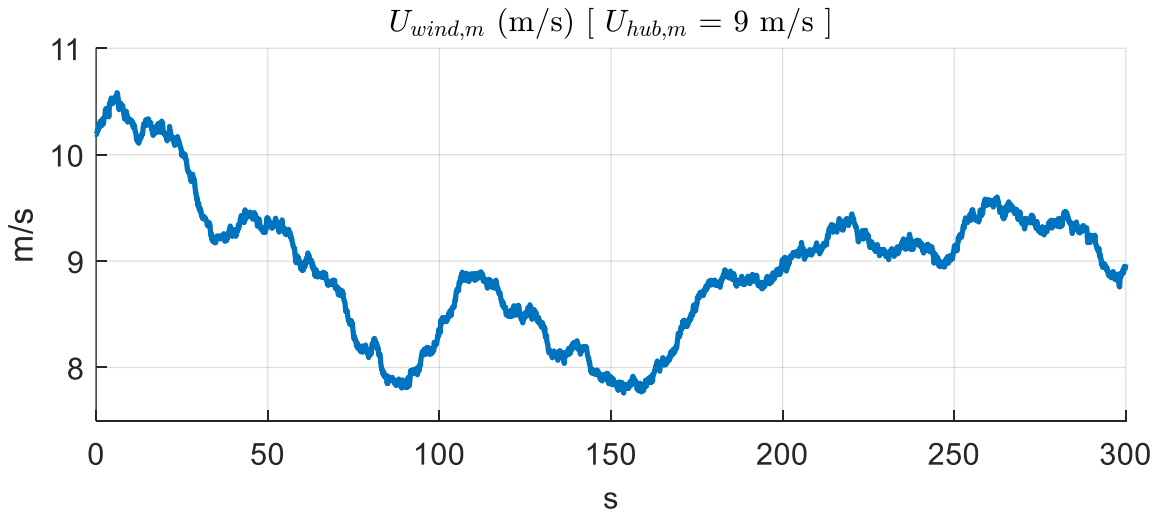

Figure 6.4: Time series of $U_{wind,m}$, spatial mean wind speed (obtained from TurbSim with hub temporal mean speed $U_{hub,m} = 9\,m/s$).

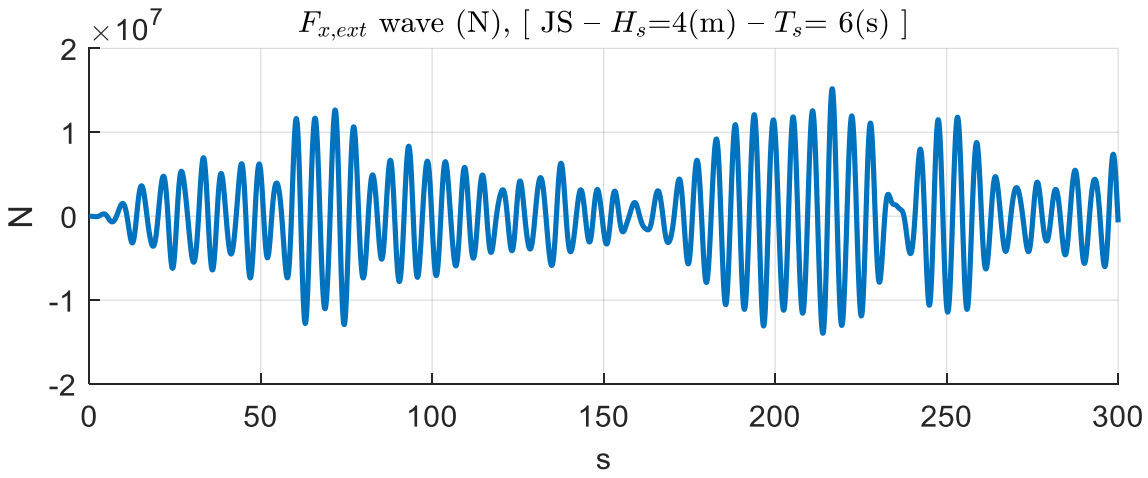

Figure 6.5: Time series of Fx,ext (wave excitation force along x direction), computed with a JONSWAP (JS) spectrum, significant height Hs=4 m and peak period Tp=6 s.

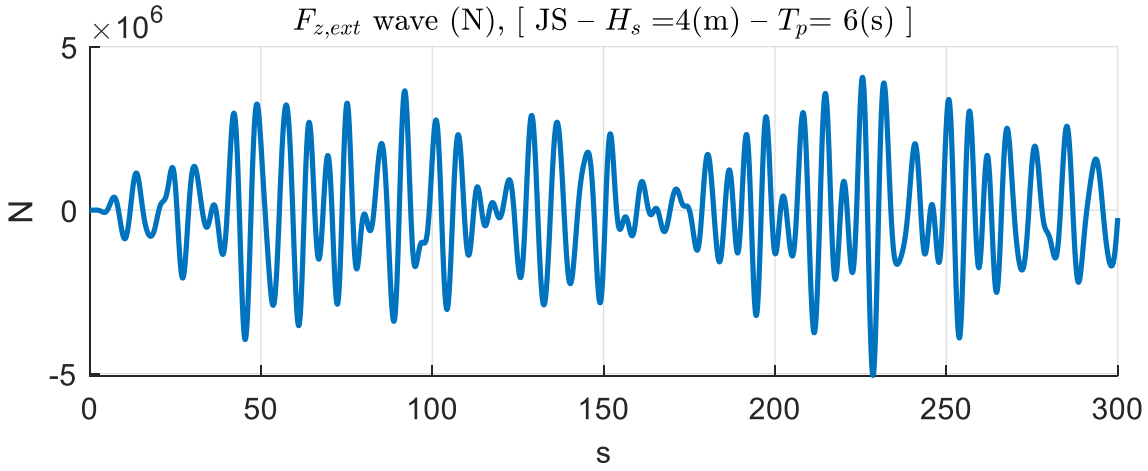

Figure 6.6: Time series of Fz,ext (wave excitation force along z direction), computed with a JONSWAP (JS) spectrum, significant height Hs=4 m and peak period Tp=6 s.

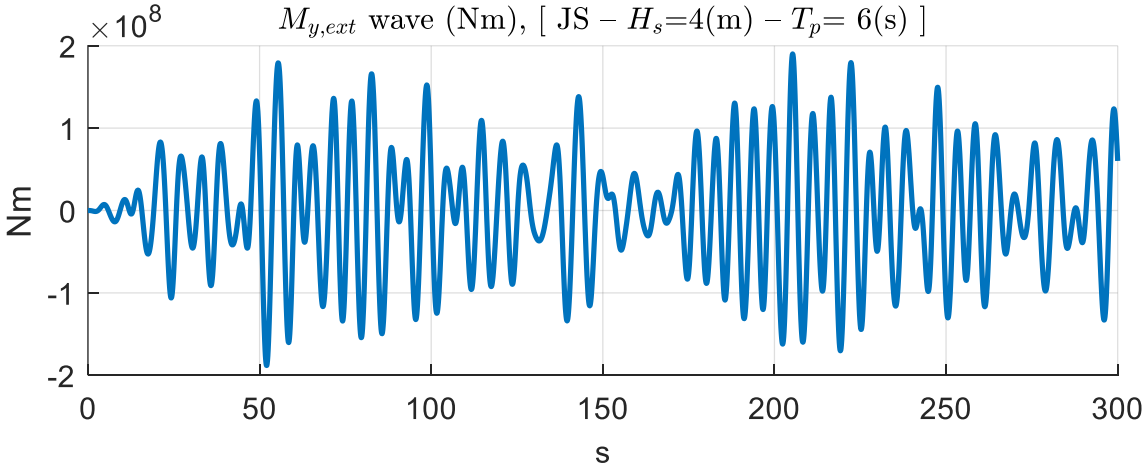

Figure 6.7: Time series of My,ext (wave excitation torque around y axis), computed with a JONSWAP (JS) spectrum, significant height Hs=4 m and peak period Tp=6 s.

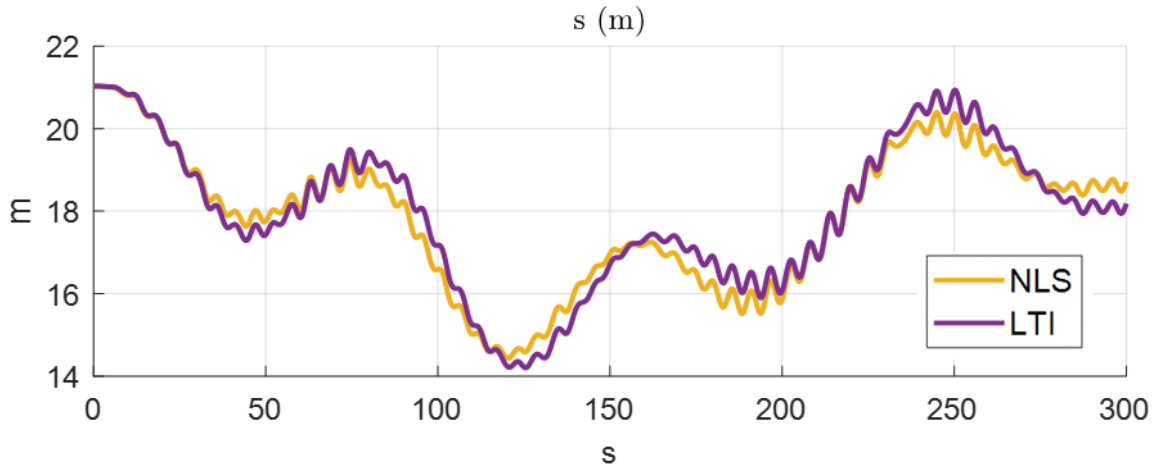

Figure 6.8: Comparison between simplified-non-linear (NLS) and linear-time-invariant (LTI) models: surge (m).

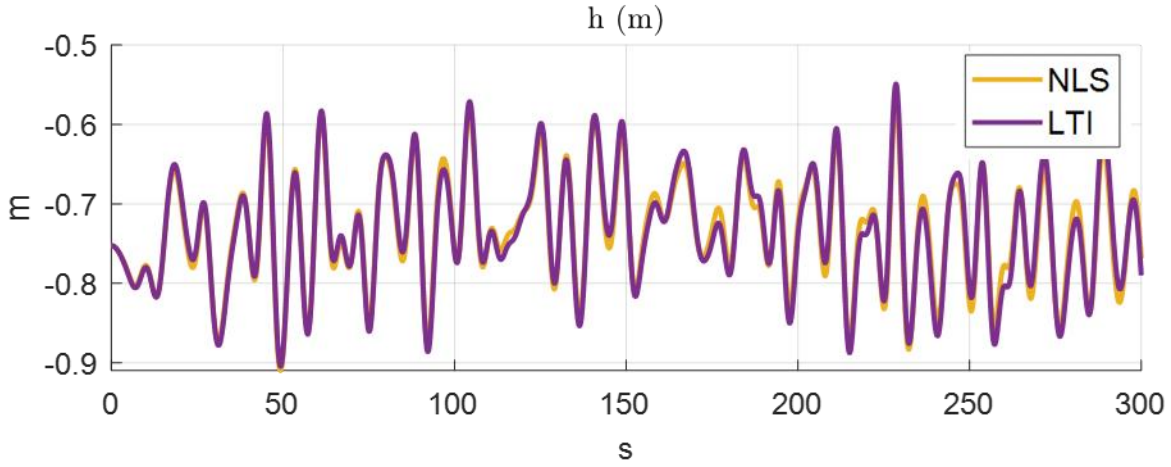

Figure 6.9: Comparison between simplified-non-linear (NLS) and linear-time-invariant (LTI) models: heave (m).

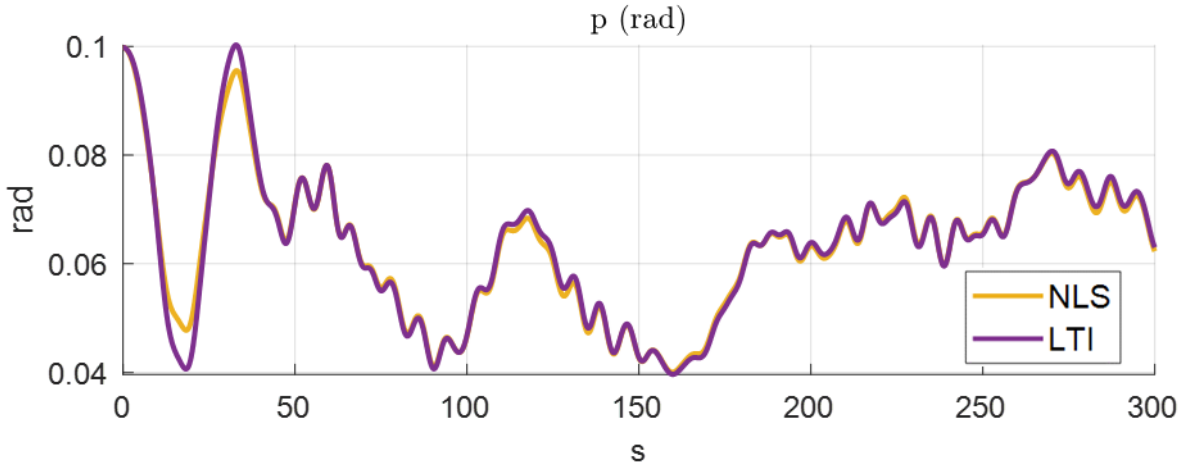

Figure 6.10: Comparison between simplified-non-linear (NLS) and linear-time-invariant (LTI) models: pitch (rad).

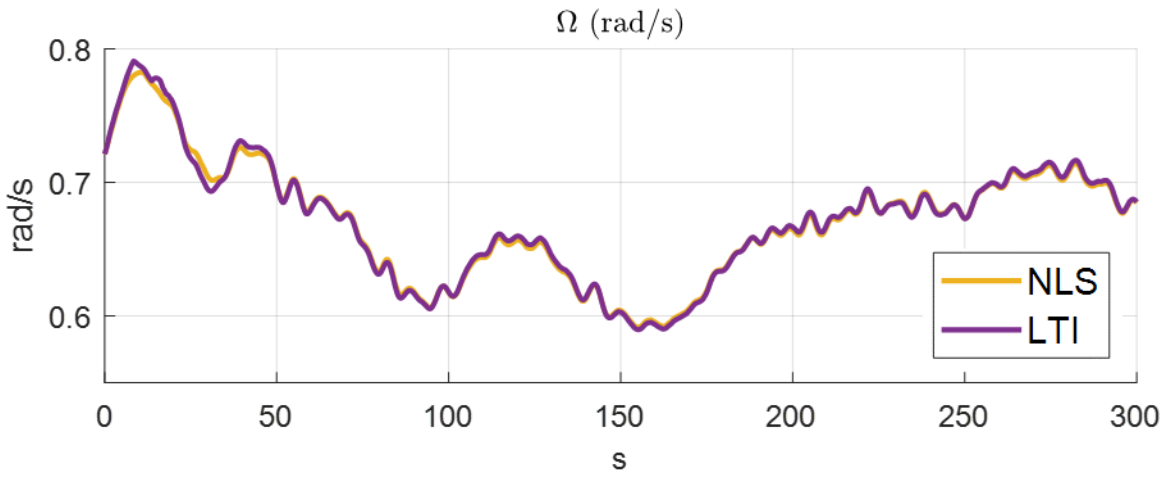

Figure 6.11: Comparison between simplified-non-linear (NLS) and linear-time-invariant (LTI) models: rotor angular speed Ω (rad/s).

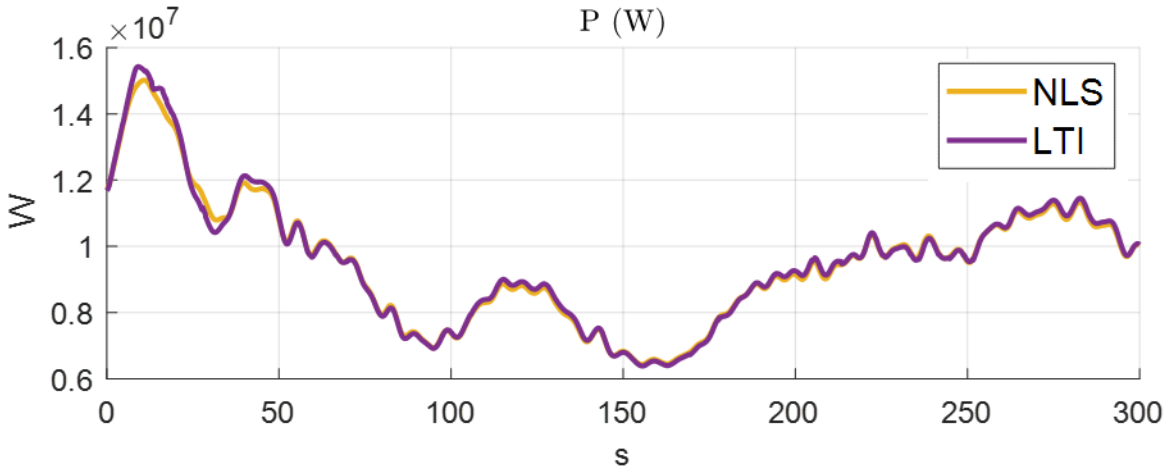

Figure 6.12: Comparison between simplified-non-linear (NLS) and linear-time-invariant (LTI) models: extracted power (W).

From the graphs, it can be seen that the two systems lead to very similar results, it can also be noted what has been said before about the choice of parameters, i.e., the surge and/or its derivative, on which largely non-linear forces depend, were not chosen as linearization grid parameters. This leads to greater differences between the models in terms of surge, however, the choices made (blade pitch angle and rotor speed) lead to greater accuracy in terms of rotor speed and extracted power.

6.2 MPC Modules

6.2.1 Estimation

In this section, we will deal with the estimation of the unknown model quantities that must be known to construct the MPC controller. In particular, the model of the system used for the constrained optimisation in the controller algorithm is derived from the LTI systems set, where each optimisation is performed with the linear model that best approximates the starting non-linear model at the instant in which the calculation is performed. As will be specified later, the only quantities that will not be measured, which are to be estimated, are the wind speed and the wave excitation forces, which, as seen in Section 6.1.2, are treated as inputs.

Furthermore, in section 5.2.6 it was shown how to estimate certain unmeasured states of a linear system by means of so-called observers, and the algorithm of one type of them, called Kalman filter, was obtained. To perform the estimation of unknown external input, two different observers will be built. For each of them, models will be constructed from the previously obtained LTI systems set, in which the unknown quantities become states of the system, so that they can be estimated by means of a linear discrete Kalman filter, whose algorithm was described in section 5.2.6.1.

6.2.1.1 Wind Kalman Filter

The starting LTI systems set from which the model for the Kalman filter will be constructed is defined as follows (obtained in section 5.2.6.1):

$$\begin{cases} \dot{\delta \boldsymbol{x}} = {}^{ip}(\dot{\boldsymbol{x}})_0 + {}^{ip}\boldsymbol{A} \cdot \delta \boldsymbol{x} + {}^{ip}\boldsymbol{B_u} \cdot \delta \boldsymbol{u} + {}^{ip}\boldsymbol{B_v} \cdot \delta \boldsymbol{v} \\ \delta \boldsymbol{y} = {}^{ip}\boldsymbol{C} \cdot \delta \boldsymbol{x} + {}^{ip}\boldsymbol{D} \cdot \delta \boldsymbol{u} \end{cases} \quad (6.9)$$

With:

- $\boldsymbol{x} = \{s, h, p, \dot{s}, \dot{h}, \dot{p}, \Omega\}^\top$,
- $\boldsymbol{y} = \{s, h, p, \dot{s}, \dot{h}, \dot{p}, \Omega, P\}^\top$
- $\boldsymbol{u} = \left\{C'_{\text{gen}}, \theta_{bl}\right\}^\top$,
- $\boldsymbol{v} = \left\{U_{wind,m}, \boldsymbol{F}^T_{\text{ext}}\right\}^\top$
- ${}^{ip}(\dot{\boldsymbol{x}})_0 = f\left({}^{ip}\boldsymbol{x}_0, {}^{ip}\boldsymbol{u}_0, {}^{ip}\boldsymbol{u}_0\right)$,
- ${}^{ip}\boldsymbol{y}_0 = g\left({}^{ip}\boldsymbol{x}_0, {}^{ip}\boldsymbol{u}_0\right)$
- $\delta \boldsymbol{x} = \boldsymbol{x} - {}^{ip}\boldsymbol{x}_0$,
- $\delta \boldsymbol{u} = \boldsymbol{u} - {}^{ip}\boldsymbol{u}_0$
- $\delta \boldsymbol{v} = \boldsymbol{v} - {}^{ip}\boldsymbol{v}_0$,
- $\delta \boldsymbol{y} = \boldsymbol{y} - {}^{ip}\boldsymbol{y}_0$

The vectors $\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}$ and $\boldsymbol{y}$ are, respectively, the states, controlled inputs, external inputs, and outputs vectors. The states are *s* (surge), *h* (heave), *p* (pitch), their derivatives and the rotor speed Ω. The controlled inputs are the generator torque C'_{gen} and the collective blade-pitch angle θ_{bl}. The external inputs are the average wind speed (across rotor area) in the direction parallel to the surge, $U_{wind,m}$, and the wave excitation forces $\boldsymbol{F}_{ext}$. Finally, the outputs are the states and the extracted power *P*.

The $\boldsymbol{x_0, u_0, v_0, y_0}$ and $(\dot{\boldsymbol{x}})_0$ vectors, called offsets, are the $\boldsymbol{x, u, v, y}$ and $\dot{\boldsymbol{x}}$ vectors calculated at the points of the linearisation grid, while the $\boldsymbol{A, B_u, B_v, C}$ and $\boldsymbol{D}$ matrices are the matrices describing the linear system obtained at these points. The superscript ***ip*** is the grid index, which identifies which linear system is being considered among those constituting the LTI systems set. There will be a module in the controller which, based on the values of the parameters chosen to construct the linearisation grid, will work out which system is best to use, obtaining the appropriate ***ip***-index at each step.

From section 5.6.2.1, we recall the state space form of a discrete model used in a linear-discrete Kalman filter, where the subscript KF means we refer to quantities of Kalman filter model, while no subscript means we refer to LTI systems quantities or to quantities that only compare in the Kalman filter. Superscript *ip* indicates the linear system from which Kalman filter quantities are computed, since at each time step we are going to use a different Kalman filter model according to parameter values. All models are computed offline, so, during working operations, the better one (according to parameter values) is chosen to be used for that time step. From these considerations, discrete Kalman filter model becomes:

$$\begin{cases} {}^{ip}_{KF}\boldsymbol{x_k} = {}^{ip}_{KF}\boldsymbol{A} \cdot {}^{ip}_{KF}\boldsymbol{x_{k-1}} + {}^{ip}_{KF}\boldsymbol{B} \cdot {}^{ip}_{KF}\boldsymbol{u_{k-1}} + \boldsymbol{G} \cdot \boldsymbol{w_k} \\ {}^{ip}_{KF}\boldsymbol{y_k} = {}_{KF}\boldsymbol{C} \cdot {}^{ip}_{KF}\boldsymbol{x_k} + \boldsymbol{v_k} \end{cases} \qquad (6.10)$$

Where:

- ${}_{KF}\boldsymbol{x}$**:** It is the Kalman filter states vector**,** which is defined as follows: ${}_{KF}\boldsymbol{x} = \{\Omega - \Omega_0\ , vw_t\ , vw_m - U_{wind,m,0}\}^{\top}$, where vw_m is the (temporal) mean part of wind speed $U_{wind,m}$, while vw_t is the turbulent part, such that their sum is the wind speed $U_{wind,m}$. All states are expressed in term of difference from values assumed at linearization condition (turbulent component at linearization condition is imposed to be zero).

- ${}_{KF}\boldsymbol{u}$ **:** It is the Kalman filter inputs vector, known by assumption. It is defined as follow: ${}_{KF}\boldsymbol{u} = \{C'_{\text{gen}} - C'_{\text{gen}\ 0},\ \theta_{bl} - \theta_{bl_0},\ p - p_0,\ \dot{s} - \dot{s}_0,\ \dot{h} - \dot{h}_0,\ \dot{p} - \dot{p}_0,\ \dot{\Omega}_0\}^{\top}$. Subscript 0 indicates values at linearization point. Last four input defined are considered known (instead of the state Ω, for which a noise is assumed) because their influence on states derivatives is lower, so, to simplify the model, we assume there is no noise on this information without committing high errors.

- $\boldsymbol{w}$**:** It is the Kalman filter process noise vector. It is a vector $\boldsymbol{w} = \{w_1, w_2, w_3\}^{\top}$, whose component are inputs of the system and whose value is normally unknown. In this case they are the "stochastic part" of the states, assumed to be random processes, which are white noises with zero mean and a certain value of variance;

- ${}_{KF}\boldsymbol{y}$ **:** It is the Kalman filter output vector (in this case with only one component), it is defined as ${}_{KF}\boldsymbol{y} = \{\Omega - \Omega_0\}$. It is the measured quantity, whose measure is composed by a deterministic part ($\boldsymbol{C} \cdot \boldsymbol{x}$) and stochastic one due to sensor noise ($\boldsymbol{v}$);

- $\boldsymbol{v}$**:** It is the Kalman filter measurement noise vector $\boldsymbol{v} = \{ v_1 \}$**,** the stochastic part of the output's measurement, which is assumed to be a white noise with zero mean and a certain value of variance, $\sigma^2(v_1)$;

- $\boldsymbol{G}$: It is the process noise matrix; in this case it is defined as the identity matrix $\boldsymbol{I}$, since, by definition, $\boldsymbol{w_k}$ is already the vector defined as the difference between the actual vector of states and that predicted by the mathematical model.

- $\boldsymbol{C}$: It is the output matrix, which, in this case, is a row vector, since there is only one output ($y = \{ \Omega - \Omega_0 \}$). $\boldsymbol{C}$ is defined as $\boldsymbol{C} = \{ 1, 0, 0 \}^\top$, since output is equal to the first state.

- T: sample time of discrete model, such that, as an example $\boldsymbol{x_k} = \boldsymbol{x}(kT)$**.**

The elements in equation 6.10 that have yet to be defined are the matrices ${}_{KF}\boldsymbol{A}$ and ${}_{KF}\boldsymbol{B}$, which express the relationship between states at time k and states and inputs at time $k-1$, as well as $\boldsymbol{w}$ at time k. Focusing on the deterministic part (thus excluding the term dependent on $\boldsymbol{w}$), the first row is the equation that results in the state $\Omega - \Omega_0$ at time k from the states and inputs at time $k-1$. This equation corresponds to the last row of the first vector equation of system 6.9, so the first rows of the matrices ${}_{KF}\boldsymbol{A}$ and ${}_{KF}\boldsymbol{B}$ can be derived by manipulating the matrices $\boldsymbol{A}, \boldsymbol{B_u}$ and $\boldsymbol{B_v}$ of the starting LTI system, after discretizing it. For other rows, on the other hand, it is necessary to define the wind model, i.e. the equations by which the states vw_t and $vw_m - U_{wind,m,0}$ at time k can be predicted from the same quantities at time $k-1$. The wind model used is inspired by the model suggested by Knudsen T, Bak T and Soltani M in [31]. Adapting this model to the discrete case under consideration, the following equations can be written:

$$vw_{t,\,k} = (1 - T\frac{\pi \cdot vw_{m,\,k-1}}{2L}) \cdot vw_{t,\,k-1} + w_{2,k} \qquad (6.11)$$

$$vw_{m,\,k} - U_{wind,m,0} = (vw_{m,\,k} - U_{wind,m,0}) + w_{3,k} \qquad (6.12)$$

L is a turbulence length scale parameter generically defined as $L = 3D$, where D is the rotor diameter. T is the discrete Kalman filter sample time, while w_2, w_3 are $\boldsymbol{w}$ vector's component, all of them defined above. As can be seen in 6.11 and 6.12, mean part of wind speed is modelled as a random walk, while turbulent part step increment is proportional (with minus sign) to mean part.

Finally, to be able to implement the discrete Kalman filter algorithm obtained in section 5.2.6.1, it is necessary to derive the matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$, which we recall being the covariance matrices of the vector $\boldsymbol{w}$ (process noise) and the vector $\boldsymbol{v}$ (measurement noise). Matrix $\boldsymbol{R}$, since we have only one measured output, is simply its variance: $\boldsymbol{R} = \sigma^2(v_1)$. On the other hand, regarding matrix $\boldsymbol{Q}$, going back to the work done in [31] and modifying the results to make them compatible with the discrete model adopted, it is defined as follows:

$$\mathbf{Q}_k = T \cdot \begin{bmatrix} 1 \times 10^{-5} & 0 & 0 \\ 0 & \frac{\pi \cdot vw_{m,k}{}^3 \cdot \mathrm{TI}^2}{L} & 0 \\ 0 & 0 & \frac{2^2}{600} \end{bmatrix} \qquad (6.13)$$

$\mathbf{Q}$ has got a diagonal form (it means that different components of state vector are uncorrelated), where diagonal elements are the variances of $\boldsymbol{w}$ vector's components. It can be seen variance of second and third state, parts of the unknown wind speed, have a greater variance due to the difficulty of modelling the phenomenon, which has a strong random component, while the first term is smaller, as any disturbances are much smaller than the deterministic part predicted by the model. Finally, it is possible to note that the variance of the turbulent component of the wind speed is greater when the average component is greater, furthermore, it depends on the TI term, which indicates the turbulence intensity. In this case, a TI value of 0.18 is adopted in accordance with IEC 614001-1 [32].

At the operational level, as mentioned above, the offsets and matrices of the LTI systems set are calculated offline for each point on the linearisation grid. Starting from these and the considerations/definitions just made, all possible quantities are calculated offline, so that they can be called up during controller operation, knowing, instant by instant, the grid index *ip* to which reference must be made. During controller operation, the inputs to the Kalman filter for wind estimation are the measurements taken (vector $\boldsymbol{y}$ in 6.9), the controlled inputs (vector $\boldsymbol{u}$ in 6.9) and the grid index *ip*. From these, the vectors ${}_{KF}\boldsymbol{x}$ and ${}_{KF}\boldsymbol{u}$ are calculated online, then the time-dependent part of the matrix ${}^{ip}_{KF}\boldsymbol{A}$ and matrix $\boldsymbol{Q}$ are calculated, both dependent on the average part of estimated speed in the previous step. Finally, the Kalman filter algorithm obtained in 5.6.2.1 is applied through which the filter model states are estimated. It is recalled here with a notation compatible with 6.10 and above definitions:

$$\begin{aligned} {}^{ip}_{KF}\hat{\boldsymbol{x}}_{\boldsymbol{k|k-1}} &= {}^{ip}_{KF}\boldsymbol{A} \cdot {}^{ip}_{KF}\hat{\boldsymbol{x}}_{\boldsymbol{k-1}} + {}^{ip}_{KF}\boldsymbol{B} \cdot {}^{ip}_{KF}\boldsymbol{u}_{\boldsymbol{k-1}} \\ \boldsymbol{K_k} &= \boldsymbol{P_{k|k-1}} \cdot {}_{KF}\boldsymbol{C}^{\boldsymbol{T}} \left[{}_{KF}\boldsymbol{C} \cdot \boldsymbol{P_{k|k-1}} \cdot {}_{KF}\boldsymbol{C}^{\boldsymbol{T}} + \boldsymbol{R}\right]^{-1} \\ {}^{ip}_{KF}\hat{\boldsymbol{x}}_{\boldsymbol{k}} &= {}^{ip}_{KF}\hat{\boldsymbol{x}}_{\boldsymbol{k|k-1}} + \boldsymbol{K_k}\left({}^{ip}_{KF}\boldsymbol{y_k} - {}_{KF}\boldsymbol{C} \cdot {}^{ip}_{KF}\hat{\boldsymbol{x}}_{\boldsymbol{k|k-1}}\right) \\ \boldsymbol{P_k} &= (\boldsymbol{I} - \boldsymbol{K_k} \cdot {}_{KF}\boldsymbol{C}) \cdot \boldsymbol{P_{k|k-1}} \\ \boldsymbol{P_{k+1|k}} &= {}^{ip}_{KF}\boldsymbol{A} \cdot \boldsymbol{P_k} \cdot {}^{ip}_{KF}\boldsymbol{A}^{\boldsymbol{T}} + \boldsymbol{Q} \end{aligned} \qquad (6.14)$$

Once ${}^{ip}_{KF}\hat{\boldsymbol{x}}_{\boldsymbol{k}}$ is obtained, the estimation of second and third states ($vw_{t,k}$ and $vw_{m,k} - U_{wind,m,0}$) are summed together, then the offset wind speed ($U_{wind,m,0}$) is added to that sum to obtain estimation of wind speed $U_{wind,m}$.

Below there is a graph showing, to do an example, the results obtained by simulating the non-linear model NL (see section 3.3) and estimating the wind speed with the Kalman filter just obtained. To carry out these tests, since the implementation of the MPC controller under examination has not yet been completed, it was decided to use the Baseline controller presented in section 4.4.1 as control algorithm. The external inputs of the model are the wave excitation forces/torques and the wind speed (see sections 3.2.2 and 3.2.1), while the system characteristics are those reported in section 2.

We recall that the NL model receives, as wind input, the wind speeds at each point of a spatial grid over the rotor area, while the Kalman filter estimate the spatial average wind speed, or, to be more precise, the wind speed which lead to the best fit between measured output (of Kalman filter model) and those one predicted by model, also considering sensor characteristics (matrix ***R***) and model reliability (matrix ***Q***). It is therefore clear that although the algorithm works correctly, there are differences between the real average speed and the estimated speed. In any case, it can be noted that the results obtained are satisfactory.

Since wind speed isn't a priori known, an initial estimate speed has been set (in this case it is 9 m/s), which can also be very different from the actual speed at the initial instant. Despite this, as can be seen from the graph, the estimated speed quickly converges to a value close to the correct one.

To evaluate the estimation performances, a factor called "Goodness-of-Fit", GoF, is computed from simulation results and reported on graph; it is defined as:

$$GoF = 1 - \frac{\sqrt{\sum_{k=1}^{N}(y_k - \hat{y}_k)^2}}{\sqrt{\sum_{k=1}^{N}(y_k - \overline{y_k})^2}} \qquad (6.15)$$

Where $\hat{y}_k$ is the estimation of y_k at time k, $\overline{y_k}$ is the average value of y and N are the number of samples.

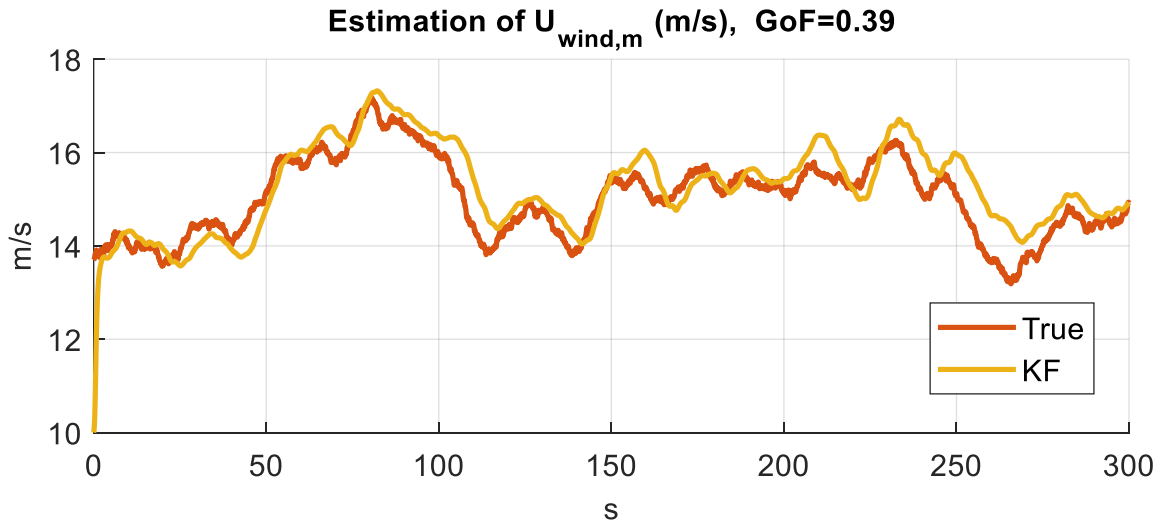

Figure 6.13: Comparison between $U_{wind,m}$, average wind speed across rotor area (mean of wind speed computed at grid points), and its estimation through Kalman filter.

6.2.1.2 Wave Kalman Filter

As with the Kalman filter for wind speed estimation, the LTI systems set (see section 5.2.6.1) is called up from which the construction of the mathematical model adopted in the observer algorithm starts:

$$\begin{cases} \dot{\delta \boldsymbol{x}} = {}^{ip}(\dot{\boldsymbol{x}})_0 + {}^{ip}\boldsymbol{A} \cdot \delta \boldsymbol{x} + {}^{ip}\boldsymbol{B_u} \cdot \delta \boldsymbol{u} + {}^{ip}\boldsymbol{B_v} \cdot \delta \boldsymbol{v} \\ \delta \boldsymbol{y} = {}^{ip}\boldsymbol{C} \cdot \delta \boldsymbol{x} + {}^{ip}\boldsymbol{D} \cdot \delta \boldsymbol{u} \end{cases} \quad (6.16)$$

With:

- $\boldsymbol{x} = \{s, h, p, \dot{s}, \dot{h}, \dot{p}, \Omega\}^\top$,
- $\boldsymbol{y} = \{s, h, p, \dot{s}, \dot{h}, \dot{p}, \Omega, P\}^\top$
- $\boldsymbol{u} = \left\{C'_{\text{gen}}, \theta_{bl}\right\}^\top$,
- $\boldsymbol{v} = \left\{U_{wind,m}, \boldsymbol{F}_{\text{ext}}^T\right\}^\top$
- ${}^{ip}(\dot{\boldsymbol{x}})_0 = f\left({}^{ip}\boldsymbol{x}_0, {}^{ip}\boldsymbol{u}_0, {}^{ip}\boldsymbol{u}_0\right)$,
- ${}^{ip}\boldsymbol{y}_0 = g\left({}^{ip}\boldsymbol{x}_0, {}^{ip}\boldsymbol{u}_0\right)$
- $\delta \boldsymbol{x} = \boldsymbol{x} - {}^{ip}\boldsymbol{x}_0$,
- $\delta \boldsymbol{u} = \boldsymbol{u} - {}^{ip}\boldsymbol{u}_0$
- $\delta \boldsymbol{v} = \boldsymbol{v} - {}^{ip}\boldsymbol{v}_0$,
- $\delta \boldsymbol{y} = \boldsymbol{y} - {}^{ip}\boldsymbol{y}_0$

To be clear, let us remember that the vectors $\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}$ and $\boldsymbol{y}$ are, respectively, the states, controlled inputs, external inputs and outputs vectors. The states are *s* (surge), *h* (heave), *p* (pitch), their derivatives and the rotor speed Ω. The controlled inputs are the low-shaft generator torque C'_{gen} and the collective blade-pitch angle θ_{bl}. The external inputs are the average wind speed (across rotor area) in the direction parallel to the surge, $U_{wind,m}$, and the wave excitation forces $\boldsymbol{F}_{ext}$. Finally, the outputs are the states and the extracted power *P*.

The $\boldsymbol{x}_0, \boldsymbol{u}_0, \boldsymbol{v}_0, \boldsymbol{y}_0$ and $(\dot{\boldsymbol{x}})_0$ vectors, called offsets, are the $\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{y}$ and $\dot{\boldsymbol{x}}$ vectors calculated at the points of the linearisation grid, while the $\boldsymbol{A}, \boldsymbol{B_u}, \boldsymbol{B_v}, \boldsymbol{C}$ and $\boldsymbol{D}$ matrices are the matrices describing the linear system obtained at these points. The superscript ***ip*** is the grid index, which identifies which linear system is being considered among those constituting the LTI systems set. There will be a module in the controller which, based on the values of the parameters chosen to construct the linearisation grid, will work out which system is best to use, obtaining the appropriate ***ip***-index at each step.

From section 5.6.2.1, we recall the state space form of a discrete model used in a linear-discrete Kalman Filter, where the subscript KF means we refer to quantities of Kalman filter model, while no subscript means we refer to LTI systems set quantities or to quantities that only compare in the Kalman filter. Superscript ***ip*** indicates the LTI linear system from which Kalman filter quantities are computed, since at each time step we are going to use a different Kalman filter model according to parameter values. All models are computed offline, so, during working operations, the better one (according to parameter values) is chosen to be used for that time step.

From these considerations, discrete Kalman filter model adopted to estimate waves excitation forces/torques becomes:

$$\begin{cases} {}_{KF}\boldsymbol{x_k} = {}_{KF}\boldsymbol{A} \cdot {}^{ip}_{KF}\boldsymbol{x_{k-1}} + \boldsymbol{G} \cdot \boldsymbol{w_k} \\ {}^{ip}_{KF}\boldsymbol{y_k} = {}^{ip}_{KF}\boldsymbol{C} \cdot {}_{KF}\boldsymbol{x_k} + {}^{ip}_{KF}\boldsymbol{D} \cdot {}^{ip}_{KF}\boldsymbol{u_{k-1}} + \boldsymbol{v_k} \end{cases} \quad (6.17)$$

Where:

- ${}_{KF}\boldsymbol{x}$ **:** It is the Kalman filter states vector**,** which is defined as follows: ${}_{KF}\boldsymbol{x} = \{ \mathrm{F}_{ext,x}/f_x \, , \mathrm{F}_{ext,z}/f_z \, , \mathrm{M}_{ext,y}/f_y \}^{\top}$, where $\mathrm{F}_{ext,x}$ and $\mathrm{F}_{ext,z}$ are the wave excitation forces along x (surge) and z (heave) directions, while $\mathrm{M}_{ext,y}$ is the wave excitation torque around y (pitch) axis. Factors f_x, f_z and f_y are three factors used to scale forces and torque, in order to have similar values, around unity, for the three states.

- ${}_{KF}\boldsymbol{u}$ **:** It is the Kalman filter inputs vector, known by assumption. It is defined as follow: ${}_{KF}\boldsymbol{u} = \{s - s_0,\ h - h_0,\ p - p_0,\ \dot{s} - \dot{s}_0,\ \dot{h} - \dot{h}_0,\ \dot{p} - \dot{p}_0,\ C'_{\text{gen}} - C'_{\text{gen}\,0},\ \theta_{bl} - \theta_{bl0},\ U_{wind,m} - U_{wind,m,0},\ \ddot{s}_0,\ \ddot{h}_0,\ \ddot{p}_0\}^{\top}$. Subscript 0 indicates values at linearization point. All these quantities are considered to be known, although, actually, the first six can be affected by measurement errors, while wind speed is only an estimate (made by the Kalman filter dedicated to it).

- $\boldsymbol{w}$**:** It is the Kalman filter process noise vector. It is a vector $\boldsymbol{w} = \{ w_1, w_2, w_3 \}^{\top}$, whose component are inputs of the system and whose value is normally unknown. In this case they are the "stochastic part" of the states, assumed to be Gaussian processes, which are white noises with zero mean and a certain value of variance;

- ${}_{KF}\boldsymbol{y}$ **:** It is the Kalman filter output vector, which is defined as follows: ${}_{KF}\boldsymbol{y} = \{ \dot{s} - \dot{s}_0,\ \dot{h} - \dot{h}_0,\ \dot{p} - \dot{p}_0 \}$. It is the measured quantity, composed by a deterministic part ($\boldsymbol{Cx} + \boldsymbol{Du}$) and stochastic one due to sensor noise ($\boldsymbol{v}$);

- $\boldsymbol{v}$**:** It is the Kalman filter measurement noise vector**,** the stochastic part of the output's measurement, which is assumed to be a white noise with zero mean and a certain value of variance;

- $\boldsymbol{G}$: It is the process noise matrix; in this case it is defined as the identity matrix $\boldsymbol{I}$, since, by definition, $\boldsymbol{w_k}$ is already the vector defined as the difference between the actual vector of states and that predicted by the mathematical model.

- T: sample time of discrete model, such that, as an example $\boldsymbol{x_k} = \boldsymbol{x}(kT)$**.**

The elements in equation 6.17 that have yet to be defined are the matrices ${}_{KF}\boldsymbol{A}$, ${}^{ip}_{KF}\boldsymbol{C}$ and ${}^{ip}_{KF}\boldsymbol{D}$. Focusing on the matrix ${}_{KF}\boldsymbol{A}$, as can be seen in 6.17, it relates state vectors at time k and at time $k-1$, according with the mathematical model chosen to represent wave forces/torques. In this case, we assume a simple random walk, that is we consider the forces/torque at time k equal to those at time $k-1$, to which is then added a white noise, i.e. a random term, $\boldsymbol{w}$, with zero mean and a certain value of variance, in mathematical terms:

$$ {}_{KF}\boldsymbol{x_k} = {}_{KF}\boldsymbol{x_{k-1}} + \boldsymbol{w_k} \qquad (6.18) $$

It is clear, from equation 6.18 that matrix ${}_{KF}\boldsymbol{A}$ is the identity matrix $\boldsymbol{I}$. As for the matrices C and D, they are derived by manipulating the matrices of LTI systems set 6.16, in particular from the rows corresponding to the equations linking the deviations of the derivatives of $\dot{s}$, $\dot{h}$ and $\dot{p}$ from their values at linearization points to the states vector, $\delta\boldsymbol{x}$, to the inputs ones, $\delta\boldsymbol{u}$ and $\delta\boldsymbol{v}$, and to states derivatives at linearization points, $(\dot{\boldsymbol{x}})_0$.

Finally, lasts quantities must be defined to implement the wave Kalman filter are the matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$, which are the covariance matrices of process and measurements noise ($\boldsymbol{w}$ and $\boldsymbol{v}$). They are defined as diagonal matrices, i.e., it is assumed that two different components of the same vector are statistically uncorrelated, i.e., their covariance is zero. The principal diagonals of the $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrices, on the other hand, are non-zero and are the variances of their components. Given the way in which the wave excitation force model has been defined, i.e., a random walk, one wants to give more importance to the ${}^{ip}_{KF}\boldsymbol{y_k}$ measurements rather than the states' values predicted by the model. This means that the Kalman filter constant $\boldsymbol{K_k}$ (see second relation of 6.19) should be similar to $\boldsymbol{C}^{-1}$. For this to happen, the ${}_{KF}\boldsymbol{C} \cdot \boldsymbol{P_{k|k-1}} \cdot {}_{KF}\boldsymbol{C^T}$ term should be much greater than $\boldsymbol{R}$. Matrix $\boldsymbol{P}$, covariance matrix of 'a priori' error, however, tends to converge quickly to small values with the result that more importance is given to the prediction of states made by the model rather than to the information received from the measurements. To cope with this problem, a so-called forgetting factor, $\boldsymbol{ff}$, has been introduced into the algorithm, which is between 0 and 1 and modifies the way in which the matrix $\boldsymbol{P}$ evolves. This modification results in slower convergence at larger values.

The Kalman filter algorithm resulting from definition and consideration just made, and according with those one obtained in section 5.6.2.1, is:

$$
\begin{gathered}
{}_{KF}\widehat{\boldsymbol{x}}_{\boldsymbol{k|k-1}} = {}_{KF}\widehat{\boldsymbol{x}}_{\boldsymbol{k-1}} \\
\boldsymbol{K_k} = \boldsymbol{P_{k|k-1}} \cdot {}_{KF}\boldsymbol{C^T} \left[{}_{KF}\boldsymbol{C} \cdot \boldsymbol{P_{k|k-1}} \cdot {}_{KF}\boldsymbol{C^T} + \boldsymbol{R} \cdot \boldsymbol{ff} \right]^{-1} \\
{}_{KF}\widehat{\boldsymbol{x}}_{\boldsymbol{k}} = {}_{KF}\widehat{\boldsymbol{x}}_{\boldsymbol{k|k-1}} + \boldsymbol{K_k} \left({}^{ip}_{KF}\boldsymbol{y_k} - {}^{ip}_{KF}\boldsymbol{C} \cdot {}_{KF}\widehat{\boldsymbol{x}}_{\boldsymbol{k|k-1}} - {}^{ip}_{KF}\boldsymbol{D} \cdot {}^{ip}_{KF}\boldsymbol{u_{k-1}} \right) \qquad (6.19) \\
\boldsymbol{P_k} = \left(\boldsymbol{I} - \boldsymbol{K_k} \cdot {}^{ip}_{KF}\boldsymbol{C} \right) \cdot \boldsymbol{P_{k|k-1}} \\
\boldsymbol{P_{k+1|k}} = \boldsymbol{P_k}/\boldsymbol{ff} + \boldsymbol{Q}
\end{gathered}
$$

Once ${}_{KF}\widehat{\boldsymbol{x}}_{\boldsymbol{k}}$ is obtained, the estimation of is computed by multiplying the latter with the scaling factor previously introduced: f_x, f_z and f_y.

Below there are some graphs showing, to do an example, the results obtained by simulating the non-linear model NL (see section 3.3) and estimating the wave excitation forces/torques with the Kalman filter just obtained. To carry out these tests, since the implementation of the MPC controller under examination has not yet been completed, it was decided to use the Baseline controller presented in section 4.4.1 as control algorithm. The external inputs of the model are the wave excitation forces/torques and the wind speed (see sections 3.2.2 and 3.2.1), while the system characteristics are those reported in section 2. Wave forces/torques inputs of simulation are generated through a JONSWAP spectrum with a significative height H_s=4 m and a peak period T_p=6 s.

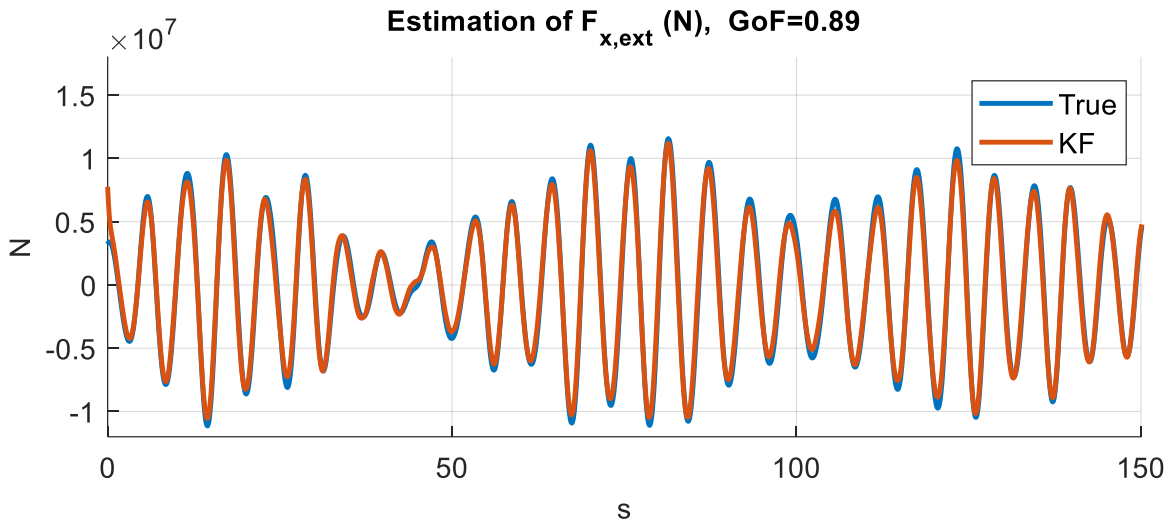

Figure 6.14: Comparison between $F_{ext,x}$, wave excitation force along x (surge) direction, and its estimation obtained through Kalman filter.

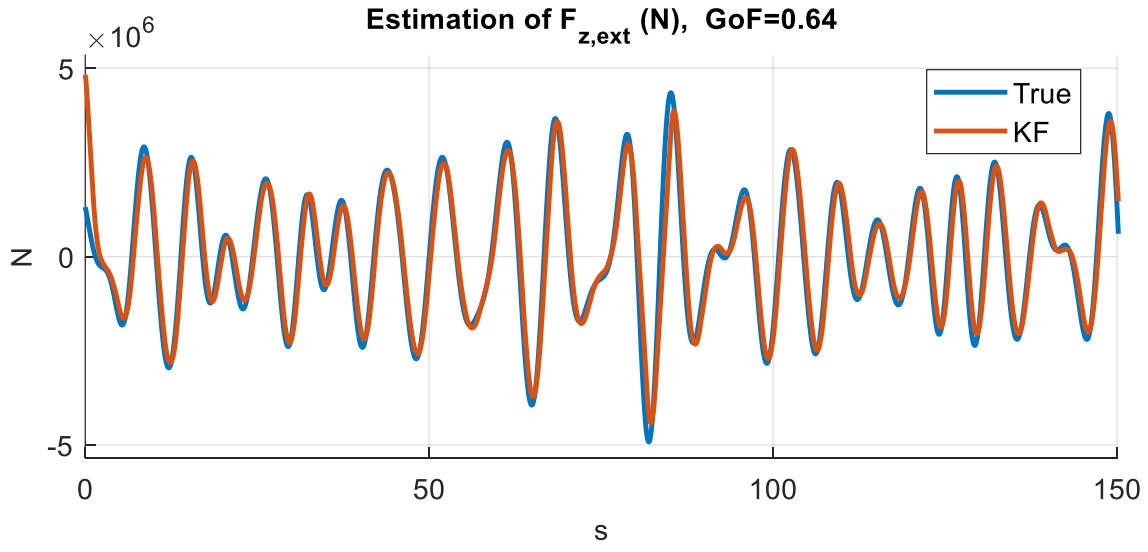

Figure 6.15: Comparison between $F_{ext,z}$, wave excitation force along z (heave) direction, and its estimation obtained through Kalman filter.

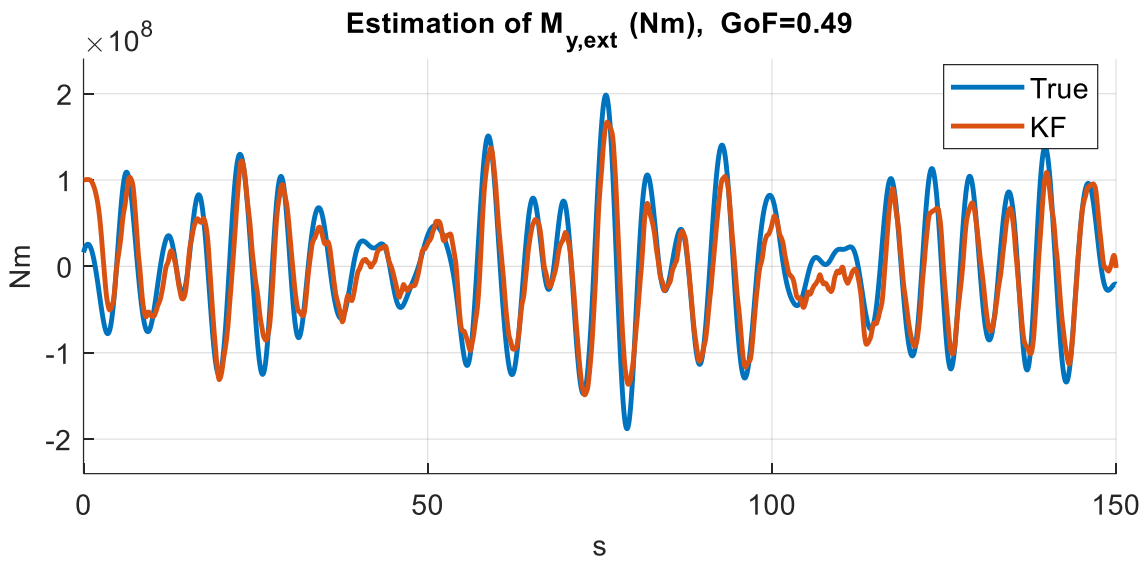

Figure 6.16: Comparison between $M_{ext,y}$, wave excitation torque around y (pitch) axis, and its estimation obtained through Kalman filter.

From these results the fit between the actual and estimated values is quite good, however, the simulated measurements are free of noise and some quantities are defined as input that are not actually known, which means that the results could actually be worse. To obtain an estimate that produces good results even without these simplifications, it may be necessary to use a better model than a simple random walk. One solution might be to model the wave excitation forces as the summation of a certain number of harmonics at a certain frequency, where the states of the system are the amplitudes of these harmonics. The problem with this mode is that the frequency of the harmonics cannot vary and the number of them cannot be too high for reasons of computational onerousness. Another solution could be to reformulate the model in such a way as to have both the amplitudes and the frequencies of a certain number of harmonics as states, or to adopt different types of models, such as autoregressive models in which the value at a certain instant depends, according to coefficients used as states, on the previous values of the estimated quantity. Before arriving at the solution adopted in this work, an attempt was made to implement the first solution mentioned above, which was, however, discarded because the results produced were not sufficiently better to justify the greater complexity and thus the greater computational effort. It was therefore preferred to opt for a filter in which the model produces less reliable results and in which more importance is given to the measurements taken.

6.2.2 Forecasting

As mentioned in the introductory part of this chapter, in order to correctly perform numerical optimisation within the MPC controller for the calculation of optimal control actions, it is necessary to know the value of the model's external inputs used for the controller within the entire prediction horizon. In the previous section, we explored how to estimate their values if they are not measured, whereas in this section, we define how to estimate their values in the future from those in the past (and present). In particular, from the estimates (made by the Kalman filters defined in 6.2.1 and 6.2.2) of the wind speed and wave excitation forces, their values will be forecast throughout the prediction horizon provided by the MPC optimisation algorithm.

Regarding wave excitation forces, the forecasting method chosen is based on an auto-regressive model (AR) and a recursive least square algorithm to estimate model's coefficients. For what concerning wind speed forecasting, on the other hand, an actual forecast is not done, but we simply assume a constant wind speed within prediction horizon, which value corresponds to estimated wind speed at the beginning of the interval. The reason of this choice is the highly stochastic nature of wind speed, especially if the prediction interval is small, as happens in the case of the implemented MPC controller, which deals with time horizons in the order of seconds. Different forecasting methods have been tried, but with poor results, for example the method used for waves or based on ARI (Auto-Regressive with Integrator) models, or still based on the study of signal frequency. Even using the wind model used for the related Kalman filter, this time to perform a forecast, i.e. to estimate the future values of the states based solely on its the model, has not led to good results.

6.2.2.1 Wave Forecasting

As just mentioned, the estimation of the future values of the excitation forces of the waves is performed by means of autoregressive (AR) models and a recursive algorithm for the least squares estimation of the coefficients describing these models. In particular, three separate estimates are made for the two unknown forces and for the unknown torque. An autoregressive model is a discrete model, through which it is possible to estimate a certain quantity at time step t+1 knowing the values in the previous n steps, where n indicates the order of the model, which corresponds to the number of coefficients needed for its description.

The auto-regressive model of order *n*, denoted as AR(*n*), of a general signal y is defined as follow:

$$y(t+1) = \bar{y} + \beta_1 \cdot (y(t) - \bar{y}) + \beta_2 \cdot (y(t-1) - \bar{y}) + \cdots + \beta_n \cdot (y(t-n+1) - \bar{y}) + e(t+1) \quad (6.20)$$

Where $y(t+1)$ is the signal's value at time step t+1 (the same holds for other time steps), $\bar{y}$ is the average value of y , β_i $(\text{with } i = 1, \dots, n)$ are the model coefficients and $e(t+1)$ is the value at time t+1 of a with noise, that is a Gaussian signal with zero mean. The idea of an AR model is that the signal at a certain time step t+1 is the sum of a deterministic part, determined thanks to knowledge of model coefficients and signal's values at previous n steps, and of a random portion, of which only the variance may be known.

From now on, a signal with zero mean is considered ($\bar{y} = 0$), given that the quantities to be estimated (the excitation forces of the waves) actually have a zero mean and that in any case the generality of the treatment is not lost by making this hypothesis. Furthermore, considering that e is a white noise and that signal's values at time steps before that one at which prevision is done are known, the expected value of the signal at the time step t+1, called $\hat{y}(n+1)$, can be written as:

$$\hat{y}(t+1) = E[y(n+1)] = \beta_1 \cdot y(t) + \cdots + \beta_n \cdot y(t-n+1) \qquad (6.21)$$

Equation 6.21 will be used to find future values of y once coefficients vector $\boldsymbol{\beta}$ is known (and with present and past values of y). To obtain $\boldsymbol{\beta}$, a set of equation of the same type of 6.21 can be write, all of them with known past values of y and with vector $\boldsymbol{\beta}$ as the unknown variable. Normally the number of equations, which we define *m*, is higher than the order, so the resulting linear system is an over-dimensioned one, which can be solved by least-squares. To do so, we can define:

$$\boldsymbol{\beta} = \{\beta_1, \cdots, \beta_n\}^\top, \quad \boldsymbol{\beta} \in \mathbb{R}^{n\times 1} \qquad (6.22)$$

$$\boldsymbol{y}(k) = \{y(k-m+1), y(k-m+2), \cdots, y(k)\}^\top, \ \boldsymbol{y}(k) \in \mathbb{R}^{m\times 1} \qquad (6.23)$$

$$\hat{\boldsymbol{y}}(k) = \{\hat{y}(k-m+1), \hat{y}(k-m+2), \cdots, \hat{y}(k)\}^\top, \ \hat{\boldsymbol{y}}(k) \in \mathbb{R}^{m\times 1} \qquad (6.24)$$

$$\boldsymbol{e}(k) = \boldsymbol{y}(k) - \hat{\boldsymbol{y}}(k), \ \boldsymbol{e}(k) \in \mathbb{R}^{m\times 1} \qquad (6.25)$$

$$\boldsymbol{\phi}(k) = \{y(k-1), y(k-2), \cdots, y(k-n)\}, \ \boldsymbol{\phi}(k) \in \mathbb{R}^{1\times n} \qquad (6.26)$$

$$\boldsymbol{\Phi}(k) = \begin{bmatrix} \boldsymbol{\phi}(k-m+1) \\ \boldsymbol{\phi}(k-m+2) \\ \vdots \\ \boldsymbol{\phi}(k) \end{bmatrix}, \boldsymbol{\Phi}(k) \in \mathbb{R}^{m\times n} \qquad (6.27)$$

The goal is to find the vector $\boldsymbol{\beta}$ such that the norm of vector $\boldsymbol{e}$ is minimized, to do so, we define a cost function $J(k)$:

$$J(k) = \frac{1}{2}\, \boldsymbol{e}(k)^\top \boldsymbol{A}(k)\ \boldsymbol{e}(k) \qquad (6.28)$$

Where $\boldsymbol{A}(k)$ is a weight matrix through which we can give less importance to samples more distant in the past. It is a diagonal matrix defined by λ, called forgetting factor, whose value must be between 0 and 1 ($0 < \lambda \leq 1$).

$$\boldsymbol{A}(k) = \begin{pmatrix} \lambda^{m-1} & 0 & \dots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \lambda & 0 \\ 0 & \dots & 0 & 1 \end{pmatrix}, \ \boldsymbol{A}(k) \in \mathbb{R}^{m\times m} \qquad (6.29)$$

By imposing $\frac{\partial J(k)}{\partial \boldsymbol{\beta}} = 0$ we can obtain value of vector $\boldsymbol{\beta}$ which minimize $J(k)$:

$$\boldsymbol{\beta}(k) = (\boldsymbol{\Phi}(k)^\top \cdot \boldsymbol{A}(k) \cdot \boldsymbol{\Phi}(k))^{-1} \cdot \boldsymbol{\Phi}(k)^\top \cdot \boldsymbol{A}(k) \cdot \ \boldsymbol{y}(k) \qquad (6.30)$$

In equation 6.30 we named coefficients vector $\boldsymbol{\beta}(k)$ and not only $\boldsymbol{\beta}$ to underlying it is computed with information at time k, composed by samples from time $k-m-n+1$ to time k. A method can be used to compute $\boldsymbol{\beta}$ at any time step is building $\boldsymbol{\Phi}$, $\boldsymbol{A}(k)$ and $\boldsymbol{y}$ at each time and then calculating $\boldsymbol{\beta}$ with 6.30. This method is called "Batch Least Square", (BLS), and requires the calculation of the inverse of a matrix at each step, as well as the construction of the above-mentioned matrices. However, to reduce computational demand, a different method can be adopted, through which, starting from vector $\boldsymbol{\beta}(k)$ (computed at time step k with, as an example, a BLS method), it is possible to compute coefficient vector at next time step, $\boldsymbol{\beta}(k+1)$, by modifying $\boldsymbol{\beta}(k)$ only with new information available at time step $k+1$. This method is called "Recursive Least Square" (RLS) and it is that one used in this work.

To implement the RLS algorithm some quantities at time step $k+1$ should be defined:

$$\mathbf{y}(k+1) = \begin{bmatrix} \mathbf{y}(k) \\ y(k+1) \end{bmatrix}, \quad \mathbf{y}(k+1) \in \mathbb{R}^{(m+1)\times 1} \tag{6.31}$$

$$\boldsymbol{\Phi}(k+1) = \begin{bmatrix} \boldsymbol{\Phi}(k) \\ \boldsymbol{\phi}(k+1) \end{bmatrix}, \quad \boldsymbol{\Phi}(k+1) \in \mathbb{R}^{(m+1)\times n} \tag{6.32}$$

$$\boldsymbol{A}(k+1) = \begin{bmatrix} \lambda \cdot \mathbf{A}(k) & Z_{m\times 1} \\ Z_{1\times m} & 1 \end{bmatrix}, \ \boldsymbol{A}(k+1) \in \mathbb{R}^{(m+1)\times(m+1)} \tag{6.33}$$

Where $Z_{1\times m}$ is a zero matrix of dimension $1 \times m$. Then we define the matrix $\mathbf{P}$ through its inverse:

$$\mathbf{P}^{-1}(k) = \boldsymbol{\Phi}^{\mathsf{T}}(k)\mathbf{A}(k)\boldsymbol{\Phi}(k), \quad \mathbf{P}(k) \in \mathbb{R}^{n\times n} \tag{6.34}$$

From 6.30-6.34, the inverse of $\mathbf{P}$ at time $k+1$ is:

$$\mathbf{P}^{-1}(k+1) = \boldsymbol{\Phi}^{\mathsf{T}}(k+1)\mathbf{A}(k+1)\boldsymbol{\Phi}(k+1) = \lambda\mathbf{P}^{-1}(k) + \boldsymbol{\Phi}^{\mathsf{T}}(k)\,\boldsymbol{\Phi}(k) \tag{6.35}$$

Matrix $\mathbf{P}(k+1)$ can be obtained by rule $(\mathbf{A}+\mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{DA}^{-1}\mathbf{B} + \mathbf{C}^{-1})^{-1}\mathbf{DA}^{-1}$:

$$\mathbf{P}(k+1) = \frac{1}{\lambda}(\mathbf{P}(k) - \mathbf{P}(k)\cdot\boldsymbol{\phi}^{T}(k+1)\cdot\boldsymbol{\Gamma}(k)\cdot\boldsymbol{\phi}(k+1)\cdot\mathbf{P}(k)) \tag{6.36}$$

With: $\boldsymbol{\Gamma}(k) = (\boldsymbol{\phi}(k+1)\cdot\mathbf{P}(k)\cdot\boldsymbol{\phi}^{T}(k+1) + \lambda)^{-1}, \ \boldsymbol{\Gamma}(k) \in \mathbb{R}$ (6.37)

Now, from 6.30, we can write unknown coefficient vector $\boldsymbol{\beta}$ at time $k+1$:

$$\boldsymbol{\beta}(k+1) = (\boldsymbol{\Phi}(k+1)^{\top} \cdot \boldsymbol{A}(k+1) \cdot \boldsymbol{\Phi}(k+1))^{-1} \cdot \boldsymbol{\Phi}(k+1)^{\top} \cdot \boldsymbol{A}(k+1) \cdot \boldsymbol{y}(k+1) \qquad (6.38)$$

Substituting equation 6.35 and applying definitions 6.31-6.33 we have:

$$\boldsymbol{\beta}(k+1) = \mathbf{P}(k+1) \cdot \left(\lambda\, \boldsymbol{\Phi}^{\top}(k) \cdot \mathbf{A}(k) \cdot \mathbf{y}\,(k) + \boldsymbol{\phi}^{\top}(k+1) \cdot y(k+1)\right) \qquad (6.39)$$

Finally, by further developing equation 6.39, we arrive at the following final expressions:

$$\boldsymbol{\beta}(k+1) = \boldsymbol{\beta}(k) + \mathbf{P}(k+1) \cdot \boldsymbol{\phi}^{\top}(k+1) \cdot \left(y(k+1) - \boldsymbol{\phi}\,(k+1)\boldsymbol{\beta}(k)\right) \qquad (6.40)$$

$$\mathbf{P}(k+1) = \frac{\mathbf{P}(k)}{\boldsymbol{\phi}\,(k+1) \cdot \mathbf{P}(k) \cdot \boldsymbol{\phi}^{T}(k+1) + \lambda} \qquad (6.41)$$

To summarise: assuming we are at time $k+1$ and knowing the values of $y(k+1)$, $\boldsymbol{\phi}\,(k+1)$ and, $\mathbf{P}(k)$; it is possible to calculate the updated values of the coefficients $\boldsymbol{\beta}(k+1)$ of the AR auto-regressive model using 6.40 and 6.41. Having known $\boldsymbol{\beta}(k+1)$, it is then possible to calculate the expected values of y at subsequent instants $(y(k+2), y(k+3), \ldots etc.)$ with expression 6.21.

At the operational level, the forecasting algorithm initially accumulates the estimates produced by the Kalman filter until enough equations are available for estimation. When this condition is reached, $\mathbf{P}$ and $\boldsymbol{\beta}$ are calculated using the BLS method (Equation 6.30 and 6.34). After that, from the next step, $\boldsymbol{\beta}$ is calculated using 6.40 and 6.41. As for future estimates of wave forces, these are calculated with 6.21 from the first instant at which $\boldsymbol{\beta}$ is available (they are set to zero before that time).

We have seen that by means of the previous samples, the coefficients of a model can be estimated by which the future values of the sampled signal can be predicted, i.e., it is possible to derive a model describing the signal. It is therefore natural to think of exploiting this model in the observer dedicated to the estimation of wave forces in order to try to improve the overall performance and to integrate the two modules together. By doing so, one could structure the filter differently to give more confidence to the model's predictions, thus having fewer problems in the case of incorrect measurements. However, the first attempts in this direction proved to be inconclusive because instability problems arose, since an incorrect model leads to incorrect values of the observer's estimate, leading itself to models that are even further away from the correct one.

Below there are, to do an example, some graphs representing the results produced by the wave excitation force forecasting module. They come from a simulation of the non-linear model NL (see section 3.3) with a Baseline controller (see section 4.4.1), since the implementation of the MPC controller under examination has not yet been completed. The external inputs of the model are the wave excitation forces/torques and the wind speed (see sections 3.2.2 and 3.2.1), while the system characteristics are those reported in Section 2. Wave forces/torques inputs of simulation are generated through a JONSWAP spectrum with a significative height H_s=4 m and a peak period T_p=6 s. Average wind speed at hub height is 7 m/s.

The main choices made for the forecasting algorithm are: a 30th-order AR model (n=30), a sample time of 0.2 s, 60 equation to do the first estimation of $\boldsymbol{\beta}$ (m=60) and a forgetting factor λ =0.996.

The graphs show "present" values estimated by the Kalman filter (grey curve) and "future" values computed at the starting point of each line (coloured lines). In this case a prediction horizon of 6 seconds is chosen. Moreover, only some of the previsions are represented, but during real operations, a new computation is done every controller optimization, since we will adopt the receding horizon method (see section 5).

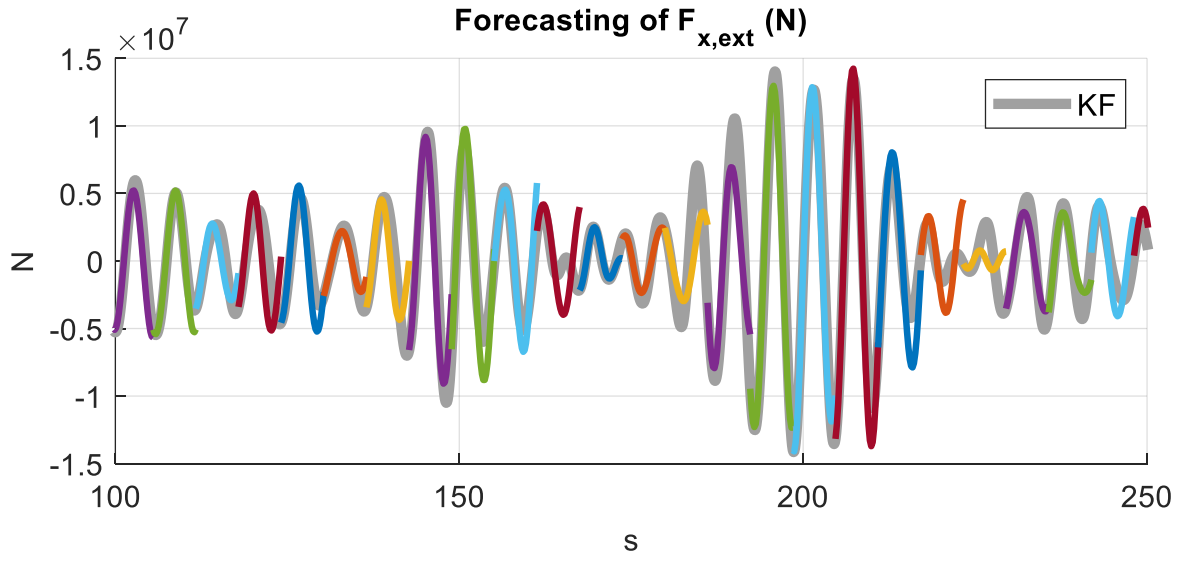

Figure 6.17: Predicted values of $F_{ext,x}$, wave excitation force along x (surge) direction. Grey line shows "present" values estimated by dedicated Kalman filter, whereas colored plots are the predicted values within prediction horizon. Starting point of those plots are the time step at which the computation of future value is done.

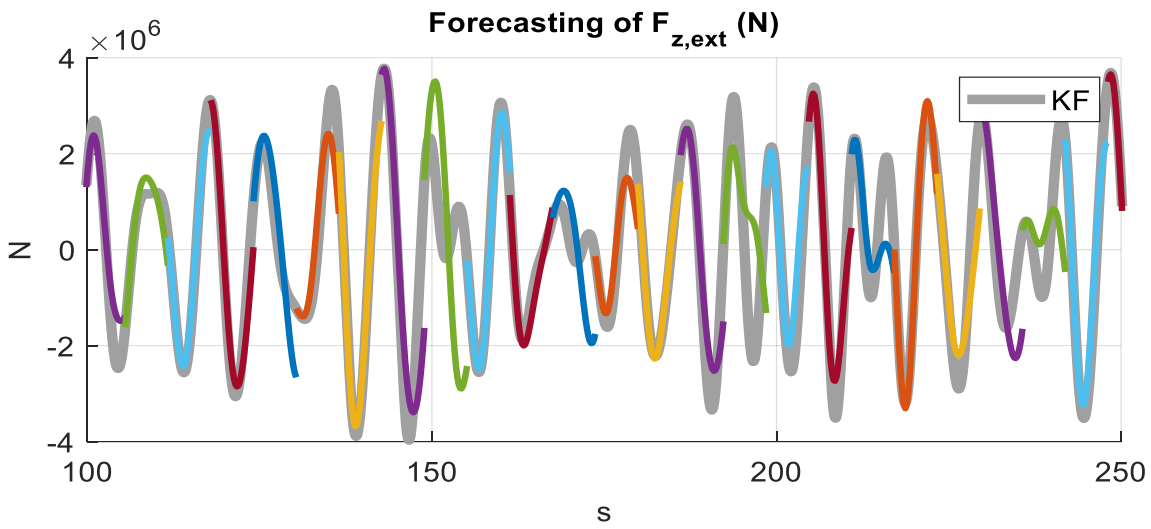

Figure 6.18: Predicted values of $F_{ext,z}$, wave excitation force along z (heave) direction. Grey line shows "present" values estimated by dedicated Kalman filter, whereas colored plots are the predicted values within prediction horizon. Starting point of those plots are the time step at which the computation of future value is done.

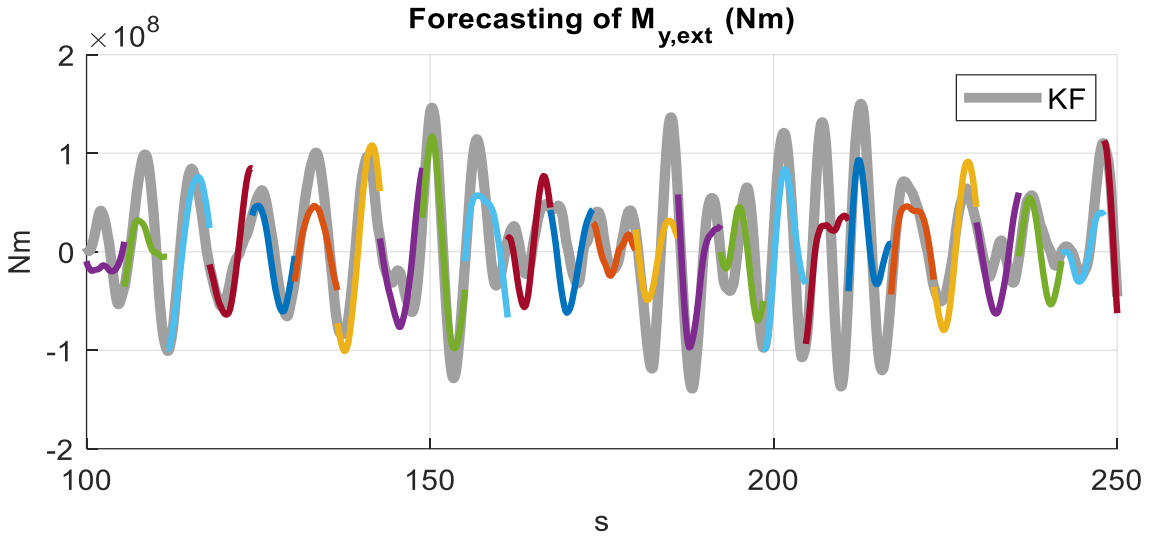

Figure 6.19: Predicted values of $M_{ext,y}$, wave excitation torque around y (pitch) axis. Grey line shows "present" values estimated by dedicated Kalman filter, whereas colored plots are the predicted values within prediction horizon. Starting point of those plots are the time step at which the computation of future value is done.

From the graphs it can be seen that the results are qualitatively correct, i.e., the prediction is successful, but with not too precise fits, especially for of $M_{ext,y}$, for which mean value of GoF may be under 0.5 (see definition 6.15). However, we must consider that, given that a receding horizon method will be adopted for our MPC controller (see section 5), the important thing is to have good results in the firsts instants of the future. Moreover, optimal knowledge of these forces does not lead to a significant improvement in controller performance, which is why suboptimal prediction performance does not lead to major problems.

6.2.4 Optimization

As mentioned in the introductory part of this chapter, now that the various ingredients required to be able to implement a model predictive control have been defined and explained, it is possible to reconnect with the basic concepts presented in Chapter 5 in order to define the control algorithm that is the subject of this work, which is used for the management of floating wind turbines. In particular, in this section we will define the mathematical model used for numerical optimisation and the cost function, which determines the objectives to be pursued.

At first, the mathematical model of the system used for the optimisation module will be defined, after which the cost function and constraints will be defined, and finally, the method by which the optimal solution will be found will be explained.

6.2.4.1 MPC Model

In Chapter 5, we saw that a fundamental element of MPC controllers is the mathematical model that describes the system to be controlled. As mentioned earlier, the entire control algorithm is based on a set of linear systems, each of which is valid around certain operating conditions. This set consists of a number of linear systems, each of which describes the system well when certain quantities (called parameters) have certain values, close to those used to linearise the starting non-linear system. The idea is therefore to create a set of models for the MPC algorithm and to use the most appropriate one each time the optimisation is to be performed. To do this, the starting point is the LTI systems set obtained in section 6.2.1, through the linearization of the non-linear model of the system called NLS, defined in section 3.4. By adding the LTI subscript, to distinguish itself from the other models that will be defined, the LTI systems set model is defined as follows (see equations 6.1 and 6.8):

$$\begin{cases} \delta\dot{\boldsymbol{x}}_{LTI} = {}^{ip}\boldsymbol{A}_{LTI} \cdot \delta\boldsymbol{x}_{LTI} + {}^{ip}\boldsymbol{B}_{\boldsymbol{u}\,LTI} \cdot \delta\boldsymbol{u}_{LTI} + {}^{ip}\boldsymbol{B}_{\boldsymbol{v}\,LTI} \cdot \delta\boldsymbol{v}_{LTI} \\ \delta\boldsymbol{y}_{LTI} = {}^{ip}\boldsymbol{C}_{LTI} \cdot \delta\boldsymbol{x}_{LTI} + {}^{ip}\boldsymbol{D}_{LTI} \cdot \delta\boldsymbol{u}_{LTI} \end{cases} \quad (6.42)$$

With:

- $\boldsymbol{x}_{LTI} = \{s, h, p, \dot{s}, \dot{h}, \dot{p}, \Omega\}^{\top}$,
- $\boldsymbol{y}_{LTI} = \{s, h, p, \dot{s}, \dot{h}, \dot{p}, \Omega, P\}^{\top}$
- $\boldsymbol{u}_{LTI} = \left\{C'_{\text{gen}}, \theta_{bl}\right\}^{\top}$,
- $\boldsymbol{v}_{LTI} = \left\{U_{wind,m}, \boldsymbol{F}^{T}_{\text{ext}}\right\}^{\top}$
- ${}^{ip}(\dot{\boldsymbol{x}})_0 = f\left({}^{ip}\boldsymbol{x}_0, {}^{ip}\boldsymbol{u}_0, {}^{ip}\boldsymbol{u}_0\right)$,
- ${}^{ip}\boldsymbol{y}_0 = g\left({}^{ip}\boldsymbol{x}_0, {}^{ip}\boldsymbol{u}_0\right)$
- $\delta\boldsymbol{x}_{LTI} = \boldsymbol{x}_{LTI} - {}^{ip}\boldsymbol{x}_0$,
- $\delta\boldsymbol{u}_{LTI} = \boldsymbol{u}_{LTI} - {}^{ip}\boldsymbol{u}_0$
- $\delta\boldsymbol{v}_{LTI} = \boldsymbol{v}_{LTI} - {}^{ip}\boldsymbol{v}_0$,
- $\delta\boldsymbol{y}_{LTI} = \boldsymbol{y}_{LTI} - {}^{ip}\boldsymbol{y}_0$

We recall the vectors $\boldsymbol{x}_{LTI}, \boldsymbol{u}_{LTI}, \boldsymbol{v}_{LTI}$ and $\boldsymbol{y}_{LTI}$ are, respectively, the states, controlled inputs, external inputs and outputs vectors. The states are *s* (surge), *h* (heave), *p* (pitch), their derivatives and the rotor speed Ω.

The controlled inputs are the generator torque C'_{gen} and the collective blade-pitch angle θ_{bl}. The external inputs are the average wind speed (across rotor area) in the direction parallel to the surge, $U_{wind,m}$, and the wave excitation forces $\boldsymbol{F}_{ext}$. Finally, the outputs are the states and the extracted power *P*.

The $\boldsymbol{x}_0, \boldsymbol{u}_0, \boldsymbol{v}_0$ and $\boldsymbol{y}_0$vectors, called offsets, are the $\boldsymbol{x}_{LTI}, \boldsymbol{u}_{LTI}, \boldsymbol{v}_{LTI}$ and $\boldsymbol{y}_{LTI}$ vectors calculated at the points of the linearisation grid, which do not have LTI subscript since they compare only in the LTI set. ${}^{ip}\boldsymbol{A}_{LTI}$, ${}^{ip}\boldsymbol{B}_{\boldsymbol{u}\,LTI}$, ${}^{ip}\boldsymbol{B}_{\boldsymbol{v}\,LTI}$, ${}^{ip}\boldsymbol{C}_{LTI}$ and ${}^{ip}\boldsymbol{D}_{LTI}$ matrices are the matrices describing the linear system obtained at these points. The superscript ***ip*** is the grid index, which identifies which linear system is being considered among those constituting the LTI systems set. As it can be noted comparing equation 6.42 with 6.8, the vector $(\dot{\boldsymbol{x}})_0$ (i.e. values of states' derivatives at linearization points) is now omitted for sake of simplicity, since it will be anyhow simplified in the next calculations.

Having defined the starting LTI systems set, it is now possible to define the mathematical model used for the optimization of the MPC controller, indicated with the subscript m. According to equation 5.1, the linear model in the state space form is shown below. It is somewhat different from the one defined in that section: according to 6.42, controlled inputs are distinguished from external ones. Furthermore, as explained in Chapter 5, the matrix $\boldsymbol{D}_{m,c}$ is omitted for reasons of causality, given that in this case the control law $\boldsymbol{u_m}$ in a certain instant will have to depend on the outputs (and therefore on the states) in that instant. From now on the apex ip will be omitted for simplicity, however it must be remembered that each model that will be defined will be one of many (defined in a similar way), each of which is valid around a given linearization point, i.e., valid for a certain value of the chosen parameters and therefore a certain $\boldsymbol{ip}$ index.

$$\begin{cases} \dot{\boldsymbol{x}}_{\boldsymbol{m}}(t) = \boldsymbol{A}_{m,c}\,\boldsymbol{x}_{\boldsymbol{m}}(t) + \boldsymbol{B}_{\boldsymbol{u}_{m,c}}\,\boldsymbol{u}_{\boldsymbol{m}}(t) + \boldsymbol{B}_{\boldsymbol{v}_{m,c}}\,\boldsymbol{v}_{\boldsymbol{m}}(t) \\ \boldsymbol{y}_{\boldsymbol{m}}(t) = \boldsymbol{C}_{m,c}\,\boldsymbol{x}_{\boldsymbol{m}}(t) \end{cases} \tag{6.43}$$

Where:

$$\boldsymbol{x_m} = \left\{\delta \boldsymbol{x}_{LTI},\ \int_0^t P - P_0\, dt\right\}^{\boldsymbol{T}}$$

$$\boldsymbol{u_m} = \delta \boldsymbol{u}_{LTI}$$

$$\boldsymbol{v_m} = \delta \boldsymbol{v}_{LTI}$$

$$\boldsymbol{y_m} = \boldsymbol{x_m}$$

From definition above, and from 6.42, matrices of model 6.43 can be defined as follows ($\boldsymbol{Z}$ are zero-matrices of needed dimensions):

$$\boldsymbol{A}_{m,c} = \begin{bmatrix} \boldsymbol{A}_{LTI} & \boldsymbol{Z} \\ \boldsymbol{C}_{LTI}(end,:) & \boldsymbol{Z} \end{bmatrix} \tag{6.44}$$

$$\boldsymbol{B}_{\boldsymbol{u}_{m,c}} = \begin{bmatrix} \boldsymbol{B}_{\boldsymbol{u}\,LTI} \\ \boldsymbol{D}_{LTI}(end,:) \end{bmatrix} \tag{6.45}$$

$$\boldsymbol{B}_{\boldsymbol{u}_{m,c}} = \begin{bmatrix} \boldsymbol{B}_{\boldsymbol{v}\,LTI} \\ \boldsymbol{Z} \end{bmatrix} \quad (6.46)$$

$$\boldsymbol{C}_{m,c} = \boldsymbol{I} \quad (6.47)$$

The reason why last state of the system is the integral of difference between extracted power and its linearization value, $\int_0^t P - P_0\, dt$, instead of $P - P_0$, is linked with causality issue above mentioned, since $P - P_0$ also depends on $\boldsymbol{u_m}$. It can't be only an output, so it would have to compare as a state, and to do so, since we have to find a relation between state derivative with states and inputs, we have chosen to consider integral $\int_0^t P - P_0\, dt$ as a state, such that its derivative, $P - P_0$, can be expressed in terms of $\boldsymbol{x_m}$ and $\boldsymbol{u_m}$. Reasoning in terms of the discrete system, with this method the integral at time step $k+1$ is given by the sum of the integral at time step k and the quantity $(P - P_0) \cdot T$ at time step k, i.e. the forward Euler method is used.

Once the continuous state space model adopted for MPC optimization is obtained, it can be discretized and then augmented according with what has been explained in section 5.2.1. Following system is the discrete form of 5.43, with discretization time T such that, as an example, $\boldsymbol{x_m}^k = \boldsymbol{x_m}(kT)$ and $\boldsymbol{x_m}^{k+1} = \boldsymbol{x_m}\big((k+1)\cdot T\big)$. Discretization method which was used is ZOH (zero-older-hold).

$$\begin{cases} \boldsymbol{x_m}^{k+1} = \boldsymbol{A}_{m,d}\, \boldsymbol{x_m}^k + \boldsymbol{B}_{\boldsymbol{u}_{m,d}}\, \boldsymbol{u_m}^k + \boldsymbol{B}_{\boldsymbol{v}_{m,d}}\, \boldsymbol{v_m}^k \\ \boldsymbol{y_m}^k = \boldsymbol{C}_{m,d}\, \boldsymbol{x_m}^k \end{cases} \quad (6.48)$$

Finally, augmented system is:

$$\begin{cases} \boldsymbol{x}^{k+1} = \boldsymbol{A}\, \boldsymbol{x}^k + \boldsymbol{B_u}\, \Delta \boldsymbol{u}^k + \boldsymbol{B_v}\, \Delta \boldsymbol{v}^k \\ \boldsymbol{y}^k = \boldsymbol{C}\, \boldsymbol{x}^k \end{cases} \quad (6.49)$$

Where following definitions hold:

$$\boldsymbol{x}^k = \left[\Delta \boldsymbol{x_m}^{k\,T} \;\; \boldsymbol{y_m}^{k\,T}\right]^T \quad (6.50)$$

$$\boldsymbol{y}^k = \boldsymbol{y_m}^k \quad (6.51)$$

$$\Delta \boldsymbol{x_m}^k = \boldsymbol{x_m}^k - \boldsymbol{x_m}^{k-1} \quad (6.52)$$

$$\Delta \boldsymbol{u}^k = \boldsymbol{u_m}^k - \boldsymbol{u_m}^{k-1} \quad (6.53)$$

$$\Delta \boldsymbol{v}^k = \boldsymbol{v_m}^k - \boldsymbol{v_m}^{k-1} \quad (6.54)$$

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_{m,d} & \mathbf{Z} \\ \boldsymbol{C}_{m,d}\,\boldsymbol{A}_{m,d} & \boldsymbol{I} \end{bmatrix} \qquad (6.55)$$

$$\boldsymbol{B_u} = \begin{bmatrix} \boldsymbol{B_u}_{m,d} \\ \boldsymbol{C}_{m,d}\,\boldsymbol{B_u}_{m,d} \end{bmatrix} \qquad (6.56)$$

$$\boldsymbol{B_v} = \begin{bmatrix} \boldsymbol{B_v}_{m,d} \\ \boldsymbol{C}_{m,d}\,\boldsymbol{B_v}_{m,d} \end{bmatrix} \qquad (6.57)$$

$$\boldsymbol{C} = [\boldsymbol{Z} \quad \boldsymbol{I}] \qquad (6.58)$$

The difference between this system with that one derived in section 5.2.1 is the separation of input vector in two vectors to decouple controlled and external inputs. As can be noted from above definitions, they are mathematically treated in the same way, reason why mathematical steps to obtain augmented system matrices are not given here.

Once augmented model is derived, we can define vectors containing future values of controlled inputs, $\Delta\boldsymbol{U}$, external inputs, $\Delta\boldsymbol{V}$, and outputs predicted by the model, $\boldsymbol{Y}$**:**

$$\Delta\boldsymbol{U} = \left[\Delta\boldsymbol{u}^{k^T}, \Delta\boldsymbol{u}^{k+1^T}, \dots, \Delta\boldsymbol{u}^{k+N_c-1^T}\right]^T,\ \Delta\boldsymbol{U} \in \mathbb{R}^{(N_c \cdot \mathrm{mu})\,\mathrm{x}\,1} \qquad (6.59)$$

$$\Delta\boldsymbol{V} = \left[\Delta\boldsymbol{v}^{k^T}, \Delta\boldsymbol{v}^{k+1^T}, \dots, \Delta\boldsymbol{v}^{k+N_p-1^T}\right]^T,\ \Delta\boldsymbol{V} \in \mathbb{R}^{(N_p \cdot \mathrm{mv})\,\mathrm{x}\,1} \qquad (6.60)$$

$$\boldsymbol{Y} = \left[\boldsymbol{y}^{k+1^T}, \boldsymbol{y}^{k+2^T}, \dots, \boldsymbol{y}^{k+N_p{}^T}\right]^T,\ \boldsymbol{Y} \in \mathbb{R}^{(N_p \cdot \mathrm{q})\,\mathrm{x}\,1} \qquad (6.61)$$

Outputs and external inputs vectors contain N_p elements, to cover the prediction horizon H_p with a sample time T (corresponding to sample time used to obtain discrete model from continuous one), while controlled inputs vector contains N_c elements, to cover the control horizon H_c with the same sample time. Furthermore, we assume that the control variables remain constant over the time span between the control and prediction horizons, so, remembering relation 6.53, $\Delta\boldsymbol{u}$ is set to zero over this time. mu, mv and q are, respectively, the number of controlled inputs, external inputs and outputs.

Remembering relations 5.32, we can express $\boldsymbol{Y}$ in terms of current state vector at time step k, $\boldsymbol{x}^k$, future control movements $\Delta\boldsymbol{U}$ and future external inputs $\Delta\boldsymbol{V}$:

$$\boldsymbol{Y} = \boldsymbol{F} \cdot \boldsymbol{x}^k + \boldsymbol{\Phi_u} \cdot \Delta\boldsymbol{U} + \boldsymbol{\Phi_v} \cdot \Delta\boldsymbol{V} \qquad (6.62)$$

Where matrices $\boldsymbol{F}$ and $\boldsymbol{\Phi_u}$ are those obtained in 5.33 and 5.34, while $\boldsymbol{\Phi_v}$ can be derived following the same way of $\boldsymbol{\Phi_u}$. Matrix $\boldsymbol{F}$ has dimension $q \cdot N_p \times n$, $\boldsymbol{\Phi_u}$ has dimension $q \cdot N_p \times mu \cdot N_c$ and $\boldsymbol{\Phi_v}$ has dimension $q \cdot N_p \times mv \cdot N_p$ (n is the dimension of state vector $\boldsymbol{x}^k$).

6.2.4.2 Cost Function

In this section, we will define the cost function to be minimised to find the vector of future controls U. It is based on the cost function defined in section 5.2.2, which set as its objective the minimum difference between the model's predicted outputs and the imposed targets, and the minimisation of the rate of change of the inputs. Weight matrices were also defined to give different importance to different targets. In this case, the cost function is slightly different, in fact, a term is added by which we try to have the controlled inputs close to the corresponding reference values. The cost function used is defined as follows:

$$J = \frac{1}{2}\left[(\boldsymbol{T_Y} - \boldsymbol{Y})^T \cdot \boldsymbol{W_Y} \cdot (\boldsymbol{T_Y} - \boldsymbol{Y}) + \Delta \boldsymbol{U}^T \cdot \boldsymbol{W_{\Delta U}} \cdot \Delta \boldsymbol{U} + (\boldsymbol{T_U} - \boldsymbol{U})^T \cdot \boldsymbol{W_U} \cdot (\boldsymbol{T_U} - \boldsymbol{U})\right] \qquad (6.63)$$

$\Delta \boldsymbol{U}$ and $\boldsymbol{Y}$ are those defined in 6.59 and 6.51, while $\boldsymbol{T_Y}$, $\boldsymbol{T_U}$ and $\boldsymbol{U}$ are:

$$\boldsymbol{T_Y} = \left[{\boldsymbol{t_y}^{k+1}}^T, {\boldsymbol{t_y}^{k+2}}^T, \dots, {\boldsymbol{t_y}^{k+N_p}}^T \right]^T, \ \boldsymbol{T_Y} \in \mathbb{R}^{(N_p \cdot \mathrm{q}) \times 1} \qquad (6.64)$$

$$\boldsymbol{T_U} = \left[{\boldsymbol{t_U}^{k+1}}^T, {\boldsymbol{t_U}^{k+2}}^T, \dots, {\boldsymbol{t_U}^{k+N_p}}^T \right]^T, \ \boldsymbol{T_U} \in \mathbb{R}^{(N_c \cdot \mathrm{mu}) \times 1} \qquad (6.65)$$

$$\boldsymbol{U} = \left[{\boldsymbol{u_m}^k}^T, {\boldsymbol{u_m}^{k+1}}^T, \dots \ {\boldsymbol{u_m}^{k+N_c-1}}^T \right]^T, \ \boldsymbol{U} \in \mathbb{R}^{(N_c \cdot \mathrm{mu}) \times 1} \qquad (6.66)$$

Vectors $\boldsymbol{t_y}$ are output target vectors ($\in \mathbb{R}^{q \times 1}$) for each time step within prediction horizon, while vectors $\boldsymbol{t_U}$ are controlled input reference vectors ($\in \mathbb{R}^{mu \times 1}$) for each time step within control horizon. Matrices $\boldsymbol{W_Y}$ ($\in \mathbb{R}^{(N_p \cdot \mathrm{q}) \times (N_p \cdot \mathrm{q})}$), $\boldsymbol{W_{\Delta U}}$ ($\in \mathbb{R}^{(N_c \cdot \mathrm{mu}) \times (N_c \cdot \mathrm{mu})}$) and $\boldsymbol{W_U}$ ($\in \mathbb{R}^{(N_c \cdot \mathrm{mu}) \times (N_c \cdot \mathrm{mu})}$) are diagonal weight matrices through which different importance can be given to output and input targets and to the rate of change of inputs.

Vector $\boldsymbol{U}$ can be expressed in terms of vector $\Delta \boldsymbol{U}$ and vector ${\boldsymbol{u_m}^k}$, as we have seen in section 5.2.5.1:

$$\boldsymbol{U} = \boldsymbol{C_1} \cdot {\boldsymbol{u_m}^{k-1}} + \boldsymbol{C_2} \cdot \Delta \boldsymbol{U} \qquad (6.67)$$

With:

$$\boldsymbol{C_1} = \begin{bmatrix} I_m \\ I_m \\ I_m \\ \vdots \\ I_m \end{bmatrix}$$

$$\boldsymbol{C_2} = \begin{bmatrix} I_m & 0 & 0 & \dots & 0 \\ I_m & I_m & 0 & \dots & 0 \\ I_m & I_m & I_m & \dots & 0 \\ \vdots & & & & \\ I_m & I_m & \dots & I_m & I_m \end{bmatrix}$$

Targets T_Y and T_U

The targets T_Y and T_U are considered constant along the corresponding time horizons and their value depends on the wind speed $U_{wind,m}$ at the initial instant of these horizons. The trend of the targets with respect to wind speed is that of the steady state values dictated by the control strategy. A strategy close to that of the ROSCO controller presented in section 4.4.2.1 was chosen, in which the possible wind speed range was divided into 4 main regions. Regions 1 and 4 are inherent below the cut-in speed and above the cut-out speed. The former involves zero generator torque and rotor start-up, while the latter involves the feathering manoeuvre, with maximum blade angle to reduce loads. Region 2 is the constant TSR ratio region at its optimum value, while region 3 is the constant power region at its rated value. Finally, regions 1.5 and 2.5 are connection regions.

The image below shows the values of the T_Y and T_U targets versus wind speed according to the control strategy adopted. Some of the T_Y targets are not represented, essentially for two reasons: the targets concerning the outputs s (surge), h (heave) and p (pitch) are not represented because their weight will be considered null, because no particular goals are set for these quantities. Concerning the outputs $\dot{s}, \dot{h}$ and $\dot{p}$, the corresponding targets are set to zero for each wind speed, as the aim is to stabilise the structure.

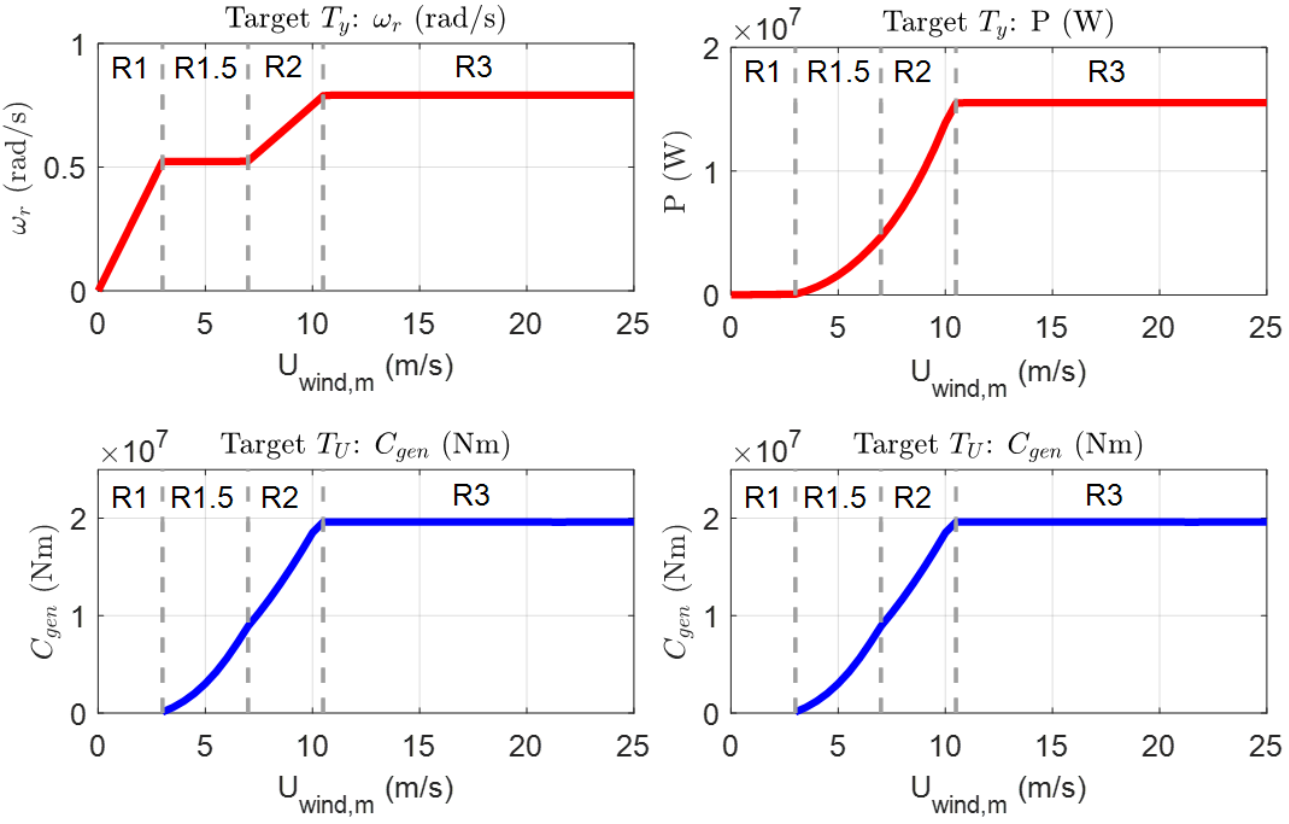

Figure 6.20: Targets T_Y and T_U with respect to average wind speed across rotor area $U_{wind,m}$.

Weight Matrices $\boldsymbol{W_Y}$, $\boldsymbol{W_{\Delta U}}$ and $\boldsymbol{W_U}$

It was seen in the previous sections that the model of the system used for the MPC changes with each optimisation performed. At each step, the model chosen is the appropriate LTI one, i.e. that one obtained under the conditions closest to those in which the real system is. The $\boldsymbol{W_Y}$, $\boldsymbol{W_{\Delta U}}$ and $\boldsymbol{W_U}$ matrices also change according to this logic, so it is possible to give different weights according to the region of the linearisation grid in which the system is situated.

Specifically, for each matrix, a base matrix is defined, after which it is multiplied by factors dependent on the zone in which plant is located. In this case, the parameters used to create the grid are wind speed, collective blade angle and generator torque. The first and third parameters are divided into three zones, while the factor related to the second parameter is always 1. However, there could be as many regions as the number of linearisation grid points.

The chosen zones for wind speed $U_{wind,m}$ are:

- **Zone 1**: $U_{wind,m} < 9$ m/s;
- **Zone 2**: 9 m/s $\leq U_{wind,m} < 11$ m/s;
- **Zone 3**: 11 m/s $\leq U_{wind,m}$.

Considering $\boldsymbol{W_Y}$ matrix, factors other than 1 were only chosen for the rotor speed and power targets. In the first and second zones, similar weights are given to the speed and power targets. In the third zone, on the other hand, a higher weight is given to the power target, while maintaining a non-zero weight for the speed target.

Considering $\boldsymbol{W_{\Delta U}}$ matrix, in the first zone, a high weight is given to the rate of change of the collective bladepitch angle, because we want to operate more through the generator torque, vice versa for zone 3. In zone 2, on the other hand, approximately equal weight is given to the two rates of change.

Finally, regarding the $\boldsymbol{W_U}$ matrix, it was chosen to give a very high weight to the target related to the blade pitch in the first region, so as to have the angle value very close to the target one because in this region we want to operate by varying the generator torque. Vice versa for region 3, where it is preferred to operate via the blade pitch.

The chosen zones for rotor speed Ω are:

- **Zone 1**: $\Omega < \Omega_t - 0.1\, rad/s$
- **Zone 2**: $\Omega_t - 0.1\, rad/s \leq \Omega < \Omega_t + 0.1\, rad/s$;
- **Zone 3**: $\Omega_t + 0.1\, rad/s \leq \Omega$.

Where Ω_t is the target rotor speed according with specification mentioned in the previous subsection. The strategy is to give more weight to the speed target and less weight to the power target when rotor speed is far from its target value, in order to be sure not to go too far from it. Moreover, weights related to inputs and their rate of chance are reduced in these zones to reach target rotor speed in a more effective way.

All these choices are made in accordance with traditional wind turbine control strategies, discussed during the presentation of the Baseline and ROSCO controllers, in which the aim is to optimise the power tracking the correct rotor speed by mainly varying the generator torque when the wind speed is lower than nominal one. On the contrary, when the wind speed is higher than rated, we want to operate more through the blade pitch trying to maintain the power at its rated value, giving less importance to the rotor speed. Furthermore, the combination of the weights of the different arrays is designed to reduce conflicts between the two actuations.

6.2.4.3 Constraints

Implemented MPC considers constrain on both controlled inputs $\boldsymbol{U}$, their rate of change $\Delta\boldsymbol{U}$ and on predicted outputs $\boldsymbol{Y}$, and they can be equality or inequality constrain. In accordance with what was obtained in section 5.2.5.1, where we defined how to express the constraints, and adapting equation 5.70 to this case, where there are also external inputs, we arrive at the following expression:

$$\boldsymbol{M} \cdot \Delta\boldsymbol{U} \leq \boldsymbol{\gamma} \quad (6.68)$$

With:

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{M_1} \\ \boldsymbol{M_2} \\ \boldsymbol{M_3} \end{bmatrix}; \quad \boldsymbol{\gamma} = \begin{bmatrix} \boldsymbol{N_1} \\ \boldsymbol{N_2} \\ \boldsymbol{N_3} \end{bmatrix};$$

$$\boldsymbol{M_1} = \begin{bmatrix} -I_{m \cdot N_c} \\ I_{m \cdot N_c} \end{bmatrix}; \quad \boldsymbol{M_2} = \begin{bmatrix} -\boldsymbol{C_2} \\ \boldsymbol{C_2} \end{bmatrix}; \quad \boldsymbol{M_3} = \begin{bmatrix} -\boldsymbol{\Phi_u} \\ \boldsymbol{\Phi_u} \end{bmatrix};$$

$$\boldsymbol{N_1} = \begin{bmatrix} -\Delta\boldsymbol{U}^{min} \\ \Delta\boldsymbol{U}^{max} \end{bmatrix}; \quad \boldsymbol{N_2} = \begin{bmatrix} -\boldsymbol{U}^{min} + \boldsymbol{C_1} \cdot \boldsymbol{u_m}^{k-1} \\ \boldsymbol{U}^{max} - \boldsymbol{C_1} \cdot \boldsymbol{u_m}^{k-1} \end{bmatrix}; \quad \boldsymbol{N_3} = \begin{bmatrix} -\boldsymbol{Y}^{min} + \boldsymbol{F} \cdot \boldsymbol{x}^k + \boldsymbol{\Phi_v} \cdot \Delta\boldsymbol{V} \\ \boldsymbol{Y}^{max} - \boldsymbol{F} \cdot \boldsymbol{x}^k - \boldsymbol{\Phi_v} \cdot \Delta\boldsymbol{V} \end{bmatrix}$$

$\boldsymbol{M_2}$ and $\boldsymbol{N_2}$ matrices are derived from Equation 6.67, while $\boldsymbol{M_3}$ and $\boldsymbol{N_3}$ from Equation 6.62. Finally, $I_{m \cdot N_c}$ indicates the identity matrix of dimension $m \cdot N_c \times m \cdot N_c$, where m is the number of controlled inputs and N_c is the number of samples related to control horizons.

In this case, to reduce the required calculations, constraints can only be imposed for certain quantities and for a time that can be less than the control or prediction horizons. Main decisions made about constrains of our MPC are:

- The constrained outputs $\boldsymbol{Y}$ are rotor speed and power. In both cases, a maximum and minimum differences to the target values are imposed.

- The rates of change of the controlled inputs $\Delta\boldsymbol{U}$ are all bound with absolute values.

- Finally, the constrained inputs $\boldsymbol{U}$ are the minimum and maximum difference of generator torque and blade pitch angle from their rated values.

It is underlined that although only some of the input constraints may be active in the optimisation procedure, all of them are then saturated to the maximum and minimum values allowed by the actuators.

6.2.4.4 Optimization Algorithm

Once all the ingredients for constrained optimization have been obtained, i.e. the cost function and the constraints, it is possible to combine equations 6.62, 6.63, 6.67 and 6.68 and arrive at a canonical formulation for solving problems of this type such as the one presented in section 5.2.5.2. It consists of a quadratic cost function with symmetric and positive definite Hessian and linear constraints:

$$\begin{cases} J(\boldsymbol{z}) = \frac{1}{2}\,\boldsymbol{z}^T \cdot \boldsymbol{E} \cdot \boldsymbol{z} + \boldsymbol{z}^T \cdot \boldsymbol{f} \\ \boldsymbol{g}(\boldsymbol{z}) = \boldsymbol{M} \cdot \boldsymbol{z} - \boldsymbol{\gamma} \leq 0 \end{cases} \tag{6.69}$$

Where: $\boldsymbol{z} = \Delta\boldsymbol{U}$

$$\boldsymbol{E} = \left(\boldsymbol{\Phi_u}^T \cdot \boldsymbol{W_Y} \cdot \boldsymbol{\Phi_u} + \boldsymbol{W}_{\Delta U} + \mathbf{C_2}^T \cdot \boldsymbol{W_U} \cdot \mathbf{C_2}\right)$$

$$\boldsymbol{f} = -\boldsymbol{\Phi_u}^T \cdot \boldsymbol{W_Y} \cdot \left(\boldsymbol{T_Y} - \boldsymbol{F} \cdot \boldsymbol{x}^k - \boldsymbol{\Phi_v} \cdot \Delta\boldsymbol{V}\right) - \mathbf{C_2}^T \cdot \boldsymbol{W_U} \cdot \left(\boldsymbol{T_U} - \mathbf{C_1} \cdot \mathbf{u_m}^{k-1}\right)$$

The last thing to be defined is the resolution method, which may be one presented in section 5 or a different one. In this case, the Hildreth procedure presented there is adopted.

7 Results

In this chapter, we will analyse the results from some simulations of the system controlled by the MPC controller defined in the previous chapter. The model adopted for the system is the non-linear model, called NL, obtained from the system equations in Chapter 3 and defined in the state space form in Section 3.3. The external inputs used for the simulation are those described in Section 3.2, i.e., the wind speed at N points on a spatial grid covering the rotor area and the wave excitation forces and torques.

In a first step, we will look at some examples of results obtained from the simulations, to understand how the controller works and its potentialities related to the unconventional working approach. Then, in a second step, we will compare the performance of the MPC controller with the two conventional controllers presented in Chapter 4, that is, the Baseline and ROSCO controllers. In order to compare their performance, a number of indicators will be defined which go beyond the kind of control adopted, trying to make a reasonable comparison between the results obtained.

7.1 Examples of MPC Results

As just mentioned, below there are some examples in which the system controlled by the MPC controller is simulated, in order to emphasise some important aspects.

Example n.1

This example shows the results of two simulations where the average speed (over time) at the hub height is 7 m/s, and the wave characteristics are: H_s=4 m, T_p=6 s (see section 3.2 for details regarding these quantities). The two tests differ in the settings adopted in the controller, in particular, regarding in the weights for the rotor speed and power targets (all other settings are the same in the two cases).

The choices made are:

- **Case 1**: Rotor speed weight much greater than power one;
- **Case 2**: Comparable rotor speed and power weights.

For this first example only, graphs of external inputs (wind speed and wave excitation forces) will be shown. It should be noted that although the wind input concerns the speed at N points on a grid covering the rotor area, the average speed across this area is shown for simplicity. The results regarding the external input estimation module are also shown. They refer to case 1, we do not show those related to case 2 because they should correspond, although, due to the different evolution of the system, they differ, even if only slightly. Furthermore, the GoF (Goodness-of-Fit) is showed, to quantify how correct the estimate is (see definition 6.15).

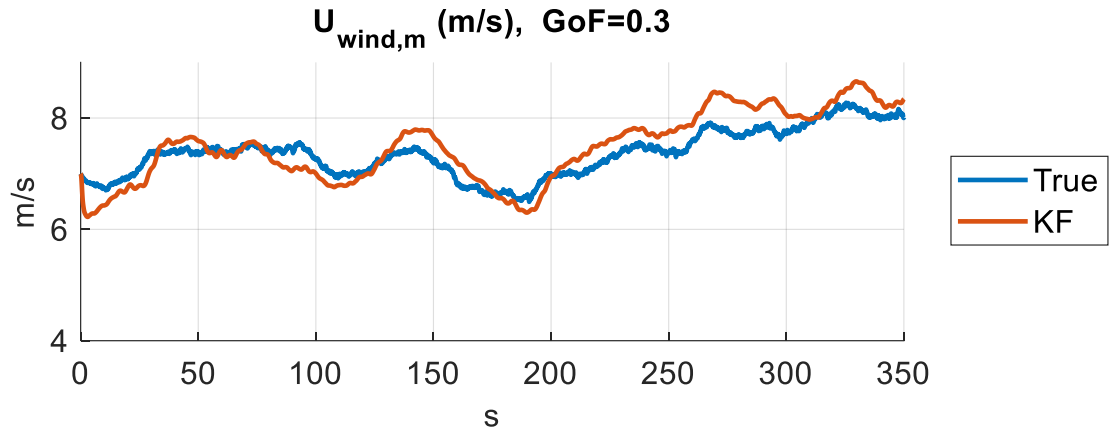

Figure 7.1: Mean wind speed (across rotor area) and its estimation through dedicated Kalman filter.

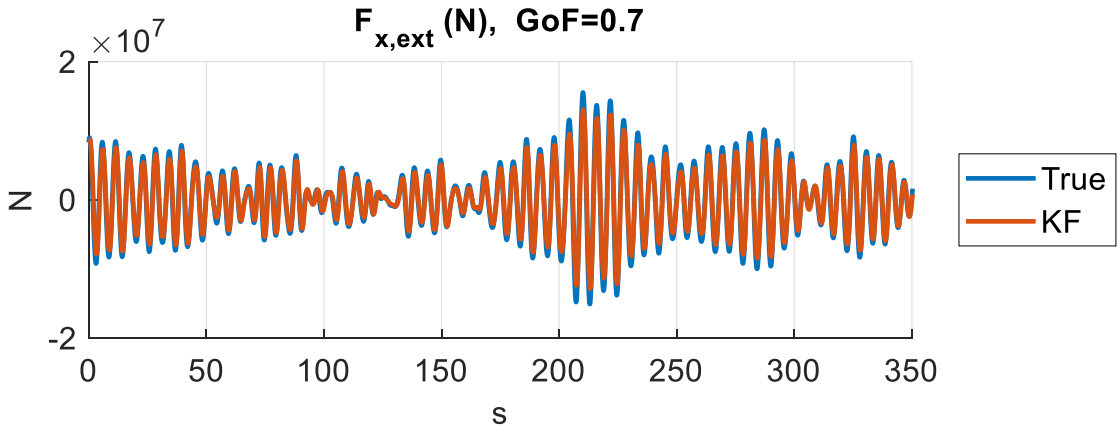

Figure 7.2: Wave excitation force along x (surge) direction and its estimation through dedicated Kalman filter.

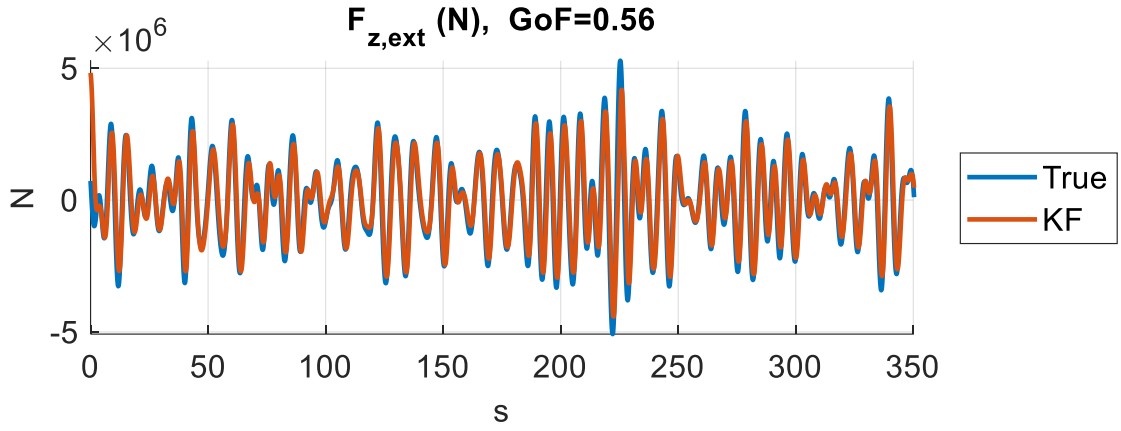

Figure 7.3: Wave excitation force along z (heave) direction and its estimation through dedicated Kalman filter.

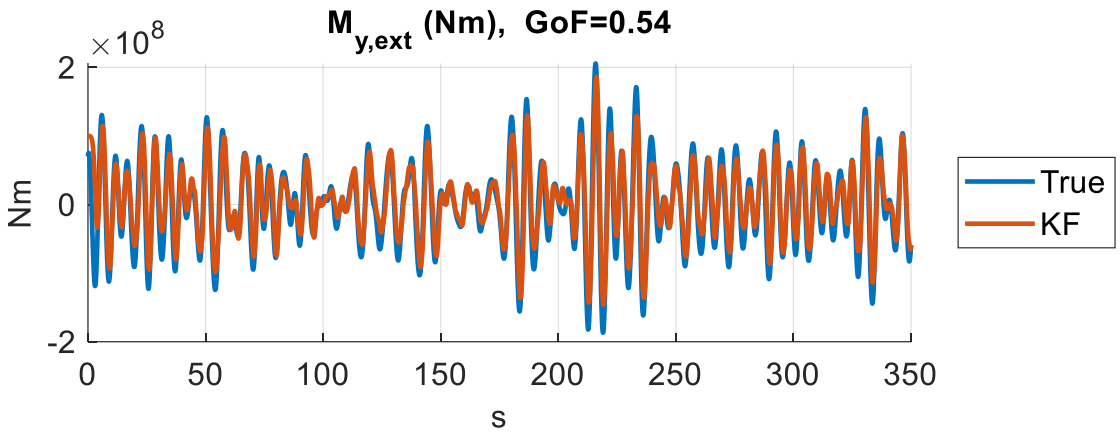

Figure 7.4: Wave excitation torque around y (pitch) axis and its estimation through dedicated Kalman filter.

It is important to remember that the main differences between actual and estimated wind speeds are due to the former is an average of the N speeds of the grid points, while the latter is the speed (the only one considered in the filter model, see section 6.2.1.1) that leads to the best match between the measurements taken and those predicted by the model. If the actual velocities were all the same (for each grid point), the estimate would be closer to the true value.

The two images below show the time trends of rotor speed, generator torque and power in the two examined cases, the relative targets (subscript T in legend) are also shown, which differ only because of the slightly different wind speed estimation.

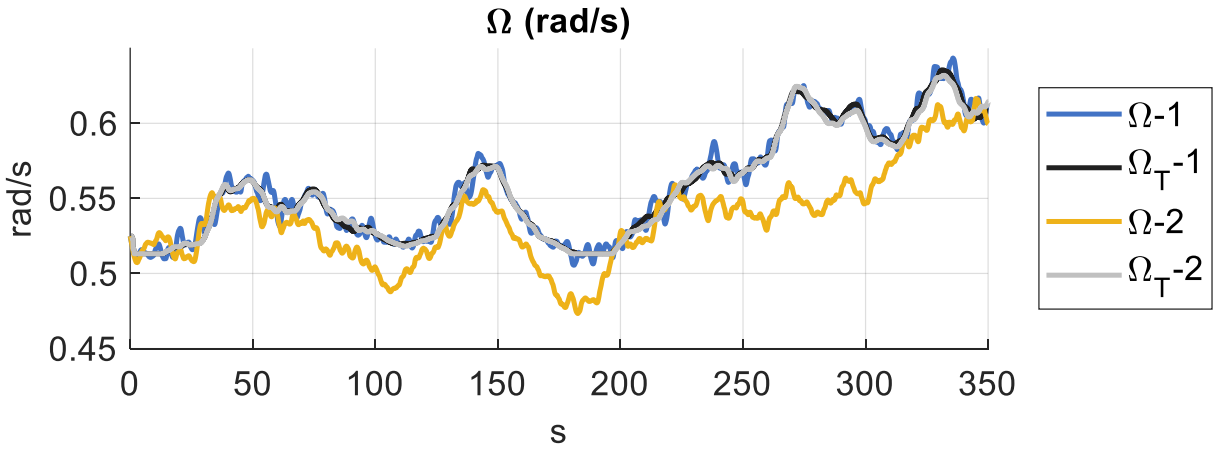

Figure 7.5: Rotor angular speed of the two examined cases and their respective targets.

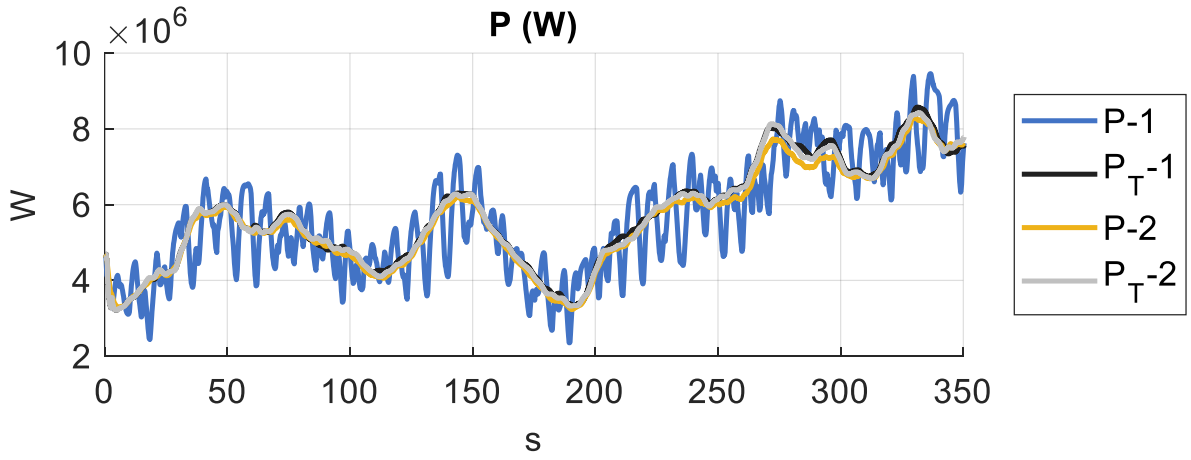

Figure 7.6: Extracted power of the two examined cases and their respective targets.

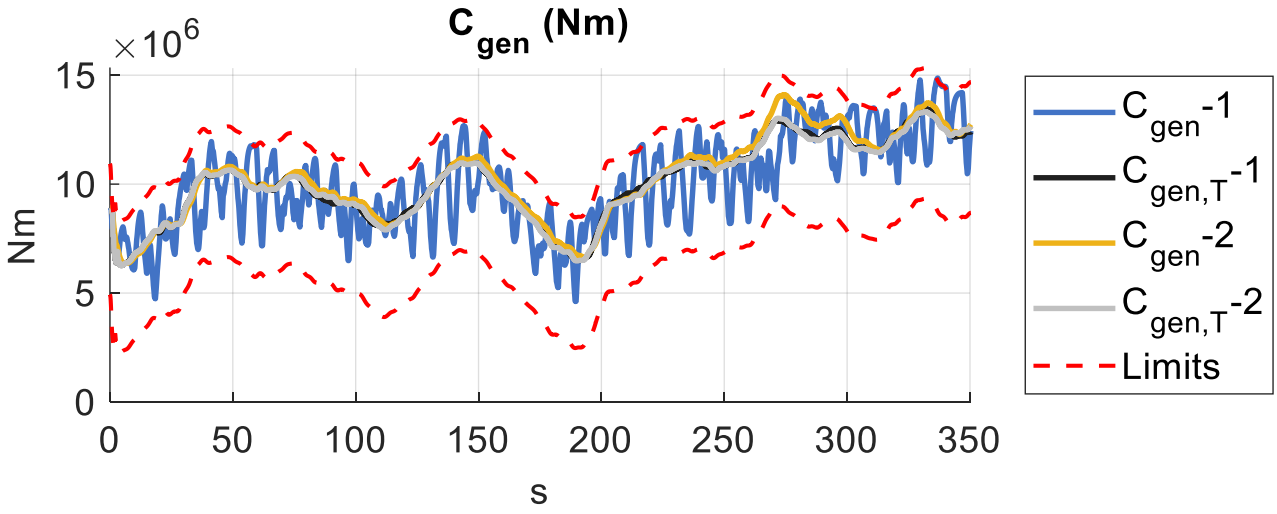

Figure 7.7: Torque exerted by the generator in the two examined cases and their respective reference values.

From the graphs shown, it can be seen that the choice of weights associated with the different targets is very important in order to achieve the objectives set. Although having the correct speed is sought to maximise power, because it leads to the optimum tip speed ratio, raising the associated weight too much leads to torque fluctuations (necessary for tracking the speed), resulting in high power oscillations. Conversely, raising its associated weight too much could lead to speed values too far from the nominal ones, resulting in operating conditions outside the linearisation grid with consequent problems for the controller, because invalid models would be used, generating incorrect results. A compromise situation must therefore be found, as happens, for example, in the second case, where the speed error is acceptable, and the power is sufficiently close to the target.

Example n.2

This example shows the results of two simulations performed with wind speed (average, at hub height) of 19 m/s and the following wave characteristics: H_s=4 m, T_p= 6 s. As before, two different cases are compared, with different approaches to the weights adopted in the cost function of the controller, to realize how important it is to find the right compromise between the various objectives pursued. The cases adopted have the following peculiarities (the other settings are the same in both cases):

- **Case 1**: Weight associated with the structure pitch speed very high compared to that associated with other targets;
- **Case 2**: Weight associated with power very high compared to the weights associated with the other targets.

Below there are the power and structure's pitch angular speed trends resulting from the simulations.

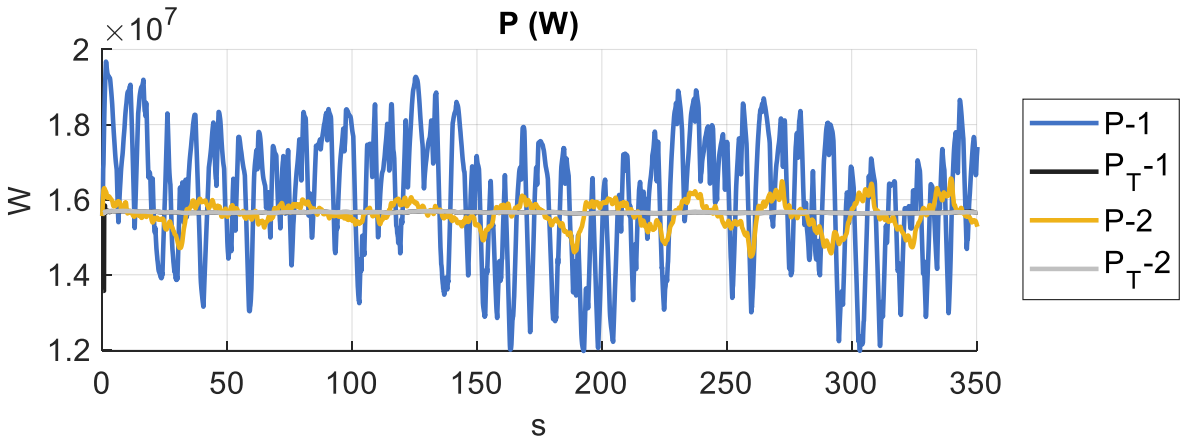

Figure 7.8: Extracted power of the two examined cases and their respective targets.

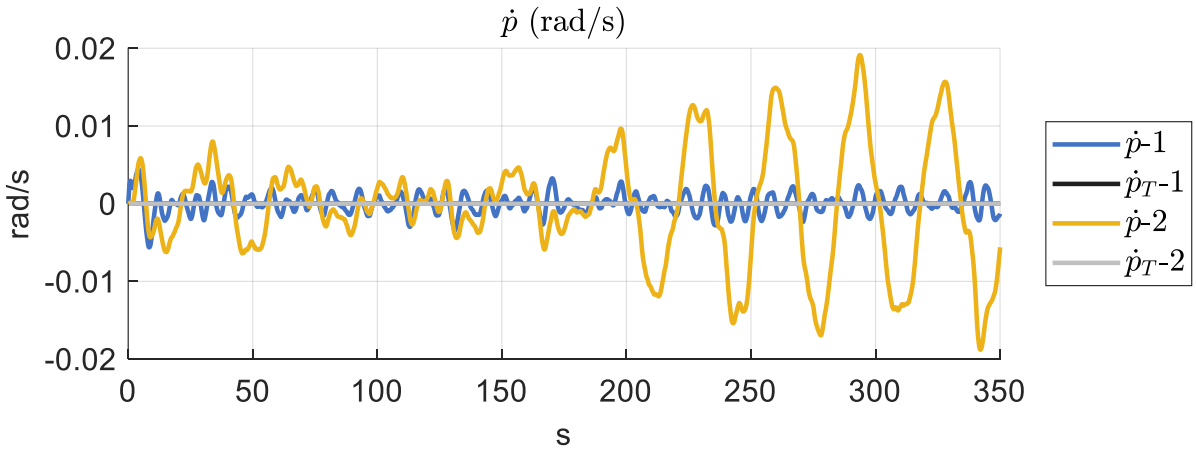

Figure 7.9: Structure pitch angular speed of the two examined cases and their respective targets

With this example, it is possible to appreciate the potential of the MPC control logic, indeed, from the graphs it can be seen that the desired targets can be reached with good precision. By giving more weight to a specific target, it is possible to make the system evolve towards that goal. Thus, by means of this type of control, it is possible to obtain the desired characteristics in a simple manner (only by varying the value of the weights of the cost function), unlike traditional controllers, where it is only possible, by varying the gains, to indirectly pursue the set targets, making it more difficult to reach the final objective, when this is feasible.

To further emphasise the MPC control's ability to achieve its goals, below there is a graph of case 1, where the goal with the greatest weight is the structure's pitch speed. This graph shows the actual trend of this speed (blue) compared with the relative target (red) and the trend predicted by the model within the prediction horizon PH (green). In this case the prediction horizon is 3.2 s, while the time step with which the control actions vary is 0.4 s, i.e., each horizon consists of 8 steps. It can be seen that within each prediction horizon, with the control actions determined by numerical optimisation, the target of zero speed is reached, and the actual speed follows quite well what the model predicts, despite a certain difference mainly due to the fact that along that time the actual external inputs deviate from those predicted.

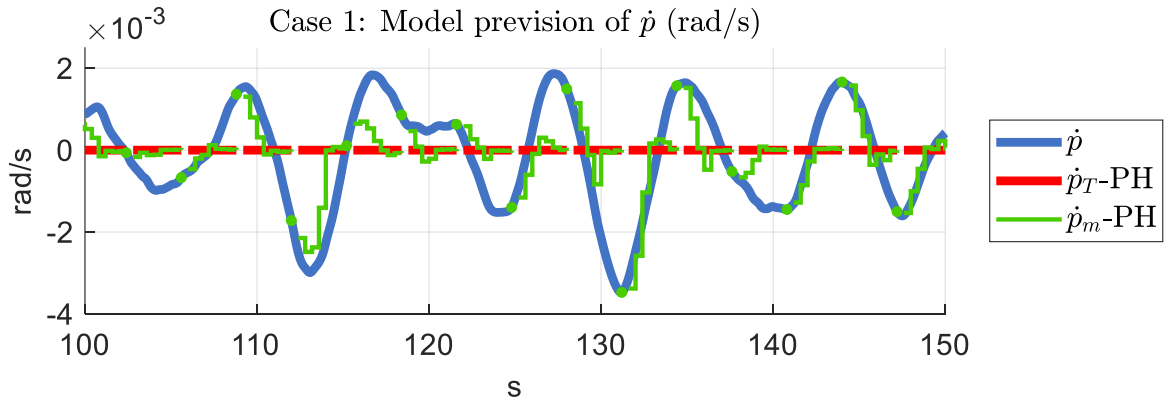

Figure 7.10: Actual angular pitch speed of the structure (blue) compared with the related target (red) and the trend predicted by the model within the PH prediction horizon (green).

Through these two examples, therefore, we have seen the good potential of this type of control, through which it is easy to pursue the objectives set, we have also seen, however, that if there are many objectives (among which are also the objectives concerning control actions, for which it is possible to provide limits in order to moderate them for reasons of wear and energy expenditure) it is necessary to find the right weight for each in order to obtain an overall satisfactory performance.

As an example, in these last two analyzed cases, the target on which most of the focus was placed is tracked very well, but to the detriment of the other targets, so it is clear that the settings adopted here are not the optimal ones in order to perform well overall. The main challenge, therefore, is to find the best weights for the various targets, and since there are numerous variables (consider that different weight matrices can be defined for each parameter combination, see section 6.2.4.2), finding the right compromise may be difficult.

7.2 Comparison Between MPC, Baseline and ROSCO

In this section, the performances of the MPC controller are compared with that of the Baseline and ROSCO controllers presented in Chapter 4. In summary, these are two gain scheduled PID controllers for variable speed and collective blade pitch angle turbines. In both cases, control strategy consists of power maximization at low wind speed (below rated value) and rotor speed and power tracking at their rated values when the wind speed is higher than rated one.

The operating principle is similar, however, the ROSCO controller has additional modules with which performance can be improved. For example, it has a module consisting of a Kalman filter for estimating wind speed. This is used to calculate the reference rotor speed in order to optimise the tip speed ratio and thus the extracted power. In addition, it has a module or floating turbines, through which an attempt is made to reduce the oscillations of the structure and increase the power extracted.

To compare controllers, the same system model and external inputs are used. Specifically, the external inputs are the wind speed (at N points on a spatial grid covering the rotor area) and the wave excitation forces defined in section 3.2. The system's model used is the non-linear model called NL and defined in section 3.3. In addition, the limits on the torque that can be delivered by the generator and the blade angle, as well as those on their rates of change, are also considered to be the same in all cases. Moreover, the data from which the Baseline and ROSCO controllers are built (structure geometry and inertias, power coefficient, etc.) are the same, and the stationary nominal values of the quantity of interest (rotor speed, power, blade pitch, etc.) at different wind speeds used for ROSCO implementation are the values adopted for the MPC controller targets (Figure 4.10 and 6.20). These values are also used to define the torque-speed characteristic used in the Baseline. Finally, with regard to the estimation of wind speed in the ROSCO controller, it was decided to use the same Kalman filter as in the MPC controller, so that the performance of the latter would not affect the comparison between the two different control algorithms.

Although the general objectives of all controllers are the same, it may not be easy to compare their performance, as the working principles are different. For example, the MPC controller explicitly tries to track the imposed targets, while the other two pursue these objectives due to the chosen gain values and, for the ROSCO, due to the various add-on modules. Therefore, to compare performance, it was decided to define a set of indices that provide information beyond these differences, which is of a general nature and does not favour or penalise a specific control strategy.

The performance indices chosen for the controller comparison are:

- $\boldsymbol{E_{\Omega,\ rms}}$: root-mean square value of rotor speed relative error $E_\Omega = \frac{\Omega_T - \Omega}{\Omega_T}$, where Ω is the actual rotor speed, while Ω_T is its target value. The latter is given by steady state values of rotor speed with respect to wind speed, the same for all controllers and computed with actual wind speed. The root-mean square value of rotor speed relative error E_Ω is defined as: $E_{\Omega,\ rms} = \sqrt{\frac{1}{N} \cdot \sum_{k=1}^{N} \left(\frac{\Omega_T - \Omega}{\Omega_T}\right)^2}$, where N is the number of samples obtained during simulations.

- $\boldsymbol{E_{P,\ rms}}$: root mean square value of power relative error $E_P = \frac{P_T - P}{P_T}$, where P is the actual rotor speed, while P_T is its target value. The latter is given by steady state values of rotor speed with respect to wind speed, the same for all controllers and computed with actual wind speed. Power root-mean square value is defined in the same way as rotor speed one.

- $\boldsymbol{\dot{p}_{rms}}$: root mean square value of structure pitch angular speed. It provides information about the stability of the structure; a low value means, on average, low rotational speeds and smaller amplitudes of oscillations, considering the frequency dictated by the forcing action (mainly wind). $\dot{p}_{rms}$ is defined as follows: $\dot{p}_{rms} = \sqrt{\frac{1}{N} \cdot \sum_{k=1}^{N} \dot{p}^2}$.

- $\boldsymbol{P_{mean,n}}$: normalized mean extracted power, it is the ratio between mean power and nominal one: $P_{mean,n} = {P_{mean}}/{P_{rated}}$, it should be as high as possible when wind speed is lower than its rated value, whereas it should be as close as possible to the nominal value when the wind speed is above the rated value.

- $\boldsymbol{P_{max,n}}$: normalized maximum extracted power, it is the ratio between maximum power and limit one: $P_{max,n} = {P_{max}}/{P_{limit}}$. The first is the maximum value reached during simulation, while the second one is the maximum allowed value, which is chosen taking into account that high power peaks could overload and damage the energy conversion system. This index will be accompanied by the percentage of time for which the power exceeds the limit: $\boldsymbol{P_{out,\%}}$.

- $\boldsymbol{\dot{\theta}_{bl,\ rms}}$: root mean square value of blade pitch angular speed. This indicator takes into consideration the use of the servomotors used for the blade angle variation. With the same other goals achieved, it would be better to have low values, because they mean less use of the actuators, therefore less energy spent and less wear. Furthermore, a high utilization of this control often leads to disadvantageous fatigue cycles at the root of the blades. $\dot{\theta}_{bl,\ rms}$ is defined as $\dot{\theta}_{bl,\ rms} = \sqrt{\frac{1}{N} \cdot \sum_{k=1}^{N} \dot{\theta}_{bl}^{\ 2}}$, where $\dot{\theta}_{bl}$ is the collective blade pitch angular speed.

- $S_{f,25y}$: safety coefficient to have a useful life of 25 years. It is defined as the ratio between the stress leading to a useful life of 25 years and the effective stress. This calculation refers to the root of the blades, i.e. the tension mentioned is calculated at the connection between hub and blade. In detail, for each simulation carried out, the forces and torques at the blade root are computed, and the time history of the equivalent stress at certain points of the section is calculated (considering the section a circular crown, some angularly equally spaced points of the periphery are taken). For each point, we then look for the safety coefficient which, multiplied by the stress time series, leads to a service life of 25 years. This calculation is performed by means of an optimisation function and the useful life is obtained from the stress by means of the rainflow method for counting load cycles and by means of the constant life diagram (CLD) of the chosen material. For each cycle, the damage related to it is calculated using this diagram, then the various damages are added together, and the useful life is obtained considering the time taken to accumulate this total damage. Finally, the lowest safety coefficient is chosen from the various points considered for the calculation of the indicator.

  The constant life diagram (CLD) is a representation of the results of fatigue analysis on a certain material. Normally, several fatigue tests are performed with a certain value of R (ratio of minimum to maximum stress, $R = \sigma_{min}/\sigma_{max}$). S-N relationships are then obtained for each R, where S is normally the alternate stress and N is the number of cycles to failure. Finally, the CLD diagram is constructed by defining several curves for different values of N, each obtained by joining the points for each value of R considered [33]. To better understand the nature of this diagram, an example relating to the material chosen for the calculation of the indicator is shown in figure 7.11. The material under consideration is a typical fiberglass laminate that is called DD-16 in the DOE/MSU Database [34], and it is a material usually used for wind turbine blades. This laminate has a [90/0/±45/0] configuration with a fiber volume fraction of 0.36 [35], [36].

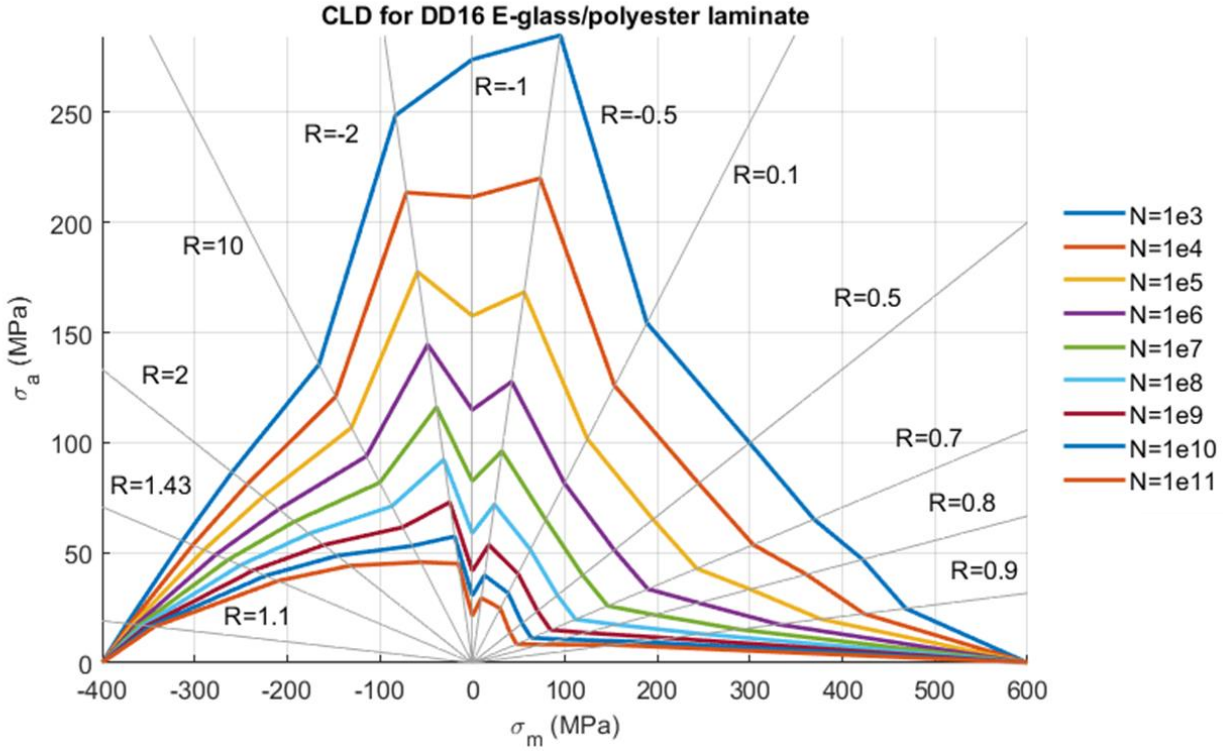

Figure 7.11: Constant life diagram (CLD) of DD16 E-glass/polyester laminate for some values of N and R.

It is important to emphasise that we chose to perform the calculation of the safety coefficient at the root of the blades essentially for simplicity of calculation, given the simple section and the simplicity of calculating forces and torques at this point, since the results from BEM theory for calculating aerodynamic forces and torques refers to this section. Actually, the most stressed point of the blade is probably at a position with a larger radius. Furthermore, the section at the root of the blades is different from that adopted in the calculations, as the actual geometry is more complex, given the need for connection to the hub and the presence of the actuator system for varying the blade pitch angle. For this reason, the calculated safety coefficient is higher than the real one, but it remains a good indicator of the system's useful life, especially considering that the main need is to make a comparison between different controllers, where the same above assumptions have been adopted.

In the next sections the results related to the comparison between the controllers will be reported, specifically the values of the indicators defined above and some examples of results of the simulations comparing the different control algorithms, to comment on and underline some aspects that can be deduced from the indicators. Unlike in the previous section, where the weights and limits of the MPC controller were chosen to highlight some important aspects that were wanted to be discussed, in this section we will use the values of these quantities that allow to have the best possible overall performance, considering all important aspects and looking for a good compromise between them. It should be noted that these values were chosen after many tests, but were not optimised, for example through optimization algorithms, mainly because of the difficulty of this operation, given the many variables and the high computational time required for simulations. It is therefore reasonable to think that with an adequate optimization the performance can improve further.

The results obtained will be divided into three sections, relating to tests where the average wind speed is lower, around, or greater than the nominal speed, since the main objectives differ depending on whether we are above or below that speed. The central category, moreover, is considered apart because it is important to understand how the controllers work when the external conditions are those for which the turbine has been designed.

Below Rated Conditions

In this area of operation, the targets of greatest interest are the errors related to rotor speed and power, as well as the average power produced, since under these conditions the main goal is to maximize the latter. The pitch angular speed of the structure is also important, although for low wind speeds it remains low compared to conditions close to nominal, where the thrust force is maximum (see figure 4.10). The same applies to the safety factor at the blades' root. With regard, on the other hand, to the maximum power achieved compared to the limit one and the root mean square value of blades pitch speed, these indices have less relevance in this zone, since the powers involved are low compared to the other zones and since the control strategy in these conditions does not involve the use of the blade angle as a control action.

Below are some graphs comparing the most relevant indices for these conditions for the three controllers examined. They are divided according to the simulations carried out, each of them with a different average hub height wind speed ($U_{wind,hub,mean}$). Notice that the input wind speeds at the various grid points covering the rotor area are generated from this average speed using the TurbSim software (see section 3.2.1).

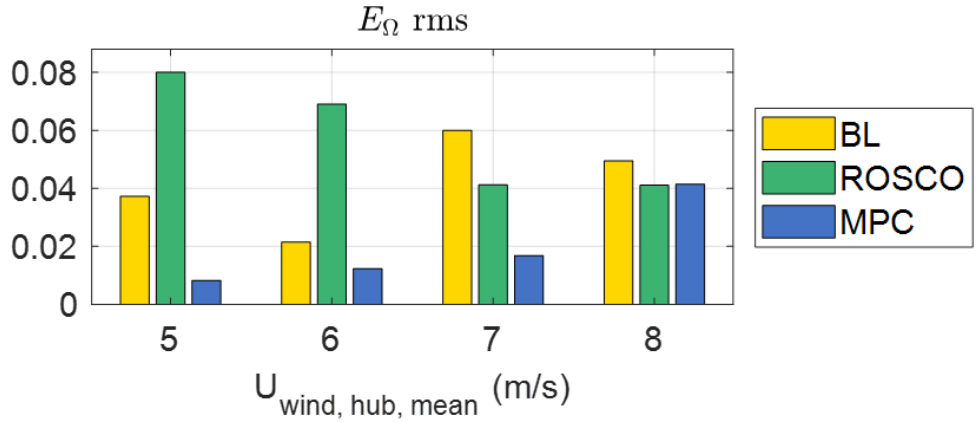

Figure 7.12: Comparison between Baseline, ROSCO and MPC, below rated conditions, $E_{\Omega,\ rms}$.

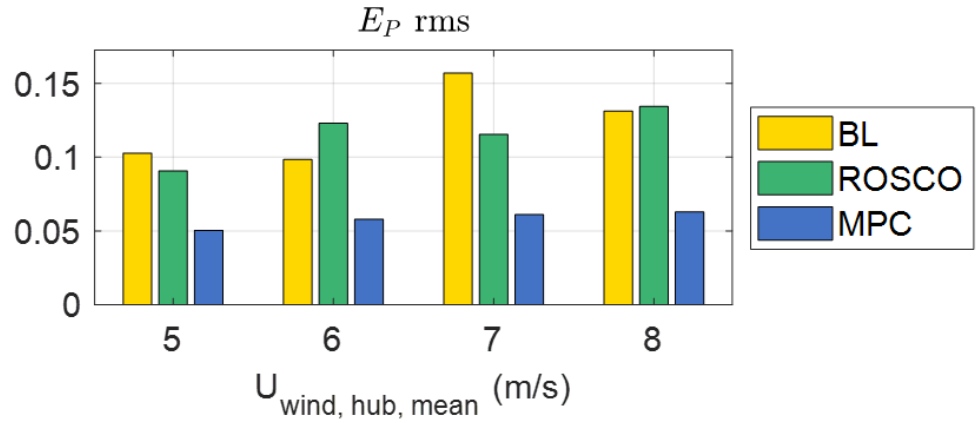

Figure 7.13: Comparison between Baseline, ROSCO and MPC, below rated conditions, $E_{P,\ rms}$.

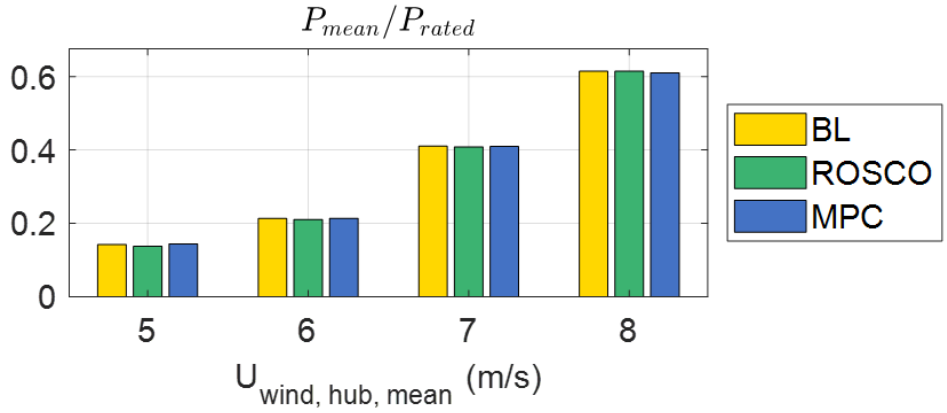

Figure 7.14: Comparison between Baseline, ROSCO and MPC, below rated conditions, $P_{mean,n} = {}^{P_{mean}}/_{P_{rated}}$.

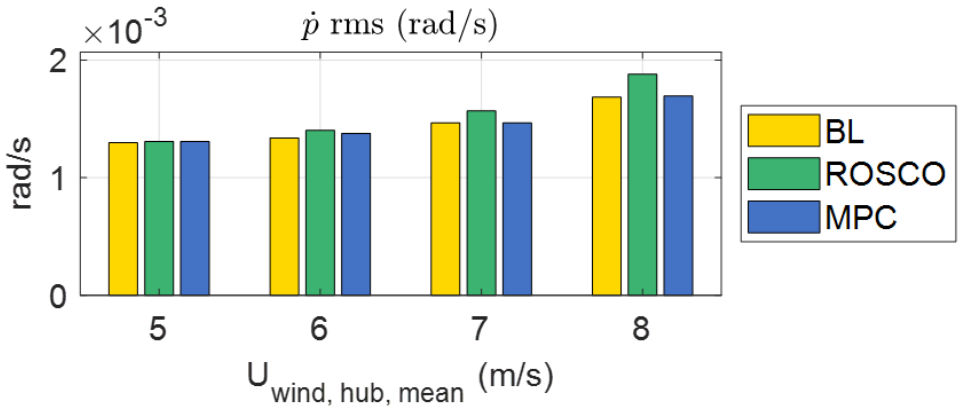

Figure 7.15: Comparison between Baseline, ROSCO and MPC, below rated conditions, $\dot{p}_{rms}$.

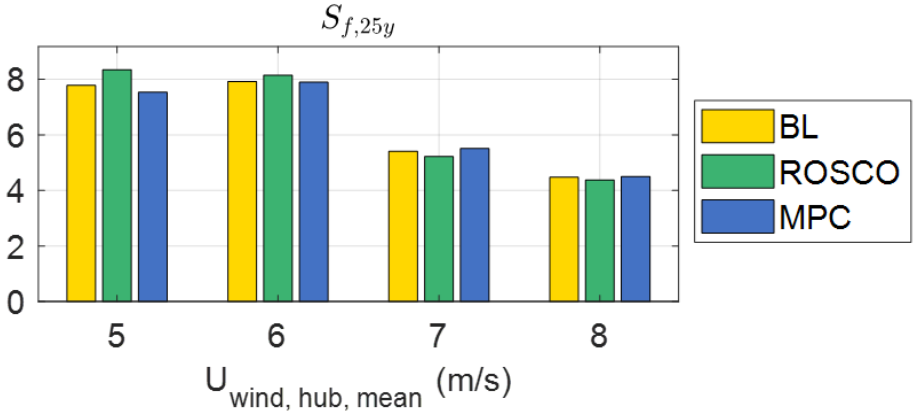

Figure 7.16: Comparison between Baseline, ROSCO and MPC, below rated conditions, $S_{f,25y}$.

From the graphs shown, it can be seen that the MPC controller achieves lower angular rotor speed and power errors than the other two controllers. In particular, the power output trend of the Baseline controller shows greater fluctuations around the local average value, while the trend of the ROSCO controller shows greater differences between the target and the actual value. To support this consideration, a graph depicting the power trend of the three different cases is shown, for the simulation with an average wind speed at hub height of 6 m/s.

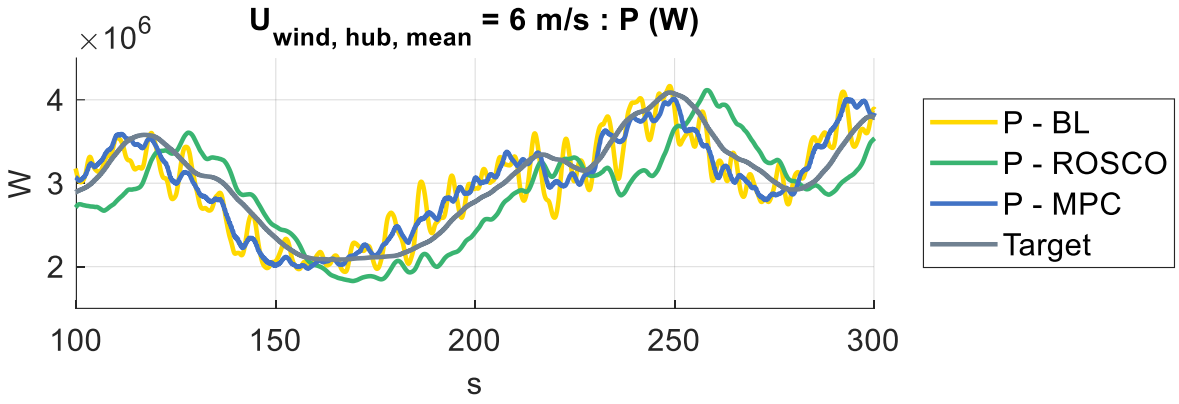

Figure 7.17: Extracted power trends for the three different controllers when simulated with an average speed at hub height of 6 m/s.

Although lower rotor speed and power errors can be achieved with the MPC controller than with the other controllers, it can be seen from Figure 7.14 that the average power extraction is very similar in the three cases, which suggests that better results are not actually achieved with the MPC. This could have been expected, since the two conventional controllers considered are optimized to extract the maximum possible power when under the conditions analyzed here. Different considerations, however, can be made regarding structure oscillations, an indication for which they are not directly optimized. In this case, there are better results in the case of the MPC controller, especially when compared to the ROSCO. Considering, however, that under these operating conditions in all cases the pitch speed of the structure is reduced, due to the low wind speed, the fact that the MPC is slightly improved does not lead to consider it much better than the others. Finally, regarding the safety factor (Figure 7.16), few considerations can be drawn since the hierarchy between the three controllers does not remain constant. However, where the coefficient presents lower values (with average wind speed at hub height of 7 and 8 m/s) MPC performances are slightly better.

Near Rated Conditions

This section will analyse the performance of the three controllers when the wind speed is close to the nominal wind speed, which in this case is approximately 10 m/s. In this case, the main objective is to maximise power, but other objectives also become important, such as limiting the pitch rate of the structure (since this zone is the one with the maximum thrust force) and limiting the use of the blade pitch angle (since in this zone this control action begins to be used because it is possible to encounter wind speeds higher than the nominal one). Furthermore, in this zone it starts to become important not to exceed the maximum power, which in this case is set at 110% of the nominal power, for the reasons explained above.

Below there are some graphs comparing the performance indices for these conditions for the three controllers examined. They are divided according to the simulations carried out, each of them with a different average hub height wind speed ($U_{wind,hub,mean}$). Notice that the input wind speeds at the various grid points covering the rotor area are generated from this average speed using the TurbSim software (see section 3.2.1).

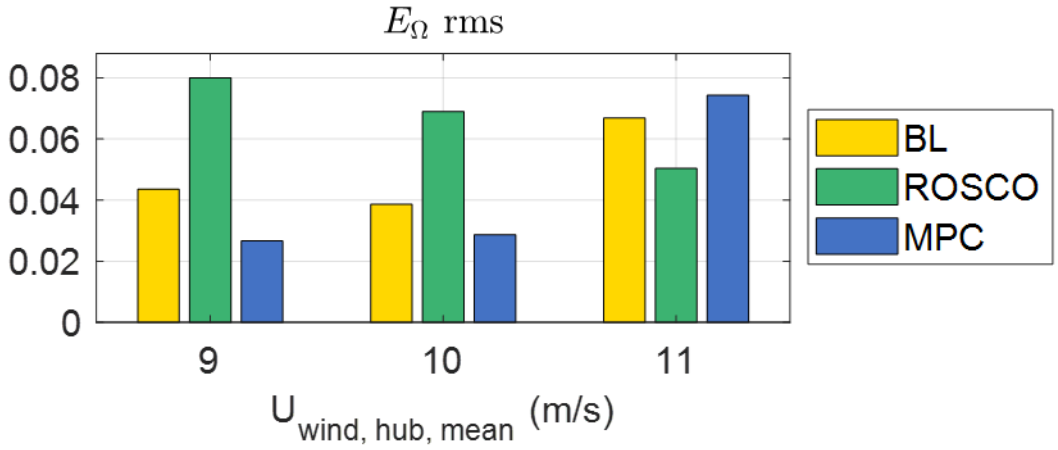

Figure 7.18: Comparison between Baseline, ROSCO and MPC, near rated conditions, $E_{\Omega,\ rms}$.

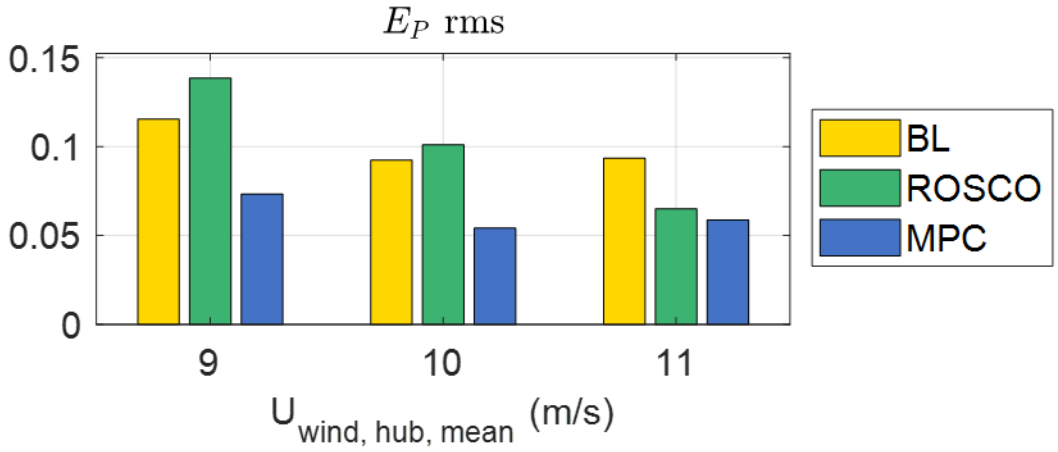

Figure 7.19: Comparison between Baseline, ROSCO and MPC, near rated conditions, $E_{P,\ rms}$.

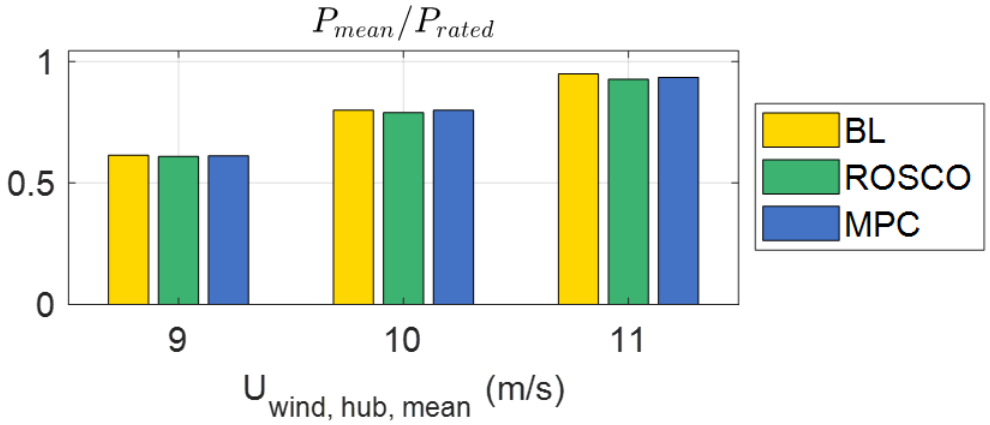

Figure 7.20: Comparison between Baseline, ROSCO and MPC, near rated conditions, $P_{mean,n} = {P_{mean}}/{P_{rated}}$.

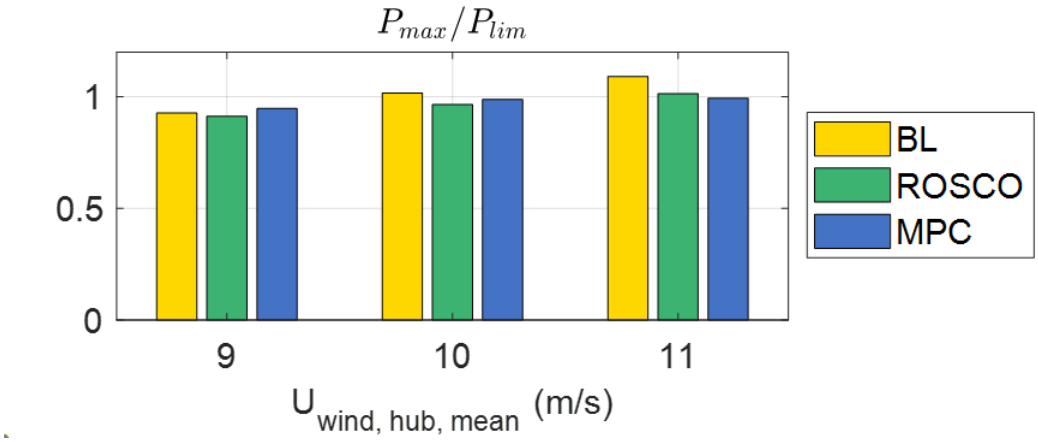

Figure 7.21: Comparison between Baseline, ROSCO and MPC, near rated conditions, $P_{max,n} = {P_{max}}/{P_{lim}}$.

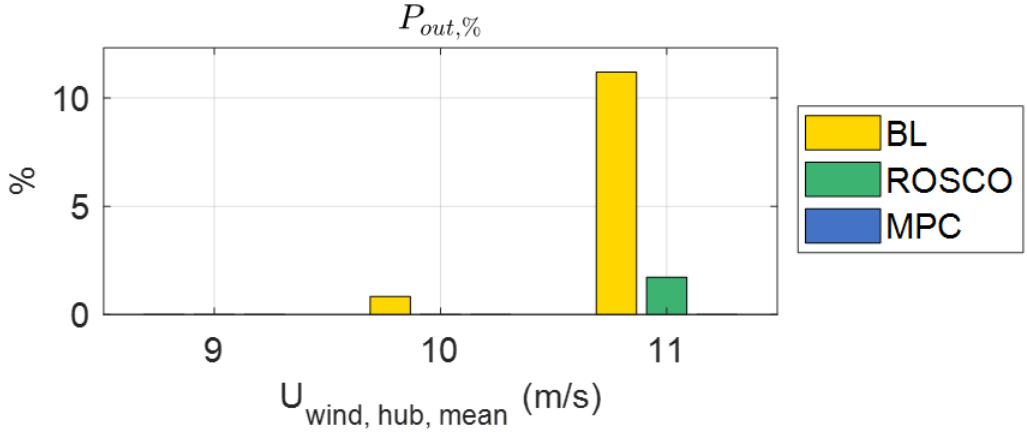

Figure 7.22: Comparison between Baseline, ROSCO and MPC, near rated conditions, percentage of time for which the power exceeds the limit: $P_{out,\%}$.

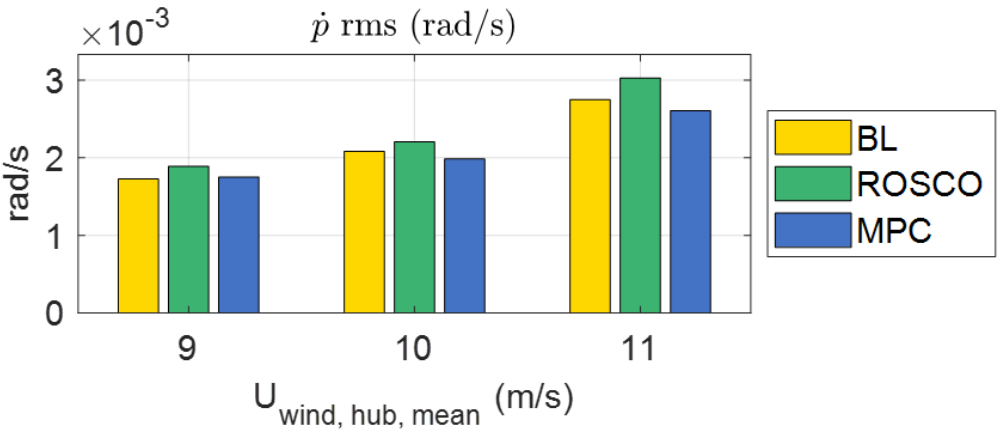

Figure 7.23: Comparison between Baseline, ROSCO and MPC, near rated conditions, $\dot{p}_{rms}$.

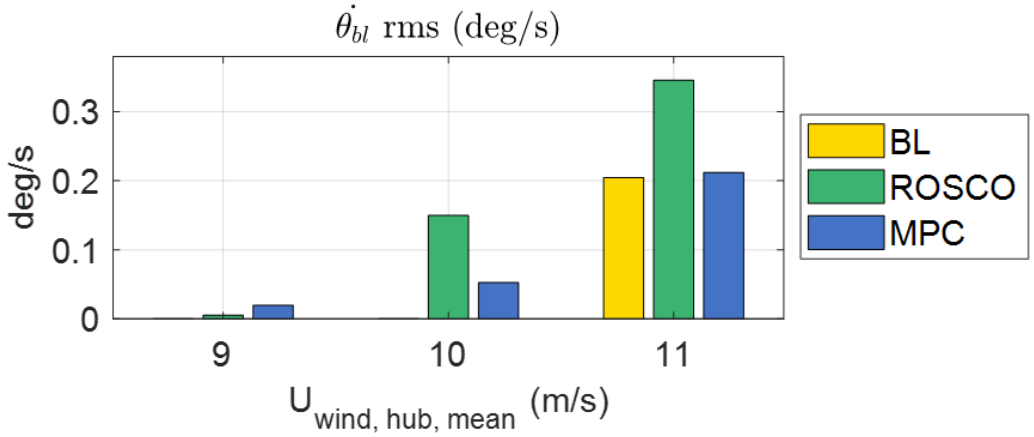

Figure 7.24: Comparison between Baseline, ROSCO and MPC, near rated conditions,$\dot{\theta}_{bl,\ rms}$.

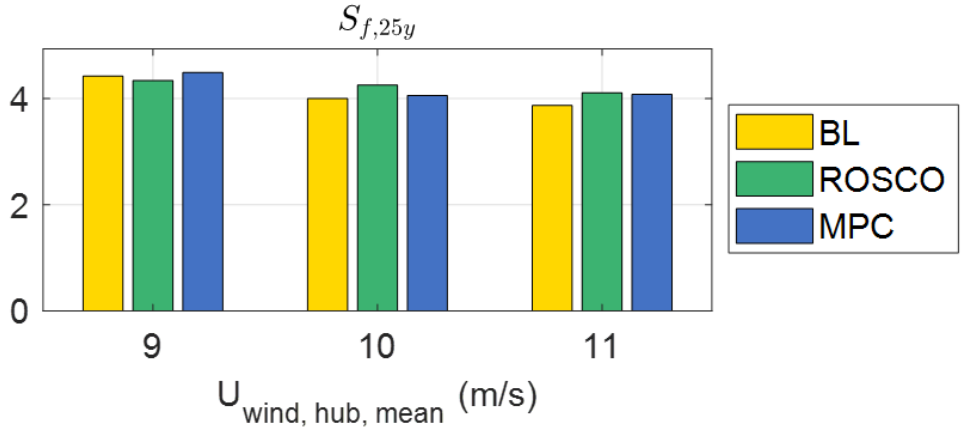

Figure 7.25: Comparison between Baseline, ROSCO and MPC, near rated conditions, $S_{f,\ 25y}$.

The graphs show that the MPC controller overall performs slightly better than the other two controllers. In favour of this thesis, one can note, for example, the generally lower values for power and speed errors. Regarding the average power output, however, the MPC never reaches the highest values. It should be noted, however, that the other two controllers present situations in which the power exceeds the limit, which is why average power tends to be higher than that produced with the MPC (it should be noted, however, that the differences are very slight). To better understand this, Figure 7.26 shows the time trend of the power in the three cases when the average wind speed at the hub is 11 m/s, as an example. Here, it can be seen that in the two conventional controllers, the power occasionally exceeds the limit. This fact is understandable, since there is no constraint on the output quantities in these controllers, as is the MPC case (this is, actually, a strength of this type of control and one of the main reasons why it was decided to try to implement an MPC controller for floating wind turbines).

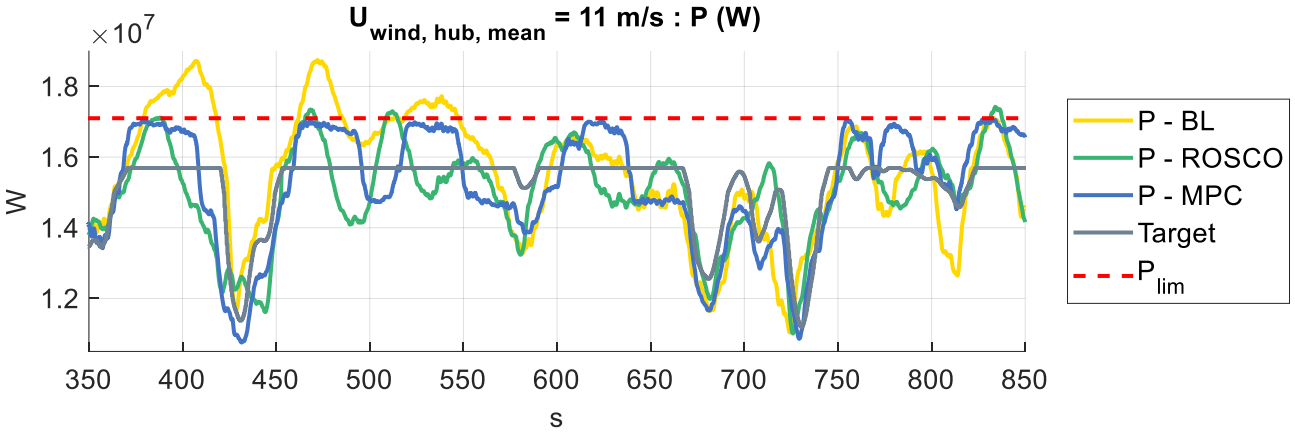

Figure 7.26: Extracted power trends for the three different controllers when simulated with an average speed at hub height of 11 m/s.

Another fact to consider is that the MPC controller achieves the power objectives no much worse than the other two, but performs better with regard to the angular pitch speed of the structure, for which it is the controller that performs overall better. In addition, regarding the use of actuators for varying the blade pitch angle, there are consistently better results than with the ROSCO controller, which is another point in favour of the MPC controller. This ties in with the discussion about finding the settings that give the best overall performances. In fact, focusing more on power, for example, would probably result in higher average powers for the MPC than for the other controllers, but worse values for other indices, such as structure stability and blade pitch actuator wear.

Above Rated Conditions

This section will analyse the performance of the three controllers when the wind speed is above the nominal wind speed, which in this case is approximately 10 m/s. In this case, the main objective is to track the nominal power, but other objectives also are important, such as limiting the pitch rate of the structure and limiting the use of the blade pitch angle Furthermore, in this zone it is fairly important not to exceed the maximum power, which in this case is set at 110% of the nominal power, for the reasons explained above.

Below are some graphs comparing the performance indices for these conditions for the three controllers examined. They are divided according to the simulations carried out, each of them with a different average hub height wind speed ($U_{wind,hub,mean}$). Notice that the input wind speeds at the various grid points covering the rotor area are generated from this average speed using the TurbSim software (see section 3.2.1).

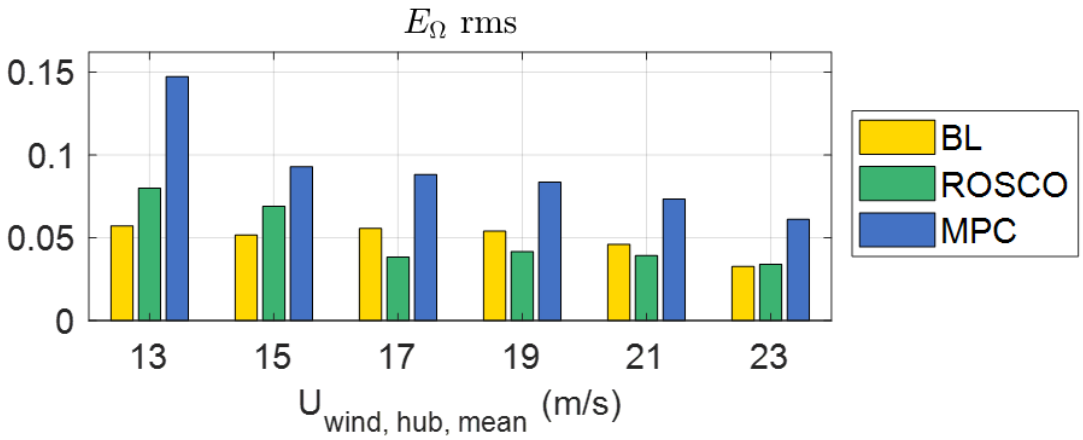

Figure 7.27: Comparison between Baseline, ROSCO and MPC, above rated conditions, $E_{\Omega,\ rms}$.

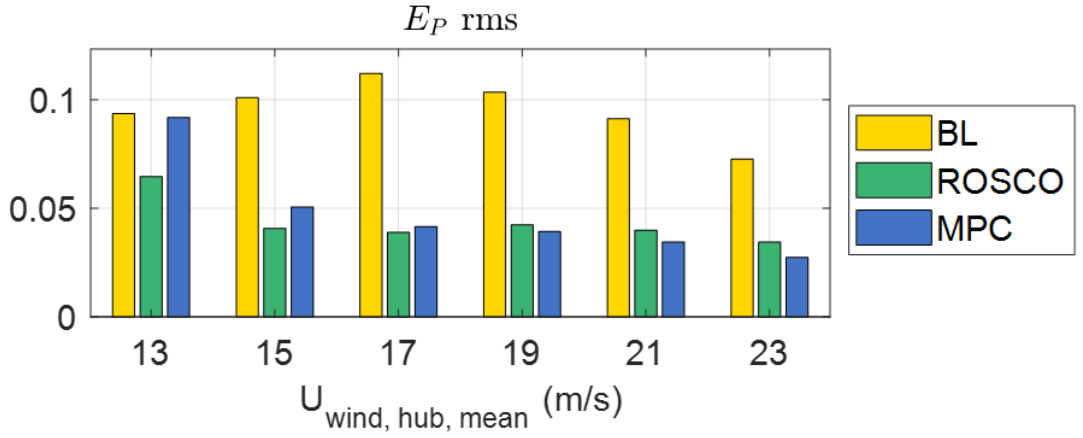

Figure 7.28: Comparison between Baseline, ROSCO and MPC, above rated conditions, $E_{P,\ rms}$.

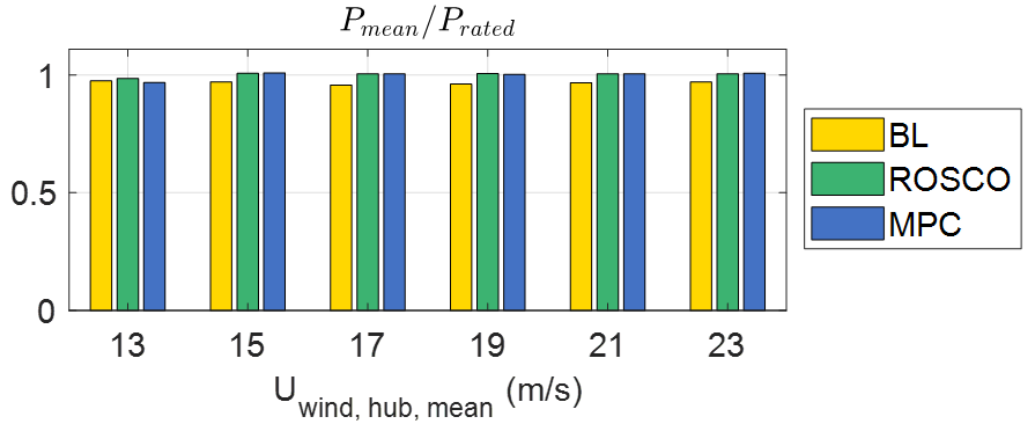

Figure 7.29: Comparison between Baseline, ROSCO and MPC, above rated conditions, $P_{mean,n} = {}^{P_{mean}}/_{P_{rated}}$.

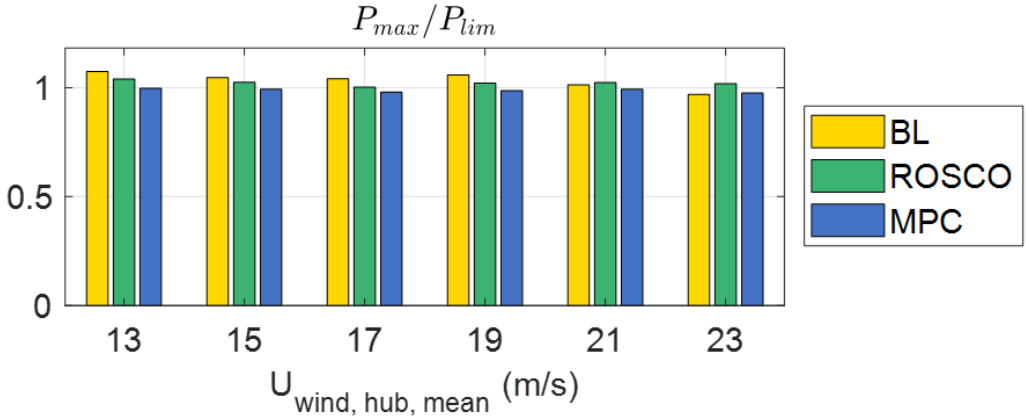

Figure 7.30: Comparison between Baseline, ROSCO and MPC, above rated conditions, $P_{max,n} = {}^{P_{max}}/_{P_{lim}}$.

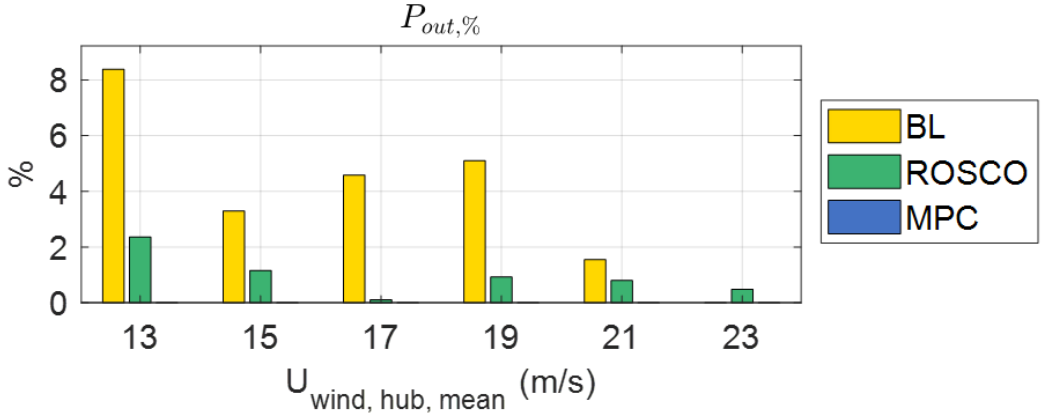

Figure 7.31: Comparison between Baseline, ROSCO and MPC, above rated conditions, percentage of time for which the power exceeds the limit: $P_{out,\%}$.

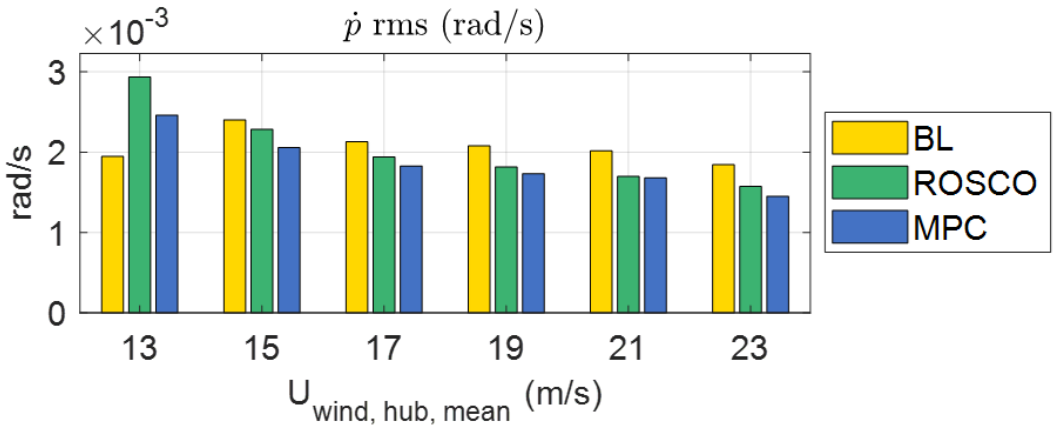

Figure 7.32: Comparison between Baseline, ROSCO and MPC, above rated conditions, $\dot{p}_{rms}$.

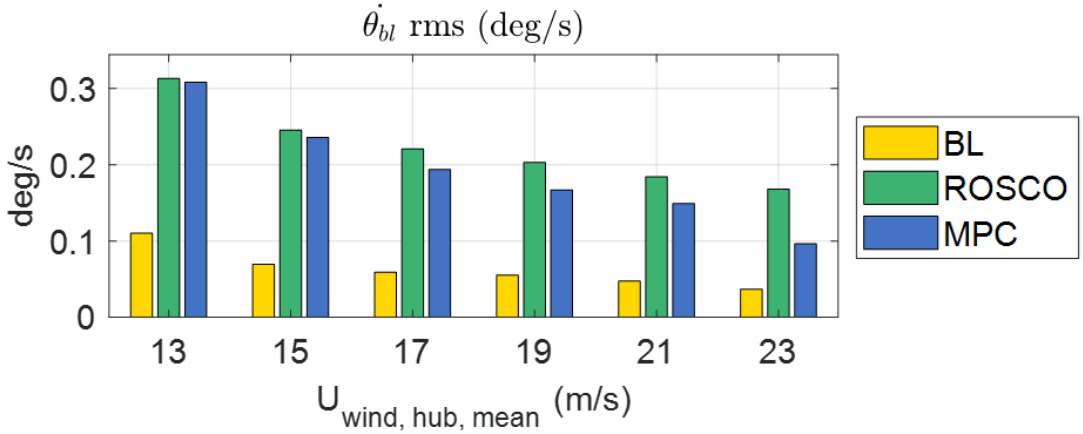

Figure 7.33: Comparison between Baseline, ROSCO and MPC, above rated conditions,$\dot{\theta}_{bl,\ rms}$.

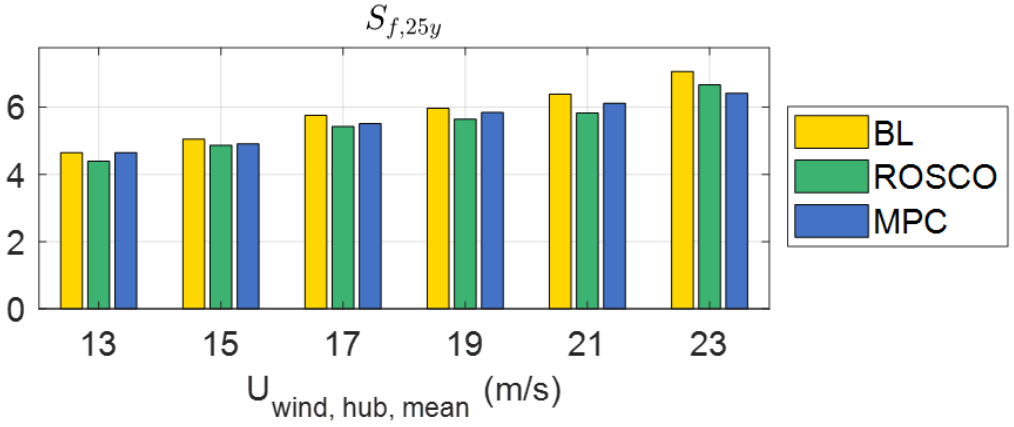

Figure 7.34: Comparison between Baseline, ROSCO and MPC, above rated conditions, $S_{f,\ 25y}$.

Even in the case with wind speeds higher than the nominal wind speed, we can see that the MPC controller overall performs slightly better. Apart from the error associated with rotor speed, for which a low weight is imposed in this area, since the main objective is not to track the latter, the other indicators tell us that, with a few exceptions, the MPC performs better than the other two. For example, the error associated with power is almost always lower, which leads to average power almost always being closer to the nominal one, without ever exceeding the limit, as is the case with the other controllers. As far as the Baseline is concerned, it can be seen that the lowest average power values are almost always obtained, despite the fact that it is the controller for which the percentage time in which the average power exceeds the limit is always the highest, a point against this type of controller. The ROSCO controller, on the other hand, performs better overall than the Baseline in terms of average power and maximum power, but even in this case there are cases in which the power limit is overcome, albeit with low percentages.

With regard to the stability of the structure, which is closely linked to the index relating to its angular pitch speed, the MPC almost always offers the best results, and in any case always better than ROSCO, which is in fact the only one that competes with the MPC since the Baseline performs poorly in terms of power, as pointed out above. Unlike the other indicators, that relating to the safety factor at the root of the blades offers us limited points of reflection, as its trend does not present a constant pattern along wind speeds. Finally, the index relating to the use of the variation in the blades pitch angle tells us that MPC always performs better than ROSCO. The two (MPC and ROSCO), however, always have much higher values of this index than the Baseline, whose strength is that it makes minimal use of this type of control action. This characteristic, however, is of little relevance when other more important performance indices, such as those for average and maximum power, are considerably worse.

8 Conclusions

In recent times, renewable resources have been increasingly exploited, with wind energy being second only to hydro and solar power. Over the last decade, thanks to technological advances, the average cost of energy has come down considerably, making this sector increasingly competitive. Although onshore wind and fixed offshore are more widespread due to their greater development and fewer technological challenges, floating offshore is also starting to progress, also thanks to the possibility of exploiting areas with good wind resources but with a deep seabed and to the possibility of reaching large sizes without visual pollution or environmental impact problems. With the development of floating offshore wind turbines (FOWT), it therefore becomes necessary to adapt control strategies, starting from conventional controllers used in the non-floating counterpart, or developing new types of control, such as those based on PID logic, or unconventional ones.

The objective of this thesis is the realization of an unconventional controller based on MPC (Model Predictive Control) logic. This type of controllers exploits the knowledge of the mathematical model of the system to find the optimal values of the control actions with the aim of maximizing its performances. The main reasons behind the choice of this type of controller is the natural ability to deal with MIMO (Multiple-Input-Multiple-Output) systems and the possibility of having constraints on output quantities, a feature not possessed, for example, by conventional controllers such as those based on PID logic.

After a brief introduction on the current situation of wind energy and the description of the main components of a wind turbine (Chapter 1), the case study adopted for the implementation of the controller was described (Chapter 2).

After that, the mathematical model of the latter was obtained, based on a pre-existing model, and validated through it (Chapter 3). Specifically, the model of a four degree-of-freedom system (3 for the structure: surge, heave, and pitch, plus 1 associated to rotor rotation) was implemented. To obtain it, the equations of motion were written, and some external software were used, in particular, to solve aerodynamic forces (Blade Element Momentum theory), hydrostatic and hydrodynamic forces (NEMOH software and WEC-Sim Matlab-Simulink library) and forces exerted by moorings (MAP software). Once the state-space form of the system's non-linear model was obtained, a second model of the system was built from the first one by applying some simplifications to facilitate the further work.

In a second step, the control strategies for wind turbines were analysed (Chapter 4) and an LTI (Linear Time Invariant) MPC controller was studied from a theoretical point of view (Chapter 5).

Finally, exploiting the knowledge of the system’s model and of the theoretical foundations above mentioned, an MPC controller for floating offshore wind turbines was implemented (Chapter 6). The implemented controller consists of several modules, mainly a numerical optimization module through which the optimal control actions are calculated, two modules for the estimation of external inputs that are not a priori known, one for the wind speed and one for the wave excitation forces, and, finally, a module for the forecasting of the latter forces along the prediction horizon of the controller, which exploits an autoregressive model and a recursive least square algorithm to compute its coefficients.

The optimization and estimation modules (the estimation modules consist of two Kalman filters) are of the discrete linear type and are based on Linear Time Invariant (LTI) models. These models are part of a set of LTI models obtained by linearizing the simplified non-linear model of the system mentioned above, and the idea is to call, at each time step, the most appropriate LTI model according to the conditions in which the physical system is located, i.e. the one obtained from the linear model of the system obtained by linearizing the non-linear model under the conditions closest to the actual conditions. The resulting final MPC controller is thus a set of MPC LTI controllers, each of which is used when most appropriate, based on the value of certain parameters that uniquely identify one of the models obtained by linearization. This technique is made possible by the fact that the controlled system has relatively slow dynamics with respect to the time step with which it changes from one LTI MPC to another, thus neglecting the fact that the matrices adopted for numerical optimization do in fact vary over the prediction horizon. The choice of models to be used in the various modules at each time step is made by a dedicated module, which receives the values of the above-mentioned parameters as input.

Finally, the results from the simulation of the system controlled by the implemented MPC controller were observed and analyzed (Chapter 7), comparing its performance with that of other widely used conventional controllers. In this comparison, it was seen that the implemented controller offers good performance, overall, more satisfactory than that offered by the conventional controllers with which it was compared. It was seen that the logic with which it works, by means of which it is possible to impose limits on output quantities and through which it is possible to explicitly pursue certain objectives, is effective, boding well for future developments in this direction, for example by further improving the controller implemented here or by building more advanced MPC controllers, such as adaptive or LPV MPCs.

9 Future Works

The controller implemented in this thesis has proven to be quite effective, and offers good performances, which are slightly better than those of other conventional controllers used today for the management of wind turbines, such as the Baseline and ROSCO used here as a term of comparison. However, the implemented controller does not differ much from the simplest existing version of MPC, which may suggest that by adopting different versions, better results may be obtained, although it must be borne in mind that such versions may require much more computing time and thus prove to be unfeasible.

Basically, three roads can be pursued for possible future developments, the first by making improvements to the controller implemented here, the second by opting for more complex types of MPC controllers, such as adaptive or non-linear MPCs, and the third by opting for different control strategies.

Regarding the first way, some operations that can be performed may be to improve the various module that make up the controller. For example, a very simple numerical optimisation method (Hildreth's procedure) was used here, but this often leads to sub-optimal solutions if the procedure is stopped to reduce calculation time. It might be a good idea to use different methods that lead to better solutions, albeit taking longer. Regarding the Kalman filters used for estimating external inputs, it might be a good idea to improve the models describing them, especially the one relating to wave forces, which in this case is a simple random walk. Furthermore, if these models were improved, they could be used to predict these quantities and thus integrate the estimation and forecasting modules together, thus being able to carry out forecasting also for wind speed, which is currently missing. On the other hand, improvements to the forecasting module could be made, for example by also implementing this type of module for wind speed, which is currently lacking, since no effective method could be found due to the random nature of this quantity. Instead of using AR autoregressive models or derivatives (e.g., ARI, ARIMA, etc.), other methods not explored here could be adopted, such as neural networks or others. On the other hand, with regard to the switching method used, improvements could be made by interpolating the matrices constituting the linear models used more effectively, e.g., by means of linear interpolation of the latter if the current operating conditions do not correspond to the conditions used for linearization but deviate slightly from them. A good way forward could be to adopt an LPV MPC, for which, however, it is necessary to know the model matrices as the parameters change explicitly, a non-trivial undertaking. Finally, improvements could also be made by adopting models of the controlled system that are more faithful to reality, making fewer simplifications than those adopted here. However, it must be borne in mind that doing so could lead to situations in which the calculation times required to operate the controller could become too high, leading to unfeasible situations.

Regarding the third way mentioned, a good idea might be to implement a control that provides a different pitch angle for each blade. Although this involves more computational effort, through such a technique it might be possible to best combine performance related to extracted power and performance related to structure stability or blade root fatigue. By doing so, one could vary the average angle of the blades to maximize power performance and at the same time exploit the variation between the angles to pursue the other objectives.

Bibliography

[1] GWEC, “Global Wind Report 2022,” 2022.

[2] GWEC, “Global Offshore Wind Report 2022,” 2022. [Online]. Available: www.gwec.net

[3] GWEC, “FLOATING OFFSHORE WIND-A GLOBAL OPPORTUNITY,” 2022.

[4] E. Gaertner *et al.*, “Definition of the IEA Wind 15-Megawatt Offshore Reference Wind Turbine Technical Report,” 2020. [Online]. Available: www.nrel.gov/publications.

[5] C. Allen *et al.*, “Definition of the UMaine VolturnUS-S Reference Platform Developed for the IEA Wind 15-Megawatt Offshore Reference Wind Turbine Technical Report,” 2020. [Online]. Available: www.nrel.gov/publications.

[6] W. E. Cummins, “The impulse Response Function and Ship Motions,” *Technical Report 1661, David Taylor Model Basin-DTNSRDC*, 1962.

[7] T. Perez and T. I. Fossen, “A Matlab Tool for Frequency-Domain Identification of Radiation-Force Models of Ships and Offshore Structures,” 2009.

[8] A. Babarit and G. Delhommeau, “Theoretical and numerical aspects of the open source BEM solver NEMOH. In Proc. of the 11th European Wave and Tidal Energy Conference (EWTEC 2015).” Nantes, France, 2015.

[9] T. F. Ogilvie, “Recent Progress towards the Understanding and Prediction of Ship Motions,” 1964.

[10] A. Betz, “Das maximum der theoretisch moglichen ausnutzung des windes durch windmotoren.,” *Zeitschrift f¨ur das gesamte Turbinenwesen.*, 1920.

[11] T. Burton, D. Sharpe, N. Jenkins, and E. Bossanyi, *Wind energy handbook*. 2001.

[12] J. Ledoux, S. Riffo, J. Salomon, and A. Riffo, “Analysis of the Blade Element Momentum Theory,” *Analysis of the Blade Element Momentum Theory. SIAM Journal on Applied Mathematics, Society for Industrial and Applied Mathematics*, vol. 81, no. 6, 2021, doi: 10.1137/20M133542Xï.

[13] D. Spera, “Wind Turbine Technology: Fundamental Concepts in Wind Turbine Engineering, Second Edition. ASME,” New York, NY, 2009.

[14] P. M. Masciola, J. Jonkman, and A. Robertson, “Implementation of a Multisegmented, Quasi-Static Cable Model,” 2013. [Online]. Available: http://www.osti.gov/bridge

[15] M. Masciola, “MAP++ Documentation Release 1.15,” 2018.

[16] B. J. Jonkman, “TurbSim User’s Guide: Version 1.50,” 2009. [Online]. Available: http://www.osti.gov/bridge

[17] J. M. Jonkman, “Dynamics modeling and loads analysis of an offshore floating wind turbine,” 2007.

[18] International Electrotechnical Commission., *IEC 61400-3 “Wind turbines-Part 3: Design requirements for offshore wind turbines.” Draft 1st edition. Geneva, Switzerland: International Electrotechnical Commission, January 2006*.

[19] Tony Burton, David Sharpe, Nick Jenkins, and Ervin Bossanyi, *WIND ENERGY HANDBOOK*. 2001.

[20] W. J., Jr. and M. L. A. Pierson, “Proposed Spectral Form for Fully Developed Wind Seas Based on the Similarity Theory of S. A. Kitaigorodskii,” *ournal of Geophysical Research*, 1964.

[21] K. Hasselmann and D. Olbers, “Measurements of wind-wave growth and swell decay during the Joint North Sea Wave Project (JONSWAP),” *Ergänzung zur Deut. Hydrogr. Z., Reihe A (8)*, vol. 12, pp. 1–95, 1973.

[22] “Mathworks. MatLab-Simulink. Available online: https://it.mathworks.com/help/control/ug/linear-parameter-varying-models.html.”

[23] M. Sirigu, E. Faraggiana, A. Ghigo, and G. Bracco, “Development of MOST, a fast simulation model for optimisation of floating offshore wind turbines in Simscape Multibody.”

[24] “‘WEC-Sim GitHub Webpage.’ [Online]. Available: https://github.com/WEC-Sim/WEC-Sim.”

[25] T. Salic, J. F. Charpentier, M. Benbouzid, and M. le Boulluec, “Control strategies for floating offshore wind turbine: Challenges and trends,” *Electronics (Switzerland)*, vol. 8, no. 10. MDPI AG, Oct. 01, 2019. doi: 10.3390/electronics8101185.

[26] M. H. Hansen and Danmarks Tekniske Universitet. Risø DTU, *Control design for a pitch-regulated, variable speed wind turbine*. Risø DTU - National Laboratory for Sustainable Energy, 2005.

[27] J. Jonkman, S. Butterfield, W. Musial, and G. Scott, “Definition of a 5-MW Reference Wind Turbine for Offshore System Development,” 2009. [Online]. Available: http://www.osti.gov/bridge

[28] N. J. Abbas, D. S. Zalkind, L. Pao, and A. Wright, “A reference open-source controller for fixed and floating offshore wind turbines,” *Wind Energy Science*, vol. 7, no. 1, pp. 53–73, Jan. 2022, doi: 10.5194/wes-7-53-2022.

[29] Q. Truong, “Continuous-time Model Predictive Control,” 2007.

[30] Liuping Wang, *Model Predictive Control System Design and Implementation Using MATLAB®*. Springer London, 2009. doi: 10.1007/978-1-84882-331-0.

[31] T. Knudsen, T. Bak, and M. Soltani, “Prediction models for wind speed at turbine locations in a wind farm,” *Wind Energy*, vol. 14, no. 7, pp. 877–894, 2011, doi: 10.1002/we.491.

[32] IEC, *Wind energy generation systems – Part 1: Design requirements, IEC 614001-1, 4th edn., Internation Energy Commission, Geneva, Switzerland, 2019*. International Electrotechnical Commission, 2019.

[33] R. P. Louis. Nijssen and F. of A. Engineering. D. and P. of C. S. Group. TU Delft, *Fatigue life prediction and strength degradation of wind turbine reactor blade composites.* Knowledge Centre Wind turbine Materials and Constructions, 2006.

[34] “Mandell, J. F., and Samborsky, D. D., 2008, DOE/MSU Fatigue of Composite Materials Database.”

[35] D. D. Samborsky, T. J. Wilson, and J. F. Mandell, “Comparison of tensile fatigue resistance and constant life diagrams for several potential wind turbine blade laminates,” *Journal of Solar Energy Engineering, Transactions of the ASME*, vol. 131, no. 1, pp. 0110061–01100610, Feb. 2009, doi: 10.1115/1.3027510.

[36] H. J. Sutherland and J. F. Mandell, “UPDATED GOODMAN DIAGRAMS FOR FIBERGLASS COMPOSITE MATERIALS USING THE DOE/MSU FATIGUE DATABASE SAND2011-8342C.” [Online]. Available: http://www.sandia.gov/wind/.

[37] HENRIK SVENSSON, “DESIGN OF FOUNDATIONS FOR WIND TURBINES,” 2010. [Online]. Available: http://www.byggmek.lth.se

[38] O. Y. Miñambres and J. U. Garcia, “Assessment of Current Offshore Wind Support Structures Concepts-Challenges and Technological Requirements by 2020,” 2012.