

UNIVERSITÀ DEGLI STUDI DI NAPOLI
FEDERICO II

Field and Service Robotics: Final project

Aerial Robots: Quadrotor Control with External Disturbances Estimations
and Ground and Ceiling Effects Simulation

Riccardo Stucovitz, P38000181: <https://github.com/Stuko98>
Nicola Monetti, P38000238: <https://github.com/NicoMonetti>

A.Y. 2024/2025

Contents & Figures

Abstract	4
Chapter 1	4
Simulation Tools and Environment	4
X3D-BL Quadrotor	4
Environment Scenario	5
Trajectory planner	6
Collision detector	7
Chapter 2	8
Hierarchical control	8
Geometric control	12
Passivity-based control	14
Impact of "Filter and Derivative" Block on system performance	14
Robustness comparison	16
Chapter 3	16
Chapter 4	19
Hierarchical estimator-based control	19
Geometric estimator-based control	22
Passivity-and-estimator-based control	23
Chapter 5	23

List of Figures

1 X3D-BL quadrotor.	4
2 UAV scenario: (a) top view and (b) 3D perspective view scenario with trajectory path.	5
3 Wind external disturbance	6
4 Overall time law and its time derivatives	8
5 Hierarchical control system	9
6 Desired and effective pose, Hierarchical controller. $K_p = K_e = [I_3 \ I_3]$	10
7 Linear and angular errors, Hierarchical controller. $K_p = K_e = [I_3 \ I_3]$	10
8 Total thrust and torques, Hierarchical controller. $K_p = K_e = [I_3 \ I_3]$	11
9 Linear and angular errors, Hierarchical controller. $K_p = 10 [I_3 \ I_3]$, $K_e = 100 [I_3 \ I_3]$	11
10 Geometric control system	12
11 Linear and angular errors, Geometric controller. $K_p = K_v = K_r = K_w = I_3$	13
12 Linear and angular errors, Geometric controller. $K_p = K_v = 100I_3$, $K_r = 10I_3$, $K_w = I_3$	13
13 Passivity-based control system	14
14 Linear and angular errors, Passivity-based controller. $K_p = [I_3 \ I_3]$, $D_0 = I_3$, $K_0 = \sigma D_0 = I_3$	15
15 Linear and angular errors, Passivity-based controller. $K_p = [20I_3 \ 10I_3]$, $D_0 = I_3$, $K_0 = \sigma D_0 = 20I_3$	15
16 Linear and angular errors, Hierarchical controller. $K_p = 100 [I_3 \ I_3]$, $K_e = 100 [I_3 \ I_3]$	16
17 Linear and angular errors, Hierarchical controller. $I_b \times 5$, $K_p = 10 [I_3 \ I_3]$, $K_e = 100 [I_3 \ I_3]$	17
18 Linear and angular errors, Passivity-based controller. $I_b \times 5$, $K_p = 10 [I_3 \ I_3]$, $K_e = 100 [I_3 \ I_3]$	17

19	Linear and angular errors, Passivity-based controller. $I_b \times 50$, $K_p = 10 [I_3 \ I_3]$, $K_e = 100 [I_3 \ I_3]$.	17
20	Hierarchical control system with momentum-based estimator	18
21	Geometric control system with momentum-based estimator	18
22	Passivity-based control system with momentum-based estimator	19
23	$G_i(s)$'s Bode diagram for $r = 1$, $w_c = 1$	19
24	Ground and ceiling effects ratios	20
25	External generalized forces estimate	20
26	Total thrust and torques, Hierarchical estimator-based controller. $K_p = 10 [I_3 \ I_3]$, $K_e = 100 [I_3 \ I_3]$.	20
27	Total thrusts: generated solely by the control input and acting on the drone	21
28	Desired and effective pose, Hierarchical estimator-based controller. $K_p = 10 [I_3 \ I_3]$, $K_e = 100 [I_3 \ I_3]$.	21
29	Linear and angular errors, Hierarchical estimator-based controller. $K_p = 10 [I_3 \ I_3]$, $K_e = 100 [I_3 \ I_3]$.	21
30	Linear and angular errors, Geometric estimator-based controller. $K_p = K_v = 100I_3$, $K_r = 10I_3$, $K_w = I_3$.	23
31	Linear and angular errors, Passivity-and-estimator-based controller. $K_p = [20I_3 \ 10I_3]$, $D_0 = I_3$, $K_0 = \sigma D_0 = 20I_3$.	23



Figure 1: X3D-BL quadrotor.

Abstract

In recent decades, the development of unmanned autonomous vehicles (UAVs) has been of great interest for field and service robotics applications. This project deals with the tracking problem for a quadrotor, supposing to have perfect knowledge of its current absolute pose in the world frame in every time instant. Moreover, this UAV moves in a well-known environment with several obstacles, and in order to realize a more realistic analysis, external disturbances and several physical effects are emulated, too. To counteract these model uncertainties, a momentum-based estimator has been designed and integrated with the controllers mentioned above. In addition, a collision detection mechanism has been implemented. With this premise, the goal of the project is to simulate the flight control of a quadrotor with different approaches, analyze their performance, and relate their advantages and drawbacks.

Chapter 1: Simulation Environment and Elements

Simulation Tools and Environment

All simulations were carried out using MATLAB R2024b and Simulink. The models, control algorithms, and environment dynamics were implemented entirely within the MATLAB/Simulink environment.

X3D-BL Quadrotor

The adopted drone is the miniature UAV developed by the German company *Ascending Technologies*, **X3D-BL**, known for its use in research and development of control systems for quadrotors (Figure 1). This vehicle is characterized by a set of basic parameters, some of which are provided in Table 1¹.

¹S. R. B. dos Santos, C. L. N. Júnior, S. N. G. Júnior, "Design of Attitude and Path Tracking Controllers for Quad-Rotor Robots using Reinforcement Learning", Instituto Tecnológico de Aeronáutica e Royal Military College of Canada.

Symbols	Definitions	Values	Units
m	Mass	0.5	kg
I_{xx}, I_{yy}	Moment of inertia about the x and y axis	0.0023	$\text{kg} \cdot \text{m}^2$
I_{zz}	Moment of inertia about the z axis	0.00509	$\text{kg} \cdot \text{m}^2$
h	Height	100	mm
ρ	Radius of blade propeller	100	mm
l	Distance between the axis of rotation of the rotors and the center of the mass	170	mm
d	Distance between adjacent propeller axis	240	mm
w	Maximum width with propeller	570	mm

Table 1: Physical model parameters of X3D-BL.

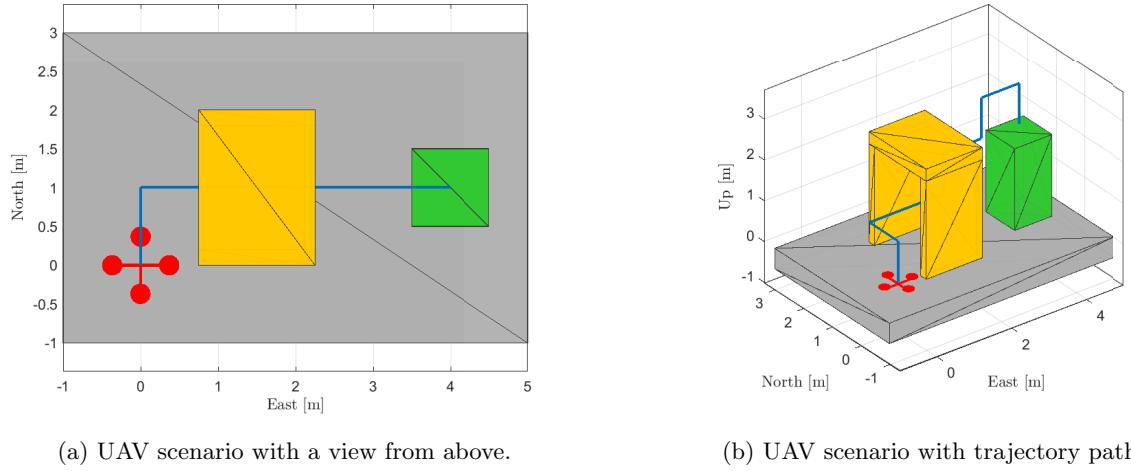


Figure 2: UAV scenario: (a) top view and (b) 3D perspective view scenario with trajectory path.

Environment Scenario

An environment with obstacles (e.g., an urban area) has been designed using the MATLAB *UAV Toolbox* (Figure 2), and both the geometry of all its elements and all the introduced phenomena are supposed to be perfectly known. It includes two obstacles: a **portal frame** composed of two vertical supports and a horizontal beam, centered at (1.5, 1) meters in the world frame; and a **cylindrical column** centered at (4, 1) meters. In addition, a **noisy wind effect** (Figure 3) acting in the xy plane has been emulated: after 5 s from the beginning of future simulations, it arises linearly, and once it reaches a mean value for the applied force of 0.1 N along both of the above specified directions, it stabilizes in it. Whenever the drone is inside the portal frame, the wind intensity along the y direction is supposed to halve, since the robot is covered by one of the two supports. Furthermore, **ground** and **ceiling effects** are included:

- *Ground effect* is a phenomenon that affects a UAV when it flies close to the ground. It consists of an additional upward thrust that tends to further lift up the multirotor: flying near the ground, each propeller generates a downwash that impacts the ground surface and reflects upward, creating an opposing airflow that contributes to the lift. Assuming a co-planar quadrotor parallel to the ground, this effect depends on the radius of its propellers ρ , its flight height z , and on d the distance between its adjacent propellers: the larger the propellers, the greater the minimum distance required for the ground effect to be negligible; and knowing that the airflow from them may interfere with each other, as d increases the mutual interference between propellers becomes negligible. Naming $u_{T,\text{IGE}}$ and $u_{T,\text{OGE}}$

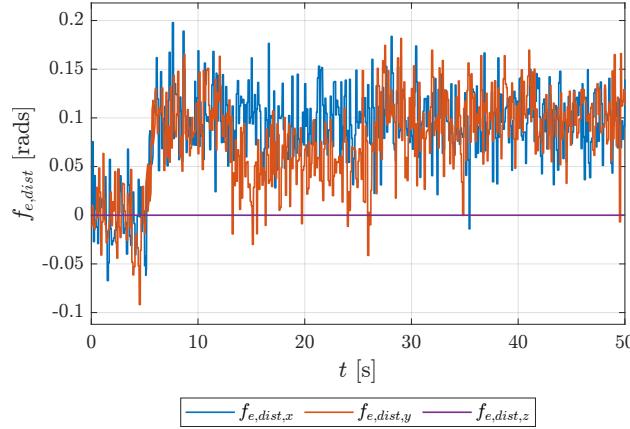


Figure 3: Wind external disturbance

respectively the thrusts of the quadrotor inside and outside the ground effect, the following relation² holds:

$$\frac{u_{T,IGE}}{u_{T,OGE}} = k_{GE} = \frac{1}{1 - \left(\frac{\rho}{4z}\right)^2 - \frac{z\rho^2}{\sqrt{(d^2+4z^2)^3}} - \frac{z\rho^2}{2\sqrt{(2d^2+4z^2)^3}}}$$

- *Ceiling effect* can arise when a UAV needs to physically interact with the environment from below, as in bridge inspection by contact. Suppose a single propeller: this effect depends on the radius of the propeller ρ and the distance to the ceiling z . This phenomenon is due to a vacuum effect, which decreases the propeller drag when it is very close to the obstacle: propellers rotate faster, consequently the thrust increases, and UAV will be pushed even closer to it. Therefore, it is important to consider this effect in order to avoid a possible crash! Instead, assuming a multirotor, it produces a different effect than adding the single effects of each individual rotor, due to the coupling between the flow of the propellers. Naming $u_{T,ICE}$ and $u_{T,OCE}$ respectively the thrusts of the quadrotor inside and outside the ceiling effect and given the possibility of approximating this effect for a quadrotor to that of a single propeller, the following relation³ holds:

$$\frac{u_{T,ICE}}{u_{T,OCE}} = k_{CE} = \frac{1}{2} + \frac{1}{2}\sqrt{1 + \frac{1}{8\hat{z}^2}}$$

where $\hat{z} = z/\rho$.

Trajectory planner

Given the above-mentioned scenario, a suitable desired trajectory $x_d(t) = [p_d(t) \quad \eta_d(t)]^T \in \mathbb{R}^6$, where $p_d(t) \in \mathbb{R}^3$ represents the desired position of the center of mass of the UAV and $\eta_d(t) \in \mathbb{R}^3$ its desired

²A. Ollero, B. Siciliano (Eds.), *Aerial Robotic Manipulation: Research, Development and Applications*, Springer Tracts in Advanced Robotics, Vol. 129, Springer, 2019.

³Rhiannon Elliott-Roe, Kieran Wood, Ozgun Ozer, *Effects of Wind Speed on Quadcopter Ceiling Effect*, 15th Annual International Micro Air Vehicle Conference and Competition (IMAV), University of Manchester, 2024, Paper ID: IMAV2024-37.

orientation described in terms of the Euler angle triplet $\eta = [\phi \ \theta \ \psi]^T$, should have been designed. Actually, quadrotors are differentially flat systems, with (x, y, z, ψ) their flat outputs: given a desired behavior for the UAV, represented by the desired trajectory itself, the control inputs necessary to achieve it are its function. But, as one can see, unfortunately only specifying a desired behavior only for the full position $p_d(t)$ and part of the attitude (specifically the yaw angle $\psi_d(t)$), allows one to take advantage of this property, while the desired roll and pitch angles $\phi_d(t)$ and $\theta_d(t)$ must be derived from $p_d(t)$, $\psi_d(t)$ and their time derivatives. Therefore, only $p_d(t)$ and $\psi_d(t)$ have been designed: $p_d(t)$ has been implemented combining adjacent *position* segments $p_k(t)$ each with a length of 1 meter, while for simplicity $\psi_d(t) = 0$ rad for the whole simulation period. Therefore, the successive analysis focuses on the singular *position* segments composing it. Generally, a trajectory $p(t)$ can be broken down into a **geometric path** $p(s)$ describing its geometry, and a **time law** $s(t)$ along this path, describing the time evolution of a point along the assigned path. To assign an analytic expression for $s(t)$, the fifth-order (quintic) polynomial has been chosen:

$$s(t) = a_5 t^5 + a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0$$

respecting the conditions:

$$s(t_0) = s(0) = 0, \quad \dot{s}(t_0) = 0, \quad \ddot{s}(t_0) = 0, \quad s(t_f) = s(5) = 1, \quad \dot{s}(t_f) = 0, \quad \ddot{s}(t_f) = 0,$$

where $t_f = 5$ s, the time period necessary to complete the desired trajectory segment, has been chosen as the minimum time value to maintain a *hovering* flight state.

Instead, concerning the Geometric point of view, the parametric representation of the rectilinear path, its velocity and acceleration follow:

$$p(s) = p_i + \frac{s}{\|p_f - p_i\|}(p_f - p_i), \quad \dot{p}(s) = \frac{\dot{s}}{\|p_f - p_i\|}(p_f - p_i), \quad \ddot{p}(s) = \frac{\ddot{s}}{\|p_f - p_i\|}(p_f - p_i),$$

where p_i and p_f are, respectively, the starting and ending position points for the specific segment, and $\|p_f - p_i\|$ its length. In the end, ten *position* segments (including the final one, designed to keep the drone hovering at the final desired point) will be properly connected to generate an overall positional trajectory $p_d(t)$ starting from $p_0 = (0, 0, 0.2)$ m with respect to the world frame, avoiding obstacles, and ending at the top of the cylindrical column in $p_F = (4, 1, 2.2)$ m, where it remains hovering for further 5 s, as shown in Figure 2b. The related overall time law $s(t)$ and its time derivatives have been derived (Figure 4).

Collision detector

Supposing that one perfectly knows the pose of the drone and all the obstacles of the environment, it is necessary to implement a method to detect every possible collision. For this reason, a proper algorithm has been deployed: whenever the impact occurs, the simulation stops, and an error message is displayed. In order to be as realistic as possible, not only the contact with the center of mass, but with the entire volume occupied by the drone, has been considered: the height $h = 0.1$ m and the maximum width with propellers $w = 0.57$ m have been taken into account.

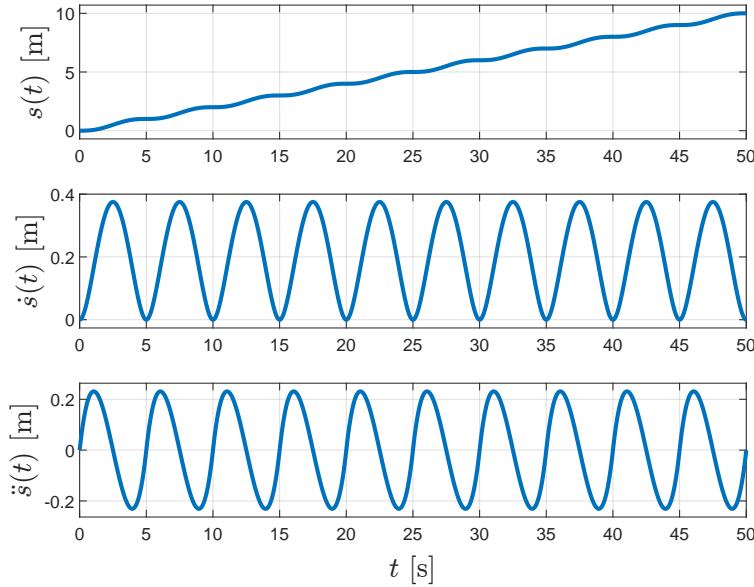


Figure 4: Overall time law and its time derivatives

Chapter 2: Controllers

Several controllers have been implemented: the **Hierarchical**, the **Geometric** and the **Passivity-based** ones⁴. It is worth to mention that, since quadrotor is an underactuated system, these architectures are characterized in each case by two PD+ sub-controllers:

- an Outer (Linear) control loop, responsible for translational control. It has slow dynamics, and provides the remaining attitude references to the Inner loop, and the required total thrust to the UAV.
- an Inner (Angular) control loop, responsible for attitude (orientation) control. This loop has a faster dynamics, and generates the required control torques to the UAV.

As one can notice, these sub-controllers separate the translational and rotational controls of the dynamics of the vehicle. Moreover, the formers are computed in different ways depending on the specific controller. In this chapter, the differences, advantages, drawbacks and the performance of the controllers are reported, without taking into account any external disturbance and/or physical effect.

Hierarchical control

This control technique consists of an outdated approach in modern engineering practices (reference scheme in Figure 5). The Hierarchical controller is built upon the *RPY* attitude representation of the drone's body frame, that is, through the associated Euler angles and their first and second time derivatives. It is worth pointing out, in fact, that the angular error $e_\eta \in \mathbb{R}^3$ is defined in \mathbb{R}^3 and the inner control loop directly gets the desired yaw angle ψ_d from the planner for their computations. This is connected to its suffering from representation singularities, characteristic of the use of a minimal representation (Euler angles) for the angular variables, which complicates control when the system's orientation approaches certain critical angles (in this case $\theta = \pm \frac{\pi}{2}$ rad). For this reason, this controller is not suitable for acrobatic flights, and can be used only in hovering hypothesis. Regarding the robustness, the angular part of the dynamic model is subjected

⁴Inspired by the Field and Service Robotics course.

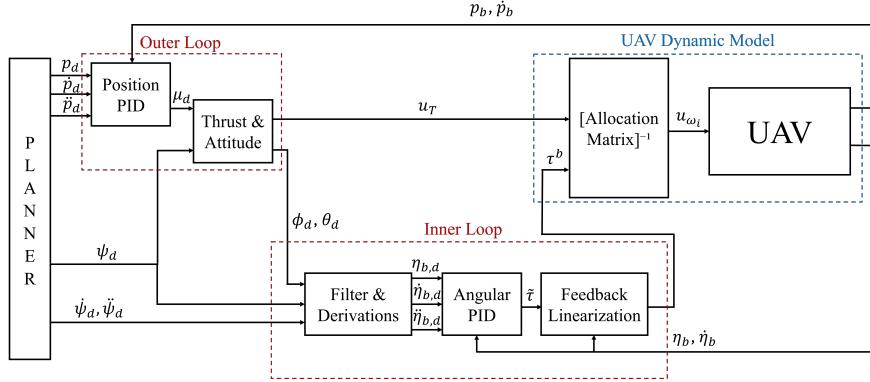


Figure 5: Hierarchical control system

to a *partial feedback linearization* (that is, its linearization based on the attitude feedback), therefore to a cancellation of the dynamics: this procedure makes these control systems lack robustness with respect to model uncertainties and external disturbances. Advantages:

- It is made up of two PID sub-controllers (inner and outer loops): tuning the gains variable, one can deduct easily the effects on control action
- Its stability is proved just employing PD+ sub-controllers (even if in practice, an integral term is required)

Drawbacks:

- It needs the implementation of a "Filtering and Derivation" operation of the angular references. In fact, numerical derivation is necessary to obtain the rates of change of the Euler angles, but being this kind of computation not so accurate, it introduces some noise. The latter must then be filtered, but this ulterior operation degrades the performance of the overall control system.
- It is based on a *partial feedback linearization*, which is a process based on the cancellation of the angular part of the dynamic model through the use of feedback from the drone itself. This characteristic makes the system not robust with respect to model uncertainties. In other words, the controller is highly reliable only under nominal conditions.
- It suffers from representation singularities (avoidable only in hovering hypothesis), that make the entire system not suitable for acrobatic flights.

As mentioned above, limiting the performance analysis to the only controller, and taking into account unitary gain values ($K_p = K_e = [I_3 \ I_3]$), one obtains as shown in Figs. 6 - 8. With these tuning parameters, satisfactory trajectory tracking is achieved for both position ($e_p(t)$ has a maximum order of magnitude of 10^{-2} m) and attitude ($e_\eta(t)$ does not exceed the order of 10^{-3} rad). However, for these gain values, it can be observed that the asymptotic convergence of the orientation error to zero is not achieved: consequently, the position error does not converge either, and remains at most bounded. Moreover, a general trend for the total thrust and torques that the control should apply to the drone model to track this trajectory can be identified. Given the time law assigned to cover the singular trajectory segments from Chapter 1, as the control requires the drone to lift off, the total thrust u_T necessary to achieve such movement increases in the first instants of time in order to push and accelerate the drone itself upwards, and decreases to decelerate it once it starts approaching the segment endpoint, while keeping the control torques null. In order to make the opposite movement, the opposite trend is followed. Instead, to move with the same altitude in the xy plane, it is necessary to provide to the model drone a nearly constant value for the total thrust,

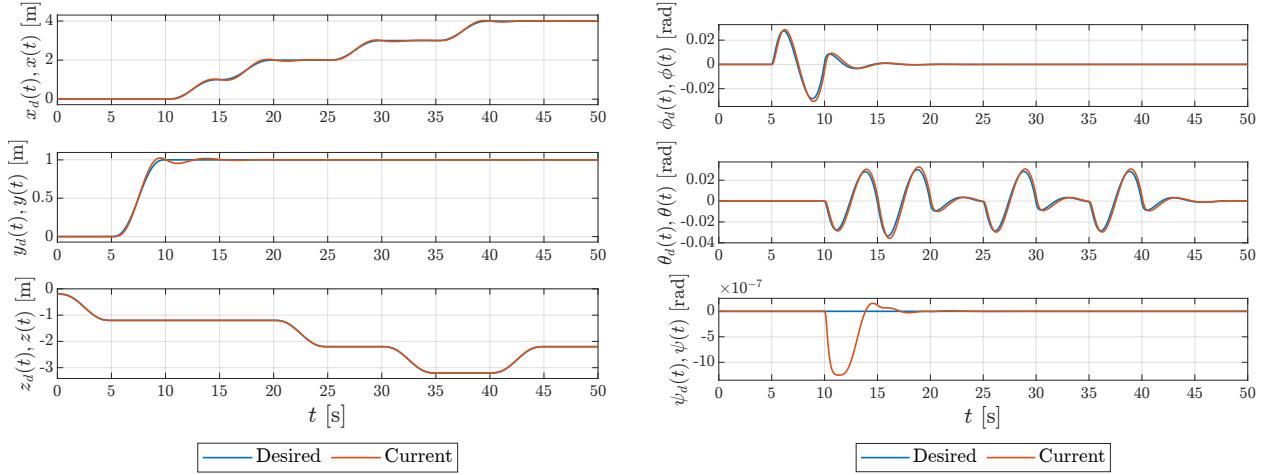


Figure 6: Desired and effective pose, Hierarchical controller. $K_p = K_e = [I_3 \quad I_3]$.

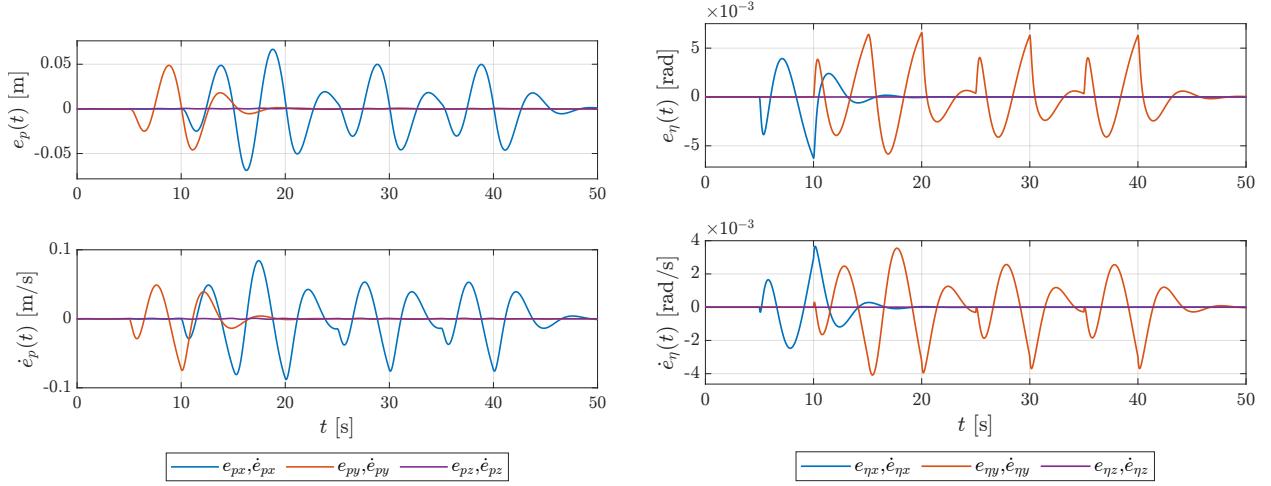


Figure 7: Linear and angular errors, Hierarchical controller. $K_p = K_e = [I_3 \quad I_3]$.

while properly varying control torques: for instance, to move along the y direction towards the East, one should first increase the torque applied around the x axis of the body frame, in order to properly incline the drone and accelerate along the desired direction, and then decrease it to decelerate once approaching the desired segment endpoint. It is worth mentioning that these trends remain similar for all successive chosen controllers and their relative tuned gain parameters, and for this reason, in the remaining part of this chapter, their analysis is omitted.

In order to improve the performance of the control system, after several trials, new values have been assigned to the related gain matrices. First, the values of K_e have been increased ($K_e = 100 [I_3 \quad I_3]$), while leaving K_p unchanged: this way, all of the errors asymptotically converge to zero values! Moreover, to further enhance the performance of the system, the position errors have been made comparable to attitude ones (now both do not exceed the order of 10^{-3}), assigning $K_p = 10 [I_3 \quad I_3]$. The related results are plotted in Figure 9.

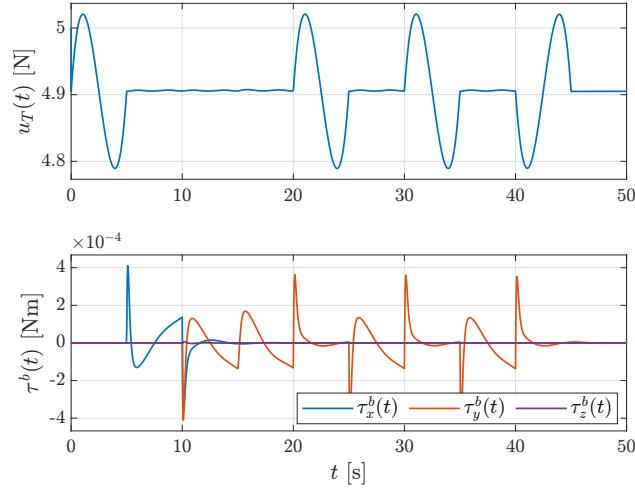


Figure 8: Total thrust and torques, Hierarchical controller. $K_p = K_e = [I_3 \quad I_3]$.

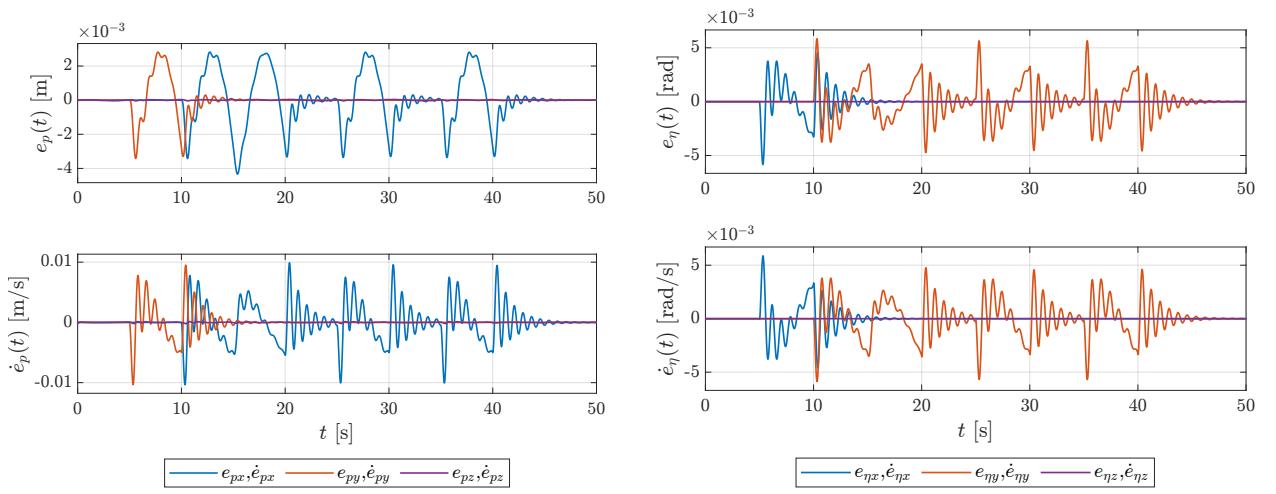


Figure 9: Linear and angular errors, Hierarchical controller. $K_p = 10 [I_3 \quad I_3]$, $K_e = 100 [I_3 \quad I_3]$.

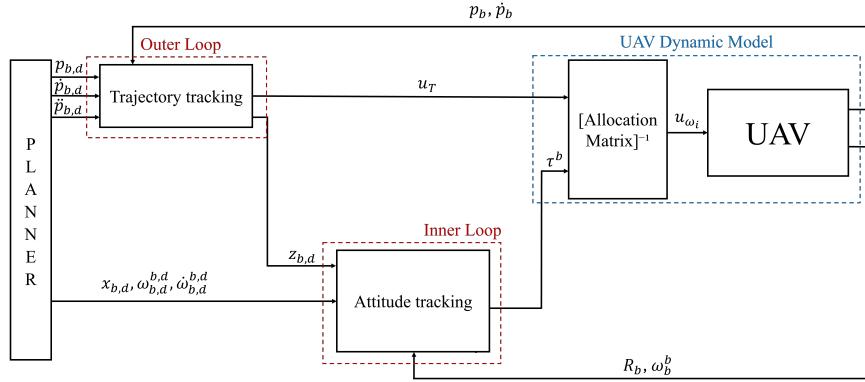


Figure 10: Geometric control system

Geometric control

This control technique consists of an advanced approach aligned with current engineering standards (reference scheme in Figure 10). Differently from the previous case, this controller is based on the *geometric* mathematical model of the drone, characterized by attitude, angular velocities and angular accelerations represented in a geometrical way, that is, through rotation matrices and angular velocities and accelerations vectors. In fact, $e_R \in \mathbb{R}^3$ is defined in $\mathbb{SO}(3)$, and even if the planner provides ψ_d , some processing must be performed to provide the associated vector $x_{b,d}$. For this reason, the Geometric controller is not subject to any singularity. However, this representation has the drawback of being non-minimal. Regarding the robustness, everything said for the Hierarchical controller, is valid in this case, too.

Advantages:

- two linear sub-controllers advantages (already explained)
- the "Filtering and Derivation" operation is not requested, preserving the entire system's performance
- it does not suffer from representation singularities problems, since attitude, angular velocities and angular accelerations are represented in a Geometrical way. For this reason, the control system is suitable for acrobatic flights.

Drawbacks:

- it is based on the *partial feedback linearization* of the dynamic model's angular part, and consequently it is not robust with respect to model uncertainties (already explained)
- exponential stability of the closed-loop system can be proven only if the initial attitude error is less than 90°

N.B. Before proceeding, it is worth clarifying that the controller considered here was implemented based on the one assigned in Homework 3 of the Field and Service Robotics course. However, this time the controller is tasked with tracking a more complex trajectory, and for this reason, its tuning must be revised accordingly. Furthermore, in Chapter 4, the full control system will be integrated with an estimator and subjected to external disturbances.

Its performance for unitary gain values ($K_p = K_v = K_r = K_w = I_3$) and, after several trials, the optimal ones ($K_p = K_v = 100I_3$, $K_r = 10I_3$, $K_w = I_3$), has been analyzed, respectively in Figs. 11, 12.

Focusing on the latter, one can notice how assigning these high values to K_p and K_v drastically decreases e_p and \dot{e}_p , bounded in a range of values even narrower than in the Hierarchical case (for instance, $e_p(t)$ has a maximum order of magnitude of 10^{-4} m). With such values, we will recognize that this controller, supposing

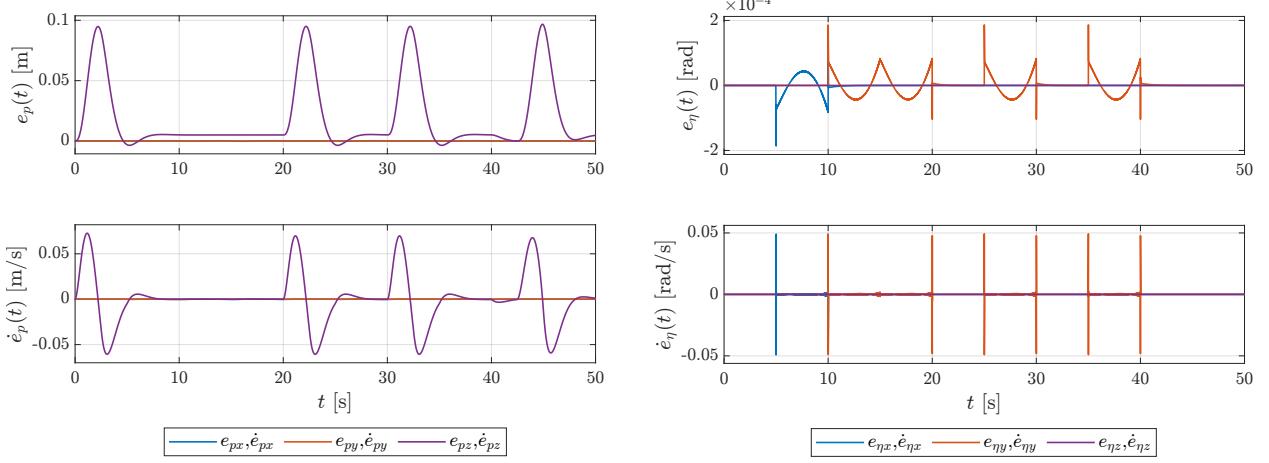


Figure 11: Linear and angular errors, Geometric controller. $K_p = K_v = K_r = K_w = I_3$.

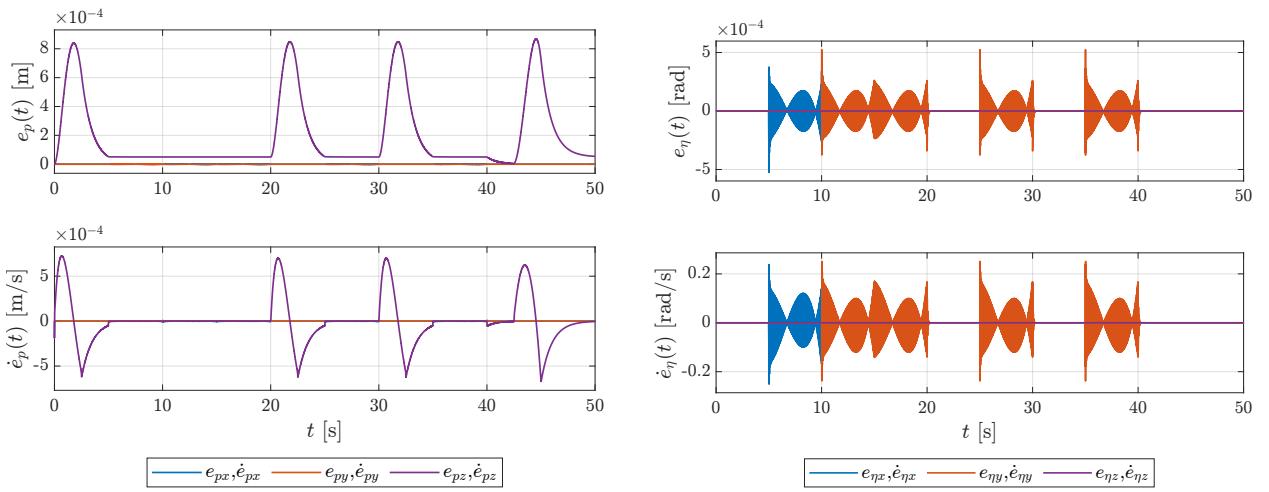


Figure 12: Linear and angular errors, Geometric controller. $K_p = K_v = 100I_3$, $K_r = 10I_3$, $K_w = I_3$.

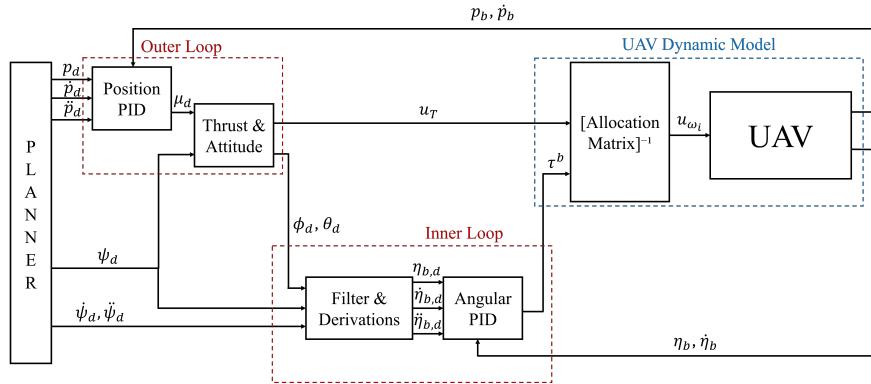


Figure 13: Passivity-based control system

to have perfect knowledge of the dynamic model of the drone, ensures the best performance among all the controllers.

Passivity-based control

This control technique consists of a robust approach, increasingly adopted in advanced engineering applications (reference scheme in Figure 13). As for the Hierarchical case, Passivity-based controller is built upon the *RPY* attitude representation, and for this reason it is subject to representation singularity in the same configurations. Instead, as it concerns for robustness, this controller is not based on a partial feedback linearization of the system, making it more robust than the other two previous approaches! Advantages:

- two linear controllers advantages (already explained)
- it is not based on any kind of feedback linearization: this characteristic makes the control system robust with respect to model uncertainties

Drawbacks:

- the necessity of implementation of a "Filtering and Derivation" operation of the angular references, that degrades its performance (already explained).
- the suffering from representation singularities (avoidable only in hovering hypothesis), that make the entire system not suitable for acrobatic flights (already explained).

Its performance for unitary gain values ($K_p = [I_3 \ I_3]$, $D_0 = I_3$, $K_0 = \sigma D_0 = I_3$) and, after several trials, the optimal ones ($K_p = [20I_3 \ 10I_3]$, $D_0 = I_3$, $K_0 = \sigma D_0 = 20I_3$), has been analyzed, respectively in Figs. 14, 15.

With this tuning, its performances and the Hierarchical ones become comparable.

Impact of "Filter and Derivative" Block on system performance

In the above-seen simulations, employing a second-order low-pass filter inside the "Filter and Derivative" blocks as suggested by the theory would have implied being forced to assign the filters themselves enormous cut-off frequency values (≥ 50 rad/s) to obtain reasonable results for controllers' performance. Therefore,

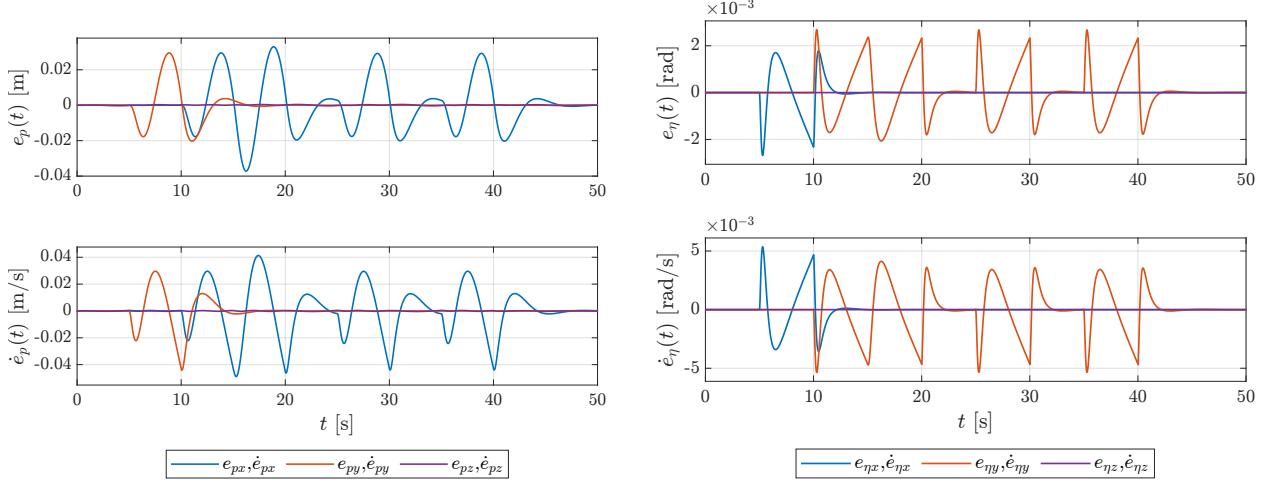


Figure 14: Linear and angular errors, Passivity-based controller. $K_p = [I_3 \quad I_3]$, $D_0 = I_3$, $K_0 = \sigma D_0 = I_3$.

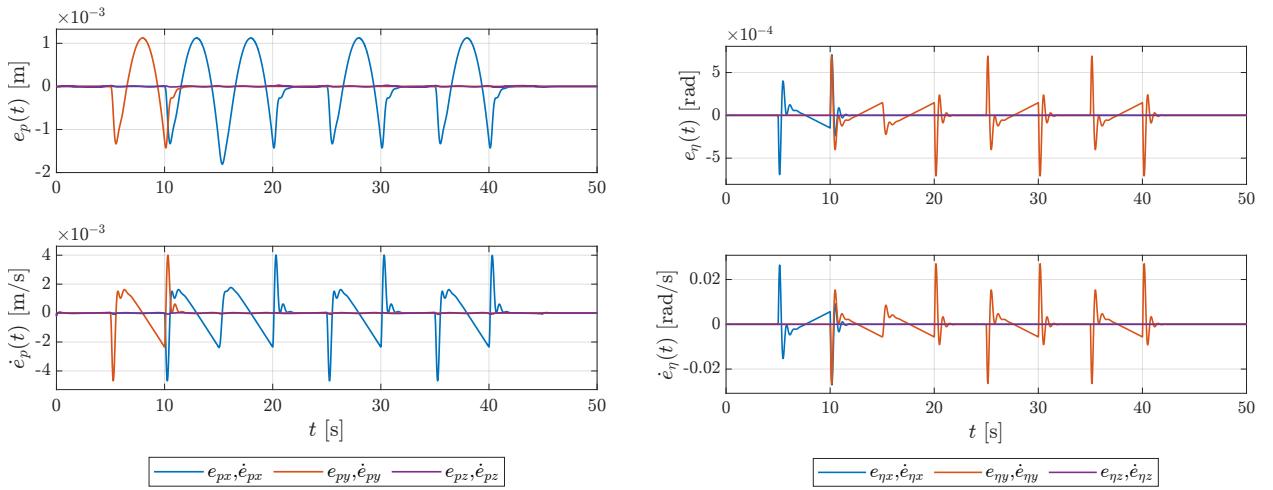


Figure 15: Linear and angular errors, Passivity-based controller. $K_p = [20I_3 \quad 10I_3]$, $D_0 = I_3$, $K_0 = \sigma D_0 = 20I_3$.

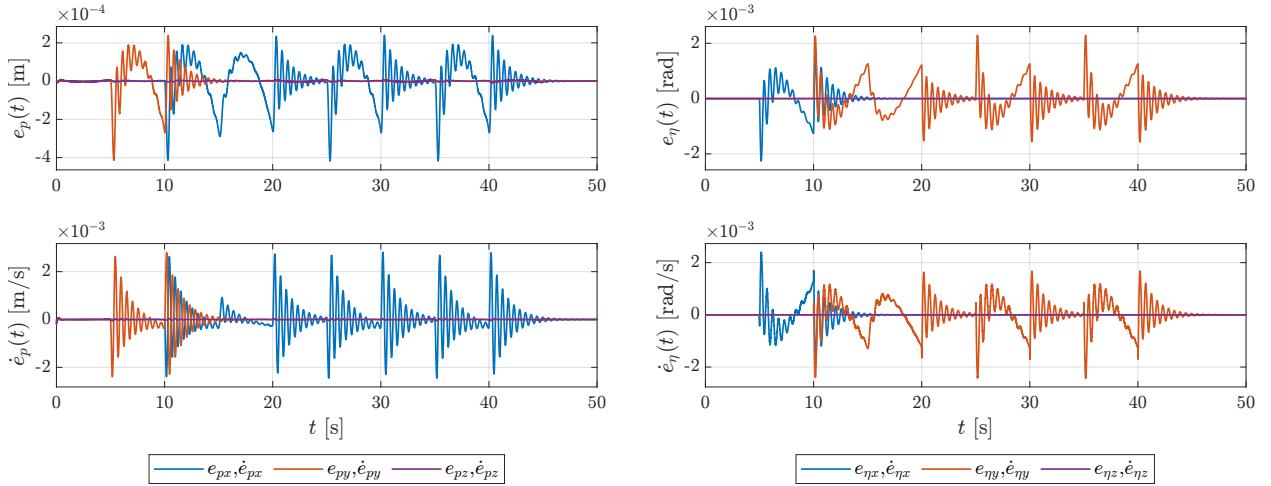


Figure 16: Linear and angular errors, Hierarchical controller. $K_p = 100 [I_3 \ I_3]$, $K_e = 100 [I_3 \ I_3]$.

first-order filters have been adopted, and the related lowest value of the cut-off frequency that has ensured acceptable performance for the entire control system is $w_c = 10$ rad/s. However, it can be proved that this value degrades the control action itself. In fact, assigning, for instance, $w_c = 30$ rad/s and adopting again a trial-and-error approach to tune the Hierarchical controller, tracking performance has been improved: now, it has been possible to assign $K_p = [100I_3 \ 10I_3]$ (not previously feasible, as values higher than $K_p = 10 [I_3 \ I_3]$ led the control system to instability), $K_e = 100 [I_3 \ I_3]$, and as a consequence the mean value of the tracking linear and angular error slightly lower than the ones characterizing the optimal tuning for the Geometric controller. This can be observed by comparing the orders of magnitude in Figs. 12, 16.

Robustness comparison

Hierarchical and Passivity-based controllers robustness with respect to model uncertainties has been compared by increasing the model's inertia by a factor of five in both cases, while applying the above-found optimal values to their respective gains. Concerning the Hierarchical case, as can be seen in Figure 17, the deviation of the model inertia from the one expected in the "Partial Feedback Linearization" block causes the divergence of the drone from the desired trajectory. Instead, as one can notice in Figure 18, the Passivity-based controller performance remains completely unaffected by this change. Several other tests have been executed with the latter controller: even increasing the inertia by a factor of fifty (Figure 19), in spite of the errors arising as well, it still ensures satisfactory trajectory tracking!

Chapter 3: Momentum-based Estimator

In this chapter, an estimator of external generalized forces (force and moments) acting on the aerial platform, and based on the momentum of the mechanical system⁵⁶, is proposed for the control of UAV together with the previous control architectures. With this implementation, it is possible to estimate the external wrench acting on the drone (exerted by external disturbances, physical effects, or even caused by model uncertainties), and

⁵Inspired by the Field and Service Robotics course.

⁶F. Ruggiero, J. Cacace, H. Sadeghian, and V. Lippiello, "Impedance Control of VTOL UAVs with a Momentum-based External Generalized Forces Estimator", available at: <http://wpage.unina.it/fabio.ruggiero/Papers/C14.pdf>

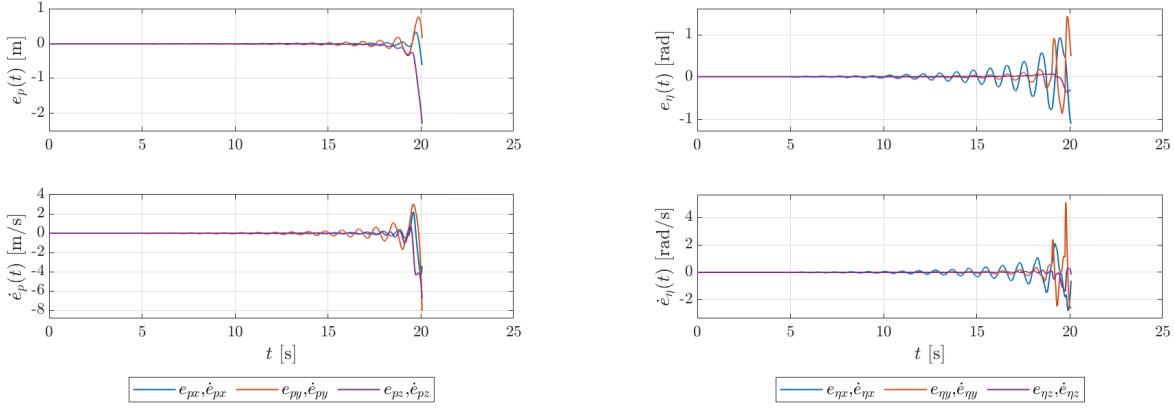


Figure 17: Linear and angular errors, Hierarchical controller. $I_b \times 5$, $K_p = 10 [I_3 \quad I_3]$, $K_e = 100 [I_3 \quad I_3]$.

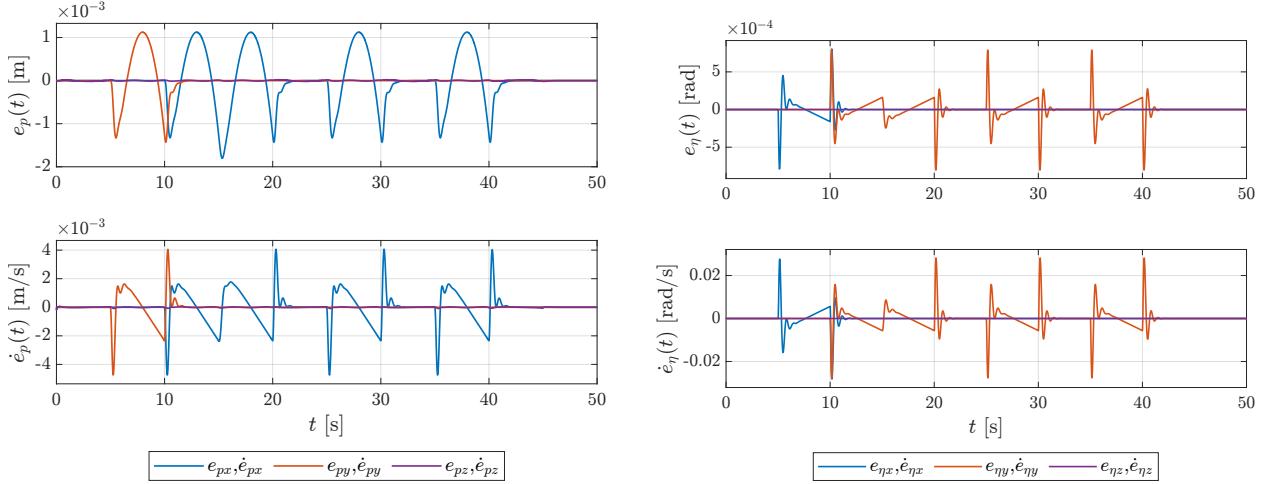


Figure 18: Linear and angular errors, Passivity-based controller. $I_b \times 5$, $K_p = 10 [I_3 \quad I_3]$, $K_e = 100 [I_3 \quad I_3]$.

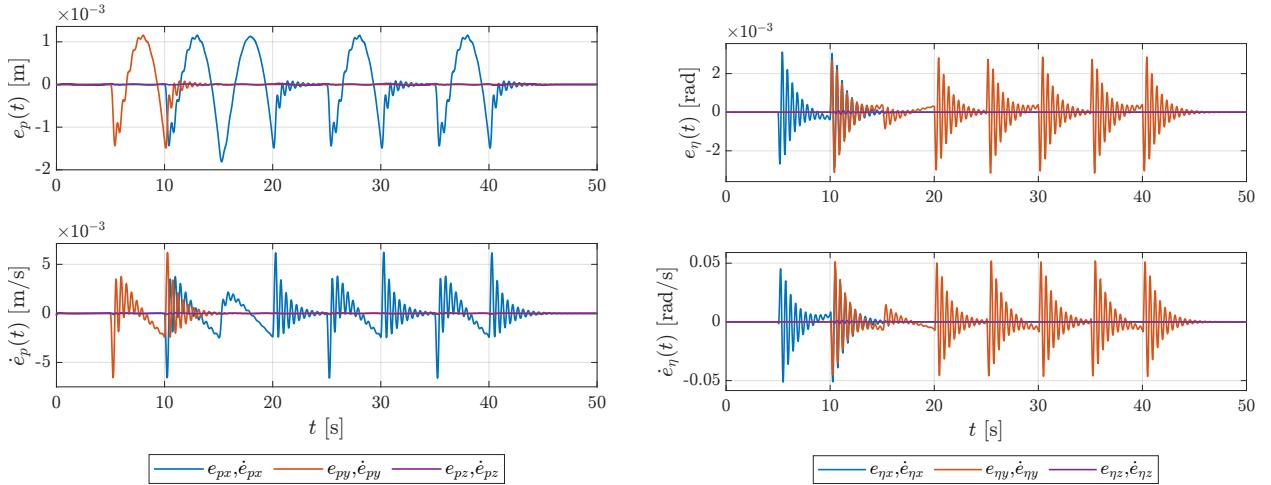


Figure 19: Linear and angular errors, Passivity-based controller. $I_b \times 50$, $K_p = 10 [I_3 \quad I_3]$, $K_e = 100 [I_3 \quad I_3]$.

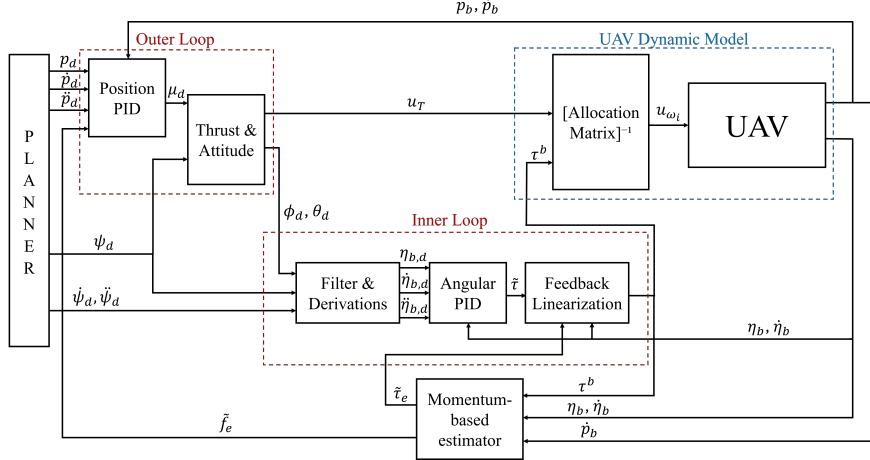


Figure 20: Hierarchical control system with momentum-based estimator

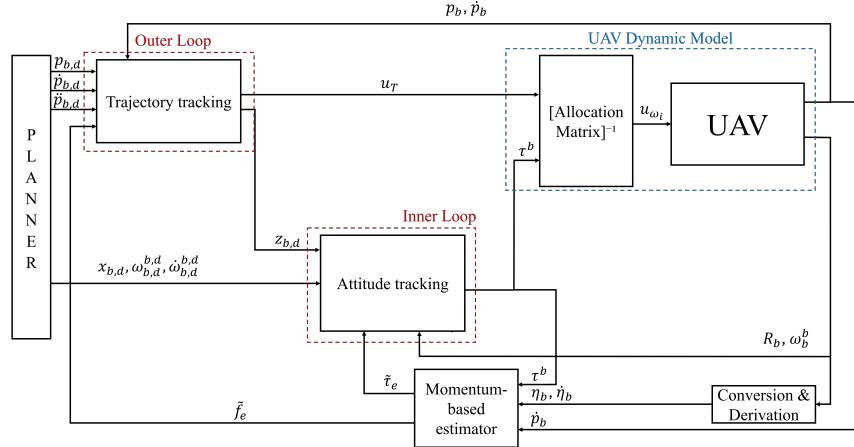


Figure 21: Geometric control system with momentum-based estimator

compensate for them by adding such measures in the PID+ sub-controllers, designate to compute the control action (Figs. 20 - 22).

Its implementation is necessary since there does not exist any sensor that can be equipped on drones capable to measure external disturbances not considered in the model: using an estimator of this kind represents a possible solution. This estimator computed this way realizes a linear relationship between the external wrench and its estimation in the Laplace domain:

$$\mathcal{L} \begin{bmatrix} \hat{f}_e \\ \hat{\tau}_e \end{bmatrix} = G(s) \mathcal{L} \begin{bmatrix} f_e \\ \tau_e \end{bmatrix} = \text{diag}(G_1, G_2, \dots, G_6) \mathcal{L} \begin{bmatrix} f_e \\ \tau_e \end{bmatrix}$$

with $G(s) \in \mathbb{C}^{6 \times 6}$ diagonal matrix of transfer functions $G_i(s)$. The chosen form for the latter is that of a first-order low-pass filter:

$$G_i(s) = \frac{w_c}{s + w_c}, \quad i = 1, \dots, 6$$

where $w_c = 1$ is the chosen cut-off frequency. The related Bode diagrams (Figure 23), illustrate how the estimator operates in reconstructing the behavior of external disturbances, modeled as external forces and torques acting on the system. These estimates exhibit low-frequency components that preserve both the magnitude and phase of the corresponding components of the external wrench. Instead, starting from the

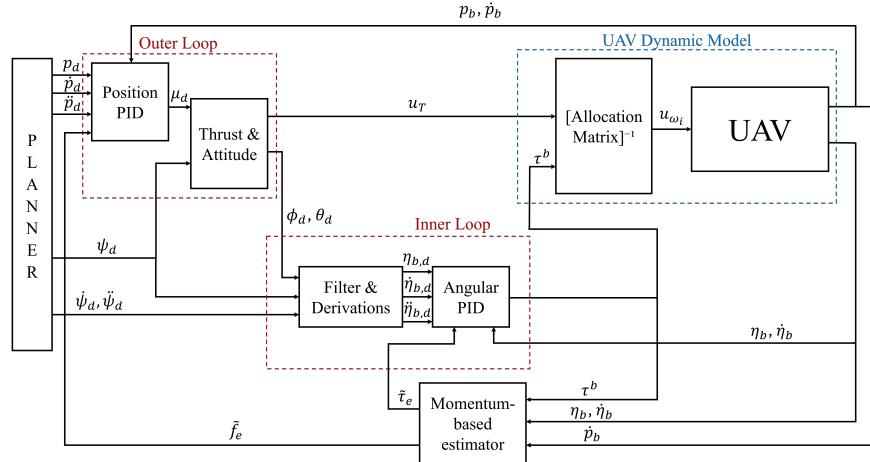
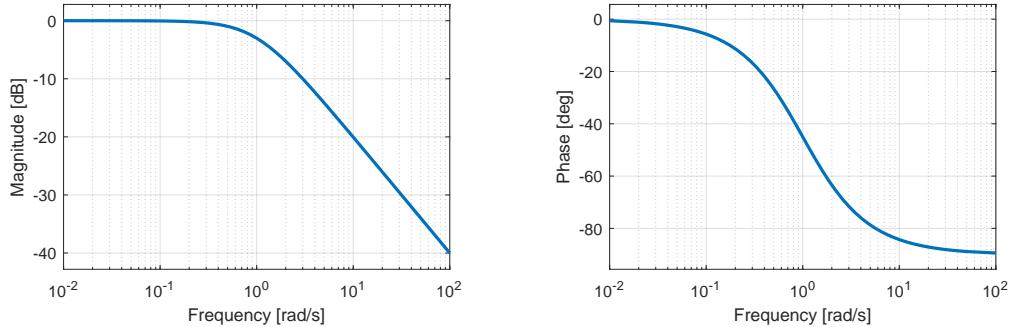


Figure 22: Passivity-based control system with momentum-based estimator


 Figure 23: $G_i(s)$'s Bode diagram for $r = 1$, $w_c = 1$

cutoff frequency $w_c = 1$ and upwards, the higher-frequency components become increasingly attenuated in magnitude (the higher their characteristic frequency, the stronger the attenuation) and experience a phase lag of up to 90 degrees. The graphical results of this estimation will be presented in the following chapter.

Chapter 4: Simulations and considerations

In this final chapter, all the elements introduced in the previous ones are taken into account for the purpose of the simulations.

Hierarchical estimator-based control

Its performance, for the optimal gain values of Chapter 2 ($K_p = 10 [I_3 \ I_3]$, $K_e = 100 [I_3 \ I_3]$), has been analyzed in Figs. 24 - 29.

First of all, let us analyze from Figure 24 the consequences of the ground and ceiling effects on the UAV on the characteristic quantities of the drone. Starting from the ground effect, the ratio k_{GE} (see Chapter 1) has an initial peak since the drone is suppose to start from an off state (propellers plane placed at a height of 0.2 m with respect to the world frame). After this, it drops to nearly unitary values while reaching a height of 1.2 m since it moves away from the ground, and doing the same until a height of 2.2 m its value further

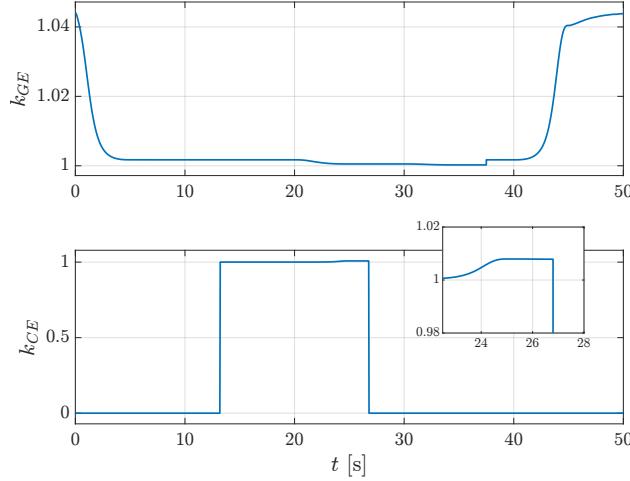


Figure 24: Ground and ceiling effects ratios

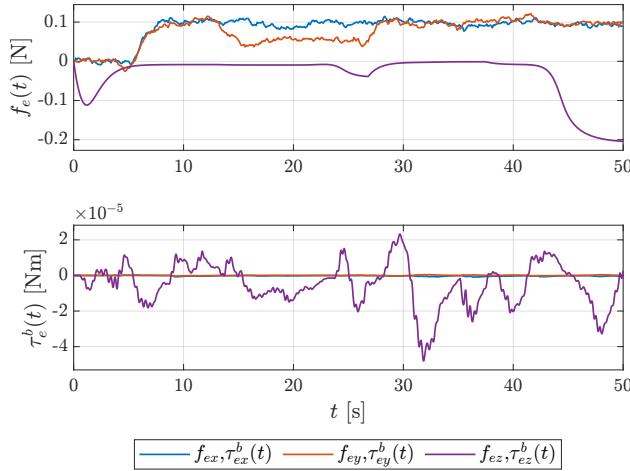
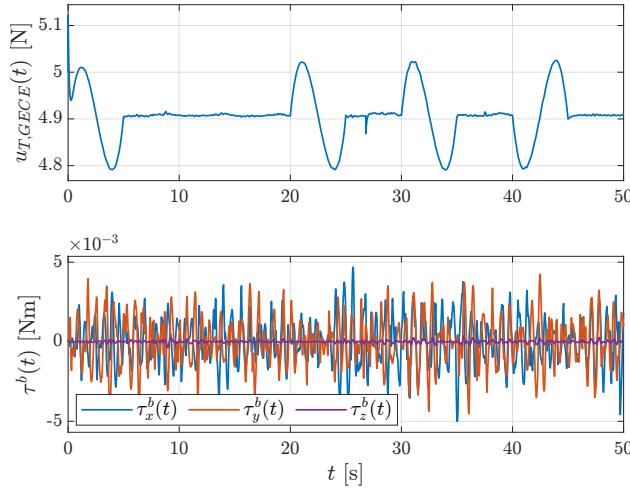


Figure 25: External generalized forces estimate

Figure 26: Total thrust and torques, Hierarchical estimator-based controller. $K_p = 10 [I_3 \ I_3]$, $K_e = 100 [I_3 \ I_3]$.

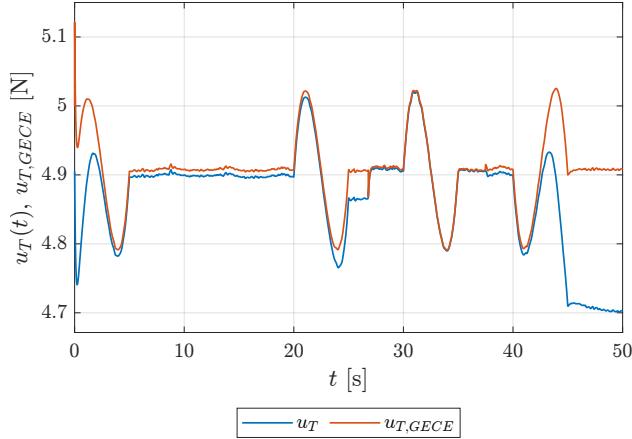


Figure 27: Total thrusts: generated solely by the control input and acting on the drone

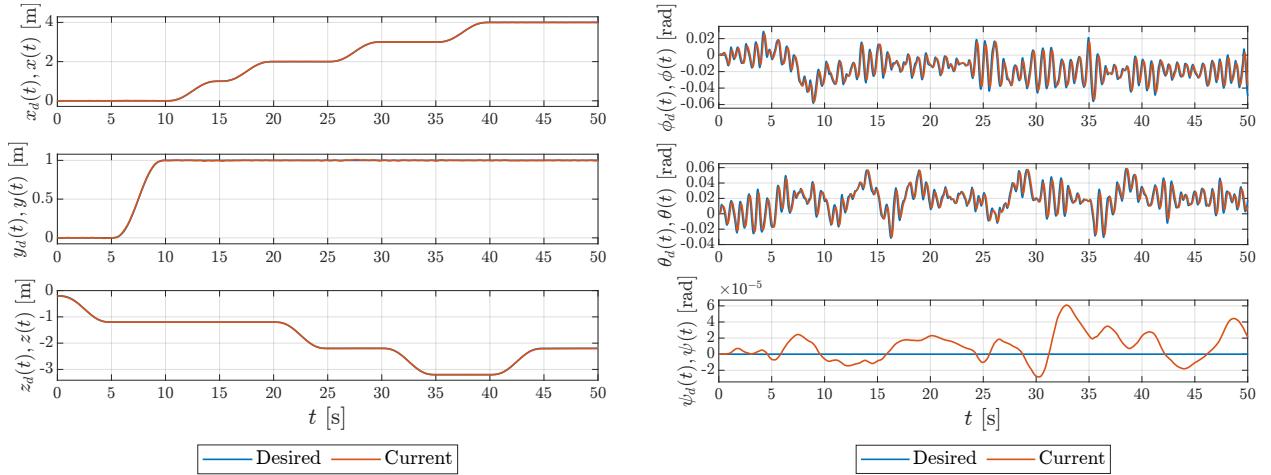


Figure 28: Desired and effective pose, Hierarchical estimator-based controller. $K_p = 10 [I_3 \ I_3]$, $K_e = 100 [I_3 \ I_3]$.

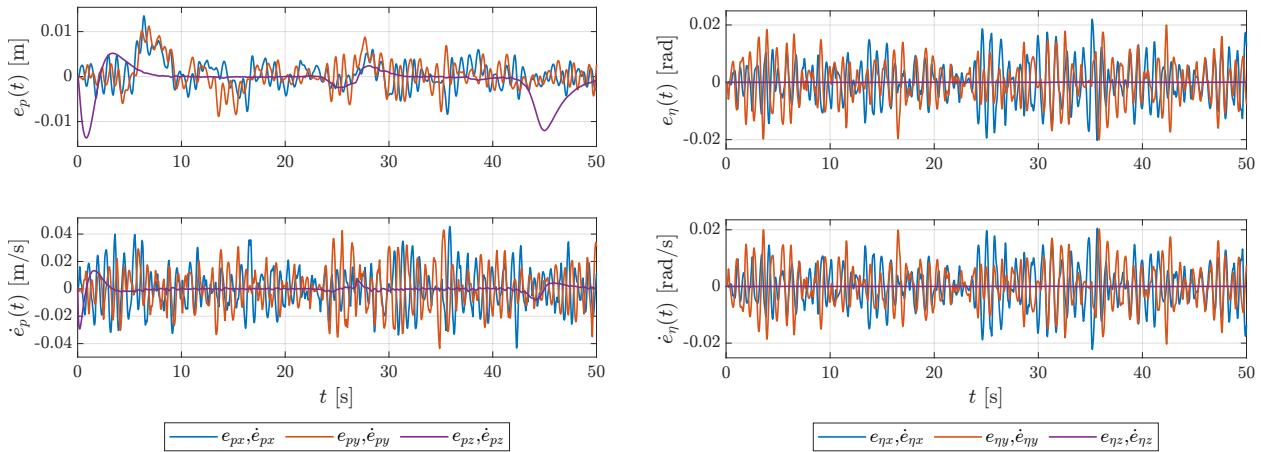


Figure 29: Linear and angular errors, Hierarchical estimator-based controller. $K_p = 10 [I_3 \ I_3]$, $K_e = 100 [I_3 \ I_3]$.

reduces. Once approaching from above the cylindrical column, due to proximity to the latter upper base, k_{GE} again increases until, during landing on the column itself, the nearly same starting value is reached. Concerning the ceiling effect, k_{CE} is supposed to be non null until the UAV remains inside the portal frame (and consequentially under its ceiling), where it assumes the already mentioned trend⁷. As can be seen in the zoomed-in frame, when the UAV reaches a height of 2.2 m and approaches the ceiling from below, k_{CE} increases noticeably. Moreover, it is worth noticing that the control action opposes to such an effect, never letting the drone impact the obstacle.

After that, Figure 25 shows the external wrench estimate computed with the estimator. Comparing these trends with the information from Figs. 3, 24, one can verify that wind (acting in the xy plane), ground, and ceiling effect (both acting along the z direction) are all well estimated.

Figure 26 shows the total thrust and torques applied to the UAV model to reach the desired trajectories: ground and ceiling effects mainly affect the total thrust trend, and this can be noticed above all when both vary abruptly ($t_1 \approx 0$ s, $t_2 \approx 27$ s, $t_3 \approx 38$ s).

Whenever ground and ceiling effects act on the drone, they can also bring benefits to UAV applications: it makes possible to develop more thrust for the same power if the multirotor is subjected to these. Similarly, to reach the same height, a *lesser* value of the control thrust u_T is required! According to this, Figure 27 shows the difference between the total thrust generated exclusively by the control input u_T and the total thrust applied on the UAV model $u_{T,GECE}$. According to what said above, the former is always less than the latter, since an ulterior thrust generated by these phenomena always exists and provides a positive contribution, above all when the ground effect mostly increases.

Figs. 28, 29 show the trends of the tracking parameters: the compensation of the external wrench made possible by the integration of the chosen estimator with the control system tuned with the optimal gains values set in Chapter 2, allows one to perform a satisfactory trajectory tracking in spite of the external disturbances acting on the UAV (e_p , \dot{e}_p , e_η , \dot{e}_η all have a maximum order of magnitude of 10^{-2}). The attention is particularly focused on the positional part of Figure 29: at the exact instant when a given disturbance affects the model, the controller is not fully capable of compensating for it, due to the latency of the estimator in computing the disturbance estimate. This can be observed at the following time instants:

- $t_1 \approx 0$ s: ground effect (along the z -axis) suddenly increases,
- $t_2 \approx 7.5$ s: wind effect (in the xy plane) appears,
- $t_3 \approx 15$ s: wind effect along the y -axis is halved due to the passage through the obstacle,
- $t_4 \approx 25$ s: ceiling effect (along the z -axis) reaches its peak,
- $t_5 \approx 28$ s: wind effect along the y -axis returns to full intensity as the robot exits the obstacle,
- $t_6 \approx 45$ s: the robot starts landing on the cylindrical column, and the ground effect begins to rise again.

Geometric estimator-based control

Its performance, for the optimal gain values from Chapter 2 ($K_p = K_v = 100I_3$, $K_r = 10I_3$, $K_w = I_3$), has been analyzed in Figure 30.

⁷Rhiannon Elliott-Roe, Kieran Wood, Ozgun Ozer, *Effects of Wind Speed on Quadcopter Ceiling Effect*, 15th Annual International Micro Air Vehicle Conference and Competition (IMAV), University of Manchester, 2024, Paper ID: IMAV2024-37.

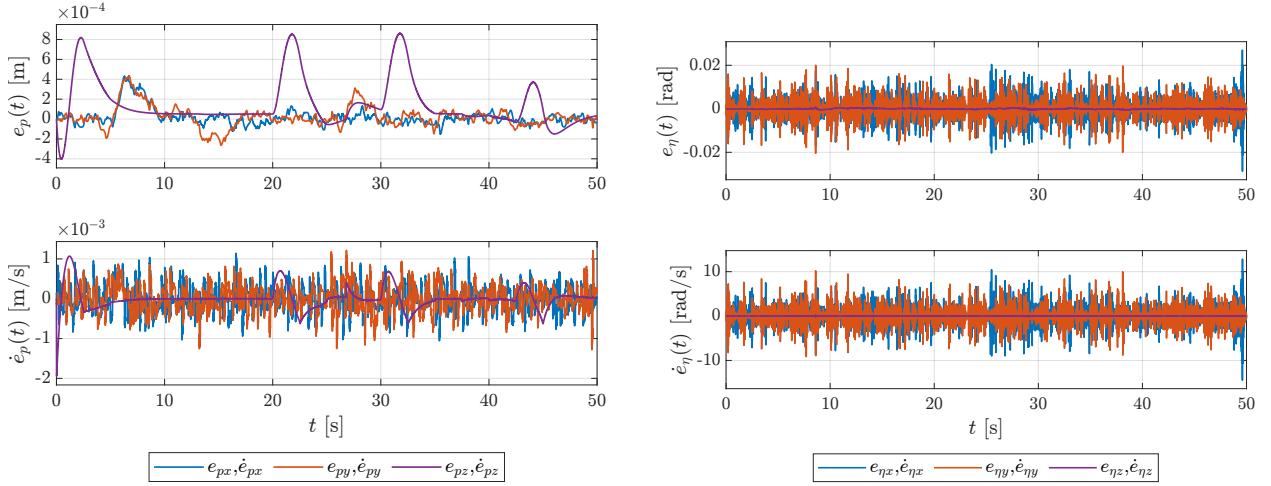


Figure 30: Linear and angular errors, Geometric estimator-based controller. $K_p = K_v = 100I_3$, $K_r = 10I_3$, $K_w = I_3$.

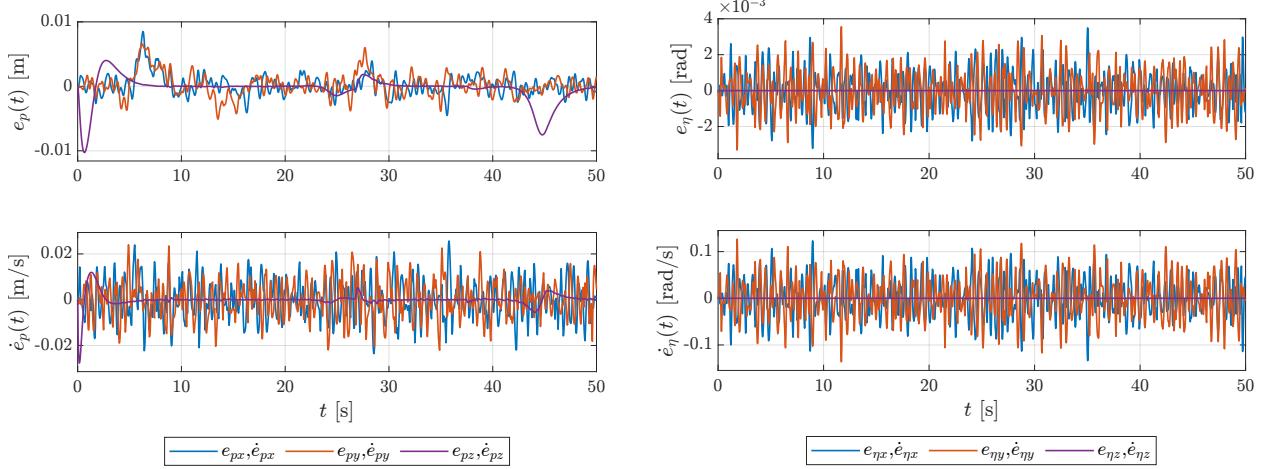


Figure 31: Linear and angular errors, Passivity-and-estimator-based controller. $K_p = [20I_3 \quad 10I_3]$, $D_0 = I_3$, $K_0 = \sigma D_0 = 20I_3$.

Passivity-and-estimator-based control

Its performance, for the optimal gain values from Chapter 2 ($K_p = [20I_3 \quad 10I_3]$, $D_0 = I_3$, $K_0 = \sigma D_0 = 20I_3$), has been analyzed in Figure 31: nearly comparable to the Hierarchical controller!

Chapter 5: Conclusions

A comparison has been conducted among several controllers for a quadrotor, both in an ideal, disturbance-free environment and in a realistic, challenging scenario, under the assumption that the system remains in a hovering flight state. The results indicate that — assuming a perfectly known model — the Geometric controller delivers the best overall performance in the ideal case (Figs. 9, 12, 15). In contrast, in the realistic scenario, the Geometric controller provides the best performance for the positional control, while

the Passivity-based controller excels in the angular control (Figs. 29, 30, 31). On the other hand, when model uncertainty is taken into account, both theoretical considerations and the analysis in the ideal environment (Figs. 18, 19) clearly indicate that the Passivity-based controller consistently achieves superior performance.

Author Contributions

- **Riccardo Stucovitz:** worked on the trajectory construction based on the concepts of geometric path and time law, the implementation of nominal control systems and their tuning, and the modeling of wind disturbance.
- **Nicola Monetti:** worked on the construction of the simulation environment using MATLAB's UAV Toolbox, the implementation and integration of the control systems with the estimator, the modeling of ground and ceiling effects, and the design of the collision detector.
- Both authors actively contributed to reviewing and discussing each other's work, ensuring consistency and alignment with the shared objectives of the project.