ADAPTIVE CONTROL AND MISSION PLANNER DESIGN FOR UAV OPERATIONS WITH BATTERY MANAGEMENT

Alessandro Boldrini[1], Giovanni Gozzini[1] & Davide Invernizzi[1]

[1]Dipartimento di Scienze e Tecnologie Aerospaziali, Politecnico di Milano, Via La Masa 34, 20156 Milano

Abstract

The primary objective of this paper is to leverage adaptive control techniques to effectively counteract the reduction in thrust caused by the battery discharge in Unmanned Aerial Vehicles (UAVs) enhancing their reliability and autonomy while extending their operational range and mission success rates. In addition to this, the paper aims to devise a mission planner capable of ensuring the safe execution of trajectory tracking tasks, such as those related to monitoring/surveillance missions. One of the main contributions of the paper is the augmentation of classic baseline controllers with adaptive control mechanisms to dynamically adjust and optimize the control inputs of the UAV as the battery discharges. This approach continuously monitors and adapts to the changing thrust capabilities the control inputs, aiming to maintain the UAV's performance within acceptable parameters throughout the mission. Moreover, regarding the autonomous mission planner, the thrust loss estimate, which can be considered as a key parameter of the battery's status, becomes crucial also for the decision-making algorithms that assess when it is prudent to terminate the mission, ensuring the UAV's safe return to the charging station without risking a depleted battery during flight. Therefore, the mission management system is completed by breaking down the overall mission into different modes and defining a hybrid automaton that enables the transition from one mode to another and a control input that allows the UAV to perform the task autonomously, including when it necessitates to return to the charging base. Lastly, this control scheme and the mission management real-world effectiveness is assessed through a series of tests under different simulation conditions.

Keywords: Autonomous UAV, Adaptive Control, Battery Discharge, Hybrid Automaton

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 possibility of continuing 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 presented in this paper is the use of adaptive control techniques to ensure a high level of performance despite the progressive discharge of the battery and to 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.

Therefore, 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, is presented.

Related works

The problem of trajectory tracking and, more broadly, the control of position and attitude in small quadrotor UAVs has garnered significant attention in academic research. Authors in [1, 2] provide a comprehensive overview of UAV control design principles. Given the underactuated characteristics of quadrotor UAVs, position and attitude cannot be independently controlled. Several nonlinear trajectory tracking controllers use separate control loops for attitude and position, as demonstrated for example in [3, 4]. A frequently employed method involves linearizing the system around an equilibrium point and designing a feedback controller as done in [5]. However, this approach suffers significant performance degradation with large errors and lacks robustness against unmodelled dynamics and parametric uncertainties. A possible modification to compensate for these effects, is an adaptive approaches that can be used to estimate parameter uncertainties, taking into account unmodelled dynamics as well [6].

Considering the reliance of small UAVs, particularly quadrotors, on battery power, understanding the coupling between the battery and drone dynamics is crucial. Various experimental power consumption models are available in the literature [7, 8]. Typical small battery-powered UAVs face limited endurance due to weight constraints on the battery pack. Authors in [9] examined parameters influencing the performance of battery-powered UAVs and formulated mathematical expressions to estimate range and endurance under battery discharge conditions. Moreover, a battery model-based thrust controller to compensate for the effects generated by battery dynamics is presented in [10].

To ensure the energy safety of UAV flight missions, consistent estimation of the battery state is necessary. Various parameters and estimation methods have been defined in the literature due to the impossibility of direct measurements. Authors in [11] used the estimation of the State of Charge (SOC) of the battery, [12] formulated an online estimator of the State of Energy (SOE), and [7] used an experimental power consumption model to predict the End of Discharge time, indicating the remaining battery life. Authors in [13] compared SOC and SOE, highlighting the differences in their trends under various conditions and determining which parameter is more indicative of the actual battery state.

When deployed on autonomous missions, drones require intelligent decision-making systems for trajectory planning and tour optimization. Autonomous drones must rapidly compute feasible and energy-efficient paths to avoid collisions. A review of key aspects and current trends in drone path planning research, identifying factors that need to be addressed to develop practical path planners, can be found in [14]. Given the constrained capacity of onboard batteries, it is crucial to optimize mission objectives and incorporate recharging operations during extended flights. In [15] the joint problem of flight mission planning and recharging optimization is formulated as a multi-criteria Asymmetric Traveling Salesman Problem (ATSP), thus, ensuring the drone visits all target sites and refuels as necessary. Various strategies for power supply and energy management for UAVs are described in [16].

2. Problem Formulation

For the formulation of the control problem, a mathematical model capturing both the UAV flight dynamics and the battery dynamics is developed.

2.1 Quadrotor model

Starting from the drone model, two different frames can be defined, the inertial Cartesian frame $\mathscr{F}_I \coloneqq (O_I, \{\mathbf{x}_I, \mathbf{y}_I, \mathbf{z}_I\})$ and the body-fixed Cartesian one attached to the center of mass of the quadrotor $\mathscr{F}_B \coloneqq (O_B, \{\mathbf{x}_B, \mathbf{y}_B, \mathbf{z}_B\})$, where O_k refers 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 $\mathbf{R} \coloneqq [\mathbf{b_1}, \mathbf{b_2}, \mathbf{b_3}] \in \mathrm{SO}(3)$ describing the attitude of $\mathscr{F}_B$ with respect to $\mathscr{F}_I$ can be defined in order to fully describe the configuration of the quadrotor through the definition of the tuple $(\mathbf{p}, \mathbf{R}) \in \mathbb{R}^3 \times \mathrm{SO}(3)$.

As shown in [17], given the full actuation capabilities of the UAV attitude dynamics, classic control designs such as PID loops have been demonstrated to be effective in controlling the attitude dynamics of the UAV also in highly dynamic motions and under the assumption that the angular velocity dynamics is controlled at a sufficiently fast rate [18], the angular velocity $\omega_B \in \mathbb{R}^3$ describing the rotation of $\mathscr{F}_B$ with respect to $\mathscr{F}_I$ can be considered directly as the control input ω_c.

Therefore, the nonlinear dynamical model of the UAV can be described by the following simplified set of nonlinear differential equations:

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

where $\mathbf{p} \in \mathbb{R}^3$ and $\mathbf{v} \in \mathbb{R}^3$ are respectively the position and the velocity of the UAV both expressed in the inertial frame, $m \in \mathbb{R}_{>0}$ represents the UAV mass, $T_c \in \mathbb{R}_{>0}$ is the overall thrust applied by the propellers, $\mathbf{f}_e \in \mathbb{R}^3$ represents the external disturbances, $g = 9.81\,\mathrm{m/s^2}$ the gravitational acceleration, $\mathbf{e}_3 = [0,0,1]^\top$, $\lambda \in \mathbb{R}_{>0}$ is the control effectiveness, which gradually decreases during nominal operating conditions due to battery discharge, and $\mathbf{S}(\cdot)$ is the Skew operator defined as the map $\mathbf{S}(\cdot) : \mathbb{R}^3 \to \mathfrak{so}(3) := \{W \in \mathbb{R}^{3\times 3} : W = -W^\top\}$ that given $x, y \in \mathbb{R}^3$ is such that $S(x)y = x \times y$.

2.2 Battery model

Battery discharge effects play an important role in the behaviour and performance of multirotor UAVs. For this reason, the relation between the parameters of the battery state and the thrust has been deeply investigated in literature. The two most common parameters used to described the battery state in literature are the State of Energy (SOE) and the State of Charge (SOC).
The SOE is the normalized remaining available energy of Lithium-Ion batteries and is defined as follows:

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

where, E_{crit} represents the maximum available energy of the battery and $P_{tot}(t)$ the overall power consumption at the selected time instant, which is composed of the power delivered to the actuator $P(t)$ together with the dissipated one.
The SOC is instead defined as the level of charge of an electric battery relative to its capacity and is mathematically expressed as:

$$SOC(t) := \frac{Q_c - \int_{t_0}^{t} I_L(\tau)\, d\tau}{Q_c}, \tag{3}$$

where, Q_c is the current capacity of the battery and $I_L(t)$ is the instantaneous delivered current.
As shown in literature, the trend of these two parameters during discharge is different, with the SOE lower than the SOC due to the fact that the battery voltage behaviour is considered in SOE computation. Therefore, when choosing a minimum threshold to end a mission, monitoring the SOE instead of the SOC is justified. Nevertheless, the difference between these two parameters is quite small, at most 2/3% ([13]). Having found in the literature an equivalent circuit of the battery that through a power consumption model explains the variation of SOE during different flight stages [7], and a model of thrust loss in relation with the SOC of the battery [10], to leverage both, for the sake of simplicity these two parameters and their evolution over time can be considered equivalent.
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. In this paper the $Rint$ model formalized in [7] and shown in Figure 1 is used. 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.
Thanks to the definition of the SOE, the state-space model of the battery can be expressed as:

$$\frac{\mathrm{d}SOE}{\mathrm{d}t} = \left(P(t) + I_L(t)^2 R_0(t)\right) E_{crit}^{-1}, \tag{4}$$

$$\frac{\mathrm{d}R_0(t)}{\mathrm{d}t} = w(t). \tag{5}$$

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

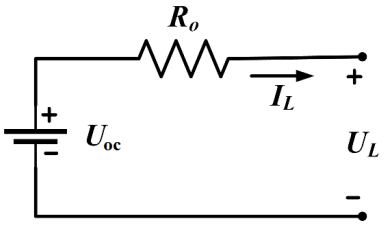


Figure 1 – *Rint* model equivalent circuit.

and the SOE has been shown that it is not as straightforward. For this reason, the dynamics 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:

$$U_L(t) = U_{oc}(t) - I_L(t)\, R_0(t), \tag{6}$$

$$U_{oc}(t) = v_L + \lambda_b\, e^{-\gamma\, SOE(t)}, \tag{7}$$

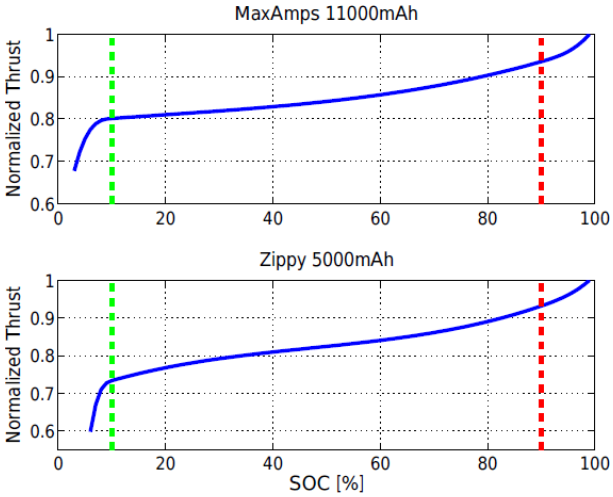
$$I_L(t) = \frac{U_{oc}(t) - \sqrt{U_{oc}(t)^2 - 4\, R_0(t)\, P(t)}}{2\, R_0(t)}, \tag{8}$$

where $U_L(t)$ and $I_L(t)$ are the delivered voltage and current, $U_{oc}(t)$ represents Open Circuit Voltage, v_L, λ_b and γ are parameters that can be retrieved by least square fitting of an experimental dataset and $P(t)$ is derived from the power consumption model reported in [7].

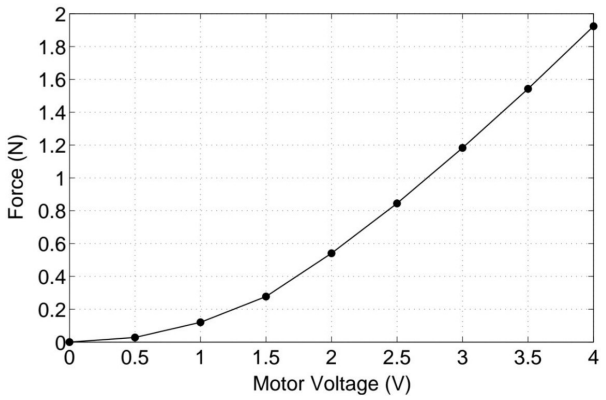
2.3 Quadrotor and battery dynamics coupling

The coupling between the quadrotor dynamics and the battery dynamics is bidirectional, in fact, the power consumption in the battery dynamics model depends on the drone's speed $\mathbf{v}$ and altitude z, while the other coupling term in the drone's dynamics is represented by the parameter λ that depends on the battery discharge. Therefore, it is necessary to understand how the battery state influences the value of the parameter λ.

For what concern the SOC-dependent model, the relation between the SOC and the delivered thrust at fixed throttle has a spline-like behaviour as the one shown in Figure 2a, which is 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. From the model in Figure 2a, by inserting the SOE level obtained from the equivalent circuit into this function, it is possible to obtain the value of the thrust loss λ directly.



(a) Relation between SOC and thrust (from [10]).



(b) Relation between voltage and thrust (from [19]).

Figure 2 – Relation between battery state parameters (SOC and voltage) and thrust.

Alternatively, 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 as shown in Figure 2b. 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 in Figure 2b, uncertainty of the thrust value can be expressed as follows:

$$\lambda := 1 - K_U \frac{U_L^{full}}{U_L}. \tag{9}$$

Both these models have been considered when testing the proposed adaptive controller, which is described in the next section.

3. Controller 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 the presented controller, however, is not only to compensate for this loss but also to estimate it, allowing to use this information in the mission management context. Exploiting the hierarchical control architecture and adopting the same attitude controller and attitude planner proposed in [17], the attitude of the drone $\mathbf{R}$ can be considered approximately equal to the desired attitude $\mathbf{R}_d$ at each time instant because of the fastness attitude dynamics with respect to the position dynamics. Thus, it is possible to focuses on the position dynamics alone. 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:

$$m\dot{\mathbf{v}} = -mg\mathbf{e}_3 + (T_c - \Delta T_c)\mathbf{R}_d\mathbf{e}_3. \tag{10}$$

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

$$\mathbf{f}_d = m(g\mathbf{e}_3 + \dot{\mathbf{v}}_d) + \mathbf{u}, \tag{11}$$

the linear velocity dynamics reads:

$$\dot{\mathbf{v}} = \dot{\mathbf{v}}_d + \frac{1}{m}(\mathbf{u} - \Delta T_c \mathbf{R}_d \mathbf{e}_3). \tag{12}$$

It can be noticed that the desired control force in (11) 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:

$$\dot{\mathbf{x}}_p = \mathbf{A}_p\mathbf{x}_p + \mathbf{B}_p(\mathbf{u} + \Theta\Phi) + \mathbf{B}_{pr}\mathbf{r}, \tag{13}$$

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

$$\mathbf{x}_p := \begin{bmatrix}\mathbf{p}^\top & \mathbf{v}^\top\end{bmatrix}^\top, \quad \mathbf{r} := \begin{bmatrix}\mathbf{p}_d^\top & \mathbf{v}_d^\top & \dot{\mathbf{v}}_d^\top\end{bmatrix}^\top, \tag{14}$$

$$\mathbf{A}_p := \begin{bmatrix}\mathbf{0}_3 & \mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{0}_3\end{bmatrix}, \quad \mathbf{B}_p := \frac{1}{m}\begin{bmatrix}\mathbf{0}_3 \\ \mathbf{I}_3\end{bmatrix}, \quad \mathbf{B}_{pr} := \begin{bmatrix}\mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3\end{bmatrix}, \tag{15}$$

where $\mathbf{I}_3$, $\mathbf{0}_3 \in \mathbb{R}^{3\times 3}$ represent the identity and the zero matrix of dimension three, respectively.
Moreover, the control input

$$\mathbf{u} := \mathbf{u}^{bl} + \mathbf{u}^{ad}, \tag{16}$$

is defined as the sum of the baseline control input designed in absence of uncertainty, denoted as $\mathbf{u}^{bl}$, and of an adaptive component, denoted as $\mathbf{u}^{ad}$, which instead will compensate for the uncertainty.

3.1 Baseline Controller

As mentioned earlier, since the adaptive controller replaces the integral contribution, the baseline is chosen as a simple PD controller:

$$\mathbf{u}^{bl} = -\mathbf{K}_p\mathbf{e}_p - \mathbf{K}_v\mathbf{e}_v, \tag{17}$$

where $\mathbf{e}_p := \mathbf{p} - \mathbf{p}_d$ is the position tracking error, $\mathbf{e}_v := \mathbf{v} - \mathbf{v}_d$ is the velocity tracking error, and the diagonal matrices $\mathbf{K}_p, \mathbf{K}_v \in \mathbb{R}^{3\times3}$ contain the gains for the position and velocity error, respectively. Finally, the closed loop position dynamics for the uncertainty-free system can be rewritten as:

$$\dot{\mathbf{x}}_\mathbf{p} = \mathbf{A}_{ref}^{PD}\mathbf{x}_p + \mathbf{B}_{ref}^{PD}\mathbf{r}, \tag{18}$$

where

$$\mathbf{A}_{ref}^{PD} := \begin{bmatrix} \mathbf{0}_3 & \mathbf{I}_3 \\ -\frac{\mathbf{K}_p}{m} & -\frac{\mathbf{K}_v}{m} \end{bmatrix}, \tag{19}$$

$$\mathbf{B}_{ref}^{PD} := \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ -\frac{\mathbf{K}_p}{m} & -\frac{\mathbf{K}_v}{m} & \mathbf{I}_3 \end{bmatrix}. \tag{20}$$

The PD controller defined in (17) guarantees global exponential tracking of the reference trajectory for the uncertainty-free system.

3.2 Adaptive Controller

In order to compensate for the uncertainty represented by $\Theta\Phi$ in (13), it must be noted that Θ is assumed to be unknown. Nevertheless, this value can be estimated by using an adaptive control law. The starting point is the definition of the parameter estimation error:

$$\Delta\Theta = \hat{\Theta}(t) - \Theta = \Delta\hat{T}_c(t) - \Delta T_c. \tag{21}$$

Then, the adaptive control input is chosen as:

$$\mathbf{u}^{ad} = -\hat{\Theta}(t)\Phi. \tag{22}$$

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: the standard Model Reference Adaptive Control (MRAC), the Closed-loop Model Reference Adaptive Control (CM-RAC) [20] and the Predictor-based Model Reference Adaptive Control (PMRAC) [21].

3.2.1 MRAC

For the MRAC, the reference model is chosen as the closed-loop uncertainty-free system response

$$\dot{\mathbf{x}}_{ref} = \mathbf{A}_{ref}\mathbf{x}_{ref} + \mathbf{B}_{ref}\mathbf{r}, \tag{23}$$

$$\text{where} \quad \begin{matrix} \mathbf{A}_{ref} := \mathbf{A}_{ref}^{PD} \\ \mathbf{B}_{ref} := \mathbf{B}_{ref}^{PD} \end{matrix}. \tag{24}$$

Then, the mismatch error can be defined as

$$\mathbf{e} = \mathbf{x}_p - \mathbf{x}_{ref}. \tag{25}$$

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

$$V(\mathbf{e}, \Delta\Theta) = \mathbf{e}^\top \mathbf{P}\,\mathbf{e} + \frac{\Delta\Theta^2}{2\gamma_\Theta}, \tag{26}$$

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

$$\mathbf{P}\,\mathbf{A}_{ref} + \mathbf{A}_{ref}^\top\,\mathbf{P} = -\mathbf{Q} \tag{27}$$

where $\mathbf{Q} = \mathbf{Q}^\top > \mathbf{0}$. As common in adaptive control design, to ensure convergence of the mismatch error $\mathbf{e}$, the following adaptive law is derived by imposing a negative semi-definite condition on the Lie derivative of the Lyapunov function:

$$\dot{\hat{\Theta}} = \gamma_\Theta\,\Phi\,\mathbf{e}^\top\mathbf{P}\,\mathbf{B}_\mathbf{p}. \tag{28}$$

3.2.2 CMRAC

As regard the Closed-loop Model Reference Adaptive Control (CMRAC) [20], the reference model is obtained from the one of MRAC by adding an output injection term as follows:

$$\dot{\mathbf{x}}_{ref} = \mathbf{A}_{ref}\mathbf{x}_{ref} + \mathbf{B}_{ref}\mathbf{r} + \mathbf{L}\mathbf{e}, \tag{29}$$

where $\mathbf{L} \in \mathbb{R}^{6\times 6}$ is a diagonal matrix of positive gains and the closed-loop uncertainty-free system response in (24) is again used as reference model.
Repeating the same steps used for the MRAC, with the same mismatch error of the (25) but defining $\mathbf{P}$ as the solution of the following Lyapunov equation, in place of the (27),

$$\mathbf{P}\ (\mathbf{A}_{ref} - \mathbf{L}) + (\mathbf{A}_{ref} - \mathbf{L})^{\top}\ \mathbf{P} = -\mathbf{Q} \tag{30}$$

with $\mathbf{Q} = \mathbf{Q}^{\top} > \mathbf{0}$, it is possible to obtain the same adaptive law in (28) for the CMRAC case.

3.2.3 PMRAC

The last technique used for the adaptation of the control input is the Predictor-based Model Reference Adaptive Control (PMRAC) [21]. It is obtained by including a state predictor $\hat{\mathbf{x}}$ in the adaptive law in order to predict the system state derivative.
The so-called predictor dynamics can be defined as follows:

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}_{pr}\,(\hat{\mathbf{x}} - \mathbf{x}_p) + \mathbf{A}_{ref}\mathbf{x}_p + \mathbf{B}_{ref}\mathbf{r}. \tag{31}$$

where $\mathbf{A}_{pr} \in \mathbb{R}^{6\times 6}$ is a diagonal matrix of positive gains and the closed-loop uncertainty-free system response in (24) is again used as reference model. In this case the prediction error is used and defined as follows:

$$\hat{\mathbf{e}} = \hat{\mathbf{x}} - \mathbf{x}_p, \tag{32}$$

A Lyapunov design approach is employed again but, in this case, choosing as candidate a quadratic positive definite function of the prediction error and of the parameter estimation error:

$$V(\hat{\mathbf{e}}, \Delta\Theta) = \hat{\mathbf{e}}^{\top}\,\mathbf{P}_{pr}\,\hat{\mathbf{e}} + \frac{\Delta\Theta^2}{2\gamma_{\Theta}}, \tag{33}$$

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

$$\mathbf{P}_{pr}\,\mathbf{A}_{pr} + \mathbf{A}_{pr}^{\top}\,\mathbf{P}_{pr} = -\mathbf{Q} \tag{34}$$

where $\mathbf{Q} = \mathbf{Q}^{\top} > \mathbf{0}$. By imposing a negative semi-definite condition on the Lie derivative of the Lyapunov function, the PMRAC adaptive law can be written as:

$$\dot{\hat{\Theta}} = \gamma_{\Theta}\,\Phi\,\hat{\mathbf{e}}^{\top}\mathbf{P}_{pr}\,\mathbf{B}_{\mathbf{p}}. \tag{35}$$

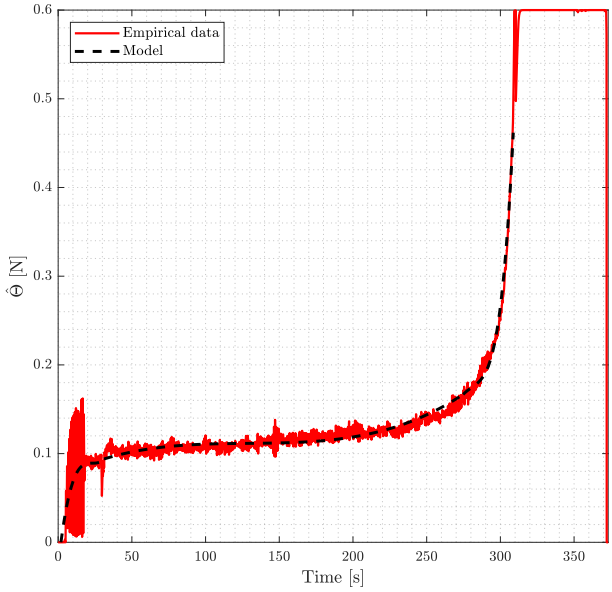
The PMRAC design improves transient performance over the standard MRAC and modifies the baseline controller input only when the plant uncertainties are not compensated for [22].

4. Simulations and experimental tests

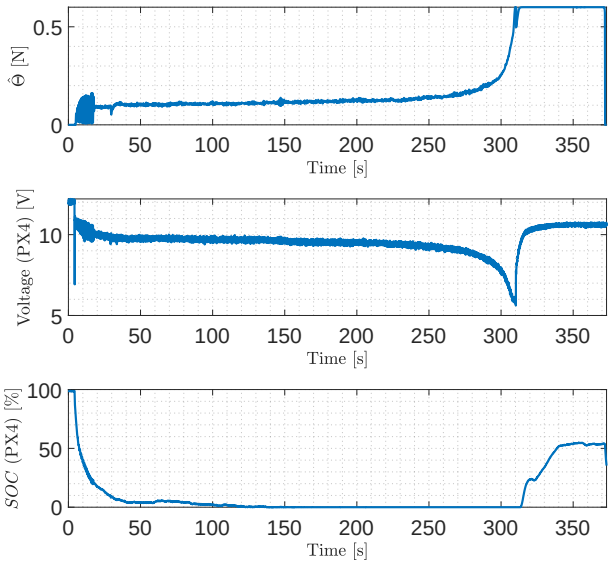
To validate the effectiveness of the presented controllers, preliminary tests were conducted by considering the plant's dynamics presented in (1).
Then, the implemented solutions were transferred and tested in the UAV simulator of the ANT-X [23] drone developed by the Aerospace Systems and Control Laboratory (ASCL) [24], which is based on the identified models of the drone's angular velocity dynamics acquired by using the blackbox Predictor-Based Subspace Identification (PBSID) 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 real drone shown in Figure 3.

Figure 3 – ANT-X [23] drone.

To validate the proposed discharge models and enable a direct comparison between the simulation and experimental data, an endurance test was performed by keeping the drone in hovering mode for as long as possible. Figure 4a illustrates the evolution of the estimated parameter $\hat{\Theta}$ over time during this test from which a model of battery discharge has been retrieved.
The actual remaining charge percentage was measured, showing a value of 8%, thus aligning with the theoretical expectation. Looking at the values, a more consistent battery performance is shown throughout the nominal battery charge region (90%–10%), maintaining a thrust loss value between 10% and 20%. Comparing this data with what is depicted in Figure 2a, it is possible to note a more restrained descent, probably due to the fact that this battery is specifically designed for drones, aiming to optimize the supplied voltage to the propellers as consistently as possible. The saturation of the estimated thrust loss $\hat{\Theta}$, is due to the presence of the projection operator [22, p. 239] in the adaptive law which constraints the evolution of the estimated parameter within a compact and convex set in order to increase robustness. Nevertheless, this saturation does not influence the drone's descent, which occurs gradually before the parameter estimate saturates, therefore increasing the bound of the projection operator does not provide benefits but could potentially reduce the robustness of the controller.



(a) Thrust lost parameter $\hat{\Theta}$ function with respect to time.

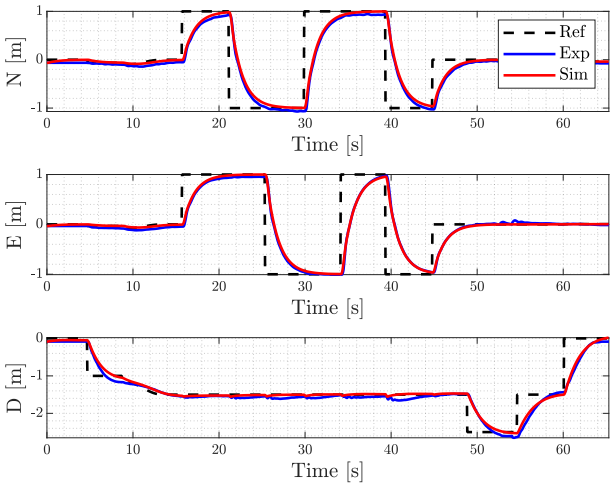


(b) Comparison between thrust loss parameter $\hat{\Theta}$, delivered voltage and SOC time histories.

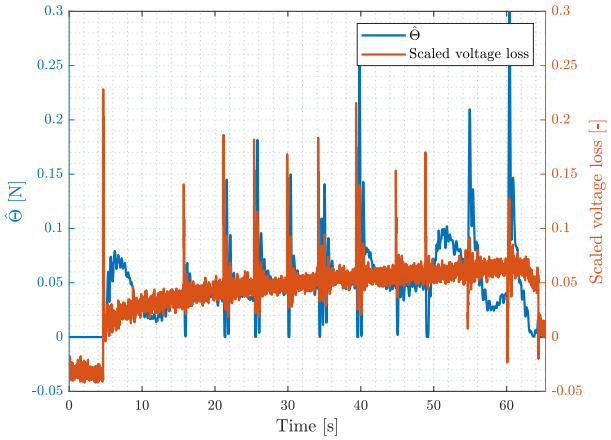
Figure 4 – Result for battery model validation during hovering flight test.

Similarly to what was done in [10], experimental data has been filtered, divided into three different curves and approximated with splines, separating the nearly linear region in the centre from the other

two highly non-linear regions at the edges, thus obtaining the model of thrust loss as a function of flight time represented by the dashed line in Figure 4a. So, through this model, it is possible to understand the remaining flight time starting from the estimate of the thrust loss, in the absence of external disturbances.
Moreover, during the same test other output coming from the drone PX4 autopilot [25] were extracted and compared to the estimated parameter of thrust loss.
Notably, the estimation of SOC turned out to be entirely inaccurate, especially when compared with the estimation of thrust loss, as can be seen in Figure 4b. This highlights how obtaining reliable estimates of SOC and SOE from simple models is not as straightforward. By using computationally expensive models the estimates can be improved but this entails an increase of the required computational power and, consequently, of weight, or an update of the estimate at lower frequencies, which does not seem advisable for short-duration missions like those of small drones, especially near complete battery discharge when thrust experiences a vertical collapse. In contrast, estimating the thrust loss parameter not only comes almost effortlessly, as it is a parameter the adaptive controller requires to enhance drone performance, but it also appears to be indicative of the battery state, as expected. Of course, the estimation of this parameter serves as a "direct" measure of the battery state only in the absence of external disturbances, such as wind, since the controller as formulated would not be able to distinguish and separate these contributions. Even observing the behaviour of the voltage curve and of the thrust loss parameter in Figure 4b, it is evident that the trends are quite similar, albeit inverted. Scaling the voltage trend and considering the voltage lost rather than supplied, the similarity between these trends is confirmed, underscoring the strong correlation between battery voltage and generated thrust.
Then, simulation results exploiting the obtained empirical model of the battery discharge have been compared with the results of experimental tests concerning a sequence of steps, highlighting the effectiveness of the simulator obtained through blackbox PBSID method.



(a) Response comparison to sequence of steps.



(b) Thrust loss parameter $\hat{\Theta}$ and voltage loss correlation.

Figure 5 – Results for the comparison between simulation and experimental response to step setpoint in position.

Lastly, the relationship between voltage and thrust loss shown in Figure 5b confirms the proportionality between these two quantities. In fact, through the same scaling, it can be observed how not only the trends are extremely similar, but also the peaks of both curves are located at the same time instants, namely when the drone is requested to perform a position step. This only further validates what was previously obtained regarding the proportionality between thrust loss and the voltage supplied to the propellers in the absence of external disturbances.

5. Mission Management

The mission that solves the persistent trajectory tracking problem, considering the battery discharge and the possibility of recharging at dedicated pads, is divided into various modes, each of which

has a defined domain based on the value of the parameter estimated by the adaptive controller, the UAV position and the resulting position error with respect to the reference trajectory. The modes are denoted by the letter $q \in [-1,0,1,2,3,4]$ and are grouped in three macro-modes: the macro mode for take-off, the main mode that is the trajectory tracking one $q = 1$ and the macro mode for safe landing. The macro mode for take-off is split in the actual take-off from ground $q = -1$ and in the approach towards the starting point of the trajectory $q = 0$, while the macro mode for safe landing, drawing inspiration from the work done in [26], is split in three sub-modes, the approach towards the charging station on which to land $q = 2$, the landing mode $q = 3$ and the disarm mode $q = 4$.
The overall mission scheme is represented in Figure 6.

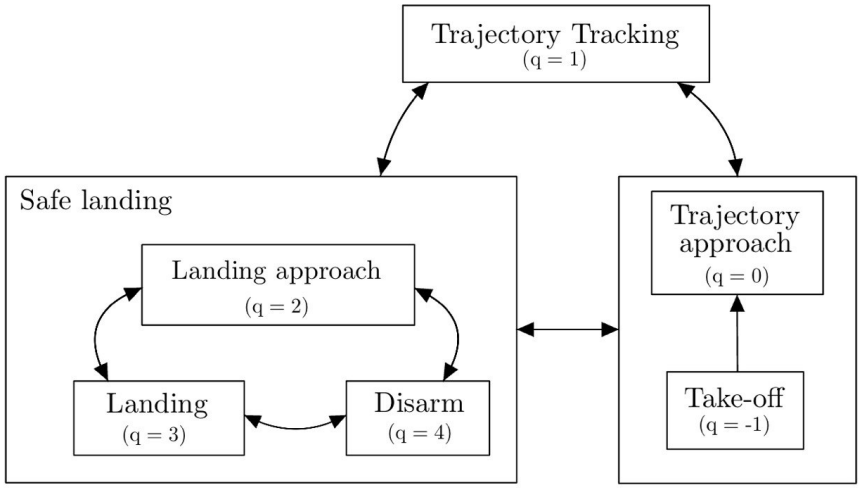


Figure 6 – Scheme of the overall switch logic.

The continuity of a surveillance mission over a large area, even after a charging stop, is ensured by applying the technique adopted in [27]. In this approach, a generalized time is defined, which only elapses when the drone is in the main mode, in the case of a surveillance mission, the trajectory tracking mode. In mathematical terms, the generalized time dynamics is described by:

$$\dot{t}_{gen} = s, \quad \text{where} \quad \begin{cases} s = 1, & \text{if} \quad q = 1 \\ s = 0, & \text{else} \end{cases}. \tag{36}$$

Before describing all the modes, some preliminary considerations can be done. The main objective of the mission is trajectory tracking performed in $q = 1$, while in all the other modes position setpoint stabilization is the goal. The solution proposed leverages the same setpoint generation control law for all the modes, adjusting in different ways the parameters to pursue either trajectory tracking or setpoint stabilization. To define the control law for the setpoint generation to be used by the controllers described in Section 3.1 and Section 3.2, the trajectory tracking is first considered.
It is assumed that control laws for T_c that enable the asymptotic tracking of any bounded velocity trajectory, denoted as $\mathbf{v}_r(t) \in \mathbb{R}^3$ exist. This assumption allows to formulate the tracking problem independently of the specific UAV actuation mechanism, relying on the kinematic model:

$$\dot{\mathbf{p}}_d = \mathbf{v}_d, \tag{37}$$

where $\mathbf{v}_d$ is a virtual input representing the drone's velocity in the inertial frame, utilized for control design purposes. Under this assumption, the following strategy can be used to track any sufficiently smooth reference trajectory, denoted as $t \to \begin{bmatrix} \mathbf{p}_r(t)^\top & \mathbf{v}_r(t)^\top & \mathbf{a}_r(t)^\top \end{bmatrix}^\top \in \mathbb{R}^9$ such that $\dot{\mathbf{p}}_r(t) = \mathbf{v}_r(t)$ and $\dot{\mathbf{v}}_r(t) = \mathbf{a}_r(t)$.
Defining the tracking error as:

$$\varepsilon(t) \coloneqq \mathbf{p}(t) - \mathbf{p}_r(t_{gen}), \tag{38}$$

the corresponding dynamics of the error can be rewritten as

$$\dot{\varepsilon}(t) = \mathbf{v}(t) - \mathbf{v}_r(t_{gen}), \tag{39}$$

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. It must be noted that the desired trajectory is parametrized by the generalized time whose dynamics is defined in (36).
The proposed control law is expressed by

$$\mathbf{v}_d(t) = -v_M \tanh\left(\frac{\mathbf{K}_\varepsilon\, \varepsilon(t)}{v_M}\right) + \mathbf{v}_r(t_{gen}), \tag{40}$$

where $v_M \in \mathbb{R}_{>0}$ represents the saturation level for the velocity, $\mathbf{K}_\varepsilon \in \mathbb{R}^{3\times 3}$ is a diagonal matrix containing positive gains and the hyperbolic tangent function has been preferred over a simple saturation function in order to obtain a smoother trajectory to be followed. It must be noted that the control law is a sum of a quasi time-optimal control law [28] and a feedforward term of the reference trajectory velocity. Finally, the acceleration setpoint $\mathbf{a}_d(t)$, can be computed by deriving (40):

$$\mathbf{a}_d(t) = -\mathbf{K}_\varepsilon \operatorname{sech}\left(\frac{\mathbf{K}_\varepsilon\, \varepsilon(t)}{v_M}\right) \dot{\varepsilon}(t) + \mathbf{a}_r(t_{gen}), \tag{41}$$

while, for the position setpoint $\mathbf{p}_d(t)$, numerical integration will be used.
So, the overall dynamics of each mode and the corresponding outputs $\mathbf{y}$ can be written as:

$$\begin{cases} \dot{\mathbf{p}}_d = \mathbf{v}_d \\ \dot{t}_{gen} = s \\ \mathbf{y} = \begin{bmatrix} \mathbf{p}_d(t)^\top & \mathbf{v}_d(t)^\top & \mathbf{a}_d(t)^\top \end{bmatrix}^\top \end{cases} \tag{42}$$

where $\mathbf{v}_d(t)$ and $\mathbf{a}_d(t)$ are computed as in (40) and in (41), respectively. The output $\mathbf{y}$ is used as setpoint for the controllers described in Section 3.1 and Section 3.2.

Remark 1. It can be observed that through this common choice of the control input, each mode can have different values in the $\mathbf{K}_\varepsilon$ matrix, in the errors ε and $\dot{\varepsilon}$, and in the saturation velocity v_M to adapt to the required task and performance. Additionally, excluding the main mode where tracking a trajectory is required, in the other modes the goal is setpoint stabilization and thus $\mathbf{v}_r(t_{gen})$ and $\mathbf{a}_r(t_{gen})$ will be zero, and the hyperbolic tangent will only create a smooth trajectory leading to the required point $\mathbf{p}_r(t_{gen})$.

A brief overview of the presented modes is reported hereafter.

5.1 Trajectory tracking mode

This mode represents the main goal of the mission, and actual trajectory tracking is pursued through control laws (40) and (41) with ε and $\dot{\varepsilon}$ computed as in (38) and (39).
The domain for the mode can be defined as follows:

$$C_1 := \left\{\hat{\Theta} < \Delta T_M;\ \ \mathbf{p} \in \mathbb{R}^3 : |\mathbf{p} - \mathbf{p}_r(t_{gen})| \leq R_t\right\}, \tag{43}$$

where R_t represents the maximum acceptable tracking error for considering successful the required task, ΔT_M is the threshold value of the estimated parameter at which the drone must interrupt the mission in order to return safely to the charging station.
Letting ΔT_M to be a function of the distance from the charging station, the drone can optimize 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 is possible to choose the right point to end the mission based on the remaining time until battery discharge. In order to consider flight stages different from hovering, a factor must be used to scale this threshold to increase safety, at the expense of longer endurance. The selected formula is:

$$\Delta T_M = \min(\Delta T_M{}^{Max}, \Delta T_{fun}(K(T_{batt} - T_{land} - |\Psi_\perp|/v_M))). \tag{44}$$

In the above formula ΔT_{fun} represents the empirical model of Figure 4a, 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, $|\Psi_{\perp}|$ represents the planar component of the distance from the charging station such that the last term considers the time that the drone implies to reach the charging station from the point where the mission is interrupted (T_{dist}). The minimum function and the threshold ${\Delta T_M}^{Max}$ are used to further increase the reliability of the autonomous mission management as explained in the next section. A graphical representation of the function is shown in Figure 7.

5.2 Landing macro mode

The macro mode for safe landing, drawing inspiration from the work done in [26], is split in three sub-modes, the approach towards the charging station on which to land $q = 2$, the landing mode $q = 3$ and the disarm mode $q = 4$.
The domain definition for all these modes depends mainly on the value of the thrust loss estimated parameter. Some threshold are defined in order to distinguish if it is possible to carry out a safe landing or if an emergency landing is necessary. Once the estimated parameters is greater than the threshold ΔT_M computed as explained in (44), the safe landing is initiated. Furthermore, the threshold ΔT_{crit} that triggers the landing procedure at the current location of the drone, given the impossibility of a safe landing on the pad is also defined. The entire decision process can be explained looking at Figure 7.

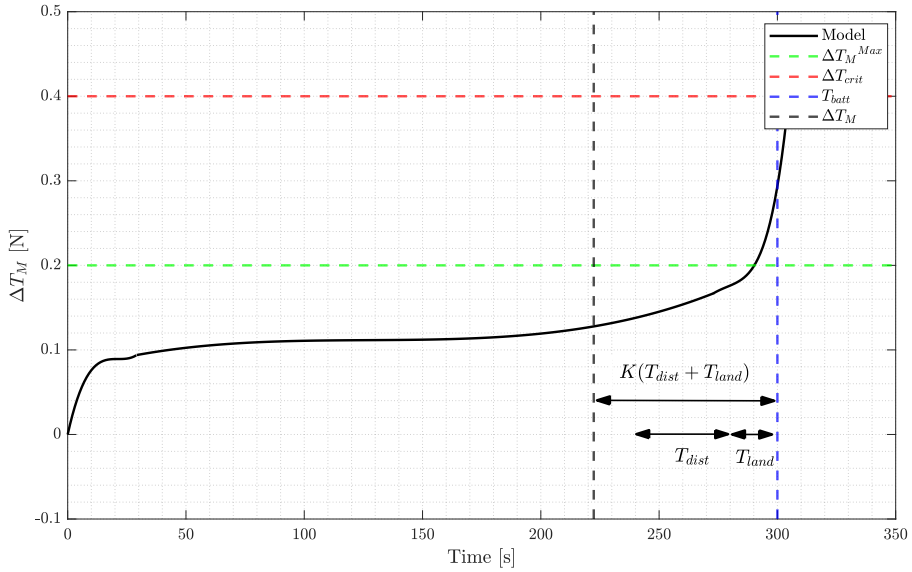


Figure 7 – Visual representation of ΔT_M.

In Figure 7 the continuous black line represents the battery model, the red dashed line represents the maximum acceptable threshold value ΔT_{crit} that triggers the landing procedure at the current location of the drone, given the impossibility of a safe landing on the pad. Moreover, the blue dashed line represents the total time of the ideal mission, including take-off and landing, ensuring that the drone lands with 30% of remaining charge, and the black dashed line represents the time associated with the variable threshold value: if this value is greater, and therefore above the red line, the threshold will be equal to ${\Delta T_M}^{Max}$, while if it falls below, it will represent the value itself in which the safe landing mode starts. So, applying the formula in (44), the goal would be to land when the thrust loss is approximately 30%. Also, it can be noticed that by using the minimum function, a maximum value of ${\Delta T_M}^{Max}$ has been set for the activation of the landing mode to further increase safety in the final stages of the mission.

5.2.1 Landing approach mode

In sub-mode $q = 2$, the so-called landing approach mode, the follower is far from the landing/charging station and has to get close to a safe distance. Therefore, its domain of application will be defined as

follows:

$$C_2 := \left\{ \Delta T_M \leq \hat{\Theta} \leq \Delta T_{crit};\ \mathbf{p} \in \mathbb{R}^3 : \left\{ \begin{array}{lll} |\Psi_\perp| \geq r_m & \text{if} & z \geq h_a \\ |\Psi_\perp| \geq r_t & \text{if} & 0 \leq z \leq h_a \end{array} \right\} \right\}. \tag{45}$$

As can be seen, the conditions that determine the domains, and therefore also dictate the switches between modes, are the position of the drone and the thrust loss estimated parameter. Regarding the first one h_a represents the altitude at which the approach phase should end, while $r_m < r_t$ is the distance at which the landing approach phase should end, $|\Psi_\perp|$ represents the planar component of the distance from the ground station and z the altitude of the drone. With the choice of $r_m < r_t$ the logical switch between modes occur with hysteresis in order to avoid chattering phenomena, caused by possible fluctuations in the estimated parameter, following what was proposed in [26].
The control laws applied during this mode are (40) and (41), but with $\mathbf{v}_r(t_{gen}) = \mathbf{a}_r(t_{gen}) = 0$, such that

$$\mathbf{v}_d(t) = -v_M \tanh\left(\frac{\mathbf{K}_\varepsilon\, \varepsilon(t)}{v_M}\right) \tag{46}$$

$$\mathbf{a}_d(t) = -\mathbf{K}_\varepsilon \operatorname{sech}\left(\frac{\mathbf{K}_\varepsilon\, \varepsilon(t)}{v_M}\right) \dot{\varepsilon}(t), \tag{47}$$

$\dot{\varepsilon} = \mathbf{v}$ and ε is defined as:

$$\varepsilon = \begin{bmatrix} \Psi_\perp \\ 0 \end{bmatrix} \tag{48}$$

so that stabilization of the position at an height z above the charging station is performed.

5.2.2 Landing mode

In the landing sub-mode, indicated with $q = 3$, the drone starts to decrease its altitude, maintaining itself over the surface of the charging platform. The domain related to this mode can be expressed as:

$$C_3 := \left\{ \mathbf{p} \in \mathbb{R}^3 : \left\{ \begin{array}{llll} |\Psi_\perp| \leq r_t, & z \geq h_a & \text{if} & \Delta T_M \leq \hat{\Theta} \leq \Delta T_{crit} \\ & z \geq h_a & \text{if} & \hat{\Theta} \geq \Delta T_{crit} \end{array} \right\} \right\}. \tag{49}$$

It can be noted, as specified earlier in the definition of ΔT_{crit}, that once the estimated parameter exceeds this threshold value, the landing will no longer occur directly above the charging base. Instead, the drone will be required only to land nearby its actual position, without imposing any specific coordinates in the North-East plane. So, even the disarm mode will use the same division, distinguishing between a nominal landing and an emergency landing.
The control laws applied during this mode are (40) and (41), but with $\mathbf{v}_r(t_{gen}) = \mathbf{a}_r(t_{gen}) = 0$ in order to achieve setpoint stabilization, eventually changing the velocity limit v_M and the gains $\mathbf{K}_\varepsilon$ to enhance the required performance of this mode.
In case of nominal landing the UAV is stabilized at the position of the charging platform while in case of emergency landing at the ground position below its current height. To do this, the error ε in this mode is computed as follows, depending on whether it is a nominal or emergency landing:

$$\begin{cases} \varepsilon = \begin{bmatrix} \Psi_\perp^\top & z \end{bmatrix}^\top & \text{if} \quad j = 0 \\ \varepsilon = \begin{bmatrix} 0 & 0 & z \end{bmatrix}^\top & \text{if} \quad j = 1 \end{cases}, \tag{50}$$

where $j \in \{0, 1\}$ is a logical operator that assumes the value 1 when the drone starts the emergency landing procedure. The introduction of another logical operator was mandatory in order to avoid chattering issue between the nominal and emergency mode, also making the introduction of new modes unnecessary, allowing the utilization of the existing landing mode, thus minimizing the final logic scheme without compromising its functionality.

5.2.3 Disarm mode

The disarm sub-mode will be the last phase, which is indicated with $q = 4$. The drone has reached an altitude sufficiently close to the ground and so the disarm command is activated both for nominal landing and for emergency landing. The domain of the disarm mode is then defined as:

$$C_4 := \left\{ \mathbf{p} \in \mathbb{R}^3 : \begin{Bmatrix} |\Psi_\perp| \leq r_t, & z \leq h_a & \text{if} & \Delta T_M \leq \hat{\Theta} \leq \Delta T_{crit} \\ & z \leq h_a & \text{if} & \hat{\Theta} \geq \Delta T_{crit} \end{Bmatrix} \right\}. \tag{51}$$

After disarming the drone, the battery could be recharged using a dedicated charging pad or replaced. The control law is the same as the landing mode. It must be noted that, while the actual disarm is performed turning off the motors, for simulation purposes, the drone does not land on the ground/pad but hovers at zero height in the coordinates of the charging base. Once this phase is completed, the drone engages the take-off mode, which is yet to be defined, so that the mission can be resumed.

5.3 Take-off macro mode

The macro mode for take-off is split in the actual take-off from ground $q = -1$ and in the approach towards the starting point of the trajectory $q = 0$.

5.3.1 Take-off mode

The first sub-mode of the take-off concerns the actual take-off ($q = -1$), which from the ground level brings the drone to the altitude of the trajectory z_{traj}. Its domain is defined as

$$C_{-1} := \left\{ \hat{\Theta} \leq \Delta T_M;\ \mathbf{p} \in \mathbb{R}^3 : \left| z - z_{traj} \right| \geq \varepsilon_z \right\}, \tag{52}$$

where ε_z represents the maximum error acceptable for the drone vertical position to match the trajectory altitude in order to switch to the trajectory approach mode.
The control laws applied during this mode are (40) and (41), but with $\mathbf{v}_r(t_{gen}) = \mathbf{a}_r(t_{gen}) = 0$ in order to achieve position setpoint stabilization, eventually changing the velocity limit v_M and the gains $\mathbf{K}_\varepsilon$ to enhance the required performance of this mode as in the landing mode case.
In case the UAV performed a nominal landing, during the following take-off the UAV is stabilized at the in-plane charging station position but at the height of the desired trajectory, while in case an emergency landing was performed, the drone is stabilized at the trajectory height above its current position. To do this, the error ε in this mode is computed as follows, depending on whether there was a nominal or emergency landing:

$$\begin{cases} \varepsilon = \begin{bmatrix} \Psi_\perp^\top & z_{traj} \end{bmatrix}^\top & \text{if} \quad j = 0 \\ \varepsilon = \begin{bmatrix} 0 & 0 & z_{traj} \end{bmatrix}^\top & \text{if} \quad j = 1 \end{cases}, \tag{53}$$

where $j \in \{0, 1\}$ is a logical operator that assumes the value 1 when the drone performed the emergency landing procedure.

5.3.2 Trajectory approach mode

After take-off, the subsequent phase is the so-called trajectory approach ($q = 0$), that brings the drone to the initial point of the trajectory, from which the main mode will subsequently restart.
The domains is defined as:

$$C_0 := \left\{ \hat{\Theta} \leq \Delta T_M;\ \mathbf{p} \in \mathbb{R}^3 : |\mathbf{p} - \mathbf{p}_r(t_{gen})| \geq R_m \right\}, \tag{54}$$

where R_m represent the maximum error acceptable for the drone position to match the trajectory initial setpoint $\mathbf{p}_r(t_{gen})$ for the tracking mode. Therefore, by choosing $R_m < R_t$ the chattering phenomena can be prevented in the switch between the trajectory approach mode and the trajectory tracking one.
It can be noted that thanks to the choice of the generalized time, the value of t_{gen} will remain unchanged for the entire time spent outside the tracking mode, allowing for the resumption of the mission to restart tracking from where it was interrupted.
To perform position stabilization of the drone at the starting point of the desired trajectory, the control law used are the one of (46) and (47) with the velocity error $\dot{\varepsilon} = \mathbf{v}$ and the position error computed as $\varepsilon = \mathbf{p} - \mathbf{p}_r(t_{gen})$.

5.4 Hybrid automaton

In the definition of the hybrid automaton that describes the mission management system, once that the domains of all nodes and the related control inputs are known, it is necessary to define the Guard conditions that allow transitioning from one mode to another.
As defined in [29], considering the set of modes Q which is composed by all the modes described previously $q \in Q := \{-1,0,1,2,3,4\}$ and the domain map $C_q : Q \rightrightarrows \mathbb{R}^3$ which gives for each $q \in Q$ the set C_q in which the continuous state $\mathbf{p} \in \mathbb{R}^3$ evolves, the set of edges $E \subset Q \times Q$ identifies the pairs (q,q') such that a transition from the mode q to the mode q' is possible.
First of all, it is possible to start with the safe landing mode taking as a reference what was done in [26].
The set $E_{SL} := \{(2,3),(3,2),(2,4),(4,2),(3,4),(4,3),(4,4)\}$ contains all possible combinations internal to the safe landing mode. Therefore, a Guard map, which contain all the Guard condition Guard:$E \rightrightarrows \mathbb{R}^3$, identifies for each edge $(q,q') \in E$ the set Guard(q,q') to which the continuous state must belong so that a transition from q to q' can occur. The various Guard conditions that allow to switch from a mode to another one inside the safe-landing overall mode, are:

$$\text{Guard}(2,3) := \left\{ \mathbf{p} \in \mathbb{R}^3 : |\Psi_\perp| \leq r_m, \quad \begin{matrix} z \geq h_a & \text{if} \quad \Delta T_M \leq \hat{\Theta} \leq \Delta T_{crit} \\ z \geq h_a & \text{if} \quad \hat{\Theta} \geq \Delta T_{crit} \end{matrix} \right\},$$

$$\text{Guard}(3,2) := \left\{ \mathbf{p} \in \mathbb{R}^3 : |\Psi_\perp| \geq r_t, \quad z \geq 0, \quad \left\{ \begin{matrix} \Delta T_M \leq \hat{\Theta} \leq \Delta T_{crit}, & \text{if} \quad j = 0 \\ \Delta T_M \leq \hat{\Theta} \leq \Delta T_{crit} - \delta_{crit}, & \text{if} \quad j = 1 \end{matrix} \right\} \right\},$$

$$\text{Guard}(2,4) := \left\{ \mathbf{p} \in \mathbb{R}^3 : \left\{ \begin{matrix} |\Psi_\perp| \leq r_t, & 0 \leq z \leq h_a & \text{if} \quad \Delta T_M \leq \hat{\Theta} \leq \Delta T_{crit} \\ & 0 \leq z \leq h_a & \text{if} \quad \hat{\Theta} \geq \Delta T_{crit} \end{matrix} \right\} \right\},$$

$$\text{Guard}(4,2) := \left\{ \begin{matrix} \mathbf{p} \in \mathbb{R}^3 : \left\{ \begin{matrix} |\Psi_\perp| \geq r_m & \text{if} \quad z \geq h_a \\ |\Psi_\perp| \geq r_t & \text{if} \quad 0 \leq z \leq h_a \end{matrix} \right\}, \left\{ \begin{matrix} \Delta T_M \leq \hat{\Theta} \leq \Delta T_{crit}, & \text{if} \quad j = 0 \\ \Delta T_M \leq \hat{\Theta} \leq \Delta T_{crit} - \delta_{crit}, & \text{if} \quad j = 1 \end{matrix} \right\} \end{matrix} \right\},$$

$$\text{Guard}(3,4) := \left\{ \mathbf{p} \in \mathbb{R}^3 : \left\{ \begin{matrix} |\Psi_\perp| \leq r_t, & 0 \leq z \leq h_a, & \Delta T_M \leq \hat{\Theta} \leq \Delta T_{crit} & \text{if} \quad j = 0 \\ |\Psi_\perp| \leq r_t, & 0 \leq z \leq h_a, & \Delta T_M \leq \hat{\Theta} \leq \Delta T_{crit} - \delta_{crit} & \text{if} \quad j = 1 \\ & z \leq h_a & \text{if} \quad \hat{\Theta} \geq \Delta T_{crit} & \end{matrix} \right\} \right\},$$

$$\text{Guard}(4,3) := \left\{ \mathbf{p} \in \mathbb{R}^3 : \left\{ \begin{matrix} |\Psi_\perp| \leq r_t, & z \geq h_a, & \Delta T_M \leq \hat{\Theta} < \Delta T_{crit} & \text{if} \quad j = 0 \\ |\Psi_\perp| \leq r_t, & z \geq h_a, & \Delta T_M \leq \hat{\Theta} \leq \Delta T_{crit} - \delta_{crit} & \text{if} \quad j = 1 \\ & z \geq h_a & \text{if} \quad \hat{\Theta} \geq \Delta T_{crit} & \end{matrix} \right\} \right\},$$

$$\text{Guard}(4,4) = \text{Guard}(3,4),$$

where $\delta_{crit} \in \mathbb{R}_{>0}$ is a small value compared to ΔT_{crit}, which is used to create hysteresis in the switch, thereby avoiding chattering phenomena.
Regarding the switch between the sub-modes contained in the take-off macro mode, a unique Guard condition is defined for the set of edges $E_{TO} = \{(-1,0),(0,-1)\}$, so once the take-off is completed, the drone will no longer use this mode. In mathematical terms, it can be expressed as:

$$\begin{aligned} &\text{Guard}(-1,0) := \left\{ \hat{\Theta} \leq \Delta T_M;\ \mathbf{p} \in \mathbb{R}^3 : \left| z - z_{traj} \right| \leq \varepsilon_z \right\}, \\ &\text{Guard}(0,-1) := \varnothing. \end{aligned}$$

Then, to increase the mission management robustness, from any modes between take-off, trajectory approach and trajectory tracking, the drone must always be able to decide whether to abort the mission if the estimated loss exceeds the established threshold values, both for nominal and emergency landings. So, all subsequent modes must lead to the modes of landing. In fact, the switch should

bring the drone to the proper safe landing mode ($q = \{2,3,4\}$) depending on its altitude z, distance from the base and estimated thrust loss $\hat{\Theta}$. This logic can be formulated mathematically as:

$$\begin{aligned}
&\text{Guard}(-1,2) = \text{Guard}(0,2) = \text{Guard}(1,2) = C_2 \\
&\text{Guard}(-1,3) = \text{Guard}(0,3) = \text{Guard}(1,3) = \text{Guard}(2,3) \\
&\text{Guard}(-1,4) = \text{Guard}(0,4) = \text{Guard}(1,4) = \text{Guard}(2,4)
\end{aligned}$$

Proceeding in chronological order, the condition that allows entering the tracking mode from the take-off one can be defined as:

$$\text{Guard}(0,1) \coloneqq \left\{\hat{\Theta} \leq \Delta T_M;\ \ \mathbf{p} \in \mathbb{R}^3 : |\mathbf{p} - \mathbf{p}_r(t_{gen})| \leq R_m\right\},$$

$$\text{Guard}(-1,1) = \text{Guard}(0,1).$$

While regarding the transition from the landing mode to the tracking mode, the switch conditions can be defined as:

$$\text{Guard}(2,1) \coloneqq \left\{\hat{\Theta} \leq \Delta T_M - \delta_M;\ \ \mathbf{p} \in \mathbb{R}^3 : |\mathbf{p} - \mathbf{p}_r(t_{gen})| \leq (R_t - R_m)/2\right\},$$

$$\text{Guard}(3,1) = \text{Guard}(4,1) = \text{Guard}(2,1),$$

where, as done previously, $\delta_M \in \mathbb{R}_{>0}$ has been defined to avoid chattering phenomena. Moreover, as can be seen, the intermediate value between R_t and R_m has been chosen as the error to evaluate whether to return to tracking mode or trajectory approach mode. In fact, even the switches that lead from the safe landing mode to the latter can be defined as:

$$\text{Guard}(2,0) \coloneqq \left\{\hat{\Theta} \leq \Delta T_M - \delta_M;\ \ \mathbf{p} \in \mathbb{R}^3 : |\mathbf{p} - \mathbf{p}_r(t_{gen})| \geq (R_t - R_m)/2\right\},$$

$$\text{Guard}(3,0) = \text{Guard}(4,0) = \text{Guard}(2,0).$$

It can be noted that the choice of this intermediate parameter will not create chattering problems, given that the switch from the landing mode to one of the other modes depends on the value of the estimated parameter.
The last switch to consider is the one that leads from the tracking phase to the trajectory approach phase. This connection between the two modes is necessary so that if the drone fails to follow the expected trajectory, perhaps due to sudden external disturbances such as wind gusts, the trajectory is frozen until the drone returns in a position within an acceptable error:

$$\text{Guard}(1,0) \coloneqq \left\{\hat{\Theta} \leq \Delta T_M;\ \ \mathbf{p} \in \mathbb{R}^3 : |\mathbf{p} - \mathbf{p}_r(t_{gen})| \geq R_t\right\}.$$

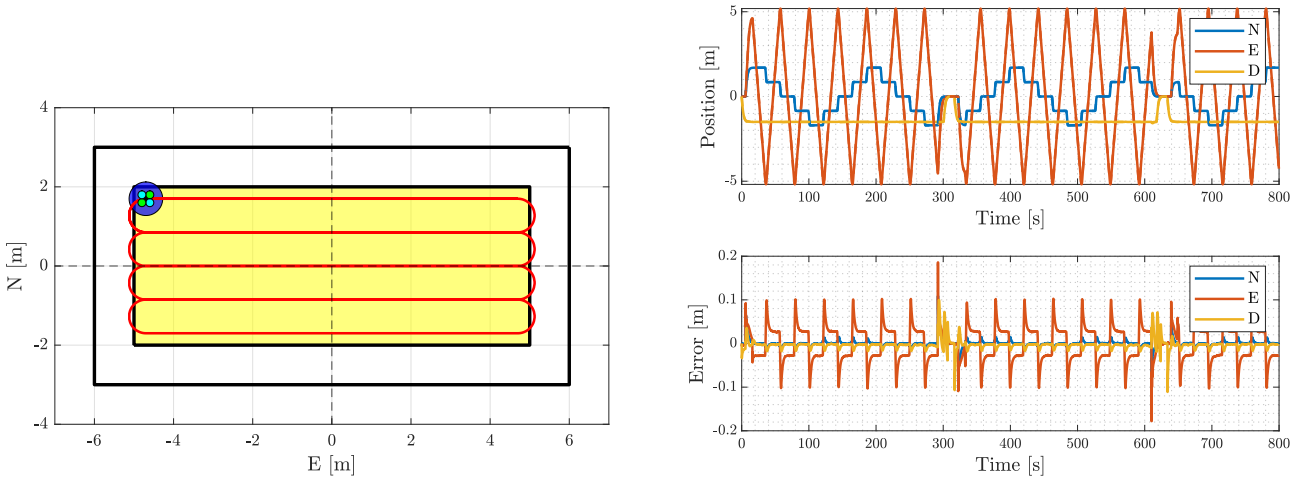
Finally, to complete the definition of the hybrid automaton, the definition of reset map will be necessary. The reset map Reset:$E \times \mathbb{R}^3 \to \mathbb{R}^3$ identifies for each edge $(q,q') \in E$ the value to which the continuous state is set during a transition from mode q to mode q'. For all the modes presented in this paper, the continuous state remains constant at each jump, therefore the Reset$(q,q',\cdot)$ map can be taken to be the identity.

6. Mission simulation results

Different trajectories have been simulated to prove the effectiveness of the mission manager regardless of the trajectory: a circular trajectory simulating loitering around a specific point, and a more complex trajectory simulating the surveillance of a rectangular area using a pattern of line segments. The parameters used in the simulations are the ones reported in Table 1.
The results obtained for the simulation of the pattern trajectory shown in Figure 8a are shown hereafter.
Comparing the endpoint of the previous missions with the starting point of the subsequent ones in Figure 8b, it will be immediate to assess the effectiveness of the mission management system in resuming the mission from where the previous one was interrupted.

Table 1 – Values of the parameters used in simulations.

Thrust loss threshold	Reference trajectory	Landing	Tracking errors
$\Delta T_M{}^{Max} = 0.20$	$\mathbf{K}_\varepsilon = 0.5\,\mathbf{I}_3$	$r_m = 0.15\,\mathrm{m}$	$R_m = 0.07\,\mathrm{m}$
$\delta_M = 0.05$	$v_{traj} = 0.5\,\mathrm{m/s}$	$r_t = 0.25\,\mathrm{m}$	$R_t = 0.13\,\mathrm{m}$
$\Delta T_{crit} = 0.35$	$v_M = 0.8\,\mathrm{m/s}$	$h_a = 0.05\,\mathrm{m}$	$\varepsilon_z = 0.1\,\mathrm{m}$
$\delta_{crit} = 0.03$	$z_{traj} = -1.5\,\mathrm{m}$		



(a) Pattern trajectory used in simulation.

(b) Pattern trajectory tracking errors.

Figure 8 – Pattern trajectory and position results in simulation.

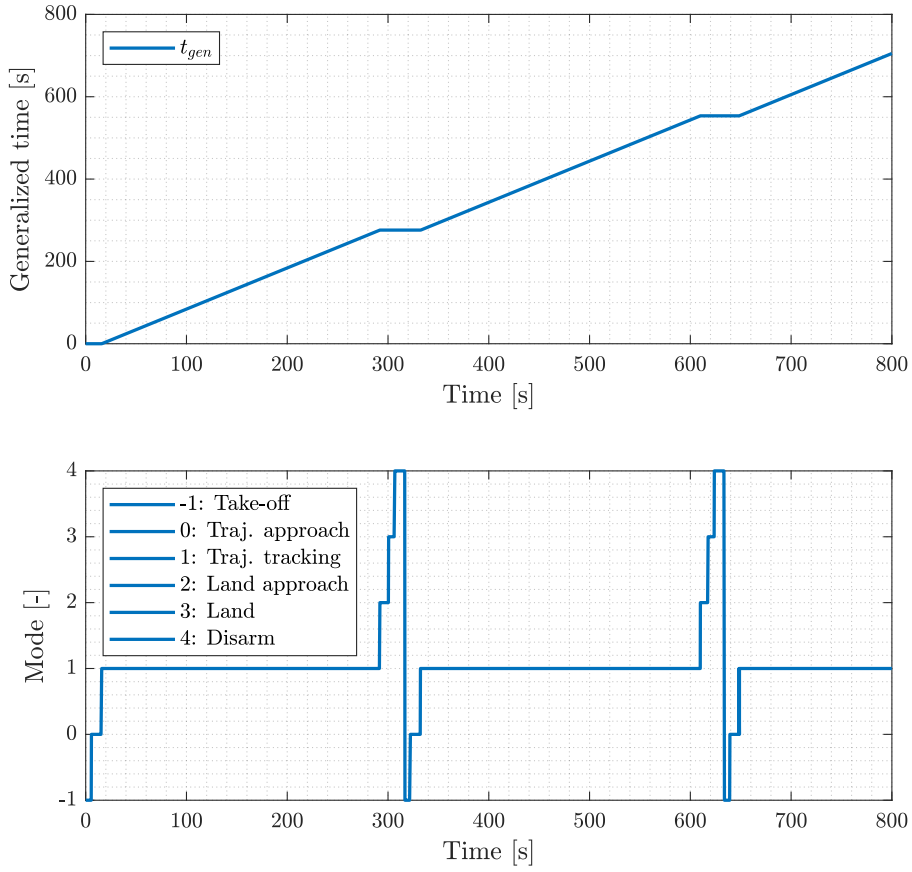


Figure 9 – Generalized time and active modes.

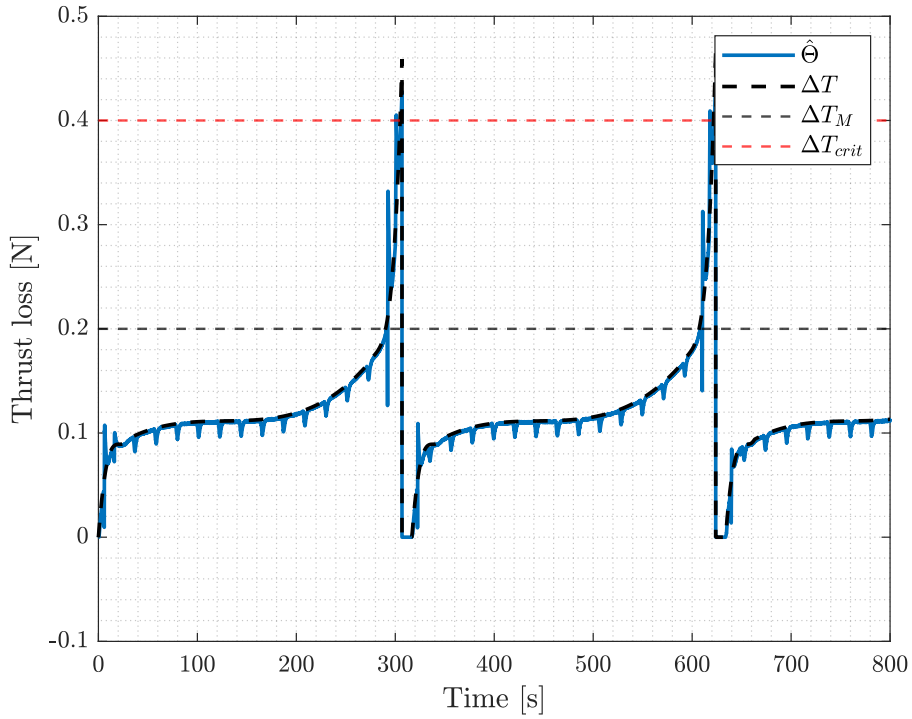


Figure 10 – Estimated thrust loss and real one in consecutive missions.

All of this is, of course, due to the generalized time dynamics resuming from the last value assumed when the mission was interrupted, as can be seen in Figure 9, where the generalized time is shown together with the time history of the active modes.
Furthermore, the drone was kept stationary on the charging base for about $10\,\mathrm{s}$, resetting the battery charge to the maximum, simulating a real battery recharge or replacement, as can be seen by looking at Figure 10. Certainly, in reality, recharging will take much longer than simulated, but this phase was added solely to validate the planner and not to simulate a realistic recharge. It has to be noted, that although the terrain is not modelled, and therefore the drone does not land on the ground/pad but, for simulation purposes, hovers at zero height in the coordinates of the charging base performing the battery charge reset, these two actions can be considered as equivalent. After demonstrating the correct functioning of the logic in the case of nominal discharge, additional tests have been carried out 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, instead, the drone will enter the emergency landing mode, simply attempting to land in its current location.

7. Conclusions

One of the objectives of this paper was to formulate a control law capable of compensating for performance degradation due to battery discharge in multirotor UAVs while solving trajectory tracking tasks. The adaptive control techniques presented have demonstrated the ability to maintain performance levels achieved under nominal conditions, even when considering the effects of battery discharge. Furthermore, this control technique not only achieves this result but also provides online estimation of thrust loss. This parameter has been successfully used in decision-making algorithms, enabling the drone to complete missions autonomously. The estimated parameter is particularly valuable when compared to what can be obtained from SOC or SOE estimators already implemented in autopilots such as PX4, where obtaining reliable estimates from simple models is challenging.
The other objective of the paper was to define a mission management system. The choice to divide the mission into various modes with different domains of application and objectives has proven to be effective. This success is partly due to the introduction of hysteresis in the transition functions between modes, which prevents chattering and subsequent performance degradation. Additionally, the robustness of the algorithm for generating the threshold value for mission termination has been note-

worthy, ensuring mission completion even under significantly anomalous conditions with discharge rates much faster than expected.

8. Contacts

Email addresses: alessandro.boldrini@mail.polimi.it, {davide.invernizzi,giovanni.gozzini}@polimi.it

9. Acknowledgement

This study was carried out within the Agritech National Research Center and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.4 – D.D. 1032 17/06/2022, CN00000022). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

10. Copyright Statement

References

[1] Hua M.-D, Hamel T, Morin P, and Samson C. Introduction to feedback control of underactuated VTOL vehicles: A review of basic control design ideas and principles. *IEEE Control systems magazine*, 33(1):61–75, 2013.

[2] Mahony R, Kumar V, and Corke P. Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE Robotics & Automation Magazine*, 19(3):20–32, 2012.

[3] Roza A and Maggiore M. A class of position controllers for underactuated VTOL vehicles. *IEEE Transactions on Automatic Control*, 59(9):2580–2585, 2014.

[4] Invernizzi D, Lovera M, and Zaccarian L. Geometric trajectory tracking with attitude planner for vectored-thrust VTOL UAVs. In *2018 Annual American Control Conference (ACC18)*, pages 3609–3614. IEEE, 2018.

[5] Li J and Li Y. Dynamic analysis and PID control for a quadrotor. In *2011 IEEE International Conference on Mechatronics and Automation (ICMA11)*, pages 573–578, 2011.

[6] Roberts A and Tayebi A. Adaptive position tracking of VTOL UAVs. *IEEE Transactions on Robotics*, 27(1):129–142, 2011.

[7] Tang D, Gong M, Yu J, and Li X. A power transfer model-based method for Lithium-ion battery discharge time prediction of electric rotatory-wing UAV. *Microelectronics Reliability*, 114:113832, 2020.

[8] Abeywickrama H. V, Jayawickrama B. A, He Y, and Dutkiewicz E. Comprehensive energy consumption model for unmanned aerial vehicles, based on empirical studies of battery performance. *IEEE Access*, 6:58383–58394, 2018.

[9] Traub L. W. Range and endurance estimates for battery-powered aircraft. *Journal of Aircraft*, 48(2):703–707, 2011.

[10] Podhradskỳ M, Bone J, Coopmans C, and Jensen A. Battery model-based thrust controller for a small, low cost multirotor unmanned aerial vehicles. In *Proceedings of the 2013 International Conference on Unmanned Aircraft Systems (ICUAS13)*, pages 105–113. IEEE, 2013.

[11] Pang S, Farrell J, Du J, and Barth M. Battery state-of-charge estimation. In *Proceedings of the 2001 American Control Conference (ACC01)*, volume 2, pages 1644–1649. IEEE, 2001.

[12] Dong G, Chen Z, Wei J, Zhang C, and Wang P. An online model-based method for state of energy estimation of Lithium-ion batteries using dual filters. *Journal of Power Sources*, 301:277–286, 2016.

[13] Hickey R and Jahns T. M. Direct comparison of state-of-charge and state-of-energy metrics for Li-ion battery energy storage. In *Proceedings of the 2019 IEEE Energy Conversion Congress and Exposition (ECCE19)*, pages 2466–2470, 2019.

[14] Gugan G and Haque A. Path planning for autonomous drones: Challenges and future directions. *Drones*, 7(3), 2023.

[15] Alyassi R, Khonji M, Karapetyan A, Chau S. C.-K, Elbassioni K, and Tseng C.-M. Autonomous recharging and flight mission planning for battery-operated autonomous drones. *IEEE Transactions on Automation Science and Engineering*, 20(2):1034–1046, 2023.

[16] Boukoberine M. N, Zhou Z, and Benbouzid M. A critical review on unmanned aerial vehicles power supply and energy management: Solutions, strategies, and prospects. *Applied Energy*, 255:113823, 2019.

[17] Meraglia S, Gozzini G, Liu S, and Invernizzi D. Adaptive control for high-performance trajectory tracking in multirotor UAVs. In *Proceedings of the 9th International Conference on Control, Decision and Information Technologies (CoDIT23)*, 2023.

[18] Hua M.-D, Hamel T, Morin P, and Samson C. A control approach for thrust-propelled underactuated vehicles and its application to VTOL drones. *IEEE Transactions on Automatic Control*, 54(8):1837–1853, 2009.

[19] Lopes R. V, Galvao R. K, Milhan A. P, Becerra V. M, and Yoneyama T. Modelling and constrained predictive control of a 3DOF helicopter. In *Proceedings of the XVI Congresso Brasileiro de Automática*, volume 123, pages 429–434, 2006.

[20] Lee T.-G and Huh U.-Y. An error feedback model based adaptive controller for nonlinear systems. In *Proceeding of the 1997 IEEE International Symposium on Industrial Electronics (ISIE97)*, pages 1095–1100. IEEE, 1997.

[21] Lavretsky E, Gadient R, and Gregory I. M. Predictor-based model reference adaptive control. *Journal of Guidance, Control, and Dynamics*, 33(4):1195–1201, 2010.

[22] Lavretsky E and Wise K. A. *Robust and adaptive control: with aerospace applications*. Springer, 2013.

[23] ANT-X website. [Online] `https://antx.it/`.

[24] ASCL-Polimi website. [Online] `http://ascl.daer.polimi.it/`.

[25] PX4 Autopilot User Guide. [Online] `https://docs.px4.io/main/en/`.

[26] Gozzini G, Invernizzi D, Panza S, Giurato M, and Lovera M. 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.

[27] Andreetto M, Fontanelli D, and Zaccarian L. Quasi time-optimal hybrid trajectory tracking of an n-dimensional saturated double integrator. In *Proceedings of the 2016 IEEE Conference on Control Applications (CCA16)*, pages 550–555, 2016.

[28] Forni F, Galeani S, and Zaccarian L. A family of global stabilizers for quasi-optimal control of planar linear saturated systems. *IEEE Transactions on Automatic Control*, 55(5):1175–1180, May 2010.

[29] Goebel R, Sanfelice R. G, and Teel A. R. Hybrid dynamical systems. *IEEE Control Systems Magazine*, 29(2):28–93, 2009.