

POLITECNICO DI TORINO

Department of Engineering
Master of Science in Mechatronic Engineering



Master Thesis

Set-up of a mechatronic test bench
for Hardware-In-the-Loop simulation
of floating offshore wind turbine

Supervisor: Giovanni Bracco

Advisor: Giuliana Mattiazzo

Candidate: Marco Serra
263580

Academic Year 2020-2021

Abstract

The ongoing global environmental crisis is thrusting both the research & development (R&D) and the industry to put an increasing committed effort into the implementation of the energetical transition, with the aim to cut the fossil-fuel related emissions of carbon dioxide within due time, hence avoiding the so-called “tipping points” to be reached.

Floating offshore wind (FOW), among the sources of renewable energy, is the one whose development is proceeding with the highest rapidity. FOW, indeed, is characterized by an enormous potential (in terms of capacity factor and accessibility for new markets) which is in fact still far from being fully exploited - especially in the Mediterranean Sea, where not a single FOW turbine has been deployed and added to the grid yet.

The purpose of this master thesis is to develop and set-up a mechatronic test bench (located in the Energy Center Lab of Politecnico di Torino) intended for the Hardware-In-the-Loop (HIL) simulation of the floating offshore wind turbine *NREL5MW*, with the eventual goal of an actual deployment in a suitable site near Pantelleria Island, in the southern-western Sicilian territorial waters. Concurrently, a Simulink model of the same system (developed in previous works) has been modified accordingly, in order to be compiled and imported in such test bench and therefore be validated in a HIL framework. The first two chapters of this dissertation contain, respectively, a deep introduction about the state of the art and the motivations behind this works, and a complete characterization of both the wind turbine system and its Simulink model.

The third chapter discusses the test bench architecture, which is composed by a hardware plant reproducing the turbine mechanics (asynchronous electrical motor, shaft, torque meter, synchronous generator), whose outputs are meant to represent in a faithful manner the actual output of the simulated device. On the other hand, the test bench is controlled via NI CompactRIO (cRIO-9040), which is programmed in NI VeriStand. The data transfer between the cRIO controller and the mechanical plant is implemented in a high-speed and deterministic

fashion thanks to the use of the EtherCAT communication protocol. The fourth chapter deepens the whole programming procedure and the tests that have been performed to validate the motor behavior when controlled by means of the selected controller.

The fifth chapter, instead, describes the compiling workflow performed to import the original *NREL5MW* wind turbine Simulink model within the VeriStand environment. In the same chapter, three different cross-simulations are reported, that are meant to compare the VeriStand model with its Simulink counterpart in different environmental conditions.

As a result to this thesis work, the developed test rig proved to be a powerful tool to validate the control logic of the wind turbine in an effective way, providing a support for the engineers that will design the controller in the actual turbine; nevertheless, the research still has to be completed in future works. In particular, the bench needs to be completed with part of the plant hardware (torque meter, generator) and the EtherCAT communication between cRIO and drive needs to be refined to enlarge the features and testing flexibility of the test bench.

Table of contents

List of Tables.....	IV
List of Figures.....	V
Chapter one: Introduction and state of the art.....	1
1.1 Scenario.....	1
1.2 Offshore wind power.....	3
1.2.1 Worldwide situation: growth and previsions.....	5
1.3 Floating offshore wind.....	8
1.4 The Mediterranean situation.....	12
1.4.1 The Pantelleria case study.....	15
1.5 The purpose of this work.....	17
Chapter two: The system definition.....	18
2.1 The physical model.....	19
2.1.1 The wind turbine.....	19
2.1.2 The floating platform.....	22
2.1.3 The mooring subsystem.....	23
2.1.4 The overall system and the reference frame.....	24
2.2 The Simulink representation.....	25
2.2.1 Overview.....	25
2.2.2 Wave input.....	26
2.2.3 Wind input.....	28
2.2.4 Definition of the simulation campaign.....	29
2.2.5 Wind turbine.....	30
2.2.6 Moorings.....	34
2.2.7 Hull.....	35
2.2.8 Example simulation.....	37
Chapter three: The test bench setup.....	42
3.1 Hardware-In-The-Loop testing.....	42

3.2 The plant hardware	44
3.2.1 The motor	46
3.2.2 The drive	47
3.2.3 The transmission subsystem	49
3.2.4 The generator	51
3.2.5 Mechanical constraints	53
3.3 The controller	54
3.3.1 CompactRIO	54
3.3.2 Current input module: NI 9253	56
3.3.3 NI VeriStand	58
3.3.4 Host PC	58
3.3.5 Electrical panel and miscellaneous	61
3.4 EtherCAT	61
Chapter four: Test bench startup and first tests	66
4.1 Preliminary operations	66
4.1.1 Set-up of a VeriStand project and target definition	66
4.1.2 EtherCAT connection to the motor drive	67
4.1.3 The drive parameter system and mapping issues	69
4.1.4 Importing time sequences from Simulink to VeriStand and vice versa	71
4.2 The first tests on the motor and PID selection	73
4.2.1 The PID controller: theoretical background	74
4.2.2 PID parameter selection for the test bench	75
Chapter five: The wind turbine VeriStand model	80
5.1 Compiling: preliminary operations and parameter settings	80
5.2 Compiling: problems and solutions	82
5.2.1 Compiling the Wind Turbine block	83
5.2.2 Compiling the Moorings block	84
5.3 The overall wind turbine VeriStand project: simulation campaign	87
5.3.1 First simulation	89
5.3.2 Second simulation	91

5.3.3 Third simulation.....	92
Chapter six: Conclusions.....	94
6.1 Future works and further developments.....	95
Bibliography.....	97
Ringraziamenti.....	104

List of Tables

Table 1.4.1: Features of the three sites of the Pantelleria case study.....	16
Table 2.1.1: Specifications of the NREL5MW wind turbine.....	19
Table 2.1.2: Undistributed blade aerodynamics and structural properties.....	21
Table 2.1.3: Geometrical quantities of the 19 radial substations of each rotor blade.....	21
Table 2.1.4: Drivetrain and generator properties.....	21
Table 2.1.5: Floating substructure properties (hexafloat).....	23
Table 2.1.6: Mooring subsystem properties.....	23
Table 2.1.7: Undistributed blade aerodynamic properties and moments of inertia.....	25
Table 2.2.1: dataset of the triads (wave height and period, wind speed) characterizing the simulation input.....	30
Table 2.2.2: Rated operating point of the NREL 5-MW wind turbine.....	32
Table 3.2.1: Rated quantities and other relevant data of the motor ABB AX 160L.4.....	46
Table 3.2.2: Testing data of the motor, collected by Comer s.r.l.....	46
Table 3.2.3: Working ratings of the drive ABB ACS880-07-0145A-3.....	48
Table 3.2.4: Rated quantities and other relevant data of the transducer 1-T22/500NM by HBM.....	49
Table 3.2.5: Rated quantities of the generator EOGEN 150/16.....	52
Table 4.2.1: PID parameter selection of the Ziegler-Nichols method.....	75

List of Figures

Figure 1.1.1: An example of offshore wind farm	2
Figure 1.1.2: Global emissions of carbon dioxide due to energy production	2
Figure 1.2.1: A technician during a maintenance operation	3
Figure 1.2.2: Comparison of the capacity factors characterizing offshore wind and other energy sources	4
Figure 1.2.3: Global offshore wind capacity in operation in June 2020 (by country)	6
Figure 1.2.4: Global offshore wind capacity in operation (cumulative), from 2011 to June 2020	6
Figure 1.2.5 Average distance from shore of installed offshore wind farms	7
Figure 1.2.6: Rolling average water depth of installed offshore wind farms	8
Figure 1.3.1: Different types of wind turbine foundations, both bottom-fixed and floating	9
Figure 1.3.2: Drone picture of the Hywind Scotland wind farm	10
Figure 1.3.3: Median LCOE cost reduction scenario of wind energy	11
Figure 1.4.1: Total theoretical offshore wind production	14
Figure 1.4.2: Wind atlas of the Italian Mediterranean	14
Figure 1.4.3: Visual pollution of a wind turbine vs. distance from shore	14
Figure 1.4.4: The peculiar location of Pantelleria Island	15
Figure 1.4.5: Wind rose of the Pantelleria Island sea	16
Figure 1.4.6: Location of the three sites that were analyzed in Pantelleria sea	16
Figure 1.5.1: Picture of the test bench and user workstation	17
Figure 2.1.1: Isometric representation of the system	18
Figure 2.1.2: Generic airfoil cross-section	20
Figure 2.1.3: NREL5MW rotor blade	20

Figure 2.1.4: Spar-buoy versus hexafloat.....	22
Figure 2.1.5: Fixed reference frame.....	24
Figure 2.2.1: Simulink model of the system.....	26
Figure 2.2.2: Graphical representation of irregular waves as superposition of different waves.....	27
Figure 2.2.3: Example of the wave input of the Simulink.....	28
Figure 2.2.4: Graphical representation of the three-dimensional wind profile.....	29
Figure 2.2.5: Wind turbine Simulink sub-model.....	30
Figure 2.2.6: Generator-torque control law and main control regions.....	33
Figure 2.2.7: Example comparison between displacements and mooring reactions.....	35
Figure 2.2.8: Example simulation. Input wind speed versus time.....	39
Figure 2.2.9: Example simulation. Displacements of the hull over time.....	39
Figure 2.2.10: Example simulation. Mooring reaction forces versus time.....	40
Figure 2.2.11: Example simulation. Rotor angular speed and blade pitch control versus time.....	40
Figure 2.2.12: Example simulation. PTO (power take-off).....	41
Figure 3.1.1: Schematic representation of the HIL feedback loop structure.....	43
Figure 3.1.2: 3D rendering of the complete HIL test bench structure.....	44
Figure 3.2.1: Scheme of the main mechanical parts of a turbine.....	45
Figure 3.2.2: Three main components of the plant hardware, inserted in the HIL feedback loop.....	45
Figure 3.2.3: Mechanical characteristics of the motor ABB AX160L.4.....	47
Figure 3.2.4: A picture of the motor ABB AX160L.4.....	47
Figure 3.2.5: A picture of the driver ABB ACS880-07-0145A-3.....	48
Figure 3.2.6: Rendered CAD of the torque meter and of the flange.....	49
Figure 3.2.7: The transmission subsystem. 3D rendering.....	50

Figure 3.2.8: The protective structure. Rendered CAD.....	51
Figure 3.2.9: Mechanical characteristic of the generator EOGEN 150/16.....	52
Figure 3.2.10: Efficiency and power characteristics from the datasheet of EOGEN 150/16.....	52
Figure 3.2.11: Comparison of the critical torque of the three main components of the test bench.....	53
Figure 3.3.1: Schematic representation of the controller set-up, within the HIL feedback framework.....	54
Figure 3.3.2: A picture of CompactRIO-9040, by National Instruments.....	55
Figure 3.3.3: A picture of the NI 9253 current input module.....	57
Figure 3.3.4: Conditioning circuit of the input channels of NI 9253.....	57
Figure 3.3.5: Internal circuitry and components of the electrical panel.....	60
Figure 3.4.1: Typical architecture of an EtherCAT-based network.....	62
Figure 3.4.2: FECA-01 EtherCAT adapter module by ABB.....	63
Figure 3.4.3: Schematic representation of the test bench overall architecture.....	65
Figure 4.1.1: EtherCAT slave added to the VeriStand project configuration.....	68
Figure 4.1.2: Example of a correctly formatted CSV file.....	72
Figure 4.1.3: Example of stimulus profile code.....	73
Figure 4.2.1: Schematic representation of a PID control loop.....	74
Figure 4.2.2: Step response of the designed PI controller.....	77
Figure 4.2.3: PID parameter selection. Post-processed step response in three different cases. Zoomed view.....	77
Figure 4.2.4: Motor speed performances. Square wave response.....	78
Figure 4.2.5: Motor speed performances. Sine wave response.....	78
Figure 4.2.6: Motor speed performances. Filtered response to an example generator speed response. Original and zoomed views.....	79
Figure 5.2.1: Original Simulink model of the moorings.....	85

Figure 5.2.2: Original architecture of the Simulink model of one of the six mooring lines.....	85
Figure 5.2.3: New version of the mooring Simulink model.....	86
Figure 5.3.1: Overall structure of the NREL5MV wind turbine model rearranged within VeriStand.....	88
Figure 5.3.2: Simulation 1. VeriStand PTO: generator speed and generated power.....	90
Figure 5.3.3: Simulation 1. Output speed compared to actual motor speed.....	90
Figure 5.3.4: Simulation 2. Expected behavior from Simulink. PTO: generator speed and generated power.....	91
Figure 5.3.5: Simulation 2. VeriStand PTO: generator speed and generated power.....	91
Figure 5.3.6: Simulation 3. Expected behavior from Simulink. PTO: generator speed, generator torque and generated power.....	92
Figure 5.3.7: Simulation 3. VeriStand PTO: generator speed and generated power.....	93
Figure 5.3.8: Simulation 3. Output speed compared to actual motor speed.....	93

Chapter one

Introduction and state of the art

The first Chapter of this thesis dissertation illustrates the state of the art of offshore wind, also providing an insight into its features and its enormous potential, with a focus on its advantages and its current weaknesses. Moreover, the present worldwide situation of the offshore wind industry is highlighted.

In paragraph 1.4, the focus is shifted on the floating offshore wind sector. The differences between floating and fixed offshore wind are described thoroughly, along with the challenges that such subsector poses.

Subsequently, paragraph 1.5 regards the current situation in the Mediterranean Sea, whereas paragraph 1.6 presents the case study of which this thesis is part: the project and design of a wind farm in the waters of Pantelleria Island, Italy. Eventually, the last paragraph of this introduction explicates the specific purpose of this thesis work.

1.1 Scenario

As we face the third decade of the twenty-first century, the frailty of planet Earth as a whole environment is today more exacerbated than ever. The wide majority of the scientific community claims that the responsibility of this scenario is mainly due to the behavior of humankind as a species, especially for what concerns air pollution and the related climate change that is progressively (and with an increasingly fast pace) destabilizing the narrow equilibria on which all life in nature is based.

According to the United Nations, in the next decades the number of human individuals is expected to increase sharply, with an exponential trend, and to reach the huge total number of eleven billion of units within the end of

this century [1]. Such upcoming demographic growth leads to the urgent necessity of accelerating the decarbonization of energy generation, in order to adequately tackle the rise in the worldwide consumption.



Figure 1.1.1: An example of offshore wind farm.

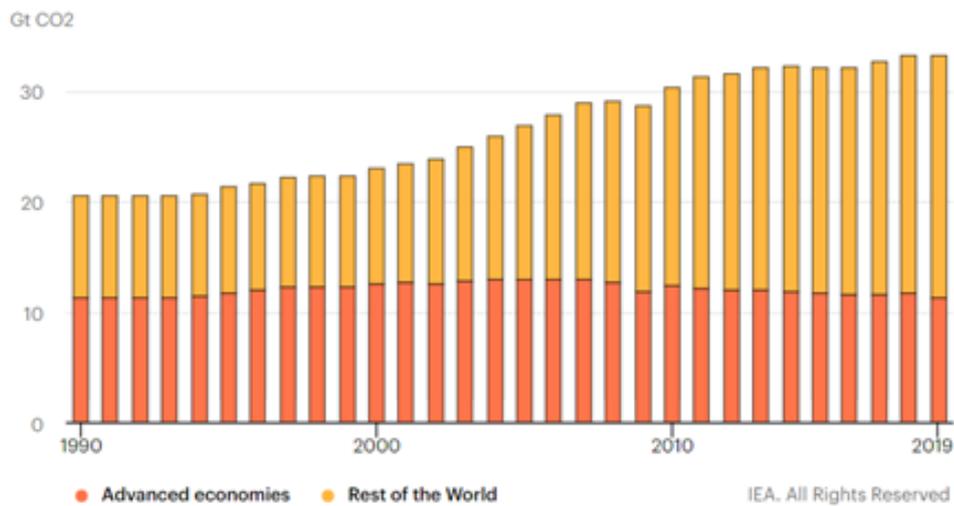


Figure 1.1.2: Global emissions of carbon dioxide (in gigatons) due to energy production [2].

Even though this criticality is nowadays well known, and despite a very sharp increase in the renewable sector, it is clear that there is still a long way to go in this direction. In 2018, the global emissions of CO₂ due to energy production reached an unprecedented peak of 33.3 gigatons per year (Figure 1.1.2) [2]. Moreover, fossil sources are still used to satisfy almost two thirds of the global electrical demand, and this proportion is not changed in the last decades because, even though the overall usage of renewable sources is increasing steadily, the same is happening for fossil fuels as well [3].

1.2 Offshore wind power

Offshore wind is based on the usage of wind farms located on water, usually in the ocean (where the wind average speed is higher than on land), seas, lakes and fjords.

Even though wind power has been used for thousands of years (the first sail boats date back to the 6th millennium B.C., while the first inland windmills have been constructed in the 900 B.C.), the idea of harvesting energy from a turbine located offshore is relatively new. Indeed, the first offshore wind farm was installed in 1991 in Vindeby, Denmark, and it was decommissioned in 2017, after a quarter of century of useful life. The reasons of this late development are due to the relatively high cost with respect to other energy sources, and to the technical difficulties for installation and maintenance, inherently linked to the offshore location and affecting both the workers (difficulty of access, harsher and more dangerous conditions, Figure 1.2.1) and the turbines themselves (humidity and saltwater lead to corrosion and oxidation, affecting the service life of the units).



Figure 1.2.1: A technician during a maintenance operation.

On the other hand, the position itself also guarantees to offshore wind energy some peculiarities that make this power source extremely valuable. First of all, the good quality wind resource implies an electric generation higher per amount of capacity installed. According to the International Energy Agency [4], if fully developed this sector would be able to generate more than 420 000 terawatt-hours per year (TWh/year), an amount of energy 18 times bigger than

the current electricity demand of the whole planet. The estimation is optimistic, because the constraints related to market and distribution (such as the availability of transmission infrastructure) have not been considered – yet this analysis gives a glimpse of the enormous potential of offshore wind energy.

Another advantage of the position of offshore wind farms is the fact that in the high seas, the breeze can be quite strong even in the afternoon, matching the moment of the day in which the most of electricity is employed. Last but not least, being the installed turbines far from the coast, no land is consumed, and at the same time the impact on the landscape is relatively low. As a consequence of these facts, historically offshore wind has encountered little opposition from NIMBY movements (“Not In My Back Yard”).

Those are not the only advantages of offshore wind compared to other energy sources. Indeed, among the renewables, it is characterized by the highest capacity factor, i.e. the ratio of electricity generated in a certain period to the maximum energy output that could have been generated in the same amount of time. More precisely, the most recent offshore wind farms are characterized by capacity factors up to 50% [4], thanks to the steady technology improvements. Such percentage is at the same level of gas-fired power plants and also coal-fired ones, in some regions. With respect to the other renewables, in average offshore wind is more efficient than onshore wind and almost twice as efficient as solar photovoltaics (the respective average capacity factors in 2018 were 33%, 25% and 14%, Figure 1.2.2).

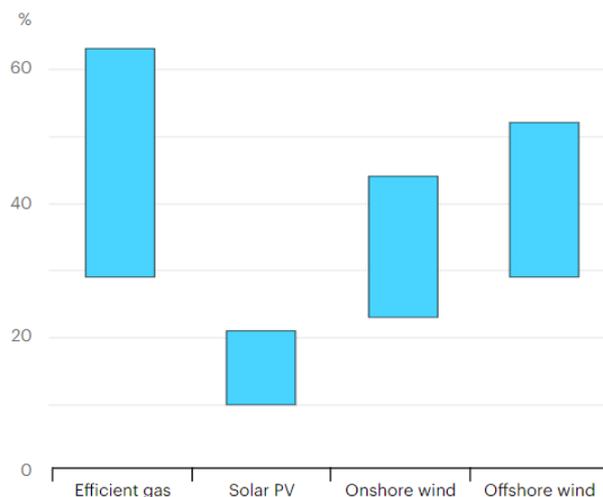


Figure 1.2.2: Comparison of the capacity factors characterizing offshore wind and other energy sources [4].

Like all renewables, offshore wind presents a certain variability, due to the natural variations in strength and direction of the wind; by the way, its hourly variability (20%) is lower than that of solar photovoltaics (40%). This means that the intermittency in the power generation, that is one of the main drawbacks of both solar PV and onshore wind, does not affect offshore wind in the same way.

Due to its high capacity factor and low hourly variability, offshore wind is classified by the International Energy Agency as the only “variable baseload technology” [4], since its working efficiency is effectively similar to that of baseload technologies.

1.2.1 Worldwide situation: growth and previsions

Offshore wind industry presented a constant growth that has lasted for the whole decade. Since 2010, indeed, this sector has grown roughly of the 30% per year, yet it still provides a low percentage of the overall global power generation (only 0.3% of the total in 2018). Nevertheless, in the next decades the offshore wind market is expected to expand, reaching a global business of one trillion of US dollars [4].

Today, globally, offshore wind capacity reaches almost 30 gigawatts (GW) [5]. Worldwide, 157 offshore wind farms are currently operative: 105 of them are situated in Europe (with a cumulative installed capacity of 22.1 GW in 2019 [6]), 50 in Asia, 2 in USA. In particular, the European Northern countries are the ones leading the industry. In the North Seas, indeed, the environmental features (namely the quality of the wind resource and the relatively shallow water), the technical knowledge, the political commitment and the economic investment guaranteed exceptional conditions for the development of offshore wind.

Going into the detail of the single countries, the UK is the absolute biggest market, with more than 10 GW of installed capacity. Germany holds the second place with 7.7 GW, and third is China with 6.4 GW (Figure 1.2.3).

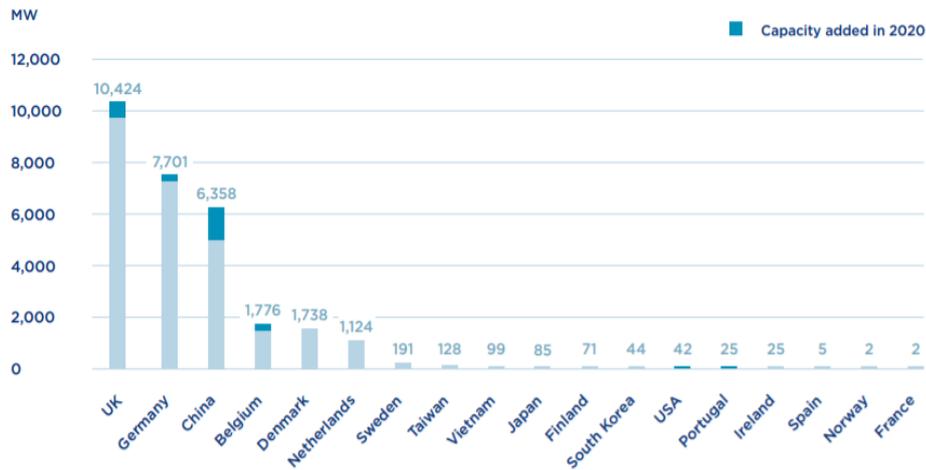


Figure 1.2.3: Global offshore wind capacity in operation in June 2020 (by country) [5].

The steady growth of offshore wind has not been affected severely by COVID-19 (Figure 1.2.4) [5]. In the first semester of 2020, indeed, more than 2.5 GW have been added to the grid (against the 5.1 GW of the whole year 2019), with a total of 10 new farms that went into operation. The country showing the steepest rise is China (1.4 GW of installed capacity from January to June 2020, roughly 56% of the global overall), and at the same time China is also, by far, the nation with the highest amount of capacity currently under construction (4.6 GW, against the Netherlands, second with 1.5 GW and the UK, third with 857 MW). For what concerns the EU, several policies have been stated to foster the growth of the sector, aiming to enlarge the European offshore wind installed capacity up to 80 GW within 2030, and to more than 125 GW within 2040 [4].

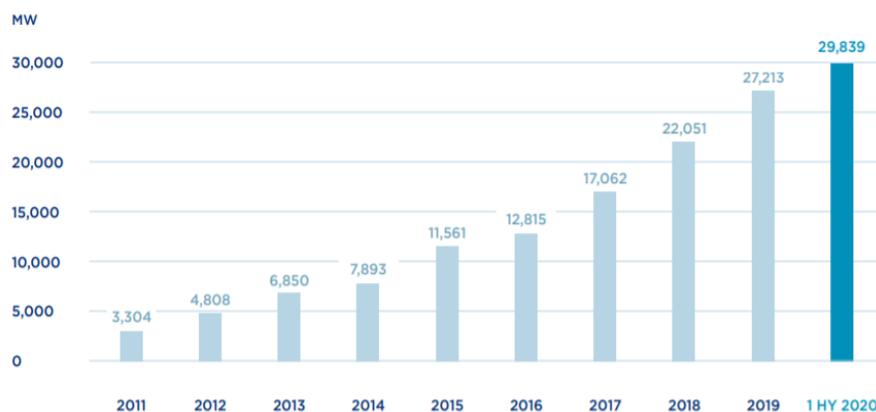


Figure 1.2.4: Global offshore wind capacity in operation (cumulative), from 2011 to June 2020 [5].

From a technical point of view, offshore wind is undergoing a continuous improvement [6] regarding several aspects.

The rated capacity of the installed turbines is increasing steadily. For example, the average rated capacity in 2019 was 7.8 MW per turbine, roughly 1 MW bigger than 2018. The capacity depends on many features, but mainly on the dimensions of the turbine: to give some figures, the blades of a 6 MW turbine are usually around 60 meters long, while those of an 8 MW turbine are approximately 90 meters.

Similarly, also the average size of wind farms has shown a significant development, almost doubling in one decade and reaching 621 MW in 2019. This improvement is possible due to the continuous enhancing of the technology, including more efficient substations, dynamic cables, larger generators and so on.

Finally, one of the most interesting development regards the distance from shore and the bathymetry. In general, indeed, the new wind farms are moving farther offshore (Figure 1.2.5) and into deeper waters (Figure 1.2.6). This trend is confirmed by the data [6]: the farms under construction in 2019 were in average 59 km away from shore, against the average of 35 km in 2018; on the other hand, the average water depth increased from 30 m to 33 m in the same year.

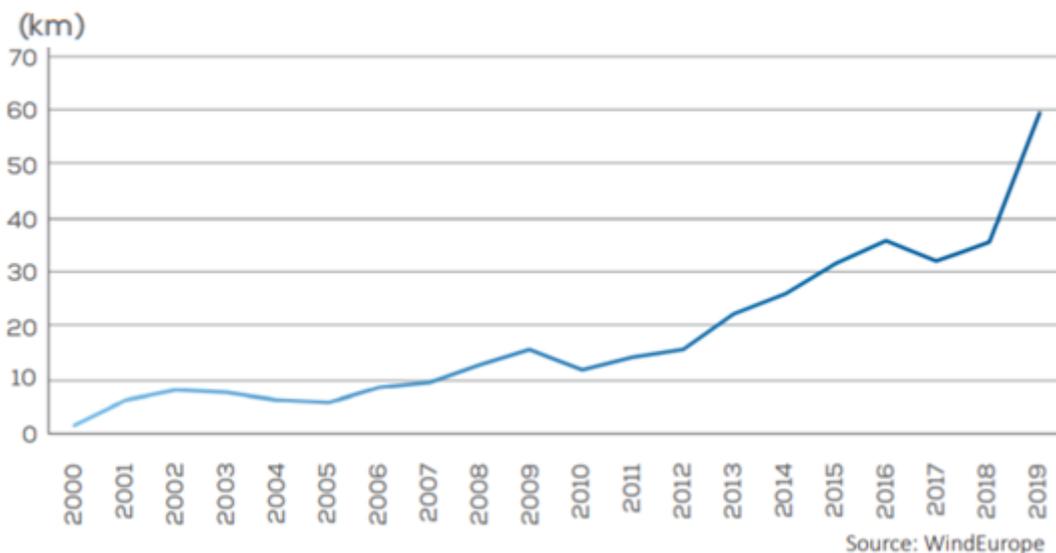


Figure 1.2.5: Rolling average distance from shore of installed offshore wind farms [6].

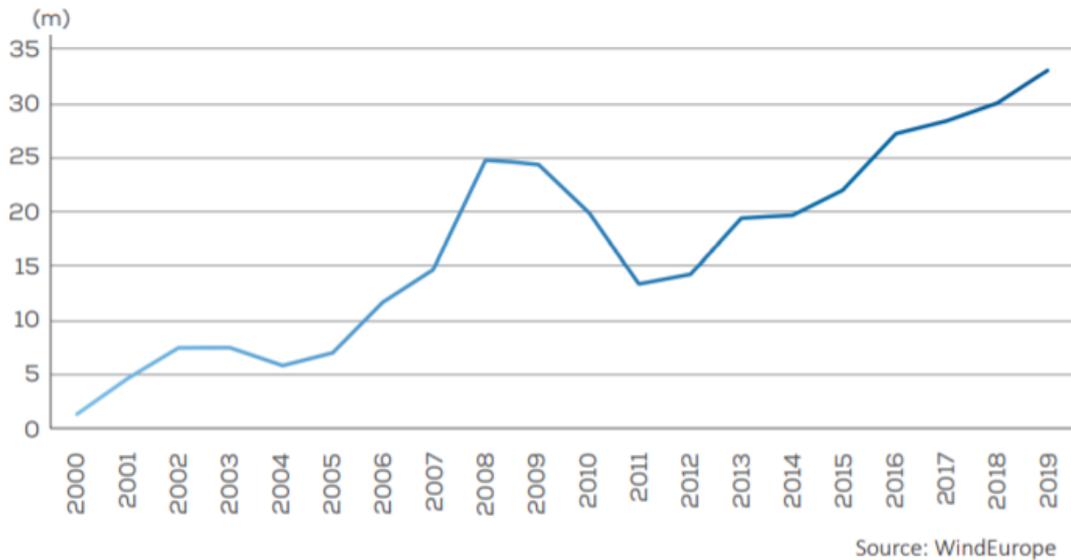


Figure 1.2.6: Rolling average water depth of installed offshore wind farms [6].

These trends are motivated by many reasons. First and foremost, in the high seas generally the wind resource is characterized by bigger capacity factor and better stability, leading therefore to greater efficiency. At the same time, the depletion of near-shore locations is pushing to install farther away the new turbines. On the other hand, the increasing water depth is crucial for some of the largest potential markets (such as United States and Japan) that possess few shallow-water locations. Lastly, farther wind farms are less impacting, both visually and in terms of noise.

As a consequence, an increasingly significant effort is being put into the development of floating offshore wind.

1.3 Floating offshore wind

Floating offshore wind (FOW) is the last frontier of offshore wind [8]. While the latter uses turbines rooted to the seabed either with monopile or jacket foundations, FOW consists in turbines with floating foundations. Solutions of this kind are already well known and widely deployed in the oil & gas sector, that quite often requires the employment of floating platforms even in harsh environments. However, platforms for FOW require adaptation due to its

distinct dynamic characteristics and loading pattern. Several concepts of floating structures have been developed in the last years, and the most common in the industry are spar-buoy, semi-submersible (Semi-Sub) and tension leg platform (TLP, Figure 1.3.1) [9].

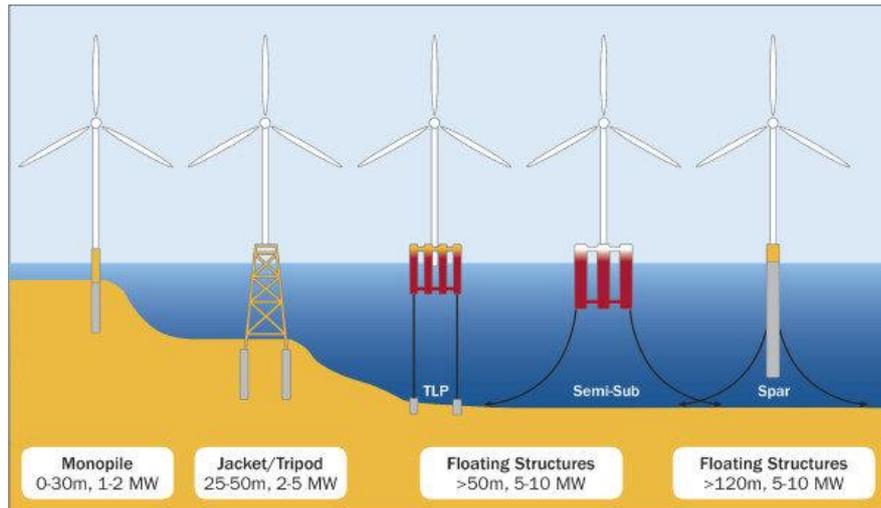


Figure 1.3.1: Different types of wind turbine foundations, both bottom-fixed and floating. [9]

Whereas offshore wind is restricted to waters up to 50 m deep, FOW allows access to deep-water sites, with all the consequent advantages that have been analyzed in the previous paragraph. Additionally, according to the International Renewable Energy Agency, even in mid-depth conditions (30-50 m) floating platforms may in time represent a low-cost alternative with respect to fixed-bottom foundations, thanks to the potential standardization of the design of the turbines and their simpler deployment. Moreover, floating foundations generally represent a less invasive solution with respect to fixed-bottom designs, because the installation does not imply any damage to the seabed.

Despite being a possible game changer in the renewable energy sector, floating offshore wind must face some barriers to development, mainly related to its cost. Often the smaller companies do not have the economic resources to push their innovative products; on the other hand, the financial features of FOW generally do not match the needs of most private investors. Indeed, the required investment is generally very high (up to hundreds of millions only for the pre-

commercial projects), and the period required to attain profitability is longer. As a result, global energy majors are often not inclined to invest in a business with such a long path to market.

Nevertheless, huge steps ahead have been made lately, in a relatively short interval of time. The idea of floating wind turbine has been introduced in 1972 by William E. Heronemus, professor at the University of Massachusetts Amherst. After that, only in 2007 the first prototype in the world was deployed, by Blue H Technologies (Netherlands). The chosen location was a spot 21.3 km off the shore of Apulia, in waters 113 m deep.

Eventually, in October 2017, the Norwegian state-owned energy company Statoil (now Equinor) commissioned the first worldwide commercial floating offshore wind farm, in UK. Its name is Hywind Scotland (Figure 1.3.2), it is situated 25 km off the Scottish shore, in a spot with a maximum water depth of 120 m, and it has a total capacity of 30 MW (5 turbines, 6 MW each). According to the data shared by Equinor, over the first two years of operation its average capacity factor was 56% [11].

Currently, FOW is a mature technology, no longer confined to Research & Development, and it has already started to be integrated into the energy market. It represents the most powerful renewable resource to lead the energy transition and to accelerate the industry development - both in Europe and globally. However, the right conditions for FOW to fully unleash its potential have not been created yet.



Figure 1.3.2: Drone picture of the Hywind Scotland floating offshore wind farm. [11]

First and mostly, the costs must fall significantly in the years to come. Commitment from institutions, policy makers, investors, researchers and industry is imperative to allow such cost reduction, concerning structures and technology (moorings, electrical cables and grid connections) but also facilities, infrastructures and processes. A robust market strategy will be crucial, and it will likely require the exploitation of suitable steppingstone markets.

Nevertheless, FOW travels on a clear path towards a wide deployment, thanks to the lessons recently learned in the other sub-sectors of wind energy. The experience made in the last decade showed that speeding up the commercialization implies a significant cost reduction; moreover, FOW will certainly benefit from the downward trend occurred in both onshore wind and bottom-fixed offshore wind. Indeed, FOW will be able to exploit the already well-developed economies of scale of such sectors.

The International Energy Agency experts claim that the LCOE (acronym for Levelized Cost of Energy, i.e. an index of the average cost of electricity generation cost, taking into account all the cost items of the lifetime of the considered technology) related to FOW will halve within 2050, while WindEurope says that it is expected to decrease by 38% over the same period (Figure 1.3.3) [12]. This diminishment will also be fostered by some technical aspects such as the high capacity factors of the turbines and their continuous growth in size.

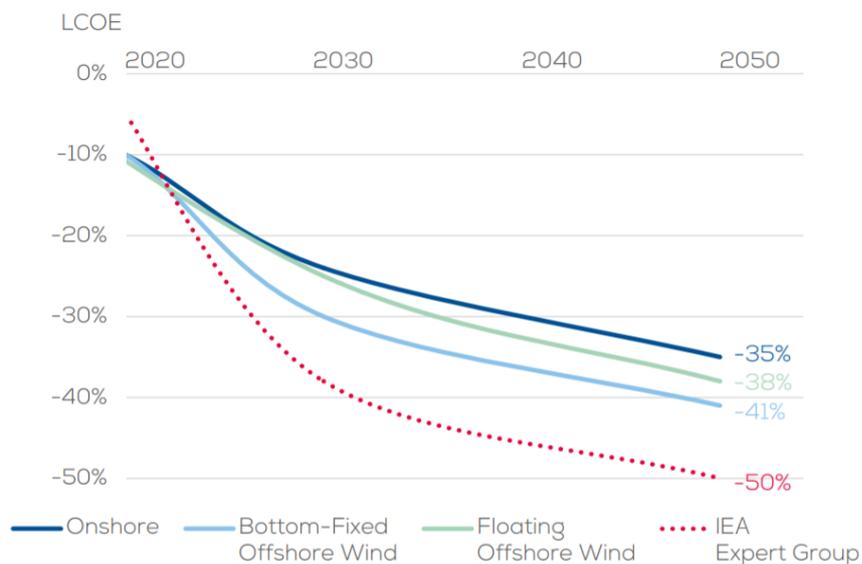


Figure 1.3.3: Median LCOE cost reduction scenario of wind energy. [12]

1.4 The Mediterranean situation

So far, in this dissertation, the Mediterranean Sea has not been mentioned. The reason is quite simple: at present, the number of offshore wind farms in operation in the whole Mediterranean region (including Italy) is zero. This implies that all the achievements reached so far by Europe were in fact accomplished either in the Northern Seas (namely North Sea, Baltic Sea, Norwegian Sea) or in the Atlantic Ocean. Some countries of the Mediterranean basin are actually already in the offshore wind market (France, Portugal, Spain), though such offshore farms are all installed in their Atlantic territorial waters.

This gap is due to the particular features of the Mediterranean area. It is well known that the quality of the wind resource is in general much higher in proximity of the Poles; moreover, the morphology of the Mediterranean does not allow the generation of powerful winds, as it happens in the open ocean [14]. The result is that, in average, the capacity factors that could be attained in the Mediterranean are lower, and therefore an investment is less attractive in comparison with other locations.

Another reason that prevented the development of offshore wind in the region is bathymetry. Indeed, the windiest areas of the Mediterranean are often deeper with respect to the most exploited seas. In this regard, it is clear that the incoming growth of floating offshore wind (FOW) will be a significant turning point for the industry in southern Europe.

The Mediterranean Sea is in fact definitely not poor from a wind resource standpoint. According to Pantusa and Tomasicchio [15] (2019), the total theoretical annual offshore wind energy production of the whole Mediterranean is 741.87 TWh/year. This study has been performed on the basis of geographic information, bathymetric data and average wind speeds, and it does not take into account further concrete constraints such as vessels traffic, presence of ports, underwater power grids, and so on. Nevertheless, the results are impressive and encouraging, because even though the path to market in the Mediterranean Sea is long, the energetic potential of this region is abundant, and for this reason the development of offshore wind must be pursued.

In the research of Pantusa and Tomasicchio, moreover, the incidence of the single countries on the overall energy production is quantified. Italy, in particular, holds the first position and it stands on 21% of the total (Figure 1.4.1), which equals to roughly 155.79 TWh/year. Considering that in 2019 the total energy generation in the country was 284.0 TWh [16], it follows that offshore wind, alone, would be able to satisfy almost half of the overall Italian need for electricity.

In the Italian Mediterranean Sea, the environmental and geographical features are suitable for the deployment of offshore wind farms in broad areas of the Ionian, Adriatic and Tyrrhenian Sea; in particular, the quality of the wind resource is optimal in the South-West of Sardinia and Sicily (Figure 1.4.2), where the average wind speed over a year reaches roughly 10 meters per second (m/s).

The upcoming rise of offshore wind is a great opportunity for Italy, which has the chance to tackle the energetic transition and to capture at the same time a steadily growing market which would ensure to the Italian industrial sector many benefits in the years to come. In the recent past, several projects for offshore wind farms in the Mediterranean have been developed and are currently undergoing the last pre-construction steps.

The three main projects currently waiting for approval are located in Apulia (near the port of Taranto), Sardinia (in the southern-western waters), and in Sicily (35 km off Marsala, in the South-West). All those projects have been debated for a long time, and even though their design is in fact terminated, the start of their construction has been postponed several times. The reasons of this delay are related to bureaucracy on one hand, and on the other one to the alleged impact on wildlife, landscape, and environment.

The effect of offshore wind on wildlife, in particular, remains an unclear argument. Indeed, despite the fact that scientific research has addressed thoroughly such topic in the last decades, the results are still ambiguous [18]. For this reason, the impact of this technology needs further assessments, also through empirical evidence, in order to clarify the exposure of animals (both marine and avian) to offshore wind.

Instead, for what concerns the visual pollution, obviously the greater is the distance from shore of the wind farm, the smaller will be the effects on landscape (Figure 1.4.3).

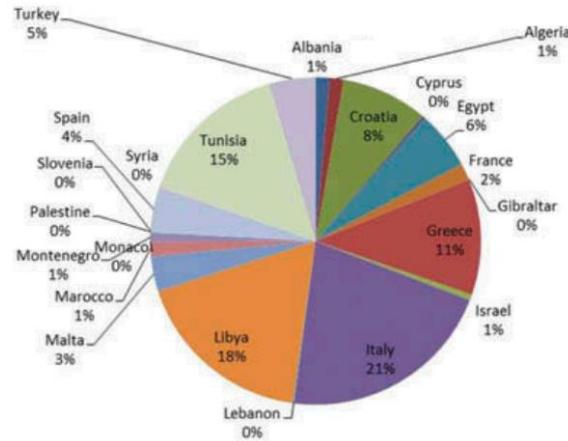


Figure 1.4.1: Total theoretical offshore wind production, percentage by country. [15]



Figure 1.4.2: Wind atlas of the Italian Mediterranean. [17]



Figure 1.4.3: Visual pollution of a wind turbine vs. distance from shore.

1.4.1 The Pantelleria case study

This thesis work is inserted in the context of the development of a floating offshore wind farm in the Italian Mediterranean Sea. In past studies, the choice of the most suitable location has been performed [19], and the designated site is in Pantelleria Island, Sicily. This small island is situated in the Strait of Sicily, 100 km away from the Sicilian shore and 60 km from the Tunisian shore (Figure 1.4.4). As stated in the previous paragraph, the South-West of Sicily is one of the locations among the Italian territorial waters with the greatest wind resource. Pantelleria is a volcanic island, hence its waters are relatively deep. This makes the location suitable for the deployment of floating turbines.

The choice has been performed on the basis of all the relevant parameters, that are, in summary:

- Productivity (considering as reference a NREL 5 MW wind turbine and the wind rose in Figure 1.4.5);
- Distance from shore and sea depth;
- Other (geo-politics, air and sea traffic, wildlife).

After a first survey, the three best sites were chosen (Figure 1.4.6 and Table 1.4.1). Site B, although being the one with the highest productivity, has been discarded because it is quite close to the shore. Also, its sea depth is much bigger than the other two, and this leads to bigger mooring costs. Eventually, between sites A and C, the designated one was A, because of its slightly greater efficiency.



Figure 1.4.4: The peculiar location of Pantelleria Island (in red).

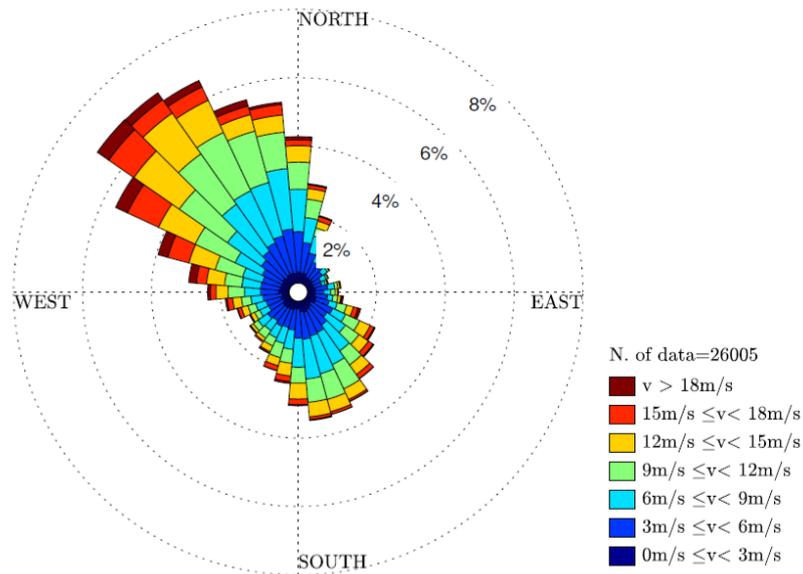


Figure 1.4.5: Wind rose of the Pantelleria Island sea.

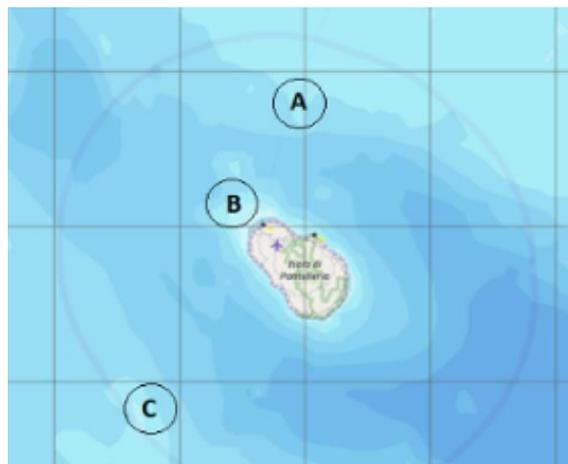


Figure 1.4.6: Location of the three sites that were analyzed in the Pantelleria sea.

	Site A	Site B	Site C
Sea depth	100 m	250 ÷ 500 m	100 m
Distance from shore (m)	21.5 km	8.3 km	26.5 km
Productivity	4235 MWh/MW	4320 MWh/MW	4205 MWh/MW
C.F.	48.3%	49.3%	48.0%
Coordinates	37.01N 12.02E	36.54N 11.54E	36.37N 11.48E

Table 1.4.1: Features of the three sites of the Pantelleria case study.

1.5 The purpose of this work

The work presented in this thesis dissertation is aimed to set up a mechatronic HIL test bench for testing and validating the simulation model and the control logic of wind turbines that will possibly be deployed in Pantelleria, in the discussed site. Those turbines would compose a floating offshore wind farm intended to provide electrical energy to the island, possibly satisfying its whole energetic demand.

The task has been performed in the Energy Center, in Via Paolo Borsellino 38/16. Such structure is an inter-departmental facility of Politecnico di Torino, launched in 2016 to host research and development activities in the field of energy technology, with the goal of a more sustainable society. In particular, the test bench has been built in the EC-Lab, the dedicated laboratory of the Energy Center. Moreover, the whole work has been supported, during all its stages, by MORE (Marine Offshore Renewable Energy Lab, a multi-disciplinary research group whose headquarters is hosted by the Energy Center), and by DIMEAS (Dipartimento di Ingegneria Meccanica e Aerospaziale).



Figure 1.5.1: Picture of the test bench and user workstation

Chapter two

The system definition

In this Chapter, the system that has been modeled and simulated is described thoroughly. The Chapter is subdivided in two sections: in the first one, the wind turbine of interest is presented from a physical standpoint, along with several tables that depict in detail all the structural data. In the second subsection, on the contrary, the Simulink model of the system is illustrated.

The topics treated in this Chapter have been deepened in previous MSc theses [19] [20]. They are reported here because they represent the starting point from which this thesis work has been developed.

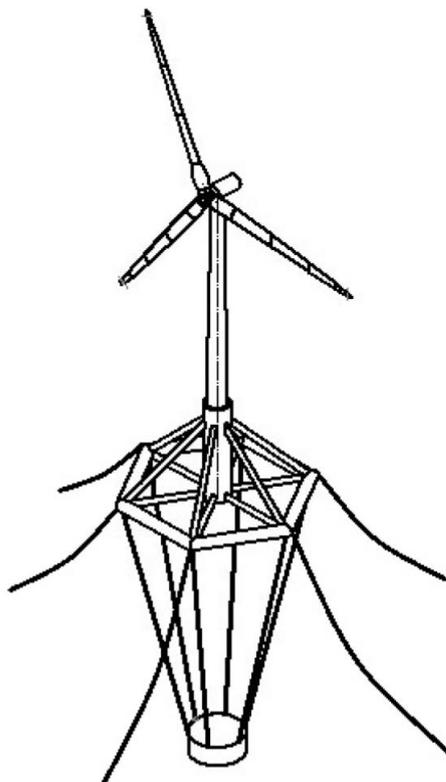


Figure 2.1.1: Isometric representation of the system.

2.1 The physical model

The system, in first approximation, consists in three main elements:

1. Wind turbine (WT);
2. Floater (or hull);
3. Mooring lines.

These three parts interact among themselves (Figure 2.1.1) and at the same time with the surrounding environment. In the scenario of a floating offshore WT, the environment affects the system by means of both wind and waves (that are in this case considered as inputs of the system).

2.1.1 The wind turbine

The chosen WT is the *NREL offshore 5-MW baseline wind turbine* (*NREL5MW* for short). It is a standardized turbine for offshore wind [21], developed by the National Renewable Energy Laboratory (NREL) of the US Department of Energy. In Table 2.1.1 all its specifications are summarized.

Rating	5 MW
Rotor orientation, configuration	Upwind, three blades
Control	Variable speed, collective pitch
Drivetrain	High speed, multiple-stage gearbox
Rotor, hub diameter	126 m, 3 m
Cut-in, rated, cut-out wind speed	3 m/s, 11.4 m/s, 25 m/s
Cut-in, rated rotor speed	6.9 rpm, 12.1 rpm
Rotor mass (hub mass)	110 t (56.78 t)
Nacelle mass	240 t
Tower mass	250 t
Overall mass	600 t
Hub inertia about rotor axis	115,926 kg m ²
Hub CM coordinates in shaft CS	(0 m, 0 m, -5.0191 m)
Nacelle CM coordinates in nacelle CS	(1.75 m, 0 m, 1.9 m)
Tower height	77.6 m
Distance from nacelle base to rotor axis	2 m
Distance from rotor axis to tower base	87.6 m

Table 2.1.1: Specifications of the NREL5MW wind turbine [21].

For what concerns the aerodynamics of the blades, it is provided by NREL. One rotor blade is discretized into 19 radial substations, each of which is modeled by a certain type of airfoil. In Figure 2.1.2, a typical airfoil is depicted. Figure 2.1.3, instead, shows one whole blade and all its characterizing airfoils.

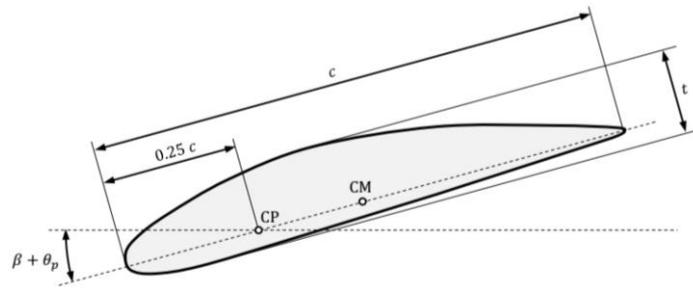


Figure 2.1.2: Generic airfoil cross-section [19].

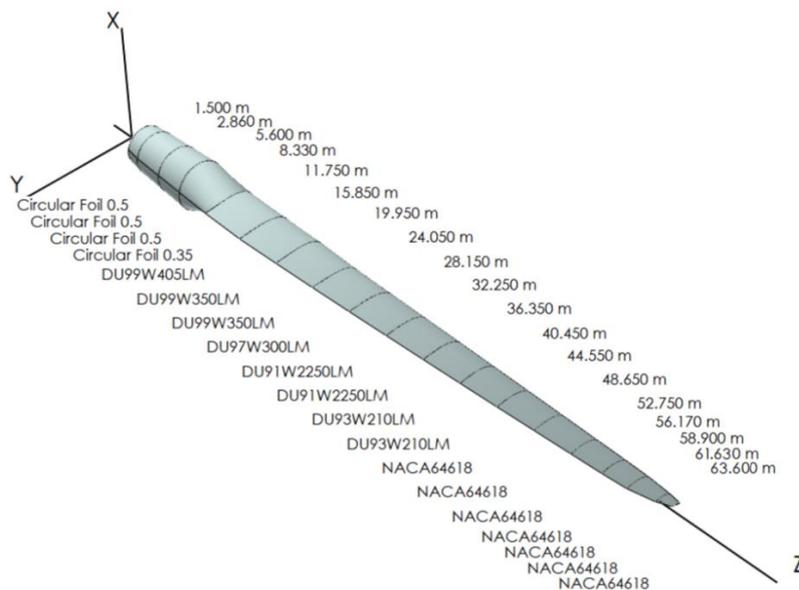


Figure 2.1.3: NREL5MW rotor blade (together with airfoils and respective coordinates) [19].

With reference to Figure 2.1.1, it is possible to highlight the geometrical parameters that define the shape of an airfoil cross-section. Typically, the pitch axis intersects the airfoil in the point CP, belonging to the chord c and spaced from the leading edge of $0.25 \cdot c$. The angles θ_p and β are respectively the active pitch and the twist, and the latter measures the misalignment between c and the rotor plane, when θ_p is null. Finally, t is the thickness, defined as the maximum encumbrance normal to c .

Moreover, some critical quantities are not depicted in Table 2.1.1. The distance between one substation and the rotor center is defined as z_{airf} , while Δz_{airf} is the blade span covered by each substation, spanwise. Eventually, $EdgeCMOff$ is the edgewise offset of the center of mass with respect to the leading edge. The geometrical features described so far are listed in Table 2.1.3. On the other hand, Table 2.1.2 presents the resulting undistributed properties for the blade, modeled neglecting the internal mechanical properties such as stiffness, damping and non-rotational inertia.

Length	61.5 m
Overall mass	17.740 t
Moment of inertia (w.r.t. root)	11,776,047 kg m ²
CM location	20.475 m

Table 2.1.2: Undistributed blade aerodynamics and structural properties [19].

Airfoil N.	z_{airf} [m]	Δz_{airf} [m]	Mass [kg/m]	Twist [°]	Chord [m]	Relative thickness [%]	Aerodynamic table	EdgeCMOff [mm]
1	1.50	2.73	802.6	13.31	3.54	100	Cylinder1	18.6
2	2.87	2.73	802.6	13.31	3.54	100	Cylinder1	18.6
3	5.60	2.73	634.8	13.31	3.85	100	Cylinder1	-57.4
4	8.33	2.73	427.9	13.31	4.17	100	Cylinder2	-81.4
5	11.75	4.10	445.2	13.31	4.56	41	DU40	-161.5
6	15.85	4.10	368.3	11.48	4.65	35	DU35	-264.3
7	19.95	4.10	353.5	10.16	4.46	35	DU35	-228.1
8	24.05	4.10	335.1	9.01	4.25	30	DU30	-198.3
9	28.15	4.10	306.3	7.80	4.01	25	DU25	-150.9
10	32.25	4.10	272.4	6.54	3.75	25	DU25	-132.7
11	36.35	4.10	245.5	5.36	3.50	21	DU21	-140.1
12	40.45	4.10	201.2	4.19	3.26	21	DU21	-159.9
13	44.55	4.10	167.8	3.13	3.01	18	NACA64	-253.9
14	48.65	4.10	140.6	2.32	2.76	18	NACA64	-226.7
15	52.75	4.10	107.5	1.53	2.52	18	NACA64	-211.6
16	56.17	2.73	90.8	0.86	2.31	18	NACA64	-229.4
17	58.90	2.73	70.8	0.37	2.09	18	NACA64	-153.6
18	61.63	2.73	48.4	0.11	1.42	18	NACA64	-72.3
19	63.00	2.73	48.4	0.11	1.42	18	NACA64	-72.3

Table 2.1.3: Geometrical quantities of the 19 radial substations of each rotor blade [19].

Rated generator speed	1173.7 rpm
Gearbox ratio	97
Generator efficiency	94.4%
Generator inertia about high speed shaft	534.116 kg m ²

Table 2.1.4: Drivetrain and generator properties [19].

The last part of the wind turbine to be specified is the drivetrain and generator. From a simulation perspective, the drivetrain is simplified and considered as an undamped and frictionless gearbox which transfers torque from the rotor to the generator (respectively the low-speed shaft and the high-speed shaft). The motion transmission is characterized by a gearbox ratio of 97:1. The other relevant properties are listed in Table 2.1.4.

2.1.2 The floating platform

In previous studies, a deep analysis of all the floating substructures present in the market has been carried out [19]. Initially, the research focused on two concepts: spar-buoy and hexafloat (by Saipem), which are the two guaranteeing the best stability to cost ratio (Figure 2.1.4).

Then, through a suitable genetic algorithm, the two substructures have been optimized to obtain the greatest possible stability at the lowest cost. The optimized substructures have been eventually compared both from a performance and an investment standpoint, and the final choice fell on the hexafloat (Table 2.1.5). Indeed, it is characterized by a slightly higher capacity factor (45.5% versus 45.2%) and it more convenient also in terms of LCOE (121.70 €/MWh versus 136.82 €/MWh). This LCOE projections have been derived from the specific case-study of Pantelleria Island, and in particular relatively to the site A defined in Chapter one of this dissertation.

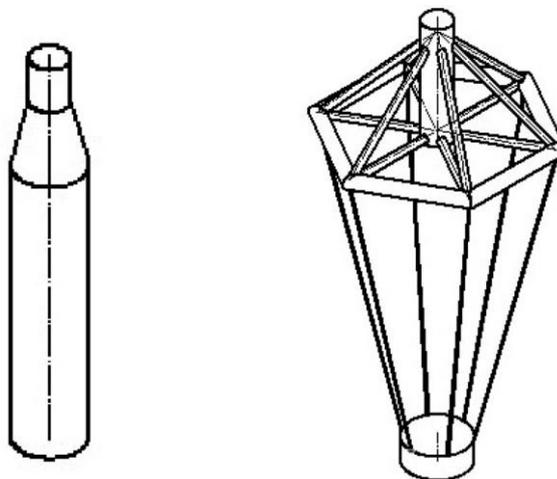


Figure 2.1.4: Spar-buoy (left) versus hexafloat (right) [19].

Central column diameter	8.38 m
Central column height	35.25 m
Side length	30 m
Ballast z_{CoG}	-99.45
Ballast diameter	10 m
Ballast mass	3304 t
Density magnetite	5200 kg/m ³

Table 2.1.5: Floating substructure properties (hexafloat) [19].

2.1.3 The mooring subsystem

The last main part of the system is the mooring subsystem, which in floating offshore WT is intended to prevent the structure from drifting away due to the forces of waves, currents and wind. Moreover, it is designed to maximize the rotational stability of the whole turbine.

In this thesis work, the choice fell on a catenary mooring system, composed by a set of six lines. These mooring lines develop radially with respect to hexagonal floating structure and are uniformly distributed in the water plane, spanning all the 360° and thus keeping the structure as still as possible. The lines, from a structural point of view, consist in metal chains, which implies a good stiffness to cost ratio.

All the properties of the mooring system that has been taken into account for the simulation are listed in Table 2.1.6.

number of mooring lines	6
angle between adjacent lines	60°
seabed depth	150 m
mooring leg composition	chain
unstretched lengths	473.3 m
type	chainup
diameter	80.9 mm
mass per unit length	138.7 kg/m
min. breaking load	3380 kN
axial stiffness	1071 MN
cable/seabed friction coefficient	1

Table 2.1.6: Mooring subsystem properties [19].

2.1.4 The overall system and the reference frame

Now that the structure of the turbine is completely described, the system reference frame has to be defined. The chosen frame is a right-handed, fixed reference axes (FRA) frame $Oxyz$ (Figure 2.1.5). This reference frame is inertial, because it does not move in space. The origin O is located in the point in which the mean water plane intersects the vertical axis of the turbine; the z axis points upwards, whereas the x axis points backwards and downwind.

A second reference frame is defined in order to describe univocally the motion of the system with respect to the FRA $Oxyz$. Such second reference is the local structure axes (LSA) frame $Gxyz$ and it is not inertial, since it moves in space. Its origin G is coincident with O .

The definition of the axes is critical because the moments of inertia, that affect the rotations of the system in the three directions, are defined with respect to those axes. When referring to rotations, it is worth noting that the x , y and z axes are usually named roll, pitch and yaw respectively.

Table 2.1.7 presents the last structural properties of the wind turbine to be listed: the moments of inertia with respect to the three directions and the undistributed aerodynamic properties of the blade.

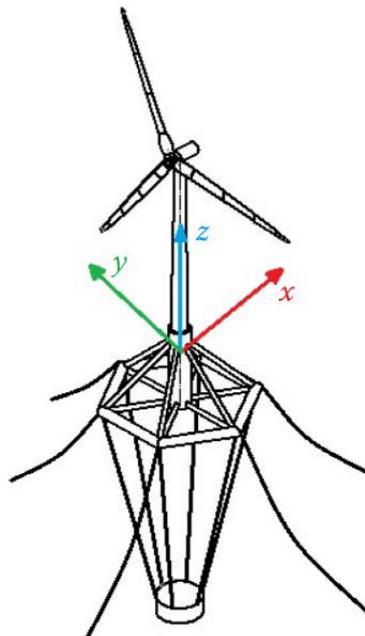


Figure 2.1.5: Fixed reference frame (FRA) [19].

Mass	4,605 t
Displacement	5,250 m ³
Draft	12 m
Center of gravity from SWL	(0, 0, -55) m
Roll moment of inertia	34,803,381,981 kg m ²
Pitch moment of inertia	1,579,004,565 kg m ²
Yaw moment of inertia	34,818,228,666 kg m ²

Table 2.1.7: Undistributed blade aerodynamic properties and moments of inertia [19].

2.2 The Simulink representation

The physical system described so far has been then modeled and represented in Simulink; in this paragraph the whole model is shown, along with a description of the structure and function of each of the main blocks. Also, a glimpse of the theoretical background behind the modelling process is given.

2.2.1 Overview

At the top level (Figure 2.2.1), the Simulink model presents three blocks, each one of them implementing the physical behavior of the three main parts of the system as described in Paragraph 2.1: wind turbine, mooring lines and hull (floater). The inputs feeding the systems are two data structures representing the evolution of wind and waves over a certain time horizon. All the relevant data needed for the simulation are contained in a workspace, that is generated by a Matlab main script along with a set of secondary scripts, workspaces, functions, libraries etc.

In the system, a pose vector is used to describe univocally the position, velocity and acceleration of the wind turbine hull. The degrees of freedom (DOFs) selected for the representation are 6 (thus leading to a pose vector with dimensions 18x1): surge (x), sway (y), heave (z), roll (rx), pitch (ry) and yaw (rz) respectively.

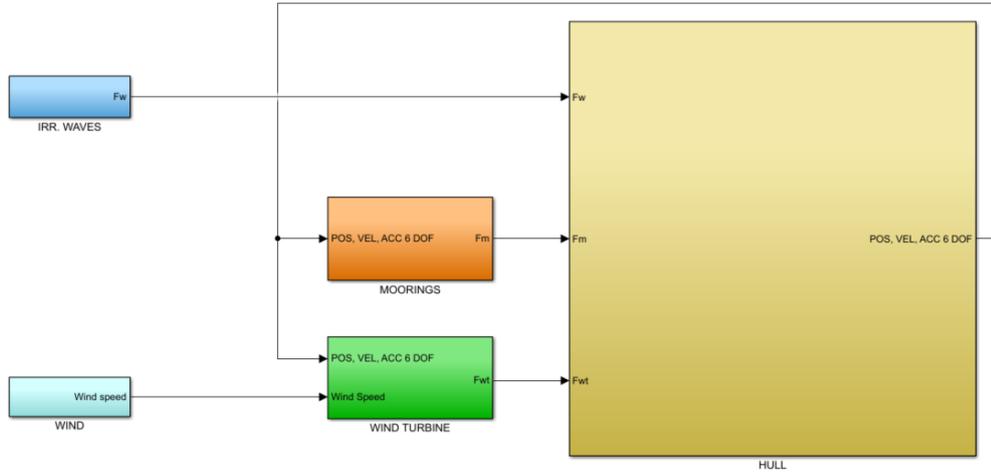


Figure 2.2.1: Simulink model of the system.

2.2.2 Wave input

First of all, the sources of excitation of the floating wind turbine are modeled: namely wind and waves. To do so, a deep research about the resources available in the Pantelleria case study site has been performed.

For what regards the waves, they exert on the hub an external force inducing a certain motion and limiting the stability properties. Waves can be modeled, in general terms, in three different fashions of increasing complexity and accuracy:

- Linear, monochromatic regular waves;
- Stokes waves (linearized to the second order);
- Irregular waves, the one considered in this work.

In the irregular waves (Figure 2.2.2), through the superposition principle, a complex wave profile is modeled as the combination of a theoretically infinite set of linear regular wave components (i.e. the spectrum). Each component is characterized by different heights, frequencies, wavelengths, and phases. The obtained wave η is represented by means of the following formula, which is valid for one particular direction x :

$$\eta(x, t) = \sum_N^{n=1} a_n \sin(\omega_n t - k_n x + \varphi_n)$$

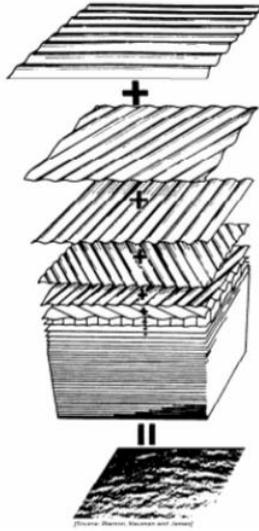


Figure 2.2.2: Graphical representation of irregular waves as superposition of different waves [22].

In the formula, the subscript n indicates the n^{th} component of the wave spectrum. Therefore, a_n is the amplitude, k_n the wave number, φ_n the phase and ω_n the frequency of such particular element of the wave summation. Note that the sum is finite (up to the N^{th} component) because in practice the specific contribution of each monochromatic wave decreases for higher frequencies.

Modeling irregular waves as described so far is a complex task from a computational point of view, and usually it is easier to obtain statistical data from a sample time history of the wave behavior. Therefore, the study of irregular waves becomes a statistical analysis, based on some significant data as average wave period, mean wave height and so on [22]. Schematically, the irregular wave history $\zeta(t)$ can be associated to a certain wave energy spectrum $S_\zeta(\omega)$, that is the statistical function that associates a certain energy to the wave, for each frequency.

Then, from the wave spectrum it is possible to retrieve a deterministic time history of the wave, without losing the information about the statistical properties represented by the energy density. This step is referred to as the inverse problem in wave statistics, and it leads to a reconstruction of the wave history that is constituted by randomly chosen phases φ_n . This means that the instantaneous values of the reconstructed wave are likely different from the original ones but are equivalent from a statistical (and therefore energetical) point of view.

From a programming standpoint, the reasoning described so far leads to a set of Matlab scripts (whose further analysis is outside the scope of this thesis) that allow the generation of the time history of the irregular waves in 6DOFs, and the resulting forces acting on the system. An example of the obtained waveforms is depicted in Figure 2.2.3. The considered time horizon for the simulations, in this work, is 300 seconds.

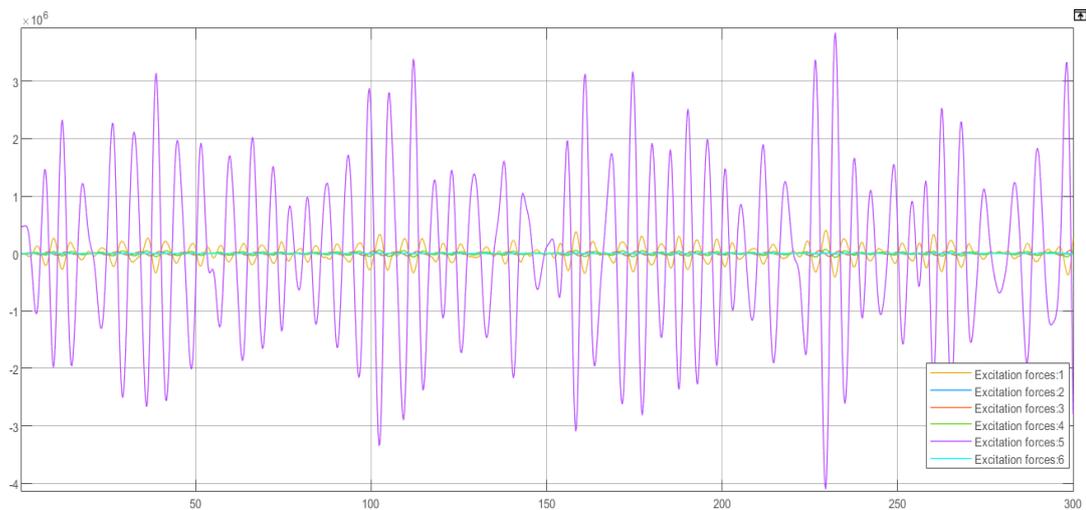


Figure 2.2.3: Example of the wave input of the Simulink model of the NREL5MW.

2.2.3 Wind input

The second input of the system is the wind resource; from a mathematical standpoint, the amount of energy carried by the wind is proportional to the cube of its speed. As already highlighted in Paragraph 1.2, in comparison with the other energy sources the wind is characterized by a higher variability, with respect of time and space. This is due to both macroscopic (e.g. geographic features, daily evolution) and microscopic causes (local and/or instantaneous gradients and variations).

In particular, a critical phenomenon that greatly affects the turbine effectiveness is the turbulence, that happens when the wind speed undergoes variations that are short term from a time point of view, but concurrently very high in amplitude. As written in the following of this paragraph, turbulence can be modeled by means of suitable software.

The wind is modeled in a three-dimensional, spatial-discretized fashion. This means that such input is defined, in Simulink, as a sequence of matrices, in which each matrix (13x13 elements) describes the values of the instantaneous wind speeds in a certain set of point, which belongs to a vertical 2D plane of the turbine environment (Figure 2.2.4), with dimensions $145 \times 145 \text{m}^2$. In this framework, the number of matrices that are part of the sequence corresponds to the number of timing instants taken into account, while each matrix represents a certain spatial discretization.

Afterwards, the *Turbsim* software [23] has been employed to simulate a complex turbulent environment to be added to a pre-defined 3D wind spectrum structured as expressed above.

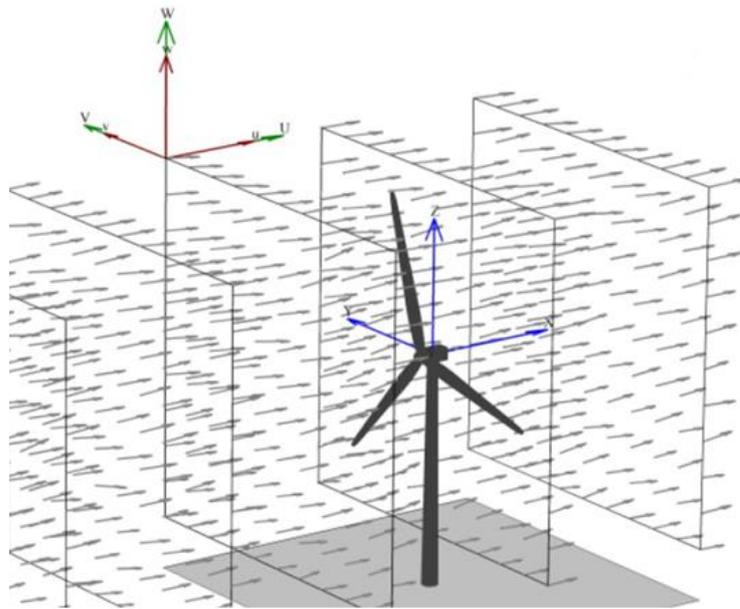


Figure 2.2.4: Graphical representation of the three-dimensional wind profile. [23]

2.2.4 Definition of the simulation campaign

Eventually, the *ERA5* database of ECMWF (“European Centre for Medium-Range Weather Forecasts”) [24] has been exploited to extract the hourly marine data in the location of interest discussed in Chapter one.

By joining the obtained data, one could form a set of triads composed by wave height (H_s), wave period (T_p) and wind speed V_0 (Table 2.2.1). Afterwards, a probabilistic study has been performed to sort the triads in increasing order of

probability, and to discard those with lower probability. From a user standpoint, it is possible to select the triad to be considered as simulation environment directly in the Matlab main script, simply changing an identification index.

H_s (m)	T_p (s)	V_0 (m/s)	
0.25	0.50	0.50	14.50
0.75	1.50	1.50	15.50
1.25	2.50	2.50	16.50
1.75	3.50	3.50	17.50
2.25	4.50	4.50	18.50
2.75	5.50	5.50	19.50
3.25	6.50	6.50	20.50
3.75	7.50	7.50	21.50
4.25	8.50	8.50	22.50
4.75	9.50	9.50	23.50
5.25	10.50	10.50	24.50
5.75		11.50	25.50
		12.50	26.50
		13.50	

Table 2.2.1: Dataset of the triads (wave height and period, wind speed) characterizing the simulations.

2.2.5 Wind turbine

A wind turbine is, in a nutshell, the electromechanical system intended to obtain electrical power through the conversion of the kinetic energy given by the wind. In the following, the structure of the sub-model (Figure 2.2.5) is presented, together with an overview about the modeling process.

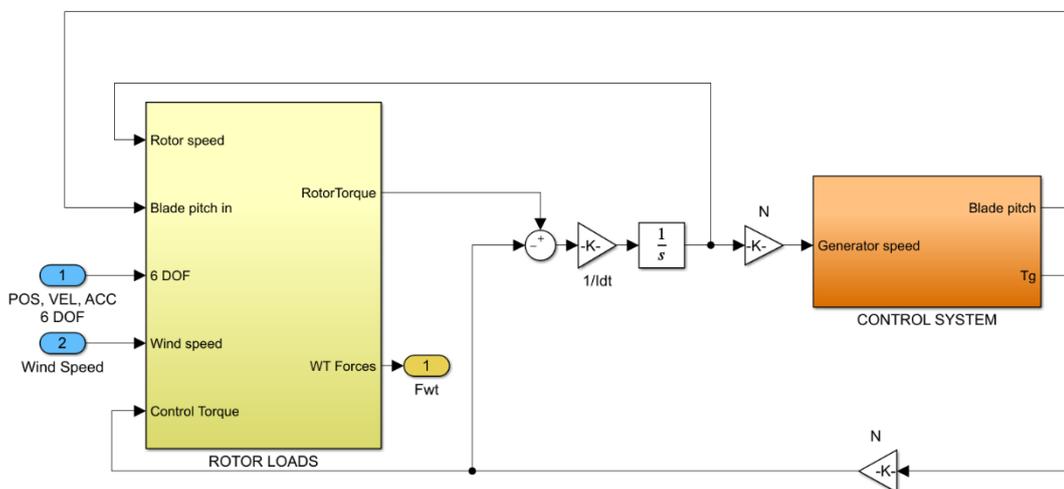


Figure 2.2.5: Wind turbine Simulink sub-model.

- **ROTOR LOADS block**

From a top-level point of view, the wind turbine receives as input the 6DOFs pose vector and the wind speed time sequence, and it gives as output a set of forces (WT forces) acting on the hull as tower base loads. The ROTOR LOADS block is the part intended for the computation of WT forces and the rotor torque. Such computation relies on the steady BEM theory (Blade Element Momentum), a methodology for calculating speeds and loads for any set of data (rotor speed, wind speed, pitch angle and position of the turbine) [25].

BEM is based, from a theoretical standpoint, on two concurrent assumptions. First, each blade is considered as a set of two-dimensional airfoils, whose behavior is influenced only by local events. This is referred to as *blade element model*. In other words, the blades are divided in sections that are independent from each other.

The second assumption on which BEM is based is the *momentum theory*. Here, the rotor is considered as an actuator disc, frictionless, and it is considered to subtract the kinetic energy from the wind, and therefore the stream speed decreases steadily. On the other hand, the flow is stationary, incompressible, and frictionless.

The BEM theory is affected by some limitations, and for this reason in the Simulink model some corrections are implemented. Summarizing, three corrections have been considered, and they are described in the following.

1. Prandtl's tip-loss factor: the BEM method, basically, overestimates the blade performances near the two extremities of the blade (root and tip). With this correction, a compensation for the tip loss is taken into account [25]. The root loss is usually neglected because it does not lead to a remarkable error.

2. Glauert and Buhl correction: it is implemented to manage a case in which BEM ceases to be valid. In detail, large deflections of the blade out of the rotor plane lead to turbulence of the wake, and therefore the thrust on the rotor increases [26].

3. Skewed wake correction: BEM neglects the deflections of the wind coming from the direction of the rotor axis [27]. In other terms, the BEM model is designed for an asymmetrical case. This correction takes into consideration

such deviations, whose effect is to increase the induced velocity and decrease the forces as the blade gets deeper into the wake.

- **CONTROL SYSTEM block**

Going back to the Simulink scheme of Figure 2.2.5, to the rotor torque obtained from the ROTOR LOADS block is subtracted the generator torque, and the result is integrated obtaining the rotor speed. Then, the result is multiplied for a certain gearbox ratio N and by doing so the generator speed is calculated, which is the input of the second block (CONTROL SYSTEM).

The implemented control system is actually composed by two different control systems. They are independent of one another from a functioning point of view, but they work in a mutual exclusive manner, based on the current operating point. Indeed, when below the rated point, a generator-torque controller is utilized, so that the power extraction is enhanced; on the contrary, when above such point, a full-span rotor-collective blade-pitch controller is activated.

rated wind speed	11.4	m/s
rated rotor speed	12.1	rpm
rated generator speed	1173.7	rpm
rated generator torque	43094	Nm
rated mechanical power	5.30	MW
rated electric power	5.00	MW

Table 2.2.2: Rated operating point of the NREL5MW wind turbine.

It is worth underlining that the rated (or nominal) point is defined as the operative point taken as reference for the maximum continuous power conversion [20]. In general, it is possible to state that the control system aims to reach the rated point as a sort of equilibrium. In Table 2.2.2, the values characterizing the nominal point are depicted. In the following, the two control algorithms are described.

In the *generator-torque control law*, the generator torque is computed through a look-up table representing a function of the speed. It presents three main regions:

1. In region 1 the torque is null; therefore, no power is generated. The energy given by the wind accelerates the rotor for the start-up. The cut-in wind speed is 670 rpm, above this value, the transition occurs.

2. In region 2 the torque is computed as the square of the generator speed times a suitable coefficient k_{opt} . This leads to an optimal constant tip-speed ratio, which maximizes the converted power. Region 2 is located between 871 rpm and 1162 rpm.

3. In region 3 the generator torque and speed are inversely proportional, thus generating a constant power. This behavior is triggered for speeds greater or equal to 1173 rpm.

Between regions 1 and 2, and regions 2 and 3 respectively, there are two linear transition regions. They are needed to limit the operating speed range of the wind turbine (in the first case) or the noise emission at rated power (in the second). The overall look-up table is graphed in Figure 2.2.6.

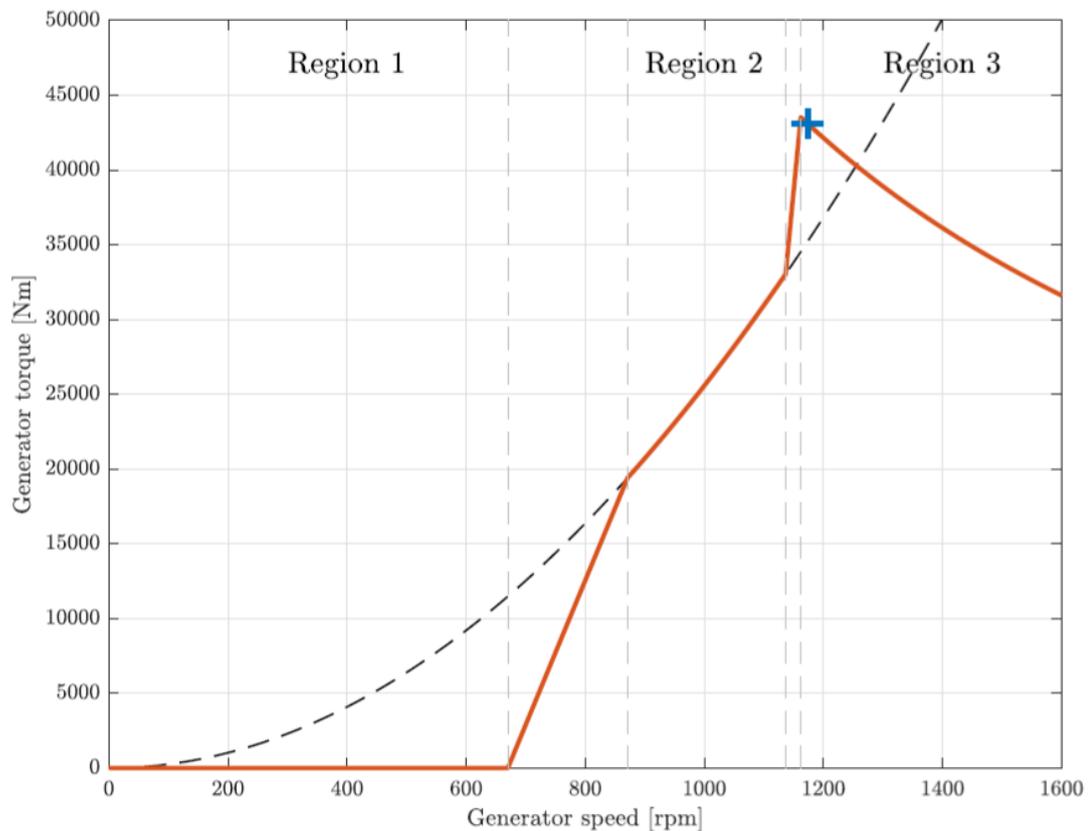


Figure 2.2.6: Generator-torque control law and main control regions [20].

It is important to notice that when in region 3, the generator speed is greater than the nominal speed, and as already anticipated this is the conditional statement that activates the full-span, rotor-collective, *blade-pitch controller*. In this case, a proportional-integral controller with variable gains is used in order to maintain the nominal point as operating point. Practically, the instantaneous generator speed is compared with the rated one continuously, and when the former is greater than the latter, the PI controller is triggered, and it modifies the blade-pitch eventually decreasing the generator speed.

2.2.6 Moorings

The modelling of the mooring system has been performed by means of the so-called *MAP++* theory [28]. *MAP++* is a mooring model developed by the US National Renewable Energy Laboratory (NREL) and the American Bureau of Shipping. It is a relatively simple multisegmented quasi-static model, which allows a robust evaluation of a mooring system.

Being quasi-static, the dynamic effects are neglected by this theory. Similarly, the loads of the mooring systems are assumed to be constant at each time step, and this leads to a uniform and linear motion of the platform between two static positions. Moreover, the model is multisegmented in the sense that each single mooring line is considered as constituted by a set of nodes and elements (the latter are the components connecting two adjacent nodes).

The *MAP++* theory is in practice implemented by a suitable program whose algorithm works as follows. The inputs of the program are the position of the platform (6DOFs) and the environmental features; from these, the program is able to solve two distinct sets of equation. The first set is composed by the catenary functions, depending on the configuration of the line and on the number of elements.

The second set, instead, is the one containing the force-balance equations for each node. From this, a triad of forces in the three directions is computed; afterwards, the moments produced on the structure by these forces are calculated and eventually all the moments of the six different mooring lines are summed.

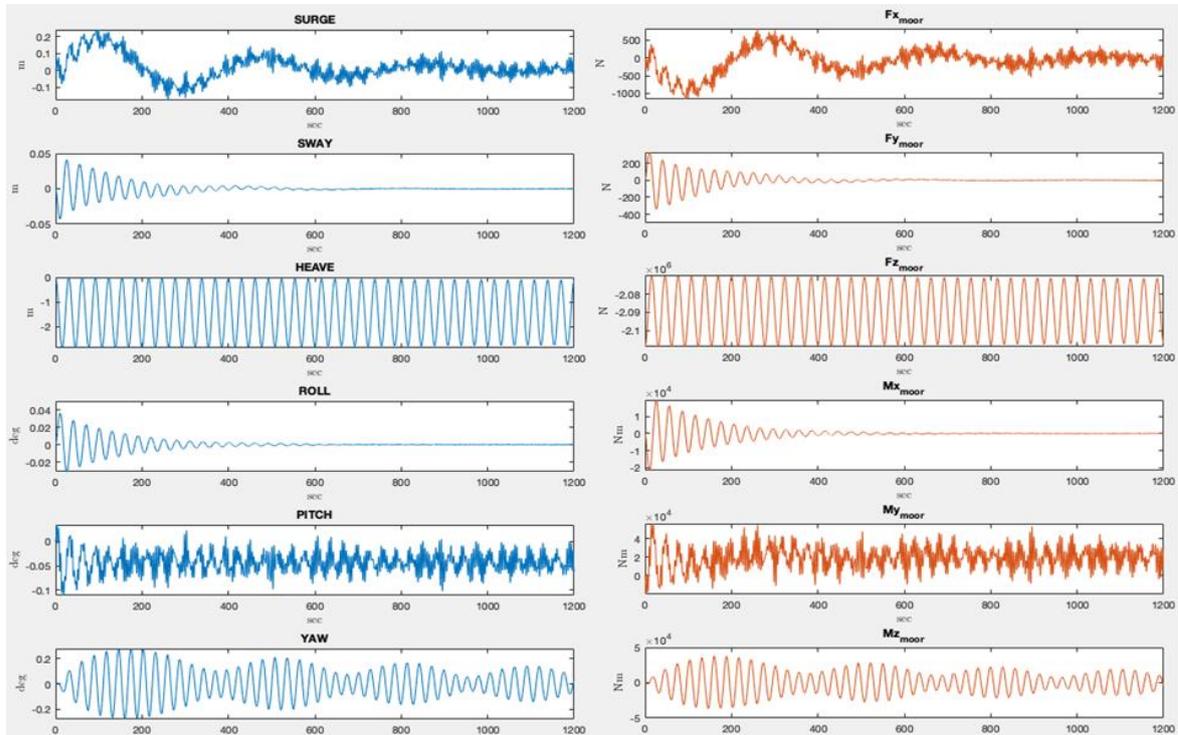


Figure 2.2.7: Example comparison between displacements and mooring reactions.

From an operational point of view, the mooring system is intended to allow the hull to maintain a stationary position regardless the external forces. In other terms, the forces and moments given by the mooring reaction have the task to counteract any movement of the platform. This can be visualized in Figure 2.2.7, in which the movements and the mooring forces and moments present a clearly specular trend.

2.2.7 Hull

The HULL block contains the model of the floating platform of the wind turbine, that has been already discussed previously in this Chapter. Obtaining a model of the hull is not a trivial task, because it involves complex physical phenomena regarding different kinds of hydrostatic loads affecting a floating body in a marine environment. In the following, a quick overview of the theoretical background of the modelling process is presented.

- **Hydrostatic restoring force**

This force is due to the well-known Archimedes' principle: *“any body, fully or partially submerged in still water, receives an upward buoyant load equal to the weight*

of the fluid displaced by the body and applied on the center of buoyancy” [19]. Such force is proportional to the water density, the submerged volume and the gravitational acceleration. It is possible to demonstrate that the hydrostatic restoring force can be expressed, in matricial form, as follows:

$$\mathbf{F}_{hrf} = -\mathbf{K} \mathbf{x}$$

In this equation, \mathbf{x} is the vector representing the displacements of the center of gravity of the platform (with respect to the fixed-axis reference frame), while \mathbf{K} is a 6x6 matrix representing the hydrostatic stiffness in 6DOFs. After some manipulation, the resulting matrix is diagonal:

$$\mathbf{K} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & K_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & K_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & K_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- **Froude-Krylov force**

It is the force contribution due to the unsteady pressure field which is generated by the incident waves. It is computed as a surface integral of the incoming wave pressure field times the surface normal unit vector.

- **Diffraction force**

This disturbing force contribution is due to the disturbance waves generated by the interaction of the floating platform and the incident wave. It is calculated as the surface integral of the unsteady pressure field due to the diffracted wave.

- **Radiation force**

This kind of force is due to radiated waves that are a consequence of the buoy motion. From an analytical point of view, in this work such contribution is computed in the form developed by Cummins [30].

- **Drag force**

This contribution considers the damping effects due to the viscosity of water, and for the presence of vortices that dissipate the energy of the flow.

The hydrodynamic analysis of the overall system is developed on the basis of *Boundary Element Method* (BEM). It is one of the most common theories used to model floating platforms. In this work, the ANSYS Aqwa software has been utilized [31]. It is a commercial software that runs numerically the BEM method, in order to obtain the physical quantities of interest regarding the system in analysis. To do so, some assumptions have been made (inviscid and incompressible fluid, irrotational flow, small steepness of the waves, motion with small amplitude).

The obtained hydrodynamic model has been then analyzed both in the frequency domain (to obtain the 6x6 inertia matrix) and in the time domain. Solving the model in the latter case could require a high computational effort. For this reason, FOAM been employed in this work. It is an algorithm based on finite-order moment matching [32] which approximates the problem by means of an LTI SSM (linear time-invariant state-space model) of linear ordinary differential equations.

2.2.8 Example simulation

In this last Paragraph, the Simulink model just described is tested and its results reported. The wind turbine system under examination is located in a marine environment characterized by the following triad of working conditions:

- wave height $H_s = 0.75\text{m}$;
- wave period $T_p = 5.5\text{s}$;
- wind speed $V_0 = 8.5\text{m/s}$.

It is useful to recall that, as expressed in Paragraph 2.2.4, such triad belongs to the dataset of environmental conditions which has been defined for the simulation campaign, and that the selected triad can be changed simply modifying a suitable index in the Matlab main script. In the following, the most relevant graphs are extrapolated from the Simulink model. The considered timing horizon, for this simulation, is 5 minutes (300s).

First, the input profiles can be defined. For what concerns the wave forces, they have already been depicted in Figure 2.2.3; on the other hand, in

Figure 2.2.8 the wind speed is represented. Here, the Y-axis is expressed in m/s, and it is possible to verify that the average value of the whole wind speed profile is 8.5m/s, which is consistent with the triad that has been selected for the simulation.

Such inputs act on the wind turbine system basically in two manners. First, on the floating platform are exerted some forces, whose result, together with the wave forces, is a platform displacement (Figure 2.2.9); they are counteracted by the mooring reaction (Figure 2.2.10), which conversely tends to keep the hull stationary.

The second consequence of the inputs on the turbine system is the rotation of the blades, that has been modeled through the BEM model. The resulting rotor angular speed (rpm) is depicted in Figure 2.2.11. In the same Figure it is also possible to notice that, since the rated rotor speed is not exceeded (12.1 rpm, see table 2.2.2), the blade-pitch controller is never activated during the whole simulation. Consequently, the braking system is not triggered to reduce the velocity, and in this scenario the exploited control law is always the generator-torque one, which has been already described in Figure 2.2.6.

The generator angular speed (rpm) can be obtained mathematically as the rotor speed times the gearbox ratio. Then, the generator torque (Nm) and eventually also the extracted electrical power (MW) are obtained. In Figure 2.2.12 the reader can visualize that the latter two have an identical behavior, apart from a contraction factor, which is given by the efficiency of the system (0.944). In these last graphs, it is possible to notice again that the nominal values are never exceeded. This implies that the rated torque of 5MW of the *NREL5MW* wind turbine is not reached during such time horizon. The maximum value of produced power is, in the simulated scenario, roughly 2.6MW. In a situation in which the nominal ratings are reached, then the blade-pitch controller would be activated almost instantly, decreasing speed, torque, and generated power, which would quickly reach an allowed value (i.e. slightly lower than 5MW).

After the complete analysis of the system of interest, both in its actual physical version and in its Simulink model, the next chapter is going to deepen the architecture of the test bench set-up, which is the actual focus of this thesis dissertation.

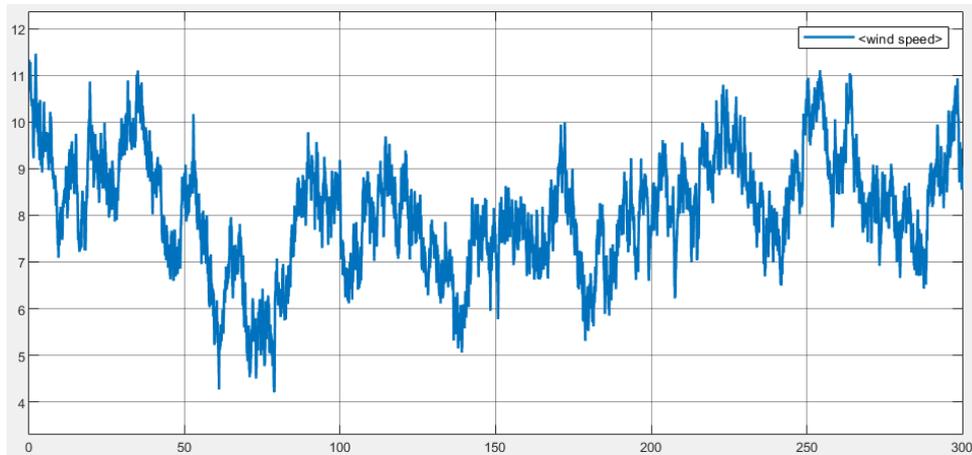


Figure 2.2.8: Example simulation. Input wind speed (m/s) versus time (s).

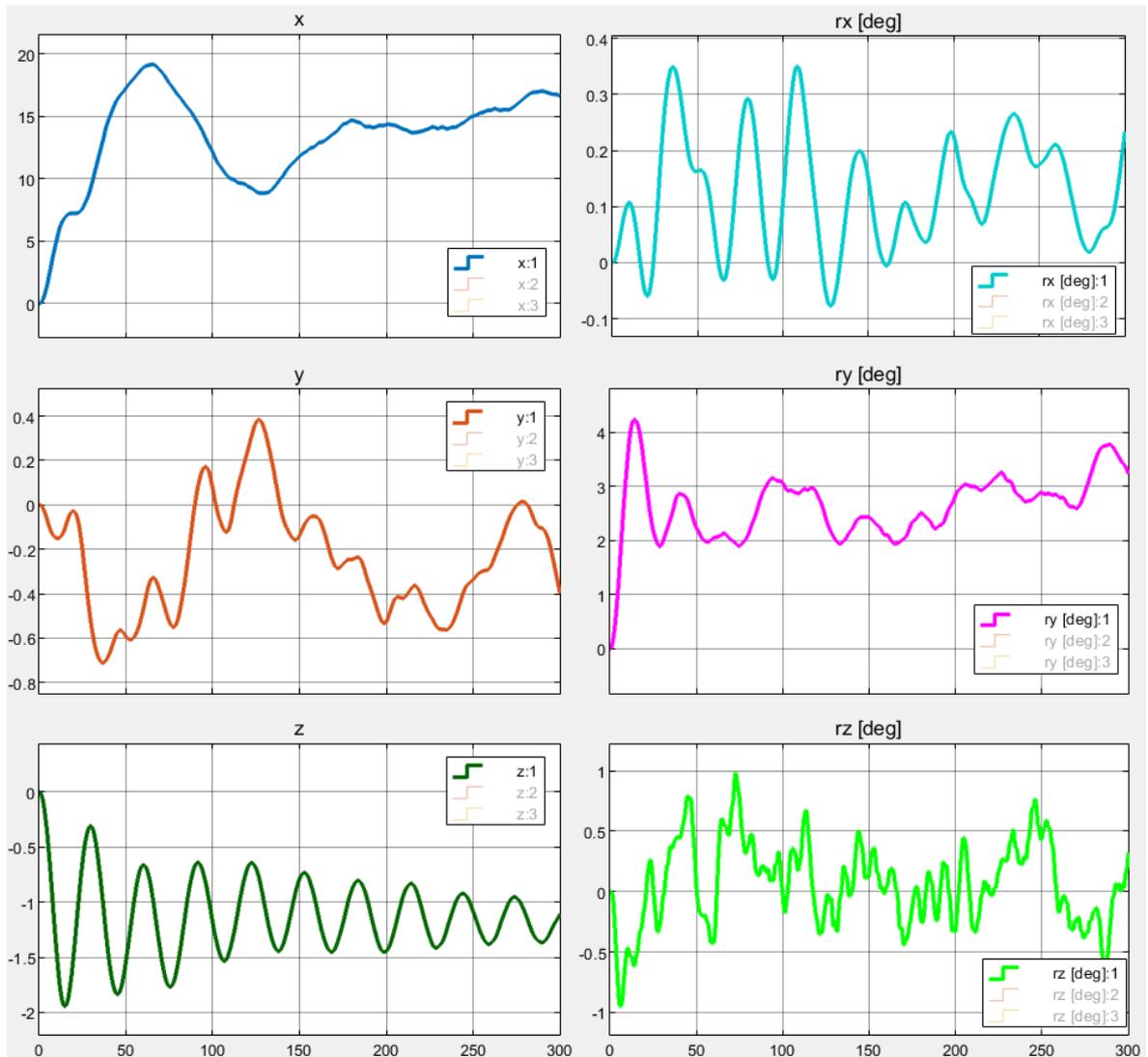


Figure 2.2.9: Example simulation. 6DOF displacements of the hull over time (s). The derivatives are not showed. The linear quantities (x, y, z) are expressed in meters, the angular ones (rx, ry, rz) in degrees.

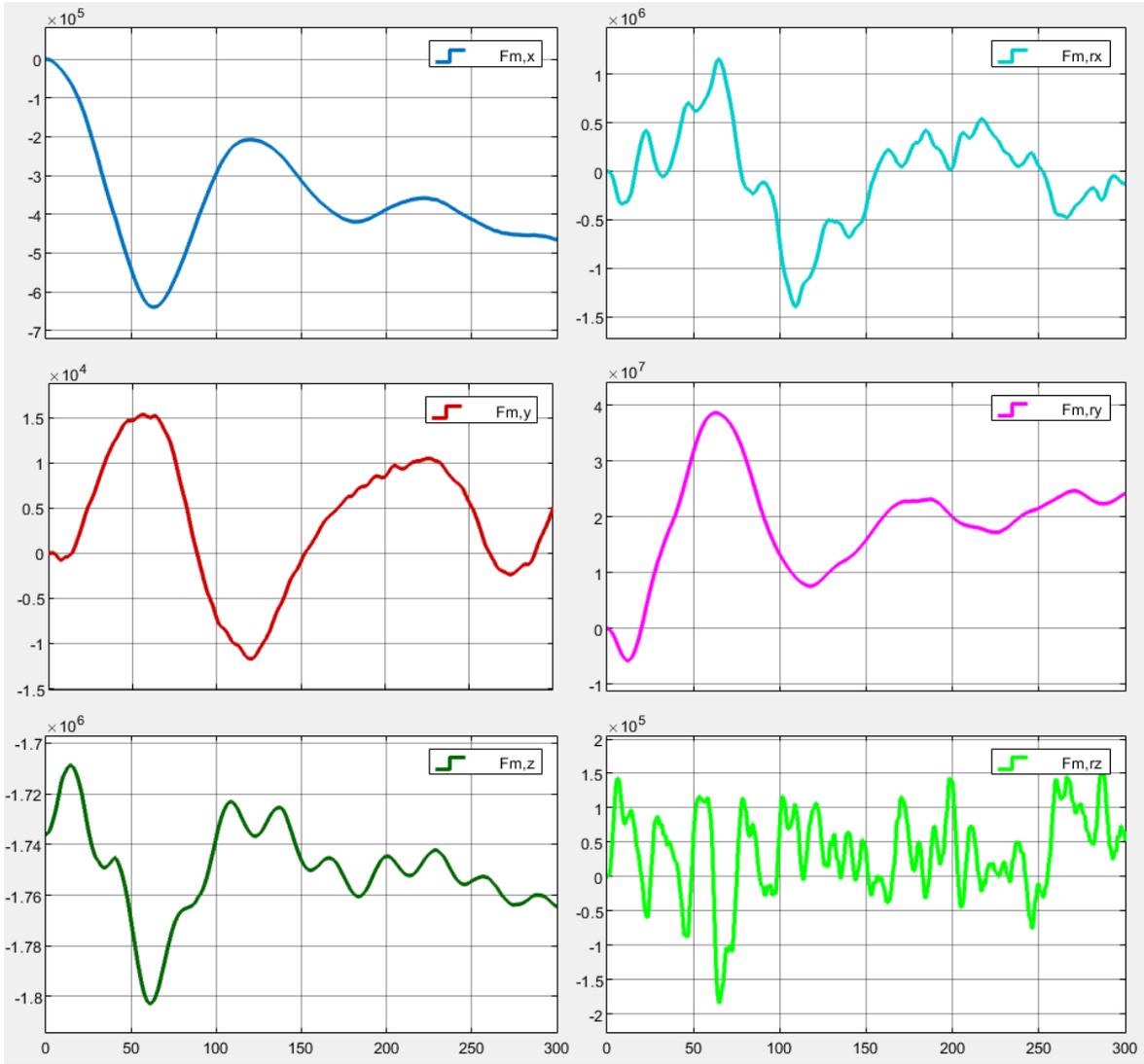


Figure 2.2.10: Example simulation. Mooring reaction forces (N and Nm) versus time (s).

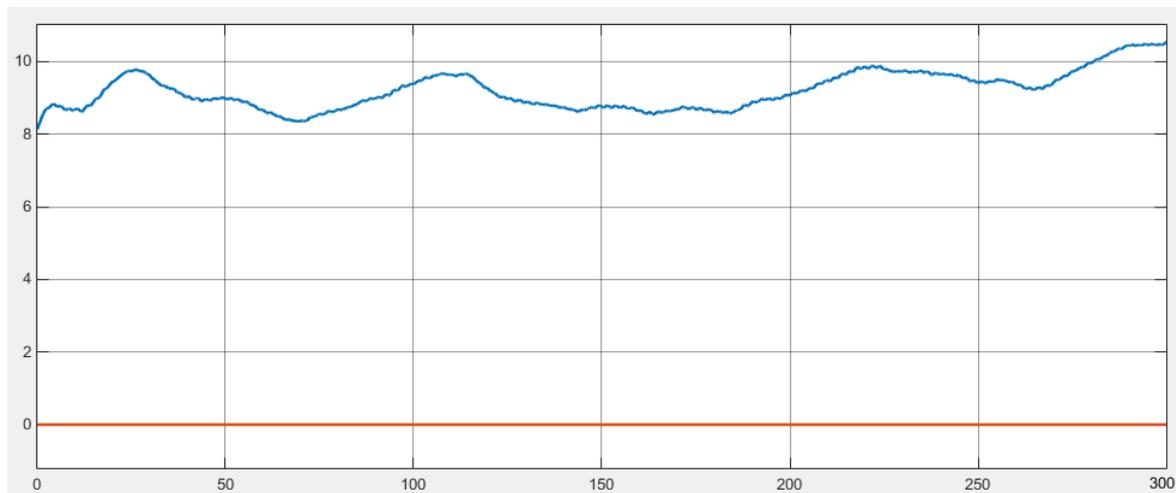


Figure 2.2.11: Example simulation. Rotor angular speed (rpm) and blade pitch control versus time (s).

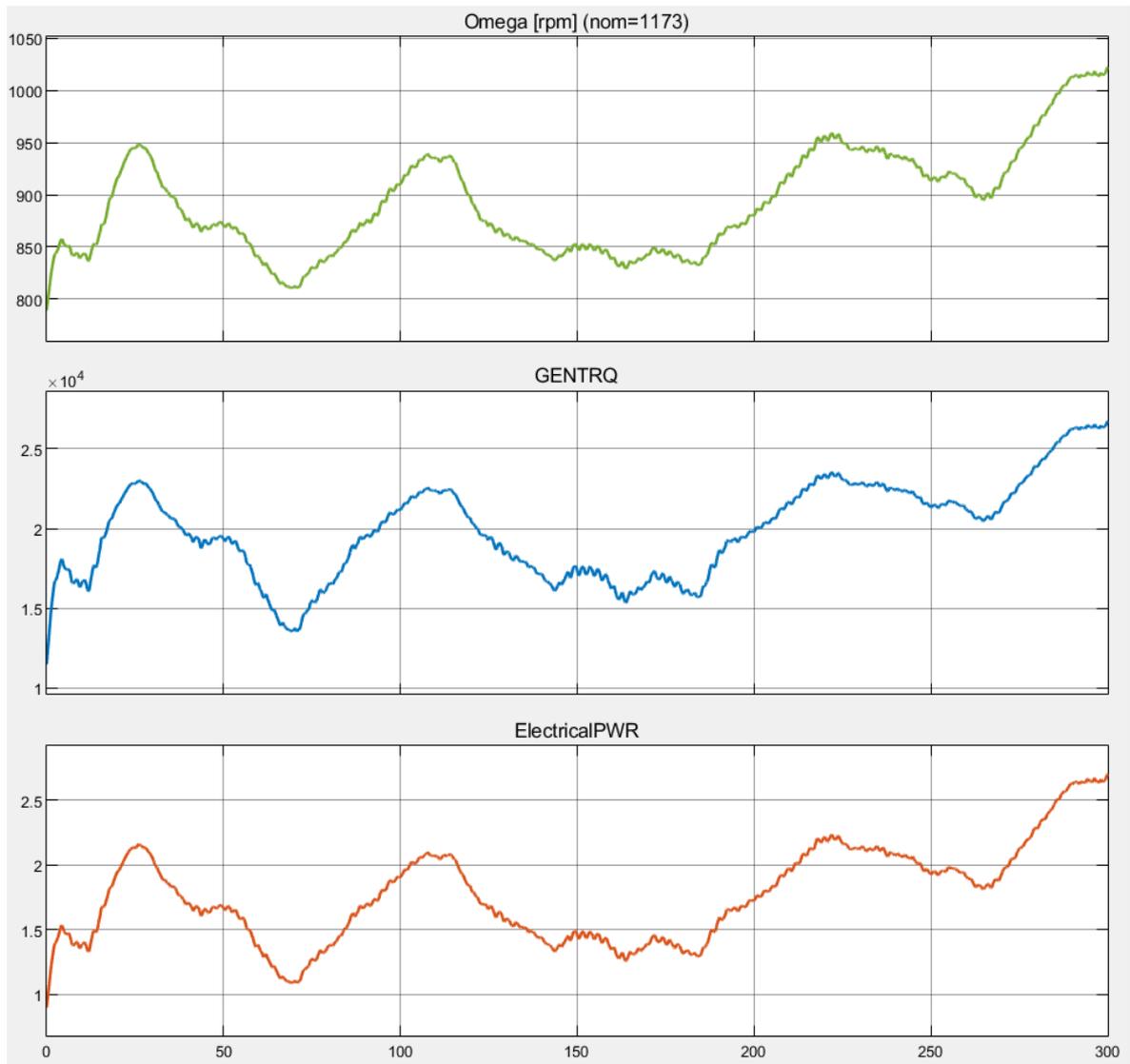


Figure 2.2.12: Example simulation. PTO (power take-off): generator speed (rpm), generator torque (Nm), and electrical generated power (MW) with respect to time (s).

Chapter three

The test bench setup

As stated in Chapter one, the purpose of this thesis is to develop and build a Hardware-In-The-Loop (HIL) test bench, aimed to verify the behavior of the floating offshore wind turbine described in Chapter two. In this Chapter, instead, the structure of the experimental test rig is depicted. First, HIL simulation is defined in a nutshell in Paragraph 3.1, along with the reasons that led to the choice of such testing approach.

Subsequently, the overall architecture of the experimental setup is presented and analyzed, in its entirety and then deepening each of the components. The hardware plant and the controller of the bench (together with the VeriStand software) are described respectively in Paragraph 3.2 and Paragraph 3.3. Then, Paragraph 3.4 focuses on the EtherCAT protocol, that has been implemented to allow the communication among the several test bench components.

3.1 Hardware-In-The-Loop testing

The testing and validating process is nowadays more critical than ever in the engineering sector. Indeed, the average complexity of the control systems used in most of the industrial applications (i.e. automotive, aerospace, automation, robotics and, of course, energy) is growing at an exponential rate, and this obviously increases the chances of possible system failures. In general terms, the latter such failures occur in the workflow and the worst the consequences, both in terms of risks and costs.

Concurrently, the time to market (defined as the time elapsed from the ideation of a product until the moment it becomes available on the market) is in

average shrinking. Therefore, the validation process must be as efficient and as fast as possible. For the reasons above, suitable and well-established strategies for simulating and testing are utilized in different stages of the design flow of embedded control systems. Hardware-In-The-Loop (HIL) is one of them. It is performed in a sort of virtual environment (representing the physical system to be controlled) to evaluate the performances of the control algorithm. This prevents the designer to create a proper physical prototype for all the needed tests to be performed, thus saving a tremendous amount of time, effort and money.

In detail, HIL has a couple of features that distinguish it from other fashions of testing techniques (such as Model-In-The-Loop and Software-In-The-Loop) [33]. First, it works in real time and it provides deterministic execution: in other terms, the control system (running on an electronic control unit, or ECU) must guarantee the delivery of the control input within a certain sampled period, that is set to be consistent with the real-application scenario. Hence, the computational burden that the controller has to manage is maximized, and this is crucial to avoid failures of both the control logic and the ECU itself. The second characteristic feature of HIL testing is the usage of real hardware in the simulation loop (Figure 3.1.1). Indeed, the plant to be controlled is completely substituted by one or several pieces of hardware, that are intended to feed back to the ECU a behavior that is once again compliant with the real application.

The implementation of a HIL test bench, although not trivial, has the significant advantage of creating a simulation environment that is able to substitute the physical application from a performance point of view, also guaranteeing a real-time behavior intended to be as accurate and faithful as possible to the actual world scenario. This, ultimately, allows to fulfil reliability and time-to-market requirements in a cost-effective, powerful manner.

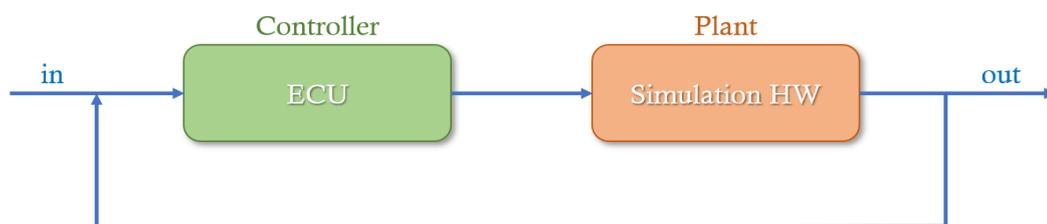


Figure 3.1.1: Schematic representation of the HIL feedback loop structure.

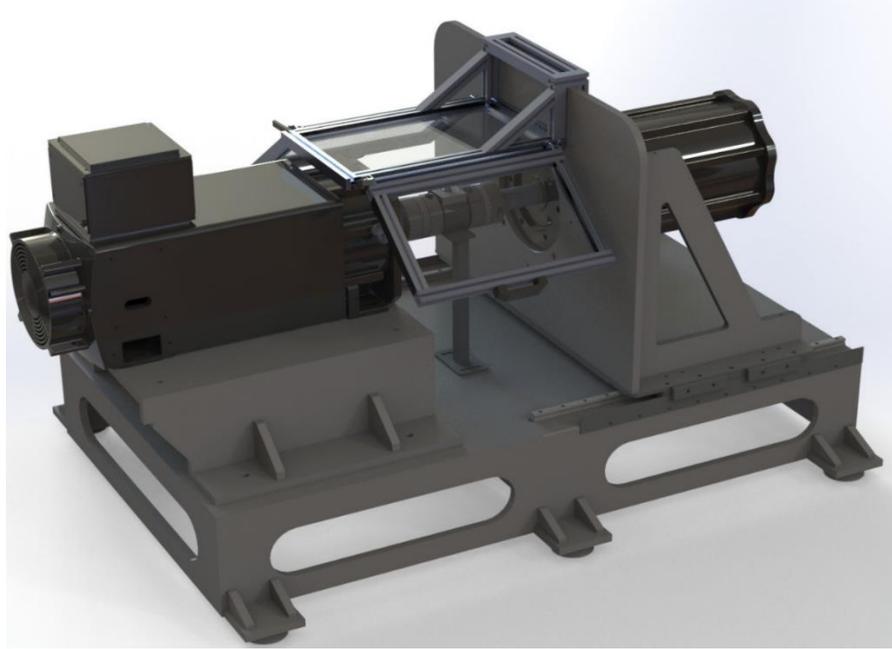


Figure 3.1.2: 3D rendering of the complete HIL test bench structure.

3.2 The plant hardware

As stated in the previous paragraph, one of the main peculiarities of the HIL testing approach is the usage of a suitable hardware in the control loop, intended for simulating the actual application plant in a manner as realistic as possible. This paragraph is aimed to outline the hardware used in the thesis work; to understand more clearly the behavior that has to be simulated, the paragraph starts with a quick overview of the electro-mechanical functioning principles of the wind turbines.

Despite the fact that in the industry there exist a huge variety of wind turbines, with various structures and exploiting several different control systems, they are all based on the same physical principle. In general terms:

- The wind acting on the blades triggers their rotation and the consequent rotation of the hub;
- Such movement is driven to a generator by means of a suitable transmission subsystem (gearboxes and shafts);
- The generator converts the mechanical input in an electrical output, thus producing a certain amount of electrical power that is then fed to the grid.

Therefore, it is possible to highlight three main mechanical parts constituting a turbine, that are rotor, generator and transmission subsystem (Figure 3.2.1). The test bench, from a structural point of view, must therefore match those three parts in order to consistently represent the system from a functional standpoint.

The test rig that has been developed is indeed composed by a motor (representing the rotor of the turbine), a shaft with a torque meter that gives as feedback the actual rotational speed, and a generator.

In the following, these three blocks are described thoroughly. Figure 3.2.2 illustrates schematically the structure of the HIL loop (already presented in Figure 3.1.1), taking into account explicitly the structure of the plant hardware.

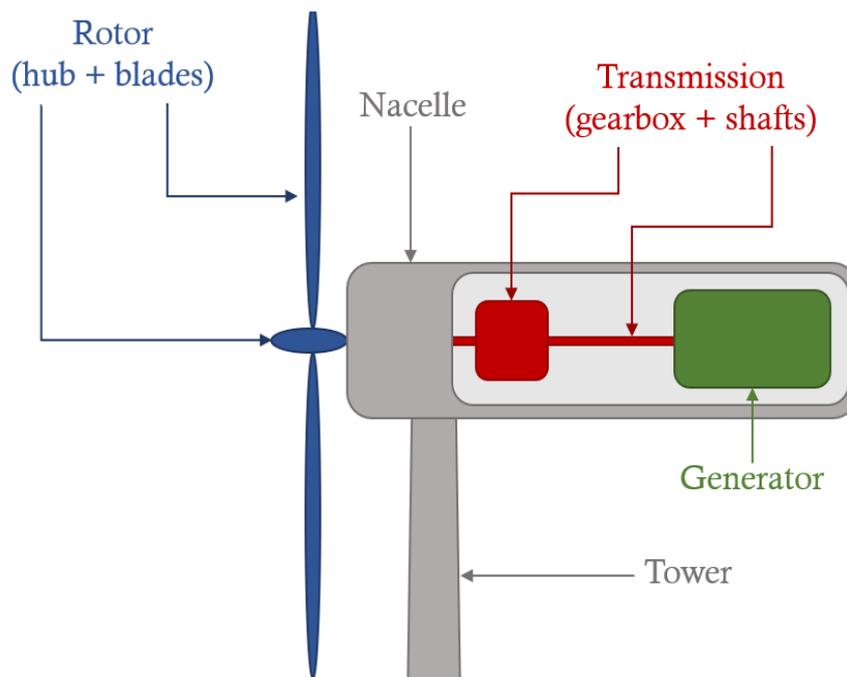


Figure 3.2.1: Scheme of the main mechanical parts of a turbine (rotor, transmission, generator).

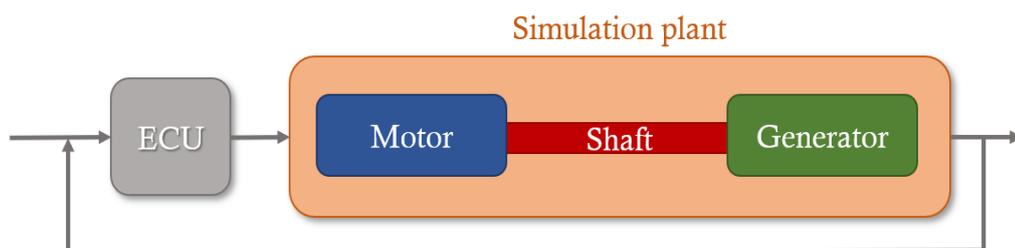


Figure 3.2.2: Three main components of the plant hardware, inserted in the HIL feedback loop.

3.2.1 The motor

The employed motor is a three-phase, four poles asynchronous motor, with squirrel cage rotor. The vendor is ABB, and the model is *AX 160L.4*. It is presented in Figure 3.2.4, where it is also possible to notice the white control cable connecting to the drive and the green cable which is, in turn, the encoder cable. In Table 3.2.1, the relevant data regarding the motor performances are listed, while Table 3.2.2 illustrates the data collected during the testing (performed by the company Comer s.r.l). Moreover, applying the following suitable equations, the behavior of the motor in terms of losses and efficiency is derived. Finally, plotting the data listed in Table 3.2.1 and taking into account the shape of the mechanical characteristic of the motor, which can be found in its datasheet, it is possible to obtain the graphical behavior of torque T and power P with respect to the angular velocity ω (Figure 3.2.3).

Rated speed ω_n	1740 rpm / 58.9 Hz
Limit speed ω_{lim}	5500 rpm / 184 Hz
Maximum speed for constant power ω_{max}	4200 rpm / 141.5 Hz
Rated torque T_n	329.9 Nm
Rated power P_n	60 kW
Rated voltage V_n	400 V
Rated current I_n	113.3 A
Power factor $\cos \varphi$	0.83

Table 3.2.1: Rated quantities and other relevant data of the motor ABB AX 160L.4.

$V[V]$	$I[A]$	$\omega [rpm]$	$f [Hz]$
400	47	1767	58.9

Table 3.2.2: Testing data of the motor, collected by Comer s.r.l.

- Absorbed electrical in rated conditions: $P_{n,e} = \sqrt{3} V_n I_n \cos \varphi = 65.15 \text{ kW}$
- Total losses in rated conditions: $P_{n,loss} = P_{n,e} - P_n = 5.15 \text{ kW}$
- Efficiency: $\eta = \frac{P_{n,e}}{P_n} = 0.921$

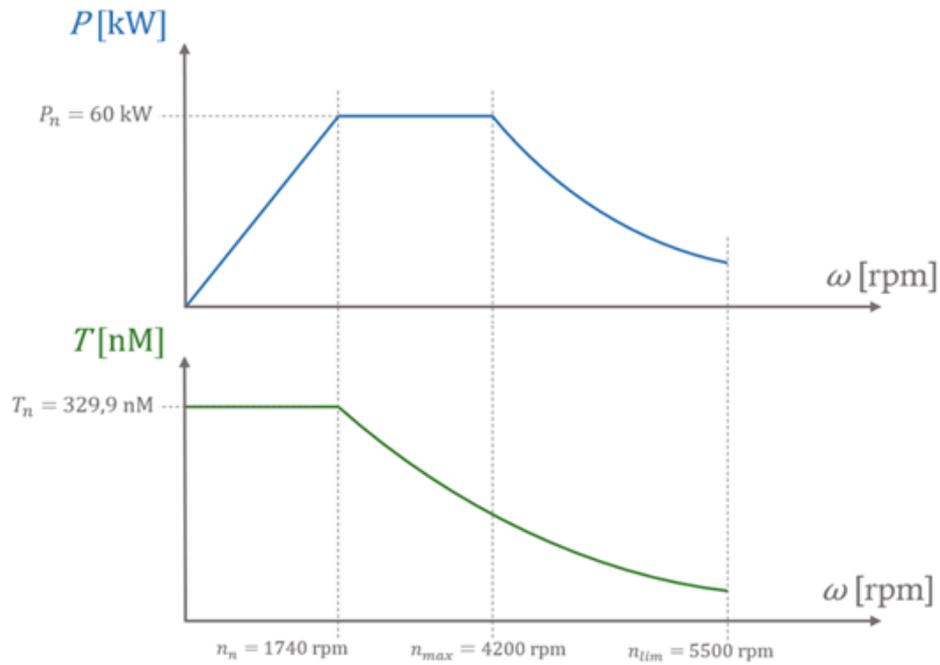


Figure 3.2.3: Mechanical characteristics of the motor ABB AX160L.4.

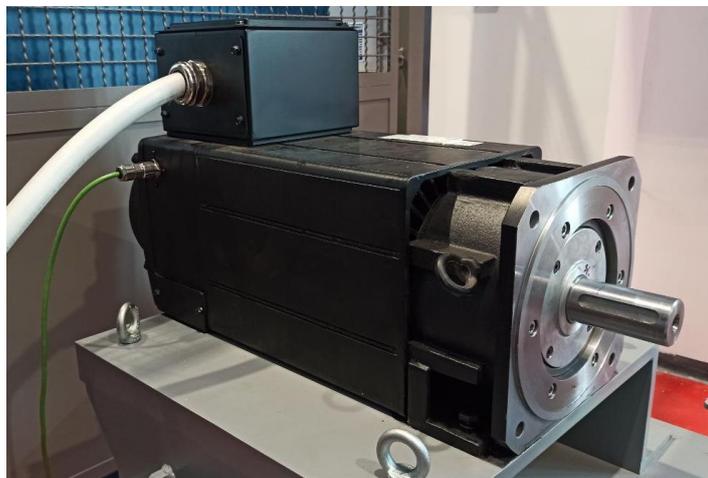


Figure 3.2.4: A picture of the motor ABB AX160L.4.

3.2.2 The drive

The motor has to be controlled by a suitable electrical drive. The one used in this thesis is an *ACS880-07-0145A-3* by vendor ABB (Figure 3.2.5). It is a six-pulse diode, three-phase drive that accepts as inputs an AC signal with frequency of 50/60 Hz and a voltage range from 380 V to 415 V. On the other hand, the output is a suitable AC signal, with a modulated frequency up to 500 Hz. In

Table 3.2.3, some of the main features of this component are listed (note that “light overload” and “heavy-duty” are defined as temporary working conditions in which the drive is working respectively at 110% and 150% of its maximum capability, for a period up to 5 mins).

Nominal ratings		Light overload		Heavy-duty	
P_n	75 kW	P_{Ld}	75 kW	P_{Hd}	55 kW
I_n	145 A	I_{Ld}	138 A	I_{Hd}	105 A
I_{max}	178 A				
V_n	400 V				

Table 3.2.3: Working ratings of the drive ABB ACS880-07-0145A-3.



Figure 3.2.5: A picture of the driver ABB ACS880-07-0145A-3.

In Figure 3.2.5 it is possible to notice at the base of the drive the presence of a support structure for the drive itself and an emergency pushbutton. They have been both specially designed for this application by the company Elatec S.a.s., together with the rest of the electrical cablings and the installation of the electrical panel (see Paragraph 3.3.4).

3.2.3 The transmission subsystem

The transmission subsystem is intended for transferring the motion generated by the motor to the generator. It is composed by a torque transducer, two shafts and two flanges which join the other parts together.

The utilized torque meter is a model by HBM called *1-T22/500NM* (Figure 3.2.6). It is the electromechanical component able to measure the torque exerted on the shafts, providing as output an electric signal that in this model can be generated either as a current (10 ± 8 mA) or as a voltage (± 5 V) [34]. In Table 3.2.4, the main mechanical features of such device are presented. It is possible to notice that the nominal torque T_n is greater than its motor homologue (329.9 Nm).

Rated torque T_n	500 Nm
Maximum torque T_{max}	1000 Nm
Breakup torque T_{lim}	1400 Nm
Rated rotational speed ω_n	9000 rpm
Limit longitudinal force $F_{long,max}$	4 kN
Limit lateral force $F_{lat,max}$	1.4 kN
Limit bending moment M_{max}	54 Nm

Table 3.2.4: Rated quantities and other relevant data of the transducer *1-T22/500NM* by HBM.

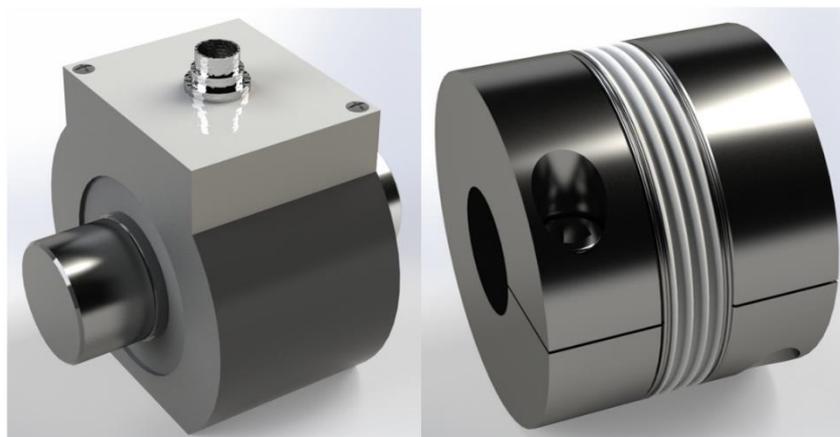


Figure 3.2.6: Rendered CAD of the *1-T22/500NM* torque meter by HBM (left) and of the *3-4412.0022* flange (right).

Such torque meter is joined to the shafts, on its two sides, by means of two flanges belonging to the category 3-4412.0022 [34] (Figure 3.2.6); the maximum torque that this kind of flange can bear is 1 kNm.

At the time of writing, the transmission system has not been implemented in the test bench yet, due to technical problems during the manufacturing of the flanges, which did not allow the correct insertion of the torque transducer, also leading to misalignment. In future works, the shafts manufacturing will be commissioned again, and then the shafts will be mounted properly. In Figure 3.2.7 is depicted a 3D rendering of how the overall transmission subsystem will look like.

Eventually, to increase the safety of the working environment, a protection (Figure 3.2.8) has been designed, manufactured by the company Meccania s.r.l, and then added to the test bench setup. It is intended for reducing the risk of reaching the rotating parts, and also to protect the workers in the unlikely possibility of failures. The protection is composed by three Lexan panels and a modular aluminum frame: this allows the user to adjust it when needed to access the shaft and torque meter, without moving the whole structure. The frame is designed to lean against the flange of the bench, and a pair of screws are employed to fix the two parts together.

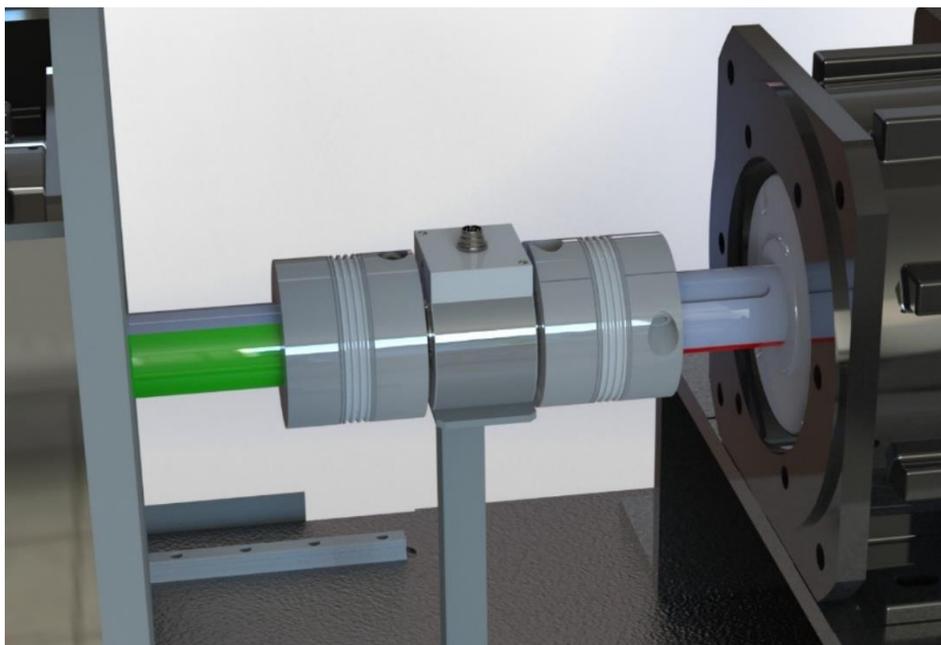


Figure 3.2.7: The transmission subsystem. 3D rendering.



Figure 3.2.8: The protective structure. 3D rendering.

3.2.4 The generator

The choice of the generator is not directly part of this thesis work, but for the sake of completeness its characteristics are briefly discussed in this paragraph. On top of that, the exact device model has not been selected yet at the time of writing, thus it has been not included in any of the experimental activities reported in this dissertation. For this reason, one suitable generator is here presented in its generality, but its feature may be different from the actual one that will be implemented in the final test bench. Therefore, in the following it will not be discussed as deeply as the other components of the experimental set-up.

The generator described here is a *EOGEN 150/16*, and it is produced by the company Mecc Alte. It is a three-phase, sixteen-poles, synchronous generator with permanent magnets. Its main performance parameters are listed in Table 3.2.5. On the other hand, the two following figures illustrate respectively the ideal mechanical characteristic (Figure 3.2.9) and the efficiency and power characteristics with respect to the angular velocity from the datasheet (Figure 3.2.10). Finally, at the end of the paragraph, the behaviour of the machine in terms of losses and efficiency is derived.

Note that, even if the component that will be eventually chosen for the actual implementation of the rig may be different from “EOGEN 150/16”, its

features and characteristics will be chosen purposely to be comparable and compliant with those of the generator presented here.

Rated speed ω_n	415 rpm
Maximum speed ω_{max}	4200 rpm
Rated torque T_n	73 Nm
Rated power P_n	3kW
Rated voltage V_n	400 V
Rated current I_n	5.7 A
Power factor $\cos \varphi$	0.84

Table 3.2.5: Rated quantities of the generator EOGEN 150/16 by Mecc Alte.

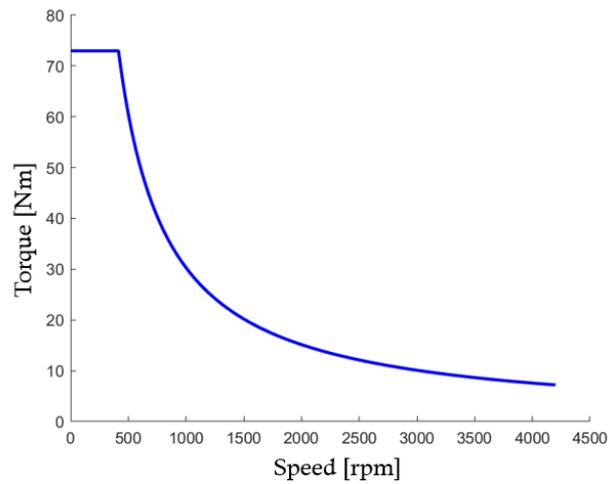


Figure 3.2.9: Mechanical characteristic of the generator EOGEN 150/16 by Mecc Alte.

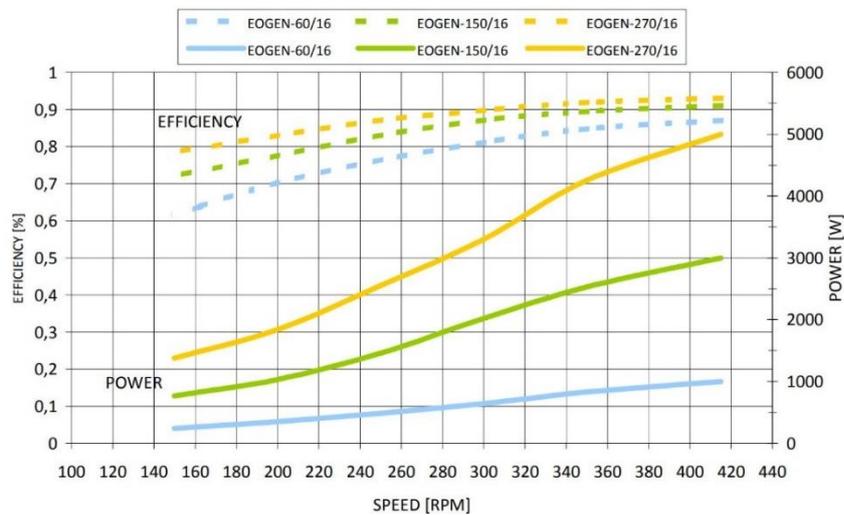


Figure 3.2.10: Efficiency and power characteristics from the datasheet of EOGEN 150/16 (in green).

- Absorbed electrical in rated conditions: $P_{n,e} = \sqrt{3} V_n I_n \cos \varphi = 3.317 \text{ kW}$
- Total losses in rated conditions: $P_{n,loss} = P_{n,e} - P_n = 0.317 \text{ kW}$
- Efficiency: $\eta = P_n / P_{n,e} = 0.904$

3.2.5 Mechanical constraints

Now that the three main components of the test rig have been defined, it is useful to perform a mechanical cross-analysis to evaluate the limit working conditions of the bench as a unique system.

In Figure 3.2.11, the nominal values of the torques of the three main components of the test rig (motor, transmission subsystem, generator) are compared, with respect to the angular speed. It is possible to notice that, whereas the torque meter has a constant T_n , regardless of the ω , the motor and the generator have a constant nominal torque only until a certain speed, and afterwards the trend decreases. Moreover, the element of the bench whose T_n is the smallest is the generator (73Nm). This means that the mechanical parameters must be finely tuned in order to avoid such limit to be crossed by the operating point, and to allow the generator to work properly without malfunctioning or breakdowns.

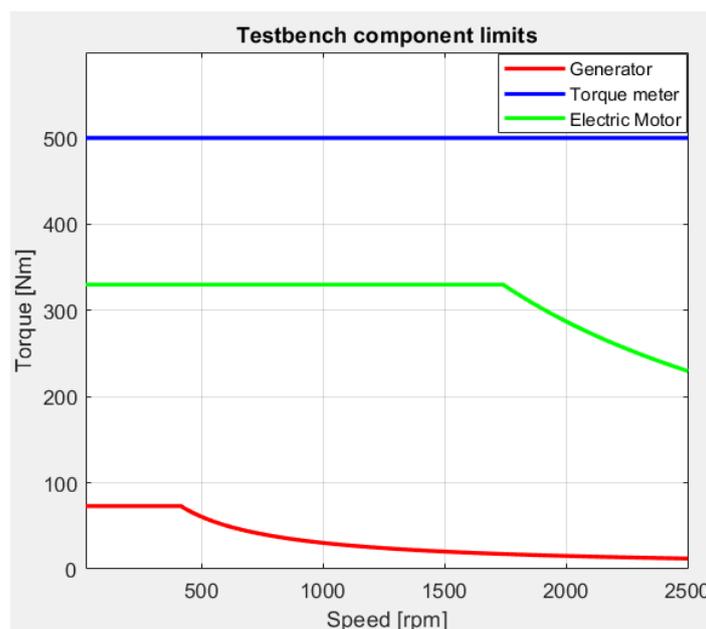


Figure 3.2.11: Comparison of the critical torque of the three main components of the test bench.

However, it is to be discussed the power supply of the actual test bench setup, in the Energy Center Lab. Here, the motor drive is connected to the electrical grid by means of a 63A current outlet. As a consequence, the actual working point of the drive (and therefore of the motor too) will be way smaller than the nominal conditions ($I_n = 145\text{A}$ for the drive), and this will help the user to reduce the risks regarding the generator limit discussed above.

3.3 The controller

After having completely defined the simulation plant, the next section to be depicted is the controller of the Hardware-In-the-Loop framework. The chosen hardware is a CompactRIO (or simply cRIO) by National Instruments (NI). Such controller is programmed with a suitable development personal computer, by means of the software NI VeriStand. Figure 3.3.1 represents in a schematic way the controller set-up, and the single components are described thoroughly in the following paragraphs.

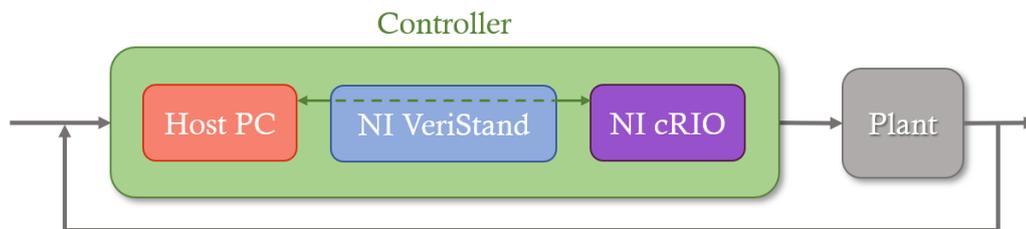


Figure 3.3.1: Schematic representation of the controller set-up, within the HIL feedback framework.

3.3.1 CompactRIO

The cRIO controller is defined by NI as “*a rugged, reliable, high-performance, industrial-grade embedded controller with industry-standard certifications. It is ideal for applications that need waveform acquisition, high-speed control or signal processing, hardware algorithm acceleration, hardware reliable tasks, or unique timing and triggering*” [35]. The chosen model is, in particular, cRIO-9040 (Figure 3.3.2)

which is managed by the NI Linux Real-Time operating system (OS), in the version 64-bit [36].

From a structural point of view, there exist in the market many different cRIO versions, but they all are characterized by a single chassis containing these main components:

- **CPU** (Central processing unit). cRIO-9040, in particular, has a dual-core Intel Atom E3930 processor with a base frequency of 1.3GHz.
- **FPGA** (Field-Programmable Gate Array), i.e. a kind of integrated circuit which is suitable for being custom-configured directly by the user after manufacturing. cRIO-9040 has a reconfigurable Kintex-7 7K70T FPGA by Xilinx.
- Multiple slots for **input/output modules** (I/O). This the feature of cRIO controllers makes them exceptionally flexible, thus extensively used in the industry. A high number of different I/O modules are produced by NI itself, which covers most of the possible applications. In addition, cRIO controllers can also be employed with third-party modules. cRIO-9040, in particular, has a total of four I/O slots.



Figure 3.3.2: A picture of CompactRIO-9040, by National Instruments. [35]

The chosen controller also contains a user interface featuring several ports for communication and data logging (RS-232 and RS-485 serial ports, three USB ports, SD slot, four LEDs and one pushbutton), together with the power supply terminal block. Indeed, with this kind of architecture, an external hardware intended for powering-up the device is needed. In the laboratory set-up, a National Instrument industrial power supply, model PS-15, has been employed (24 VDC, 5A).

Last but not least, cRIO-9040 has two Ethernet ports (RJ-45). This is a crucial point, because those models of cRIO having multiple Ethernet jacks can be exploited as masters in an EtherCAT network. Such feature has been used in this work, as discussed in Paragraphs 3.4 and 4.1.2.

From a performance standpoint, one of the key characteristics of cRIO controllers is their real-time reliability, which makes them suitable for critical applications where an accurate and precise timing is mandatory, with minimal latency. cRIO-9040, for example, guarantees a clock accuracy of 40 ppm (parts per million, i.e. $\pm 0.004\%$) at 25°C.

cRIO-9040, moreover, presents multiple programming modes. Such modes represent the ways to access each I/O module slot, and they provide different programming scenarios according to the peculiarities of the modules and to the usage they are intended for. The modes are three, and are here listed in increasing order of control loop speed [37]:

- **Scan Engine Mode:** it is designed for applications requiring I/O control loops up to 1 kHz. It is the easiest mode from a development point of view, but also the slowest.
- **Real-Time Mode (NI-DAQmx):** it has control loops up to 5kHz and it is optimized for waveform acquisitions and hardware timing.
- **FPGA Mode:** it is the fastest mode and at the same time the most complex, since it demands FPGA programming to the user. It attains I/O updates in the MHz range; thus, it is suitable for high-performance applications.

3.3.2 Current input module: NI 9253

As written earlier, NI cRIO controllers have multiple slots intended for hosting I/O modules. In our set-up, only one module has been utilized. It is, in particular, an analog, differential current input module, whose model number is C-series NI 9253 [38].

In the architecture of the test bench, the NI 9253 module (Figure 3.3.3) has been exploited to send to the controller a feedback from the torque meter transducer,

which is able, as stated in Paragraph 3.2.3, to give as output either a voltage or a current signal. In this case it has been decided to opt for a current signal, as it is known to be a more stable and robust solution when it comes to transmitting small signals. The output range of the torque meter is indeed $10 \pm 8 \text{mA}$.



Figure 3.3.3: A picture of the NI 9253 current input module, by National Instruments. [38]

NI 9253 provides eight analog differential input channels that are completely independent from one another, allowing the user to acquire up to eight signals simultaneously. Each input is buffered, conditioned and sampled by one distinct ADC (analog to digital converter); they can also be treated by means of built-in digital programmable filters. The overall conditioning circuitry is depicted in Figure 3.3.4.

Each ADC has the following characteristics:

- Signal range of $\pm 20 \text{mA}$;
- Resolution of 24 bit;
- Conversion frequency of 50kS/s (i.e. samples per second).

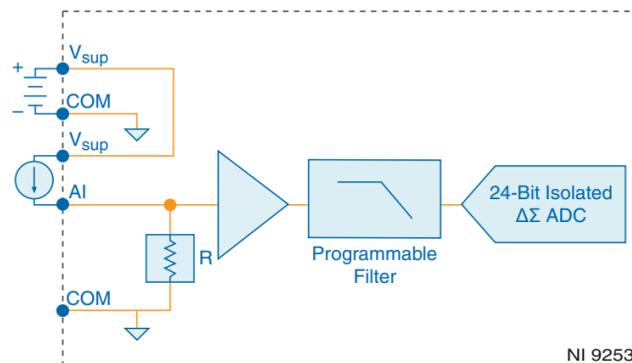


Figure 3.3.4: Conditioning circuit of the input channels of NI 9253. [38]

3.3.3 NI VeriStand

As already written in paragraph 3.3.1, cRIO controllers (together with their I/O modules) are meant to be configured by the user in order to fit the custom-specific framework they are going to be inserted. This can be done by means of apposite pieces of software developed by National Instruments.

Generally, the most used software in this context is LabView, because it provides the greatest variety of functionalities. Despite this, in the laboratory set up of this test bench, the VeriStand software has been utilized instead. Indeed, it is specifically designed for HIL simulation, and, as stated in the NI website, *“VeriStand application software helps you configure I/O channels, data logging, stimulus generation, and host communication for NI real-time hardware. You can also import simulation models and control algorithms”* [39]. In other terms, VeriStand allows the user to import, within such programming environment, simulation models written in a wide range of different programming languages, including Simulink.

In this regard, the user can simply compile the model of interest, in order to obtain an equivalent model written in a programming language that can be read by VeriStand. On the other hand, in LabView there is no direct way to import smoothly a Matlab/Simulink model. The Model Interface Toolkit add-on could have been used for this purpose, but the programming endeavor would have been significantly greater, therefore the final choice fell on VeriStand.

By the way, the importing process has not been a trivial task, for several reasons. The overall process is described deeply in Chapter four.

3.3.4 Host PC

The set-up is completed by a development PC (or host PC). Basically, it is the interface that the user exploits to configure the CompactRIO controller (via VeriStand).

In a first moment, when the test bench was still not available and the greatest part of the effort was being put in the model translation from Simulink to VeriStand, an Acer Aspire 3 notebook with processor Intel Core i7-10510U has been utilized as development PC. Nevertheless, afterwards it has been

decided to add to the laboratory set-up a suitable electrical panel (Paragraph 3.3.4) containing all the needed pieces of hardware, hence eventually an Intel NUC 9 with processor Intel i5QNX has been purchased to be employed as host computer.

This choice is due to many reasons: of course, its small dimensions (238 x 216 x 96mm) make it suitable to be inserted inside the electrical panel, whereas its performances are still satisfactory for the purpose. Moreover, the NUC family is a modular computer and therefore it has been possible to purchase separately all the needed peripherals, then mounting a host PC with customized and suitable characteristics.

In particular, an additional 480 GB SSD (solid-state drive) memory and a 16 GB RAM memory stick have been mounted inside the computer chassis. Eventually, the workplace has been completed with a desk hosting the monitor, mouse and keyboard.

On the NUC host computer, the Windows 10 OS (operating system) has been installed. Then, two fundamental pieces of software has been installed too: *NI-MAX* (Measurement and Automation Explorer, version 20.0.0) and *NI Package Manager* (version 20.6.0). The first is included with the basic NI drivers, such as *NI-DAQmx*, *NI-VISA* etc., and it is used as interface between the host computer and the hardware provided by National Instruments (cRIO-9040 and the NI-9253 module, in our case).

NI Package Manager, instead, is used to download and install all the other needed pieces of software, add-ons, device drivers etc. The following list describes shortly all the programs that have been installed, making sure to respect the software/hardware compatibility tables.

- *NI LabView* 2020 f2 (32-bit), with the add-ons *LabView FPGA Module* 2020 f1 and *LabView Real-Time Module* 2020.
- *VeriStand* 2020 R3, with the add-on *Scan Engine and EtherCAT* 20.3.0.
- *NI CompactRIO Device Drivers* 20.5.0

Afterwards, it is possible to install on the target hardware the suitable drivers, programs and engines. This step is performed through *NI-MAX*. The set of pieces of software is reported here:

- *LabView Real-Time 20.0.0*
- *NI VeriStand Engine 20.3.0*
- *NI-RIO 20.0*
- *NI Scan Engine 9.0*
- *NI-Industrial Communications for EtherCAT 20.0.0*
- *I/O Variable Remote Configuration Web Service 20.0*

All the information about the installation procedure, the software to download, and the compatibility among releases of different products has been found on the National Instruments website [35].



Figure 3.3.5: Internal circuitry and components of the electrical panel. The cRIO-9040 controller with its power supply (to the left) are situated in the middle level. On the bottom, instead, there is the Intel NUC9i5QNX host PC.

3.3.5 Electrical panel and miscellaneous

Eventually, the laboratory set-up is completed by some miscellaneous elements. First, an electrical panel (Figure 3.3.5) has been designed to contain the cRIO-9040 controller, its power supply, and the Intel NUC host PC, together with all the needed circuitry, cables, switches. The electrical panel also features a set of wheels to be moved, a cooling system, an electromagnetic shielding, Schuko sockets, and an external interface with cable entries (USB, Ethernet RJ-45, HDMI) that allow the communication between the components that are internal to the panel with the external ones, or user PC, data storage devices et cetera.

Lastly, an emergency pushbutton has been added to the set-up. It is directly connected to the motor drive STO (safe torque off). The latter, indeed, is a standard electronic signal, commonly integrated in drives for safety reasons. It basically avoids the energy to reach the motor, therefore no torque can be generated when the STO signal is at logical one. This implies that, when the pushbutton is pressed, unwanted starts of the motor are avoided; otherwise, if already running, it decelerates until reaching a null speed.

3.4 EtherCAT

The communication between the CompactRIO and the motor drive has been implemented exploiting the *EtherCAT* protocol (“Ethernet for Control Automation Technology”, ECAT for short). It is a fieldbus, Ethernet-based, open industrial technology which was invented by Beckhoff Automation in 2003 and then defined as an international standard by IEC (International Electrotechnical Commission) in 2007 [40]. ECAT has in common with the Ethernet standard only its first layers: the physical layer (i.e. the fieldbus) and the data link layer. The basic architecture of a EtherCAT network is a closed ring topology with master/slave dependency, where the master is the only module able to send data (in the form of frames) downstream in the network.

EtherCAT is part of the Fieldbus family (IEC 61158), namely a standard of industrial computer networks exploited for distributed complex automated

systems. With respect to other members of such family, ECAT is characterized by multiple features which make it one of the most common protocol in the market, such as:

- **Simplicity.** The configuration, diagnostics and maintenance are easier from the perspective of the user, who can avoid setting manually the addresses, detecting disturbances and so on, while also reducing significantly the costs.
- **Bandwidth utilization.** EtherCAT is able to exploit the Ethernet layers in a more efficient manner; indeed, it can read and write data while the frame is moving downstream (“*on-the-fly*”), with a minimal delay being introduced. In this sense, while the common Ethernet standard can be considered a *half-duplex* technology (meaning that it can transmit data in both directions but not simultaneously), ECAT is instead a *full-duplex* technology.

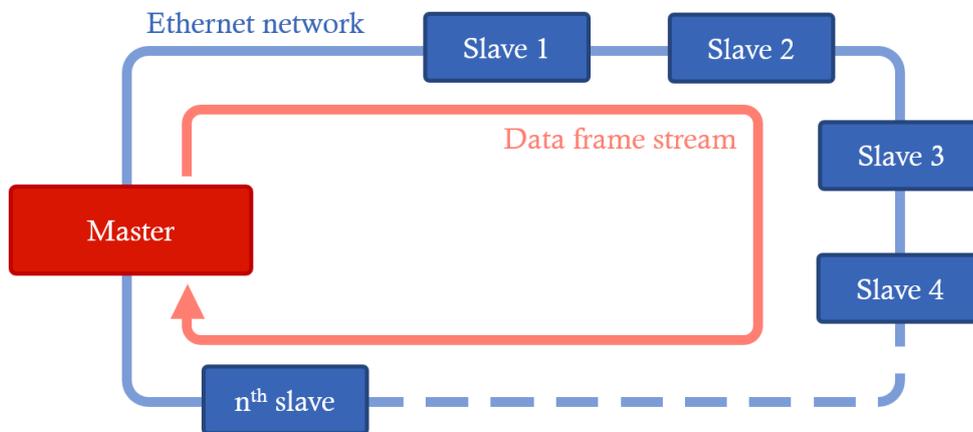


Figure 3.4.1: Typical closed ring architecture of an EtherCAT-based network.

- **Real-time and deterministic performances.** The wider band makes EtherCAT the fastest industrial Ethernet technology in the market, with an accuracy in the order of nanoseconds [40]. This implies that ECAT is particularly suitable for applications controlled via bus system, in which deterministic performances are requested (e.g. hard real-time requirements). Moreover, high-precision synchronization is achieved by means of a system of distributed clocks intended for reducing furtherly the jitter.

- **Flexibility and robustness.** ECAT does not need hubs or switches, therefore it is virtually limitless in terms of network topology and it can be employed with any combinations of line, tree and star linking. EtherCAT also features automatic link detection. Moreover, a bunch of communication profiles have been implemented in order to enlarge the spectrum of device and applications supported by ECAT (whose deepening is outside the scope of this thesis):
 1. CANopen application protocol over EtherCAT (CoE);
 2. Servo drive profile (SoE);
 3. Ethernet over EtherCAT (EoE);
 4. File access over EtherCAT (FoE);
 5. ADS over EtherCAT (AoE).

In our case, the chosen communication profile is CoE; the reason is going to be clarified later in this paragraph.

As stated in Paragraph 3.3.1, our cRIO-9040 can be exploited as EtherCAT master. This means, in practice, that an ECAT network could be utilized to allow the data transmission among the components of the test bench with no need to use another intended component (such as a PLC) that would have represented a great effort in term of time and money.

The network is completed by the motor drive, which is at the time of writing the only ECAT slave connected to the master. In future works, it will be possible to insert additional slaves (e.g., the generator drive) in a relatively easy manner.



Figure 3.4.2: FECA-01 EtherCAT adapter module by ABB. [41]

The communication is made possible, on the motor drive side, by a suitable adapter module, *FECA-01 EtherCAT* by ABB (Figure 3.4.2). It is “*an optional device for ABB drives which enables the connection of the drive to an EtherCAT network*” [41]. By means of such adapter module, among other things, the user can utilize the ECAT network to:

- Command and control the drive (e.g. Start, Stop, Run, Reset, Enable, etc.);
- Provide reference profiles, such as speed, torque or position references, to the drive (and therefore to the motor);
- Read information and values from the drive;
- Modify the parameters and values of the drive during run-time.

To establish the communication with the FECA-01 adapter module, the programmer has to import in the programming environment (VeriStand, in our case) a certain XML file, provided by the manufacturer, that is intended for allowing the master (i.e. cRIO-9040) to recognize the drive as a slave in the ECAT network. The whole procedure is going to be described in Chapter four.

On the other hand, the module has to be programmed using a certain device profile, namely the set of parameters which define the configuration of the link between the hardware and software. The FECA-01 module that has been employed in the test bench is based in particular on the CiA 402 standardized device profile, which is part of the CANopen specification, used for controlled motion products such as drives [41]. This is also the reason why the selected communication profile of the EtherCAT network is CoE, as stated earlier in this paragraph.

After having described thoroughly the whole structure of the Hardware-In-the-Loop test bench, the discussion will continue, in the next Chapter, with a focus on the programming environment: VeriStand. Figure 3.4.3 depicts in a simplified manner the entire architecture of the rig. It is worth underlying again that the generator and the transmission systems are still not present at the moment of writing. Nevertheless, in future developments of the work, when those parts will be available, thanks to the EtherCAT features it will be relatively easy to add to the ECAT network the generator drive, if any, and then to control it by means of the cRIO controller.

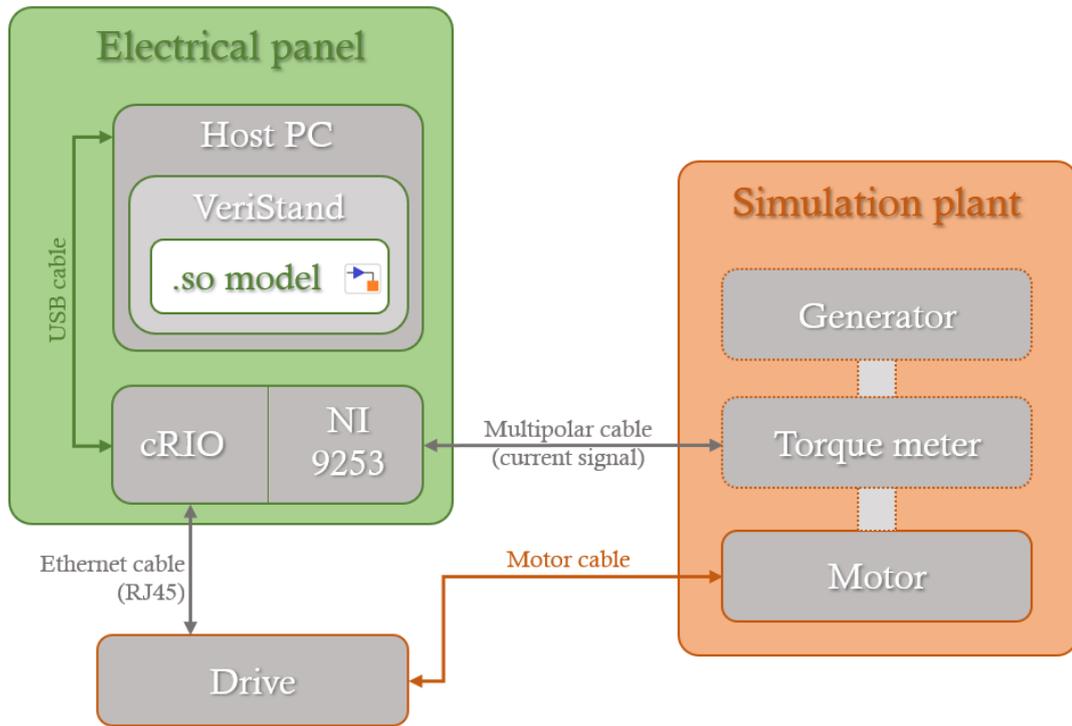


Figure 3.4.3: Schematic representation of the overall architecture of the test bench.

Chapter four

Test bench startup and first tests

In the present Chapter, the preliminary steps to be carried out in order to startup the test bench are described. In the first part of this section, the relevant features of the VeriStand software are presented, highlighting how it has been used throughout the thesis. In the second part, instead, the first operational tests are reported, with a particular emphasis on the process for setting the PID parameters of the drive. Finally, some relevant tests and data for the validation of the motor performances are highlighted.

4.1 Preliminary operations

This paragraph explains how to create a VeriStand project having as target the cRIO-9040, and how to set the latter as master in an EtherCAT network with the ACS880 motor drive as slave. Then, the drive itself is taken into account and its parameter system (i.e. the data that are exchanged with the rest of the test bench) is described. Eventually, the procedure that has been used throughout the simulations to exchange data between VeriStand and Simulink is discussed.

4.1.1 Set-up of a VeriStand project and target definition

The very first action to be performed is selecting the desired target of the VeriStand project, that in our scenario is the CompactRIO-9040 controller. This can be done in the *System Explorer* tool, which is the section of VeriStand in which is possible to define the overall system (declare targets and/or custom devices, import external simulation models, define aliases et cetera). This step can be

done either manually, selecting *Add Target* under the item *Targets* of the project tree, or automatically, by pressing *Hardware Discovery Wizard*. In either case, if the cRIO is correctly connected and all the pieces of software, add-ons and device drivers have been installed consistently (both on the target and on the host PC itself), then VeriStand will be able to communicate with such hardware, and it will be able to deploy the project.

Under the item representing the newly added target, the user can customize some critical settings, such as the operating system, IP address, the username and password (if any) of the controller. Also, here it is possible to assign modes to the processors (PCL, Primary Control Loop and DPL, Data Processing Loop), and set the timing rates of the target. In this work, all the settings have been left in default, except for the target timing definition, which has been modified according to the particular simulation performed.

4.1.2 EtherCAT connection to the motor drive

To establish the connection to the motor drive, the EtherCAT protocol has been exploited. As already discussed in Chapter three, in this test bench architecture the cRIO-9040 controller is also the master of the network, and it has to be set accordingly in the NI-MAX software. Here, the user must select the connected cRIO and choose *EtherCAT* from the drop-down menu. By doing so, the controller is set to behave as an ECAT master.

Afterwards, the existence of the ECAT network must be added in the *System Explorer* of the VeriStand project. This can be done adding a *Scan Engine and EtherCAT* item as a *Custom Device* (which in turn can be found under the *Controller* item). Indeed, as stated in the VeriStand software itself, “*the Scan Engine and EtherCAT custom device allows you to communicate with I/O in local and remote chassis using the NI Scan Engine. This is primarily used to communicate with CompactRIO chassis and EtherCAT hardware*”.

After that, the user can add manually several masters and slaves to the network. In our architecture, in particular, one master and one slave are sufficient for the purpose. When the ECAT master is added, it appears in the

System Explorer, it is important to verify that its address is consistent with the one that has previously been set in NI-MAX. The default address is 0.

On the other hand, to add slaves to the system definition, the developer must firstly import their respective ESI (“EtherCAT Slave Information”) files, both to the host PC and to the target. Such files are XML text files, and they are provided by the vendor (ABB in this case) along with the drive and the ECAT module adapter (FECA-01). The ESI file is needed by the master in order to acknowledge that particular adapter as a slave, and therefore to properly control the data exchanged through the network. In Paragraph 4.1.3 such topic is going to be addressed more in detail.

To add the ESI file to the VeriStand project, the programmer has to click on *Manage 3rd Party ECAT Slaves*, select the path of the required XML file and import it both to the host computer and the real-time target. Afterwards, it is possible to click *Auto-Detect Modules*, and if the imported ESI file is consistent with the featured slave, and if the slave itself is connected properly, then the slave will appear in the project tree, together with those inputs and outputs that can be controlled by means of the VeriStand software. In Figure 4.1.1, a screen capture displays the final appearance of the System Explorer tree menu after the described procedure.

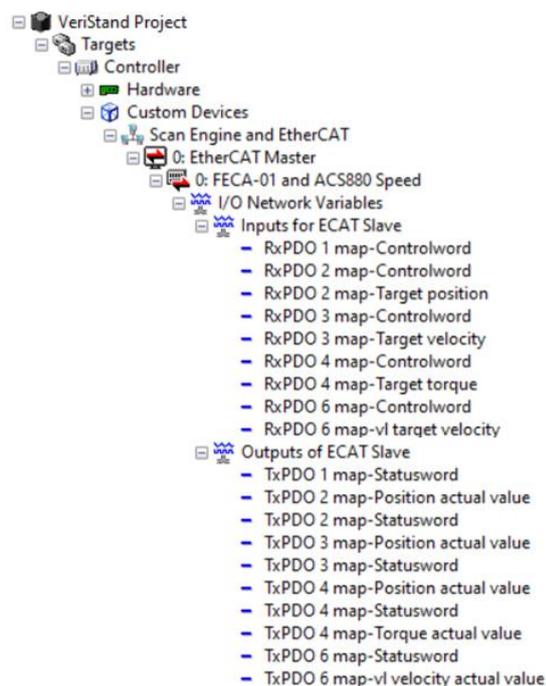


Figure 4.1.1: EtherCAT slave added to the VeriStand project configuration.

By the way, the step that has been just described can be performed successfully only if *I/O Variable Remote Configuration Web Service* has been previously installed on the CompactRIO device. Such software allows the controller to constitute the EtherCAT network.

When the programmer clicks on *Auto-Detect Modules*, aside from the slaves present in the configuration, also the I/O modules inserted in the cRIO controller will be recognized by VeriStand. During this step, the user can also select which programming mode will be used for each of the employed modules (see Paragraph 3.3.1). In our architecture, the only module is NI-9253, and the selected programming mode is *Scan Engine*. This choice, indeed, even not being the best in terms of I/O acquisition velocity, guarantees a maximum control loop rate of 1 kHz, which is sufficient for our purposes.

4.1.3 The drive parameter system and mapping issues

The internal logic of the drive is organized by means of several parameters, which contain data representing the motor behavior and defining a wide range of operating conditions, limitations, internal architectures of the drive circuitry etc. The parameter architecture widely expands the possibilities for the engineer to customize the hardware and its relationship with the other components.

For the drive user it is crucial to know how to access, read, and modify those parameters. One possibility to do so is with the interface situated on the drive front panel; otherwise, it is possible to connect via USB the drive to the host PC and communicate exploiting *Drive Composer*, a software by ABB. The latter option provides some additional features, such as a monitoring tool that allows to quickly plot the timing behavior of the parameters.

By the way, in this thesis work, VeriStand itself has been employed to interact with the drive. In this regard, the ESI file that has been imported in the controller defines the mapping of a certain set of parameters, which then appear in the VeriStand *Project Explorer* as inputs and outputs of the slave device. In detail, the process data that at each cycle is transferred by the ECAT slave constitutes the so-called PDOs (process data objects) [42]. To allow a correct

master/slave communication, the ESI file is used to define which PDOs are transferred, their size, type, location etc.

Alas, the standard ESI XML given by ABB for the FECA-01 ECAT adapter module showed some issues in our architecture, due to the fact that the controller is in this case third-party (i.e., not manufactured by ABB itself). Therefore, the ECAT master is only able to access four of the drive parameters. In other words, by means of the VeriStand project only those four parameters can actually be exploited by the programmer. The solution to this problem (which requires a high-level knowledge in the field of the XML language) has not been found yet by the time of writing, thus limiting the range of tests that could be carried on so far. In future works this topic is going to be addressed, in order to allow a greater variety of drive parameters to be accessible from cRIO.

It is useful to highlight once again that anyhow all the parameters can be accessed directly either manually with the front interface of the drive, or by means of the *Drive Composer* software. Instead, the four parameters accessible in VeriStand are described in the following. They are all mapped into the PDO 6 group by the XML file (the parameters contained in other PDOs are visible but do not actually carry information due to the ESI problem described above). In particular, they are subdivided in two group of two parameters:

- **RxPDO 6** is the group of parameters that the slave receives from the master (i.e., the drive inputs that can be set via VeriStand).
 - **Control word** (a sequence of bits that the user configures to set the working condition of the drive).
 - **Target velocity** (the input value of speed that the motor is asked to follow).
- **TxPDO 6**, on the other hand, is the group of parameters transmitted by the slave back to the master (i.e., the drive outputs).
 - **Status word** (the feedback of the control word, i.e. a sequence of bits that communicates to the user the drive working condition).
 - **Actual velocity** (the output speed of the motor).

The latter parameter is intended to represent the speed read by the encoder, which is built-in inside the motor. There are several drive parameters

that are linked to the encoder velocity, and in fact the one which is mapped within the XML file as “Actual velocity” is an integer that continuously toggles between two adjacent values. The result is a non-optimal signal, noisy but still precise. For this reason, in some of the performed tests, a filter has been applied to smooth the signals of interest and allow an easier visualization.

This last issue is intended to be solved in future developments of the research, together with the other critical aspect of the XML mapping. Indeed, it would be sufficient to select, rather than the integer parameter currently linked to “Actual velocity”, another one with a greater accuracy.

4.1.4 Importing time sequences from Simulink to VeriStand and vice versa

VeriStand features an advanced tool for building stimulus profiles to be fed to the system of interest. Such tool is called *Stimulus Profile Editor*, and it is a VeriStand executable that allows the user to exploit several elements such as real-time sequences and subsequences, decisions, pre-defined profiles etc., to the aim of stimulating and evaluating the system channels. It also enables the customer to import CSV (comma separated values) files and to read them as real-time sequences; this has been particularly useful in our case, since it is fairly straightforward to generate this kind of files from Matlab/Simulink.

Using the *To workspace* Simulink block, indeed, the programmer can save into the Matlab workspace the needed signals; then, the CSV file can be generated by means of the *writematrix()* Matlab command. The data used to build in such a manner the CSV files has been obtained from one Simulink simulation of the entire system, so that each one of the created stimulus profiles represent a realistic working scenario for the model.

The CSV file to be imported in the Stimulus Profile Editor, by the way, must be formatted in a specific way. The first column is intended for the timestamps; for example, if a certain row of the file has 5 as first element, it means that the input there contained are inserted in the real-time sequence after that a period of 5 ms is elapsed from the start of the stimulus profile.

On the other hand, the first row must contain a series of headers, that are needed for VeriStand to identify univocally which parameter is represented by that particular column. Therefore, the first header of the first column is always ‘*timestamp*’. Figure 4.1.2 represents an example of CSV file that is suitable for the Stimulus Profile Editor. The first column must be consistent with the time-step defined in Simulink. In our simulations, $\Delta t = 0.05$ s, and this is why the ‘*timestamp*’ column contains a linearly spaced vector with spacing equal to 50 ms.

	A	B	C	D	E	F	G	H
	timestamp	x	dx	d2x	y	dy	d2y	z
	Number	Number	Number	Number	Number	Number	Number	Number
1	timestamp	x	dx	d2x	y	dy	d2y	z
2	0	0	0	0	0	0	0	0
3	50	0	0	0.17895860...	0	0	-0.0054098...	0
4	100	0.00022386...	0.00895798...	0.17949875...	-6.7881090...	-0.0002720...	-0.0053827...	-5.2006750...
5	150	0.00089630...	0.01794287...	0.17903469...	-2.7139368...	-0.0005424...	-0.0052966...	-0.0002081...
6	200	0.00201740...	0.02690386...	0.17630978...	-6.0906643...	-0.0008087...	-0.0056434...	-0.0004682...
7	250	0.00358312...	0.03572770...	0.17165396...	-0.0001084...	-0.0010921...	-0.0060736...	-0.0008322...

Figure 4.1.2: Example of a correctly formatted CSV file. Picture captured from the Matlab viewer.

Then, the created CSV file has to be called as a real-time sequence within the Stimulus Profile Editor. In this environment, the programmer can build the stimulus as a sequential piece of code. The code consists of three pre-set parts, that are *Setup*, *Main* and *Clean up*. In our simulations, only the *Main* has been coded, while the other two parts have been left blank.

An important consideration regards the timing between the deployment of the VeriStand project (i.e. the moment in which the model starts running on the target), and the actual beginning of the stimulus profile. If the user simply deploys the project on the cRIO controller, and then launches the profile, there will be a certain elapsed time between these two events which will result in an inconsistent behavior of the system. To avoid this kind of discrepancies, the programmer has to include in the stimulus profile code the command called *Deploy Active VeriStand Project*, right before the call to the real-time sequence. By doing so, the sequential execution of the code will include itself the deployment of the project, minimizing in a satisfactory way the elapsed time.

Figure 4.1.3 depicts an example of code used to generate the profiles used to stimulate the motor. It is possible to notice that a logging operation has been added, in order to save in a text file the results of the simulations.

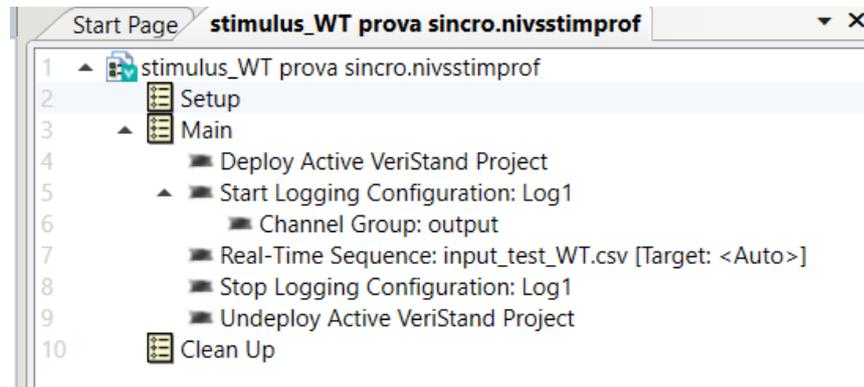


Figure 4.1.3: Example of stimulus profile code. Picture captured from the Stimulus Profile Editor tool.

The data obtained by means of the simulations has been then plotted in a VeriStand workspace, i.e. a tool which allows a smart interface between user and machine. Among the other things, it allows to display the real-time behavior of relevant signals belonging to the VeriStand project, through customizable plots. Exploiting this feature, it has been possible to visualize the needed graphs representing the I/O, and other critical quantities, if any.

Eventually, the graphed output has been exported in an Excel file (XLSX) which has been afterwards imported in Matlab/Simulink. By doing so, it has been possible to plot in one single graph the data obtained simulating different versions of the same experiment, thus detecting in a more effective manner pros and cons of each of them.

4.2 The first tests on the motor and PID selection

After all the preliminary steps, it is finally possible to actually startup the motor. The control word has to be set in order for the motor to start running; afterwards, the target velocity can be modified (using the numeric controls of VeriStand) and therefore the actual velocity of the motor will follow.

Among the drive parameters, the user can find the definition of the PID parameters that regulate the response of the drive (in particular of the speed

control loop, in our scenario). In the following, the procedure that has been applied for setting those parameters is described. Subsequently, some motor tests are reported. In those preliminary tests, the floating offshore wind turbine was not added to the VeriStand project yet, therefore their experimental results are only related to the validation of the mechanical behavior and the PID performances of the motor, prior to including of the actual model.

4.2.1 The PID controller: theoretical background

To regulate the motor behavior, the inner circuitry of the drive has been exploited. The motor has been controlled in speed employing the speed control loop integrated within the drive; such loop contains a proportional-integrative-derivative (PID) control logic. In this paragraph, such mechanism is briefly introduced, and the method used to set those parameters is depicted.

In general, the feedback control systems are based on the idea of reading the actual output and comparing it to the desired one. Subtracting the two, the so-called error is obtained; the control systems are indeed categorized by how such error is managed. The PID controller, in particular, is characterized by the fact that the computed error is processed to obtain three terms that are eventually summed up and used to apply corrections to the system output. Figure 4.2.1 depicts the typical structure of a PID control loop, and then the typical equation of the control function $u(t)$ is reported [43].

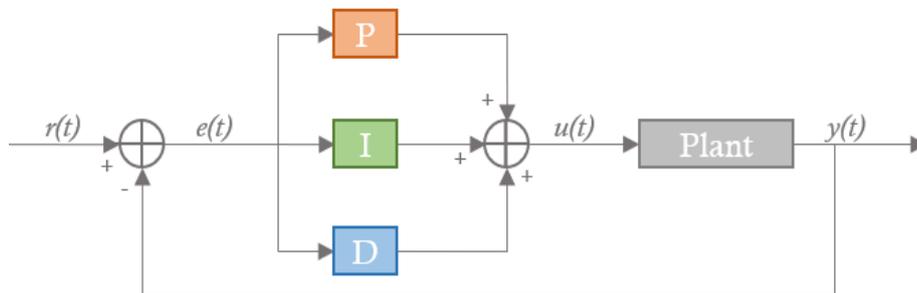


Figure 4.2.1: Schematic representation of a PID control loop.

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(t') dt' + T_d \frac{de(t)}{dt} \right)$$

The advantage of the PID control logic is that it only needs the tuning of three parameters (i.e. the proportional gain K_p , the integration time T_i , and the derivation time T_d) in order to completely define the effect of the controller on the process or plant of interest.

One of the most common procedure to tune such parameters in an effective way is provided by the so-called Ziegler-Nichols closed-loop method [44]. The algorithm is the following:

- Start with a purely proportional controller by neglecting the integrative and derivative terms, imposing $T_i = \infty$ and $T_d = 0$;
- Increase K_p until the output $y(t)$ is characterized by constant and stable (i.e. not divergent) oscillations around the desired output. Such value of the proportional gain is often referred to as the ultimate gain (K_u).
- Measure the oscillation period T_u and eventually set the value of the three parameters according to the following table, depending on the desired behavior:

Controller	K_p	T_i	T_d
P	$0.5 K_u$	-	-
PI	$0.48 K_u$	$0.8 T_u$	-
PD	$0.8 K_u$	-	$0.125 T_u$
PID	$0.6 K_u$	$0.5 T_u$	$0.125 T_u$

Table 4.2.1: PID parameter selection of the Ziegler-Nichols method.

4.2.2 PID parameter selection for the test bench

The Ziegler-Nichols method has the drawback of being heuristic and not deterministic, meaning that in some cases, especially in complex systems as the one described in this thesis, it is likely to lead to an undesired behavior. In particular, such method tends to push the loop to its stability limit, often leading to noise and/or instability issues.

This indeed happened in the case in analysis. The whole Ziegler-Nichols procedure has been followed, but the obtained PID controller ($K_p = 114,6$, $T_i =$

0.02 s, $T_d = 0.005$ s) showed a significant vibration which worsened its performances. For this reason, a more conservative approach has been eventually applied, starting from the default PID definition of the drive ($K_p = 10$, $T_i = 0.02$ s, $T_d = 0$ s) and then tuning those parameters. It has been noticed that the derivative action (although small) induced a motor vibration even for low velocities, therefore a PI controller has been selected. The final parameters are the following:

- $K_p = 15$;
- $T_i = 0.02$ s;
- $T_d = 0$ s.

Those values have been chosen with a trial-and-error procedure. Figure 4.2.2 shows the step response behavior of the designed PI controller, while 4.2.3 compares of three alternative PI controllers that have been taken into account during the designed procedure. In the examples in figure, T_i is constant (0.02 s) while the proportional gain K_p varies (10, 15, 20), and the step amplitude is 750 rpm. From Figure 4.2.2 it is possible to notice that the overall performance of the controller is satisfying (with a step in the order of magnitude of the actual wind turbine rotor velocity) both in terms of promptness and accuracy. In Figure 4.2.3, instead, to better visualize the response graphical features, the output motor velocity (that is as already discussed a noisy signal) has been post-processed in Matlab, applying a moving average filter with the aim of smoothing the characteristics.

Then, the performances of the motor have been tested in several conditions, using as target velocities both square waves and sine waves with many different amplitudes, frequencies, offsets etc. Some examples are showed in Figure 4.2.4 and 4.2.5.

Finally, Figure 4.2.6 has been obtained setting as target velocity a certain generator speed profile obtained by the Simulink model of the turbine *NREL5MW* (environmental conditions: average wave height speed $H_s = 0.75$ m, average wave period $T_p = 5.5$ s, average wind speed $V_0 = 8.5$ m/s). In particular, such figure depicts both the original view and a zoomed capture of the comparison plot between the wind profile and the post-processed output motor speed, smoothed with a moving average filter.

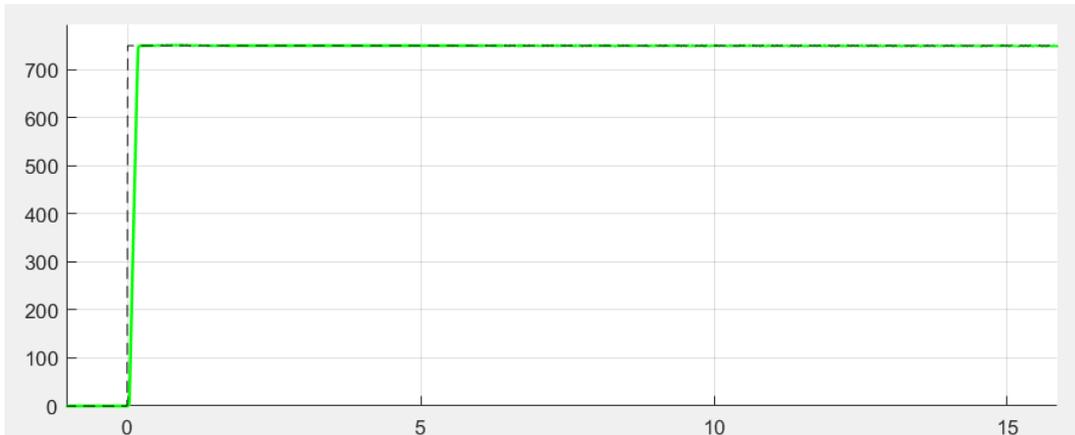


Figure 4.2.2: Step response (rpm vs. s) of the designed PI controller
($K_p = 15$; $T_i = 0.02$ s; $T_d = 0$ s).

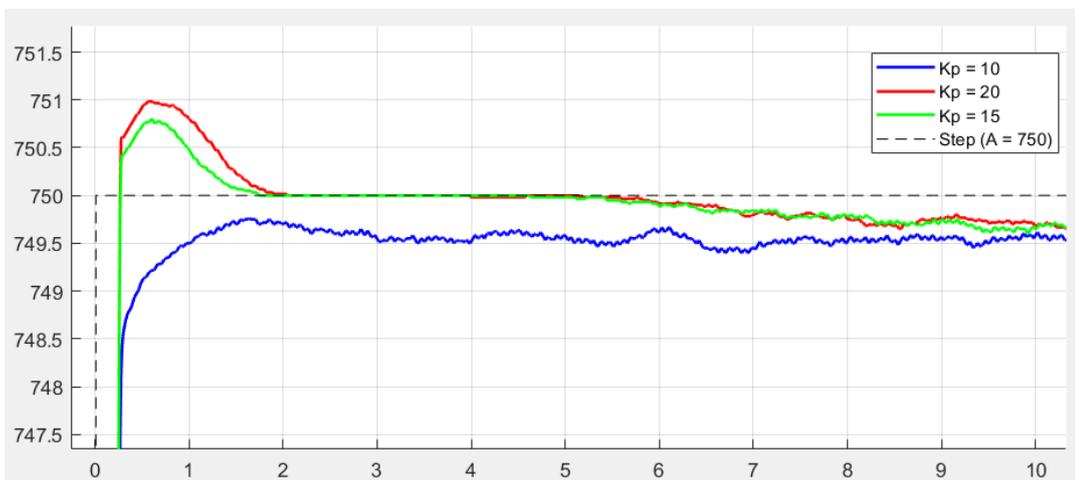


Figure 4.2.3: PID parameter selection. Post-processed step response (rpm vs. s) in three different cases
($K_p = 10, 15, 20$; $T_i = 0.02$ s; $T_d = 0$ s). Zoomed view.

In Figure 4.2.3. it is possible to visualize the step response features of the three considered controllers. The PI controller with $K_p = 10$, $T_i = 0.02$ s, $T_d = 0$ s (in blue) has no overshoot, but on the contrary it shows a certain slowness in reaching the steady state value. On the other hand, the controller with $K_p = 20$, $T_i = 0.02$ s, $T_d = 0$ s (in red) has better performances in terms of readiness, although presenting a small overshoot. Eventually it has been decided to opt for a PI controller with $K_p = 15$, $T_i = 0.02$ s, $T_d = 0$ s (in green) in order to slightly diminish the overshoot and still guaranteeing an optimal timing promptness. This choice is due to the fact that the input wind profile, in a turbulent scenario, could be characterized by step changes even with great amplitudes.

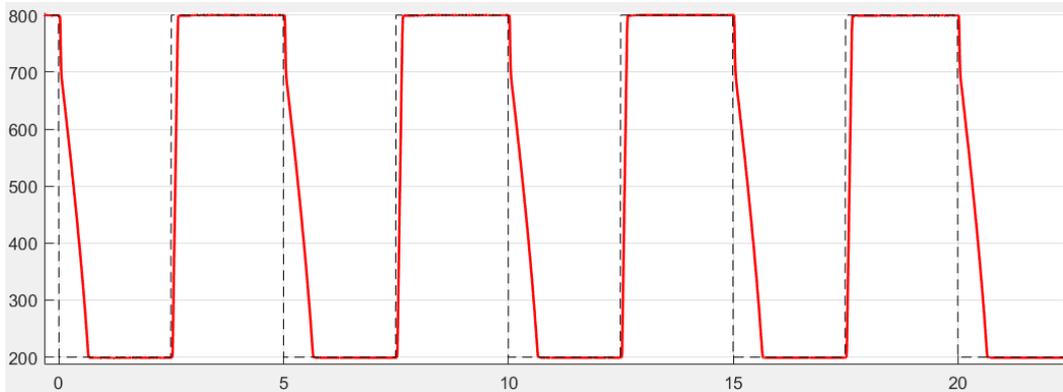


Figure 4.2.4: Motor speed performances. Square wave response (with peak-to-peak amplitude of 600 rpm, offset of 500 rpm, period of 5 s and duty cycle of 0.5).

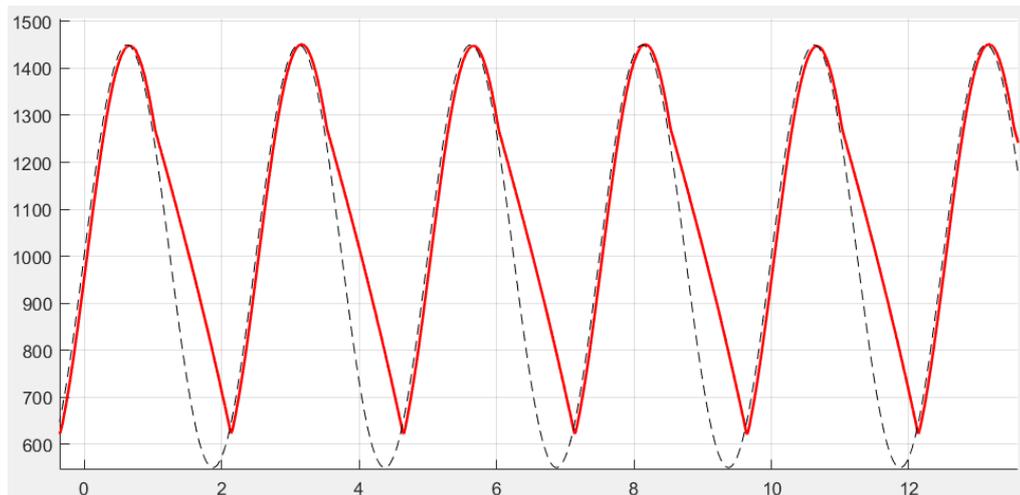


Figure 4.2.5: Motor speed performances. Sine wave response (with peak-to-peak amplitude of 900 rpm, offset of 1000 rpm, period of 2.5 s and frequency of 0.4 Hz).

Among the many motor speed tests performed with square wave and sine wave signals, in Figure 4.2.4 and 4.2.5 are depicted the two cases that were most demanding in terms of amplitude and frequency. From the first plot, it can be noticed that the output velocity presents a neglectable overshoot and undershoot. On the rising edge of the square wave, the motor reaches the target value with a rise time of 0.15 s, therefore accelerating at 4000 rpm/s. On the other hand, the deceleration time is significantly higher (0.35 s). This is due to a two-fold reason: on one hand, the internal structure of the motor (squirrel cage rotor with braking resistors) implies high ripples in the DC voltage during steep braking. On the other hand, the drive is set by default to apply a DC voltage limitation (by

modifying the torque accordingly) in order to avoid risky motor responses. The result is a slower deceleration, that could have been avoided by switching the drive settings; anyway, eventually this variation has not been implemented, because the motor performances are satisfying for our purposes.

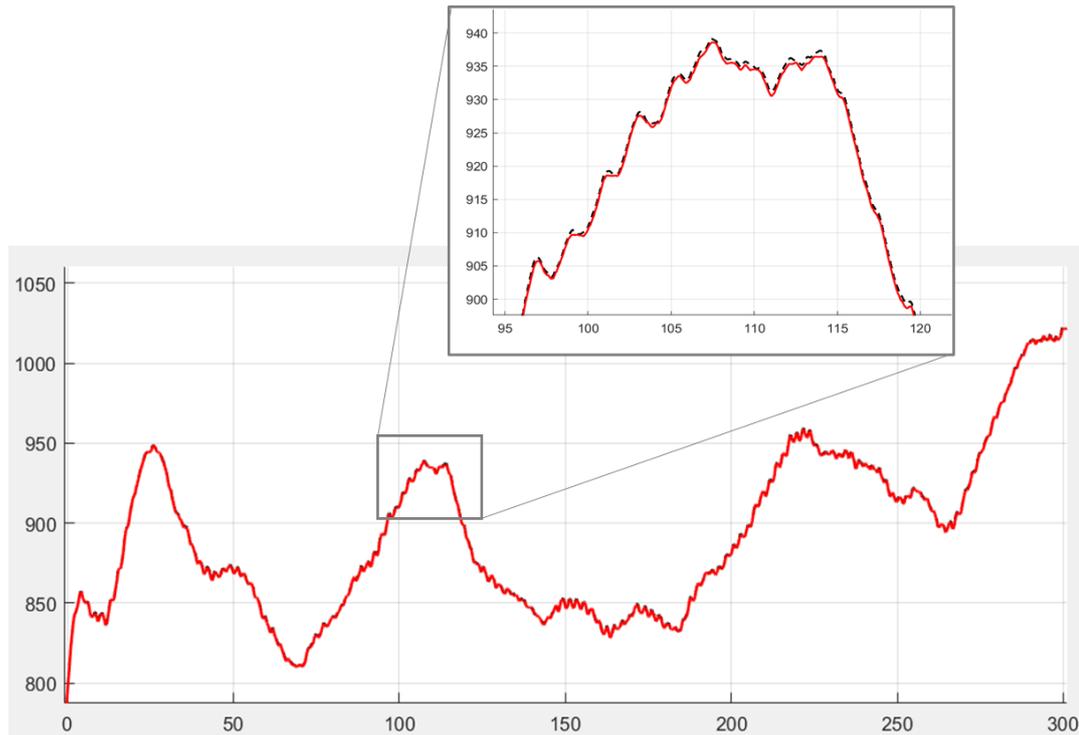


Figure 4.2.6: Motor speed performances. Filtered response to an example generator speed response ($H_s = 0.75 \text{ m}$, $T_p = 5.5 \text{ s}$, $V_o = 8.5 \text{ m/s}$). Original and zoomed views.

In Figure 4.2.6, indeed, it is noticeable how the generator speed profile used as example (that is characterized by a one-dimensional wind velocity which ranges between 800 rpm and 1020 rpm) is followed almost perfectly by the motor output speed (which has been post-processed in this case for the sake of visibility). This last plot validates the motor for the purposes of employing it in our test bench architecture; after such test, the research continued with the building procedure of the overall VeriStand project of the wind turbine.

Chapter five

The wind turbine VeriStand model

In this Chapter, the procedure followed to build a NI VeriStand model of the wind turbine showed in Chapter two is described, together with the problems that have been faced, the implemented solutions and some significant considerations. In particular, the compiling process of the model from Simulink to VeriStand is defined in detail.

5.1 Compiling: preliminary operations and parameter settings

As stated in Chapter three, VeriStand has been chosen because it allows the direct importation of Simulink models. This makes such software apt for the scenario of this thesis, considering that the wind turbine to be tested has been modeled in the Simulink software itself. In this paragraph, the compiling procedure is described.

In computer science, *compiling* means translating a certain piece of code from one programming language to another. In such framework, the starting artifact is the *source code*, while the final one is the so-called *target code*. Usually, a certain computer program (the *compiler*) is exploited to automatically manage the whole process, providing the target code without a great programming effort from the user point of view.

This is exactly the case of this thesis. Here, the source code is the Simulink model of the wind turbine *NREL5MW* (see Chapter two), which is written in the Simulink native language (*.slx*), while the target code has to be written in a programming language capable of being read by a VeriStand project running on a real-time target controller (i.e. the cRIO-9040).

The whole procedure has been performed following the instructions that can be found in the NI VeriStand manual [45]. The compiler to be chosen depends on the operating system (OS) of the target controller and on the kind of source code: for a Simulink file to be imported on a cRIO-9040 (running on a NI Linux Real-Time x64 OS), the suitable one is called *C/C++ Development Tools, Eclipse edition* [46], which can be downloaded and installed to the host PC online [47].

Such compiler works properly, in the Simulink scenario, not as a stand-alone but only with another compiler, *Simulink Coder*, which is instead developed by MathWorks. It is intended for generating C and C++ code from an *.slx* file. Once that the two compilers just mentioned are prepared, one can finally start the compiling.

Inside the Simulink model of interest, the programmer has to set the *Model Configuration Parameters* as expressed in the official guidelines of NI VeriStand [48]:

- **Solver.** The simulation must cover the whole positive axis of time. Thus, the simulation time shall start at 0.0s and last indefinitely (i.e. stop time = ‘inf’). Apart from that, the solver type has to be *discrete* (meaning that no continuous states are used during the solving algorithm) with *fixed-step* size. The used Δt (namely the step size or the fundamental sample time of the file), in our case, is inherited from the original Simulink model of the wind turbine ($\Delta t = 0.05\text{s}$).
- **Code generation.** This drop-down menu appears only if the Simulink Coder is installed correctly; here, the programmer can declare the features of the C-code to be created. In our scenario, the *System target file* (that is basically the specific compiler which is going to be used) is *VeriStand.tlc*. Such option will not be available unless the *C/C++ Development Tools, Eclipse edition* described above is downloaded. Eventually, the last mandatory setting is *VeriStand Compiler for Linux64 targets* under the item *Toolchain*.

The last option is needed because as already stated, the cRIO-9040 runs on a Linux64 RTOS (real-time operating system). This choice will also affect the

file extension of the generated target code, which in the case of interest must be an *.so* file (the other possibility would be a *.dll* file extension, which in turn is not suitable for our RTOS).

Finally, everything is set and the compilation process itself can be started, simply by building the Simulink model.

5.2 Compiling: problems and solutions

First, an attempt of compiling the whole Simulink model of the *NREL5MW* wind turbine (see Chapter two) has been made. As a result, however, the build process was not completed and returned an error, as the compiler “failed to generate all binary outputs”. In other terms, the model was too bulky in order for the toolchain to manage the creation of the desired *.so* code.

To overcome this issue, the flexibility of the NI VeriStand software has been of help. Indeed, it is possible to import there several simulation models, and to map their respective inputs and outputs in a custom way, therefore creating in practice a new model which is still runnable on the target hardware.

Such reasoning led to the decision of splitting the overall Simulink model into smaller sub-systems, compiling them independently, and finally rebuilding the top-level architecture of the system within the VeriStand project, connecting properly the inputs and outputs of the generated *.so* versions of those sub-systems.

To this end, for the sake of simplicity it has been decided to consider as sub-models to be compiled as stand-alone the three main blocks of the Simulink file. Such blocks, which represent the main parts of *NREL5MW* (see paragraph 2.2), are:

- Wind Turbine;
- Moorings;
- Hull.

By the way, this splitting operation was not enough for the compiler to run without errors, because the newly created sub-models were still too cumbersome. After a trial-and-error procedure, the cause has been individuated:

the original Simulink system contained multiple Look-Up Tables (LUTs), which were generated by means of the Matlab main script, and which contained millions of elements each. In the following paragraphs, the modification that have been made, in this regard, to allow the compiling are described for each of the three Simulink blocks.

On top of that, it also had to be considered the fact that the original Simulink model included continuous states, such as continuous-time integrators (and, in one case, also a state-space representation of continuous differential equations). This was due to the exploited solver algorithm, that was a fixed step *ode4*. Namely, the ordinary differential equations included in the system were solved in an approximated but still accurate manner using the so-called Runge-Kutta methods (i.e. iterative methods suitable for time-discrete framework, just like Simulink), which are suitable in a continuous-time scenario.

Nevertheless, in our case, the solver to be employed is fixed-step discrete (as stated earlier in this Chapter), and this implies that all the references, within the models to be compiled, to continuous states had to be switched to their discrete-time counterpart. For this reason, all the continuous-time integrators had to be replaced by discrete-time ones, with a unitary gain and an inherited sample time.

Using compiled models, by the way, leads to one criticality. Indeed, each Simulink model to be compiled is associated with a particular workspace that is respective to a certain environmental condition. This means that to test the VeriStand wind turbine model in different scenarios, it is necessary to compile the three models several times (once for each working condition, i.e. for each Matlab workspace) and then import them in the VeriStand project. This whole procedure is certainly time consuming, and in upcoming further developments of this work, such procedure is intended to be automatized.

5.2.1 Compiling the Wind Turbine block

Within the original Simulink system, the blades of the *NREL5MW* wind turbine were modeled through the BEM theory, as stated in Chapter two. For each of the three blades, three look-up tables were used to compute the time

evolution of the wind speed vector V_0 . This led to problems regarding the compiling process, because each of the nine LUTs contained several millions of elements which did not allow the compiler to allocate the needed bits.

To overcome this issue, the dimensions of the original workspace, which was linked to the Simulink model and generated by the main Matlab script, have been reduced. The simulation time has been decreased multiple times; then, the model has been compiled again. Such process has been repeated, with a trial-and-error procedure, until the compiling ran correctly, creating the desired `.so` version of the model, along with a wide set of companion files.

The newly set T_{sim} is 300s, and this is indeed the timing horizon characterizing the simulations that are going to be showed in Paragraph 5.3.

Another option that could have been selected to decrease the dimensions of the critical LUTs was to define in a different manner the spatial discretization of the wind input (see Paragraph 2.2.3). In detail, instead of representing the wind input as a time sequence of 13x13 matrices covering an area of 145x145m² each, one could have chosen to decrease the amount of elements of the matrices, reducing the covered area and/or selecting a wider spatial discretization (e.g. 12x12, 11x11 and so on). By the way, this option implied a severe degradation of the wind turbine performances, therefore eventually it has been decided to reduce the timing horizon instead, as written above.

5.2.2 Compiling the Moorings block

The mooring subsystem has been the most challenging one to be compiled. The original physical system being represented in Simulink, indeed, contained six mooring lines (Figure 5.2.1), and each one of them has been modeled by means of three look-up-tables (for a total of 18 LUTs). In a single mooring line (Figure 5.2.2), the three LUTs were needed to compute the forces along that line in each of the three directions x, y, z (T_x, T_y, T_z); afterwards, the forces acting in the remaining three degrees of freedom rx, ry, rz (namely the three moments M_x, M_y, M_z) were calculated by means of a vectorial product between each force T_i and the vector C_{rot} representing the distance between the

center of the floating platform and the point in which that particular line is linked to the platform itself (i.e. the six vertices of the hexagon).

The choice of computing in such a manner the moments was due to the fact that the MAP++ library that has been used to this end (see Chapter two) only computes directly the linear forces. By the way, the architecture described so far had to be re-designed for the purpose of compiling.

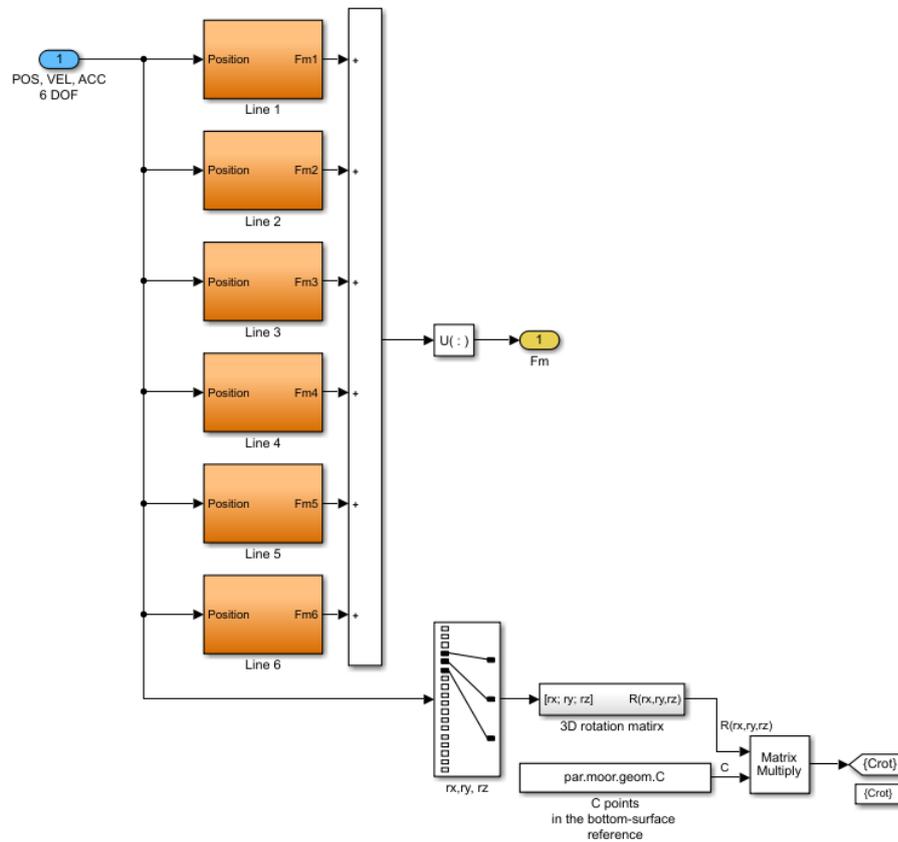


Figure 5.2.1: Original Simulink model of the moorings.

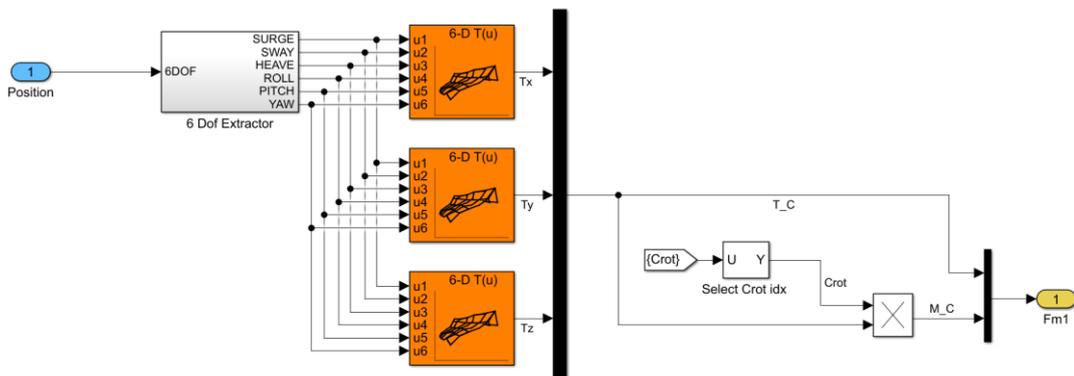


Figure 5.2.2: Original architecture of the Simulink model of one of the six mooring lines.

The first option that has been taken into account consisted in exploiting a Matlab Level-2 S-function. Such function is basically a Simulink block able to make a call to external libraries written in a programming language different from the Matlab/Simulink native language (indeed the MAP++ library is written in C++ language). By doing this, it was possible to avoid the LUTs (whose elements are indeed computed by MAP++) to be directly loaded in the compilation workspace, thus decreasing the computational effort being managed by the compiler.

Such approach encountered another problem though: the S-functions need, to be compiled, a companion *.tlc* file containing the piece of information required by the compiler to manage the call to the external libraries. This operation had to be performed by hand and it would have been too time-consuming for our purposes. For this reason, another method had to be found.

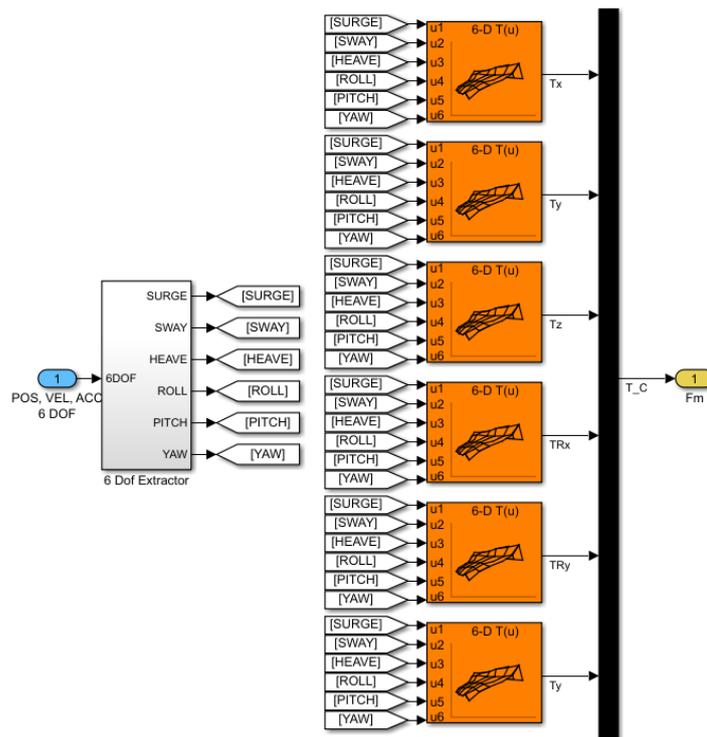


Figure 5.2.3: New version of the mooring Simulink model.

The solution that has eventually been implemented originated by the following reasoning. In the original model, the 6DOF generalized forces (T_x , T_y , T_z , M_x , M_y , M_z) obtained for each of the six lines are then summed up to get the vector F_m of the overall mooring reactions. By the way, to reduce the overall

number of exploited LUTs, one could bypass the contribution of the single lines and compute directly the summation of the generalized forces. This concept led to the architecture showed in Figure 5.2.3.

In this new design only 6 LUTs have been used, thus allowing the compilation to happen flawlessly. Each of the first three LUTs contain the summation of the LUTs of the six lines. By doing so, the three generalized forces acting on the overall platform (T_x , T_y , T_z) have been directly computed, without explicitly taking into account each mooring line. Similarly, the remaining three LUTs are employed to directly obtain the overall moments M_x , M_y , M_z . These tables are computed through a suitable script which takes into account, for each line, the vectorial product described previously.

After this procedure, the mooring reactions obtained by means of the new model have been compared to the ones of the original model, and the two versions proved to be identical. Therefore, the new architecture has eventually undergone the build procedure, which was managed correctly by the compiler.

On the other hand, the Hull block has been compiled in a relatively easy way, only modifying the continuous states into discrete-time ones. The two so-obtained compiled *.so* models (Hull and Moorings) have been added to the VeriStand project, together with the one representing the Wind Turbine, described in the previous paragraph.

5.3 The overall wind turbine VeriStand project: simulation campaign

After having created the three *.so* sub-models, they can be imported in the VeriStand project representing the model of the *NREL5MW* wind turbine (presented in Chapter two) in its entirety. This step is, as usual, performed by means of the System Explorer tool (under *Targets*, *Controller*, *Simulation Models*, *Add a Simulation Model*), where it is also possible to customize several settings such as the initial state of the model (either running or paused), signals and parameters importing options, vector port specifications etc.

Subsequently, the I/O of each sub-model needed to be mapped with the aim of reproducing the architecture of the original turbine Simulink model. Again, this can be done through System Explorer, under the icon *Configure Mappings*. There, under the respective menu tree, the user can identify the sources and destinations to be mapped. To sum up, the following connections have to be created:

- The mooring forces F_m , output of the Moorings block, are fed as input of the Hull block;
- Similarly, the wind turbine forces F_{wt} acting on the platform are fed to the Hull block as inports;
- Last, the output of the Hull block (i.e. the 18x1 vector representing the 6DOF positions, velocities and accelerations) is fed back both to the Moorings block and to the Wind Turbine one.

In Figure 5.3.1, the overall structure (comprehensive of the three imported sub-models with mapped channels) is depicted as it is represented in the User Interface (UI) of the VeriStand project.

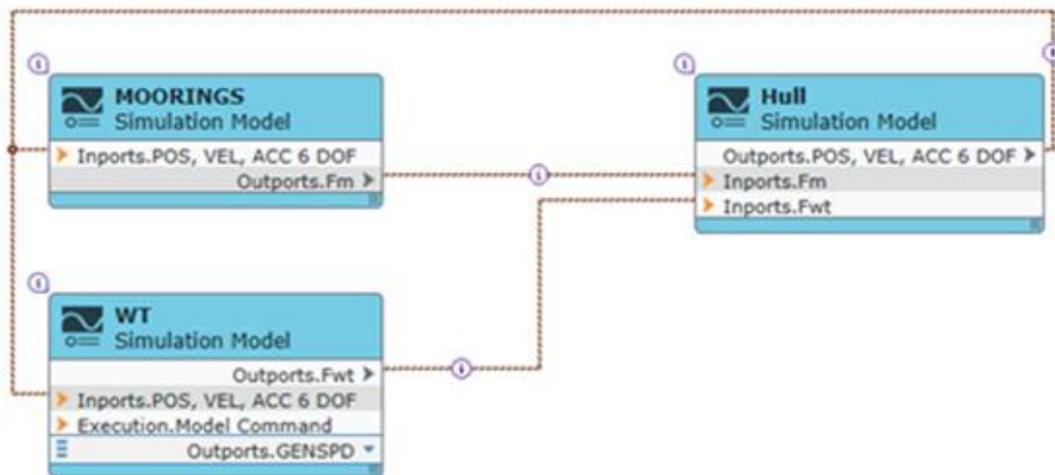


Figure 5.3.1: Overall structure of the NREL5MV wind turbine model, rearranged within VeriStand.

The last setting to be modified in the VeriStand project before the deploy, can be found under *System Explorer, Controller, Timing Source Settings, Target Rate*. This parameter specifies the update rate (i.e. the internal clock) of the controller. In our case the setting has to be consistent with the timing performances of the

wind turbine Simulink model (whose simulation step size is $\Delta t = 0.05\text{s}$). Therefore, the chosen value is in this case 20 Hz.

Then, it is eventually possible to perform the first simulations with the complete wind turbine system. In the following, some experimental tests that have been performed to validate the bench are reported and discussed.

The wind turbine model has been tested in three different scenarios, each one representing a different set of environmental conditions. In each case, the three subsystems have been compiled with reference to the respective workspace associated to that working condition (such workspace being generated, in turn, by means of the original Matlab main script). The three triads that have been selected (see Paragraph 2.2.4) are the following:

- 1) $H_s = 0.75\text{ m}$, $T_p = 5.5\text{ s}$, $V_0 = 8.5\text{ m/s}$;
- 2) $H_s = 0.75\text{ m}$, $T_p = 5.5\text{ s}$, $V_0 = 4.5\text{ m/s}$;
- 3) $H_s = 2.25\text{ m}$, $T_p = 6.5\text{ s}$, $V_0 = 15.5\text{ m/s}$.

In such simulations, the time horizon is $T_{sim} = 300\text{ s}$; in each case, the VeriStand model output is compared to its Simulink counterpart with the aim to validate the whole project building.

5.3.1 First simulation

The first simulated environmental scenario is the same that has been used in all the simulation showed in this dissertation so far. The expected model behavior has been already showed in Chapter two.

The wind turbine generator speed output has been mapped within the VeriStand project to the target velocity of the motor. By doing so, when the generator will be implemented in the test bench structure, it will be fed with a rotating speed consistent with the one that the wind turbine would produce in that particular scenario.

In the following, to check whether the procedure has been applied properly, the graphs obtained by capturing the VeriStand workspace are depicted. The first (Figure 5.3.2) represents the PTO (power take-off) of the simulated turbine, with the generator speed (rpm vs s) and the generated

electrical power (MW vs s). The second, instead, represents the comparison between the output signal and the corresponding physical behavior of the motor.

From the last figure it is possible to notice that the rotational velocity of the test bench motor follows in a satisfactory manner the output computed by the model. For this reason, in the following simulations the same graph will not be reported.

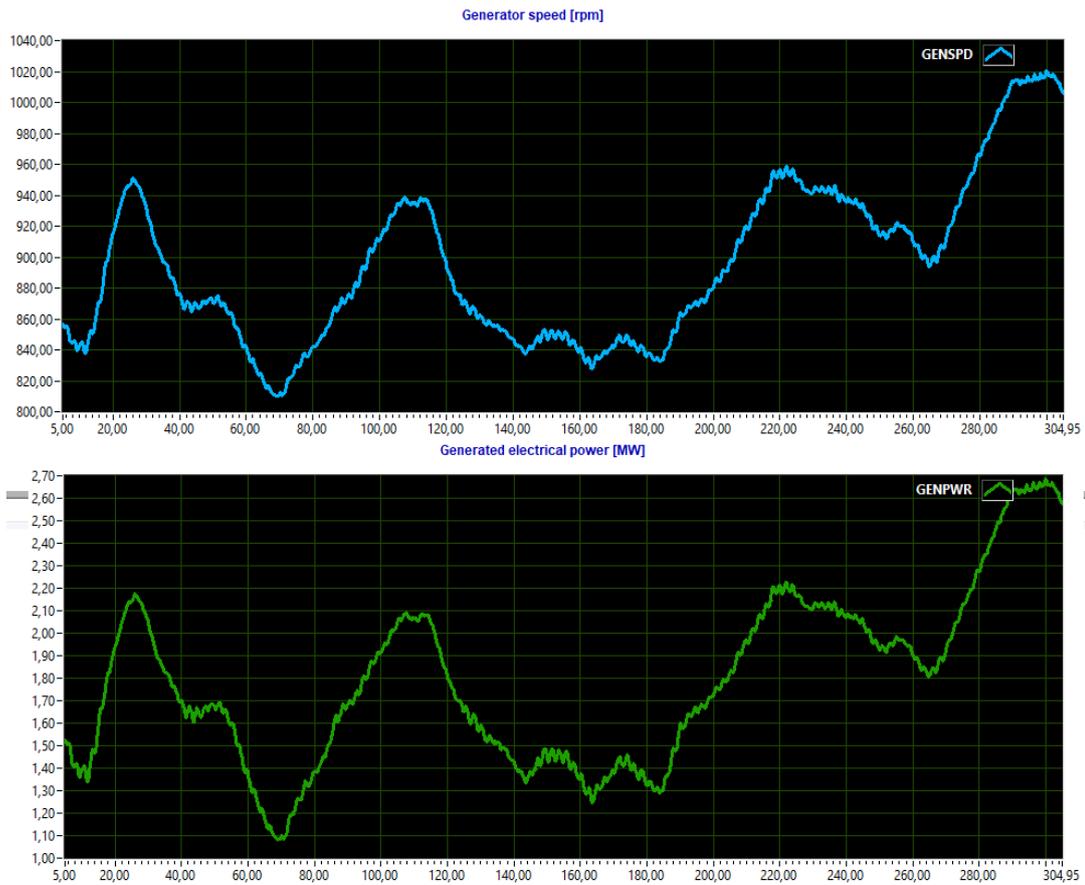


Figure 5.3.2: Simulation 1. VeriStand PTO: generator speed [rpm vs s] and generated power [MW vs s].

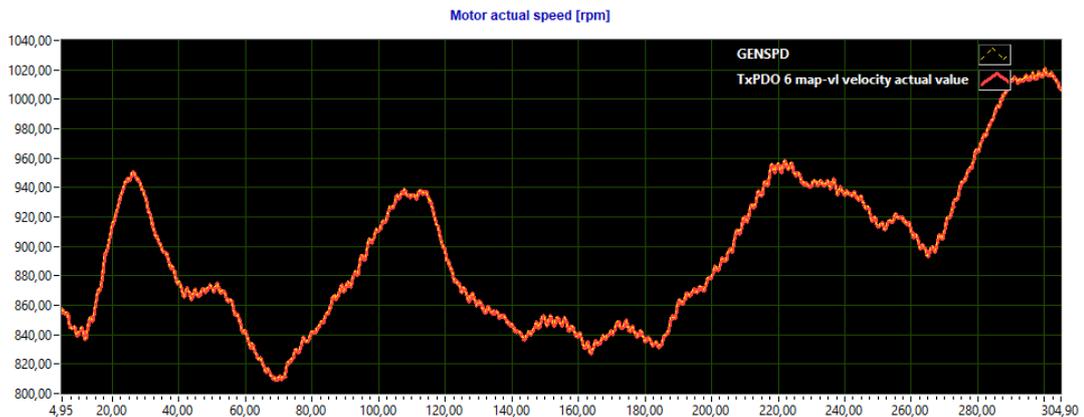


Figure 5.3.3: Simulation 1. Output speed (yellow) compared to the actual motor speed (red) [rpm vs s].

5.3.2 Second simulation

The second simulation represented a slightly different scenario, with the same average wave height and period, but with a smaller average wind velocity. The following figures represent, respectively, the expected PTO behavior from Simulink (Figure 5.3.4) and the obtained VeriStand PTO (Figure 5.3.5).

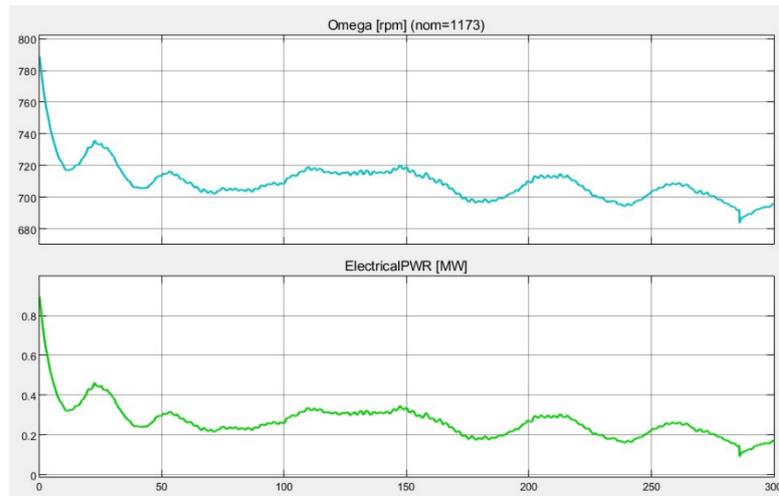


Figure 5.3.4: Simulation 2. Expected behavior from Simulink. PTO: generator speed [rpm vs s] and generated power [MW vs s].

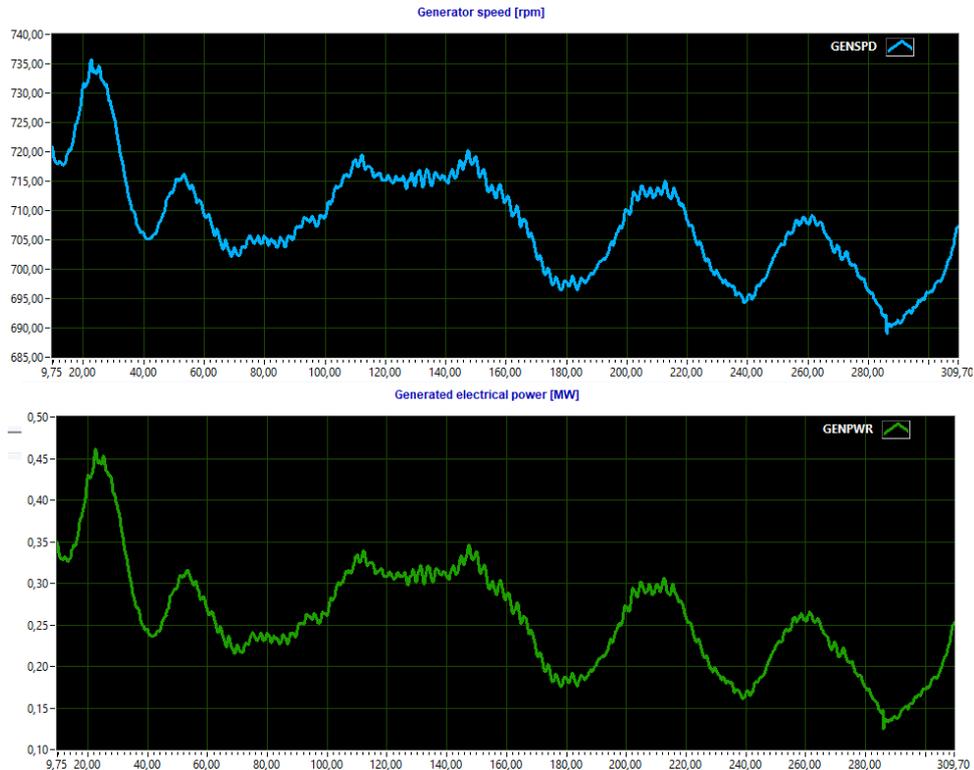


Figure 5.3.5: Simulation 2. VeriStand PTO: generator speed [rpm vs s] and generated power [MW vs s].

5.3.3 Third simulation

For the third simulation, a highly demanding working condition has been chosen. Indeed, in this scenario (that is typical of floating offshore wind), due to the relatively strong winds the wind turbine rated velocity ($\omega_{nom} = 1173.7$ rpm) happens to be often overcome by the actual generator speed. As described thoroughly in Paragraph 2.2.5, in this scenario the power take-off reaches the rated power ($T_{nom} = 5$ MW) and when this threshold is crossed, the blade-pitch controller is activated to decrease the generator torque and therefore limit the generated power to the nominal value.

In Figure 5.3.6 the PTO of the Simulink version is depicted. Here, also the torque behavior is showed to better underline the effect of the blade-pitch controller activation. Then, Figure 5.3.7 and Figure 5.3.8 display respectively the VeriStand PTO and the velocity response of the test bench motor with respect to the target generator velocity. In the latter image, it is possible to notice that even in this scenario the motor follows the target in a satisfactory manner.

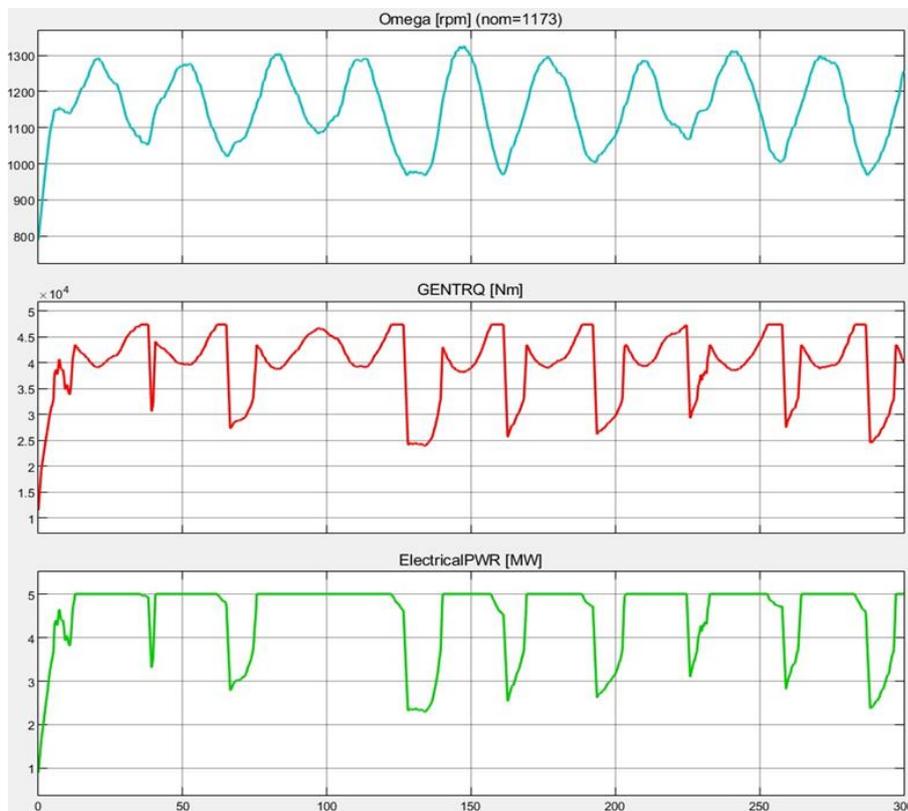


Figure 5.3.6: Simulation 3. Expected behavior from Simulink. PTO: generator speed [rpm vs s], generator torque [Nm vs s] and generated power [MW vs s].

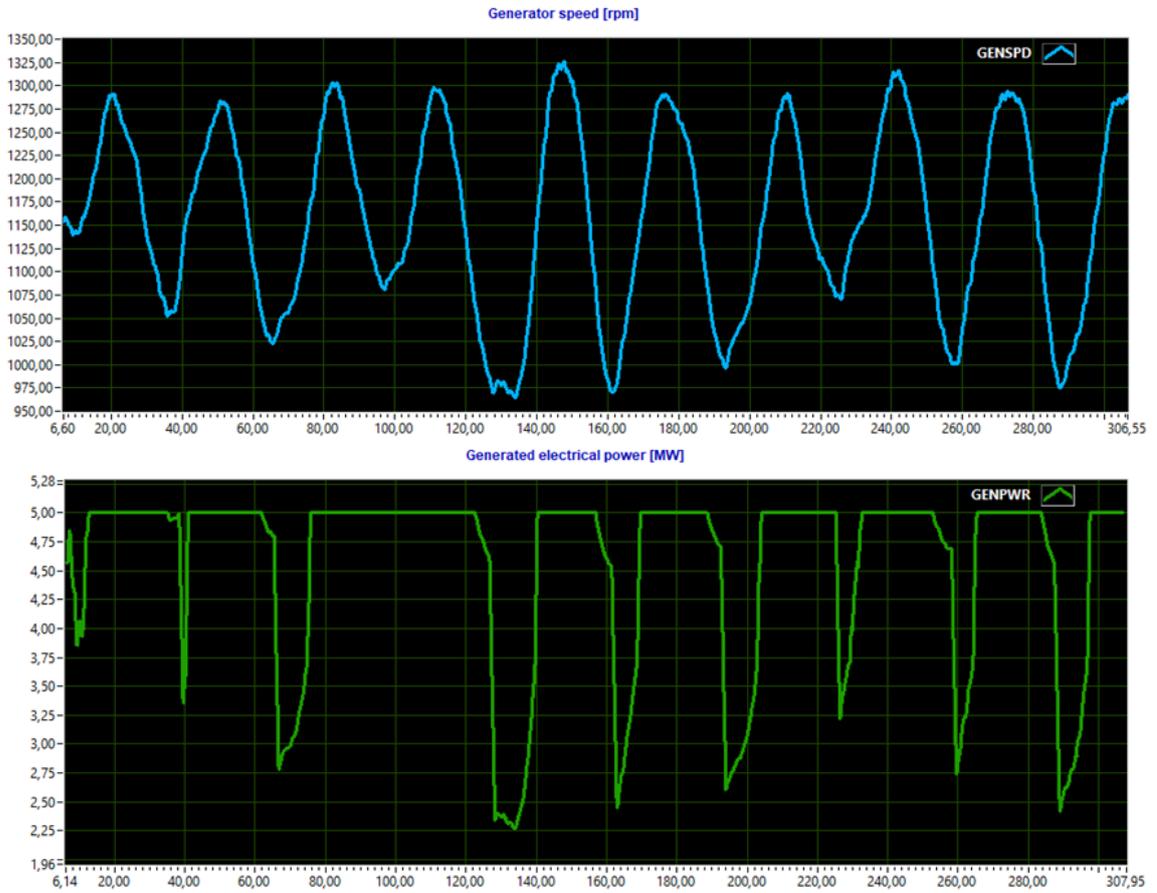


Figure 5.3.7: Simulation 3. VeriStand PTO: generator speed [rpm vs s] and generated power [MW vs s].

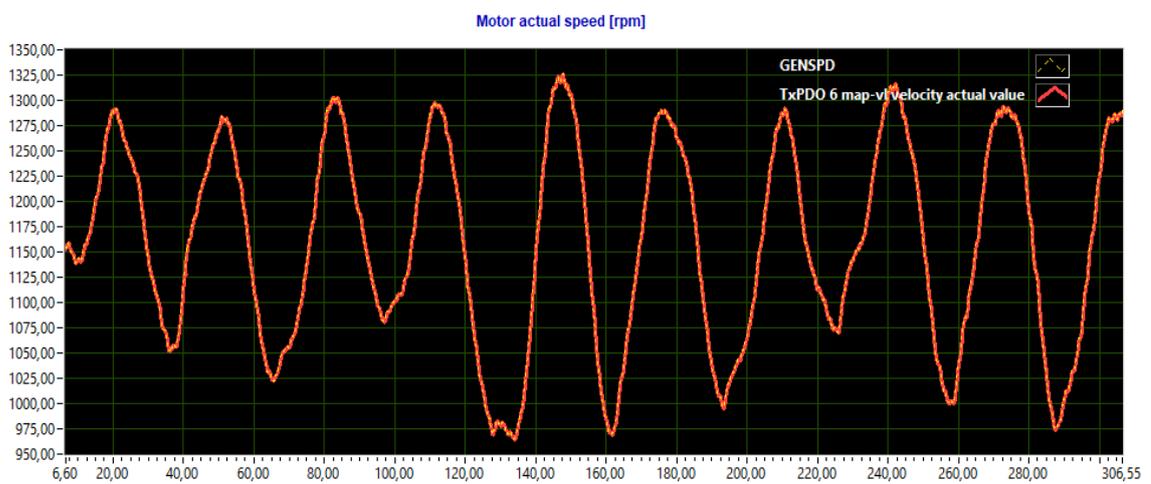


Figure 5.3.8: Simulation 3. Output speed (yellow) compared to actual motor speed (red) [rpm vs s].

Chapter six

Conclusions

The thesis work described throughout the present dissertation consisted, in a nutshell, in laying the foundation of a mechatronic test bench for Hardware-In-the-Loop (HIL) simulations for floating offshore wind turbines. The test rig has been designed, developed and built starting from scratch, and the work has been carried on during a period of eight months.

The HIL architecture has been characterized completely, defining the hardware plant (composed by the electrical motor and its drive, the transmission system and the generator) which represents the mechanical plant of the simulated turbine *NREL5MW*. The selected controller for the loop is a CompactRIO-9040 by National Instruments, with a current input module (NI-9253) to read during run time the current feedback from the torque meter mounted on the shaft.

The cRIO-9040 (which is the master of an EtherCAT network that allows the real-time communication with the motor drive), is programmed by means of the NI VeriStand software. The VeriStand project deployed on the controller contains a model which is composed in turn by three sub-systems (wind turbine, hull and moorings). They have been compiled from a Simulink model designed in previous works and representing the dynamic behavior of the physical overall system. Thanks to this hardware/software architecture, the tester can import within the VeriStand project several sets of realistic environmental conditions (identified by a triad of average wind speed, wave height and wave period) in order to test the control logic of the turbine on the hardware plant.

At the time of writing, despite the rig is still not complete, the first tests have been already carried out. First, the motor proportional-integrative-derivative (PID) controller integrated within the drive has been tuned in order to guarantee the best response in terms of timing performance and accuracy. Then,

the motor behavior has been tested with a set of standardized time sequences (i.e. step, square wave, sine wave) and in several frequency scenarios, in order to check the precision of its response.

Moreover, for what concerns the VeriStand model of the wind turbine, it has been tested in three different environmental conditions. In each case, the obtained behavior has been compared to the one of its Simulink counterpart, to validate the effectiveness of the building project. These last simulations have demonstrated that, indeed, the model has been built flawlessly. Also, the simulated generator speed of the models has been fed to the drive as target velocity of the motor, and the latter responded as expected to the solicitations, thus simulating in real time the actual rotation of the wind turbine blades.

Therefore, ultimately, the test bench proved to be a powerful tool to be used during the design workflow of a floating offshore wind turbine (especially for testing and debugging purposes), with the final goal of a deployment in the Pantelleria territorial waters.

6.1 Future works and further developments

Despite the steps ahead towards the final purpose, some aspects of the work still have to be fulfilled and they will be object of future research. From a constructive point of view, the chosen electrical generator of the test bench has still not been inserted in the architecture, together with its drive which will need to be added to the EtherCAT network in order to communicate with the controller. After having added the generator to the test bench, it will represent the power take-off (PTO) of the wind turbine, and the generated electrical power will be accounted for as a critical part of the acquisitions.

Also, aside from the generator, also the transmission subsystem (composed by the shafts, the flanges and the torque meter) still needs to be mounted due to some technical manufacturing issues that delayed the procedure. The implemented torque meter will be connected with a signal cable to the NI-9253 current input module of the cRIO, thus providing to the controller an additional feedback regarding the rotational behavior of the mechanical components of the bench.

For what concerns the functioning, the test rig has been validated in this thesis work only with respect to the components that have already been implemented physically on the bench. Hence, as the mounting flow will progress, more simulations will be needed for validating both the generator and the torque meter. After that, the test bench will be completed, and eventually a proper suitable simulation campaign will be launched.

Another aspect that needs further analysis is the procedure for importing the models from Simulink to VeriStand. Indeed, the three sub-models (wind turbine, hull, moorings) when compiled are automatically linked to a Matlab workspace that represents an arbitrary environmental condition. This means that the sub-models need to be recompiled with respect to a different workspace every time that the turbine needs to be simulated in a different working condition. As a consequence, the compiling flow becomes time consuming, and it lacks in flexibility.

For this reason, in future developments such issue will be tackled, either automatizing the compiling process, or even better creating an *.so* model of the *NREL5MW* wind turbine that is not associated to any workspace. In this last case, the test bench user would be allowed to customize the working and environmental conditions directly from within the VeriStand project, from a dedicated interface, thus being able to tune arbitrarily the simulated scenario according to his or her particular interests, also during runtime.

Eventually, the last element that is still to be refined is the EtherCAT communication between controller and motor drive. Indeed, although the real-time data transmission works, due to technical issues regarding the mapping implemented by the XML file that has been used to interface cRIO and ACS880, the number of parameters that can be accessed by means of third-party hardware and software is limited. Modifying the XML text, either manually or with an intended software, in future works it will be possible to overcome this limitation, thus enlarging the potential and flexibility of the developed test bench.

Bibliography

- [1] United Nations, *“The World Population Prospects 2019: Highlights”*, 2019. Available online at: <https://population.un.org/wpp/>

- [2] International Energy Agency, *“Global CO2 emissions in 2019”*, 2020. Available online at: <https://www.iea.org/articles/global-co2-emissions-in-2019>

- [3] H Ritchie, M Roser, *“Fossil Fuels”*, 2020. Available online at: <https://ourworldindata.org/fossil-fuels>

- [4] International Energy Agency, *“Offshore Wind Outlook 2019”*, 2019. Available online at: <https://www.iea.org/reports/offshore-wind-outlook-2019>

- [5] World Forum Offshore Wind, *“Global Offshore Wind Report: 1st half 2020”*, 2020. Available online at: https://wfo-global.org/wp-content/uploads/2020/08/WFO_Global-Offshore-Wind-Report-HY1-2020.pdf

- [6] WindEurope, *“Offshore wind in Europe. Key trends and statistics 2019”*, 2020. Available online at: <https://windeurope.org/about-wind/statistics/offshore/european-offshore-wind-industry-key-trends-statistics-2019/>

- [7] Global Wind Energy Council, *“Global Wind Report 2018”*, 2019. Available online at: <https://gwec.net/wp-content/uploads/2019/04/GWEC-Global-Wind-Report-2018.pdf>

- [8] International Renewable Energy Agency, “*Floating foundations: a game changer for offshore wind power*”, 2016. Available online at: https://www.irena.org/-/media/Files/IRENA/Agency/Publication/2016/IRENA_Offshore_Wind_Floating_Foundations_2016.pdf
- [9] H Bailey, P M Thompson, K L Brookes, “*Assessing Environmental Impacts of Offshore Wind Farms: Lessons Learned and Recommendations for the Future*”, 2014. Available online at: https://www.researchgate.net/publication/266086383_Assessing_Environmental_Impacts_of_Offshore_Wind_Farms_Lessons_Learned_and_Recommendations_for_the_Future
- [10] U.S. Department of Energy, “*Wind Vision: A New Era for Wind Power in the United States*”, 2015. Available online at: https://www.energy.gov/sites/prod/files/WindVision_Report_final.pdf
- [11] Equinor, “*Equinor and ORE Catapult collaborating to share Hywind Scotland operational data*”, 2019. Available online at: <https://www.equinor.com/en/news/2019-11-28-hywind-scotland-data.html>
- [12] Wind Europe, “*Floating Offshore Wind Vision Statement*”, 2017. Available online at: <https://windeurope.org/wp-content/uploads/files/about-wind/reports/Floating-offshore-statement.pdf>
- [13] Wind Europe, “*Floating Offshore Wind Energy. A policy blueprint for Europe*”, 2019. Available online at: <https://windeurope.org/wp-content/uploads/files/policy/position-papers/Floating-offshore-wind-energy-a-policy-blueprint-for-Europe.pdf>

- [14] Energydata.info, “*Global Wind Atlas*”, 2020. Available online at: <https://globalwindatlas.info/>
- [15] D Pantusa, G R Tomasicchio, “*Large-scale offshore wind production in the Mediterranean Sea*”, Cogent Engineering, 2019.
- [16] Terna, “*Lo storico dei dati statistici sull’energia elettrica e l’ultimo bilancio elettrico*”, 2019. Available online at: <https://www.terna.it/it/sistema-elettrico/statistiche/pubblicazioni-statistiche>
- [17] Ricerca Sistema Energetico, “*Atlante Eolico Interattivo*”, 2020. Available online at: <http://atlanteeolico.rse-web.it>
- [18] M Wing Goodale, A Milman, “*Assessing the cumulative exposure of wildlife to offshore wind energy development*”, Journal of Environmental Management, Elsevier, 2019.
- [19] L Cottura, “*Dynamic modeling of an offshore floating wind turbine for application in the Mediterranean Sea*”, PhD Thesis, Politecnico di Torino, Italy, 2020.
- [20] R Caradonna, “*Development and Validation of an Aero-Hydrodynamic Model for a Floating Offshore Wind Turbine*”, PhD Thesis, Politecnico di Torino, Italy, 2020.
- [21] National Renewable Energy Laboratory, “*Definition of a 5-MW Reference Wind Turbine for Offshore System Development*”, Technical Report NREL/TP-500-38060, 2009.
- [22] J Journée and W Massie, “*Offshore Hydromechanics*”, Delft University of Technology, 2001. Available online at: <https://ocw.>

tudelft.nl/wp-content/uploads/OffshoreHydromechanics_Journee_Massie.pdf

- [23] B J Jonkman, L Kilcher, “*Turbsim user’s guide: version 1.06.00*”, NREL, 2012. Available online at: <https://www.nrel.gov/wind/nwtc/assets/pdfs/turbsim.pdf>

- [24] ECMWF, ERA5 dataset. Available online at: <https://www.ecmwf.int/en/forecasts/datasets/reanalysis-datasets/era5>

- [25] H Glauert, “*Airplane Propellers*”. In: “*Aerodynamic Theory*”. Springer, Berlin, Heidelberg, 1935. DOI: https://doi.org/10.1007/978-3-642-91487-4_3

- [26] D Spera, “*Wind Turbine Technology: Fundamental Concepts in Wind Turbine Engineering, Second Edition*”. ASME Press, New York, 1994. DOI: <https://doi.org/10.1115/1.802601>

- [27] R Ning, G Hayman, J Jonkman, “*Development and Validation of a New Blade Element Momentum Skewed-Wake Model within AeroDyn*”. NREL, Brigham Young University, 2015. Available online at: <https://www.nrel.gov/docs/fy15osti/63217.pdf>

- [28] F Wendt (NREL), A Robertson (NREL), J Jonkman (NREL), M T Andersen (Aalborg University), “*Verification and Validation of the New Dynamic Mooring Modules Available in FAST v8*”. Presented at the Twenty-sixth International Ocean and Polar Engineering Conference (ISOPE). Rhodes, Greece, June 26–July 2, 2016. Available online at: <https://www.nrel.gov/docs/fy16osti/65822.pdf>

- [29] M Masciola, “*MAP++ Documentation*”. 2008.

- [30] W E Cummins, “*The impulse response function and ship motions*”. Navy Department, David Taylor Model Basin, 1962.
- [31] ANSYS, Inc., “*Aqwa Theory Manual*”, Release 15.0, 2013.
- [32] N Faedo, Y Peña-Sanchez, J V Ringwood, “*Finite-Order Hydrodynamic Model Determination Using Moment-Matching*”. Ocean Engineering, volume 163, 2018. Available online at: <http://www.eeng.nuim.ie/jringwood/Respubs/J300FOAMM.pdf>
- [33] M Basic, “*On hardware-in-the-loop simulation*”, Proceedings of the 44th IEEE Conference on Decision and Control, Seville, Spain, 2005.
- [34] HBK, “*T22 datasheet*”. Available online at: <http://www.shr-etest.com/upload/product/201812/11/201812111107444266.pdf> (retrieved: 11/02/21)
- [35] NI official website. <https://www.ni.com/en-us/shop/hardware/products/compactrio-controller.html?modelId=229876> (retrieved: 13/01/21)
- [36] NI, cRIO 9040 datasheet. Available online at: https://www.ni.com/pdf/manuals/377158f_02.pdf (retrieved: 13/01/21)
- [37] NI, “*Understanding NI CompactRIO Scan Mode*”, updated Oct 2, 2020. Available online at: <https://www.ni.com/en-us/innovations/white-papers/08/understanding-ni-compactrio-scan-mode.html> (retrieved: 13/01/21)
- [38] NI, NI 9253 datasheet. Available online at: https://www.ni.com/pdf/manuals/378060a_02.pdf (retrieved: 13/01/21)

- [39] NI official website. <https://www.ni.com/en-us/shop/data-acquisition-and-control/application-software-for-data-acquisition-and-control-category/what-is-veristand.html> (retrieved: 13/01/21)
- [40] EtherCAT Technology Group, “EtherCAT - the Ethernet Fieldbus”. Commercial brochure, March 2020. Available online at: www.ethercat.org/download/documents/ETG_Brochure_EN.pdf (retrieved: 21/01/21)
- [41] ABB, “*FECA-01 EtherCAT adapter module – User’s manual*”. Available online at: <https://search.abb.com/library/Download.aspx?DocumentID=3AUA0000068940&LanguageCode=en&DocumentPartId=1&Action=Launch> (retrieved: 22/01/21)
- [42] Beckhoff, “*EtherCAT slave process data settings (PDO)*”. Available online at: <https://infosys.beckhoff.com/english.php?content=../content/1033/ep3356/1652713483.html> (retrieved: 05/04/21)
- [43] M Lellić, Z Gajic, “*A reference guide to PID controllers in the Nineties*”. IFAC Proceedings Volumes, Volume 33, Issue 4, 2000. Available online at: <https://www.sciencedirect.com/science/article/pii/S147466701738223X> (retrieved: 05/04/21)
- [44] G Ellis, “*Control System Design Guide (Fourth Edition)*”. Butterworth-Heinemann/Elsevier, 2012. DOI: <https://doi.org/10.1016/C2010-0-65994-3>
- [45] NI, “*VeriStand 2020 R3 Manual*”. Available online at: <https://www.ni.com/documentation/en/veristand/20.3/manual/manual-overview/> (retrieved: 14/01/21)

- [46] NI, “*VeriStand 2020 R3 Manual: Choosing Compiler Tools for a Model*”. Available online at: <https://www.ni.com/documentation/en/veristand/20.3/manual/select-compiler/> (retrieved: 14/01/21)
- [47] NI official website. <https://www.ni.com/it-it/support/downloads/software-products/download.c-c---development-tools.html#333407>
- [48] NI, “*VeriStand 2020 R3 Manual: Compiling a Model in MathWorks Simulink® Software*”. Available online at: <https://www.ni.com/documentation/en/veristand/latest/manual/compile-model-simulink/> (retrieved: 15/01/21)

Ringraziamenti

Sento il bisogno di concludere questa tesi (e questo enorme capitolo della mia vita) dedicando qualche riga alle persone che, in un modo o nell'altro, sono state fondamentali nel mio percorso.

Innanzitutto, grazie a Gabriele (aka Gino), con cui ho condiviso questo lavoro dall'inizio alla fine. Senza la tua serietà e il tuo impegno, ma soprattutto senza la tua ironia, certe giornate sarebbero state probabilmente troppo difficili per me. Sei una "brava persona".

Grazie alla dottoressa Peresso, perché tra il fare il proprio mestiere e il farlo con l'anima c'è di mezzo un mare, e ora mi sembra di avere gli strumenti per attraversare il mio e vedere com'è la vita dall'altra parte.

Grazie ai miei amici. Non mi lancerò in stucchevoli elenchi: chi c'è sa. Non sono mai stato bravo a rispondere puntualmente ai messaggi, ad essere presente, o a chiedere aiuto quando mi serviva davvero, e conoscete a memoria i miei mille altri difetti. Ma ci tengo a ringraziarvi perché comunque ci siete, e perché mi ricordate sempre che certe cose nemmeno il tempo le può cambiare davvero. Vi voglio bene.

Grazie a mamma e papà. Grazie per tutti i sacrifici che fate, la cui grandezza e profondità riesco appena ad intuire. Sono fiero di chiudere il mio percorso accademico e regalarvi un orgoglio tanto grande; io, dal canto mio, sono orgoglioso di avere genitori così. So che mi supportereste qualunque cosa decidessi di fare del mio futuro, ma spero soprattutto che possiate continuare ad essere fieri di me vedendomi fiorire e mettere il cuore in ciò che amo fare.

Grazie al resto della mia famiglia. A mio fratello, perché siamo simili e diversissimi, e gli voglio un bene che non sono mai riuscito a spiegargli. Ai nonni: sono incredibilmente fortunato e grato di poter condividere con voi questo traguardo. A nonno Pippo: non so bene dove, se nel cielo o nella terra, se fuori o dentro di me, ma so che ci sei anche tu. Agli zii e ai cugini, a Roma e a Pizzo.

Infine, grazie ad Elena. Grazie per esserci sempre stata, per avermi infuso coraggio in tutti i lunedì blu ed avermi aiutato a godere del Sole nei martedì dorati. Con me ti sei messa in gioco con una sincerità e una forza che ammiro e da cui prendo spunto tutti i giorni. Diventare grandi è un viaggio assurdo, stupendo e spaventoso; con te è molto più bello e molto meno difficile.

Marco