POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Modelling and Control Tuning of an Airborne Wind Energy System with On-Board Generation

Tesi di Laurea Magistrale in
Control and Automation Engineering

Author: **Giacomo Calciolari**

Student ID: 946636
Advisor: Prof. Lorenzo Mario Fagiano
Academic Year: 2020-21

Abstract

In the last decade, the first consequences of climate change became evident worldwide, bringing into the open the need of a dramatic change into an energy market that relies too much on polluting sources like fossil fuels. Thereby, the scientific community started to investigate the renewable sector, to find a greener alternative to coal and oil.
Conceived in 1970s, Airborne Wind Energy (AWE) systems represent an innovative technology for wind energy conversion, not yet fully developed. These new-generation wind power plants use autonomous tethered kites to drive electric generators, exploiting the lift force developed by the kite when flying in crosswind at high-altitudes (500-800 m), where winds are stronger and more reliable with respect to the traditional turbines hub altitude. Several companies have been investing in AWE technology since the early 2000s, when the main AWE principles were reconsidered after the pioneering theoretical work of Miles L. Loyd in 1980. Among those companies there was the Californian Makani Power, that built the M600 system. The M600 system was designed to produce up to 600 kW of rated power and was the first ever AWE system able to complete an offshore flight.

This thesis aims to provide a new approach for the modelling and control of On-Board Generation (OBG) systems, which represent an AWE branch that has not been deeply investigated yet, to which the M600 belong. The model employs the dynamical equations of an octocopter rotor-craft, adding a procedure for the identification of the aerodynamic forces coefficients and an accurate tether model, in order to obtain realistic results. The proposed control system employs a cascade scheme with three nested loops and is tuned with a data-driven optimization procedure, based on the data available from the flight tests of the Makani OBG M600 that the company publicly distributed in 2020.
The final outcome of this work is an optimally-tuned simulator, able to emulate the working activities of the M600 OBG drone in the initial take-off phase and validated on the actual flight data.

Keywords: Airborne Wind Energy, AWE, Makani, Autonomous Drones

Sommario

Nell'ultimo decennio, le prime conseguenze del cambiamento climatico sono diventate evidenti in tutto il mondo, portando in primo piano la necessità di un drastico cambiamento in un mercato dell'energia che si basa troppo su fonti inquinanti come i combustibili fossili. Così, la comunità scientifica ha iniziato a studiare il settore delle rinnovabili, per trovare un'alternativa più verde al carbone e al petrolio.
Concepiti negli anni '70, i sistemi Airborne Wind Energy (AWE) rappresentano una tecnologia innovativa per la produzione di energia eolica, ma non ancora completamente sviluppata. Questi impianti eolici di nuova generazione utilizzano aquiloni autonomi per azionare generatori elettrici sfruttando la portanza sviluppata dai venti d'alta quota (500-800 m), più forti e affidabili rispetto a quelli presenti solitamente all'altitudine in cui operano le turbine tradizionali. Diverse aziende hanno investito nella tecnologia AWE fin dai primi anni 2000, quando i principi AWE sono stati riconsiderati dopo il pionieristico lavoro teorico di Miles L. Loyd nel 1980. Tra queste troviamo la californiana Makani Power, che costruì il sistema M600 con l'obiettivo di produrre fino a 600 kW di potenza nominale. Il sistema M600 è stato il primo sistema AWE in grado di completare un volo offshore.

Questa tesi mira a fornire un nuovo approccio per la modellazione e il controllo dei sistemi On-Board Generation (OBG), che rappresentano una branca AWE che non è stata ancora approfondita, a cui appartiene il sistema M600. Il modello utilizza le equazioni dinamiche di un octocopter rotor-craft, aggiungendo una procedura per l'identificazione dei coefficienti delle forze aerodinamiche e un modello di cavo accurato, al fine di ottenere risultati realistici. Il sistema di controllo proposto utilizza uno schema a cascata con tre loop innestati, i cui parametri sono impostati con una procedura di ottimizzazione basata sui dati raccolti durante i test di volo del Makani M600, resi disponibili dall'azienda nel 2020.
Il risultato finale di questo lavoro è un simulatore in grado di emulare il volo del drone M600 nella fase iniziale di decollo e convalidato sui dati di volo reali.

Parole chiave: Airborne Wind Energy, AWE, Makani, Energia Eolica, Droni Autonomi

Ringraziamenti

Alla fine di questo viaggio, desidero ringraziare il Professor Fagiano per il continuo supporto e per avermi dato la possibilità di sviluppare questa Tesi nell'ambito dell'energia rinnovabile, facendomi appassionare ad un settore innovativo con concetti incredibilmente stimolanti e che spero possa diventare parte del mio futuro.

Ringrazio dal profondo del mio cuore mio padre Alberto per i suoi saggi consigli, mia madre Carla per il costante incoraggiamento, mio fratello Davide per esserci sempre stato quando ho avuto bisogno di lui, mia cugina Letizia e mia zia Patrizia per il sostegno e la comprensione che mi hanno sempre dimostrato.

Ringrazio mia nonna Lia e mio nonno Nedo per i valori che mi hanno trasmesso e per avermi cresciuto come un figlio, insegnandomi l'amore per le cose semplici e per la Natura.

Infine ringrazio i miei amici, con cui ho condiviso tanto in questi anni fatti di gioie, sofferenze, esami e avventure indimenticabili, e che continuano a essere la pietra d'angolo della mia vita.

Contents

1 | Introduction

Energy security has always been a key topic for any country in the world and, with the unceasing advance of climate change and its effects becoming more and more evident, the spotlights are on the renewable sources like wind, solar and biofuels, in the attempt to replace (or at least reduce) the employment of fossil fuels and consequently decrease the CO_2 emissions. This intention was formally stated by the United Nations in the Agreements of Paris in 2015, as the joint goal of the UN members of reaching the net-zero greenhouse gas (GHG) emissions by 2050.

A strong position was taken also by the European Union that, on July 21^{th} 2021, set a legally binding target of net zero greenhouse gas emissions by 2050 with the European Climate Law [1]: the EU Institutions and the Member States are now bound to take the necessary measures at European and national level to meet the target.

The European Climate Law writes into law the goal set out in the *European Green Deal* [2] to become climate-neutral by 2050. The law also establishes the intermediate target of reducing net greenhouse gas emissions by at least 55 % by 2030, compared to 1990 levels, and aims to ensure that all EU policies contribute to this goal and that all sectors of the economy and society play their part by increasing the efforts toward the innovation in the energy field, especially into the expansion of the renewable sector.

In the net zero pathway, according to the International Energy Association [3], the energy sector is indeed based largely on renewable energy: two-thirds of total energy supply in 2050 is expected to be from wind, solar, bioenergy, geothermal and hydro energy. Solar photovoltaic (PV) capacity increases 20-fold between now and 2050, and wind power 11-fold, with almost 90 % of global electricity generation that comes from renewable sources, against the actual 29 % produced worldwide. Wind and solar PV together will account for 90 % of renewable electricity generation so, the annual capacity additions of wind and solar between 2020 and 2050 have to be five-times higher than the average over the last three years. In other words, the energy mix is going to change, mainly by decreasing the dependence on fossils fuels with a bigger share of the market occupied by the renewables and by a large increase in energy supply from nuclear power, which nearly doubles between 2020 and 2050.

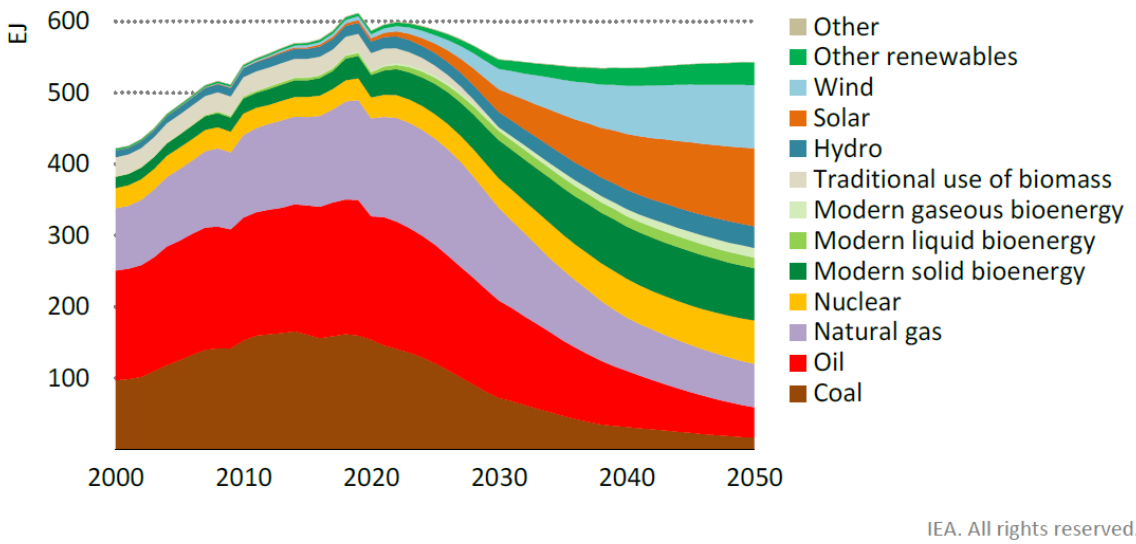


Figure 1.1: Total energy supply according to the Net Zero Emission scenario of IEA ([3])

The transformation of the electricity sector is central to achieve net-zero emissions in 2050. Today, electricity generation is the single largest source of energy-related CO_2 emissions, accounting for 36 % of the total: CO_2 emissions from electricity generation worldwide totalled 12.3 Gt in 2020, of which 9.1 Gt was from coal-fired generation, 2.7 Gt from gas-fired plants and 0.6 Gt from oil-fired plants.
Following the Net-Zero Scenario (NZE), renewables contribute most to decarbonising electricity: solar PV and wind will become the primary sources of electricity globally before 2030, each one generating over 23000 TWh by 2050, equivalent to about 90 % of all electricity produced in the world in 2020. Clearly, pairing advanced battery storage systems with solar PV and wind becomes a key factor to improve the power systems flexibility and maintain electricity security.
For what concerns hydropower, today it is the largest low-carbon source of electricity and steadily grows, doubling by 2050 in the NZE scenario proposed in [3]. Also the generation using bioenergy (in dedicated plants and in form of biomethane delivered through gas networks) doubles to 2030 and increases nearly fivefold by 2050.
The transition to net-zero emissions by 2050 requires a substantial ramp up in the investment on electricity generation: in NZE scenario, the annual investment increases from about USD 0.5 trillion over the past five years to USD 1.6 trillion in 2030 and, by 2030, the annual investment in renewables in the electricity sector will be around USD 1.3 trillion, slightly more than the highest level ever spent on fossil fuel supply (USD 1.2 trillion in 2014). This huge expenditure is mainly to build a new generation of infrastructures for low-emissions electricity and to finance the technology advance of renewable energy.

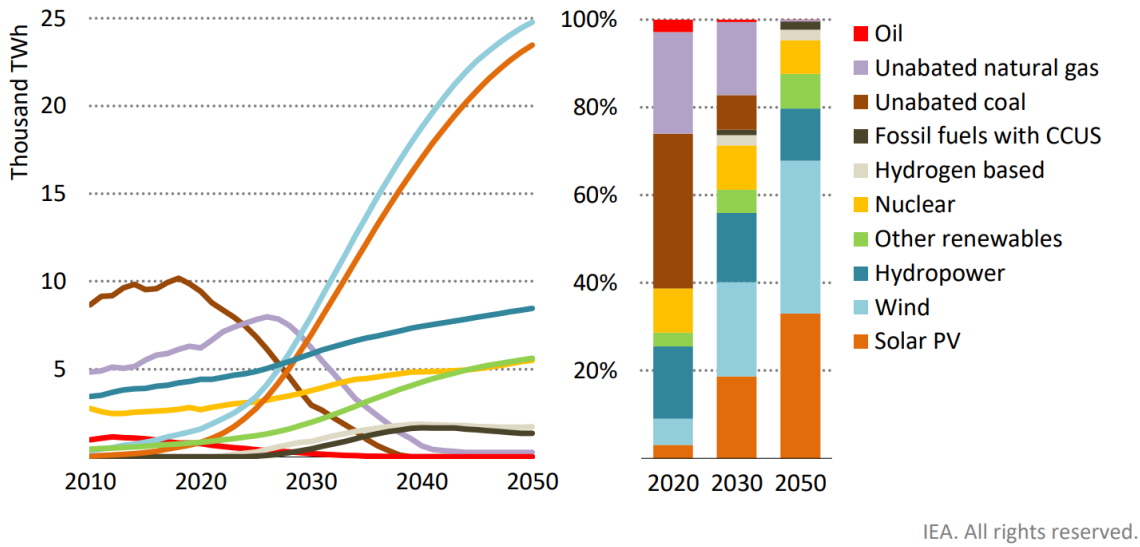


Figure 1.2: Global electricity generation by source (from [3])

After 2030, the forecasts report that annual electricity generation investment will fall by one-third to 2050, since the cost of renewables will continue to decline (as already happening today) once that the infrastructure is set up.
However, actions made to date fall far short of what is required by the Net-Zero pathway, but the number of countries that have pledged to achieve net-zero emissions has grown rapidly over the last year and now covers around 70 % of global emissions of CO_2.
A country that is concretely acting to achieve climate neutrality is Germany, which set its deadline for the net zero by 2045 and aims to obtain a 65 % national emissions reduction below 1990 levels by 2030. The total investment volume estimate is of €6 trillions, *i.e.* an average annual investment of roughly €240 billions through 2045 (about 7% of Germany's Gross Domestic Product) will be made to fight the climate emergency.
As a consequence, in January 2021, Germany's *Renewable Energy Act (EEG) 2021* came into effect: the plan includes raising the capacity for wind power from 3 *GW* per year to 4 *GW* per year and for solar photovoltaics from 4 *GW* to 6 *GW*, bringing solar PV capacity to 100 *GW* (from 56 *GW*), onshore wind to 71 *GW* (from 55 *GW*), offshore wind to 20 *GW* and biomass to 8.4 *GW* by 2030.

Generally speaking, wind and solar capacity more than doubled between 2015 and 2020, increasing by around 800 *GW*, *i.e.* to an average annual increase of 18 %, accordingly to the bp Statistical Review of World Energy 2021 ([4]). Referring to 2020, is necessary to underline the strong increase of the energy produced from wind power in Europe, with the annual growth rate that reached 10.6 %.
Indeed, wind energy is experiencing an impressive global growth, confirming to be a

promising player in the future energy mix: only in 2020, the installation of new plants brought the total power capacity to 743 GW, a growth of 14 % with respect to 2019, as we can read in the annual report of the Global Wind Energy Council (GWEC, see [5]).

Together with the technology advance, the Levelized Cost Of Energy (LCOE) is a key variable in the spreading of wind power: we define as LCOE the average revenue per unit of generated electricity that would be required to recover the costs of building and operating a generating plant during an assumed financial life.
The LCOE associated to wind energy is significantly decreasing and we expect the average LCOE of onshore wind to continue declining by 25% while offshore wind LCOE will shrink 55% with respect to the 2018 level, reaching the value of $54 USD/MWh$ in 2030, according to the Global Renewables Outlook [6] by the International Renewable Energy Agency (IRENA). These are important achievements for the global goal of carbon neutrality and show how much potential wind energy has.

1.1. Wind Turbines

The dominant technology for wind power are the Horizontal Axis Wind Turbines (HAWT), since they represent the most established way to harvest energy from the wind.

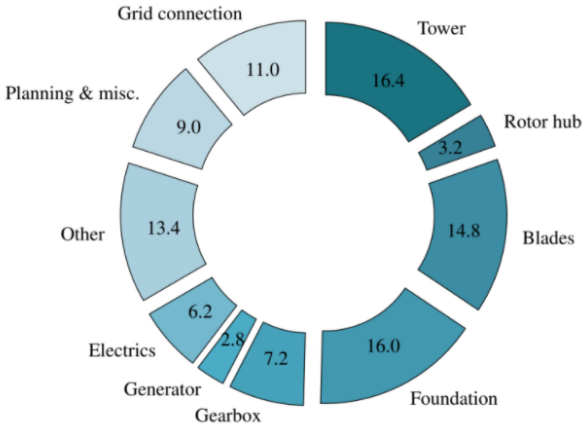


Figure 1.3: Breakdown of the capital cost of wind turbines in percent ([7])

Nevertheless this kind of power generators are well known since almost 200 years, there are some drawbacks inherent to the nature of HAWTs:

- The installation cost of a wind turbine is still high, mainly due to the amount of materials needed for its building rather than for what concerns the power generation elements (such as the electrical components or the generator itself), as shown in Figure 1.3. Offshore plants, that are the most efficient ones since the advanta-

geous wind conditions available in open sea, require complex and expensive floating foundations to sustain a structure that can reach up to 825 tons.

- Generally wind speed increases with altitude, because at higher heights above the ground wind can flow more freely and with less friction, usually given by obstacles on the earth surface such as trees, buildings and mountains. For example, the average wind speed recorded at the largest offshore site ever built (London's Array, UK) is 9.2 m/s at 100 m (tower hub height), while at a height of 500 m, the average wind speed is 11.6 m/s. Hence, today the main trend is to realize taller and taller wind turbines equipped with bigger blades to exploit the stronger high altitude wind. To understand this, it's enough to consider that the most powerful wind turbine (16 MW), the Chinese MySE 16.0-242, will have a diameter of 242 m and its blades will be 118 m long. To realize such enormous work of engineering, an impressive quantity of materials (often polluting, such as cement and steel for the tower) and surrounding free space are required, increasing costs and land occupation.
- The capacity factor, *i.e.* the average power output divided by the maximum power capability of a wind turbine, today ranges from 26% to 52%. It is partly determined by the turbine design (which is continuously improving) but is strongly influenced by the wind availability, limiting the use of wind power to few areas that can guarantee a high power density at low altitude (and so competitive prices for energy).

These are the main limitations of the actual wind power production, and there is where new concepts, such as Airborne Wind Energy Systems (AWES), are coming into play.

1.2. Airborne Wind Energy Systems

Airborne Wind Energy (AWE) is a novel technology based on rigid or soft kites that autonomously fly at high altitude attached to the ground by a tether, with the goal of converting wind energy into electrical energy via ad-hoc generating systems (discussed in Section 1.2.1).

The idea of exploiting tethered wings to produce electricity is known since the '70s, and in 1980 Miles Loyd published the paper *Crosswind kite power* [8], demonstrating the viability of its intuitions about crosswind power and suggesting that this technology may overcome the classic wind turbine. After decades of silence, in the early 2000s the Airborne Wind Energy concepts started to be developed by researchers from all over the world that were fascinated by Loyd's vision, also thanks to the technology advance reached in the field of autonomous vehicles and advanced control systems.

AWE systems main advantage is the relative small size for a complete autonomous working apparatus, if compared to a classic HAWT with the same power generation potential: the Google Makani 600 kW wing (M600) weighs more or less 2 tons (including the tether and on-board generators) while a 600kW wind turbine weighs between 50 and 100 tons without considering the foundations, namely the heaviest part. The lightness implies less costs for the realization of an AWE system, starting from the required materials: kites are typically made of cheap materials (or even recycled, see [9]), that can be soft fabric, aluminum or carbon fiber, while the tether is realized with elastic polymers and the ground station can be set up in the space of a common container; according to [10], in terms of materials AWE systems needs around 20% of what is required for traditional HAWT.

Consequently, the installation of an AWE system is easier than an equivalent wind plant, since neither cranes nor particular infrastructures are needed. For example, the complete Kitepower Falcon system (Figure 1.4, from [11]) fits in a single shipping container (6.06m length x 2.44m width x 2.60m height) and weighs 9.5 tons, hence can be transported by a single truck, bringing the plant where is usually logistically impossible or very expensive to install HAWTs: in principle, AWE systems could autonomously power remote islands and mining operations in the mountains, or electrically sustain small and isolated communities.

In the latter application, AWE systems could also represent an alternative to traditional polluting fuels for house-heating and cooking and, at the same time, provide an independent source of electricity to the 770 million of people that live in developing countries and that don't have access to the grid for logistical reasons.

Indeed, the main potential advantage of AWE technology is the independence from the traditional requisites for wind power sites, *i.e.* the presence of abundant and strong wind

Figure 1.4: Kitepower Falcon (see [11]), a soft kite with ground level generation and its ground station/container (6.06m length x 2.44m width x 2.60m height), with an overall weight of 9.5 tons)

at low altitude: through the drones, AWE plants can access to high altitude winds up to 800 m, that are far stronger and more reliable than winds at HAWTs level, and can continuously modify their working altitude to harvest the most powerful winds available at the moment. Thanks to this feature, AWE could achieve impressive performance: according to [12], considering 100-m altitude versus 500-m altitude, the wind speed increases on average of more or less 20%, meaning that more power will be generated if the high altitude winds are exploited. As exposed in [13]:

> "The theoretical global limit of wind power at high altitude has been estimated to be about 4.5 times greater than what could be harvested at ground level"

Indeed, wind power strongly depends on wind speed according to the formula:

$$P_w = \frac{1}{2}\rho\, S\, V^3 \tag{1.1}$$

where P_w is the wind power, ρ is the air density, S is the swept area (by the AWE drone or HAWT blades) and V is the wind speed. From Equation 1.1, we notice a cubic dependence between V and P_w, which implies that considering the 20%-increase of wind speed observed in [12] when moving from 100 m to 500 m of altitude, 73% more power is generated, keeping constant the swept area. This concept locates AWE system at the opposite side with respect to HAWT research direction: AWE main goal is to reach significantly high altitudes and so faster winds, while traditional wind power research is focused in producing largest blades to increase the value of S in Equation 1.1.
In other words, AWE experts are converging towards the design of better control sys-

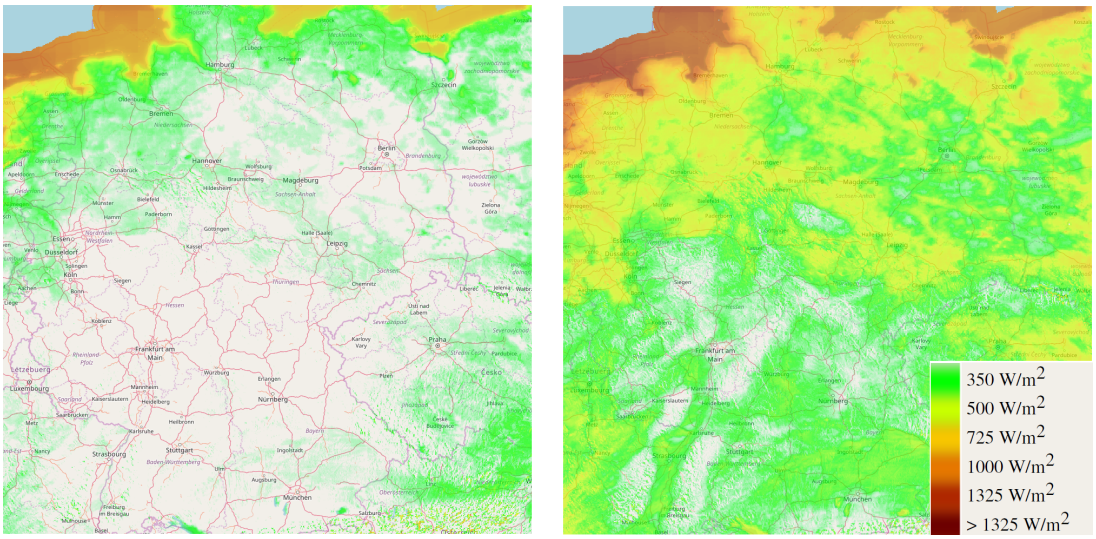


Figure 1.5: Average wind power density in Germany at 100 m (left figure) and 200 m (right figure), from [7]

tems and estimation algorithms to fly tethered autonomous drones efficiently and safely, whereas the wind power traditional technology efforts are heading towards building taller cement towers in the middle of the ocean.
In the same way, we can assess AWE advantages by considering the wind power density (WPD) variation with altitude increase by looking at Figure 1.5 from [7]: at 200 m, almost all Germany territory becomes suitable for harvesting wind energy by achieving a consistent value of WPD, while at 100 m only the coastal area in the North is convenient for wind power plants. Expanding the previous rationale, it is clear how AWE could completely transform the global energy system, by making a clean renewable energy source available all over the world, from the Northern Sea offshore parks to the small villages in Central Africa.
To make another example, we can take as reference Figure 1.6, where wind data from the city of London are reported: around 100 m, we have 210 W/m^2 and 7 m/s of wind speed, while at 500 m the wind power density more than triples, reaching 924 W/m^2, and the wind speed touches 11.6 m/s. To help visualizing how incredible is this concept, it is enough to think that these numbers would qualify London as better than any other wind park ever built in terms of wind speed, since the actual record is 10.46 m/s registered at the HyWind Scotland offshore wind farm.

For all these reasons, today AWE is a promising technology that is making its first steps into the energy market thanks to developing companies like Ampyx Power [14], KitePower [11], SkySails Power [15] and Kitekraft [9].

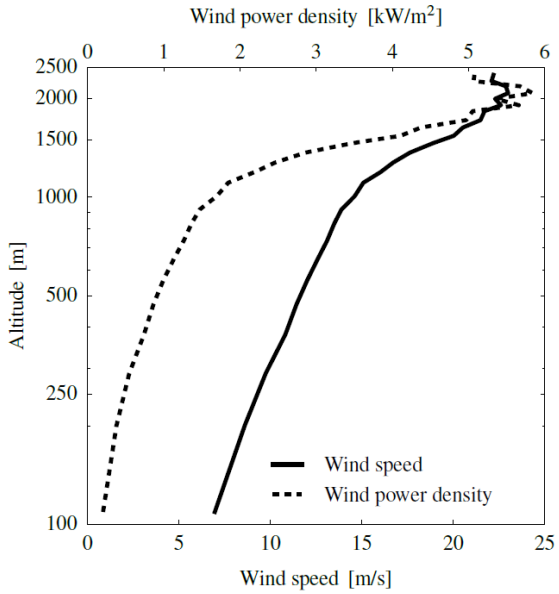


Figure 1.6: Average wind speed and wind power density above London, from [7]

1.2.1. Generation Principles

Nowadays, there are two main working principles for the AWE systems: the ground-level generation (GLG) and the on-board generation (OBG). Both methods exploit the lift force of the wind to produce energy (like classic HAWTs) but with substantial differences in the practice.

In ground-level generation systems, the kite operates in pumping cycles, alternating between a phase in which energy is generated and a phase where a small fraction of power is instead consumed. Going deeper in detail, during the so-called traction phase the wing is driven in a way to produce a lift force and consequently a traction force that pulls the tether. Usually, this is achieved by flying the kite in figure-8 paths in *crosswind*, since it has been proved that crosswind flight induces a stronger apparent wind, hence a stronger aerodynamic force that results in more pulling force on the tether. The latter, unwinding from a drum located at ground level, causes the rotation of an electrical generator allowing the power production. Then, the tether is reeled back by the winch motor (retraction phase), hence consuming energy, to bring the kite to the starting point. The combination of the two phases leads to a pumping motion, also called "yo-yo" motion (right picture in Figure 1.7).

However, pumping kite generators present a highly discontinuous power output, since the working cycle alternates power generation and consumption for periods of tens of seconds. Such a feature makes it necessary to resort to electrical rectification techniques, employing batteries or large capacitors before sending the power to the grid, constituting a major

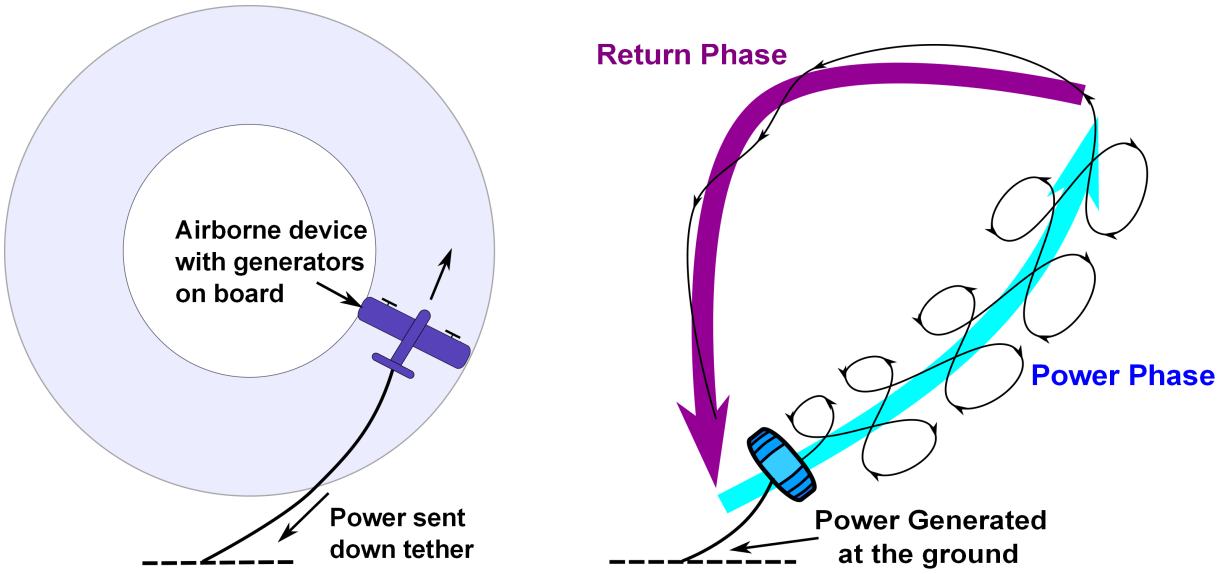


Figure 1.7: Illustration of the basic operations of OBG (left picture) vs GLG (right picture), from [16]

disadvantage for the technology. GLG systems are mainly realized using soft kites, that can be:

- Single-layer kites equipped with inflatable structures, to enhance the wing aerodynamics;
- Double-layer kites made of canopy cells, which are open on the leading edge to allow the air to inflate all cells during the flight and giving the kite the necessary stiffness;
- Kites made of a single layer of fabric material (called Delta kites), reinforced by a rigid frame that provides better aerodynamic efficiency, which in turn results in a higher efficiency for the wind power extraction.

An interesting Italian company, Kitenergy [17], is developing a prototype that employs a reinforced fabric kite with two tethers and features 60 kW of rated power.
In the same way, Skysails Power [15] designed a system that employ soft kites able to generate 80 to 200 kW of rated power, and is planning to realize an offshore system in the next years. SkySails' AWE system is based on a foil kite controlled with one tether and a control pod which regulates the lengths of kite bridles for steering the kite and changing its angle of attack.
On the other hand, there are companies that are developing GLG using rigid kites. For example, the system developed by the Dutch company Ampyx Power [14] employs a glider with carbon fiber body equipped with on-board actuators for the control surfaces and two small propellers to help the drone in the climb towards the working altitude.

Figure 1.8: AP3 glider from Ampyx Power [14]

The latest prototype realized by Ampyx Power is the AP3 system, which flies a 12-m wingspan glider (Figure 1.8) between 200 and 450 m and has a rated power of 150 kW. Kitemill [18] and TwingTec [19] are instead exploiting the fixed-wing design in order to introduce a vertical take-off and landing (VTOL) strategy for the drone, installing multiple propellers on their kites. The VTOL take-off and landing approach is indeed very convenient for this kind of systems, since it allow to reduce the space required for the maneuvers and hence the size of the plant.
The second most diffused working principle is the on-board generation, whereby the kites are equipped with small wind generators mounted on the flying wing. In this framework, the tether is made of conductive material and brings the produced electricity to the ground station, which finally connects the plant to the grid.
The main feature of this technology is the use of the on-board wind generators not only to produce power, but also as motors to move the drone in the initial maneuvers, driving the wing to the optimal conditions to start the power generation phase, while in the meantime the tether is unwound. Indeed, many OBG systems are designed to perform autonomous vertical take-off and landing operations, where the propellers actively push the drone in the air, similarly to the commercial drones take-off. Then, the production phase begins and the drone harvests the wind energy while flying in crosswind, usually in circular loops. OBG systems are realized with rigid kites, since the drone must sustain the generators weight: however, rigid wings result to be intrinsically more aerodynamic efficient than soft kites, because the design can be optimized to provide convenient lift and drag coefficients to the wing.

In this moment, there are two companies that investigate OBG systems:

- Kitekraft [9] proposes a small aluminum wing with a boxplane structure and a truss-like shape, minimizing weight but providing sufficient rigidity.

- WindLift [20] is building a four-propellers rigid wing that can produce up to 15 kW and is being developed in partnership with the U.S. Marine Corps.

Figure 1.9: Kitekraft

Figure 1.10: WindLift

An interesting airborne wind energy project is Makani [21], a Californian company born in 2006. After years of independent research, Makani was acquired by Google in May 2013 and in 2019 Royal Dutch Shell made a minority investment to begin a partnership with the company.
During its activity, Makani team investigated the OBG technology focusing on the creation of a fully autonomous drone able to generate energy. The peak of Makani work was the realization of the M600 system, theoretically able to produce up to 600 kW using a custom-made drone (that will be described in the next section). With the M600, in 2019 Makani was the first company to achieve an offshore flight of an airborne wind turbine (Figure 1.11).
Makani was then shut down by Google Alphabet in February 2020, since as declared by the owners of the company, "the road to commercialization is longer and riskier than hoped".

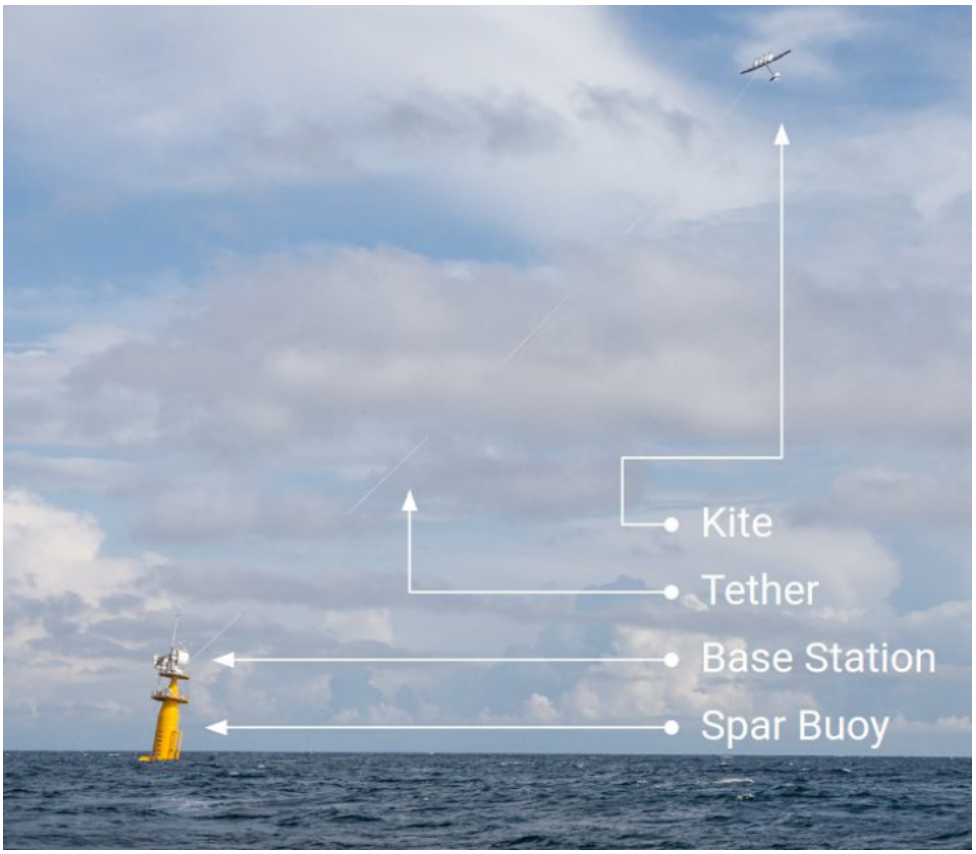


Figure 1.11: M600 during the offshore flight in the Norwegian Sea in 2019

1.2.2. State of Art and Thesis Purposes

Airborne wind energy has been deeply investigated in the last decade, with many contributions from the scientific AWE community, that also realized a literature review [22] which collects and analyses the main recent papers and technological findings.
The physics of the tethered drone flight was firstly explored in Loyd's work [8], where the author focused on the crosswind flight motion and power analysis.
About modelling, an interesting work regarding soft kites is [23], in which online identification is used to implement an adaptive control scheme capable of handling the typical uncertainties in the crosswind flight dynamics of the kite.
In [24], the authors propose a realistic kite model that includes the kite aerodynamic characteristics and the effects of tether weight and drag forces. Then, a nonlinear model predictive control is employed to maximize the energy produced during the pumping cycle while satisfying input and state constraints.
Generally, the presence of tether reduces AWE performance, because the tether drag force severely limits the power production: to address this issue, in [25] an AWE system with two airfoils and a shared tether is proposed to reduce overall system drag. This choice

would allow the main tether (*i.e.* the longest segment) to remain almost still in the air, while two smaller segments are effectively moving and so generating the inconvenient drag force. To achieve satisfactory results with this technique, a properly balancing of the system trajectories and parameters is required, so the coordination of the two drones is addressed using optimal control techniques.
Examining the literature, we find many different control approaches regarding AWE systems, since their typical non-linearity do not allow to use classic off-the-shelf strategies. For example, in [26] the control of a rigid boxed-wing kite with on-ground generation by Skypull SA [27] is addressed by using a hierarchical approach for all the operational phases of the system, including a steering-authority analysis. The authors also investigated a vertical take off strategy for the same rigid drone in [28], while in [29] we find an approach for the autonomous launch and landing of a tethered rigid aircraft by using a linear motion system made of a slide translating on rails.
Several take-off approaches are gathered and deepened in [30], where the authors highlight pros and cons of each strategy on the basis of quantitative and qualitative criteria (to assess their technical and economic viability) and relate the power required for the take-off functionality to the overall generated power.
For what concern AWE systems testing, [31] presents the systematic approach and analysis of the flight test verification of a small-scale rigid wing by Ampyx Power ([14]).

About on-board generation systems, literature offers far less material, due to the higher-complexity level of these systems with respect to GLG plants and the bigger difficulties in realizing low-cost prototypes.
In [32], the main physics of AWE is investigated via analytical equations, both for OBG and GLG rigid-wing systems, to derive a model for economical considerations. Then, in [33] a nonlinear inverse model is developed for the performance analysis of OBG systems, focusing on the power production.

In September 2020 Makani [21] released the Energy Kite Collection, a three-part report [34] [35] [36] with a collection of open source code repositories, flight logs and technical videos from the project that will be the starting point for the entire development of this Thesis. The Makani database offers indeed the possibility to tackle the modelling and control problems with a data-driven approach which, combined with optimization techniques, can bring to significant results and insights about the typical non-linear interactions between the drone, the tether and the ground station without the necessity to develop an accurate white box model.

The idea behind this Thesis is thus exploit the large amount of available data to create a simulator that can effectively reproduce the behaviour of the M600, starting from a simple

octocopter model (see [37]) and proposing an optimization problem to tune the control system using the real flight data. In this way, we aim to contribute to the advance of the modelling and control for on-board generation AWE systems, by providing a new tool to study these kind of systems.

In Chapter 2, we present the models of each subsystem used for the realization of the AWE simulator:

- The M600 is considered as a octocopter, using a 6-degrees of freedom model that involves non-linear couplings between the states;
- The tether is assumed as made by Kelvin-Voigt material, *i.e.* as multiple nodes connected by springs and dampers in parallel;
- The base station is modelled as a winch driven by an electric motor (supposed ideal), taking into account the pulling force exerted by the drone on the tether.

In the same Chapter the employed control systems are described: we adopted linear strategies as pole placement and PI control, referring to the linearized model of the subsystems as presented in [37]. In addition, in Section 2.6 we discretize the control system, while in Section 2.5 we propose an approach for the modelling of aerodynamic forces based on a method employed in the study of HAWTs.

In Chapter 3, we analyse the data released by Makani. First, we retrieve the system parameters and second, we produce smaller datasets from the flight data in order to capture the hovering phase of the M600. In particular, we will extract the reference position signals and the thrust command generated by the real M600 controller, the corresponding actual drone trajectory and the wind measures.

The new datasets will be used in Chapter 4 to set up an optimization routine for the optimal tuning of control system for the hovering phase. The optimization is performed by giving as input to the simulator the previously cited reference signals and comparing the simulator response with the available data. We hence obtain a constrained non-linear optimization problem , that is solved via Sequential Quadratic Program (SQP) with backtracking line-search, returning the optimally tuning parameters for the kite control system.

Finally, in Chapter 5 we present the results by quantifying the performance of the simulator using error indicators and we validate the optimization outcome.
Then, in Chapter 6 we propose future possible developments, such as the extension of the optimization procedure to the power generation phase.

1.2.3. Innovative Contributions

The main original concepts introduced in this Thesis are:

- The non-linear modelling an OBG AWE system, validated on the available flight data for the Makani M600 (Section 2.2);
- The implementation of a cascaded control scheme, originally designed for standard drones and now adapted for the M600 apparatus (Section 2.2.2 - Section 2.2.5);
- The integration of different submodels, namely the kite, the tether and the ground station together with the respective control systems, to develop a complete simulator of an OBG AWE system;
- The identification of the aerodynamic lift and drag coefficients and the characterization of the lift and drag curves on an interval of 360° of the angle of attack α. The procedure is based on experimental results for HAWTs and Makani flight data (Section 2.5);
- The data-driven tuning of the drone control system for the take-off and hovering stages, exploiting the Makani flight data to define an optimization procedure (Chapter 4).

2 | System Model and Control

The considered system is composed of three main elements: the drone, the tether and the ground station.

The M600 drone is developed with an OBG approach, allowing the use of a high performance rigid wing and enabling easy launch and land operation by using the power system in propulsion mode to hover the kite. The propulsion/generation system uses a high voltage DC drive with a control scheme that enables it to transfer up to 1 MW of electrical power to and from the electrical grid, through a tether that is less than 30 mm in diameter.
As depicted in Figure 2.1, the M600 drone autonomously launches from the ground station in a vertical take-off and landing (VTOL) configuration and hovers as an octocopter as the tether is reeled out from the winch mounted on the ground station. The tether supports the hovering phase by feeding the motors with electricity from the grid.
Once the tether is fully reeled out, the system transitions into power-generating crosswind flight with the so-called trans-in maneuver, and starts flying in circular loops as a traditional plane, as long as wind conditions allow a satisfactory power production.
At the end of the generation cycle, the kite transitions out of crosswind (trans-out maneuver), hovers as the tether is reeled in, and lands on the perch present on the ground station, which also manages electrical inverters to allow bidirectional current flow with the electrical grid.

2.1. Reference System and Notation

Since the model is made by many subsystems, multiple reference frames have been adopted to facilitate the modelling. First of all, let's define the global inertial right-handed reference frame $G \coloneqq (x_G, y_G, z_G)$, centered at the ground station with the Z-axis pointing upwards. Then, a local frame $L \coloneqq (x_L, y_L, z_L)$ is introduced to better describe the drone position and orientation, as depicted in Figure 2.2. The drone orientation will be represented using Euler angles $\phi(t)$,$\theta(t)$ and $\psi(t)$, *i.e.* roll, pitch and yaw angles.

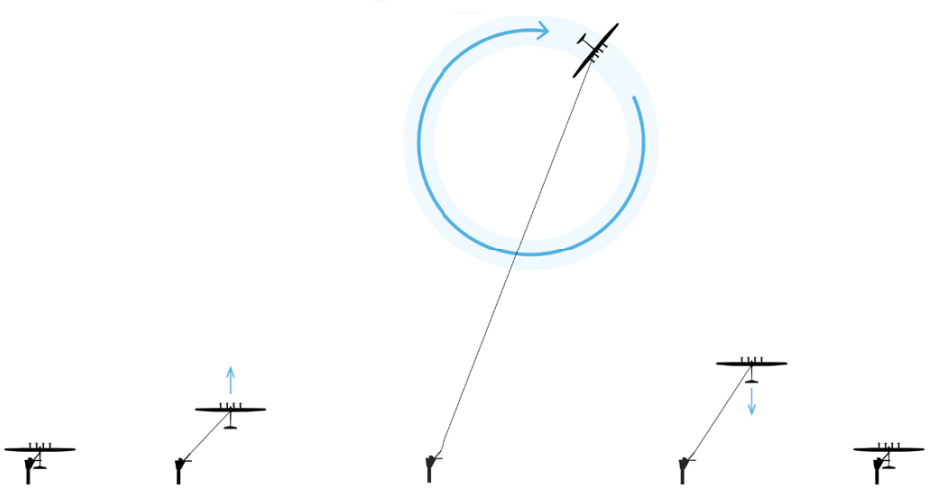


Figure 2.1: Representation of the five main maneuvers of the M600 cycle. From left to right: autonomous VTOL takeoff, hover/trans-in maneuver, generation phase, trans-out maneuver, VTOL landing on perch.

To relate the global and local frames , it is necessary to define a rotation matrix $R(t)$, that is a function of the Euler angles:

$$R = \begin{bmatrix} c(\psi)c(\theta) & s(\psi)c(\theta) & -s(\theta) \\ c(\psi)s(\theta)s(\phi) - s(\psi)c(\phi) & s(\psi)s(\theta)s(\phi) + c(\psi)c(\phi) & c(\theta)s(\phi) \\ c(\psi)s(\theta)c(\phi) + s(\psi)s(\phi) & s(\psi)s(\theta)c(\phi) - c(\psi)s(\phi) & c(\theta)c(\phi) \end{bmatrix} \tag{2.1}$$

where $c(\cdot) = \cos(\cdot)$, $s(\cdot) = \sin(\cdot)$ and the time dependency of R and the Euler angles is omitted for brevity.
Matrix R is used to move from the inertial reference frame to the local one, in order to translate a vector $\vec{P}_G$ to its analogous $\vec{P}_L$ in the local frame as:

$$\vec{P}_L = R\,\vec{P}_G \tag{2.2}$$

Due to the rotation matrices properties, to move from the local to the global frame, it is enough to compute

$$\vec{P}_G = R^T\,\vec{P}_L \tag{2.3}$$

since for a rotation matrix $R^{-1} = R^T$.
Then, to help the modelling of the aerodynamic forces, the wind reference frame $W := (x_W, y_W, z_W)$ is introduced, where x_W is aligned with the apparent wind direction, y_W is parallel to the wings and z_W pointing up.
Given a wind velocity vector in the inertial frame, namely $\vec{W}_G$, let's define the apparent

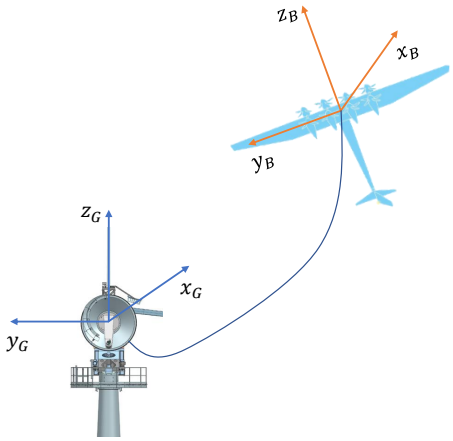


Figure 2.2: Global and Local Reference Frames

wind velocity in the local frame $\vec{W}_a = [W_{a,x}\, W_{a,y}\, W_{a,z}\,]^T$ as

$$\vec{W}_a = R\,(\vec{W}_G - \dot{\vec{P}}_G) \tag{2.4}$$

where $\dot{\vec{P}}_G$ is the drone velocity in the inertial frame. Then, we can compute the angle of attack α and the side slip angle β as

$$\alpha = \arctan\left(\frac{W_{a,z}}{W_{a,x}}\right) \tag{2.5}$$

$$\beta = \arcsin\left(\frac{W_{a,Y}}{\|W_a\|_2}\right) \tag{2.6}$$

Now, we define the rotation matrix from the wind reference frame W to the local system L as

$$H = \begin{bmatrix} \cos(\alpha) & 0 & -\sin(\alpha) \\ \sin(\alpha)\sin(\beta) & \cos(\beta) & \cos(\alpha)\sin(\beta) \\ \sin(\alpha)\cos(\beta) & -\sin(\beta) & \cos(\alpha)\cos(\beta) \end{bmatrix} \tag{2.7}$$

For what concerns the data, the quantities collected in the database (described in Chapter 3) are defined according to Makani global reference system $M_G := (x_{M,G}, y_{M,G}, z_{M,G})$ and local reference system $M_B := (x_{M,B}, y_{M,B}, z_{M,B})$, so the vectors collected in the database have to be manipulated to be used in the previously introduced frames.
M_G is a North East Down (NED) frame, hence, to express the data in the global frame G, we need to rotate the vectors of 180° about the $y_{M,G}$ axis. Let's define the rotation

matrix from M_G to G as R_κ

$$R_\kappa = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ -0 & 0 & -1 \end{bmatrix} \tag{2.8}$$

Then, we perform the rotation by pre-multiplying the vectors in the M_G reference frame by R_κ.

To deal with quantities defined in the Makani body frame M_B, we extract the time-depending rotation matrix $R_{M,BG}$ from M_B to M_G from the database and we rotate the vectors defined in M_B into the global Makani frame. Finally, we pre-multiply the result by R_κ, obtaining the vectors expressed in the G frame.

Ultimately, we define the notation for the angular velocity of the drone. To be consistent with the literature, we denote as p, q and r the angular rates of the drone about its local axes. The relationship between p, q and r and the Euler angles derivative is defined by Equation (2.9):

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin(\phi)\ \tan(\theta) & \cos(\phi)\ \tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)/cos(\theta) & \cos(\phi)/\cos(\theta) \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2.9}$$

In the following, a description of each of component is proposed, together with the modelling approach and the implemented control strategy.

2.2. Kite

The studied kite is the Makani M600 model SN4, well described in [34] and for which flight data are available. The M600 is a fixed-wing carbon fiber kite with a wingspan of 25.66 m equipped with eight rotors attached to the same number electric engines, that can work both as motors or wind generators depending on the operative phase. The kite was designed to produce up to 600 kW of electric power, so the size of the generators and their effect on the kite dynamics are not negligible and will be taken into account in the proposed model.

The kite launches in a VTOL configuration and behaves as a octocopter for the whole duration of the hovering phase, so the M600 can be modelled following the traditional Lagrange approach, as proposed in [37]. To slightly simplify the analysis, we decided to neglect the effect of the aerodynamic moments acting on the drone and to assume that all the drone states are measurable. This latter hypothesis is however not so strong as it could appear, since as reported in [34] the M600 is equipped with a large number of sensors to measure its states, often redundant for safety and fault prevention.

Figure 2.3: Makani M600 drone (picture from [34])

The employed model is the following:

$$\ddot{\vec{P}}_G = \frac{1}{m}\left(R^T \begin{bmatrix} 0 \\ 0 \\ U_1 \end{bmatrix} + F_T + F_{aero} \right) - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{2.10}$$

$$\dot{p} = \frac{I_y - I_z}{I_x}\, q\, r + \frac{U_2}{I_x} - \frac{J_p}{I_x}\, q\, \Omega_r \tag{2.11}$$

$$\dot{q} = \frac{I_z - I_x}{I_y}\, p\, r + \frac{U_3}{I_y} + \frac{J_p}{I_y}\, p\, \Omega_r \tag{2.12}$$

$$\dot{r} = \frac{I_x - I_y}{I_z}\, p\, q + \frac{U_4}{I_z} \tag{2.13}$$

where m is the drone mass, g is the gravitational acceleration, F_T is the force exerted by the tether on the drone, F_{aero} is the aerodynamic forces vector, I_x, I_y, I_z are the drone rotational moments of inertia (see Table A.1), J_p is the propellers inertia, Ω_r is the sum of the eight propellers rotor speeds.
In the model, the states are the global position $\vec{P}_G = [x_G\, y_G\, z_G\,]^T$ and the Euler angles ϕ, θ and ψ, while the control inputs are represented by the vector $U = [U_1\, U_2\, U_3\, U_4\,]^T$ and are applied by the propellers. Specifically, U_1 is the thrust, U_2 is the rolling moment, U_3 is the pitching moment and U_4 is the yawing moment, and will be converted into propeller speeds by the allocation system (see Figure 2.4).
The chosen model is highly non-linear, so we tackle the control design problem by taking as reference a linearized model as suggested in [37], to consequently realize a cascaded control system made by three nested loops as shown in Figure 2.4.
Finally, to facilitate the control-oriented models considerations, the explicit computation

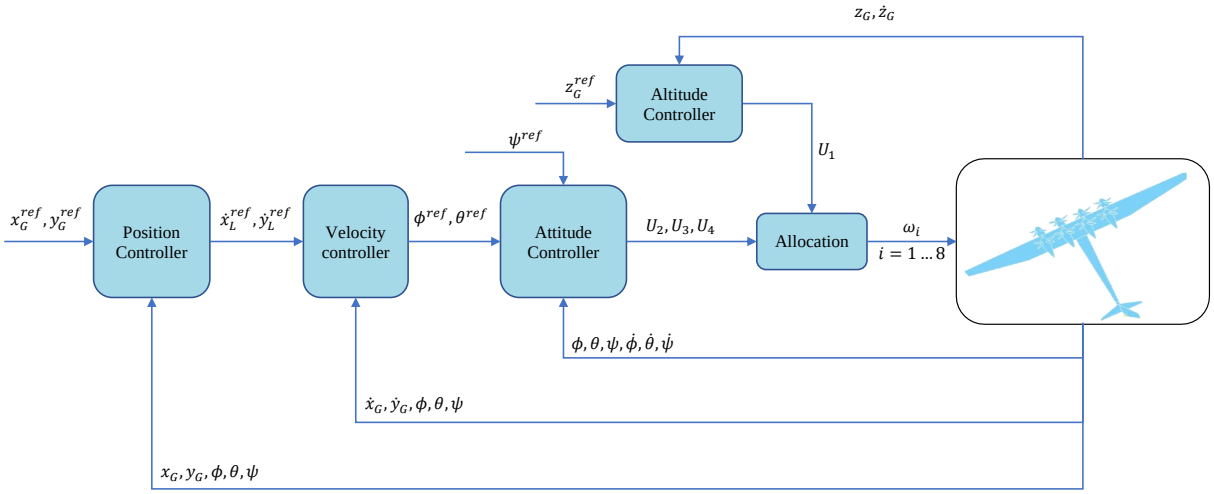


Figure 2.4: Cascade control scheme: three nested loops for attitude, altitude, velocity and position control of the drone. The allocation block translates the control signals into propellers rotational speeds.

the Equation (2.10) can be useful

$$\ddot{x}_G = (\cos(\psi)\ \sin(\theta)\ \cos(\phi) + \sin(\psi)\ \sin(\phi))\ \frac{U_1}{m} + F_{T_x} + F_{drag} \tag{2.14}$$

$$\ddot{y}_G = (\sin(\psi)\ \sin(\theta)\ \cos(\phi) - \cos(\psi)\ \sin(\phi))\ \frac{U_1}{m} + F_{T_y} + F_{side} \tag{2.15}$$

$$\ddot{z}_G = \cos\theta\ \cos\phi\ \frac{U_1}{m} - g + F_{T_z} + F_{lift} \tag{2.16}$$

where F_{lift}, F_{side} and F_{drag} are the aerodynamic forces described in Chapter 2.5 and F_{T_x}, F_{T_y} and F_{T_z} are the tether force components in the global reference frame, better discussed in Chapter 2.3. The main parameters of the drone are gathered in Table A.1.

2.2.1. Allocation

The allocation block (see Figure 2.4) maps the U vector computed by the controller into the eight angular velocities commands for the eights propellers. Typically, the lift force and drag torque generated by a single drone propeller are modelled as

$$L = b\,\omega^2 \tag{2.17}$$

$$D = d\,\omega^2 \tag{2.18}$$

where ω is the rotor angular speed while b and d are constant parameters known as rotor lift and drag coefficients.

To identify these two coefficients, we use the data released by Makani, in which we find the experimental bench test performed on a single rotor: thus, we have a database that contains the thrust produced by the propeller and the absorbed electrical power with the corresponding angular rate (see Table A.5), that can be used to fit the quadratic models for the rotors lift and drag forces.

Rotor Lift Coefficient

We identify the lift coefficient b using the Least Squared technique, since our aim is finding the value of b such that the difference between the thrust data and the quadratic model (2.17) is minimized. Defining the experimental values of angular rate and thrust as $\tilde{\omega}$ and $\tilde{T}$ (*i.e.* the first and second column of Table A.5), we can write an unconstrained optimization problem

$$\min_{b} \quad \left(\tilde{T} - b\,\tilde{\omega}^2\right)^T \left(\tilde{T} - b\,\tilde{\omega}^2\right) \tag{2.19}$$

Then, we apply the Least Squares formula to compute the lift coefficient b

$$b = \tilde{\omega}^+ \tilde{T} \tag{2.20}$$

where the matrix $\tilde{\omega}^+$ is the pseudo-inverse of $\tilde{\omega}^2$.

Rotor Drag Coefficient

To obtain the d coefficient, we define the electrical efficiency of the propeller motor as η, in order to compute the mechanical power absorbed by the rotor P_m as

$$P_m = \eta\, \tilde{P}_e \tag{2.21}$$

where $\tilde{P}_e$ is the absorbed electrical power (third column of Table A.5), we can derive the torque $\tilde{D}$ produced by the rotor as

$$\tilde{D} = \frac{P_m}{\omega} \tag{2.22}$$

Now, observing the quadratic model (2.18), we notice that

$$\frac{L}{D} = \frac{b}{d} \tag{2.23}$$

so we estimate the drag coefficient d as the mean of the vector containing the element-wise division between $\tilde{L}$, $\tilde{D} \in \mathbb{R}^{50\times 1}$, multiplied by the lift coefficient b.

Allocation Matrix

Once introduced the lift and drag coefficients, we proceed computing the allocation matrix $M \in \mathbb{R}^{4\times 8}$, *i.e.* a static matrix that relates the eight angular velocity of the propellers $\omega_p = [\omega_1\, \omega_2\, \omega_3\, \omega_4\, \omega_5\, \omega_6\, \omega_7\, \omega_8\,]^T$ and the vector $U = [U_1\, U_2\, U_3\, U_4\,]^T$, which gathers the four control variables of the system. The relation is hence the following

$$U = M\, \omega_p^2 \tag{2.24}$$

where $\omega_p^2 = \omega_p \odot \omega_p$ and the symbol $\odot$ is the Hadamard (element-wise) product. To actually build the allocation matrix, we apply the D'Alembert principle computing one force balance and three torque balances, keeping in mind the meaning of the elements of U as described in Section 2.2.
U_1 is the thrust command, so we can write

$$U_1 = \sum_{i=1}^{8} L_i = \sum_{i=1}^{8} b\, \omega_i^2 = b \sum_{i=1}^{8} \omega_i^2 \tag{2.25}$$

Then, we compute the torque balances in the local reference frame, about x_L and y_L axes

$$U_2 = l_{out}\, b\, (\omega_1^2 - \omega_4^2 - \omega_5^2 + \omega_8^2) + l_{in}\, b\, (-\omega_2^2 + \omega_3^2 + \omega_6^2 - \omega_7^2) \tag{2.26}$$

$$U_3 = h_{top}\, b\, (\omega_1^2 - \omega_2^2 - \omega_3^2 + \omega_4^2) + h_{bot}\, b\, (\omega_5^2 - \omega_6^2 - \omega_7^2 + \omega_8^2) \tag{2.27}$$

where the symbols l_{out}, l_{in}, h_{top} and h_{bot} refer to the geometrical parameters reported in Figure 2.5.
Finally, recalling that U_4 is the control action responsible for the yaw movement, we can express it by means of the drag coefficient

$$U_4 = d\, (-\omega_1^2 + \omega_2^2 + \omega_3^2 - \omega_4^2 + \omega_5^2 - \omega_6^2 - \omega_7^2 + \omega_8^2) \tag{2.28}$$

The allocation matrix is thus defined as

$$M = \begin{bmatrix} b & b & b & b & b & b & b & b \\ b\, l_{out} & -b\, l_{in} & b\, l_{in} & -b\, l_{out} & -b\, l_{out} & b\, l_{in} & -b\, l_{in} & b\, l_{out} \\ b\, h_{top} & -b\, h_{top} & -b\, h_{top} & b\, h_{top} & b\, h_{bot} & -b\, h_{bot} & -b\, h_{bot} & b\, h_{bot} \\ -d & d & d & -d & d & -d & -d & d \end{bmatrix} \tag{2.29}$$

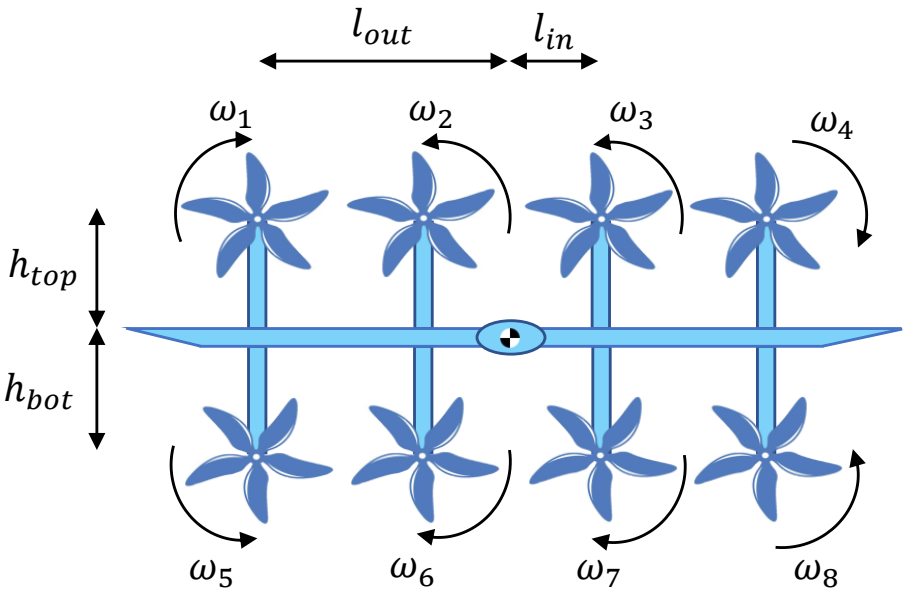


Figure 2.5: Front-view representation of the drone. In the figure are reported the geometrical parameters that describe the position of the rotors with respect to the center of mass.

Now, we need to invert Equation (2.24), in order to translate the command vector U into suitable propellers speed commands. However, since M is a 8×4 rectangular matrix (and hence not invertible), there are infinite solutions for the vector ω_p. To tackle this issue, we can consider that limiting the drone power consumption is all our interest, so we resort to the concept of Moore-Penrose pseudo-inverse, through which we compute the minimum-norm value of ω_p^2, as

$$\omega_p^2 = M^+ U \tag{2.30}$$

where M^+ is the pseudo-inverse of M. From this point, we obtain ω_p as

$$\omega_p = \sqrt{sgn(\omega_p^2) \odot \omega_p^2} \tag{2.31}$$

where $sgn(\cdot)$ is the signum function, employed to force the squared root argument to be positive.

2.2.2. Altitude Controller

The altitude controller is realized with reference to a linearized model based on Equation (2.10), that is obtained by assuming that pitch and roll angles are small enough to have

$\cos(\theta) \approx 1$ and $\cos(\phi) \approx 1$. Then, from Equation (2.10) one obtains:

$$\ddot{z}_G = \frac{U_1}{m} - g \tag{2.32}$$

Now, defining $z_1 = z_G$ and $z_2 = \dot{z}_G$ and considering in a first moment the gravity term as an external disturbance, we can write the state-space equations as:

$$\begin{cases} \dot{z}_1 = z_2 \\ \dot{z}_2 = \dfrac{U_1}{m} \end{cases} \tag{2.33}$$

From here, we derive the altitude controller via pole placement and, to achieve static precision on the altitude tracking, we enlarge the state vector to add an integral state, obtaining the state-feedback gains $K_z = [K_{z_1}\ K_{z_2}]$ and the integral gain K_z^{int}. Now, by compensating in open loop the gravity term, we get the final control law

$$U_1 = -K_z \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + K_z^{int} \left(z_G^{ref} - z_1\right) + m\, g \tag{2.34}$$

Finally, we implement the controller adding an anti-windup scheme to prevent the integral saturation since the value of U_1 is bounded by the intrinsic limits of the propellers speed between a value $\pm\, \bar{U}_1$, defined as

$$\bar{U}_1 = 8\, b\, \bar{\omega}_p^2 - m\, g \tag{2.35}$$

where $\bar{\omega}_p$ is the propeller maximum speed in absolute value.
The reference signal for the speed $\dot{z}_G$ is set to zero, while z_G^{ref} is the desired vertical position of the drone.

2.2.3. Attitude Controller

The attitude controller design is based on three assumptions, with reference to Equations (2.11), (2.12), (2.13):

- The non linear couplings between the rotational velocities are negligible;
- The contribution given by the propellers inertia is small compared to the drone one;
- The Euler angles are small enough to approximate $sin(\cdot) \approx 0$ and $cos(\cdot) \approx 1$, so

Equation (2.9) becomes

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \approx \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \tag{2.36}$$

In this way, we obtain the linear control-oriented model:

$$\begin{aligned} \ddot{\phi} &= \frac{U_2}{I_x} \\ \ddot{\theta} &= \frac{U_3}{I_y} \\ \ddot{\psi} &= \frac{U_4}{I_z} \end{aligned} \tag{2.37}$$

Following the same approach as in Section 2.2.2, we obtain the state space matrices to realize a state feedback control law via pole placement, in order to track the Euler angles reference signals. In this controller, we decide to don't add any integrator since the outer loops ensure smooth references that are easily followed.
The control law is:

$$\begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} = -K_{ang} \begin{bmatrix} \phi^{ref} - \phi \\ \dot{\phi} \\ \theta^{ref} - \theta \\ \dot{\theta} \\ \psi^{ref} - \psi \\ \dot{\psi} \end{bmatrix} \tag{2.38}$$

where $K_{ang} \in \mathbb{R}^{3\times 6}$ is the gain constant matrix.
Here, we set the reference signals for the angular rates to zero, while ψ^{ref} can be imposed by an external guidance system. ϕ^{ref} and θ^{ref} are instead computed by the velocity control loop.

2.2.4. Velocity Controller

The velocity controller regulates the local velocities $\dot{x}_L$ and $\dot{y}_L$ by using as control signals the reference Euler angles θ^{ref} and ϕ^{ref}, that will be then given as input to the Attitude Controller described in 2.2.3.
Now, assuming that the roll and pitch angles are so small that

$$\cos\phi \approx 1 \quad \cos\theta \approx 1 \quad \sin\phi \approx \phi \quad \sin\theta \approx \theta \tag{2.39}$$

we obtain from Equation (2.10) the model

$$\begin{aligned} \ddot{x}_G &= (\cos(\psi)\,\theta + \sin(\psi)\,\phi)\frac{U_1}{m} \\ \ddot{y}_G &= (\sin(\psi)\,\theta - \cos(\psi)\,\phi)\frac{U_1}{m} \end{aligned} \tag{2.40}$$

Writing Equations (2.40) in matrix form

$$\begin{bmatrix} \ddot{x}_G \\ \ddot{y}_G \end{bmatrix} = \frac{U_1}{m} \begin{bmatrix} \cos(\psi) & \sin(\psi) \\ \sin(\psi) & -\cos(\psi) \end{bmatrix} \begin{bmatrix} \theta \\ \phi \end{bmatrix} \tag{2.41}$$

If we define the matrix in Equation (2.41) as a rotation matrix R_ψ

$$R_\psi = \begin{bmatrix} \cos(\psi) & \sin(\psi) \\ \sin(\psi) & -\cos(\psi) \end{bmatrix} \tag{2.42}$$

we can express θ and ϕ as functions of $\ddot{x}_G$ and $\ddot{y}_G$ respectively, as

$$\begin{bmatrix} \theta \\ \phi \end{bmatrix} = R_\psi^T \begin{bmatrix} \ddot{x}_G \\ \ddot{y}_G \end{bmatrix} \frac{m}{U_1} \tag{2.43}$$

Now, recalling relation (2.2) we move to the local reference frame

$$\begin{bmatrix} \theta \\ \phi \end{bmatrix} = \begin{bmatrix} \ddot{x}_L \\ -\ddot{y}_L \end{bmatrix} \frac{m}{U_1} \tag{2.44}$$

Let's assume that $U_1 \approx m\,g$, *i.e.* that the contribute of the altitude controller is small enough compared to the weight force added in feedforward in Equation (2.34): this is reasonable if we consider the huge mass of the drone, as reported in A.1. We obtain

$$\begin{bmatrix} \ddot{x}_L \\ \ddot{y}_L \end{bmatrix} = \begin{bmatrix} \theta \\ -\phi \end{bmatrix} g \tag{2.45}$$

Supposing that the Attitude Controller performs correctly giving a good reference tracking, we can approximate $\phi \approx \phi^{ref}$ and $\theta \approx \theta^{ref}$

$$\begin{bmatrix} \ddot{x}_L \\ \ddot{y}_L \end{bmatrix} = \begin{bmatrix} \theta^{ref} \\ -\phi^{ref} \end{bmatrix} g \tag{2.46}$$

Now, let's introduce the proportional control laws for the two axes

$$\begin{aligned} \theta^{ref} &= \quad K_x^{vel}\,(\dot{x}_L^{ref} - \dot{x}_L) \\ \phi^{ref} &= -\quad K_y^{vel}\,(\dot{y}_L^{ref} - \dot{y}_L) \end{aligned} \tag{2.47}$$

where $\dot{x}_L^{ref}$ and $\dot{y}_L^{ref}$ are the reference speeds imposed by the outer position loop. Closing the loop by substituting Equation (2.47) into (2.46) and moving to the Laplace domain, we get the closed loop transfer functions for the velocity subsystem, which take $\dot{x}_L^{ref}$ and $\dot{y}_L^{ref}$ as input and return $\dot{x}_L$ and $\dot{y}_L$. The two transfer functions are

$$\begin{aligned} F_{\dot{x}_L}(s) &= \frac{K_x^{vel}\,g}{s + K_x^{vel}\,g} \\ F_{\dot{y}_L}(s) &= \frac{K_y^{vel}\,g}{s + K_y^{vel}\,g} \end{aligned} \tag{2.48}$$

From here the tuning of the system comes rather straightforward and is performed by imposing the desired closed-loop poles through the two gains K_x^{vel} and K_y^{vel}.

2.2.5. Position Controller

The position loop controls the position of the drone in the global reference frame in order to track given set-points x_G^{ref} and y_G^{ref}. It is based on the kinematic model

$$\begin{aligned} \dot{x}_G &= \dot{x}_G^{ref} \\ \dot{y}_G &= \dot{y}_G^{ref} \end{aligned} \tag{2.49}$$

The position controller outputs are the reference speeds for the velocity controller, so we use these variables as control variables to design a PI controller

$$\begin{aligned} \dot{x}_G^{ref} &= K_{px}\,(x_G^{ref} - x_G) + K_{ix}\int (x_G^{ref} - x_G) \\ \dot{y}_G^{ref} &= K_{py}\,(y_G^{ref} - y_G) + K_{iy}\int (y_G^{ref} - y_G) \end{aligned} \tag{2.50}$$

Now if we substitute the control laws (2.50) in Equation (2.49) and we re-arrange the expressions in the Laplace domain, we obtain the closed-loop transfer functions of the position loop

$$\begin{aligned} F_x(s) &= K_{px}\,\frac{K_{px}\,s + K_{ix}}{s^2 + K_{px}\,s + K_{ix}} \\ F_y(s) &= K_{py}\,\frac{K_{py}\,s + K_{iy}}{s^2 + K_{py}\,s + K_{iy}} \end{aligned} \tag{2.51}$$

From here, it is possible to select the desired closed loop behaviour for the position loop by opportunely tuning the gains K_{px}, K_{ix}, K_{py} and K_{iy}. The controller is then implemented using an anti-windup scheme, since the control variables $\dot{x}_G^{ref}$ and $\dot{y}_G^{ref}$ are bounded between saturation limits.
Finally, we have to convert the global reference signals into the local axes in order to be consistent with the (local) velocity controller described in Section 2.2.4:

$$\begin{bmatrix} \dot{x}_L^{ref} \\ \dot{y}_L^{ref} \end{bmatrix} = \begin{bmatrix} \cos(\psi) & \sin(\psi) \\ -\sin(\psi) & \cos(\psi) \end{bmatrix} \begin{bmatrix} \dot{x}_G^{ref} \\ \dot{y}_G^{ref} \end{bmatrix} \tag{2.52}$$

2.2.6. Anti-Windup Scheme

To efficiently implement the Altitude Controller 2.2.2 and the position PIs controllers 2.2.5, we add an anti-windup scheme in each regulator to improve the control system robustness. In this way, we prevent the windup of the integral action when the control action hits the saturation limits, avoiding the typically related problems as strong output oscillations or high peaks of the control variable. In particular, we adopted a back-calculation anti-windup technique (see Figure 2.6). With this approach, the anti-windup scheme comes into play only in case of saturation, since its action is based on the difference between the output of the integrator and the saturated control action (signals v and u in Figure 2.6): it is easy to understand that if v does not exceed the saturation limits, then $u = v$ and their difference is null.
The signal obtained by subtracting u to v is then multiplied by a tunable gain K_{bc}, which ultimately determines the magnitude of the anti-windup effect and the response speed of the de-saturating action. Indeed, the anti-windup output signal p (*c.f.* Figure 2.6) is then subtracted to the error e, in order to reduce the integrator input and consequently its output v, de-saturating the control action u which is then applied to the plant.

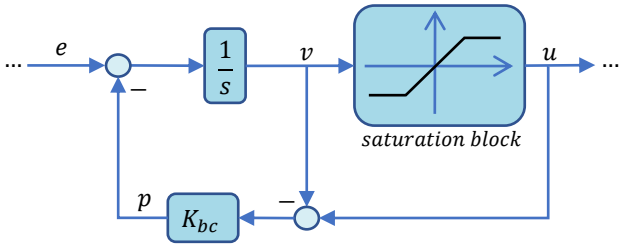


Figure 2.6: Anti-windup scheme with back-calculation approach

2.3. Tether

The tether is a key element in the overall system, since it couples the movement of the kite with the ground station.
To properly design the tether, we have to consider that the tension force that is generated during the system operations can reach up to 250 kN, so the tether has to be strong enough to support the load. On the other side, the tether is an object that moves into a fluid (the air) so we have to consider the drag force associated to it and, obviously, its weight: as a consequence, the large tension force cannot be counteracted by increasing the tether diameter, since the cable should be also lightweight and aerodynamically efficient to minimize weight and drag forces. Furthermore, the tether must carry significant electrical power (up to 600 kW) and data (30 $Mbit/sec$) so it should provide a good insulation and very small losses.
Keeping [34] as reference, the tether employed in the M600 system is made by a carbon fiber core, wrapped with insulated conductor cables and coated with an outer protective jacket that provide a drag reduction. The optimal parameters selected by Makani are reported in Table A.2.

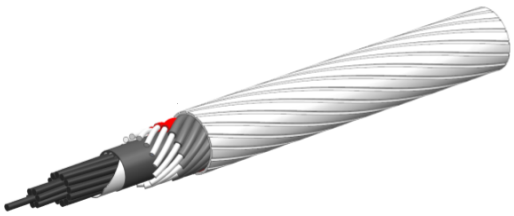


Figure 2.7: Makani Tether (from [34])

The tether behaviour is fundamental in AWE systems, since its interactions with the drone and the ground station are manifold and highly non-linear. Thus, the accuracy of the chosen model becomes crucial in the realization of the whole simulator. One of the most used models is the "straight line" model: the tether is indeed considered in an ideal way as a beam, and the forces generated by it (weight force, aerodynamic forces and tension force) are applied directly on a point mass model of the wing. This approach usually brings many advantages for the design of the control system and for simulation purposes.
However, we decided to adopt a different approach, based on the subdivision of the tether in segments joint by point mass nodes, which account for the gravity force. The segments are then supposed as made of a Kelvin-Voigt material: in this way, we consider each point mass node as connected to the next and the previous by a parallel spring-damper

system, *i.e* the interaction between two consecutive nodes is modelled through elastic and damping forces.

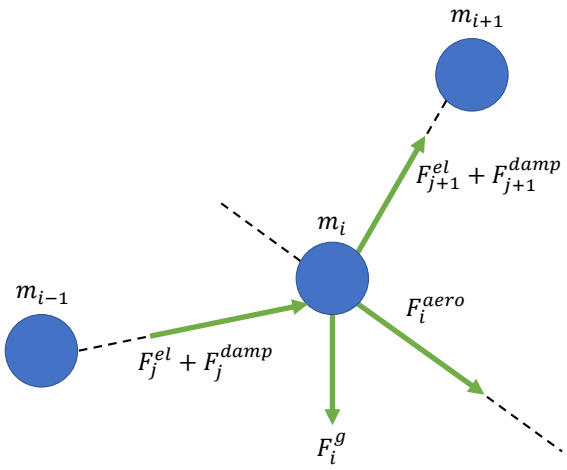


Figure 2.8: Force contribution on an i-th node of the tether: F^{aero} is the drag force, F^g is the weight force, F_j^{el} and F_j^{damp} are the elastic and damping forces exerted on the i-th node by the j-th segment while F_{j+1}^{el} and F_{j+1}^{damp} are the elastic and damping forces exerted by the i-th node on the j-th segment

This choice allows to attribute to each node a state, *i.e.* a 3D position (and velocity) in time, that we compute by integrating the force balance

$$F_i^{tot} = m_i \, a_i \tag{2.53}$$

where m_i and a_i are the mass and the acceleration of the i-th node, while F_i^{tot} is the sum of the forces acting on the i-th node. In this way, we obtain a tether model far away more accurate that the classic beam one, that allows the simulator to achieve more generality. For further information, see [38].

2.4. Ground Station

The ground (or base) station manages the reeling in and the reeling out operations of the tether during the flight maneuvers. Indeed, correctly regulating the tether length is crucial to achieve the tether maximum tension, necessary to avoid the tether to become saggy and eventually hit objects like trees or buildings during the M600 flight.
The ground station realized by Makani also features a mechanism that allow the structure to rotate around its vertical axis, following the drone movements to not damage the tether

integrity with eventual torsional forces.
In the offshore configuration, the ground station is mounted to a spar buoy: a steel tubular structure that mates with the ground station by means of a bolted flange and is anchored to the seabed by three lines.

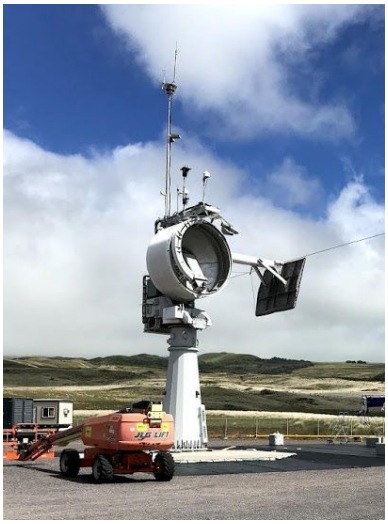


Figure 2.9: Ground Station (from [35])

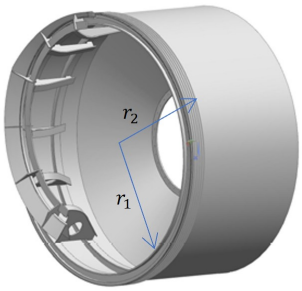


Figure 2.10: Winch Drum (from [35])

The ground station implemented in the simulator is a simplified version of the Makani apparatus, in which we ignored the ability to rotate above its vertical axes: we consider the station as a winch driven by an electric motor that reels in and out the tether.

The winch drum is modelled as a hollow cylinder: referring to Figure 2.10, we can compute the drum inertia as

$$J_{drum} = \frac{1}{2} m_w ({r_1}^2 + {r_2}^2) \tag{2.54}$$

Then we obtain the dynamical model using a torque balance, in which we take into account the tether tension, the viscous friction and the motor torque

$$J_{drum} \ddot{\lambda} = \|F_T\| \, r_2 - T_m - \beta_{drum} \, \dot{\lambda} \tag{2.55}$$

where λ, $\dot{\lambda}$ and $\ddot{\lambda}$ are the winch angular position, velocity and acceleration, J_{drum} is the drum inertia, β_{drum} is the viscous friction coefficient and T_m is the motor torque (all the parameters are reported in Table A.3).
The proposed control system for the winch (see Figure 2.11) is a pole placement scheme, where we add an integrator to ensure a satisfactory reference tracking.

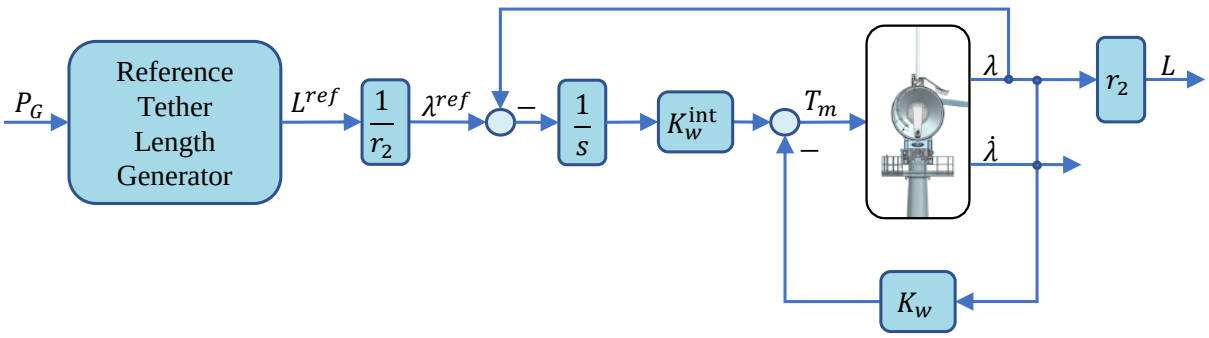


Figure 2.11: Winch Control Scheme: Pole placement with integral action

First, let's write Equation (2.55) in state space representation, by taking $x_1 = \dot{\lambda}$ and $x_2 = \lambda$ as states and T_m as control variable:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \dfrac{1}{J_{drum}} \left(\|F_T\| \, r_2 - T_m - \beta_{drum} \, x_1 \right) \end{cases} \tag{2.56}$$

Now we augment the state vector by adding an integral state $\dot{\zeta} = x_2^{ref} - x_2$, in order to follow the angular position reference x_2^{ref} with zero steady-state error

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \dfrac{1}{J_{drum}} \left(\|F_T\| \, r_2 - T_m - \beta_{drum} \, x_1 \right) \\ \dot{\zeta} = x_2^{ref} - x_2 \end{cases} \tag{2.57}$$

Then, assuming that both x_1 and x_2 are measurable and treating the torque generated by tether tension and x_2^{ref} as an external disturbances, we write the state-space matrices as

$$A_w = \begin{bmatrix} -\beta/J_{drum} & 0 & 0 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \quad B_w = \begin{bmatrix} -1/J_{drum} \\ 0 \\ 0 \end{bmatrix} \tag{2.58}$$

Now we can compute the pole placement gains, obtaining the following expression for the control variable T_m

$$T_m = -K_w \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + K_w^{int} \left(x_2^{ref} - x_2 \right) \tag{2.59}$$

where $K_w = [K_{x_1} \; K_{x_2}]$ is the state feedback gain matrix and K_w^{int} is the integral gain.

The reference signal for the winch angular position is computed as

$$\lambda^{ref} = \frac{L^{ref}}{r_2} \tag{2.60}$$

where L^{ref} is the reference tether length, defined as function of the global drone position P_G plus an offset Δ_L, which is a tuning parameter

$$L^{ref} = \|P_G\| + \Delta_L \tag{2.61}$$

In this way, we ensure that the winch reels out only the strictly necessary amount of tether in order to keep the tether almost always taut, avoiding to entangle the environmental obstacles in the working area.

2.5. Aerodynamic Forces

The aerodynamic force vector acting on the drone $\vec{F}_{aero} = [F_{drag}\ F_{side}\ F_{lift}]^T$ is computed in the wind frame as

$$\begin{aligned} F_{drag} &= \frac{1}{2}\rho S C_D(\alpha,\beta)\|\vec{W}_{a,w}\|^2 \\ F_{side} &= \frac{1}{2}\rho S C_S(\alpha,\beta)\|\vec{W}_{a,w}\|^2 \\ F_{lift} &= \frac{1}{2}\rho S C_L(\alpha,\beta)\|\vec{W}_{a,w}\|^2 \end{aligned} \tag{2.62}$$

where ρ is the air density, S is the wing surface, $C_D(\alpha,\beta)$, $C_S(\alpha,\beta)$, $C_L(\alpha,\beta)$ are the drag, side and lift coefficients, and $\|\vec{W}_{a,w}\|$ is the modulus of the apparent wind velocity in the wind frame, obtained as

$$\vec{W}_{a,w} = H^T \cdot \vec{W}_a \tag{2.63}$$

where matrix H is the rotation matrix from the wind reference system to the local frame (defined in Equation (2.7)) that depends on the angle of attack α and the sideslip angle β, which are introduced in Section 2.1. Then, we translate the set of forces $\vec{F}_{aero}$ (expressed in the wind frame) into the global reference frame passing through the local one

$$\vec{F}_{aero,G} = R \cdot H \cdot \vec{F}_{aero} \tag{2.64}$$

where R is the rotation matrix from the global reference system and the local one, expressed as function of the Euler angles (see Equation (2.1)).
For simplicity, it is assumed that F_{side} is negligible, while the drag and lift coefficients are considered as dependant only on the angle of attack α, by assuming that β is small and

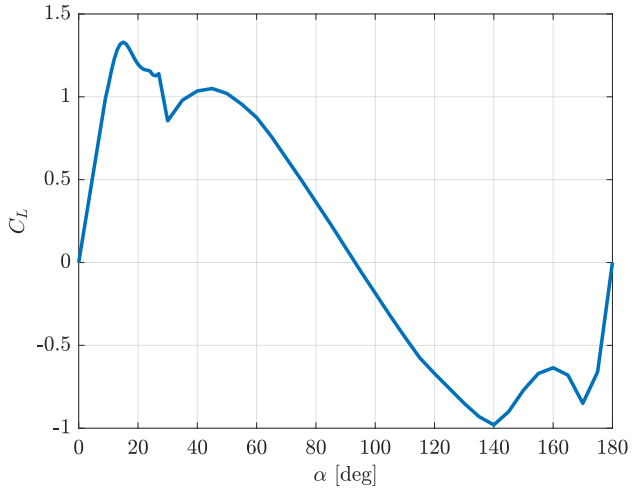


Figure 2.12: Lift coefficient function for the *NACA-0015* wing profile at $Re = 5 \cdot 10^6$ (data from [39])

its effect can be neglected, so that

$$\begin{aligned} C_D(\alpha, \beta) &\approx C_D(\alpha) \\ C_L(\alpha, \beta) &\approx C_L(\alpha) \end{aligned} \tag{2.65}$$

The relationship between the coefficients and α is obtained following the idea that the angle of attack can vary between $\pm 180°$, since the drone changes configuration from drone-like flight in hovering phase to plane-like flight during the crosswind operations. For this reason we considered the concepts exposed in [39], that are valid for HAWTs, and adapted to our context.
In particular, we used the data reported for the wing profile *NACA-0015* subject to a Reynolds number $Re = 5 \cdot 10^6$ to extract the drag and lift curves shape, and then we exploited the data released by Makani to compute the respective functions. Since [39] provides the values of the aerodynamic coefficients between 0° and 180°, the curves are firstly defined on this interval and then expanded to an interval of 360° by simple symmetrical considerations.
In the next subsections, the derivation of each curve will be discussed.

2.5.1. Lift Coefficient

Analyzing the data about the lift coefficient in [39], one can observe that the trend of the curve (reported in Figure 2.12) resembles a sine function, so the idea is to use as fitting

model a sum of sines

$$C_L(\alpha,\gamma) = \sum_{k=1}^{N_k} \gamma_k \cdot \sin((\alpha - \alpha_0)\cdot 2k) \tag{2.66}$$

where $a_0 = -19°$ is an offset angle necessary to match the Makani data with the data in [39], γ_k is the k-th scaling factor and N_k is a user-chosen parameter that represents the number of sine terms in the sum.

At this point, the vector of scaling factors $\gamma = [\gamma_1\ \gamma_2\ \dots\ \gamma_k\ \dots \gamma_{N_k}]$ is identified through an optimization problem formulated with the Least Absolute Shrinkage And Selection Operator (LASSO) approach, in order to obtain a sparse γ vector (*i.e.* with only few non-zero elements), very convenient for computational reasons since it avoids over-fitting problems.

The optimization problem to obtain the scaling factors is formulated as

$$\gamma_{LASSO} = \min_{\gamma\in\mathbb{R}^{N_k}} \|\gamma\|_1 \quad \text{subject to} \tag{2.67}$$

$$\|C_L(\alpha,\gamma) - \Phi\,\gamma\|_\infty \leq 0 \tag{2.68}$$

where $\Phi \in \mathbb{R}^{N\times N_k+1}$ is the regression matrix built with N data points from the Makani database. Then, we compute the fitted function as

$$C_{L,LASSO}(\alpha) = \Phi\,\gamma_{LASSO} \tag{2.69}$$

However, this curve presents some undesired oscillations (Figure 2.13, left picture) due to the large number of sinusoidal components N_k, so we build a new regression matrix Φ_{LS} considering only the frequencies filtered by the LASSO procedure, *i.e.* the components associated to a non-null scaling factor, and we compute the optimal γ_{LASSO} with the Least Square formula as

$$\gamma_{LS} = (\Phi_{LS}^T\Phi_{LS})^{-1}\Phi_{LS}^T\,C_{L,LASSO} \tag{2.70}$$

Finally, the function for the lift coefficient is:

$$C_L(\alpha) = \sum_{k=1}^{4} \gamma_{LS,k}\ \sin((\alpha-\alpha_0)\cdot 2\cdot f_k) \tag{2.71}$$

where $f = [f_1\ f_2\ f_3\ f_4]^T$ is the vector that gathers the frequencies of the sine components associated to each element of γ_{LS}. The curve is represented in the right picture of Figure 2.13.

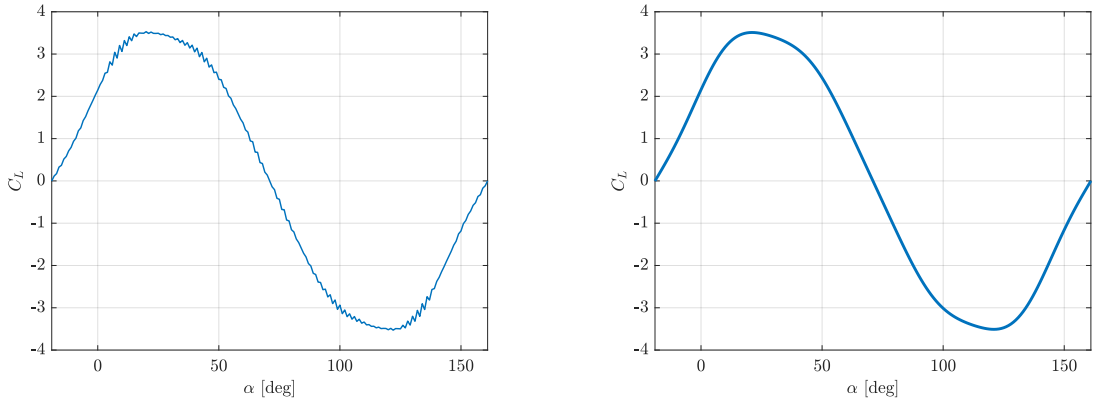


Figure 2.13: Identified curves for the lift coefficient. The left picture shows the output of the *LASSO* procedure, while the right picture represents the final curve, after the Least Squares smoothing operation

2.5.2. Drag Coefficient

Following the same approach, we observe that the data proposed in [39] can be well approximated by a Gaussian curve (see Figure 2.14), so we set the model for the identification procedure as:

$$C_D(\alpha, \delta) = \delta_1 e^{-\left(\frac{\alpha-\delta_2}{\delta_3}\right)^2} + C_{D0} \tag{2.72}$$

where $\delta = [\delta_1 \ \delta_2 \ \delta_3]^T$ is the unknown parameters vector to be identified and C_{D0} is an offset value taken from Makani data, necessary to have a drag curve always greater than zero. We hence set up a new optimization problem which simply aims to minimize the 2-norm of the error between the data $\tilde{C}_D$ published by Makani and the chosen model.

$$\min_{\delta \in \mathbb{R}^3} \ \|\tilde{C}_D - C_D(\alpha, \delta)\|_2 \tag{2.73}$$

Solving Problem (2.73), we obtain a vector δ_{CD} that, once substituted in (2.72), gives the identified drag curve, depicted in Figure 2.15. Ultimately, Figure 2.16 shows the two final curves employed by the simulator to compute the aerodynamic forces, *i.e.* the extension on a 360° interval of the identified curves in Figures 2.13 and 2.15.

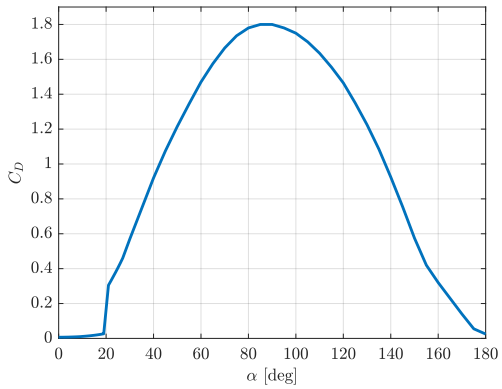


Figure 2.14: Drag coefficient function for the *NACA-0015* wing profile at $Re = 5{\cdot}10^6$ (data from [39])

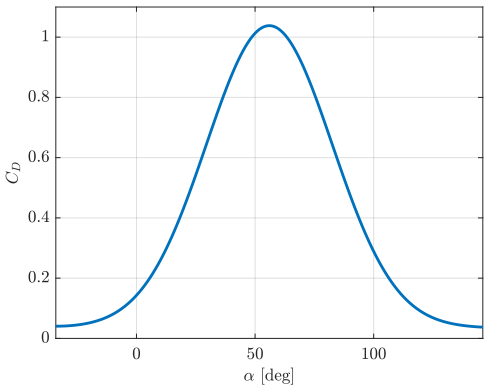
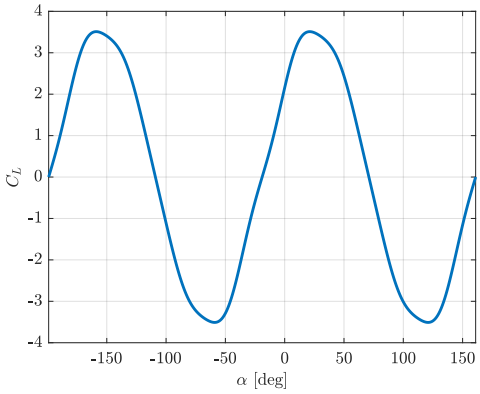


Figure 2.15: Drag curve obtained from the Gaussian fit of Makani data



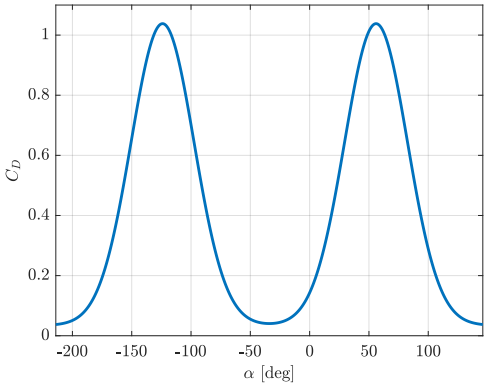


Figure 2.16: Identified lift (left picture) and drag (right picture) curves on a 360° interval

2.6. Controllers Discretization

The previously described controllers are discretized following the Forward Euler approach. With this technique, the continuous-time integrators employed in the Altitude Controller (Section 2.2.2), in the position PIs controllers (Section 2.2.5) and in the winch controller (Figure 2.11) are approximated in the discrete-time domain as functions of the *Z-transform*

$$\frac{1}{s} \approx \frac{T_s}{z-1} \tag{2.74}$$

where s is the Laplace transform, z is the Z-transform and T_s is the sampling period, defined as the inverse of the sampling frequency f_s

$$T_s = \frac{1}{f_s} \tag{2.75}$$

The sampling frequency is a crucial parameter and has to be accurately tuned: f_s has to be large enough to capture all the system dynamics but, at the same time, it should be the smallest possible to guarantee a small computational burden. Knowing that, we define as ω_c^{il} the frequency of the fastest closed-loop pole of the innermost loop (*i.e.* the loop composed of the attitude and altitude controllers, see Figure 2.4) and we select the value of f_s using the following rule of thumb

$$f_s \gg \frac{\omega_c^{il}}{2\pi} \tag{2.76}$$

3 | Database

After the shut down, Makani published the flights database and their own M600 simulator as a free resource for anyone who was interested in carrying on their work: in this Chapter, we introduce the database structure and define how to build the dataset used in Chapter 4.

The database includes both the system parameters and the logs of all the flights performed by the Makani drones. The system parameters are reported in the setting files of the Makani simulator as *JSON* database that can be easily imported in Matlab. From these files we extract:

- The parameters for the three subsystems described in Chapter 2, *i.e.* the M600 drone, the tether and the ground station, that are reported respectively in Tables A.1, A.2, A.3;
- The values of $\tilde{\omega}$, $\tilde{T}$ and P_e, reported in Table A.5, that are used to estimate the parameters of the propellers in Section 2.2.1 (see Table A.4);
- The data for the identification of the aerodynamic coefficients discussed in Section 2.5.

For what concerns flight data, Makani released the database produced by the drone avionics system, which logs to the central system the sensors data and the signals computed by the control systems. All of the avionics nodes in the system send asynchronous messages over the avionics network and Makani log files are constructed by simply recording the network traffic using the *hdf5* format, which presents an interface with aspects both of a hierarchical tree and of a time series. These data files can be inspected with Matlab, Python or other softwares.

The flights gathered in the released database are divided in four testing programs:

- The Remote Perch Crosswind (RPX), that was performed in China Lake test site (California, US) between 2016 and 2018. During this period, the Makani team completed nine tests with a rudimentary version of the system previously described

Figure 3.1: RPX program (from [34])

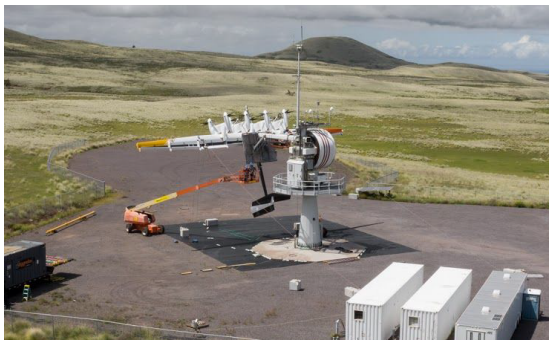


Figure 3.2: CW program (from [34])

in Chapter 2, as we can see from the base station structure in Figure 3.1

- The High Hovering (HH), that are the first two high-hover flights executed in 2018 with the complete system;
- The All-Modes Crosswind (CW), performed at Parker Ranch (Hawaii Island, US, Figure 3.2) in 2019. In these twelve test, the M600 drone flown complete working cycles from launching to landing;
- The Floating Crosswind (FC), in which Makani achieved the first off-shore crosswind flight of AWE history, that was also the last flight for Makani as a company.

In Matlab, the database is a *struct* with a field for each flight, denoted using the previously introduced acronyms (*e.g.*, *CW01* stands for the 1^{st} All-Modes Crosswind flight). Each flight is in turn a *struct* with 26 fields, filled with avionics data. The principal fields are:

1. Control input, where the Makani controller input values are logged, after being converted to real units and expressed in useful coordinate systems. The control input field includes GPS wing position, drone acceleration (via IMUs), wind (via pitot tubes), tether force (via loadcells), rotor speeds, *etc*;
2. Control output, containing all the output signals of the controller, such as rotor speed commands and the winch velocity command;
3. State estimator, where the estimated wing position, attitude, velocity, angle of attack, sideslip, ... are stored, together with the winch and the base station states variables;
4. Hover, that includes the signals used and generated by the controller in charge of the hovering phase. In these data, we find the commands for 3-D position and the thrust command that will be used in Chapter 4.

Starting from this huge amount of data, we proceed by extracting only the ones that

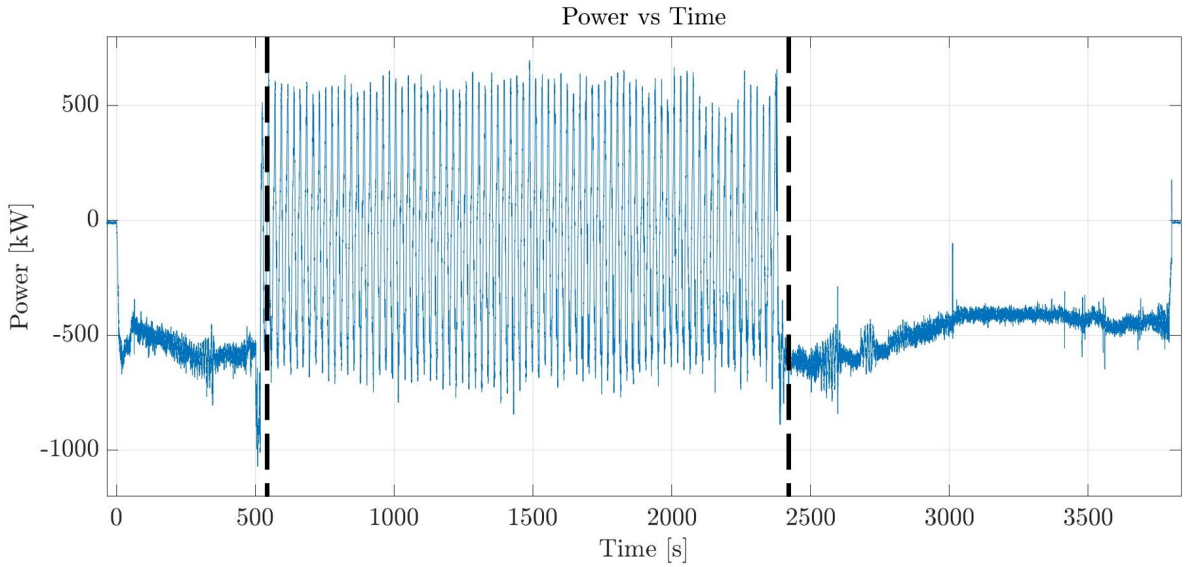


Figure 3.3: Power time course of *CW01* flight. From left to right we identify the initial power-consuming phase, the generation stage and the final landing maneuvers.

we are interested in to realize smaller (and lighter) datasets regarding *only* the hovering phase.
In order to do that, we analyse the power logs, in which we identify the operative phases of the system: at the beginning we have a power-consuming stage, that includes vertical takeoff, hovering and trans-in maneuvers (Figure 3.4); then, the M600 starts flying in crosswind, generating power but also spending a fraction to complete the circular path; finally, as Figure 3.3 shows, another power-consuming phase begins, where the drone exits from crosswind (trans-out) and approaches the ground station to land on the perch.

Now, to extract the hovering phase from the initial power-consuming stage, we consider two empirical power thresholds based on data observation, obtaining the hovering time interval in which we are interested.
After that, we create the new database selecting only the fields of the flights *struct* that will be useful for the optimization tackled in Chapter 4, *i.e.* the wind and the wing position, plus the position command vectors and the thrust computed by the actual Makani controller during the real tests.
These quantities are collected in the database according to Makani global reference frame and have to be rotated in the global frame G before being used: recalling the definition of R_κ from Section 2.1, for example the wing position in the global frame P_G is computed as

$$P_G = R_\kappa \, P_{G,M} \tag{3.1}$$

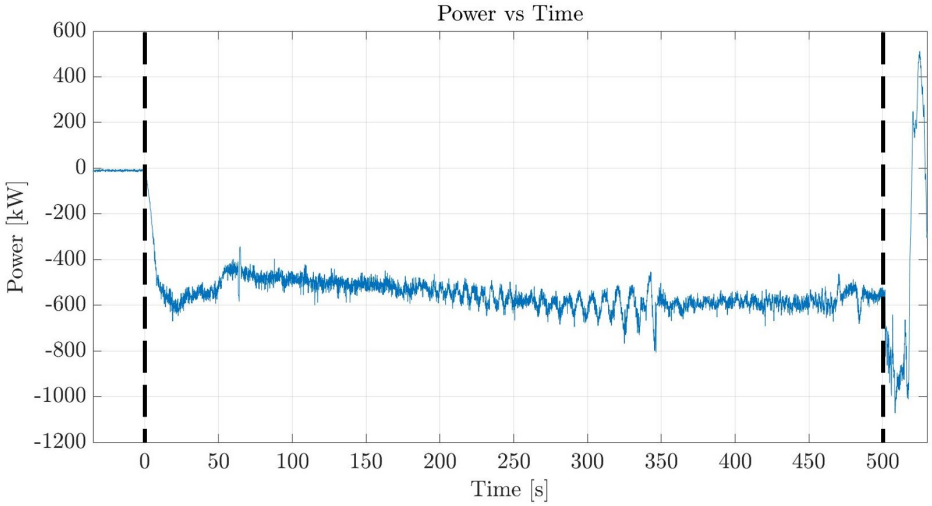


Figure 3.4: Focus on the initial power-consuming phase. From left to right: take-off, hovering and trans-in maneuvers.

where $P_{G,M}$ is the position available from the database.

Let's now focus on the wind vector: the database provides the values of the apparent wind vector $W_{app,B}$ in the Makani body frame M_B, hence some manipulations are required to express the wind vector in the G frame, in order to then use it for the aerodynamic forces computation as defined in Section 2.5.
First of all, we compute the absolute wind vector in the Makani body frame as

$$W_{M,B} = W_{app,B} + \dot{P}_{M,B} \tag{3.2}$$

where $\dot{P}_{M,B}$ is the drone velocity vector in M_B, available from the database. Then, wind vector in the G frame is obtained as

$$W_G = R_\kappa \cdot R_{M,BG} \cdot W_{M,B} \tag{3.3}$$

where we use the notation introduced in Section 2.1.

Finally, we perform the data downsampling, since the considered signals are recorded in the database with a frequency of 10 Hz, which is too computational demanding for being used in the optimizer. Hence, we create the new database by imposing the sampling frequency for each of the previously discussed signals to 1 Hz, that however guarantees a sufficient number of data points.

4 | Optimal Control Tuning

In this Chapter we present the formulation of the non-linear optimization problem that computes the optimal values of the control systems parameters. In other words, the controllers parameters are automatically chosen by the optimizer to make the simulator behaving as close as possible to the Makani M600.

The general formulation adopted optimization problem is the following:

$$\min_{\omega_c \in \mathbb{R}^7} \quad f(\omega_c) \quad \text{s. t.} \tag{4.1}$$

$$C\,\omega_c \geq d \tag{4.2}$$

where $\omega_c = [\omega_c^z\ \omega_c^{\dot{z}}\ \omega_c^{z_{int}}\ \omega_c^{\phi,\theta}\ \omega_c^{\psi}\ \omega_c^{\dot{x},\dot{y}}\ \omega_c^{x,y}]^T$ is the optimization variables vector, $f(\omega_c)$ is the cost function and (4.2) represent the linear constraints applied to the problem.

4.1. Optimization Variables

The optimization variables are related to the bandwidths of the three controllers, and are defined as:

- ω_c^z, $\omega_c^{\dot{z}}$, $\omega_c^{z\,int}$ are the tuning parameters for the altitude controller and are used to select the poles position for the Pole Placement, representing respectively the altitude position, velocity and integral poles introduced in Section 2.2.2. The actual bandwidth of this controller is determined by the smallest value between ω_c^z, $\omega_c^{\dot{z}}$, $\omega_c^{z\,int}$ in a dominant-pole approximation;
- $\omega_c^{\phi,\theta}$ and ω_c^{ψ} are used to tune the attitude controller parameters in the same way employed with the altitude controller. To reduce the computational burden, we merged the optimization variables for the roll and pitch angles into $\omega_c^{\phi,\theta}$ and we set the poles for the angular rates equal to the ones for the angular position;

- $\omega_c^{\dot{x},\dot{y}}$ refers to the velocity controller and is related with the gains K_x^{vel} and K_y^{vel} as

$$K_x^{vel} = K_y^{vel} = \frac{\omega_c^{\dot{x},\dot{y}}}{g} \tag{4.3}$$

- $\omega_c^{x,y}$ is the position loop bandwidth, through which we set the parameters of the PIs as

$$\begin{aligned} K_{px} &= K_{py} = 2\,\omega_c^{x,y} \\ K_{ix} &= K_{iy} = (\omega_c^{x,y})^2 \end{aligned} \tag{4.4}$$

  In this way, we shape the transfer functions in Equations (2.51) in order to impose the two poles of each transfer function as coincident with frequency equal to $\omega_c^{x,y}$.

4.2. Cost Function

The cost function $f(\omega_c)$ is expressed as

$$\begin{aligned} f(\omega_c) = \sum_{t=0}^{T} & w_x^2\,(x_G^{sim}(t,\omega_c) - \tilde{x}_G(t))^2 + w_y^2\,(y_G^{sim}(t,\omega_c) - \tilde{y}_G(t))^2 + \\ & + w_z^2\,(z_G^{sim}(t,\omega_c) - \tilde{z}_G(t))^2 + w_u^2\,(U_1^{sim}(t,\omega_c) - \tilde{U}_1(t))^2 \end{aligned} \tag{4.5}$$

Equation (4.5) presents four terms, where x_G, y_G and z_G are the global coordinates of the wing, U_1 is the commanded thrust and the superscript sim and the $\sim$ symbol indicate if the source of the signal is, respectively, the simulation or the database. The constant parameters w_x, w_y, w_z and w_u are weights, used to scale the errors to the same order of magnitude. In this way we measure the difference between the simulator output and the data, computing the error for each quantity of interest at each time instant $t \in [0, T]$, where T is the duration of the considered flight. In the following, we will omit the time dependency and the subscript $_G$ for the sake of brevity.

4.3. Constraints and Bounds

The employed linear inequality constraints are the following:

$$\omega_c^z \geq \omega_c^{\dot{z}} \tag{4.6}$$
$$\omega_c^{z\,int} \geq \omega_c^{\dot{z}} \tag{4.7}$$
$$\omega_c^{\dot{z}} \geq n\,\omega_c^{\phi,\theta} \tag{4.8}$$
$$\omega_c^{\dot{z}} \geq n\,\omega_c^{\psi} \tag{4.9}$$
$$\omega_c^z \geq f_s\,\omega_c^{\dot{x},\dot{y}} \tag{4.10}$$
$$\omega_c^{\dot{z}} \geq f_s\,\omega_c^{\dot{x},\dot{y}} \tag{4.11}$$
$$\omega_c^{\phi,\theta} \geq f_s\,\omega_c^{\dot{x},\dot{y}} \tag{4.12}$$
$$\omega_c^{\psi} \geq f_s\,\omega_c^{\dot{x},\dot{y}} \tag{4.13}$$
$$\omega_c^{\dot{x},\dot{y}} \geq f_s\,\omega_c^{x,y} \tag{4.14}$$
$$\underline{\omega} \leq \omega_c \leq \overline{\omega} \tag{4.15}$$

The constraints (4.6) and (4.7) set the altitude controller poles hierarchy, stating that both ω_c^z and $\omega_c^{z\,int}$ must be greater than $\omega_c^{\dot{z}}$. In other words, we select $\omega_c^{\dot{z}}$ as the closed-loop bandwidth for the inner loop, since it is the slowest: this choice has been made to limit the control action on the altitude speed, which is directly related to the thrust U_1.

With inequalities (4.8) and (4.9), we limit the attitude controller closed loop poles to be a fraction $n \in [0, 1]$ of $\omega_c^{\dot{z}}$, in order to achieve moderate values for U_2, U_3 and U_4.

The constraints from (4.10) to (4.14) impose the frequency separation between the assigned bandwidths through the user-chosen parameter $f_s \in [5, 10]$. Indeed, the control scheme is designed with a cascade approach, so the inner loop (altitude and attitude controllers) must be faster than the middle one (velocity controller), that in turn must have a bandwidth larger than the outer loop (position controller): in this way, we can neglect the dynamics of the innermost loop when tuning the outermost.

Now, we define matrix C_{con} and vector d_{con} in order to write these constraints in matrix form as $C_{con}\,\omega_c \geq d_{con}$

$$C_{con} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & n & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & n & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -f_s & 0 \\ 0 & 1 & 0 & 0 & 0 & -f_s & 0 \\ 0 & 0 & 1 & 0 & 0 & -f_s & 0 \\ 0 & 0 & 0 & 1 & 0 & -f_s & 0 \\ 0 & 0 & 0 & 0 & 1 & -f_s & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -f_s \end{bmatrix} \qquad d_{con} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.16}$$

In (4.15), we impose the upper and lower bounds on ω_c. The constant parameters $\overline{\omega}$ and $\underline{\omega}$ can be seen as tunable parameters, since are strictly related both to the control effort (too high values can bring to unnecessary power consumption) and to the quality of the system performance.

Let's now compute the matrix inequality elements C and d in (4.2) implementing the bounds in Expression (4.16)

$$C = \begin{bmatrix} C_{con} \\ I \\ -I \end{bmatrix} \qquad d = \begin{bmatrix} d_{con} \\ \underline{\omega} \\ -\overline{\omega} \end{bmatrix} \tag{4.17}$$

where $I \in \mathbb{R}^{7\times 7}$ is the identity matrix.

4.4. Solver

The optimization problem in (4.1) can be classified as constrained Non-Linear Program (NLP), since the cost function (4.5) is not linear with respect to ω_c. Indeed, $f(\omega_c)$ depends on signals obtained by integrating the non-linear model of the system exposed in Chapter 2: at each function call, the simulation is executed via *Simulink*, taking as inputs the drone reference position and the wind vector from the database and returning the values of x^{sim}, y^{sim}, z^{sim} and U_1^{sim}. To achieve enough generality, the optimization is carried out on two different datasets, namely *CW01* and *CW02*.

To solve the NLP, we choose to employ the Sequential Quadratic Programming (SQP)

approach. With this technique, the non-linear problem is locally approximated as a Quadratic Program (QP), in which the cost function $f(\omega_c)$ is approximated by a quadratic function and the constraints are linearized. The solution to such a QP is provided by the Matlab state-of-art solver `quadprog` and yields to search directions for the optimization variables. Then, a line search sub-routine computes the next iterate, using a merit function approach to find a decreasing direction for the cost function that however ensures the feasibility of the new point (with respect to constraints).
Once the direction is detected, a back-tracking strategy is employed to adjust the step size until the Armijo condition is satisfie, obtaining a new feasible point. Around the latter, a new approximated QP is derived and solved, and so on until the user-chosen convergence criteria are satisfied, *i.e.* a optimal set of variables is found.
The SQP algorithm strongly depends on derivatives computation, due to the fact that each sub-routine involves the gradient and/or the Hessian of cost function and constraints, while the optimality evaluation of the iterations is related to the Karush-Kuhn-Tucker (KKT) conditions.
Given the difficulty of obtain analytical derivatives of such a non-linear coupled system, derivatives can be numerically computed with various techniques. In this case, we choose a Forward Finite Difference (FFD) method and we set a step size of 0.1: indeed, since the optimization variable ω_c are related to the control parameters, a too small value would not lead to significant perturbations and drastically augment the necessary computational time.

Then, to improve computational efficiency Equation (4.5) can be recast in a convenient form introducing four performance indexes:

$$\begin{aligned} J_x(\omega_c) &= w_x(x_G^{sim}(\omega_c) - \tilde{x}_G) \\ J_y(\omega_c) &= w_y(y_G^{sim}(\omega_c) - \tilde{y}_G) \\ J_x(\omega_c) &= w_z(z_G^{sim}(\omega_c) - \tilde{z}_G) \\ J_u(\omega_c) &= w_u(U_1^{sim}(\omega_c) - \tilde{U}_1) \end{aligned} \tag{4.18}$$

Now, we can write the cost function (4.5) as

$$f(\omega_c) = J_x(\omega_c)^T J_x(\omega_c) + J_y(\omega_c)^T J_y(\omega_c) + J_z(\omega_c)^T J_z(\omega_c) + J_u(\omega_c)^T J_u(\omega_c) \tag{4.19}$$

Then we stack the four indexes to produce a new array $F(\omega_c)$

$$F(\omega_c) = \begin{bmatrix} J_x(\omega_c) \\ J_y(\omega_c) \\ J_z(\omega_c) \\ J_u(\omega_c) \end{bmatrix} \tag{4.20}$$

Finally we obtain the new formulation of the cost function as

$$f(\omega_c) = F(\omega_c)^T F(\omega_c) \tag{4.21}$$

With $f(\omega_c)$ expressed in this form, the problem is suitable to be solved using the Gauss-Newton approach for the Hessian computation. Indeed, computing the gradient of $f(\omega_c)$ in the form of Equation (4.21) by using the chain rule of differentiation we obtain

$$\nabla_{\omega_c} f(\omega_c) = 2\, \nabla_{\omega_c} F(\omega_c)\, F(\omega_c) \tag{4.22}$$

where $\nabla_{\omega_c}(\cdot)$ indicates the gradient with respect to ω_c. Then, differentiating again Equation (4.22) we obtain an expression for the Hessian matrix of $f(\omega_c)$

$$\nabla^2_{\omega_c} f(\omega_c) = \nabla_{\omega_c} F(\omega_c)\, \nabla_{\omega_c} F(\omega_c)^T + 2 \sum_{i=1}^{N} \nabla^2_{\omega_c} F_i(\omega_c)\, F_i(\omega_c) \tag{4.23}$$

where $\nabla^2_{\omega_c}(\cdot)$ is the Hessian and N is the number of optimization variables.
This expression is made by two terms, where the second features a sum of products between the Hessian of each component of vector $F(\omega_c)$ and the component itself. Now, assuming that, close to a local minimizer ω_c^{opt}, it holds $F(\omega_c^{opt}) \approx 0$, we can approximate Equation (4.23) as

$$\nabla^2_{\omega_c} f(\omega_c) \approx \nabla_{\omega_c} F(\omega_c)\, \nabla_{\omega_c} F(\omega_c)^T \tag{4.24}$$

In this way, we obtain an approximation for the Hessian matrix that only depends on the gradient, *i.e.* we can compute the Hessian without actually deriving twice the cost function. The advantage is easily understandable if we think that the numerical error introduced by using the FFD for the gradient computation is not incremented by obtaining also the second derivative with FFD .
Furthermore, in the SQP local approximation, the Gauss-Newton method is further accurate thanks to the fact that constraints (4.2) are linear and so the second derivative is zero (see [40] for more).

5 | Results

In the following Chapter we discuss the output of the rotors coefficients identification exposed in Chapter 2.2.1 and the results given by the solution of optimization program (4.1). About the latter, we focus on the performance improvements brought by the procedure described in Chapter 4, together with the employed validation approach for the results.

5.1. Rotor Coefficients Fitting Results

The rotor lift and drag coefficients are identified following the procedure described in Chapter 2.2.1, providing as result

$$b = 0.1385 \ \ kg \cdot m \quad d = 0.6611 \ \ s \cdot m \cdot N \tag{5.1}$$

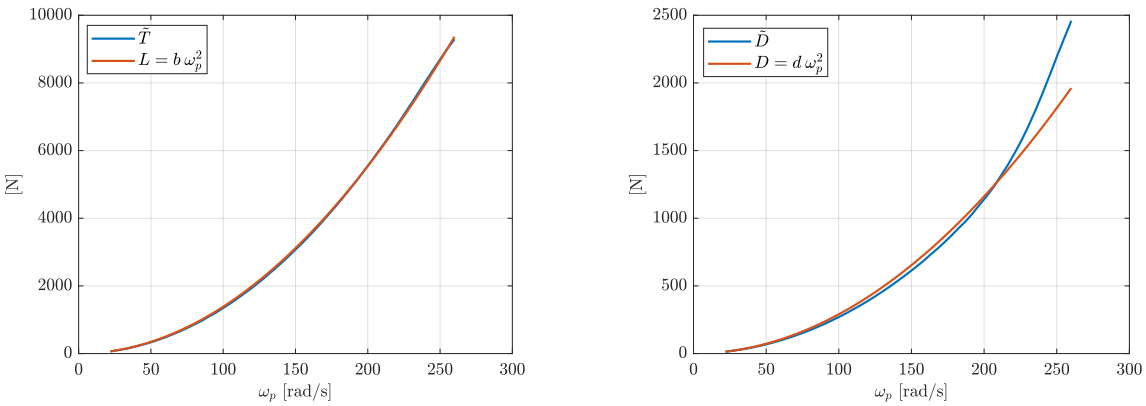


Figure 5.1: Comparison between Makani propellers data and the identified quadratic model for the lift force (left picture) and drag torque (right picture)

As we can see from the left picture of Figure 5.1, the quadratic model (2.17) fits the experimental thrust $\tilde{T}$ quite accurately for all the considered range of propellers speed ω_p. On the other hand, the model (2.18) is not that precise for the drag torque, especially at

high rotor speeds, as depicted in the right picture of Figure 5.1. Nevertheless, we decided to adopt this model to remain consistent with the literature (*c.f.* [37]) and to preserve the model simplicity, but limiting the maximum propeller speed to $\bar{\omega}_p = 210\, rad/s$, since up to this velocity we achieve a sufficient level of accuracy from such a simple model.
The parameter $\bar{\omega}_p$ is ultimately used in the computation of the saturation limits for the thrust control action U_1, as defined in Equation (2.35).

5.2. NLP Solution

In this Section, we analyze the results of the optimization procedure described in Chapter 4. First of all, let's assign values to the parameters of the constraints matrix (4.16):

Parameter	Value	Unit
n	0.9	–
f_s	5	–
$\bar{\omega}$	5	rad/s
$\underline{\omega}$	0.05	rad/s

Table 5.1: Constraints parameters for the optimization problem (4.1).

The optimization algorithm is initialized at point $\omega_c^0 = [1.39\, 1.39\, 1.39\, 1.25\, 1.25\, 0.25\, 0.05]^T$, which is computed in order to respect the constraints and stick to the lower bound. In other words, the solver starts from the point with the slowest possible bandwidths for the control system.
Figure 5.2 shows the reference tracking performance of the system on the identification dataset *CW01* when $\omega_c = \omega_c^0$: the results are satisfactory on the z coordinate, while on the other two axes the tracking is highly inaccurate. To quantify this aspect, we resort to the root mean square error (RMSE), computed as

$$\begin{aligned} RMSE_x(\omega_c) &= \sqrt{\frac{1}{N}\sum_{i=0}^{N}(\tilde{x} - x^{sim}(\omega_c))^2} \\ RMSE_y(\omega_c) &= \sqrt{\frac{1}{N}\sum_{i=0}^{N}(\tilde{y} - y^{sim}(\omega_c))^2} \\ RMSE_z(\omega_c) &= \sqrt{\frac{1}{N}\sum_{i=0}^{N}(\tilde{z} - z^{sim}(\omega_c))^2} \end{aligned} \tag{5.2}$$

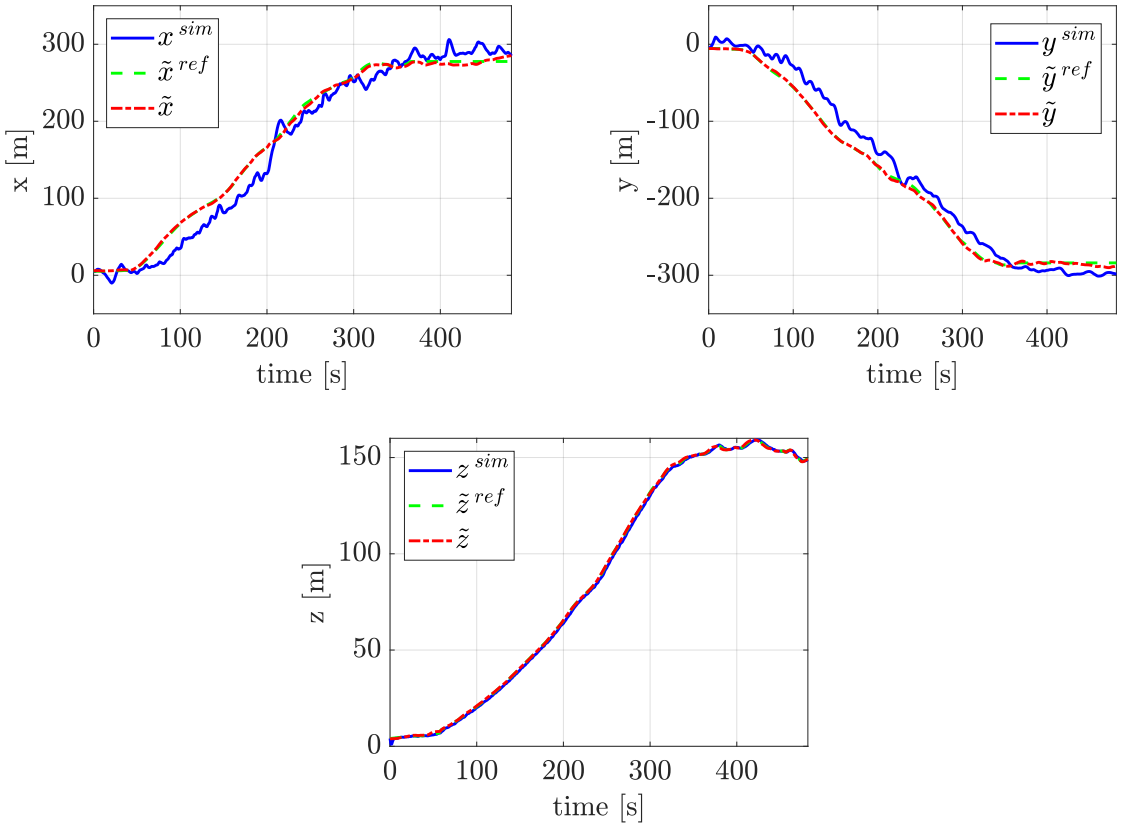


Figure 5.2: Reference Tracking with ω_c^0 on the identification dataset *CW01*

For $\omega_c = \omega_c^0$, the RMSE values are

$$\begin{aligned} RMSE_x(\omega_c^0) &= 17.9418\, m \\ RMSE_y(\omega_c^0) &= 18.5712\, m \\ RMSE_z(\omega_c^0) &= 1.0840\, m \end{aligned} \tag{5.3}$$

Now, by running the solver as described in Section 4.4, we find a a local minimizer of the cost function (4.5) that gives the optimal value ω_c^{opt}

$$\omega_c^{opt} = \begin{bmatrix} 3.58 & 3.58 & 3.63 & 3.22 & 3.22 & 0.64 & 0.13 \end{bmatrix}^T \tag{5.4}$$

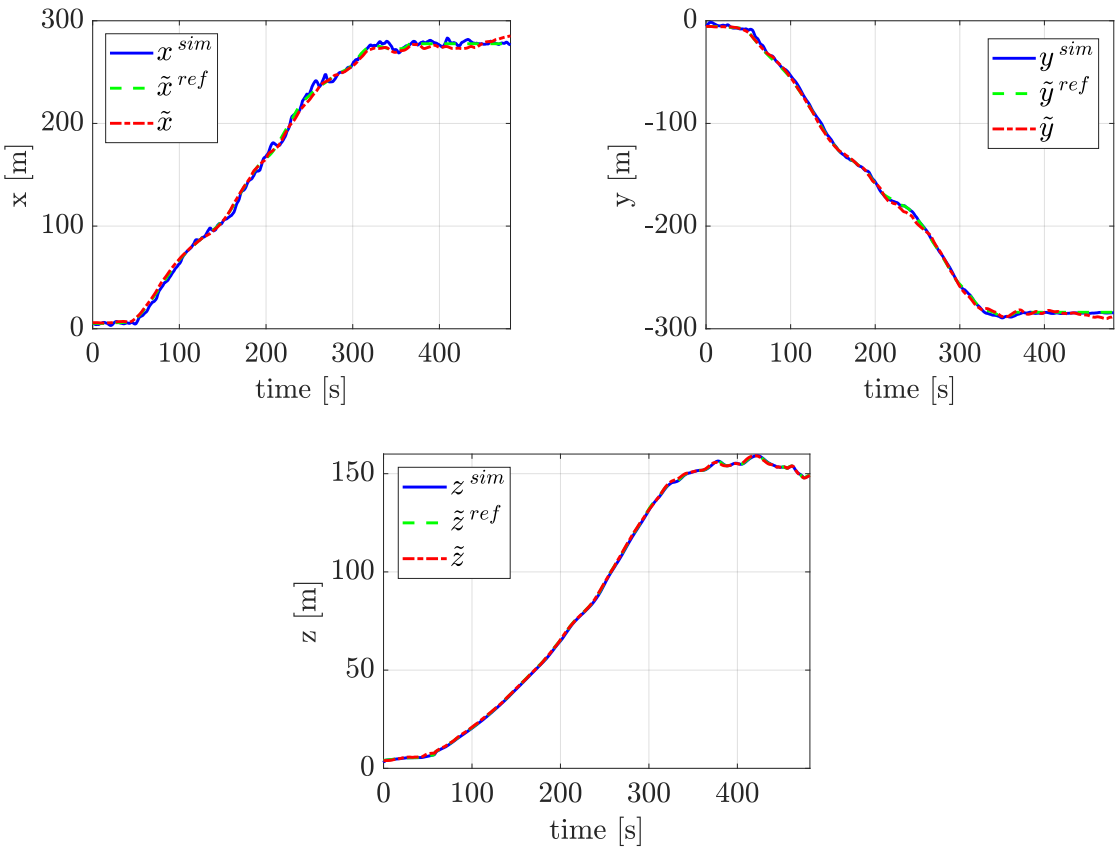


Figure 5.3: Reference tracking with ω_c^{opt} on the identification dataset *CW01*

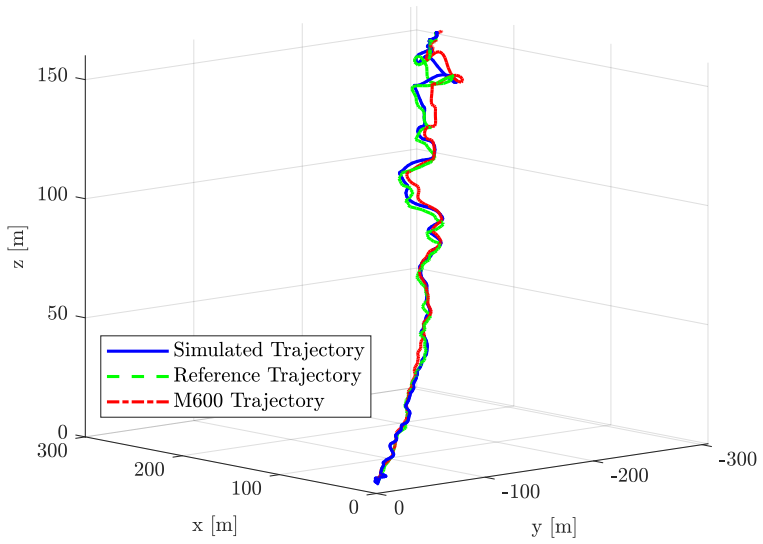


Figure 5.4: 3D representation of the reference tracking with ω_c^{opt} on the identification dataset *CW01*

This solution corresponds to the performance depicted in Figure 5.3 and 5.4, that shows a significant improvement of the reference tracking: indeed, the RMSE for the x and y coordinates drastically falls under 5 m and the already small value of $RMSE_z$ further decreases to less than 1 m (Figure 5.5),

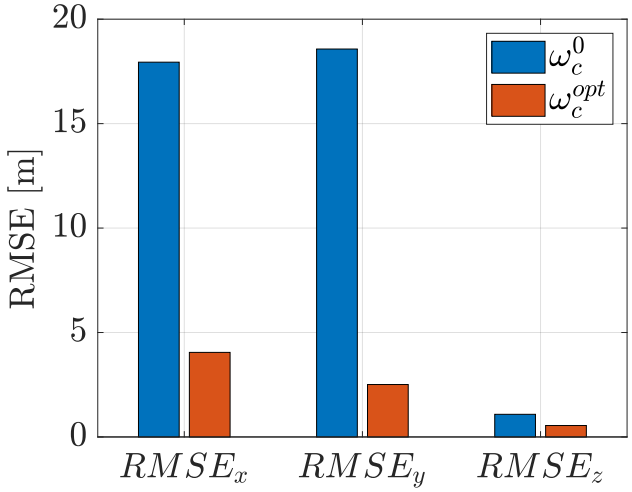


Figure 5.5: RMSE comparison between starting and optimal control parameters

Figure 5.6 shows the time course of the thrust command U_1 with ω_c^{opt} on the identification dataset *CW01*. After an initial transient, the thrust control variable converges towards a steady state value close to $20\,kN$, in order to balance the weight force acting on the drone (*i.e.* $m \cdot g \approx 17\,kN$) and the high frequency aerodynamic forces that acts as disturbances on the system.

In the same way, Figure 5.7 reports the trend over time of the moments commands U_2, U_3 and U_4: the presence of the wind considerably influences the control variables, which continuously try to adjust the drone orientation to keep tracking the reference signals despite the wind gusts.

This behaviour can be identified also by looking at the eight propellers speeds depicted in Figure 5.8, where the values of $\omega_p = [\omega_1\ \omega_2\ \omega_3\ \omega_4\ \omega_5\ \omega_6\ \omega_7\ \omega_8\ ]^T$ are computed as exposed in Section 2.2.1.

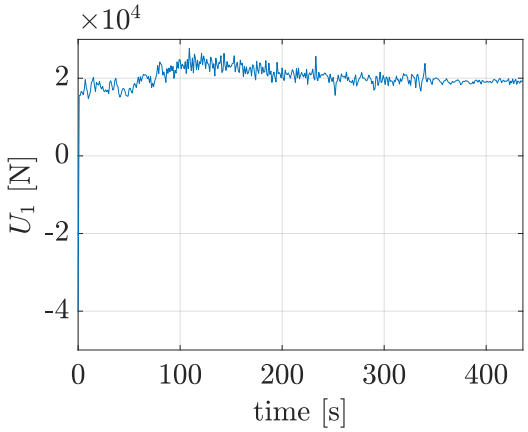


Figure 5.6: Time course of the thrust command U_1 with ω_c^{opt} on the identification dataset *CW01*

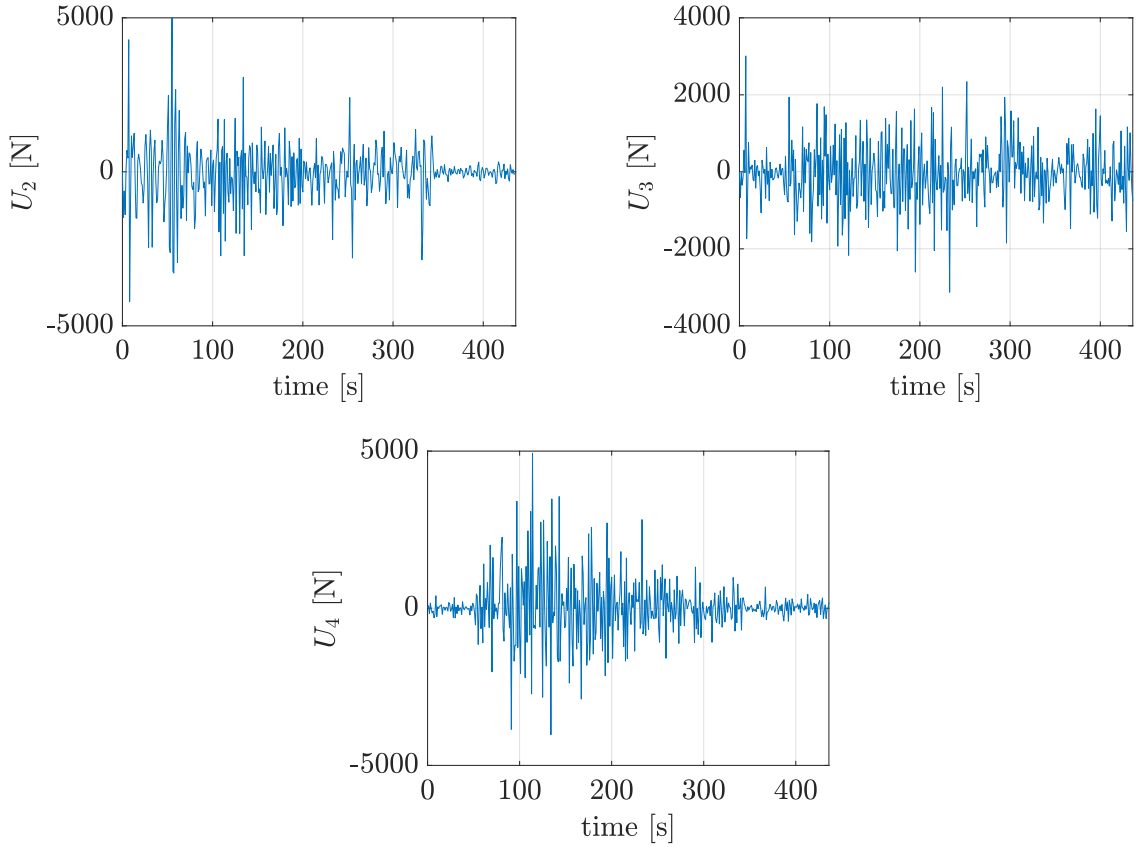


Figure 5.7: Time course of the moment commands U_2 (left picture), U_3 (central picture) and U_4 (right picture) with ω_c^{opt} on the identification dataset *CW01*

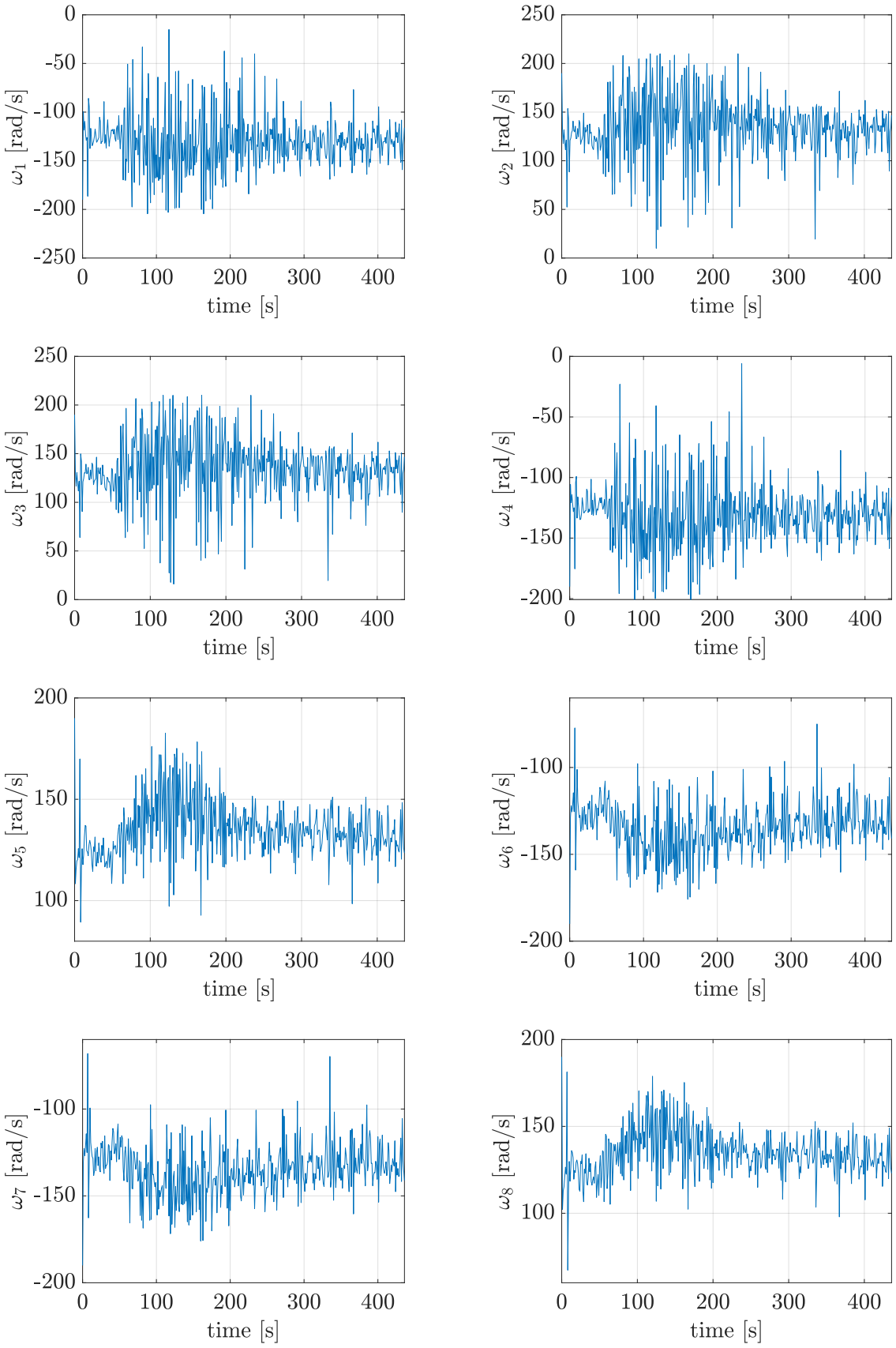


Figure 5.8: Propellers speed time course with ω_c^{opt} on the identification dataset *CW01*

5.3. Validation

The results presented in the previous section are validated exploiting the other datasets of the *CW* flight program.
The validation procedure consists in testing the control system performance with $\omega_c = \omega_c^{opt}$ on the datasets from *CW05* to *CW10* and comparing the obtained RMSE values with the results given by the *CW01* and *CW02* scenarios.
Figure 5.9 shows the RMSE index for the three position coordinates: we can see that the validation data are in the same order of magnitude of the values obtained for the optimization datasets, even if some flights data generate a higher RMSE. Indeed, for instance, when *CW06* is employed we see an increase of the $RMSE_y$ index: however, analyzing the tracking performance (depicted in Figure 5.10), we understand that the simulator is actually following quite accurately the reference, while the real drone is not precisely tracking $\tilde{y}^{ref}$, so the index as it is defined in (5.3) increases.

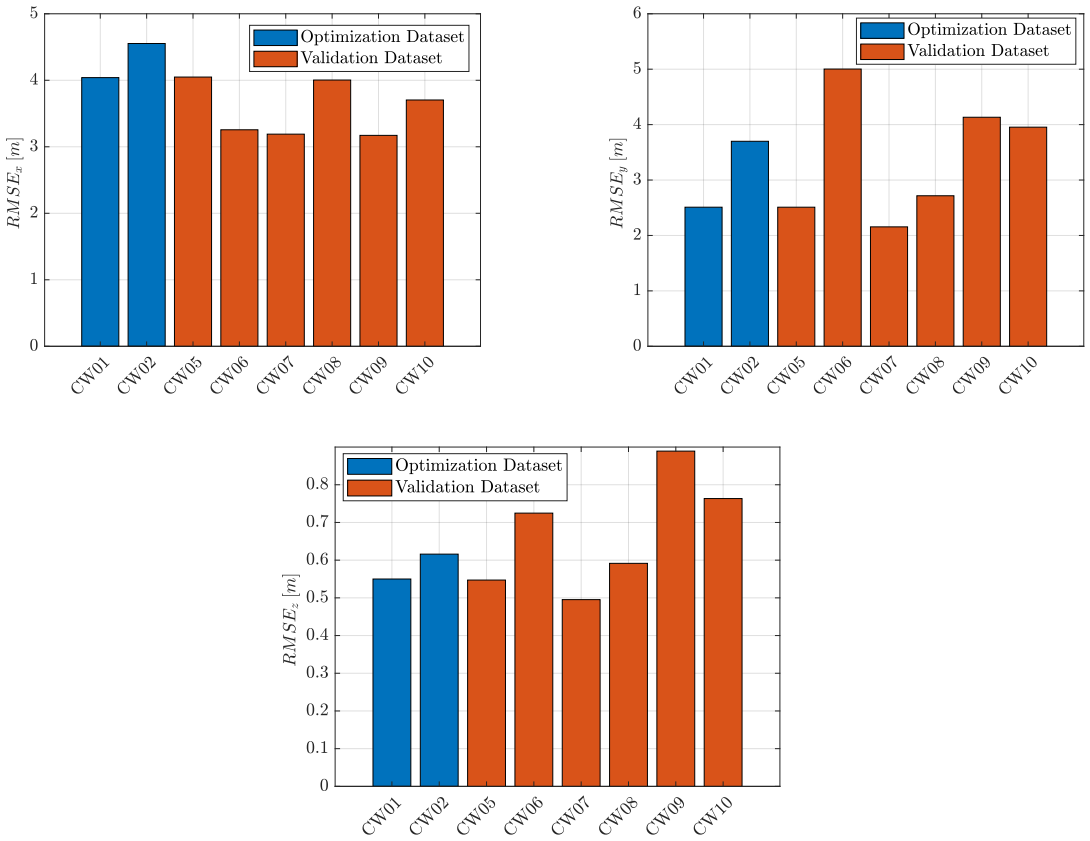


Figure 5.9: RMSE comparison between the optimization and validation datasets

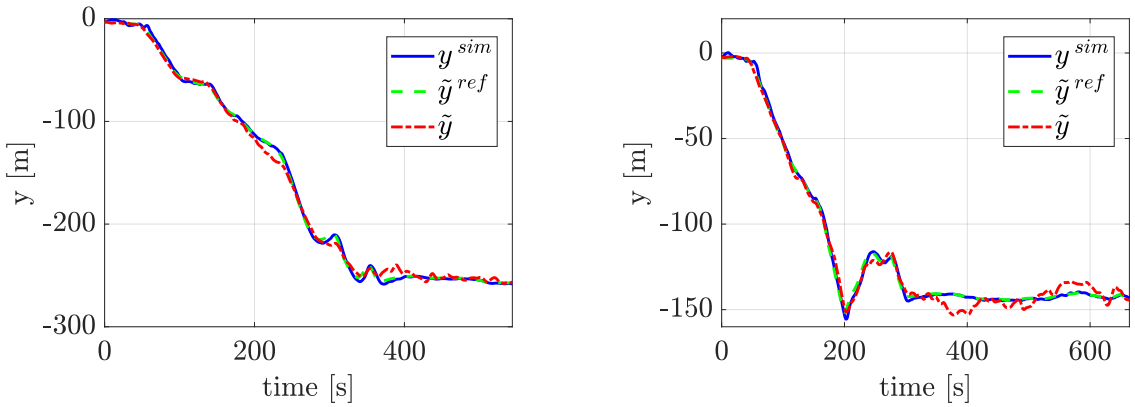


Figure 5.10: y-axis reference tracking with data from the validation datasets *CW06* (left) and *CW09* (right)

5.4. Discrete-Time Results

Finally, the optimized parameters are implemented in the discrete-time control system described in Section 2.6.
First of all, we have to select the sampling frequency f_s: given that the fastest bandwidth identified in Section 5.2 corresponds to $3.58\,rad/s$ (see Equation (5.4)), we choose f_s high enough to be sure that all the meaningful dynamics are correctly captured. After some experiments on the discretized control scheme, we select

$$f_s = 100\ Hz \tag{5.5}$$

We can assess the performance of the discretized control system analyzing Figure 5.11, that shows the RMSE values of the indexes on the three axes obtained by running the simulation on the identification dataset *CW01*: with the selected value of f_s, the controllers behave almost in the same way of their continuous counterparts, obtaining again satisfactory tracking performance with quite low RMSE values.
For completeness, Figure 5.12 shows the RMSE values obtained with the discretized control scheme when the validation datasets (described in Section 5.3) are used to run the simulator.

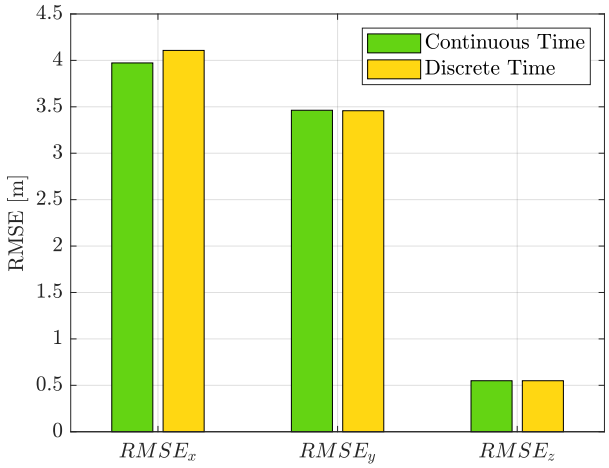


Figure 5.11: Comparison between the RMSE values obtained with the continuous-time and the discretized controllers on the identification dataset *CW01*

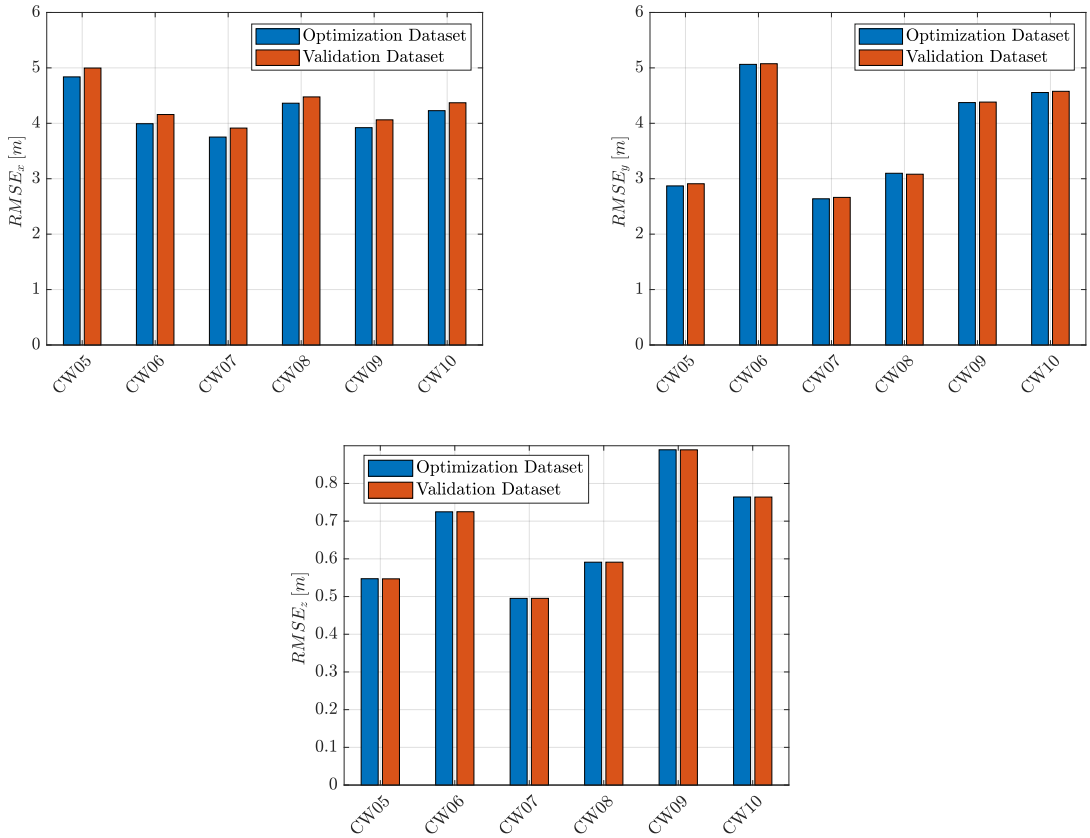


Figure 5.12: RMSE comparison between the optimization and validation datasets with the discrete-time control system

6 | Conclusions and Future Development

The goal of this Thesis was the in-depth study of an Airborne Wind Energy system with On-Board Generation, for which no established models nor control techniques are nowadays well defined as in the case of Ground Level Generation AWE plants. In particular, we consider the M600 system by Makani, the first and only OBG system for which flight data are publicly available.

In the work development, we proposed a non-linear model for the AWE drone, taking as reference a standard octocopter and adapting the dynamical equations to the problem at hand. Furthermore, we suggested an innovative approach for the aerodynamic force modelling based on HAWTs empirical results, in order to characterize the lift and drag coefficients over 360°.

Then, we developed a linear cascaded control scheme for the kite, regulating its attitude and position via three nested loops, together with an allocation system that translates the control variables into eight propellers speeds.

For completeness, we added an accurate winch model for the ground station and we used an advanced tether model (available from the literature) to achieve enough accuracy and provide high-fidelity results.

Finally, we described how to optimally tune the kite control system to emulate the real M600 behaviour, following a data-driven approach that exploits the database released by Makani.

The results shown in Chapter 5 proved that this approach is valid and can be used as a starting point for future improvements. Surely, the most straightforward one is the extension of the optimization procedure to the plane-like flight of the drone, allowing the simulation of the generation phase and the study of power output. The allocation technique could be also improved by, for example, introducing more accurate models for the propellers, to take into accounts the dynamical behaviour of a so crucial part of the system.

Bibliography

[1] European Parliament and Council. Regulation (eu) 2021/1119 of 30 june 2021 establishing the framework for achieving climate neutrality and amending regulations (ec) no 401/2009 and (eu) 2018/1999 ('european climate law'). https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32021R1119, 2021.

[2] Commission To The European Parliament, The European Council, The European Economic And Social Committee And The Committee Of The Regions. The european green deal. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=COM:2019:640:FIN, 2019.

[3] IEA. Net zero by 2050 - a roadmap for the global energy sector. *IEA*, 2021.

[4] BP p.l.c. Statistical review of world energy. Technical report, 1 St James's Square,London SW1Y 4PD,UK, 2021.

[5] Global Wind Energy Council. Global wind report 2021. Technical report, Rue Belliard 51-53,1000 Brussels, Belgium, 2021.

[6] International Renewable Energy Association. Global renewables outlook: Energy transformation 2050. Technical report, Masdar City P.O. Box 236, Abu Dhabi, United Arab Emirates, 2020.

[7] Udo Zillmann and Philip Bechtle. *Emergence and Economic Dimension of Airborne Wind Energy*, pages 1–25. 04 2018.

[8] M. L. Loyd. Crosswind kite power. *Journal of Energy 4*, 3:106–111, 1980.

[9] KiteKraft. https://www.kitekraft.de/.

[10] Eber Jochem, Melinda Marquis, Aubryn Cooperman, Caroline Draxl, Rob Hammond, Jason Jonkman, Alexsandra Lemke, Anthony Lopez, Rafael Mudafort, Mike Optis, Owen Roberts, and Matt Shields. *Airborne Wind Energy*. National Renewable Energy Laboratory. NREL/TP-5000-79992, Golden, CO, 2021.

[11] Kitepower. https://thekitepower.com/.

[12] Philip Bechtle, Mark Schelbergen, Roland Schmehl, Udo Zillmann, and Simon Watson. Airborne wind energy resource analysis. *Renewable Energy*, 141:1103–1116, Oct 2019.

[13] Nicola Jones. After a shaky start, airborne wind energy is slowly taking off. *Yale Environment 360*, 2022.

[14] Ampyx Power. `https://www.ampyxpower.com/`.

[15] SkySails Power. `https://skysails-power.com/`.

[16] S. Mann, Dr. K. Gunn, Prof. G. Harrison, Dr. B. Beare, and Dr. I. Lazakis. Wind yield assessment for airborne wind energy. *EWEA Offshore 2015*, 2015.

[17] Kitenergy. `https://kitenrg.com/`.

[18] Kitemill. `https://www.kitemill.com/`.

[19] TWingTec. `https://twingtec.ch/`.

[20] WindLift. `https://windlift.com/`.

[21] Makani Technologies LLC. `https://x.company/projects/makani/`.

[22] Chris Vermillion, Mitchell Cobb, Lorenzo Fagiano, Rachel Leuthold, Moritz Diehl, Roy Smith, Tony Wood, Sebastian Rapp, Roland Schmehl, David Olinger, and Michael Demetriou. Electricity in the air: Insights from two decades of advanced control research and experimental flight testing of airborne wind energy systems. *Annual Reviews in Control*, 04 2021.

[23] Tony Wood, Henrik Hesse, Max Polzin, Eva Ahbe, and Roy Smith. Modeling, identification, estimation and adaptation for the control of power-generating kites. *IFAC-PapersOnLine*, 51:981–989, 01 2018.

[24] Massimo Canale, Lorenzo Fagiano, and Mario Milanese. High altitude wind energy generation using controlled power kites. *Control Systems Technology, IEEE Transactions on*, 18:279 – 293, 04 2010.

[25] Mario Zanon, Sébastien Gros, Joel Andersson, and Moritz Diehl. Airborne wind energy based on dual airfoils. *IEEE Transactions on Control Systems Technology*, 21(4):1215–1222, 2013.

[26] Davide Todeschini, Lorenzo Fagiano, Claudio Micheli, and Aldo Cattano. Control of a rigid wing pumping airborne wind energy system in all operational phases. *Control Engineering Practice*, 111:104794, 2021.

[27] SkyPull. `https://www.skypull.technology/solution`.

[28] Davide Todeschini, Lorenzo Fagiano, Claudio Micheli, and Aldo Cattano. Control of vertical take off, dynamic flight and landing of hybrid drones for airborne wind energy systems. pages 2177–2182, 2019.

[29] Eric Nguyen Van, Lorenzo Fagiano, and Stephan Schnez. Autonomous take-off and landing of a tethered aircraft: a simulation study. 10 2015.

[30] Lorenzo Fagiano and Stephan Schnez. On the take-off of airborne wind energy systems based on rigid wings. *Renewable Energy*, 107:473–488, 2017.

[31] Paul Williams, S. Sieberling, and Richard Ruiterkamp. Flight test verification of a rigid wing airborne wind energy system. pages 2183–2190, 07 2019.

[32] Filippo Trevisi, Mac Gaunaa, and Michael McWilliam. Unified engineering models for the performance and cost of ground-gen and fly-gen crosswind airborne wind energy systems. *Renewable Energy*, 162:893–907, 2020.

[33] Mark Aull and Kelly Cohen. A nonlinear inverse model for airborne wind energy system analysis, control, and design optimization. *Wind Energy*, 24(2):133–148, 2021.

[34] Paula Echeverri, Tobin Fricke, Geo Homsy, and Nicholas Tucker. The energy kite: Selected results from the design, development, and testing of makani's airborne wind turbines, part 1 of 3. *Makani Technologies LLC*, 2020.

[35] Paula Echeverri, Tobin Fricke, Geo Homsy, and Nicholas Tucker. The energy kite: Selected results from the design, development, and testing of makani's airborne wind turbines, part 1 of 3. *Makani Technologies LLC*, 2020.

[36] Paula Echeverri, Tobin Fricke, Geo Homsy, and Nicholas Tucker. The energy kite: Selected results from the design, development, and testing of makani's airborne wind turbines, part 3 of 3. *Makani Technologies LLC*, 2020.

[37] Lorenzo Fagiano. Systems of tethered multicopters: Modeling and control design. *IFAC-PapersOnLine*, 2017. 20th IFAC World Congress.

[38] Andrea Berra. Optimal control of pumping airborne wind energy systems without wind speed feedback. Master's thesis, Politecnico di Milano, 2020.

[39] Sandia National Laboratories. Aerodynamic characteristics of seven symmetrical airfoils section through 180-degree angle of attack for use in aerodynamic analysis in vertical axes wind turbines. Technical report, Sandia National Laboratories, Albuquerque, NM 87185, 1981.

[40] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, NY, USA, 2e edition, 2006.

A | Appendix A

In this appendix, we report the numerical values of the subsystems parameters.

A.1. M600

Parameter	Symbol	Value	Unit
Mass	m	1730.8	kg
Inertia	J	$\begin{bmatrix} 32734 & 23 & 37 \\ 23 & 9963 & 18 \\ 37 & 18 & 32734 \end{bmatrix}$	$kg \cdot m^2$
Wing Area	S	32.9	m^2
Wing Span	-	25.66	m

Table A.1: Makani M600 Parameters (from [34])

A.2. Tether

Parameter	Symbol	Value	Unit
Tether Drag Coefficient	$C_{D,T}$	0.7	–
Linear Density	ρ_{lin}	0.917	kg/m
Diameter	d_T	29.4	mm
Length	L	425.8	m
Young's Modulus	E	26.5	GPa
Stiffness Constant	K_{el}	$4.22 \cdot 10^6$	N/m

Table A.2: Tether Parameters (from [34])

In [34], it is provided the Young's Modulus per Area as $EA = 18\,10^6\,N$. From this value, we computed the stiffness constant as $K_{el} = \frac{EA}{L}$ and we obtained the Young's Modulus

value in Table A.2 by multiplying EA by the tether area, as $E = EA\,\pi\,\frac{d_T}{4}^2$.

A.3. Ground Station

Parameter	Symbol	Value	Unit
Winch Mass	m_w	9500	kg
Drum Inner Radius	r_1	1.7	m
Drum Outer Radius	r_2	1.9	m
Drum Inertia	J_{drum}	30875	$kg \cdot m^2$
Friction Coefficient	β_{drum}	0.8	–

Table A.3: Makani Ground Station Parameters (from [34])

The data in Table A.3 are all available in the Makani database, except for β_{drum} that was arbitrarily fixed to a reasonable value.

A.4. Propellers

Parameter	Symbol	Value	Unit
Rotor Lift coefficient	b	0.1385	$kg \cdot m$
Rotor Drag Coefficient	d	0.6611	$s \cdot m \cdot N$
Propeller Inertia	J_p	0.9	$kg \cdot m^2$

Table A.4: Makani M600 Propellers Parameters (from [34])

$\omega\,[rad/s]$	Thrust $[N]$	Power $[W]$
22.00	64.13	315
26.86	95.95	567
31.71	133.95	928
36.57	177.65	1418
41.43	228.95	2048
46.29	285.95	2846
51.14	348.65	3833
56.00	418.95	5019
60.86	494.95	6426
65.71	577.60	8075
70.57	666.90	9996
75.43	761.90	12180
80.29	864.50	14700
85.14	969.00	17535
90.00	1092.50	20685
94.86	1206.50	24255
99.71	1339.50	28140
104.57	1472.50	32445
109.43	1615.00	37275
114.29	1767.00	42420
119.14	1919.00	48090
124.00	2080.50	54285
128.86	2251.50	61005
133.71	2432.00	68250
138.57	2612.50	76125

$\omega\,[rad/s]$	Thrust $[N]$	Power $[W]$
143.43	2802.50	84525
148.29	3002.00	93555
153.14	3201.50	103215
158.00	3410.50	113400
162.86	3629.00	124950
167.71	3857.00	136500
172.57	4094.50	149100
177.43	4332.00	162750
182.29	4579.00	177450
187.14	4835.50	192150
192.00	5092.00	208950
196.86	5367.50	227850
201.71	5643.00	246750
206.57	5928.00	267750
211.43	6213.00	291900
216.29	6507.50	317100
221.14	6811.50	345450
226.00	7125.00	375900
230.86	7438.50	410550
235.71	7761.50	448350
240.57	8084.50	489300
245.43	8398.00	533400
250.29	8711.50	579600
255.14	9015.50	625800
260.00	9291.00	673050

Table A.5: Propeller Database (from [34])

List of Figures

List of Tables