POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Adaptive control and mission planner design for UAV operations with battery management

Laurea Magistrale in Space Engineering - Ingegneria Spaziale

Author: Alessandro Boldrini

Advisor: Prof. Davide Invernizzi

Academic year: 2022-2023

1. Introduction

In recent years, there has been a growing interest in Unmanned Aerial Vehicles (UAVs), commonly known as drones. These aircraft operate without a pilot on board and can be either remotely controlled or autonomously guided by sophisticated flight control systems. UAVs come in various types and sizes, serving diverse mission purposes. Quadrotor UAVs, in particular, are gaining popularity due to their compact size and versatility in handling a range of scenarios, including search and rescue missions, exploration of hazardous indoor environments, surveillance operations, and delivery. Given the limited flight-time capabilities of battery-powered UAVs, the drone must be able to make instantaneous decisions regarding the continuation of the assigned mission or returning to the base for recharging, with the goal of subsequently resuming the mission from the point where it was interrupted. A possible solution explored in this thesis is the use of adaptive control techniques, to ensure a high level of performance despite the progressive discharge of the battery and to also provide a real-time estimate of a parameter indicative of the battery state for use in decision-making algorithms, since the estimate of the battery state of charge available in standard UAV autopilots is not always reliable [2]. Therefore, the goal of the work is to develop a mission management system based on an hybrid automaton that solves the persistent trajectory tracking problem, taking into consideration battery discharge and the possibility of recharging at dedicated pads.

2. Problem Formulation

For the formulation of the control problem, a mathematical model capturing both the UAV flight dynamics and the battery dynamics must be developed.
Starting from the drone model, one can define two different frames, the inertial Cartesian frame $\mathcal{F}_I := (O_I, \{\mathbf{x}_I, \mathbf{y}_I, \mathbf{z}_I\})$ and the body-fixed Cartesian one attached to the center of mass of the quadrotor $\mathcal{F}_B := (O_B, \{\mathbf{x}_B, \mathbf{y}_B, \mathbf{z}_B\})$, where with O_k we refer to the origin of the two frames, and $\mathbf{x}_k, \mathbf{y}_k, \mathbf{z}_k$ are unit vectors that form a basis of an orthogonal frame. The position vector $\mathbf{p} \in \mathbb{R}^3$ from O_I to O_B and the rotation matrix $\mathbf{R} := [\mathbf{b_1}, \mathbf{b_2}, \mathbf{b_3}] \in SO(3)$ describing the attitude of $\mathcal{F}_B$ with respect to $\mathcal{F}_I$ can be defined in order to fully describe the configuration of the quadrotor through the definition of the tuple $(\mathbf{p}, \mathbf{R}) \in \mathbb{R}^3 \times SO(3)$. is described by the

following set of nonlinear differential equations:

$$\begin{cases} \dot{\mathbf{R}} = \mathbf{R}\mathbf{S}(\boldsymbol{\omega}_c) \\ \dot{\mathbf{p}} = \mathbf{v} \\ m\dot{\mathbf{v}} = \lambda\, T_c\, \mathbf{R}\, \mathbf{e}_3 - mg\, \mathbf{e}_3 + \mathbf{f}_e \end{cases}. \quad (1)$$

$\mathbf{S}(\boldsymbol{\omega}_B)$ is called the Skew operator, $m \in \mathbb{R}_{>0}$ represents the UAV mass, $T_c > 0 \in \mathbb{R}^3$ represents the overall thrust applied by the propellers, $\mathbf{f}_e \in \mathbb{R}^3$ the external disturbances, g the gravitational acceleration, $\mathbf{e}_3 = [0, 0, 1]^{\mathrm{T}}$, $\lambda \in \mathbb{R}_{>0}$ is the control effectiveness, which gradually decreases during nominal operating conditions due to battery discharge. Moreover, as shown by S. Meraglia et al. in [7], considering $\boldsymbol{\omega}_B \in \mathbb{R}^3$, which represents the angular velocity describing the rotation of $\mathcal{F}_B$ with respect to $\mathcal{F}_I$, as the control input $\boldsymbol{\omega}_c$, the drone's orientation dynamics can be decoupled from position control. Therefore, attitude dynamics can controlled by using classic control designs such as PID loops, that have been demonstrated to be effective also in highly dynamic motions.

Regarding the battery dynamics, the most common solution in the literature is to create an equivalent circuit that can simulate the voltage and current behaviour over time of a real battery. For this thesis, we decided to use the *Rint* model formalized by D. Tang in [9]. This model, explicitly designed for drones, also incorporates a power consumption model derived from helicopter aerodynamic theory, allowing for better simulation of different discharges depending on the drone's flight stages.

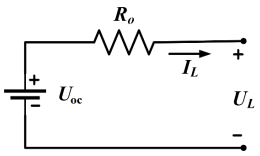

Figure 1: *Rint* model equivalent circuit

By defining the State of Energy (SOE) as the normalized remaining available energy of lithium-ion batteries, as follows:

$$SOE(t) = SOE(t_0) - \frac{\int_{t_0}^{t} P_{tot}(\tau)\, d(\tau)}{E_{crit}}, \quad (2)$$

where, E_{crit} represents the maximum available energy of the battery and P_{tot} the overall power consumption at the selected time instant, which is composed of the power delivered to the actuator P together with the dissipated one, the state-space model of the battery can be expressed as:

$$\frac{\mathrm{d}SOE}{\mathrm{d}t} = (P(t) + I_L(t)^2 R_0(t))\, E_{crit}^{-1}, \quad (3)$$

$$\frac{\mathrm{d}R_0(t)}{\mathrm{d}t} = w(t). \quad (4)$$

It can be noticed that $R_0(t)$, which represents the internal resistance, in reality depends itself on the SOE, but due to some external factor, such as the temperature, the coupling between the resistance and the SOE has been shown that it is not as straightforward. So the dynamic of the internal resistance is simulated by an increasing Gaussian noise process $w(t)$ at each time instant. The other variables characterizing the equivalent circuit can be obtained by the following set of equations:

$$U_L(t) = U_{oc}(t) - I_L(t)\, R_0(t),$$
$$U_{oc}(t) = v_L + \lambda_b\, e^{-\gamma\, SOE(t)},$$
$$I_L(t) = \frac{U_{oc}(t) - \sqrt{U_{oc}(t)^2 - 4\, R_0(t)\, P(t)}}{2\, R_0(t)},$$

where U_L and I_L are the delivered voltage and current, U_{oc} represents Open Circuit Voltage, the parameters v_L, λ_b and γ are experimental parameters and P is derived from the power consumption model reported in [9].

For the full dynamical model, the power consumed by the drone depends on the drone's speed $\mathbf{v}$ and altitude z, which will affect the battery discharge. The other coupling term is represented by the term λ. Therefore, it will be necessary to understand how the battery state influences the value of this parameter. In literature, several authors provide empirical proofs that within a specific range where the propellers are considered to work in a regime condition, the thrust is directly proportional to the voltage/current.

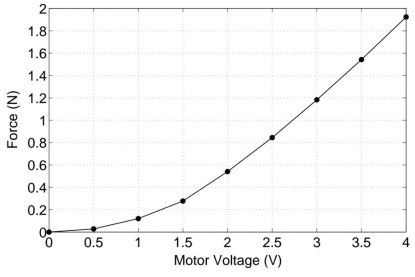

Figure 2: Relation between Voltage and Thrust

Therefore, by computing the ratio between the delivered voltage $U_L(t)$ at each instant with the voltage that would have been generated with a fully charged battery U_L^{full}, and by scaling this ratio by a factor K_U to match the Thrust-Voltage curve slope, uncertainty of the thrust value can be expressed as follows:

$$\lambda = 1 - K_U \frac{U_L^{full}}{U_L}. \tag{5}$$

Alternatively, Podhradský et al. in [8] found an empirical correlation between the State of Charge (SOC) and the delivered thrust which can be observed in the figure below:

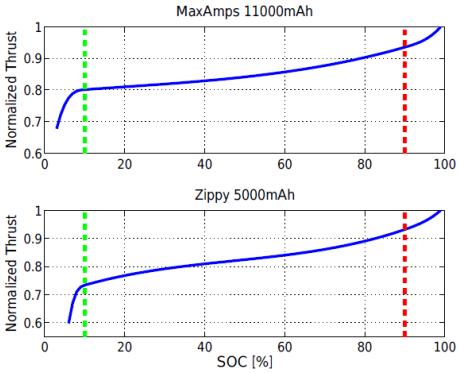

Figure 3: Relation between SOC and Thrust (from [8])

The SOC, which is another parameter exploited in the literature to characterize the battery state, is defined as the level of charge of an electric battery relative to its capacity, mathematically expressed as:

$$SOC(t) = \frac{Q_c - \int_{t_0}^{t} I_L(\tau)\, d(\tau)}{Q_c}, \tag{6}$$

where, Q_c is the current capacity of the battery. The model shown in Fig. 3 was obtained through a least-squares spline approximation of the measured thrust loss, placing two nodes at 10% and 90%, respectively, to separate the almost linear central piece from the highly nonlinear portions at the edges. So, approximating the value of SOC to that of SOE, knowing that from the literature the maximum difference is estimated to be no more than 2/3% ([4]), by inserting the SOE level into this function, it will be possible to obtain the value of λ directly. Both these models have been considered when testing the proposed adaptive controller, which is described in the next section.

3. Adaptive control design

In a typical autopilot controller, the effect of battery discharge is usually compensated for by the integral terms of the control action. The goal of our controller, however, will not only be to compensate for this loss but also to estimate it, allowing us to use this information in the mission management context. Considering that the attitude controller proposed by S. Meraglia et al. in [7] is operating properly, the attitude of the drone $\mathbf{R}$ can be considered approximately equal to the desired attitude $\mathbf{R}_d$ and we can focus on the position dynamics alone. Furthermore, to focus more closely on the impact caused by battery discharge, the external forces, included in the vector $\mathbf{f}_e$ and the uncertainties of the plant have been neglected. Considering $\lambda = 1 - \frac{\Delta T_c}{T_c}$, the thrust can be broken down into its nominal value T_c and its uncertain variation ΔT_c as follows:

$$m\dot{\mathbf{v}} = -mg\mathbf{e}_3 + (T_c - \Delta T_c)\,\mathbf{R}_d\mathbf{e}_3. \tag{7}$$

Then, setting $T_c = |\mathbf{f}_d|$ and $\mathbf{R}_d\mathbf{e}_3 = \frac{\mathbf{f}_d}{|\mathbf{f}_d|}$, and selecting:

$$\mathbf{f}_d = m\,(g\mathbf{e}_3 + \dot{\mathbf{v}}_d) + \mathbf{u},$$

the linear velocity dynamics reads:

$$\dot{\mathbf{v}} = \dot{\mathbf{v}}_d + \frac{1}{m}\,(\mathbf{u} - \Delta T_c \mathbf{R}_d\mathbf{e}_3)\,. \tag{8}$$

It can be noticed that the controller is composed by a feedforward term, $m\,(g\mathbf{e}_3 + \dot{\mathbf{v}}_d)$, which compensate for the constant gravity force and ensure tracking by providing the desired acceleration, and a feedback term $\mathbf{u}$ to be designed. Therefore, the position dynamics can be expressed in state space form as:

$$\dot{\mathbf{x}}_p = \mathbf{A}_p\mathbf{x}_p + \mathbf{B}_p\,(\mathbf{u} + \Theta\mathbf{\Phi}) + \mathbf{B}_{pr}\mathbf{r}, \tag{9}$$

where $\mathbf{\Phi} := \mathbf{R}_d\mathbf{e}_3 \in \mathbb{R}^3$ is the regressor, $\Theta := \Delta T_c \in \mathbb{R}$ is the uncertain parameter and the other components are defined as follows:

$$\mathbf{x}_p = \left[\mathbf{p}^{\mathrm{T}}, \mathbf{v}^{\mathrm{T}}\right], \quad \mathbf{r} = \left[\mathbf{p}_d^{\mathrm{T}}, \mathbf{v}_d^{\mathrm{T}}, \dot{\mathbf{v}}_d^{\mathrm{T}}\right]^{\mathrm{T}},$$

$$\mathbf{A}_p = \begin{bmatrix} \mathbf{0}_3 & \mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix}, \quad \mathbf{B}_p = \frac{1}{m}\begin{bmatrix} \mathbf{0}_3 \\ \mathbf{I}_3 \end{bmatrix},$$

$$\mathbf{B}_{pr} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix},$$

where $\mathbf{I}_3, \mathbf{0}_3 \in \mathbb{R}^{3\times3}$ represent respectively the identity and the zero matrix of dimension three. Moreover, $\mathbf{u}$, can be split by considering it as the sum of the baseline control input designed in absence of uncertainty, denoted as $\mathbf{u}^{bl}$, and an adaptive component, denoted as $\mathbf{u}^{ad}$, which instead will compensate for the uncertainty:

$$\mathbf{u} = \mathbf{u}^{bl} + \mathbf{u}^{ad}. \tag{10}$$

3.1. Baseline Controller

As mentioned earlier, since the adaptive controller will replace the integral contribution, the baseline will be chosen as a simple PD controller:

$$\mathbf{u}^{bl} = -\mathbf{K}_p\mathbf{e}_p - \mathbf{K}_v\mathbf{e}_v, \quad \text{where} \quad \begin{matrix}\mathbf{e}_p := \mathbf{p} - \mathbf{p}_d \\ \mathbf{e}_v := \mathbf{v} - \mathbf{v}_d\end{matrix},$$

where $\mathbf{K}_p, \mathbf{K}_v \in \mathbb{R}^{3\times3}$ contain the gains for the position and velocity error. Therefore, the closed loop position dynamics for the uncertainty-free system can be rewritten as:

$$\dot{\mathbf{x}}_\mathbf{p} = \mathbf{A}_{ref}^{PD}\mathbf{x}_p + \mathbf{B}_{ref}^{PD}\mathbf{r}, \tag{11}$$

where

$$\mathbf{A}_{ref}^{PD} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{I}_3 \\ -\frac{\mathbf{K}_p}{m} & -\frac{\mathbf{K}_v}{m} \end{bmatrix},$$

$$\mathbf{B}_{ref}^{PD} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ -\frac{\mathbf{K}_p}{m} & -\frac{\mathbf{K}_v}{m} & \mathbf{I}_3 \end{bmatrix}.$$

The PD controller thus defined guarantees global exponential tracking of the reference trajectory when relating to the uncertainty-free system.

3.2. Adaptive Controller

In order to compensate for the uncertainty represented by $\Theta\mathbf{\Phi}$, one must remember that Θ is assumed to be constant and unknown. Nevertheless, this value can be estimated by using an adaptive control law. The starting point is the definition of the parameter estimation error:

$$\Delta\Theta = \hat{\Theta}(t) - \Theta = \Delta\hat{T}_c(t) - \Delta T_c.$$

Then, the adaptive control input is chosen as:

$$\mathbf{u}^{ad} = -\hat{\Theta}(t)\mathbf{\Phi}. \tag{12}$$

Three different techniques, belonging to the family of Model Reference Adaptive Control (MRAC), which differ precisely in the model chosen as a reference, have been implemented. By defining the tracking error as:

$$\mathbf{e} = \mathbf{x}_p - \mathbf{x}_{ref}, \tag{13}$$

the reference model for the classic MRAC is chosen as:

$$\dot{\mathbf{x}}_{ref} = \mathbf{A}_{ref}\mathbf{x}_{ref} + \mathbf{B}_{ref}\mathbf{r}, \quad \text{where} \quad \begin{matrix}\mathbf{A}_{ref} := \mathbf{A}_p \\ \mathbf{B}_{ref} := \mathbf{B}_{pr}\end{matrix}.$$

In order to derive adaptive laws, a Lyapunov design approach is employed, choosing as candidate a quadratic positive definite function of the tracking and of the parameter estimation errors:

$$V(\mathbf{e}, \Delta\Theta) = \mathbf{e}^{\mathrm{T}}\,\mathbf{P}\,\mathbf{e} + \frac{\Delta\Theta^2}{2\gamma_\Theta}, \tag{14}$$

where γ_Θ is a scalar tunable parameter of the adaptive law and $\mathbf{P}$ satisfies the Lyapunov equation reported hereafter:

$$\mathbf{P}\,\mathbf{A}_{ref} + \mathbf{A}_{ref}^{\mathrm{T}}\,\mathbf{P} = -\mathbf{Q} \tag{15}$$

$$\text{where} \quad \mathbf{Q} = \mathbf{Q}^{\mathrm{T}} > \mathbf{0}. \tag{16}$$

As common in adaptive control design, to ensure convergence on the reference mismatch error $\mathbf{e}$, the adaptive law is derived by imposing a negative semi-definite condition on the Lie derivative of the Lyapunov function.
As regard the Closed-loop Model Reference Adaptive Control (CMRAC) [6], the reference model is obtained by adding an output injection term as follows:

$$\dot{\mathbf{x}}_{ref} = \mathbf{A}_{ref}\mathbf{x}_{ref} + \mathbf{B}_{ref}\mathbf{r} + \mathbf{L}\mathbf{e}, \tag{17}$$

where $\mathbf{L} \in \mathbb{R}^{6\times6}$ is a gain matrix. For what concerns Predictor Model Reference Adaptive Control (PMRAC) [5], it is obtained by including a state predictor $\hat{\mathbf{x}}$ in the adaptive law in order to predict the system state derivative. The PMRAC design improves transient performance over the standard MRAC and modifies the baseline controller input only when the plant uncertainties are not compensated for. The so-called predictor dynamics can be defined as follows:

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}_{pr}\,(\hat{\mathbf{x}} - \mathbf{x}_p) + \mathbf{A}_{ref}\mathbf{x}_p + \mathbf{B}_{ref}\mathbf{r}. \tag{18}$$

Repeating the same steps used for the MRAC, one can obtain the adaptive laws for the other two control techniques.

4. Simulations and experimental tests

To validate the effectiveness of the presented controllers, preliminary tests were conducted by considering the plant's dynamics presented in Section 3.

Figure 4: ANT-X drone

Then, the implemented solutions were transferred and tested in the UAV simulator of the ANT-X drone developed by the ASCL, which is based on the identified models of the drone's angular velocity dynamics acquired by using the PBSID blackbox identification method, which provides a higher level of reliability for evaluating the effectiveness of the proposed control techniques. The PMRAC controller provided the best performance and thus, it has been selected to be implemented and tested on the drone.
To validate the proposed discharge models and to enable a direct comparison between the simulation and the experimental data, an endurance test was performed, keeping the drone in hovering mode for as long as possible. The graph below illustrates the evolution of the estimated parameter $\hat{\Theta}$ over time during this test:

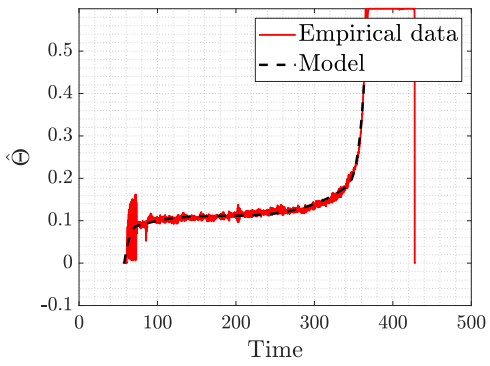

Figure 5: $\hat{\Theta}$ function with respect to time

Similarly to what was done by Podhradský et al. in [8], experimental data has been filtered and divided into 3 different curves, separating the nearly linear region in the center from the other two highly non-linear regions at the edges, thus obtaining a model of thrust loss as a function of flight time. So, through this model, it will be possible to understand the remaining flight time starting from the estimate of the thrust lost, in the absence of external disturbances.

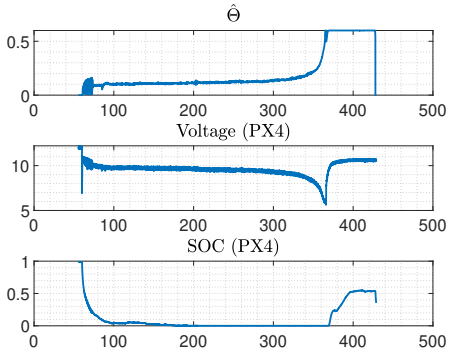

Figure 6: $\hat{\Theta}$ VS delivered voltage VS SOC

Moreover, during this test other output coming from the drone autopilot (PX4) were extracted and compared the estimated parameter of lost thrust. Notably, the estimation of *SOC* turned out to be entirely inaccurate, especially when compared with the estimation of lost thrust, as can be seen in Fig. 6. Then, simulation results exploiting the obtained empirical model have been compared with the results of experimental tests concerning a sequence of steps.

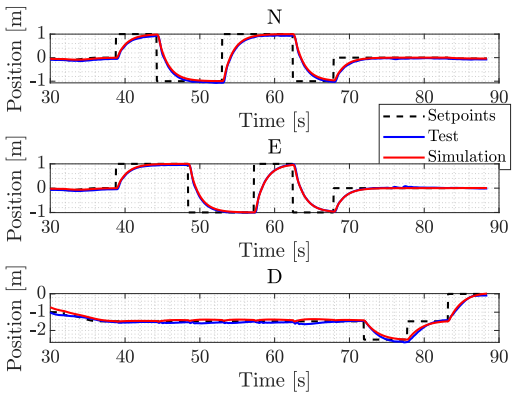

Figure 7: Response comparison to sequence of steps

Lastly, the relationship between voltage and

thrust loss confirms the proportionality between these two quantities.

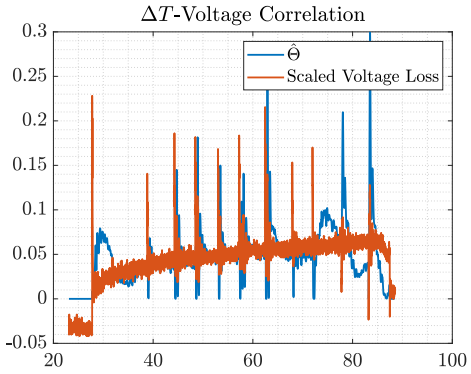

Figure 8: $\hat{\Theta}$ and voltage lost correlation

5. Mission Management

As mentioned earlier, the mission is divided into various modes, each of which has a defined domain based on both the value of the parameter estimated by the adaptive controller and its position, and the resulting error with respect to the reference trajectory. The modes are denoted by the letter $q \in [-1, 0, 1, 2, 3, 4]$, and the overall scheme is represented in the following scheme:

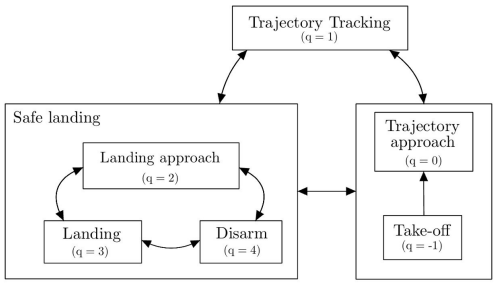

Figure 9: Scheme of the overall switch logic

Where the macro mode for safe landing has been adapted and refined to better align with the purpose of this thesis, drawing inspiration from the work done by Gozzini et al. [3].
The continuity of the mission, even after making a charging stop from where the mission was interrupted, is endured by applying the same technique that has been adopted by Andreetto et al. in [1], in which a generalized time is defined that only elapses when the drone is in the trajectory tracking mode. In mathematical terms, it can be represented as follows:

$$\dot{t}_{gen} = s, \quad \text{where} \quad \begin{cases} s = 1, & \text{if} \quad q = 1 \\ s = 0, & \text{else} \end{cases}.$$

As regards the reference trajectory, it is assumed that control laws for T_c exist that enable the asymptotic tracking of any bounded velocity trajectory, denoted as $\mathbf{v}_d(t) \in \mathbb{R}^3$. This assumption allows us to formulate the tracking problem independently of the specific UAV actuation mechanism, relying on the kinematic model:

$$\dot{\mathbf{p}}_c = \mathbf{v}_c, \tag{19}$$

where $\mathbf{v}_c$ is a virtual input representing the drone's velocity in the inertial frame, utilized for control design purposes. Under this assumption, we can use this strategy to track any sufficiently smooth trajectory, denoted as $t \to \mathbf{p}_d(t) \in \mathbb{R}^3$. Defining the tracking error as:

$$\boldsymbol{\varepsilon}(t) \coloneqq \mathbf{p}(t) - \mathbf{p}_d(t_{gen}), \tag{20}$$

the corresponding dynamics of the error can be rewritten as

$$\dot{\boldsymbol{\varepsilon}}(t) = \mathbf{v}(t) - \mathbf{v}_d(t_{gen}), \tag{21}$$

where considering the measurements of both the position $\mathbf{p}(t)$ and the velocity $\mathbf{v}(t)$ for the trajectory generation, closed-loop guidance is achieved. The proposed control law will then be expressed by

$$\mathbf{v}_c(t) = -v_M \tanh\left(\frac{\mathbf{K}_\varepsilon \boldsymbol{\varepsilon}(t)}{v_M}\right) + \mathbf{v}_d(t_{gen}),$$

where $v_M \in \mathbb{R}_{>0}$ represents the saturation level for the velocity, while $\mathbf{K}_\varepsilon \in \mathbb{R}^3$ is a diagonal matrix containing positive gains. Moreover, the hyperbolic tangent function has been preferred over a simple saturation function in order to obtain a smoother trajectory to follow. Therefore, the acceleration setpoints $\mathbf{a}_c(t)$, can be derived by deriving the previous formula:

$$\mathbf{a}_c(t) = -\mathbf{K}_\varepsilon \operatorname{sech}\left(\frac{\mathbf{K}_p \boldsymbol{\varepsilon}(t)}{v_M}\right) \dot{\boldsymbol{\varepsilon}}(t) + \mathbf{a}_d(t_{gen}),$$

while, for the position setpoints $\mathbf{p}_c(t)$, numerical integration will be used. It can be observed that through this choice, each mode can have different values in the $\mathbf{K}_\varepsilon$ matrix, saturation velocity v_M to better adapt to the required performance in the each mode. Additionally, excluding the main mode where tracking a trajectory is required, in the other modes, $\mathbf{v}_d(t_{gen})$ and $\mathbf{a}_d(t_{gen})$ will be zero, and the hyperbolic tangent will

only create a smooth trajectory leading to the required point $\mathbf{p}_d(t_{gen})$. So, the overall dynamics each mode and the corresponding outputs $\mathbf{y}$ can be represented as:

$$\begin{cases} \dot{\mathbf{p}}_c = \mathbf{v}_c \\ \dot{t}_{gen} = s \\ \mathbf{y} = (\mathbf{p}_c(t), \mathbf{v}_c(t), \mathbf{a}_c(t)) \end{cases} . \quad (22)$$

In order to return safely to the charging base, ΔT_M represents the threshold value of the estimated parameter at which the drone must interrupt the mission , while ΔT_{crit} represents a critical value that will initiate the landing procedure at the current location of the drone, given the impossibility of a safe landing on the pad. Letting ΔT_M to be a function of the distance from the charging base, the drone can increase the duration of the mission. In Section 4 a model of thrust loss dynamics as a function of time was derived from an endurance test. Therefore, knowing the battery duration in hovering, it will be possible to choose the right point to end the mission based on the remaining time until battery discharge. In order to consider different flight stages from hovering, a safety factor have to be used to scale this threshold in order to increase safety, at the expense of longer endurance. The selected formula will be:

$$\Delta T_M = \min(\Delta T_M{}^{Max}, \Delta T_{fun}(K(T_{batt} - T_{land} - |\mathbf{\Psi}_\perp| / v_M))).$$

In the above formula ΔT_{fun} represents the empirical model, K is the safety scale factor, T_{batt} is the time when thrust loss reaches 30% of the maximum thrust in hovering conditions, T_{land} represents the time taken to descend from the cruising altitude to the ground and the last term consider the time that the drone implies to reach the recharge station from the point where the mission is interrupted (T_{dist}). A graphical representation of the function is shown:

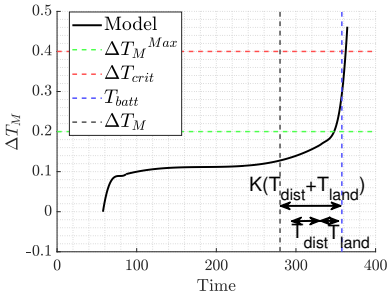

Figure 10: Visual representation of ΔT_M

Safety in the final stages of the mission can be further increased by imposing a maximum loss equal to $\Delta T_M{}^{Max}$ for the activation of the landing mode.

Finally, to avoid the phenomenon of chattering, caused by possible fluctuations in the estimated parameter, it was decided to follow what was done in [3], using hysteric switch conditions between the modes.

Different trajectories have been simulated to prove the effectiveness of the mission manager independently of the trajectory: a circular trajectory simulating loitering around a specific point, and a more complex trajectory that simulates the surveillance of a rectangular area using a pattern of line segments. The results obtained for the simulation of the pattern trajectory are shown hereafter:

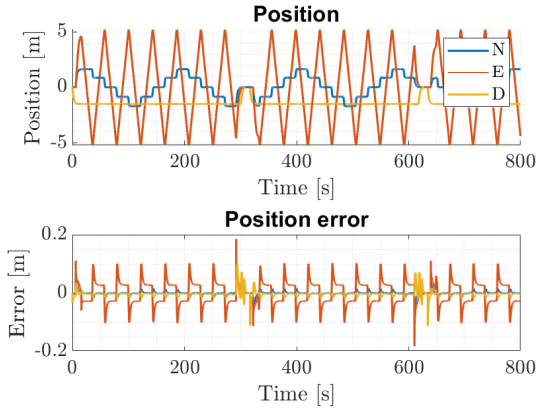

Figure 11: Pattern trajectory tracking errors

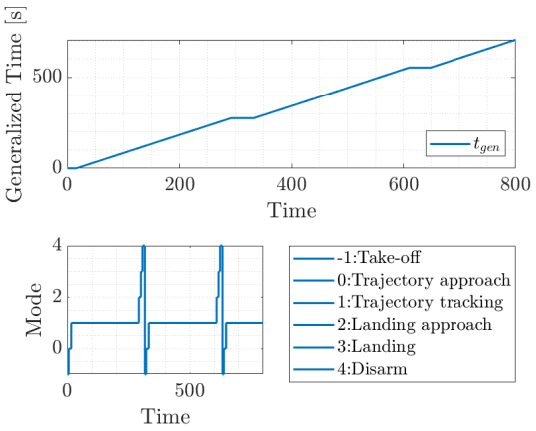

Figure 12: Generalized time and active mode

After demonstrating the correct functioning of the logic in the case of nominal discharge, we decided to test progressively faster discharges,

noting that the drone can complete the mission by landing at the predetermined point even with discharges up to about 70% faster than expected. For higher values, the drone will enter the emergency landing mode, simply attempting to land in its current location.

6. Conclusions

One of the objective of this thesis was to formulate a control law capable of compensating for the performance degradation due to battery discharge in multirotor UAVs while solving trajectory tracking tasks. The adaptive control techniques used have demonstrated the ability to extend the performance obtained under nominal conditions, even considering the effects generated by the battery discharge. Furthermore, the utility of this control technique goes beyond achieving this result, with an online estimation of thrust loss. This parameter was subsequently successfully used in decision-making within an algorithm that allowed the drone to complete a mission autonomously. This estimated parameter becomes even more important, especially when compared to what can be obtained from *SOC* or *SOE* estimators already implemented in the autopilots, as PX4, in which reliable estimates from simple models is not as straightforward. The other objective of the thesis was to define a mission management system. The choice to divide the mission into various modes with different domain of application and objectives proved to be an excellent decision. This is also thanks to the decision to introduce hysteresis in the transition functions between the various modes, avoiding the phenomenon of chattering and the degradation of performance that follows. The robustness derived from the choice of the algorithm for generating the threshold value for mission termination should also be emphasized, as it ensured the completion of the mission even in the presence of considerably anomalous conditions, with discharge rates much faster than expected.

References

[1] Marco Andreetto, Daniele Fontanelli, and Luca Zaccarian. Quasi time-optimal hybrid trajectory tracking of an n-dimensional saturated double integrator. In *2016 IEEE Conference on Control Applications (CCA)*, pages 550–555, 2016.

[2] Dronecode. Px4 battery and power module setup, 2023.

[3] Giovanni Gozzini, Davide Invernizzi, Simone Panza, Mattia Giurato, and Marco Lovera. Air-to-air automatic landing of unmanned aerial vehicles: A quasi time-optimal hybrid strategy. *IEEE Control Systems Letters*, 4(3):692–697, July 2020.

[4] Ryan Hickey and Thomas M. Jahns. Direct comparison of state-of-charge and state-of-energy metrics for li-ion battery energy storage. In *2019 IEEE Energy Conversion Congress and Exposition (ECCE)*, pages 2466–2470, 2019.

[5] Eugene Lavretsky, Ross Gadient, and Irene M Gregory. Predictor-based model reference adaptive control. *Journal of guidance, control, and dynamics*, 33(4):1195–1201, 2010.

[6] Tea-Gyoo Lee and Uk-Youl Huh. An error feedback model based adaptive controller for nonlinear systems. In *ISIE'97 Proceeding of the IEEE International Symposium on Industrial Electronics*, pages 1095–1100. IEEE, 1997.

[7] Salvatore Meraglia, Giovanni Gozzini, Shang Liu, and Davide Invernizzi. Adaptive control for high-performance trajectory tracking in multirotor uavs. In *9th International Conference on Control, Decision and Information Technologies (CoDIT'23)*, 2023.

[8] Michal Podhradskỳ, Jarret Bone, Calvin Coopmans, and Austin Jensen. Battery model-based thrust controller for a small, low cost multirotor unmanned aerial vehicles. In *2013 international conference on unmanned aircraft systems (ICUAS)*, pages 105–113. IEEE, 2013.

[9] DY Tang, MT Gong, JS Yu, and X Li. A power transfer model-based method for lithium-ion battery discharge time prediction of electric rotatory-wing uav. *Microelectronics Reliability*, 114:113832, 2020.