POLITECNICO MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE

EXECUTIVE SUMMARY OF THE THESIS

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

LAUREA MAGISTRALE IN AUTOMATION AND CONTROL ENGINEERING

Author: GIACOMO CALCIOLARI

Advisor: PROF. LORENZO MARIO FAGIANO

Academic year: 2020-2021

1. Introduction

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 600 kW of rated power and was the first ever AWE system able to complete an offshore flight.

1.1. AWE Principles

Airborne Wind Energy (AWE) is a novel technology based on rigid or soft kites that autonomously fly in crosswind 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. Nowadays, there are two main working principles for the AWE systems: the ground-level generation (GLG) and the on-board generation (OBG).

In GLG systems, the kite operates in pumping cycles. In the first phase of the cycle, the kite pulls the tether thanks to the lift force acting on the wing: in turn, the tether unwinds from the ground station winch and drives the generator, producing power. During the second phase, a small fraction of power is instead consumed to reel in the tether and move the kite at the initial position to restart the cycle.

On the other hand, OBG systems uses generators located on the kite to harvest wind power, that is then trasmitted to the ground station via the tether, that in this case is made of conductive material.

The main feature of the OBG technology is the

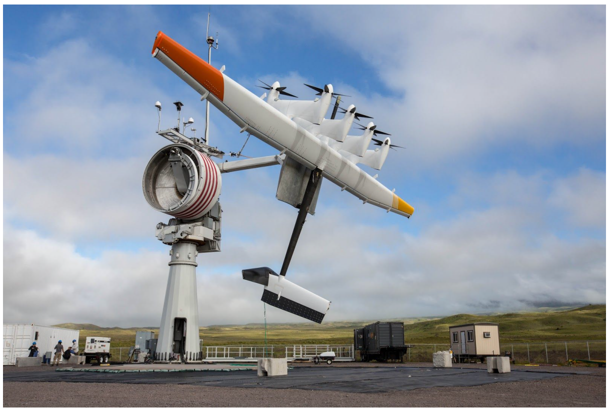

Figure 1: Makani M600 System

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 (VTOL) operations, where the propellers actively push the drone in the air, similarly to the commercial drones take-off.
In the last years, AWE research has gone in the direction of GLG systems, proposing many approaches for the modelling and control of these systems that can be easily and rapidly prototyped, unlike OBG systems that are typically heavier and involves more complex dynamics.

1.2. Objectives

This work aims to provide a new approach for the modelling and control of OBG systems, which represent an AWE branch of that has not been deeply investigated yet. The model employs the dynamical equations of a octocopter as described in [1], 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 consists in 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 and documented in [2] [3] [4] in 2020. The final outcome of the work is an optimally-tuned simulator, able to emulate the working activities of the M600 OBG drone in the initial hovering phase and validated on the actual flight data.

2. Modelling

The considered M600 system is composed by three main elements: the drone (also referred as *kite*), the tether and the ground station.
The M600 drone autonomously launches from the ground station in a vertical take-off and landing (VTOL) configuration and hovers as a common octocopter as the tether is payed out by the winch 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 in a traditional-plane configuration, 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 a perch on the ground station.

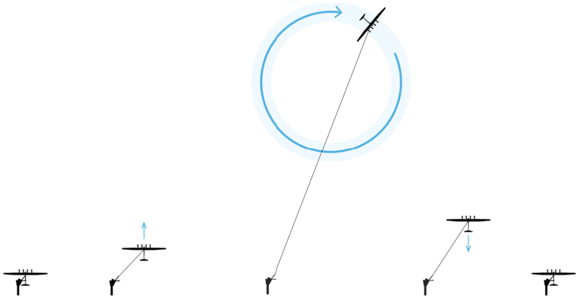

Figure 2: M600 operative cycle. From left to right: autonomous VTOL takeoff, hover/trans-in maneuver, generation phase, trans-out maneuver, VTOL landing on perch.

2.1. Kite

The kite under study is the Makani M600 model SN4, well described in [2] and for which flight data are available. The M600 is a rigid-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.

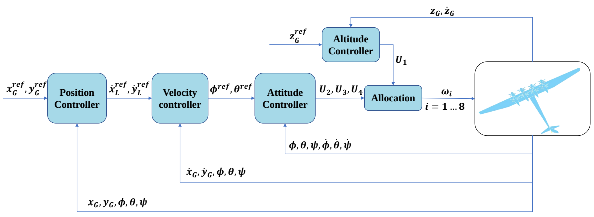
$z_G, \dot{z}_G$
z_G^{ref}
Altitude Controller
U_1
x_G^{ref}, y_G^{ref}
Position Controller
$\dot{x}_L^{ref}, \dot{y}_L^{ref}$
Velocity controller
ϕ^{ref}, θ^{ref}
Attitude Controller
U_2, U_3, U_4
Allocation
ω_i
$i = 1 \dots 8$
$\phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}$
$\dot{x}_G, \dot{y}_G, \phi, \theta, \psi$
$x_G, y_G, \phi, \theta, \psi$

Figure 3: Kite Control Scheme

The employed dynamical model for the kite is the following:

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

$$\dot{p} = \frac{I_y - I_z}{I_x}\, q\, r + \frac{U_2}{I_x} - \frac{J_p}{I_x}\, q\, \Omega_r \quad (2)$$

$$\dot{q} = \frac{I_z - I_x}{I_y}\, p\, r + \frac{U_3}{I_y} + \frac{J_p}{I_y}\, p\, \Omega_r \quad (3)$$

$$\dot{r} = \frac{I_x - I_y}{I_z}\, p\, q + \frac{U_4}{I_z} \quad (4)$$

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, 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.

Finally, R is the time-dependent rotation matrix from the global to the local frame, and is defined as

$$R = \begin{bmatrix} c(\psi)c(\theta) & s(\psi)c(\theta) & -s(\theta) \\ c(\psi)s(\theta)s(\phi) - s(\psi)c(\phi) & s(\psi)s(\theta)s(\phi) + c(\psi)c(\phi) & c(\theta)s(\phi) \\ c(\psi)s(\theta)c(\phi) + s(\psi)s(\phi) & s(\psi)s(\theta)c(\phi) - c(\psi)s(\phi) & c(\theta)c(\phi) \end{bmatrix} \quad (5)$$

where $c(\cdot) = \cos(\cdot)$, $s(\cdot) = \sin(\cdot)$ and the time dependency of R and the Euler angles is omitted for brevity.

The chosen model is highly non-linear, so we tackle the control design problem by taking as reference a linearized model as suggested in [1], to consequently realize a cascaded control system made by three nested loops, as shown in Figure 3.

The inner loop is composed by the drone attitude and altitude controllers, both realized using pole placement, with an integral action added in the altitude scheme. This loop outputs the control variables U_1, U_2, U_3, U_4 that are then translated in propeller speeds by the allocation block (see Figure 3).

Then, the middle loop controls the drone velocity on the x and y body axes with again a pole placement approach.

Ultimately, the position of the drone on the global x and y axes is regulated with a PI technique, equipped with an anti-windup back-calculation scheme.

2.2. Tether

The tether is considered as composed of segments of Kelvin-Voigt material that connect point masses: in this way we consider each point mass node as attached 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, while the gravitational and aerodynamic forces are applied on the nodes (see [5] for further details). Using the model developed in [5], we obtain an high grade of accuracy for a key element of the system, that couples the drone dynamics with the operations of the ground station.

2.3. Ground Station

The ground (or base) station manages the reeling in and the reeling out operations of the tether during the flight maneuvers. We consider the station as a winch driven by an electric motor that winds and unwinds the tether from the drum. Its dynamical model is obtained using a torque balance, in which we take into account the tether tension, the viscous friction and the motor torque

$$J_{drum}\ddot{\lambda} = \|F_T\|\, r_2 - T_m - \beta_{drum}\, \dot{\lambda} \quad (6)$$

where λ, $\dot{\lambda}$ and $\ddot{\lambda}$ are the winch angular position, velocity and acceleration, J_{drum} is the drum (supposed as an hollow cylinder) inertia, β_{drum} is the viscous friction coefficient and T_m is the motor torque.

The proposed control system for the winch is a pole placement scheme with integrator, to ensure a satisfactory reference tracking. The setpoint for the winch angular position is generated in terms of tether length, which is in turn a function of the drone position as

$$\lambda^{ref} = \frac{L^{ref}}{r} = \frac{\|P_G\| + \Delta_L}{r} \tag{7}$$

where Δ_L is an user-chosen parameter and r is the drum outer radius.

2.4. Aerodynamic Forces

The aerodynamic forces vector acting on the drone

$$\vec{F}_{aero}(\alpha, \beta) = [F_{drag}(\alpha, \beta)\ F_{side}(\alpha, \beta)\ F_{lift}(\alpha, \beta)]^T \tag{8}$$

is computed in the wind frame as

$$\begin{aligned} F_{drag}(\alpha, \beta) &= \frac{1}{2}\rho S C_D(\alpha, \beta)\|\vec{W}_{a,w}\|^2 \\ F_{side}(\alpha, \beta) &= \frac{1}{2}\rho S C_S(\alpha, \beta)\|\vec{W}_{a,w}\|^2 \\ F_{lift}(\alpha, \beta) &= \frac{1}{2}\rho S C_L(\alpha, \beta)\|\vec{W}_{a,w}\|^2 \end{aligned} \tag{9}$$

where ρ is the air density, S is the wing surface, α is the angle of attack, β is the sideslip angle, $C_D(\alpha, \beta)$, $C_S(\alpha, \beta)$, $C_L(\alpha, \beta)$ are the drag, side and lift coefficients and $\|\vec{W}_{a,w}\|$ is the modulus of the apparent wind velocity in the wind frame. In the following, we suppose that $F_{side}(\alpha, \beta)$ is negligible and that $C_D(\alpha, \beta) \approx C_D(\alpha)$, $C_L(\alpha, \beta) \approx C_L(\alpha)$.

Now, we characterize these two coefficients on an interval of 180° (easily extendable to 360°) for the angle of attack, in order to consider both the hovering and the plane flight configurations in one aerodynamic forces model. To do so, we consider the results exposed in [6], that are valid for conventional wind turbines and can be adapted to our scope: by fitting the Makani data on the curves proposed in [6] via optimization techniques like LASSO and Least Squares, we obtained the coefficients curves shown in Figures 4 and 5.

3. Makani Database

The flights data contained in the released database are divided in four testing programs: the Remote Perch Crosswind (RPX), the High Hovering (HH), the All-Modes Crosswind (CW) and the Floating Crosswind (FC). 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 first All-Modes Crosswind flight).

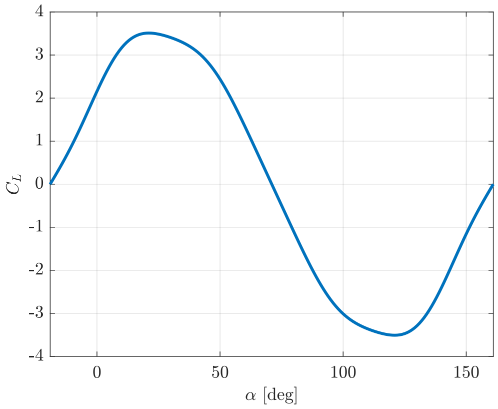

Figure 4: Lift curve obtained from the Gaussian fit of Makani data

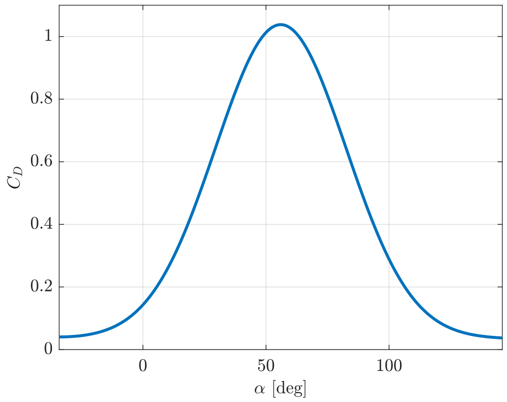

Figure 5: Drag curve obtained from the Gaussian fit of Makani data

From this huge database, we extract the signals regarding the hovering phase, since it is the flight stage we are interested into for the optimal tuning (described in the next Section). In other words, we realize smaller hovering datasets that contain the wind vector and the wing position, plus the position command vectors and the thrust computed by the actual Makani controller during the real tests, opportunely converted to the reference frames employed in Section 2.

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 Tuning

The optimal tuning of the drone control system is performed by solving a non linear optimization problem, which aims to find the control parameters that make the proposed system most similar to the behaviour of the M600 drone (described by the data).

4.1. Non Linear Program Formulation

The general formulation adopted optimization problem is the following:

$$\min_{\omega_c \in \mathbb{R}^7} \quad f(\omega_c) \quad \text{s. t.} \tag{10}$$

$$C\,\omega_c \geq d \tag{11}$$

where $\omega_c \in \mathbb{R}^7$ is the optimization variables vector, $f(\omega_c)$ is the cost function and (11) represent the linear constraints applied to the problem. The cost function $f(\omega_c)$ is expressed as

$$\begin{aligned} f(\omega_c) = \sum_{t=0}^{T} \quad & w_x^2\,(x_G^{sim}(t,\omega_c) - \tilde{x}_G(t))^2 + \\ & + w_y^2\,(y_G^{sim}(t,\omega_c) - \tilde{y}_G(t))^2 + \\ & + w_z^2\,(z_G^{sim}(t,\omega_c) - \tilde{z}_G(t))^2 + \\ & + w_u^2\,(U_1^{sim}(t,\omega_c) - \tilde{U}_1(t))^2 \end{aligned} \tag{12}$$

where x_G, y_G and z_G are the global coordinates of the wing and the superscript *sim* and the $\sim$ symbol indicate if the source of the signal is, respectively, the simulation or the database. In other words, the cost function takes as input the hovering data of the M600 and the signals generated by the simulation of the previously described model, and computes the squared errors between the M600 position and the simulated position, plus a term that quantifies the difference in the commanded thrust. In this way we measure the distance between the simulator output and the data at each time instant $t \in [0, T]$, where T is the duration of the considered flight. 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.

The constraints in Expression (11) 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 outermost loop (position controller): in this way, we can neglect the dynamics of the innermost loop when tuning the outer one. Furthermore, with inequality (11) we set the upper and lower bounds for the optimization variables.

4.2. Solver

The optimization problem (10) can be classified as constrained Non-Linear Program (NLP), since the cost function (12) is non-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_G^{sim}, y_G^{sim}, z_G^{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, implementing the Gauss-Newton Hessian approximation and the Forward Finite Differences method for the derivatives computation.

5. Results and Validation

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.

Figure 6 shows the reference tracking performance of the system on the 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

$$RMSE_x(\omega_c) = \sqrt{\frac{1}{N}\sum_{i=0}^{N}(\tilde{x}_G - x_G^{sim}(\omega_c))^2} \tag{13}$$

$$RMSE_y(\omega_c) = \sqrt{\frac{1}{N}\sum_{i=0}^{N}(\tilde{y}_G - y_G^{sim}(\omega_c))^2} \tag{14}$$

$$RMSE_z(\omega_c) = \sqrt{\frac{1}{N}\sum_{i=0}^{N}(\tilde{z}_G - z_G^{sim}(\omega_c))^2} \tag{15}$$

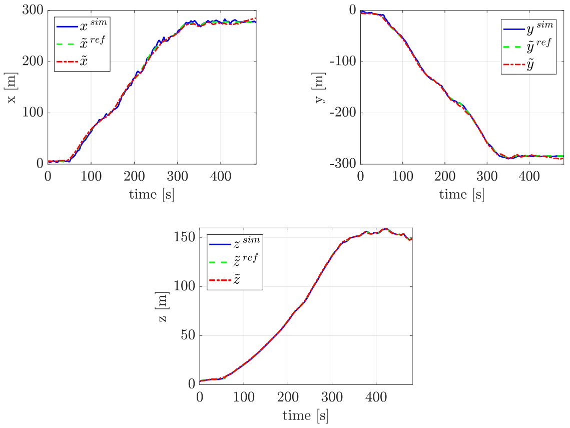

Figure 7: Reference tracking with ω_c^{opt} on the dataset *CW01*

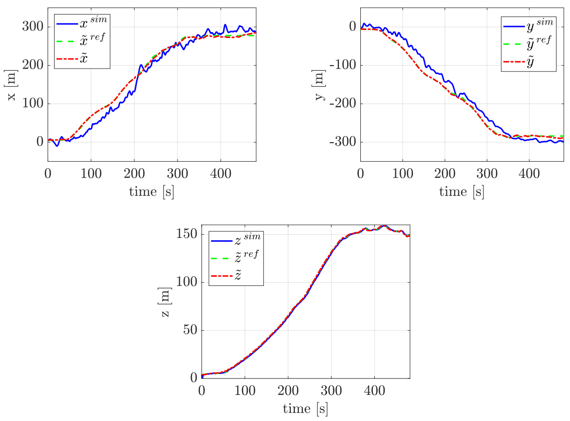

Figure 6: Reference Tracking with ω_c^0 on the dataset *CW01*

Now, by running the solver, we find a local minimizer

$$\omega_c^{opt} = [3.58\;3.58\;3.63\;3.22\;3.22\;0.64\;0.13]^T \quad (16)$$

that corresponds to the performance depicted in Figure 7, which 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 8). These results are then validated on the datasets from *CW05* to *CW10*, confirming the quality of the solution: as we can see from Figure 9, the RMSE values obtained with the validation datasets are in the same order of magnitude of the RMSE indexes produced by the optimization datasets.

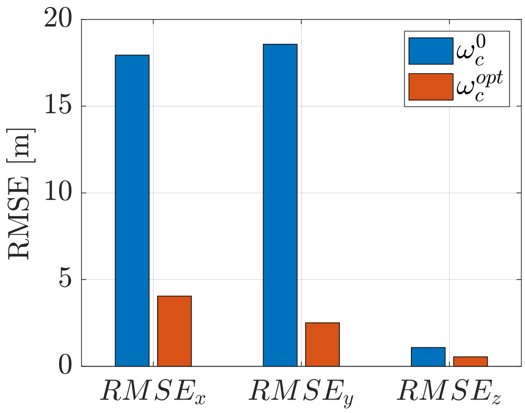

Figure 8: RMSE comparison between starting and optimal control parameters

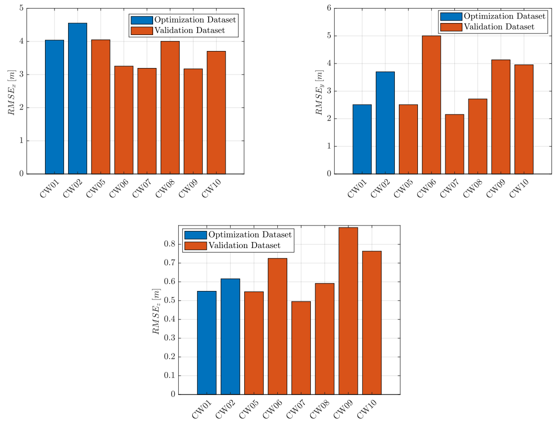

Figure 9: RMSE values for the x, y and z coordinates on the validation datasets

6. Conclusions and Future Developments

The goal of this work 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 nonlinear 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.

References

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

[2] 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.

[3] 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.

[4] 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.

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

[6] 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.