



UNIVERSITÀ DEGLI STUDI DI
NAPOLI FEDERICO II

Scuola Politecnica e delle Scienze di Base
Corso di Laurea Magistrale in Ingegneria dell'Automazione e Robotica

Field and Service Robotics

Final Project

Academic Year 2024/2025

Relatore

Ch.mo prof. Fabio Ruggiero

Candidato

Emmanuel Patellaro

P38000239

Repository

https://github.com/EmmanuelPat6/Field_and_Service_Robotics_Final_Project.git

Modeling and Control of UAV Quadrotor using
Hierarchical Control, Geometric Control and
Passivity-Based Control with External Wrench
Disturbances and APF Algorithm considering also
some Aerodynamic Effects

Abstract

This project was developed as the final assignment for the Field and Service Robotics course held by Professor Fabio Ruggiero.

The objective was to implement three robust controllers under the action of external wrench disturbances. Furthermore, an Artificial Potential Fields (APF) Algorithm was implemented in order to avoid obstacles in an unknown environment.

This topic was not covered in the course, but it was nonetheless implemented and explored in order to provide the control algorithm with a more general trajectory, allowing it to be executed within a generic and complex environment.

The controllers used for this project were the Geometric Control (provided in Homework 3) and the two newly controllers: the Hierarchical Control and the Passivity-Based Control.

Simulations showed good performance in terms of Trajectory Planning, Robustness with respect the external disturbances and Collision Avoidance. Tests were conducted with all of the three control strategies, showing differences, strengths and weakness.

Then, an Hovering task with external wrench disturbances has been implemented in order to highlight the successful behavior of the drone, which could be extremely useful for various reasons.

At the end, attention was given to some Aerodynamic Effects, in particular Ground and Ceiling Effects.

The topic was chosen out of personal interest in Aerial Robotics and as a conceptual complement to a previous optimization project focused on Drone-Assisted Parcel Delivery for the course 'MODELLI E METODI DELLA RICERCA OPERATIVA'.

Obviously, the two works are not at all related from an implementation point of view. Nevertheless, for the sake of completeness, this is the GitHub repository of the aforementioned project:

(https://github.com/EmmanuelPat6/The_Parallel_Drone_Scheduling_TSP.git).

The main focus of this project is precisely to highlight the key differences between the three control strategies and how they respond to various external disturbances, as well as their effectiveness in trajectory tracking and robustness with respect to collision avoidance and some other tasks as, for example, hovering.

Several plots were produced to analyze tracking accuracy and dynamic performance.

Contents

UAV and Dynamic Model	2
Coordinate-Free Dynamic Model	3
RPY Dynamic Model	3
Control and Estimator	5
Hierarchical Control	5
Geometric Control	6
Passivity-Based Control	7
Estimator	8
Artificial Potential Field	9
General Algorithm	9
Tasks ans Simulations	11
Environment	11
City	11
Hierarchical Control	13
Geometric Control	14
Passivity-Based Control	15
Three Obstacles and Disturbances	16
Hierarchical Control	16
Geometric Control	17
Passivity-Based Control	18
No Estimator	19
Local Minima Problem	20
Optimized APF Algorithm	21
Hovering	21
Ground Effect and Ceiling Effect	24
Conclusion and Future Works	25
Video References	26
Bibliography	27

List of Figures

1	APF	9
2	Block Scheme with APF by [2]	10
3	UAV Trajectory City	12
4	UAV Trajectory City Top View	12
5	Position and Velocity Desired Values City	12
6	Attractive and Repulsive Forces City	12
7	η and $\dot{\eta}$ Desired Values City Hierarchical Control	13
8	Position and Velocity Errors City Hierarchical Control	13
9	η and $\dot{\eta}$ Errors City Hierarchical Control	13
10	Hierarchical Inputs City	13
11	Position and Velocity Errors City Geometric Control	14
12	Geometric Control Inputs City	14
13	η and $\dot{\eta}$ Desired Values City Passivity-Based Control	15
14	Position and Velocity Errors City Passivity-Based Control	15
15	η and $\dot{\eta}$ Errors City Passivity-Based Control	15
16	Passivity-Based Inputs City	15
17	UAV Trajectory 3 Obstacles and External Wrench Disturbances	16
18	η and $\dot{\eta}$ Desired Values 3 Obstacles and External Wrench Disturbances Hierarchical Control	16
19	Position and Velocity Errors 3 Obstacles Hierarchical Control and External Wrench Disturbances	17
20	η and $\dot{\eta}$ Errors 3 Obstacles Hierarchical Control and External Wrench Disturbances	17
21	Hierarchical Control Inputs 3 Obstacles and External Wrench Disturbances	17
22	Position and Velocity Errors 3 Obstacles Geometric Control and External Wrench Disturbances	17
23	Geometric Control Inputs 3 Obstacles and External Wrench Disturbances	18
24	η and $\dot{\eta}$ Desired Values 3 Obstacles and External Wrench Disturbances	18
25	Position and Velocity Errors 3 Obstacles Passivity-Based Control and External Wrench Disturbances	18
26	η and $\dot{\eta}$ Errors 3 Obstacles Passivity-Based Control and External Wrench Disturbances	19
27	Passivity-Based Inputs 3 Obstacles and External Wrench Disturbances	19

28	Position and Velocity Errors 3 Obstacles, External Wrench Disturbances and No Estimator. At the top left Geometric Control, at the top right Passivity-Based Control, and at the bottom center Hierarchical Control	20
29	Position and Velocity Errors 3 Obstacles, External Wrench Disturbances and No Estimator, Passivity-Based Control with higher gains	20
30	UAV Trajectory Local Minima	21
31	UAV Trajectory Improved APF Top View	21
32	Wrench Estimation Hovering	22
33	Position and Velocity Errors City Hovering	22
34	u_T and τ_b Hovering	23
35	FEM Analysis	23
36	Ground Effect and Ceiling Effect results. On top left the two effects; on top right the total thrust u_T ; at the bottom the desired z -position of the UAV . .	24

UAV and Dynamic Model

Unmanned Aerial Vehicles, commonly known as *UAVs* or drones, are aircraft systems that operate *without a human pilot onboard*. They are controlled either *remotely* by a human operator or *autonomously* by onboard computers. Originally developed for military applications, UAVs have rapidly expanded into civilian sectors due to advances in technology and decreasing costs.

In recent years, UAVs have become increasingly prominent across various fields, including surveillance, precision agriculture, automated delivery, and search and rescue operations. A key component of autonomous UAV functionality is the ability to navigate safely and efficiently in complex, often unknown and dynamic environments.

The **Dynamic Model** of a UAV can be expressed in different ways, depending on the intended application. In the following, two dynamic models will be considered, which are necessary for the implementation of the two controllers that will be presented later:

- *Geometric Control* → **Coordinate-Free Dynamic Model**
- *Hierarchical Control and Passivity-Based Control* → **RPY Dynamic Model**

As can be easily imagined, the Dynamic Model of a UAV is essentially the *Dynamic Model of a Rigid Body that can fly within the environment*, and therefore the equations used to describe it are quite standard.

To proceed with the discussion, it is necessary to define a convention regarding Reference Frames. In *Aerial Robotics*, it is common to use the **NED Frame** (*North-East-Down Frame*), where the $z-$ Axis points downward, while the $x-$ and $y-$ axes point North and East, respectively.

In the following, both the *Body* and the *World Frames*¹ are in the *NED Configuration*.

¹Since the tasks considered last only a few minutes, the World Frame can be considered Fixed and Inertial

Coordinate-Free Dynamic Model

The Dynamic Model of the Quadcopter can be written as:

$$\begin{cases} m\ddot{p}_b = mge_3 - u_T R_b e_3 + f_e \\ \dot{R}_b = R_b S(\omega_b^b) \\ I_b \dot{\omega}_b^b = -S(\omega_b^b) I_b \omega_b^b + \tau^b + R_b^T \tau_e \end{cases} \quad (1)$$

where $e_3 = [0 \ 0 \ 1]^T$, $I_b \in \mathbb{R}^{3 \times 3}$ is the Inertia Matrix, m is the mass of the UAV, g is the gravity acceleration, $f^b = [f_x \ f_y \ f_z]^T \in \mathbb{R}^3$, $\tau^b = [\tau_x \ \tau_y \ \tau_z]^T \in \mathbb{R}^3$ and f_e and τ_e are the External Wrench Disturbances.

Here, there is a mixed way to express the model:

- *Linear Part into the World Frame*
- *Angular Part into the Body Frame* (in such a way to have the Inertia Matrix I_b diagonal)

The model is also called **Geometric** because *No Minimal Representation* is used here for the *Orientation*, but there is only the *Rotation Matrix* R_b without *Euler Angles* or *Quaternions*.

In this way there are *No Representation Singularities* with this type of model.

RPY Dynamic Model

The Dynamic Model of the UAV can be written also in terms of the **RPY Angels**:

$$\begin{cases} m\ddot{p}_b = mge_3 - u_T R_b e_3 + f_e \\ M(\eta_b) \ddot{\eta}_b = -C(\eta_b, \dot{\eta}_b) \dot{\eta}_b + Q^T(\eta_b) \tau^b + \tau_e \end{cases} \quad (2)$$

where $C(\eta_b, \dot{\eta}_b)$ is the Coriolis-like Matrix², $Q(\eta_b) \in \mathbb{R}^{3 \times 3}$ is the Transformation Matrix³ such that $\omega_b^b = Q(\eta_b) \dot{\eta}_b$, R_b is the Rotation Matrix from the Body Frame to the World Frame, $\eta_b = [\varphi \ \theta \ \psi]^T$ and f_e and τ_e like before.

In this model, it is not necessary to introduce the Angular Acceleration or the Angular Velocity that have a direct Geometric and Physical meaning, but it is necessary to introduce the *rate of change of the Euler Angles (RPY)* and consequently their *acceleration*. Here:

- *Same Linear Part like in the previous model because it is not affected by the considerations mentioned above*

² $C(\eta_b, \dot{\eta}_b) = Q^T(\eta_b) S(Q(\eta_b) \dot{\eta}_b) I_b Q(\eta_b) + Q^T(\eta_b) I_b \dot{Q}(\eta_b) \in \mathbb{R}^{3 \times 3}$

³ $Q = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & c_\theta s_\phi \\ 0 & -s_\phi & c_\theta c_\phi \end{bmatrix}$

- *Different Angular Part due to the presence of the Acceleration of the Euler Angles with No Physical Meaning*

Unlike before, here I may encounter *Singularity Problems* because the *Orientation* is not represented by a *Rotation Matrix* but only by the three *Euler Angles*.

In particular, singularities occur when $\det(Q) = 0 \rightarrow \theta = \pm\frac{\pi}{2}$.

Control and Estimator

The control of the UAV presents significant challenges due to their *Non-Linear Dynamic* and *complex coupling* between Rotational and Translational Motions and due to the fact that it is an *Underactuated System* because of the co-planar propellers, as explicitly highlighted in *Homework 1 (Exercise 3, point c.)* and in *Homework 3 (Exercise 1)*. In this final project, as already highlighted previously, the focus is on three *Advanced Control* approaches: **Hierarchical Control**, **Geometric Control** and **Passivity-Based Control**.

Remember that the Quadrotor is a *differentially flat system*⁴, with the position variables and the Yaw Angle as flat output: (x, y, z, ψ) .

This is a very notable property useful for the planner. In fact, in the case of an Ideal World without model uncertainties, external disturbances, or other unavoidable issues, it would be possible to provide these quantities in feedforward and achieve the desired behavior. This is not possible, and therefore *Feedback Control* is necessary.

Hierarchical Control

Let us consider now the *RPY Quadrotor Dynamic Model* in (2).

The controller's objective is to obtain two linear subsystems for the Linear and for the Angular Parts, coupled by a Non-Linear Interconnection Term.

The **Feedback Linearization** of the *Angular Part*⁵ is given by:

$$\tau^b = I_b Q(\eta_b) \tilde{\tau} + Q(\eta_b)^{-T} C(\eta_b, \dot{\eta}_b) \dot{\eta}_b - Q^{-T} \hat{\tau}_e$$

where $\tilde{\tau} \in \mathbb{R}^3$ is the **Virtual Control Input**⁶ and $\hat{\tau}_e$ the compensation for the external disturbances given by the estimator (next section). **Singularity** occur when $\theta = \pm \frac{\pi}{2}$ (so this type of controller is not good, for example, for *Acrobatic Flight*).

Considering the Errors $e_p = p_b - p_{b,d}$, $\dot{e}_p = \dot{p}_b - \dot{p}_{b,d}$, $e_\eta = \eta_b - \eta_{b,d}$ and $\dot{e}_\eta = \dot{\eta}_b - \dot{\eta}_{b,d}$, and avoiding the theoretical discussion regarding the interconnection $\delta(\eta_{b,d}, e_\eta)$ term

⁴A system is differentially flat if and only if there exists a flat output $y = h(x) \in \mathbb{C}^n$ such that it is possible to express the state and the input as a function of y and its time derivative

⁵The Feedback Linearization is one of the drawbacks of this type of controller

⁶In order to obtain $\ddot{\eta}_b = \tilde{\tau}$

and other properties, it is possible to design at the following controller

$$\begin{cases} \mu_d = -K_p \begin{bmatrix} e_p \\ \dot{e}_p \end{bmatrix} + \ddot{p}_{b,d} \\ \tilde{\tau} = -K_e \begin{bmatrix} e_\eta \\ \dot{e}_\eta \end{bmatrix} + \ddot{\eta}_{b,d} \end{cases}$$

with $K_p \in \mathbb{R}^{3 \times 6}$ and $K_e \in \mathbb{R}^{3 \times 6}$.

To conclude, considering $\bar{\mu} = [\bar{\mu}_x \quad \bar{\mu}_y \quad \bar{\mu}_z]^T = \mu_d - \frac{1}{m}\hat{f}_e$ it is possible to obtain:

$$u_T = m\sqrt{\bar{\mu}_x^2 + \bar{\mu}_y^2 + (\bar{\mu}_x - g)^2}$$

$$\phi_d = \sin^{-1} \left(\frac{m}{u_T} (\bar{\mu}_y \cos(\psi_d) - \bar{\mu}_x \sin(\psi_d)) \right) \quad \theta_d = \tan^{-1} \left(\frac{\bar{\mu}_x \cos(\psi_d) + \bar{\mu}_y \sin(\psi_d)}{\bar{\mu}_z - g} \right)$$

So in the end, the Hierarchical Control consists in the so-called **Partial Feedback Linearization** and so, basically, for the Angular Part it is possible to have any controller you want (PD, PID, ...)⁷. For this, the Angular Part should be faster than the Linear Part

The name '*Hierarchical*' come from the fact that, if it is possible to achieve the perfect tracking for the Angular Part, for sure, it is possible to achieve the perfect tracking for the Translational Part.

An important thing is that μ_z must be different from g and so $|\mu_z| < g$ because, if this were to happen, it would be as if the drone were falling downward with the same acceleration as gravity. In practice, this would result in a complete shutdown of the motors, since the controller would have virtually nothing to do.

The *Feedback* of the *Roll*, *Pitch* and *Yaw Angles* and their time derivative are easy achievable from **IMU** (*Inertia Measurement Unit*). Different thing for the *Position*. For this it is necessary some things like, for example, *GPS*, *Camera* (for indoor task) or other stuffs. Giving the position of the robot inside the environment is the **biggest problem of Field Robotics**.

The parameters used were:

$$K_p = \begin{bmatrix} 10I_3 & 0.7I_3 \end{bmatrix} \quad K_e = \begin{bmatrix} 15I_3 & 15I_3 \end{bmatrix}$$

Geometric Control

Starting from the *Coordinate-Free Quadcopter Dynamic Model* in (1), it is possible to consider controller thanks to which it is possible to have *no more filtering and numerical derivation* and, as explicitly seen before, *no more Representation Singularities*, which

⁷The PD Controller used is only a theoretical one. In practice, Integral Action is necessary

are both problems of the Hierarchical Control, avoiding this last one from the beginning, thanks to the use of the **Rotation Matrices** for the *Angular Part*.

For the discussion of this controller, its implementation, and the most important equations, it is recommended to refer to **Homework 3 Report** (specifically **Exercise 4**) here: https://github.com/EmmanuelPat6/Field_and_Service_Robotics_Homework_3.git

The same gains of the *Homework 3* have been used

$$K_p = \text{diag}([30, 30, 60]) \quad K_v = \text{diag}([4, 4, 20])$$

$$K_R = \text{diag}([40, 40, 80]) \quad K_\omega = \text{diag}([10, 10, 20])$$

Passivity-Based Control

Like in the Hierarchical Control, let us consider now the *RPY Quadrotor Dynamic Model* in (2).

The main objective of this controller is to avoid the *Feedback Linearization of the Angular Part* of the previous controller and so, the aim is not to compensate exactly the model of this part. As highlighted before, it is important to consider the *rate of change* $\dot{\eta}_b$ of the RPY Angles η_b and, substantially, the *rate of change of the rate of change* $\ddot{\eta}_b$. Defining the following reference quantities:

$$\dot{\eta}_r = \dot{\eta}_{b,d} - \sigma e_\eta \quad \ddot{\eta}_r = \ddot{\eta}_{b,d} - \nu \dot{e}_\eta$$

$$e_\eta = \eta_b - \eta_{b,d} \quad \dot{e}_\eta = \dot{\eta}_b - \dot{\eta}_{b,d} \quad v_\eta = \dot{e}_\eta + \sigma e_\eta$$

where σ and ν are two *coupling terms* and e_η is the *RPY Angles Error*, it is possible to define the **Inner (Angular) Loop Passivity-Based Control** in this way:

$$\tau^b = Q^{-T}(M(\eta_b)\ddot{\eta}_r + C(\eta_b, \dot{\eta}_b)\dot{\eta}_r - \hat{\tau}_e - D_0 v_\eta - K_0 e_\eta)$$

with $D_0, K_0 \in \mathbb{R}^{3 \times 3}$ positive definite gain matrices and $\hat{\tau}_e$ the compensation for the external disturbances given by the estimator (next chapter). Instead, defining the errors:

$$e_p = p_b - p_{b,d} \quad \dot{e}_p = \dot{p}_b - \dot{p}_{b,d} \quad \ddot{e}_p = \ddot{p}_b - \ddot{p}_{b,d} \quad \tilde{f} = f_e - \hat{f}_e$$

$$\mu_d = \begin{bmatrix} \mu_x & \mu_y & \mu_z \end{bmatrix}^T = -\frac{1}{m} u_T R_b(\eta_{b,d}) e_3 + g e_3 + \frac{1}{m} \hat{f}_e$$

it is possible to obtain, for the **Outer Loop**, that

$$\ddot{p}_b = \mu_d + \frac{1}{m} u_T \delta + \frac{1}{m} \tilde{f}$$

Forgetting for a while the fact that the system is *Underactuated* and considering μ_d simply as the desired acceleration, it is possible to design it as a simple **PD+ Feed-**

forward Controller⁸

$$\mu_d = -K_d \dot{e}_p - K_p e_p + \ddot{p}_{b,d}$$

Studying the Closed-Loop System⁹, it is clear that *the error will asymptotically converge to 0*. Sometimes, it is possible to prove only the *boundness* of the error, in particular when there is not a perfect estimation of the external disturbance.

Furthermore, there is not an exact compensation for the *Coriolis Term*.

In fact in the equations appears $(C + D_0)$ and so it is possible only to change a bit the Coriolis Matrix effect. In practice, the *Closed-Loop Equations* can be seen as **Mass-Damping-Spring** systems with *programmable stiffness* (\mathbf{K}_0) and *partially programmable damping* ($\mathbf{C} + \mathbf{D}_0$) and given mass and Inertia.

To conclude, considering $\bar{\mu} = [\bar{\mu}_x \quad \bar{\mu}_y \quad \bar{\mu}_z]^T = \mu_d - \frac{1}{m} \hat{f}_e$ it is possible to obtain the same equations like in the *Hierarchical Control*.

The parameters chosen for the implementation are

$$K_p = 1.5I_3 \quad K_D = I_3 \quad \sigma = 25 \quad \nu = 10 \quad D_0 = 15I_3 \quad K_0 = \sigma D_0$$

It is quite important the *tuning* of the parameters in this type of controller.

In fact, the drone can exhibit completely incorrect behavior when these parameters take values that are too large or too small.

σ have to be the largest as possible to avoid *vibration* problem, with, obviously an Upper Bound.

Estimator

Estimators play a crucial role in control systems by providing accurate information about *states* or *disturbances* that are *not directly measurable*. In the context of *Aerial Robotics* estimators are essential for achieving **robust** and **stable flight**, particularly in the presence of *model uncertainties* and *external disturbances*.

In particular, an **External Wrench Disturbance** refers to the combined effect of *external forces* and *torques* acting on the UAV, such as wind gusts, payload shifts, or collisions. These disturbances can significantly affect the *performance* and *stability* of the quadrotor if not properly accounted for.

To address this, a Momentum-Based Estimator has been implemented, both for Geometric and Passivity-Based Controllers. This, by estimating the external wrench, during the flight, can compensate for these disturbances, ensuring better **stability**, **trajectory tracking**, and **safety** in uncertain or dynamic environments.

A **Momentum-Based Estimator** has been used with $r = 2$ and $c_0 = 10$, to make the response dynamically faster.

⁸In the practice there is also an Integral Term

⁹ $m\ddot{e}_p + K_d \dot{e}_p + K_p e_p = u_T \delta \quad , \quad M\dot{v}_\eta + (C + D_0)v_\eta + K_0 e_\eta = 0$

Artificial Potential Field

Autonomous Navigation represents a fundamental challenge in the *Field of Robotics* and *Artificial Intelligence*, particularly in dynamic and partially unknown environments. As already mentioned in the introductory chapter, among the various approaches developed to address this issue, the **Artificial Potential Field (APF)** method has gained considerable attention due to its conceptual *simplicity*, *Real-Time performance*, and *adaptability* to a wide range of scenarios. The *APF Algorithm* models the robot as a *particle* under the influence of *Artificial Forces*, where the goal exerts an **Attractive Force** and obstacles exert **Repulsive Forces**. This method enables efficient path planning by guiding the robot toward the goal while avoiding collisions.

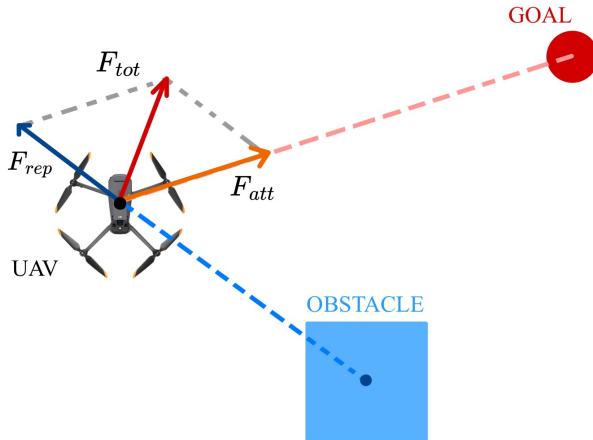


Figure 1: APF

The description provided by Oussama Khatib in [1] outlines the functioning of this algorithm in a very simple and direct manner:

The manipulator moves in a field of forces. The position to be reached is an attractive pole for the end effector and obstacles are repulsive surfaces for the manipulator parts.

General Algorithm

The APF Algorithm has three *inputs*:

- **Goal Position:** where the drone is supposed to go
- **Obstacles Position:** where the obstacles are located in the surrounding environment

- **UAV Position:** where the robot is located at that exact moment

In this project, the *Goal Position* and the *Obstacles Position* are hardcoded in the program, but for a real implementation, the use of various sensors such as *LIDAR*, *IMU*, or *GPS* is necessary, depending on the intended use of the drone.

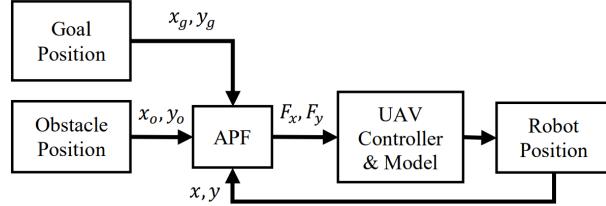


Figure 2: Block Scheme with APF by [2]

Considering $\mathbf{x} = \begin{bmatrix} x & y & z \end{bmatrix}^T$ the robot position vector and \mathbf{x}_d as the goal position vector, the control of the UAV with respect to the obstacle O , can be achieved by subjecting it to the **Artificial Potential Field**

$$U_{tot}(\mathbf{x}) = U_{att}(\mathbf{x}) + U_{rep}(\mathbf{x})$$

Then, using *Lagrange's Equation* it is possible to find the command vector F_{tot} such that

$$F_{tot} = F_{att} + F_{rep}$$

with

$$F_{att} = -\nabla U_{att}(\mathbf{x}) = K_{att}(\mathbf{x}_d - \mathbf{x})$$

$$F_{rep}(\mathbf{x}) = -\nabla U_{rep}(\mathbf{x}) = \begin{cases} K_{rep} \left(\frac{1}{\rho} - \frac{1}{\rho_0} \right) \frac{1}{\rho^2} \frac{\partial \rho}{\partial \mathbf{x}} & \text{if } \rho \leq \rho_0 \\ 0 & \text{if } \rho > \rho_0 \end{cases}$$

where K_{att} is the *Attractive Gain*¹⁰ and K_{rep} is the **Repulsive Gain**. Furthermore

$$\rho = \sqrt{(\mathbf{x} - \mathbf{x}_o)^2} = \sqrt{(x - x_o)^2 + (y - y_o)^2 + (z - z_o)^2}$$

$$\frac{\partial \rho}{\partial \mathbf{x}} = \left[\frac{\partial \rho}{\partial x} \quad \frac{\partial \rho}{\partial y} \quad \frac{\partial \rho}{\partial z} \right]^T = \frac{\mathbf{x} - \mathbf{x}_o}{\rho}$$

where $\mathbf{x}_o = \begin{bmatrix} x_o & y_o & z_o \end{bmatrix}^T$ is the obstacle position vector.

¹⁰It is necessary, for excessively large error values, to normalize the attractive force; otherwise, it would reach excessively high magnitudes

Tasks ans Simulations

Let us implement via **MATLAB** and **Simulink** what is written in the previous chapter

Environment

Regarding the environment and the creation of obstacles, the **UAV Toolbox** was used, specifically the blocks related to the *Scenario*, which allow, through appropriate lines of code, the creation of obstacles and the visualization of the UAV and its operation. Different environments have been created. In particular all the simulation have been done with all the created scenarios but, in the following, they are presented according to this criterion:

- **City-Like Environment** to fully highlight the functioning of the *Algorithm*, the *Trajectory Tracking* and several differences between the three controllers
- **Three Obstacles Environment** to highlight the functioning of the *Estimator* and how each controller react to disturbances with and without estimator
- **Single Obstacle Environment** to show the *Local Minima Problem* and a possible solution to avoid it

All the video are presented in the **Video References** Section in the end of this report.

City

As previously mentioned, to further highlight the *good performance* exhibited by the developed *APF Algorithm*, numerous obstacles and many more waypoints have been added, in order to create a scenario closely resembling a **City**. It is important to choose appropriate gain values. Too high values of K_{att} would tend to cause the UAV to oscillate once it reaches the vicinity of the final goal, as well as make the UAV move very quickly in the preceding phase.

The parameters used were: $K_{att} = 1.25$, $K_{rep} = 0.5$ and $\rho_0 = 2.0$ m, while the waypoints were $\begin{bmatrix} 0 & 20 & -6 \end{bmatrix}^T$, $\begin{bmatrix} 8 & 28 & -8 \end{bmatrix}^T$, $\begin{bmatrix} 22 & 27 & -6 \end{bmatrix}^T$, $\begin{bmatrix} 26 & 20 & -7 \end{bmatrix}^T$, $\begin{bmatrix} 26 & 2 & -9 \end{bmatrix}^T$, $\begin{bmatrix} 12 & 6 & -7 \end{bmatrix}^T$, $\begin{bmatrix} 11 & 15 & -10 \end{bmatrix}^T$, $\begin{bmatrix} 11 & 25 & -6 \end{bmatrix}^T$, $\begin{bmatrix} 17 & 11 & -3 \end{bmatrix}^T$ and $\begin{bmatrix} 4 & 0 & -7 \end{bmatrix}^T$.

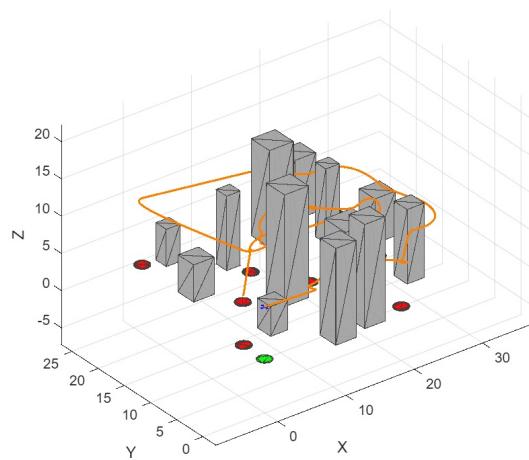


Figure 3: UAV Trajectory City

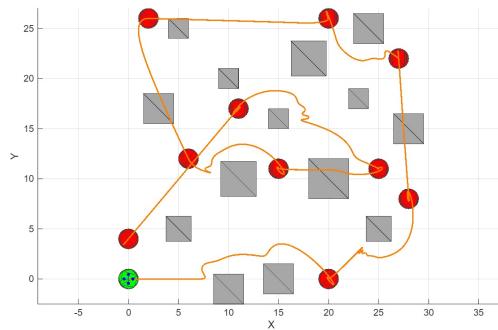


Figure 4: UAV Trajectory City Top View

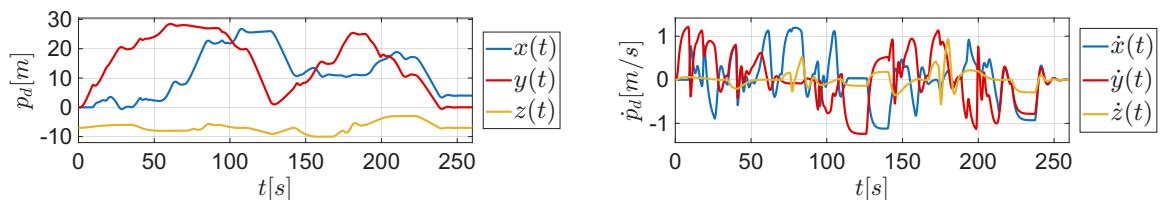


Figure 5: Position and Velocity Desired Values City

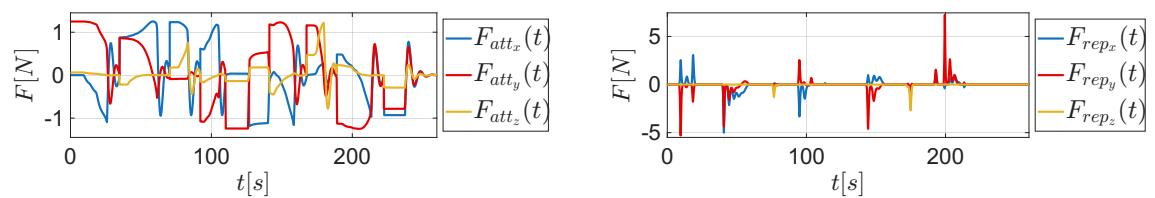


Figure 6: Attractive and Repulsive Forces City

Hierarchical Control

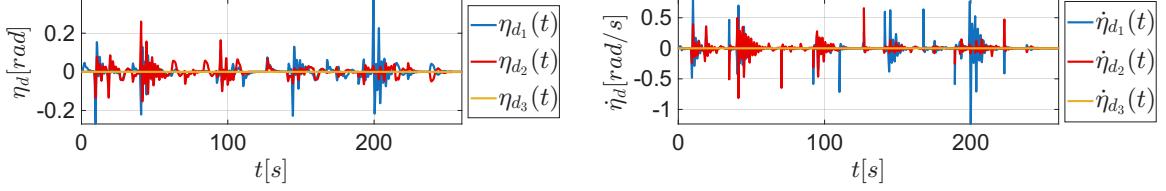


Figure 7: η and $\dot{\eta}$ Desired Values City Hierarchical Control

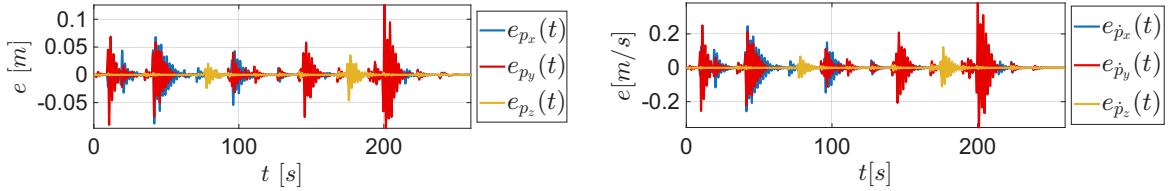


Figure 8: Position and Velocity Errors City Hierarchical Control

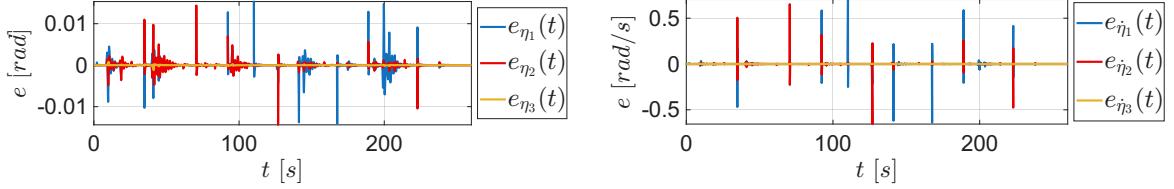


Figure 9: η and $\dot{\eta}$ Errors City Hierarchical Control

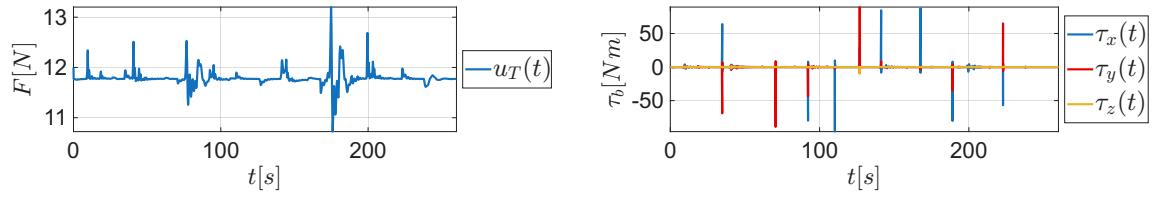


Figure 10: Hierarchical Inputs City

From the following graphs, it is possible to observe that **the obstacles are successfully overcome** with a smooth trajectory, and the final goal is reached in approximately 250 seconds. The errors are on the order of 10^{-3} , which is completely **acceptable** considering, for example, that the position is on the scale of meters while the error is only a few **millimeters**. There are only few peaks in particular in the y direction, where the drone have to *avoid the obstacle* (in this case the largest peaks are when the drone

have to pass between two obstacles that are very close to each other).

In particular, when a peak occurs, the value does not immediately return to a lower level but tends to oscillate gradually.

This *oscillatory behavior* can likely be attributed to the **filtering and numerical derivation** used in this type of controller. Specifically, when a sudden change or peak occurs, this introduces a *delayed and smoothed response*. This results in gradual oscillations since the filtering and derivative actions slow down the system's ability to react quickly.

The forces also reach relatively low values, with a *Maximum Attractive Force* of about 1.2 N and a *Maximum Repulsive Force* of approximately 7 N.

We can also observe peaks in u_T and τ_b at the moment when the UAV has to overcome the obstacles. This is entirely reasonable, as **greater dynamic effort** is required in this situation to avoid the obstacles themselves.

Geometric Control

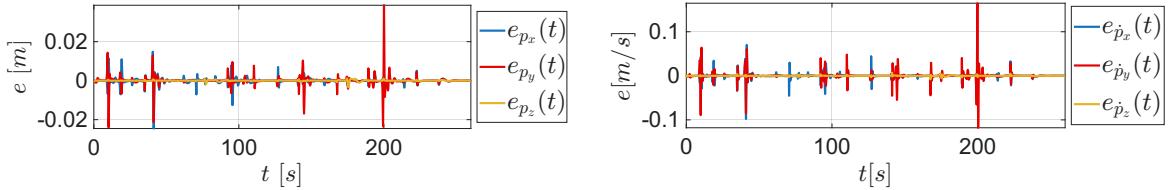


Figure 11: Position and Velocity Errors City Geometric Control

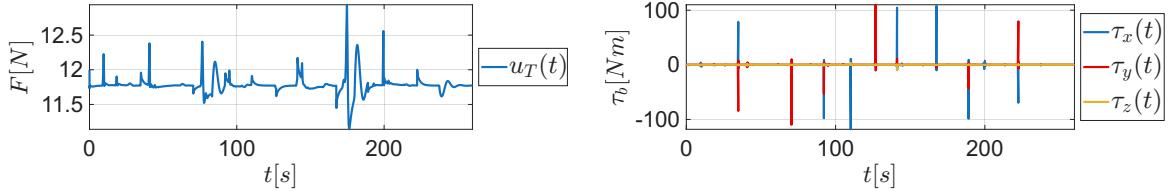


Figure 12: Geometric Control Inputs City

For the *Geometric Control*, with respect the previous one, it is possible to notice some improvements, in particular in terms of errors and oscillations. In fact the position and velocity error are much *smaller* and they have a *less oscillation* behavior. This is due to the fact that the strong point of the Geometric Control, as already seen in the previous chapter, is the fact that this controller avoid the *filtering and derivative* parts of the previous controller.

Furthermore, the control input u_T appears much more '*linear*', especially in the initial phase, compared to the *Hierarchical Control*, while τ_b is practically the same.

Passivity-Based Control

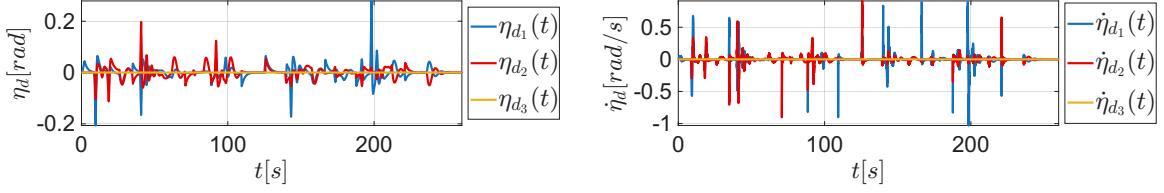


Figure 13: η and $\dot{\eta}$ Desired Values City Passivity-Based Control

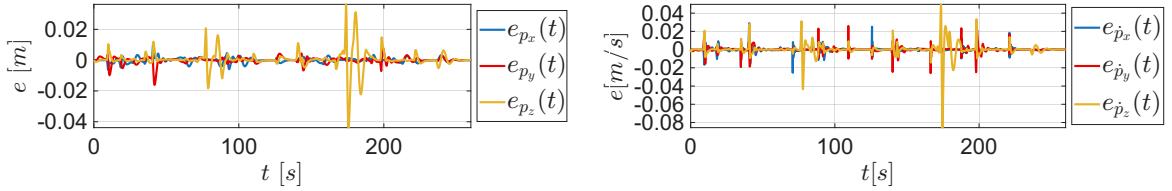


Figure 14: Position and Velocity Errors City Passivity-Based Control

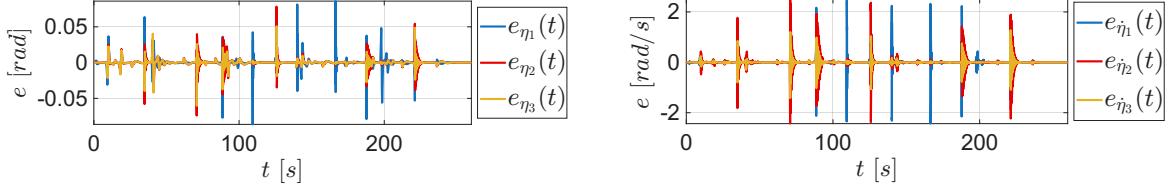


Figure 15: η and $\dot{\eta}$ Errors City Passivity-Based Control

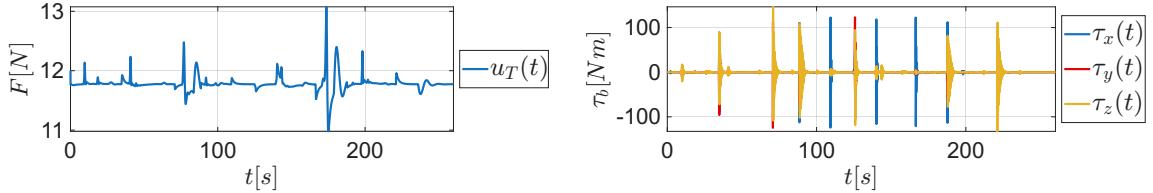


Figure 16: Passivity-Based Inputs City

For the *Passivity-Based Control*, the behavior is entirely comparable to that obtained with *Geometric Control*. With an appropriate choice of the controller parameters, as highlighted in the previous chapter, it is possible to obtain **slightly lower errors**, still on the order of 10^{-3} and with smaller peaks with respect the previous example. Also the velocity errors are significantly lower compared to the previous case. Furthermore, the resulting τ_b and the *Attitude Error* exhibit slightly more **oscillatory behavior**, due to the *Non Linearization of the Angular Part*.

Three Obstacles and Disturbances

Let us now show a slightly more complex implementation, by including, within the *Dynamic Model*, also **External Wrench Disturbances**. In particular, initially, some *constant disturbances* were implemented and then more general ones.

For this example, the results obtained with all the controllers will be shown, all of them with a *2nd-Order Momentum-Based Estimator* showed in the previous chapter.

The parameters used for this simulation were: $K_{att} = 1$, $K_{rep} = 0.5$ and $\rho_0 = 3.0$ m while the waypoints are: $\mathbf{x}_{d_1} = [0 \ 20 \ -7]^T$, $\mathbf{x}_{d_2} = [20 \ 20 \ -7]^T$ and $\mathbf{x}_{d_3} = [20 \ 0 \ -7]^T$. The disturbances applied are: $D_x = D_y = 1$ N, $D_z = -1$ N and $D_{yaw} = -0.8$ Nm.

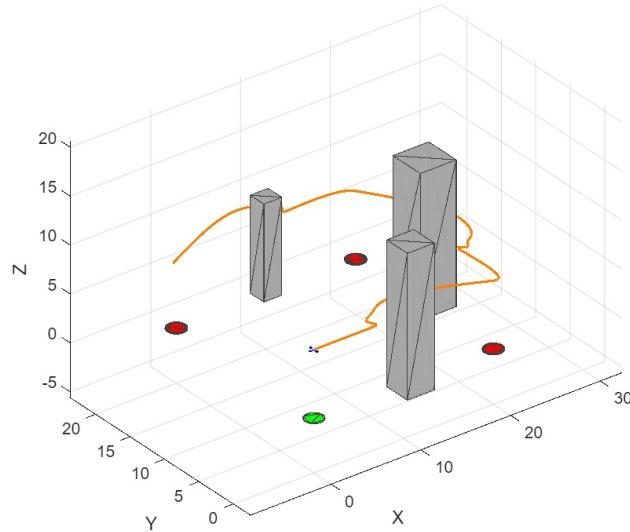


Figure 17: UAV Trajectory 3 Obstacles and External Wrench Disturbances

It is possible to notice that the trajectory, also with disturbances and a new implementation with the estimator, is successfully tracked.

Hierarchical Control

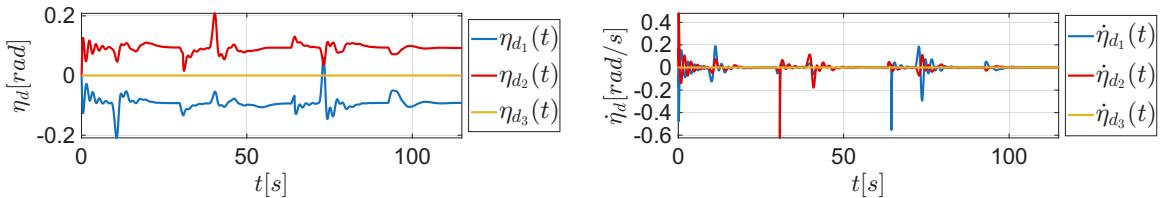


Figure 18: η and $\dot{\eta}$ Desired Values 3 Obstacles and External Wrench Disturbances
Hierarchical Control

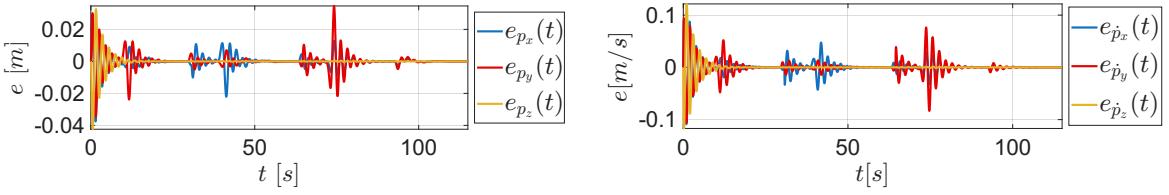


Figure 19: Position and Velocity Errors 3 Obstacles Hierarchical Control and External Wrench Disturbances

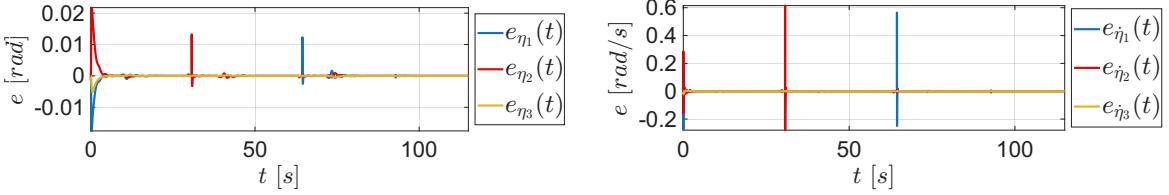


Figure 20: η and $\dot{\eta}$ Errors 3 Obstacles Hierarchical Control and External Wrench Disturbances

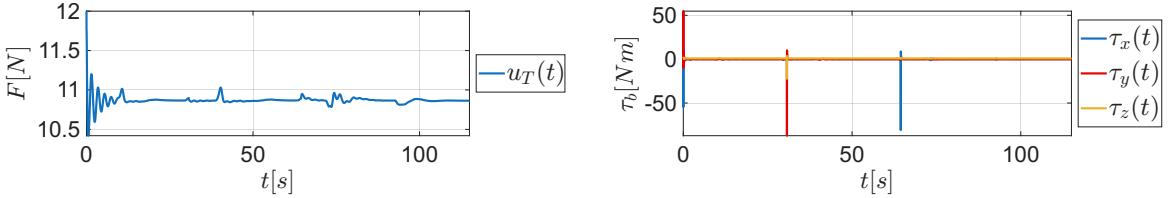


Figure 21: Hierarchical Control Inputs 3 Obstacles and External Wrench Disturbances

As before, the errors continue to exhibit a slight *oscillatory* behavior (more noticeable than before due to the shorter time span of the plot under analysis). However, an excellent **disturbance rejection** can be observed.

Geometric Control

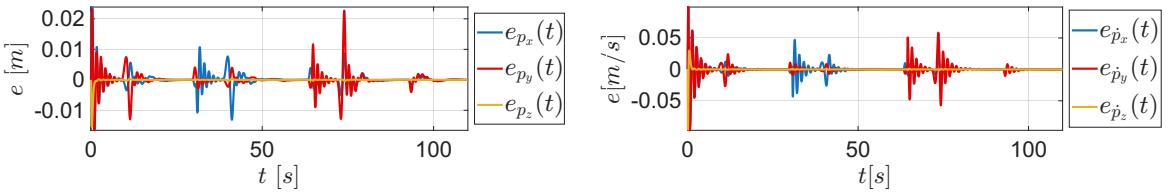


Figure 22: Position and Velocity Errors 3 Obstacles Geometric Control and External Wrench Disturbances

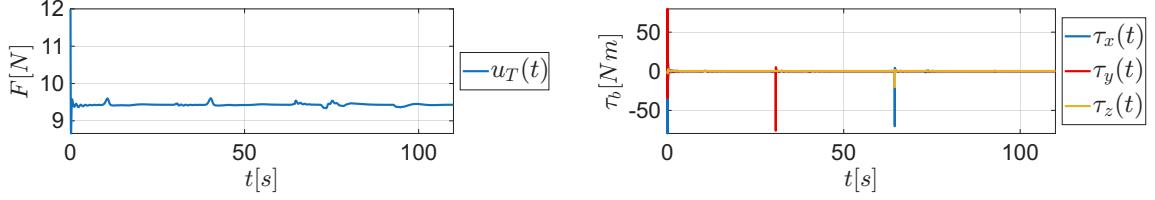


Figure 23: Geometric Control Inputs 3 Obstacles and External Wrench Disturbances

Obviously, the errors are very similar and excellent in terms of order of magnitude. This indicates that **the estimator is working very well**.

Moreover, the initial error along the z -axis is slightly higher due to the force disturbance exerted along the same axis. As highlighted in the previously mentioned *Homework*, this disturbance along the z -axis can be interpreted as a *model uncertainty* in the value of the mass.

Furthermore, the control input u_T appears much more 'linear', especially in the initial phase, compared to the hierarchical control, while τ_b is practically the same for the same reason as before.

Passivity-Based Control

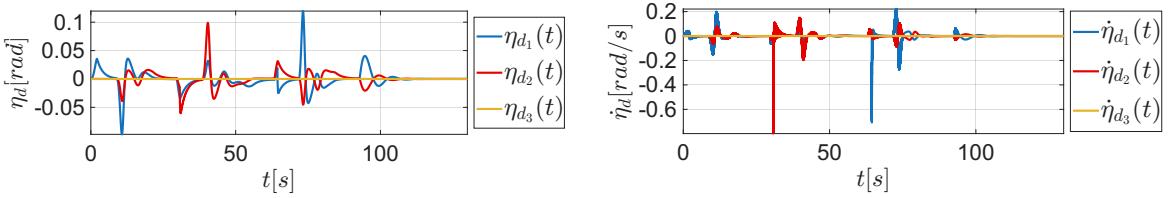
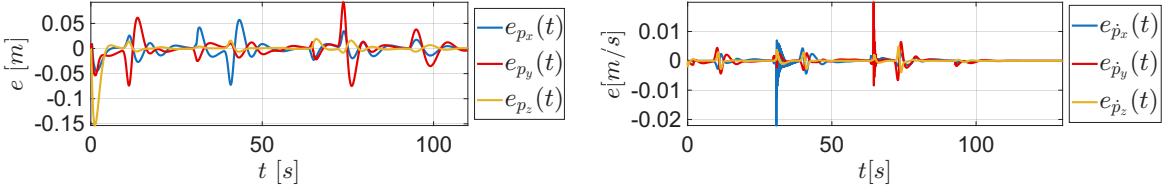
Figure 24: η and $\dot{\eta}$ Desired Values 3 Obstacles and External Wrench Disturbances

Figure 25: Position and Velocity Errors 3 Obstacles Passivity-Based Control and External Wrench Disturbances

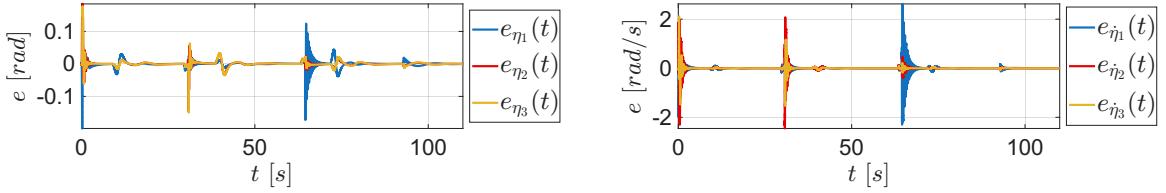


Figure 26: η and $\dot{\eta}$ Errors 3 Obstacles Passivity-Based Control and External Wrench Disturbances

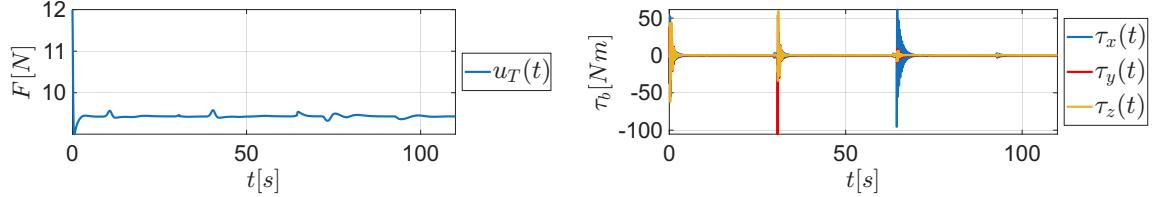


Figure 27: Passivity-Based Inputs 3 Obstacles and External Wrench Disturbances

As before, for the *Passivity-Based Control*, the behavior is entirely comparable to that obtained with *Geometric Control*.

Due to the disturbances, the errors are slightly higher but, once again, entirely acceptable (this behavior of this type of control will be analyzed in the next section).

The *Passivity-Based Control* is more sensitive to disturbances along the z -axis, leading to larger oscillations and errors compared to other control methods.

This is due to a lower choice of proportional gains compared to the other two types of control (as will be shown in the next section).

No-Estimator

Obviously, *without* the estimator, the behavior is significantly **compromised** due to the **lack of disturbance compensation**. The disturbances (in this case *constant*, but the same behavior would occur with *sinusoidal* disturbances (only more oscillatory)) introduce some **offsets** into the system that shifts the errors upward or downward, making them, naturally, larger and, in some situations, **unacceptable**.

Especially in the presence of obstacles, the drone tends to "vibrate" due to the external disturbances, which are further compounded by the repulsive force generated by the obstacles themselves.

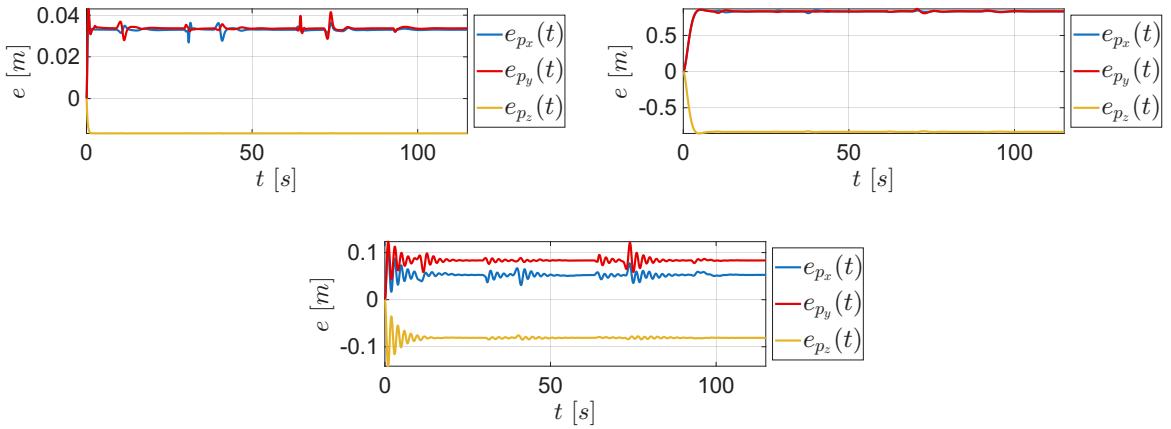


Figure 28: Position and Velocity Errors 3 Obstacles, External Wrench Disturbances and No Estimator. At the top left Geometric Control, at the top right Passivity-Based Control, and at the bottom center Hierarchical Control

From Figure 28, it is evident that, with the chosen *parameters*, the **Passivity-Based Control** is significantly **more sensitive to external wrench disturbances**. In fact, the errors, in the absence of the estimator, are one order of magnitude higher. This is due to the use and *tuning* of the gains in the **Geometric Control**, where **higher gains** were selected. These allow the system to better "*resist*" the constant disturbances and maintain a **small error**, compared to the behavior of the *Passivity-Based Control*.

In fact, for other values of the parameters such as $K_p = 10I_3$ and $K_d = 0.5I_3$, the errors, without estimator, are much smaller with respect the previous case. So, tuning is very important for every control.

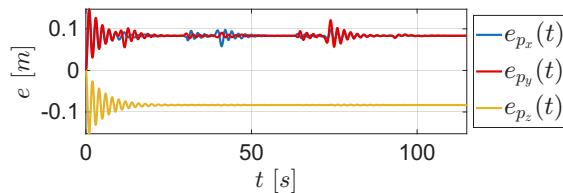


Figure 29: Position and Velocity Errors 3 Obstacles, External Wrench Disturbances and No Estimator, Passivity-Based Control with higher gains

Furthermore, the high error observed highlights the importance and necessity of implementing an estimator to ensure accurate trajectory tracking for the UAV

Local Minima Problem

As introduced earlier, one of the inherent problems of the *APF Algorithm* is the occurrence of **Local Minima**. This situation arises when the **Total Force** (*Attractive* +

Repulsive) becomes zero.

In practice, to visualize this issue, the scenario with a *Single Obstacle* was used.

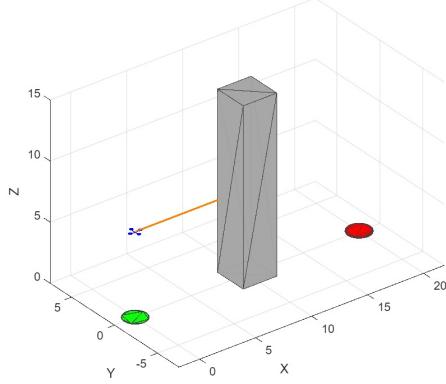


Figure 30: UAV Trajectory Local Minima

Optimized APF Algorithm

To address this issue, an improved version of the *APF Algorithm* has been implemented, based on the use of **Virtual Forces** as presented in [2].

The equations are slightly different compared to the previous version. In fact:

$$F_{rep}(\mathbf{x}) = K_{rep} \left(1 - \frac{\rho}{\rho_0} \right) \left(\frac{\mathbf{x}_o - \mathbf{x}}{\rho^3} \right) \quad F_{vir}(\mathbf{x}) = -K_{vir} \frac{1}{\rho}$$

where K_{vir} is the *Virtual Force Gain*.

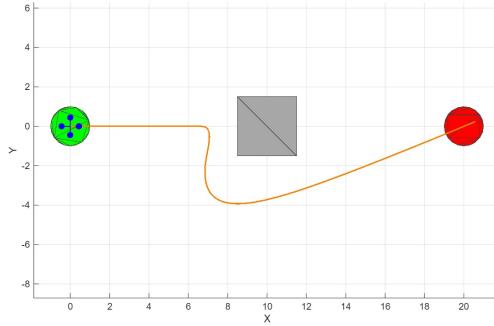


Figure 31: UAV Trajectory Improved APF Top View

Hovering

Hovering refers to the ability of a drone *to maintain a fixed position in space, remaining suspended in the air without drifting in any direction*. This means the drone

counteracts gravity and **stabilizes itself actively**, despite the presence of *external disturbances* such as wind or turbulence.

It is a fundamental capability, especially for vertical take-off and landing (*VTOL*) drones. Hovering is essential for tasks that require *precise positioning*, such as inspection, aerial photography, surveillance, or when the drone needs to pause before continuing its path. A *well-designed control system* is crucial to achieve *stable and reliable* hovering, as it must continuously adjust the *thrust* and *orientation* to maintain **equilibrium**.

In the following, the behavior of a drone in hovering will be shown, without any type of movement, but with **dynamic external disturbances along all directions and around all axes**.

From the video at the end of this report and from the following plots, it is possible to notice that **the quadrotor shows good hovering capabilities**.

A 5th-Order Estimator has been used with $r = 5$ and $c_0 = k_0 = 10$.

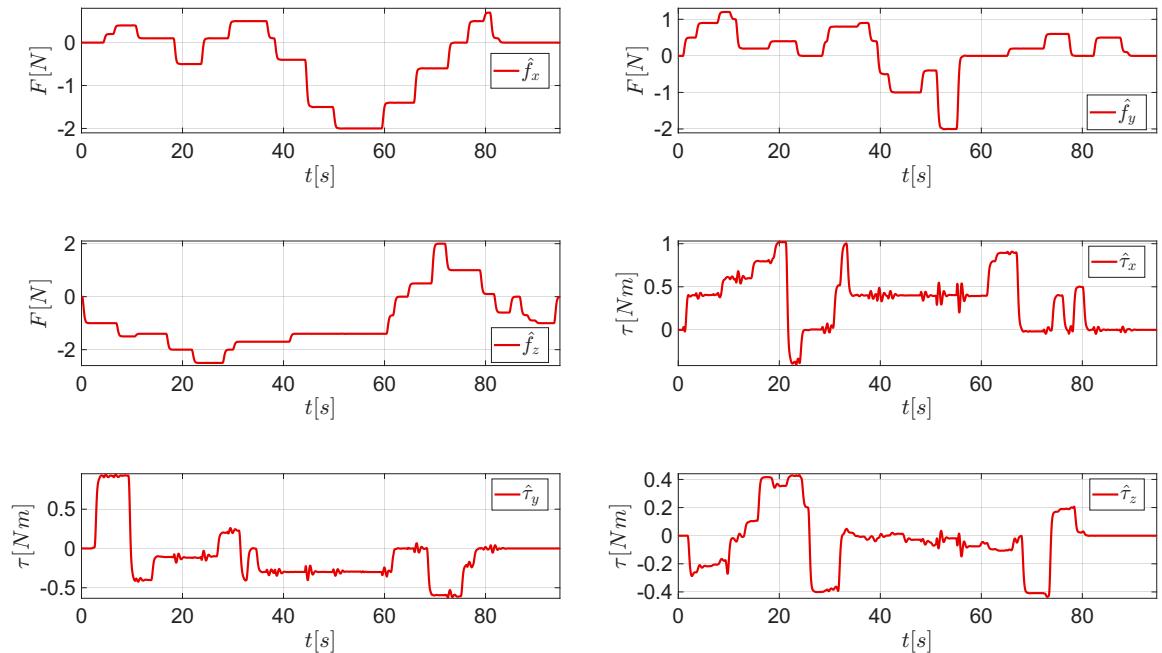


Figure 32: Wrench Estimation Hovering

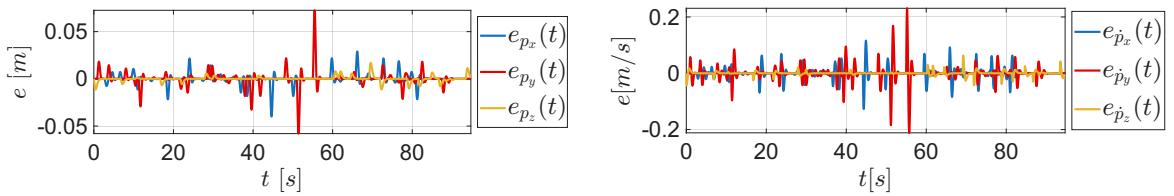
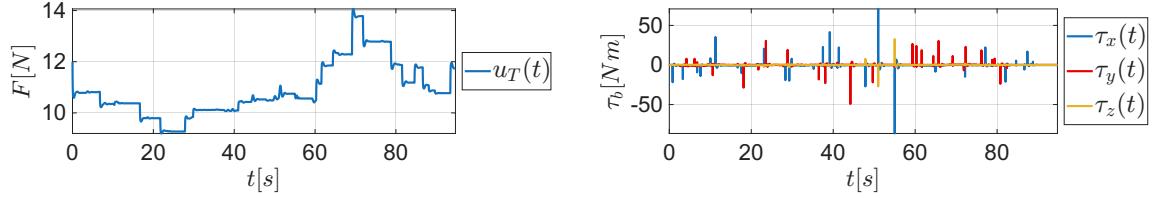


Figure 33: Position and Velocity Errors City Hovering

Figure 34: u_T and τ_b Hovering

The error is **very low**, except for a few *peaks* caused by *abrupt transitions of the disturbances* from one value to another.

It is important to note, and also quite evident, that the value of u_T closely follows the same trend as \hat{f}_z . Disturbances along the x - and y - axes cause lateral "deviations" which are stabilized in a very short time.

Among the external inputs, the drone is most affected precisely by the latter.

From an implementation point of view, this behavior can also be useful when studying a *drone equipped with a manipulator arm* and so, the so-called **Aerial Manipulator**. With a **Decentralized Control Approach**, the UAV and the manipulator can be seen as *two separated entities* with two *different own controllers*.

Doing this, it is possible to control the UAV independently from the manipulator and considering all the movement done by the manipulator simply as disturbances for the drone. In fact, thanks to some transformation and **Jacobians**, it is possible to 'propagate' the resulting forces.

Thanks to the use of the **CATIA** modeling software, it was possible to analyze how a robot of these dimensions reacts to an applied *distributed load* of 3 N from below. The resulting **Von Mises Stress** values are fully acceptable, with the maximum stresses, unsurprisingly, occurring *near the propellers* and at the *base structure*.

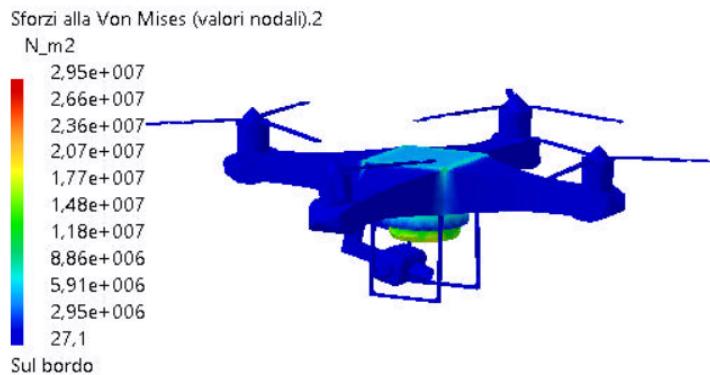


Figure 35: FEM Analysis

Ground Effect and Ceiling Effect

When a drone operates close to a surface, such as the ground or a ceiling, it experiences changes in aerodynamic behavior known as the **Ground Effect** and the **Ceiling Effect**. The formulas used were the following

$$\frac{u_{T,IGE}}{u_{T,OGE}} = \frac{1}{1 - (\frac{\rho}{4z})^2 - \frac{z\rho^2}{\sqrt{(d^2+4z^2)^3}} - \frac{z\rho^2}{2\sqrt{(2d^2+4z^2)^3}}} \quad \frac{T_{ICE}}{T_{OCE}} = \frac{1}{1 - \frac{1}{k_1}(\frac{\rho}{z+k_2})^2}$$

From [7], the parameters $k_1 = 6.924$ and $k_2 = 3.782$ have been chosen.

Now the results.

Let us assume there is a *ceiling* at a height of 10 m and a ground at a height of 0 m. Furthermore, the parameters chosen are $d = 0.2$ m and $\rho = 0.15$ m.

Considering the complete formula, we can assume that the *Ground Effect* is negligible when $z > 4\rho$, while the *Ceiling Effect* is negligible when $z > h - 4\rho$, with h the height of the ceil.

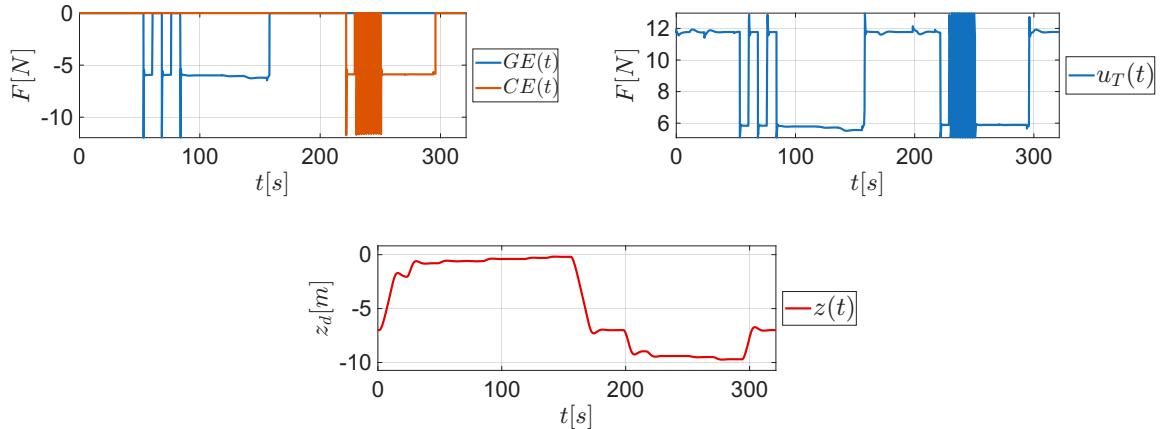


Figure 36: Ground Effect and Ceiling Effect results. On top left the two effects; on top right the total thrust u_T ; at the bottom the desired z -position of the UAV

As can be seen from the graphs, when the drone approaches the *boundary zone* where the effects come into play (around 0.6 m and 9.4 m), the thrust exhibits *oscillatory behavior* (as does the drone's z -position), since the drone alternates between being affected and not affected by the *Ground Effect* or *Ceiling Effect* while trying to stabilize itself. Once the drone fully enters the zone, here these effects are *active*, it tends to **reduce its thrust**, and consequently the **propellers speeds**, in order to remain at the *desired position*. This also allows for **battery savings**.

As can be observed, the *Ceiling Effect* tends to destabilize the drone more significantly during the transition phase. This is due to the parameter choices made earlier.

Conclusion and Future Works

This project proposes several examples in order to thoroughly study various **Control Approaches** and their **performances**.

Simulation outcomes showed that the implemented control strategy enables **stable** and **precise** UAV navigation, even when encountering unexpected obstacles. In fact, the combination of **Hierarchical Control**, **Geometric Control** or **Passivity-Based Control** with the **APF** technique effectively prevents *collisions*, significantly boosting the UAV's autonomy in *challenging* and *unpredictable environments*.

Obviously, for a **Real-Time Implementation** it is necessary to replace the predefined knowledge of obstacles with sensors such as **LiDAR**, **GPS**, or **IMU** (among others), which allow for the detection of the surrounding environment.

It can also be concluded that **Estimators** are **essential** due to the inevitable presence of *external disturbances* and *model uncertainties*, because **NO Ideal World exists**.

Various examples examined throughout this work aimed to highlight the differences between **Hierarchical Control**, **Geometric Control** and **Passivity-Based Control**, emphasizing the *strengths* and *weaknesses* of each controller, particularly in how they respond to *external disturbances* both with and without an estimator.

To conclude, some **Aerodynamic Effects** have been studied in order to highlight the necessity of accounting for these important behaviors that a drone may experience during various tasks.

Neglecting such effects could lead to serious issues, especially considering that nowadays drones are widely used for maintenance operations and other critical tasks, where malfunctions could result in catastrophic consequences.

The work carried out was of great personal interest.

Possible future developments could include, as mentioned in the final section, the implementation of a **Decentralized Control** with a *Robotic Arm* (such as those studied in the *Foundations of Robotics* Course), in order to make the system even more engaging and functionally rich.

Video References

UAV 3 Obstacles Waypoints Task: <https://youtu.be/m2e9zW4P-9M>

UAV City Waypoints Task: <https://youtu.be/033kO2RPhyQ>

Local Minima Single Obstacle: <https://youtu.be/FeFIXa-XepQ>

Local Minima Solution with Virtual Force, Single Obstacle: <https://youtu.be/6hmZGKph0FU>

Local Minima Solution with Tangential Force, Single Obstacle: <https://youtu.be/tzNrr8daEMs>

No Reachable Goal Near Obstacle: <https://youtu.be/rXLhm-J2d6Q>

No Reachable Goal Inside Obstacle: <https://youtu.be/aKbrqEWRKEM>

UAV 3 Obstacles Waypoints Task with External Disturbances: <https://youtu.be/qJ5edxEZ5BU>

UAV Hovering with External Disturbances: https://youtu.be/fhSy_LN2u-c

Bibliography

- [1] Oussama Khatib, *Real-time Obstacle Avoidance for Manipulators and Mobile Robots*. The International Journal of Robotics Research, Vol.5, No 1, Spring 1986.
- [2] Alfian Ma'arif, Wahyu Rahmani, Marco Antonio Márquez Vera, Aninditya Anggari Nuryono, Rania Majdoubi, Abdullah Çakan, *Artificial Potential Field Algorithm for Obstacle Avoidance in UAV Quadrotor for Dynamic Environment*. 2021 IEEE International Conference on Communication, Networks and Satellite (Comnetsat).
- [3] Fabio Ruggiero, *Aerial Robotics Slide*. Field and Service Robotics Course 2024/2025.
- [4] Emmanuel Patellaro, *Homework 3*. Field and Service Robotics Course 2024/2025.
- [5] Emmanuel Patellaro, *Homework 1*. Field and Service Robotics Course 2024/2025.
- [6] Fabio Ruggiero, Jonathan Cacace, Hamid Sadeghian, Vincenzo Lippiello, *Impedance Control of VToL UAVs with a Momentum-based External Generalized Forces Estimator*. 2014 IEEE International Conference on Robotics & Automation (ICRA), Hong Kong Convention and Exhibition Center, May 31 - June 7, 2014. Hong Kong, China.
- [7] P. J. Sanchez-Cuevas, G. Heredia and A. Ollero, *Multirotor UAS for bridge inspection by contact using the ceiling effect*. 2017 International Conference on Unmanned Aircraft Systems (ICUAS) June 13-16, 2017, Miami, FL, USA