Università
degli Studi
di Padova

DIPARTIMENTO DI TECNICA E GESTIONE DEI SISTEMI INDUSTRIALI

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA MECCATRONICA

Hierarchical Control of Fully-Actuated Hexarotors:

Architecture Revision and Physics-Based Validation

Relatore: Dott.ssa Giulia Michieletto

Correlatore: Dott. Massimiliano Bertoni

Laureando: Matteo Russo

Matricola: 2080496

Anno Accademico 2024-2025

Data di laurea 11/07/2025

Acknowledgements

Abstract

Nowadays, the use of aerial robots, such as multi-rotor UAVs (Unmanned Aerial Vehicles), is becoming increasingly widespread across various applications, including exploration, surveillance, monitoring, and more. For this reason, new optimal control strategies for such platforms are being actively researched. The objective of this thesis is to improve a nonlinear control solution, known as Hierarchical Control, for fully-actuated hexarotors, which are multi-rotor UAVs equipped with six propellers arranged in such a way as to achieve a fully-actuated platform, thus enabling control over all six degrees of freedom of a rigid body in three-dimensional space. Specifically, the work aims to describe the operating principles of the Hierarchical Controller, improve its existing architecture, and present a revised version that addresses current shortcomings and introduces structural enhancements.

Subsequently, to assess the performance of the improved Hierarchical Controller architecture and to carry out an objective comparison, two other state-of-the-art control strategies will be introduced, and several numerical simulations will be performed in MATLAB/Simulink. In particular, numerical simulations in MATLAB/Simulink were carried out to evaluate the performance difference between the improved Hierarchical Controller and its original version. Afterwards, the improved Hierarchical Controller is compared with the other state-of-the-art control strategies under ideal conditions, in the presence of sensor noise, and during a representative real-world tracking task.

This thesis work concludes with a physics-based simulation in Gazebo, which confirms the results obtained through numerical validation by comparing the performance of the proposed controller with that of an alternative control strategy along a predefined trajectory.

Contents

1. INTRODUCTION

In recent years, there has been a significant increase in the use of aerial robots, whose most commonly referenced name in the scientific literature appears to be *Unmanned Aerial Vehicles* (UAVs).

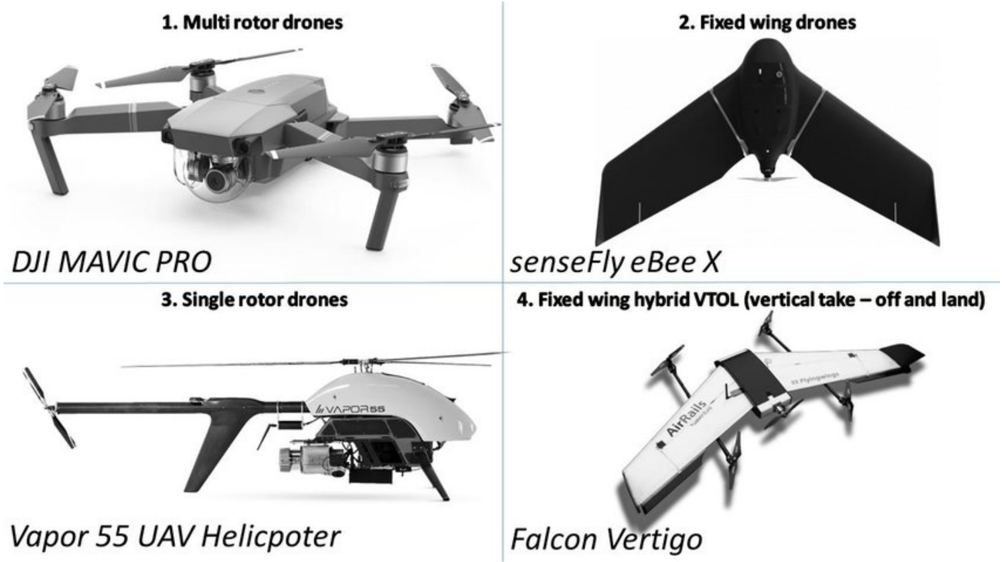


Figure 1.1: Different types of UAVs, from [1]

Typically, the term UAVs, or more commonly drones, refers to three main types of platforms (see Fig.1.1), which differ in both structure and operating principle. These are:

- **Fixed-wing drones**, which rely on the aerodynamic lift generated by their wings to remain airborne and may feature one or more engines, depending on the specific configuration. Due to their structural characteristics, they require runways for take-off and landing.
- **Rotary-wing drones**, including both single rotor and multi-rotor configurations (see Fig. 1.1), which are described in more detail in Section 1.1. These systems are capable of vertical take-off and landing (VTOL), and therefore do not require runways for operation.

- **Hybrid drones**, which combine the advantages of both fixed-wing and rotary-wing platforms.

This thesis work, as indicated by the title, focuses on multi-rotor UAVs. The choice to focus on this type of UAVs is motivated by several factors. First of all, multi-rotor UAVs feature a relatively simple and symmetric mechanical structure (resulting in lower production costs), which allows for a simpler dynamic analysis compared to other, more complex aerial systems, such as fixed-wing drones or hybrid VTOL platforms. This structural simplicity makes multi-rotor UAVs a highly versatile class with numerous potential applications. All these characteristics make them a stimulating research field for the implementation of advanced control algorithms.

1.1. Multi-Rotor UAVs

Multi-rotor UAVs are generally composed of a central body and multiple evenly spaced actuators. Typically, the rotors are arranged along parallel axes aligned with the direction of the gravitational force, to counteract it. This type of aerial robot has gained significant popularity due to its high maneuverability with capabilities such as vertical take-off/landing and hovering, its autonomy, and mechanical simplicity. A common distinction among such multi-rotor systems is based on the number of rotors they possess. The most common configurations are trirotors, quadrotors, and hexarotors, although octorotors are also available.

Another possible classification of multi-rotor UAVs can be made by considering the actuation properties of these systems. Recalling that a rigid body in three-dimensional space exhibits six degrees of freedom (DoFs), it is possible to distinguish between *underactuated* and *fully-actuated* platforms.

Denoting with n the number of rotors, the term *underactuated* refers to a UAV with $n < 6$, meaning that the UAV is unable to perform pure translational movements without altering its orientation, since it is not possible to control all six DoFs. A well-known example of an underactuated system is the quadrotor, which is the most widely used UAV due to its energy efficiency, ease of construction, and lower production cost compared to alternatives with more rotors.

On the other hand, the term *fully-actuated* refers to UAVs with $n \geq 6$, thus allowing the UAV to perform pure translational movements without changing its orientation. Therefore, to achieve full actuation, the minimum number of rotors required is trivially six, in order to control all six DoFs of a rigid body in 3D space. However, this condition alone is not sufficient if the six rotors remain oriented in parallel directions. To achieve full actuation, the orientation of the rotors must be modified. This can be accomplished using fixed rotors, meaning that each rotor is appropriately tilted by a certain angle along its corresponding arm so that its spinning axis is not parallel to the gravity vector and not parallel among them, or by employing rotors capable of varying their inclination. In this thesis, we consider a fully-actuated hexarotor with fixed rotors as the reference platform. The choice to focus on a fully-actuated system is motivated by the increasing demand for more sophisticated applications. For example, the emerging field of aerial physical interaction [3], in which UAVs are required to operate not only as flying sensors but also as aerial manipulators, has led to intensive research on achieving full actuation in UAVs. This brief introduction to multi-rotor UAVs concludes by outlining a set of applications involving both fully-actuated and underactuated aerial platforms, either currently existing or under active development, including:

- Contact-based inspection [6], [7]
- Visual control for the maintenance of hard-to-access structures [8]
- Remote agriculture [9]
- Civil applications such as rescue operations [10] and military applications [11]

1.2. Navigation Task for Multi-Rotor UAVs

The autonomous navigation of multi-rotor UAVs is a complex problem that requires careful consideration of various constraints, which depend primarily on the specific application intended for the given platform. An accurate selection of the hardware to mount on the UAV is crucial to achieve a suitable perception and understanding of the environment. If necessary, it is essential to design a path tailored to the aerial platform's mission. This includes defining an appropriate motion law to ensure smooth and continuous trajectories that respect the physical limitations of the system. Above all, an appropriate control strategy must be implemented to generate the control inputs for the UAV, which must be able both to make the UAV follow the established path and to manage external disturbances and uncertainties in the dynamic models of the platforms.

The main navigation tasks typically addressed using fully-actuated multi-rotor UAVs are defined below:

1. **Hovering**, which consists in maintaining a fixed and stable position in space with a constant arbitrary orientation, counteracting gravity and also compensating for external disturbances.

2. **Position Trajectory Tracking**, which involves following a time-varying position trajectory, ensuring that the UAV remains as close as possible to the desired position at each instant.

3. **Position Trajectory Tracking and Attitude Stabilization**, which consists in following a time-varying position trajectory while stabilizing the UAV orientation at constant desired values.

4. **Position and Attitude Trajectory Tracking**, which consists in following a time-varying trajectory both in position and orientation.

These control problems are listed in order of increasing complexity. It is precisely the growing complexity of emerging applications that has driven the development of innovative control strategies for these fully-actuated platforms. In the literature, many control strategies are proposed to enhance the stability properties of UAVs while tracking a desired trajectory. These are generally based on linear control systems such as proportional-derivative controllers (PD) or linear quadratic regulators (LQR). Recently, nonlinear control strategies have been gaining traction, with their effectiveness demonstrated through experimental tests.
Following this, Table 1.1 gathers the main existing control solutions, grouped according to the control problem to be solved:

Control Problem	**Common Control Strategies**
Hovering	PID (Proportional-Integrative-Derivative) [16] LQR (Linear Quadratic Regulator) [17]
Position trajectory tracking	PID (Proportional-Integrative-Derivative) [18] GC (Geometric Control) [15] FC (Flatness-based Control) [14] HC (Hierarchical Control) [14] NBS (Nonlinear BackStepping) [19] MPC (model predictive control) [20]
Position tracking + attitude stabilization	GC (Geometric Control) [15] FC (Flatness-based Control) [14] HC (Hierarchical Control) [14] NBS (Nonlinear BackStepping) [19] MPC (model predictive control) [20]
Position + attitude trajectory tracking	GC (Geometric Control) [15] FC (Flatness-based Control) NBS (Nonlinear BackStepping) [19] MPC (model predictive control) [20]

Table 1.1: Control strategies for multirotor UAVs depending on the control problem

The optimal resolution of the navigation task for fully-actuated hexarotors would pave the way for new application scenarios. In fact, thanks to the more precise and smoother maneuvers that a fully-actuated hexarotor is capable of performing compared to an underactuated platform, it becomes feasible to employ such UAVs in tasks that involve physical interaction with the environment, both in outdoor and indoor settings. Furthermore, investigating innovative control solutions capable of achieving equivalent navigation task performance while reducing energy consumption would make it possible to optimize the energy efficiency of UAVs, thereby extending their flight endurance. For this reason, the principal objective of this thesis is to improve the Hierarchical Controller (HC) proposed in [13], [14], which is considered highly promising for solving the navigation task of position tracking and attitude stabilization using a fully-actuated hexarotor.

1.3. Thesis Objectives

This thesis focuses on the analysis of a nonlinear Hierarchical Controller (HC) based on the Zero-Moment-Direction. In particular, the architecture of such a control strategy, previously described in [14], is revised and improved, and a novel version is proposed. Subsequently, this improved version is employed to control a fully-actuated hexarotor with fixed tilted propellers. Initially, simulations are carried out in MATLAB/Simulink to assess the trajectory tracking and orientation stabilization capabilities, comparing the performance of the improved version of the Hierarchical Controller with two other state-of-the-art control strategies, the Geometrical Controller (GC) and the Flatness-based Controller (FC). The work later move towards to physics-based simulations in Gazebo. More specifically, the structure of this thesis is defined as follows:

- **Chapter 2**: A review of theoretical concepts required for understanding this thesis work. Specifically, skew-symmetric matrices widely used in the Hierarchical Controller will be recalled. A comprehensive overview of quaternions, employed for UAV pose representation, will also be provided. Finally, Lyapunov stability is presented, as it is used to prove the stability of the Hierarchical Controller.
- **Chapter 3**: The mathematical model of a multi-rotor platform is derived and the Hierarchical Controller is presented. In particular, the original architecture proposed in [14] is described, along with the corrections made to it, concluding with the presentation of an improved version.
- **Chapter 4**: The reference Hexarotor platform is described, along with the benchmark used for validating the improved Hierarchical architecture. This includes the two other state-of-the-art control solutions adopted and the selected key performance indices. Finally, all numerical and physics-based validations are presented and discussed, along with the associated navigation tasks.
- **Chapter 5**: The main conclusions of the thesis work are drawn, based on the insights gained from the numerical and physics-based tests presented in the previous chapter. It is clarified whether the enhancements introduced to the Hierarchical Controller architecture are indeed effective. The chapter concludes by proposing possible future directions in which this thesis work can be further developed.

2. PRELIMINARIES

This chapter provides a review of the mathematical concepts necessary for a complete understanding of this thesis, with particular focus on pose representation using quaternions and on Lyapunov stability.

2.1. Skew-Symmetric Matrices

Skew-symmetric matrices, also called antisymmetric matrices, are square matrices $A \in \mathbb{R}^{n\times n}$ that satisfy the property:

$$A^\top = -A.$$

A direct consequence is that all diagonal elements are zero, i.e., $a_{ii} = 0$ for every $i = 1, \dots, n$.

In the three-dimensional case, there exists a one-to-one correspondence between vectors $\mathbf{v} \in \mathbb{R}^3$ and skew-symmetric matrices $[\mathbf{v}]_\times \in \mathbb{R}^{3\times 3}$ defined as:

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \quad \longleftrightarrow \quad [\mathbf{v}]_\times = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}.$$

Given two vectors $\mathbf{v}$ and $\mathbf{u}$, their cross product can be represented as follows:

$$\mathbf{v} \times \mathbf{u} = [\mathbf{v}]_\times \mathbf{u}.$$

This representation is widely used in mechatronics and robotics to describe rotation operators and angular velocities.

2.2. Quaternions

In general, a quaternion represents an extension of complex numbers, formally defined by an ordered quadruplet of real numbers as follows:

$$\mathbf{q} = [q_0, q_1, q_2, q_3] \tag{2.1}$$

It is possible to associate the quaternion $\mathbf{q}$ with the matrix $\mathbf{A}_q$ formed as follows:

$$\mathbf{A}_q = \begin{bmatrix} q_0 + iq_1 & q_2 + iq_3 \\ -q_2 + iq_3 & q_0 - iq_1 \end{bmatrix} = \begin{bmatrix} \alpha & \beta \\ -\overline{\beta} & \overline{\alpha} \end{bmatrix}$$

obtaining the matrix representation of a quaternion, which is useful for performing the product between quaternions, since given two quaternions $\mathbf{p}$ and $\mathbf{q}$, the matrix associated with the product of the two quaternions is computed as:

$$\mathbf{A}_{pq} = \mathbf{A}_p \mathbf{A}_q$$

Therefore, if we want to multiply two quaternions, we can multiply their corresponding matrices and then derive the quaternion associated with the resulting matrix.
Another possible representation of a quaternion is the vectorial one. Given a quaternion $\mathbf{q} = [q_0, q_1, q_2, q_3]$, q_0 is named the scalar part and $\mathbf{w} = [q_1, q_2, q_3]$ is named the vectorial part of the quaternion $\mathbf{q}$. If $\mathbf{w} = 0$, then $q \in \mathbb{R}$ and it is called a real quaternion. If $q_0 = 0$, then $\mathbf{q}$ is called a pure quaternion (or vectorial quaternion). In this case, it can be identified with the vector $\mathbf{w} \in \mathbb{R}^3$.
Finally, by introducing pure quaternions $\mathbf{i} = [0, 1, 0, 0]$, $\mathbf{j} = [0, 0, 1, 0]$,
$\mathbf{k} = [0, 0, 0, 1]$, it is possible to obtain the algebraic representation of a quaternion:

$$\mathbf{q} = q_o + \mathbf{i}q_1 + \mathbf{j}q_2 + \mathbf{k}q_3 \tag{2.2}$$

The advantage of this fourth representation is that it allows the multiplication of two quaternions using algebra.

2.2.1. Fundamental Properties

The following section recalls some fundamental properties of quaternions, which are essential for understanding certain mathematical steps not only in this thesis, but also in the related literature.

1. Given a quaternion $\mathbf{q} = [q_0, \mathbf{w}]$, the coniugate of $\mathbf{q}$ is $\mathbf{q}^* = [q_0, -\mathbf{w}]$
2. Given a quaternion $\mathbf{q} = [q_0, q_1, q_2, q_3]$, its norm is $\|\mathbf{q}\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}$.
3. If $\mathbf{q} = [q_0, \mathbf{w}]$, then $\|\mathbf{q}\|^2 = q_0^2 + \mathbf{w} \cdot \mathbf{w} = \mathbf{q}\mathbf{q}^*$
4. Named $\mathbf{A}_q$ the matrix associated with $\mathbf{q}$, then $\|q\|^2 = det\mathbf{A}_q$
5. Given two quaternions $\mathbf{p} = [p_0, \mathbf{v}]$ and $\mathbf{q} = [q_0, \mathbf{w}]$, their product is:

$$\mathbf{p}\mathbf{q} = [p_0 q_0 - \mathbf{v} \cdot \mathbf{w}, p_0 \mathbf{w} + q_0 \mathbf{v} + \mathbf{v} \times \mathbf{w}]$$

2.2.2. Quaternions and Rotations

A quaternion $\mathbf{q}$ is defined as *unit quaternion* if $\|\mathbf{q}\| = 1$.
If $\mathbf{q}$ is a *unit quaternion*, then $\mathbf{q}^{\mathbf{-1}} = \mathbf{q}^*$, because $\mathbf{q}\mathbf{q}^* = \|\mathbf{q}\|^2$.
For each unit quaternion $\mathbf{q}$, exists an angle $\varphi \in [0, 2\pi]$ and at least a versor $\mathbf{u}$ such that

$$\mathbf{q} = \left[\cos\frac{\varphi}{2}, \sin\frac{\varphi}{2}\mathbf{u}\right] \tag{2.3}$$

it is thus straightforward that unit quaternions represent rotations in 3D space, given that $\mathbf{u}$ represents the rotation axis and φ denotes the rotation angle.
Therefore, it is possible to associate to a generic unit quaternion $\mathbf{q} = [q_0, \mathbf{w}]$ a rotation matrix, belonging to the Special Orthogonal group $SO(3) := \{\mathbf{R} \in \mathbb{R}^{3\times3} | \ \mathbf{R}^T\mathbf{R} = \mathbf{I}_3, \det(\mathbf{R}) = 1\}$, computed as follows:

$$\mathbf{R(q)} = \mathbf{I}_3 + 2q_0[\mathbf{w}]_\times + 2[\mathbf{w}]_\times^2 \tag{2.4}$$

As for the rotation matrices, it is possible to compose a sequence of rotations. Given two quaternions $\mathbf{q}_1$,$\mathbf{q}_2$, it is possible to combine the rotations associated to the quaternions through the quaternion product, which is defined as follows:

$$\mathbf{q}_1 \circ \mathbf{q}_2 = \mathbf{M}(\mathbf{q}_1)\mathbf{q}_2 = \mathbf{N}(\mathbf{q}_2)\mathbf{q}_1 \tag{2.5}$$

where

$$\mathbf{M(q)} = \begin{bmatrix} q_0 & -\mathbf{w}^T \\ \mathbf{w} & q_0\mathbf{I}_3 + [\mathbf{w}]_\times \end{bmatrix}, \mathbf{N(q)} = \begin{bmatrix} q_0 & -\mathbf{w}^\top \\ \mathbf{w} & q_0\mathbf{I}_3 - [\mathbf{w}]_\times \end{bmatrix}$$

The quaternion product is not commutative, so using $\mathbf{q_3} := \mathbf{q}_1 \otimes \mathbf{q}_2$, one has $\mathbf{R}(\mathbf{q}_3) = \mathbf{R}(\mathbf{q}_1)\mathbf{R}(\mathbf{q}_2)$. In conclusion, given two 3D coordinate systems $\mathscr{F}_x$, $\mathscr{F}_y$, such that the unit quaternion $\mathbf{q}$ indicates the relative rotation from $\mathscr{F}_x$ to $\mathscr{F}_y$, the time derivative of the generic quaternion $\mathbf{q}$ results:

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \circ \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} -\mathbf{w}^\top \\ q_0\mathbf{I}_3 + [\mathbf{w}]_\times \end{bmatrix} \boldsymbol{\omega} \tag{2.6}$$

where $\boldsymbol{\omega} \in \mathbb{R}^3$ is the angular velocity of $\mathscr{F}_x$ with respect to $\mathscr{F}_y$, expressed in $\mathscr{F}_x$.
In this work, unit quaternions are adopted to represent the attitude of the UAV. This choice is motivated by the need to overcome the singularities and discontinuities associated with Euler angle representations, such as gimbal lock, which can compromise the robustness and continuity of the control laws. Moreover, compared to rotation matrices, quaternions offer a more compact and computationally efficient formulation, which simplifies both the kinematic and dynamic equations involved in the control design process.

2.3. Lyapunov Stability Theory

Lyapunov stability theory provides a fundamental framework for analyzing the stability of equilibrium points in dynamical systems without explicitly solving the system's differential equations. The main idea is to construct a scalar function $V(\mathbf{x})$, known as the Lyapunov function, which is positive definite and decreases along the trajectories of the system.
Given a dynamical system described by

$$\dot{\mathbf{x}} = f(\mathbf{x}),$$

an equilibrium point $\mathbf{x} = \mathbf{0}$ is said to be *stable in the sense of Lyapunov* if for every $\epsilon > 0$, there exists a $\delta > 0$ such that

$$\|\mathbf{x}(0)\| < \delta \implies \|\mathbf{x}(t)\| < \epsilon, \quad \forall t \geq 0.$$

If, in addition, the Lyapunov function $V(\mathbf{x})$ satisfies

$$\dot{V}(\mathbf{x}) = \frac{\partial V}{\partial \mathbf{x}} f(\mathbf{x}) < 0, \quad \forall \mathbf{x} \neq \mathbf{0},$$

then the equilibrium is *asymptotically stable.*
Lyapunov's method is extensively employed in control theory, robotics, and mecha-

tronics as a rigorous and intuitive tool for the design and analysis of stable control systems, particularly in cases involving nonlinear dynamics or complex multi-input systems where traditional linear stability analysis may be insufficient.

The Hierarchical Controller discussed in this thesis has been developed based on this theoretical framework. In fact, the parameters underlying the controller have been specifically defined to always ensure the negativity of the Lyapunov functions developed to drive the desired mismatches to zero (see Sec. 3.2).

3. HIERARCHICAL CONTROL OF FULLY-ACTUATED HEXAROTOR

This chapter represents the core theoretical contribution of the thesis. First, the mathematical model of the considered fully-actuated hexarotor platform is derived, providing the fundamental equations that describe its dynamics and serve as the basis for control design. Building upon this, the Hierarchical Controller (HC) is introduced, with particular attention devoted to the original architecture proposed in [14]. The limitations and inconsistencies identified in the original formulation are then discussed, followed by a detailed presentation of the corrections implemented to address these issues. Finally, an improved version of the controller is proposed, incorporating both structural and conceptual enhancements aimed at increasing robustness, reliability, and overall performance.

3.1. Fully-Actuated Hexarotor Modeling

In this section, using the unit quaternion formalism to represent the attitude, the model of a star-shaped hexarotor is derived. The star-shaped hexarotor features tilted propellers whose spinning axes are not parallel to the gravity vector, with an angle between each arm of 60°. This design enables the platform to achieve full actuation, as the resulting control force can be assigned in a 3D domain, allowing the UAV to effectively track position and orientation references independently.

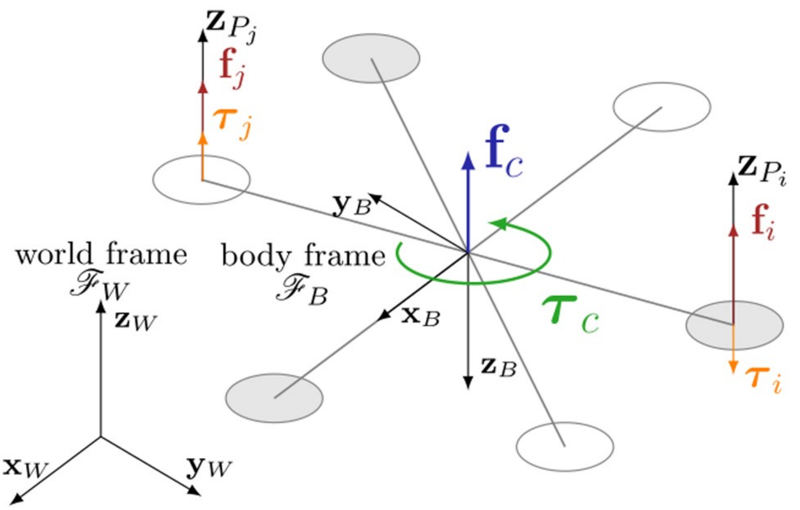


Figure 3.1: Schematic of a tilted hexarotor having CCW (gray) and CW (white) propellers, from[13]

Consider a generic hexarotor, composed of six propellers forming a hexagon centered at the *center of mass* (CoM) of the platform. As shown in Figure 3.1, two reference systems are introduced:

- $\mathscr{F}_W = \{\mathbf{O}_W - \mathbf{x}_W\mathbf{y}_W\mathbf{z}_W\}$ the inertial *world frame* (fixed).
- $\mathscr{F}_B = \{\mathbf{O}_B - \mathbf{x}_B\mathbf{y}_B\mathbf{z}_B\}$ the *body frame* with its origin coinciding with the CoM of the UAV.

The pose of the hexarotor with respect to the world frame is described by the pair $(\mathbf{p}, \mathbf{q}) \in \mathbb{R}^3 \times \mathbb{S}^3$, where the vector $\mathbf{p} \in \mathbb{R}^3$ denotes the position of O_B in $\mathscr{F}_W$, and the unit quaternion $\mathbf{q} \in \mathbb{S}^3$ represents the orientation of $\mathscr{F}_B$ with respect to $\mathscr{F}_W$. Named $\mathbf{v} \in \mathbb{R}^3$ the linear velocity of O_B in $\mathscr{F}_W$, and $\boldsymbol{\omega} \in \mathbb{R}^3$ the angular velocity of $\mathscr{F}_B$ w.r.t. $\mathscr{F}_W$, expressed in $\mathscr{F}_B$, then the position and orientation kinematics is governed by:

$$\begin{cases} \dot{\mathbf{p}} = \mathbf{v} \\ \dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \circ \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix} \end{cases}$$

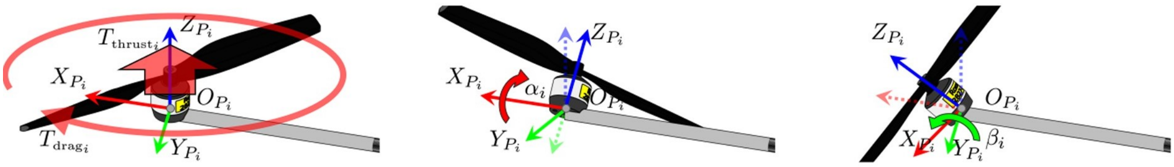
In order to derive the dynamics of the hexarotor, a single propeller is analyzed. Let the frame associated with the i-th propeller be defined as $\mathscr{F}_{P_i} = \{\mathbf{O}_{P_i} - \mathbf{x}_{P_i}\mathbf{y}_{P_i}\mathbf{z}_{P_i}\}$, where $i = 1, ..., 6$. The position $\mathbf{p}_{P_i} \in \mathbb{R}^3$ of the rotor center O_{P_i} is considered constant in the body frame $\mathscr{F}_B$ and it can be determined as follows:

$$\mathbf{p}_{P_i} = \mathbf{R}_Z(\gamma_i)\begin{bmatrix} L_{x_i} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos(\gamma_i) & -\sin(\gamma_i) & 0 \\ \sin(\gamma_i) & \cos(\gamma_i) & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} L_{x_i} \\ 0 \\ 0 \end{bmatrix}, \forall i = 1, ..., 6$$

where $L_{x_i} \in \mathbb{R}^+$ is the distance between $\mathbf{O}_{P_i}$ and $\mathbf{O}_B$. Instead, $\gamma_i = (i-1)\pi/3 \in [0, \pi]$ is the angular direction of the segment $\mathbf{O}_B\mathbf{O}_{P_i}$, with reference to the $\mathbf{x}_B\mathbf{y}_B$ plane of the CoM. In this way the six propellers lie on the $\mathbf{x}_B\mathbf{y}_B$ plane of the CoM
Lastly, it is possible to parameterize the propeller orientation by introducing the rotation matrix $\mathbf{R}_{P_i} \in SO(3)$ that represent the orientation of $\mathscr{F}_{P_i}$ with reference to the body frame $\mathscr{F}_B$:

$$\mathbf{R}_{P_i} = \mathbf{R}_z(\gamma_i)\mathbf{R}_y(\beta_i)\mathbf{R}_x(\alpha_i), \forall i = 1, ..., 6$$

where parameters α_i, β_i represent the tilt angles that defines the rotation plane of each propeller with respect to the body frame (see Fig.3.2)



(a) (b) (c)

Figure 3.2: (a): Terminal part of the i-th hexarotor arm showing the position $\mathbf{O}_{P_i}$ of the i-th propeller and the generated thrust $\mathbf{f}_i$ and drag $\boldsymbol{\tau}_i$; (b) and (c): Visualization of the possible reorientation of the i-th propeller around $\mathbf{x}_{P_i}$, with angle α_i (b), and around $\mathbf{y}_{P_i}$, with angle β_i (c). Adapted from [21].

Each i-th propeller, $i \in \{1, ..., n\}$, rotates with angular velocity $\boldsymbol{\omega}_i \in \mathbb{R}^3$ around its spinning axis, which is also assumed to have a constant direction in body frame $\mathscr{F}_B$. Indeed, the angular velocity can be expressed as $\boldsymbol{\omega}_i := \omega_i \mathbf{z}_{p_i}$, where $\omega_i \in \mathbb{R}$ indicates the controllable rotor spinning rate and $\mathbf{z}_{p_i} \in \mathbb{S}^2$ is a unit vector parallel to the rotor spinning axis, which depends also on the tilting angles as:

$$\mathbf{z}_{P_i} = \mathbf{R}_{P_i}\mathbf{z}_B \tag{3.1}$$

While spinning, each propeller exerts a *thrust/lift force* $\mathbf{f}_i \in \mathbb{R}^3$ and a *drag moment* $\boldsymbol{\tau}_i \in \mathbb{R}^3$, both oriented along the direction $\mathbf{z}_{p_i}$. These two quantities are related to the angular velocity $\boldsymbol{\omega}_i$ of the i-th propeller as follows:

$$\mathbf{f}_i = \sigma c_{f_i}|\omega_i|\omega_i \mathbf{z}_{p_i} \qquad \boldsymbol{\tau}_i = -c_{\tau_i}^{+}|\omega_i|\omega_i \mathbf{z}_{p_i} \tag{3.2}$$

where $c_{f_i}, c^+_{\tau_i} > 0$ and $\sigma \in \{-1, 1\}$ are constant parameters depending on the propeller shape; in particular, $\sigma = -1$ if the propeller rotates clockwise, and $\sigma = 1$ if the propeller rotates counterclockwise.
Defining $u_i := \sigma|\omega_i|\omega_i \in \mathbb{R}$ and $c_{\tau_i} := -\sigma c^+_{\tau_i} \in \mathbb{R}$, equation (3.2) can be rewritten as:

$$\mathbf{f}_i = c_{f_i} u_i \mathbf{z}_{p_i} \qquad \boldsymbol{\tau}_i = c_{\tau_i} u_i \mathbf{z}_{p_i} \tag{3.3}$$

By summing the contributions $\mathbf{f}_i$ of all the propellers, the *control force* $\mathbf{f}_c \in \mathbb{R}^3$ exerted at the hexarotor CoM is obtained. Conversely, summing all the contributions $\boldsymbol{\tau}_i$, while also accounting for the effect of the thrust forces, results in the *control moment* $\boldsymbol{\tau}_c \in \mathbb{R}^3$. Their expressions are:

$$\mathbf{f}_c = \sum_{i=1}^{n} \mathbf{f}_i = \sum_{i=1}^{n} c_{f_i} \mathbf{z}_{p_i} u_i \tag{3.4}$$

$$\boldsymbol{\tau}_c = \sum_{i=1}^{n} (\mathbf{p}_i \times \mathbf{f}_i + \boldsymbol{\tau}_i) = \sum_{i=1}^{n} (c_{f_i} \mathbf{p}_i \times \mathbf{z}_{p_i} + c_{\tau_i} \mathbf{z}_{p_i}) u_i \tag{3.5}$$

both expressed in $\mathscr{F}_B$, where $\mathbf{p}_i$ is the position of the CoM of the i-th propeller in $\mathscr{F}_B$.
Using the Newton-Euler approach and neglecting second-order effects, the hexarotor dynamic model is governed by the following equation set:

$$\begin{cases} m\ddot{\mathbf{p}} = -mg\mathbf{e}_3 + \mathbf{R}(\mathbf{q})\mathbf{f}_c \\ \mathbf{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \boldsymbol{\tau}_c \end{cases}$$

where $m > 0$ is the mass of the hexarotor, g is the gravitational constant, $\mathbf{e}_3 = [0, 0, 1] \in \mathbb{R}^3$ is the unit vector belonging to the canonical basis of $\mathbb{R}^3$ and identifying the direction of the gravitational force, and $\mathbf{J} \in \mathbb{R}^{3\times3}$ is the platform inertia matrix with respect to O_B expressed in $\mathscr{F}_B$.
Considering:

- State vector $\mathbf{x}_p := [\mathbf{p} \;\; \mathbf{q} \;\; \mathbf{v} \;\; \boldsymbol{\omega}]^\intercal \in \mathbb{R}^3 \times \mathbb{S}^3 \times \mathbb{R}^6$
- Control input vector: $\mathbf{u} = [u_1 \; ... \; u_6]^\intercal = [|\omega_1|\omega_1 \; ... \; |\omega_6|\omega_6]^\intercal \in \mathbb{R}^6$.

it is possible to compactly rewrite equations (3.4) and (3.5) as:

$$\mathbf{f}_c = \mathbf{F}\mathbf{u} \tag{3.6}$$

$$\boldsymbol{\tau}_c = \mathbf{M}\mathbf{u} \tag{3.7}$$

where $\mathbf{F}$, $\mathbf{M} \in \mathbb{R}^{3\times6}$ are the *control force input matrix* and the *control moment input matrix*, respectively.

The kinematic and dynamic behavior of the hexarotor can be described by the integrated nonlinear model:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{q}} \\ \dot{\mathbf{v}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ \frac{1}{2}\mathbf{q} \circ \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix} \\ -g\mathbf{e}_3 + m^{-1}\mathbf{R}(\mathbf{q})\mathbf{F}\mathbf{u} \\ \mathbf{J}^{-1}(-\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \mathbf{M}\mathbf{u}) \end{bmatrix} \tag{3.8}$$

The model based on rotation matrices is also presented, as it will be employed in the implementation of alternative control strategies. Choosing the state vector as $\mathbf{x}_p := [\mathbf{p} \quad \boldsymbol{\delta} \quad \mathbf{v} \quad \boldsymbol{\omega}]^\mathsf{T} \in \mathbb{R}^{12}$ in which $\boldsymbol{\delta} = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^\top \in \mathbb{R}^3$ is the orientation of the platform expressed in $\mathscr{F}_B$ with respect to $\mathscr{F}_W$, the model results:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\boldsymbol{\delta}} \\ \dot{\mathbf{v}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ \mathbf{T}(\boldsymbol{\delta})\boldsymbol{\omega} \\ -g\mathbf{e}_3 + m^{-1}\mathbf{R}(\boldsymbol{\delta})\mathbf{f}_c \\ \mathbf{J}^{-1}(-\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \boldsymbol{\tau}_c) \end{bmatrix} \tag{3.9}$$

where:

$$\mathbf{R}(\boldsymbol{\delta}) = \mathbf{R}_z(\psi)\mathbf{R}_y(\theta)\mathbf{R}_x(\phi), \quad \mathbf{T}(\boldsymbol{\delta}) = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} \tag{3.10}$$

3.2. Original HC Architecture

In this section, the original architerture of the hierarchical controller from [14] is presented.

Hierarchical Controller implements a state feedback nonlinear regulation, exploiting some algebraic prerequisites on the control matrices that ensure the existence of a preferential direction in the feasible force space, along which the control force and control moment are decoupled. As shown in Figure 3.3, which illustrates the architecture of the Hierarchical Controller, the control inputs are the *reference position* $\mathbf{p}_r \in \mathbb{R}^3$, its derivatives and the *reference orientation* $\mathbf{q}_r \in \mathbb{S}^3$. At first, the controller is designed to regulate the translational dynamics, aiming to counteract gravity while ensuring accurate position tracking. In this phase, an arbitrary desired orientation $\mathbf{q}_d \in \mathbb{S}^3$ is computed internally by the controller. Subsequently, a corrective action is applied to steer $\mathbf{q}_d$ towards the reference orientation $\mathbf{q}_r$, which justifies the term "Hierarchical".

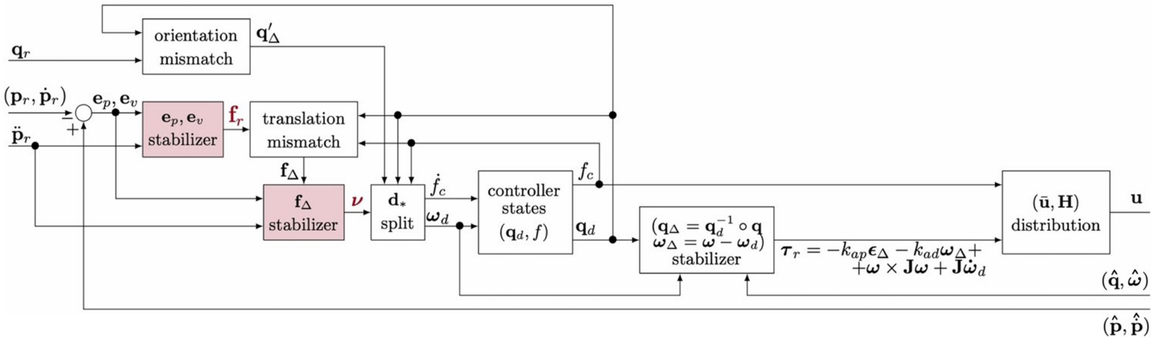


Figure 3.3: Hierarchical Controller Architecture, from [14]

As previously mentioned, the functioning of such a nonlinear controller is based on specific properties that the control matrices **F** and **M**, introduced in (3.6) and (3.7), must satisfy. As a preliminary requirement, both matrices must have full rank, a condition that is always met in the case of a tilted hexarotor when suitable tilt angles are selected. Furthermore, the matrices **F** and **M** must satisfy the following condition:

$$\mathrm{rk}(\mathbf{M}\overline{\mathbf{F}}) = 3 \tag{3.11}$$

where $\overline{\mathbf{F}} \in \mathbb{R}^{6\times 3}$ such that $\mathrm{Im}(\overline{\mathbf{F}}) = \ker(\mathbf{F})$.
This condition ensures the possibility of freely assigning the control moment $\boldsymbol{\tau}_c$. This ensures that the hexarotor can reject torque disturbances in any direction. Moreover, this condition implies the existence of a vector $\bar{\mathbf{u}} \in \ker(\mathbf{M})$ such that $\|\mathbf{F}\bar{\mathbf{u}}\| = 1$. This allows identifying a direction in the force space along which the magnitude of the control force can be arbitrarily assigned when the control moment is zero. The definition of such a *zero-moment preferential direction* is the following:

$$\mathbf{d}_* := \mathbf{F}\bar{\mathbf{u}} \in \mathrm{Im}(\mathbf{F}) \cap \mathbb{S}^2 \tag{3.12}$$

From a mathematical perspective, condition (3.11) guarantees the existence of a matrix $\mathbf{K} \in \mathbb{R}^{6\times 6}$ such that $\mathbf{M}\mathbf{K}\mathbf{M}^\mathsf{T}$ is invertible and the *generalized right pseudo-inverse of* **M** can be defined as: $\mathbf{M}_\mathbf{K}^\dagger = \mathbf{K}\mathbf{M}^\mathsf{T}(\mathbf{M}\mathbf{K}\mathbf{M}^\mathsf{T})^{-1} \in \mathbb{R}^{6\times 3}$.
By choosing the control law as:

$$\mathbf{u} = \mathbf{M}_\mathbf{K}^\dagger \boldsymbol{\tau}_r + \bar{\mathbf{u}} f_c \tag{3.13}$$

where $\boldsymbol{\tau}_r \in \mathbb{R}^3$ is the reference moment and $f_c = \|\mathbf{f}_c\| \in \mathbb{R}$ is the force intensity described in (3.6), it becomes possible to rewrite (3.6) and (3.7) as follows:

$$\mathbf{f}_c = \mathbf{F}\mathbf{u} = \mathbf{d}_* f_c \tag{3.14}$$

$$\boldsymbol{\tau}_c = \mathbf{M}\mathbf{u} = \boldsymbol{\tau}_r \tag{3.15}$$

The vector $\overline{\mathbf{u}}$ in the control law is chosen so that the zero-moment preferential direction is:

$$\mathbf{d}_* = \frac{\mathbf{d}}{\|\mathbf{d}\|}, \qquad \mathbf{d} = \mathbf{R}(\mathbf{q}_r)^T(mg\mathbf{e}_3 + m\ddot{\mathbf{p}}_r) \tag{3.16}$$

in which $\mathbf{q}_r \in \mathbb{S}^3$ is the *reference orientation.* This choice ensures that the resulting control force $\mathbf{f}_c = \mathbf{d}_* f_c$ is oriented in the body frame $\mathscr{F}_B$ in order to counterbalance the gravity force while acting along the direction of the reference position trajectory. The control architecture is shown in Fig.3.3.

The controller states are the *control force intensity* f_c and the *desired orientation* $\mathbf{q}_d \in \mathbb{S}^3$, which is the rotation that ensure that the resulting force $\mathbf{R}(\mathbf{q_d})\mathbf{f}_c$ acting on the translational dynamic (Equation 3.8), in the direction of $\mathbf{d}_*$ because of (3.14) coincides with the following reference force $\mathbf{f}_r \in \mathbb{R}^3$, chosen as a simple PD function with gravity compensation and an additional term for trajectory tracking, the expression is given by:

$$\mathbf{f}_r := mg\mathbf{e}_3 + m\ddot{\mathbf{p}}_r - k_{pp}\mathbf{e}_p - k_{pd}\mathbf{e}_v \tag{3.17}$$

where $\mathbf{e}_p = \mathbf{p} - \mathbf{p}_r \in \mathbb{R}^3$ and $\mathbf{e}_v = \mathbf{v} - \mathbf{v}_r \in \mathbb{R}^3$ represent the position and velocity error vectors, and $k_{pp}, k_{pd} \in \mathbb{R}$ are tunable positive scalar gains. It is possible to introduce the mismatch between forces, defined as:

$$\mathbf{f}_\Delta := \mathbf{R}(\mathbf{q}_d)\mathbf{f}_c - \mathbf{f}_r = \mathbf{R}(\mathbf{q}_d)\mathbf{d}_* f_c - \mathbf{f}_r \in \mathbb{R}^3 \tag{3.18}$$

With an appropriate choice of parameters f_c and $\mathbf{q}_d$, the mismatch $\mathbf{f}_\Delta$ converges to zero, thereby ensuring stabilization of the translational dynamics of the platform. Furthermore, instead of algebraically computing $\mathbf{q}_d$, its evolution is exploited through the following:

$$\dot{\mathbf{q}}_d = \frac{1}{2}\mathbf{q}_d \circ \begin{bmatrix} 0 \\ \boldsymbol{\omega}_d \end{bmatrix} \tag{3.19}$$

where $\boldsymbol{\omega}_d \in \mathbb{R}^3$ is an additional input selected to ensure $\mathbf{f}_\Delta$ converges to zero.

For parameter definition, the following Lyapunov function is chosen:

$$V_{f_\Delta} = \frac{1}{2}\mathbf{f}_\Delta^\top \mathbf{f}_\Delta \tag{3.20}$$

whose derivative is:

$$\dot{V}_{f_\Delta} = \mathbf{f}_\Delta^\top \dot{\mathbf{f}}_\Delta \tag{3.21}$$

To ensure that the mismatch $\mathbf{f}_\Delta$ converges to zero, according to Lyapunov's stability theory, the derivative of the Lyapunov function must necessarily be negative. Imposing the following equality:

$$\dot{V}_{f_\Delta} = \mathbf{f}_\Delta^\top \dot{\mathbf{f}}_\Delta := -k_\Delta \|\mathbf{f}_\Delta\|^2 \tag{3.22}$$

it follows that:

$$\dot{\mathbf{f}}_\Delta = -k_\Delta \mathbf{f}_\Delta \tag{3.23}$$

Equation (3.23) therefore represents the form that the derivative of $\mathbf{f}_\Delta$ must necessarily take in order to ensure the negativity of the time derivative of the defined Lyapunov function.

Computing this derivative from (3.18) (as reported in [14]), we obtain:

$$\begin{aligned} \dot{\mathbf{f}}_\Delta &= \mathbf{R}(\mathbf{q}_d)\mathbf{d}_*\dot{f}_c + \dot{\mathbf{R}}(\mathbf{q}_d)\mathbf{d}_* f_c - \dot{\mathbf{f}}_r \\ &= \mathbf{R}(\mathbf{q}_d)\mathbf{d}_*\dot{f}_c + \mathbf{R}(\mathbf{q}_d)[\boldsymbol{\omega}_d]_\times \mathbf{d}_* f_c - \dot{\mathbf{f}}_r \end{aligned} \tag{3.24}$$

Introducing the variable $\boldsymbol{\nu}$, which depends on the controller states:

$$\boldsymbol{\nu} = \mathbf{R}(\mathbf{q}_d)[\mathbf{d}_*\dot{f}_c + [\boldsymbol{\omega}_d]_\times \mathbf{d}_* f_c] \tag{3.25}$$

the derivative of the $\mathbf{f}_\Delta$ can be expressed as the following difference:

$$\dot{\mathbf{f}}_\Delta = \boldsymbol{\nu} - \dot{\mathbf{f}}_r \tag{3.26}$$

Proceeding with the computation of $\dot{\mathbf{f}}_r$, accounting for its definition (3.17):

$$\dot{\mathbf{f}}_r = -k_{pp}\dot{\mathbf{e}}_p - k_{pd}\dot{\mathbf{e}}_v + m\,\dddot{\mathbf{p}}_r \tag{3.27}$$

The positon error dynamic is given by:

$$\dot{\mathbf{e}}_p = \dot{\mathbf{p}} - \dot{\mathbf{p}}_r = \mathbf{e}_v \tag{3.28}$$

Instead, the velocity error dynamic is computed as follows:

$$\begin{aligned} \dot{\mathbf{e}}_v &= \dot{\mathbf{v}} - \dot{\mathbf{v}}_r \\ &= \frac{1}{m}(m\dot{\mathbf{v}} - m\dot{\mathbf{v}}_r) = \\ &= \frac{1}{m}(-mg\mathbf{e}_3 + \mathbf{R}(\mathbf{q})\mathbf{d}_* f_c - m\ddot{\mathbf{p}}_r - \mathbf{f}_\Delta + \mathbf{f}_\Delta) = \\ &= \frac{1}{m}(-mg\mathbf{e}_3 + \mathbf{R}(\mathbf{q})\mathbf{d}_* f_c - m\ddot{\mathbf{p}}_r - \mathbf{R}(\mathbf{q})\mathbf{d}_* f_c + \mathbf{f}_r + \mathbf{f}_\Delta) = \\ &= \frac{1}{m}((\mathbf{R}(\mathbf{q}) - \mathbf{R}(\mathbf{q}_d))\mathbf{d}_* f_c - k_{pp}\mathbf{e}_p - k_{pd}\mathbf{e}_v + \mathbf{f}_\Delta) \end{aligned} \tag{3.29}$$

By substituting results (3.28) and (3.29) into equation (3.27), the derivative of the reference force $\mathbf{f}_r$ becomes:

$$\dot{\mathbf{f}}_r = -k_{pp}\dot{\mathbf{e}}_v - \frac{k_{pd}}{m}\left((\mathbf{R}(\mathbf{q}) - \mathbf{R}(\mathbf{q}_d))\mathbf{d}_* f_c - k_{pp}\mathbf{e}_p - k_{pd}\mathbf{e}_v + \mathbf{f}_\Delta\right) + m\ddddot{\mathbf{p}}_r \tag{3.30}$$

This derivative simplifies when the platform attitude $\mathbf{q}$ converged toward the desired one $\mathbf{q}_d$. When this condition is met, it holds that:

$$\dot{\mathbf{f}}_r \approx \frac{k_{pd}k_{pp}}{m}\mathbf{e}_p + \left(\frac{{k_{pd}}^2}{m} - k_{pp}\right)\mathbf{e}_\mathbf{v} - \frac{k_{pd}}{m}\mathbf{f}_{\boldsymbol{\Delta}} + m\,\ddddot{\mathbf{p}}_r \tag{3.31}$$

Exploiting (3.31), the equation (3.26) results:

$$\dot{\mathbf{f}}_\Delta = \boldsymbol{\nu} - \frac{k_{pd}k_{pp}}{m}\mathbf{e}_p - \left(\frac{{k_{pd}}^2}{m} - k_{pp}\right)\mathbf{e}_\mathbf{v} + \frac{k_{pd}}{m}\mathbf{f}_{\boldsymbol{\Delta}} - m\,\ddddot{\mathbf{p}}_r \tag{3.32}$$

By imposing that $\dot{\mathbf{f}}_\Delta = -k_\Delta \mathbf{f}_\Delta$, the variable $\boldsymbol{\nu}$ results

$$\boldsymbol{\nu} = \frac{k_{pd}k_{pp}}{m}\mathbf{e}_p + \left(\frac{{k_{pd}}^2}{m} - k_{pp}\right)\mathbf{e}_\mathbf{v} - \left(\frac{k_{pd}}{m} + k_\Delta\right)\mathbf{f}_{\boldsymbol{\Delta}} + m\,\ddddot{\mathbf{p}}_r \tag{3.33}$$

with $k_\Delta \geq 0 \in \mathbb{R}$ being an additional tunable scalar gain.
In (3.33), the last term is highlighted in red to emphasize that its sign is opposite to that erroneously reported in [14].
Recalling that the variable $\boldsymbol{\nu}$ depends on the controller states as defined in (3.25), these states are given by:

$$\dot{f}_c = (\mathbf{R}(\mathbf{q}_d)\mathbf{d}_*)^\top \boldsymbol{\nu} \tag{3.34}$$

$$\boldsymbol{\omega}_d = \frac{1}{f_c}[\mathbf{d}_*]_\times \mathbf{R}(\mathbf{q}_d)^\top \boldsymbol{\nu} \tag{3.35}$$

As can be observed, the controller states in (3.34) and (3.35) themselves depend on the variable $\boldsymbol{\nu}$ defined in (3.33). This definition of the controller states was chosen to ensure that the following identity is satisfied:

$$\begin{aligned}\boldsymbol{\nu} &= \mathbf{R}(\mathbf{q}_d)\mathbf{d}_*\dot{f}_c - \mathbf{R}(\mathbf{q}_d)[\mathbf{d}_*]_\times \boldsymbol{\omega}_d f_c = \\ &= (\mathbf{R}(\mathbf{q}_d)\mathbf{d}_*)(\mathbf{R}(\mathbf{q}_d)\mathbf{d}_*)^\top \boldsymbol{\nu} - \mathbf{R}(\mathbf{q}_d)[\mathbf{d}_*]_\times f_c \frac{1}{f_c}[\mathbf{d}_*]_\times \mathbf{R}(\mathbf{q}_d)^\top \boldsymbol{\nu} = \\ &\overset{1}{=} \boldsymbol{\nu}((\mathbf{R}(\mathbf{q}_d)\mathbf{d}_*)(\mathbf{R}(\mathbf{q}_d)\mathbf{d}_*)^\top - \mathbf{R}(\mathbf{q}_d)[\mathbf{d}_*]_\times [\mathbf{d}_*]_\times \mathbf{R}(\mathbf{q}_d)^\top) = \\ &= \boldsymbol{\nu}\|\mathbf{R}(\mathbf{q}_d)\mathbf{d}_*\|^2 \mathbf{I} = \\ &\overset{2}{=} \boldsymbol{\nu}\|\mathbf{d}_*\|^2 = \\ &= \boldsymbol{\nu}\end{aligned}$$

In the first version of the Hierarchical Controller, equation (3.35) solely ensured the convergence of the platform orientation $\mathbf{q}$ towards $\mathbf{q}_d$. Subsequently, the second term $\boldsymbol{\omega}'_d \in \mathbb{R}^3$ (see Remark 3 in [13]) was added to regulate the controller state dynamics $\mathbf{q}_d$ towards $\mathbf{q}_r$.

$$\boldsymbol{\omega}_d = \boldsymbol{\omega}^0_d + \boldsymbol{\omega}'_d = \frac{1}{f_c}[\mathbf{d}_*]_\times \mathbf{R}(\mathbf{q}_d)^\top \boldsymbol{\nu} - k_q \mathbf{d}_* \mathbf{d}_*^\top \boldsymbol{\epsilon}'_\Delta \tag{3.36}$$

The term $\boldsymbol{\omega}'_d$ acts on the following orientation mismatch:

$$\mathbf{q}'_\Delta = \mathbf{q}_r^{-1} \circ \mathbf{q}_d = \begin{bmatrix} \eta'_\Delta \\ \boldsymbol{\epsilon}'_\Delta \end{bmatrix} \tag{3.37}$$

where η'_Δ is the scalar part and $\boldsymbol{\epsilon}'_\Delta$ is the vectorial part of the quaternion mismatch $\mathbf{q}'_\Delta$. The convergence to zero of $\mathbf{q}'_\Delta$ was demonstrated according to Lyapunov's stability theory. Furthermore, it was proven that its introduction does not affect the stability proof carried out previously.

The HC architecture, shown in Figure 3.3, is completed with a careful selection of $\boldsymbol{\tau}_r$ in (3.13), ensuring the convergence of $\mathbf{q}$ towards $\mathbf{q}_d$. This choice was made by introducing the following orientation mismatch:

$$\mathbf{q}_\Delta = \mathbf{q}_d^{-1} \circ \mathbf{q} = \begin{bmatrix} \eta_\Delta \\ \boldsymbol{\epsilon}_\Delta \end{bmatrix} \tag{3.38}$$

whose evolution is:

$$\dot{\mathbf{q}}_\Delta = \frac{1}{2}\mathbf{q}_\Delta \circ \begin{bmatrix} 0 \\ \boldsymbol{\omega}_\Delta \end{bmatrix} \tag{3.39}$$

where $\boldsymbol{\omega}_\Delta = \boldsymbol{\omega} - \boldsymbol{\omega}_d \in \mathbb{R}^3$ is the angular velocity mismatch.

---

[1] $\|\mathbf{x}^2\|\mathbf{I} = \mathbf{x}\mathbf{x}^\top - [\mathbf{x}]^2_\times$

[2] $\|\mathbf{R}(\mathbf{q}_d)\mathbf{d}_*\|^2 = (\mathbf{R}(\mathbf{q}_d)\mathbf{d}_*)^\top(\mathbf{R}(\mathbf{q}_d)\mathbf{d}_*) = \mathbf{d}_*^\top \mathbf{R}(\mathbf{q}_d)^\top \mathbf{R}(\mathbf{q}_d)\mathbf{d}_* = \mathbf{d}_*^\top \mathbf{d}_* = \|\mathbf{d}_*\|^2$
Ortogonality: $\mathbf{R}(\mathbf{q}_d)^\top \mathbf{R}(\mathbf{q}_d) = \mathbf{I}$

As done for the translational dynamics, a Lyapunov function was constructed, and its derivative was ensured to be negative through the following choice of $\boldsymbol{\tau}_r$:

$$\boldsymbol{\tau}_r = -k_{ap}\boldsymbol{\epsilon}_\Delta - k_{ad}\boldsymbol{\omega}_\Delta + \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \mathbf{J}\dot{\boldsymbol{\omega}}_d \tag{3.40}$$

The designed controller prioritized the regulation of the UAV orientation towards the desired one (internal controller state), rather than the reference one (external controller input), thus highlighting the lower priority assigned to attitude regulation in the hierarchical architecture. The HC architecture requires the tuning of the six scalar gains k_{pp}, k_{pd}, k_{ap}, k_{ad}, k_Δ, k_q, allowing an easier tuning process, but being a sophisticated control technique implies more difficulties in implementation.

3.3. Revised HC Architecture

With the aim of improving the performance of the Hierarchical Controller, a thorough analysis of the various versions of this control solution (presented in [12], [13], [14]) was conducted. Below, an enhancement of the architecture is proposed following the identification of a mathematical issue in Remark 3 in [14]. For the sake of clarity, equations (3.18) and (3.24), which describe the mismatch related to translational forces and its evolution, are reported below:

$$\mathbf{f}_\Delta := \mathbf{R}(\mathbf{q}_d)\mathbf{f}_c - \mathbf{f}_r = \mathbf{R}(\mathbf{q}_d)\mathbf{d}_* f_c - \mathbf{f}_r$$
$$\dot{\mathbf{f}}_\Delta = \mathbf{R}(\mathbf{q}_d)\mathbf{d}_* \dot{f}_c + \dot{\mathbf{R}}(\mathbf{q}_d)\mathbf{d}_* f_c - \dot{\mathbf{f}}_r$$

It is possible to notice the lack of a term in the derivative of $\mathbf{f}_\Delta$, associated with the derivative of the zero-moment preferential direction $\mathbf{d}_*$.

Rewriting the expression for the zero-moment preferential direction:

$$\mathbf{d}_* = \frac{\mathbf{d}}{\|\mathbf{d}\|}, \text{ with } \mathbf{d} = \mathbf{R}(\mathbf{q}_r)^\top (mg\mathbf{e}_3 + m\ddot{\mathbf{p}}_r)$$

and applying the rule for the derivative of a quotient:

$$\dot{\mathbf{d}}_* = \frac{\dot{\mathbf{d}}\|\mathbf{d}\| - \mathbf{d}\frac{d}{dt}\|\mathbf{d}\|}{\|\mathbf{d}\|^2} \tag{3.41}$$

it is possible to compute the required derivative.

The unknown derivative terms that must be evaluated in order to apply equation (3.41) are two. These terms are computed separately to enhance the clarity of the exposition.

Starting with the computation of the term $\dot{\mathbf{d}}$:

$$\begin{aligned}\dot{\mathbf{d}} &= \frac{d}{dt}(\mathbf{R}(\mathbf{q_r})^\top(mg\mathbf{e_3} + m\ddot{\mathbf{p}}_r)) = \dot{\mathbf{R}}(\mathbf{q_r})^\top(mg\mathbf{e_3} + m\ddot{\mathbf{p}}_r) + \mathbf{R}(\mathbf{q_r})^\top m\,\dddot{\mathbf{p}}_r = \\ &= -[\boldsymbol{\omega}_r]_\times \mathbf{R}(\mathbf{q_r})^\top(mg\mathbf{e_3} + m\ddot{\mathbf{p}}_\mathbf{r}) + \mathbf{R}(\mathbf{q_r})^\top m\,\dddot{\mathbf{p}}_r = \\ &= -[\boldsymbol{\omega}_r]_\times \mathbf{d} + \mathbf{R}(\mathbf{q_r})^\top m\,\dddot{\mathbf{p}}_r\end{aligned} \tag{3.42}$$

Subsequently, the second derivative term in (3.41) is evaluated. Calculating $\frac{d}{dt}\|\mathbf{d}\|$ and recalling that $\|\mathbf{d}\| = \sqrt{\mathbf{d}^T\mathbf{d}}$:

$$\begin{aligned}\frac{d}{dt}\|\mathbf{d}\| &= \frac{1}{2\|\mathbf{d}\|}\frac{d}{dt}(\mathbf{d}^\top\mathbf{d}) = \frac{1}{2\|\mathbf{d}\|}2\mathbf{d}^\top\dot{\mathbf{d}} = \frac{\mathbf{d}^\top\dot{\mathbf{d}}}{\|\mathbf{d}\|} = \\ &= \frac{\mathbf{d}^\top}{\|\mathbf{d}\|}(-[\boldsymbol{\omega}_\mathbf{r}]_\times\mathbf{d} + \mathbf{R}(\mathbf{q_r})^\top m\,\dddot{\mathbf{p}}_r)\end{aligned} \tag{3.43}$$

Substituting the results (3.42), (3.43) into (3.41), we obtain:

$$\dot{\mathbf{d}}_* = \frac{(-[\omega_\mathbf{r}]_\times\mathbf{d} + \mathbf{R}(\mathbf{q_r})^\top m\,\dddot{\mathbf{p}}_r)\|\mathbf{d}\| - \frac{\mathbf{d}\mathbf{d}^\top}{\|\mathbf{d}\|}(-[\omega_\mathbf{r}]_\times\mathbf{d} + \mathbf{R}(\mathbf{q_r})^\top m\,\dddot{\mathbf{p}}_r)}{\|\mathbf{d}\|^2}$$

Defining $\mathbf{P_d} = \mathbf{d}\mathbf{d}^\top/\|\mathbf{d}\|^2$, the derivative of the zero-moment preferential direction is given as:

$$\dot{\mathbf{d}}_* = \frac{(\mathbf{I} - \mathbf{P_d})}{\|\mathbf{d}\|}(-[\omega_\mathbf{r}]_\times\mathbf{d} + \mathbf{R}(\mathbf{q_r})^\top m\,\dddot{\mathbf{p}}_r) = \frac{(\mathbf{I} - \mathbf{P_d})}{\|\mathbf{d}\|}\dot{\mathbf{d}} \tag{3.44}$$

After computing the derivative of $\mathbf{d}_*$, it is possible to modify the derivative of $\mathbf{f}_\Delta$ as anticipated:

$$\begin{aligned}\dot{\mathbf{f}}_\Delta &= \dot{\mathbf{R}}(\mathbf{q}_d)\mathbf{d}_* f_c + \mathbf{R}(\mathbf{q}_d)\dot{\mathbf{d}}_* f_c + \mathbf{R}(\mathbf{q}_d)\mathbf{d}_*\dot{f}_c - \dot{\mathbf{f}}_r = \\ &= \dot{\mathbf{R}}(\mathbf{q}_d)\mathbf{d}_* f_c + \mathbf{R}(\mathbf{q}_d)\frac{(\mathbf{I} - \mathbf{P_d})}{\|\mathbf{d}\|}\dot{\mathbf{d}} f_c + \mathbf{R}(\mathbf{q}_d)\mathbf{d}_*\dot{f}_c - \dot{\mathbf{f}}_r\end{aligned} \tag{3.45}$$

where:

$$\dot{\mathbf{d}} = -[\omega_\mathbf{r}]_\times\mathbf{d} + \mathbf{R}(\mathbf{q}_r)^\top m\,\dddot{\mathbf{p}}_r \tag{3.46}$$

$$\mathbf{P_d} = \frac{\mathbf{d}\mathbf{d}^\top}{\|\mathbf{d}\|} \tag{3.47}$$

Recalling the required form of the time derivative of the mismatch between the total force acting on the translational dynamics and the reference force introduced in (3.17), which ensures the negativity of the derivative of the Lyapunov function defined in (3.20):

$$\dot{\mathbf{f}}_\Delta = -k_\Delta \mathbf{f}_\Delta$$

with the inclusion of the missing term, equation (3.45) then becomes:

$$\dot{\mathbf{f}}_\Delta = \boldsymbol{\nu} + \mathbf{R}(\mathbf{q}_d)\frac{(\mathbf{I}-\mathbf{P_d})}{\|\mathbf{d}\|}\dot{\mathbf{d}}f_c - \dot{\mathbf{f}}_r := -k_\Delta \mathbf{f}_\Delta \tag{3.48}$$

Through equation (3.48), a new definition of the parameter $\boldsymbol{\nu}$, previously introduced in (3.33), is obtained:

$$\boldsymbol{\nu} = \frac{k_{pd}k_{pp}}{m}\mathbf{e}_p + \left(\frac{{k_{pd}}^2}{m} - k_{pp}\right)\mathbf{e_v} - \left(\frac{k_{pd}}{m} + k_\Delta\right)\mathbf{f_\Delta} + m\,\dddot{\mathbf{p}}_r - \mathbf{R}(\mathbf{q}_d)\frac{(\mathbf{I}-\mathbf{P_d})}{\|\mathbf{d}\|}\dot{\mathbf{d}}f_c \tag{3.49}$$

For the sake of completeness, the controller states are reported below, which will be modified according to the new definition of the parameter $\boldsymbol{\nu}$:

$$\dot{f}_c = (\mathbf{R}(\mathbf{q}_d)\mathbf{d}_*)^\top \boldsymbol{\nu}$$

$$\boldsymbol{\omega}_d = \boldsymbol{\omega}_d^0 + \boldsymbol{\omega}_d' = \frac{1}{f_c}[\mathbf{d}_*]_\times \mathbf{R}(\mathbf{q}_d)^\top \boldsymbol{\nu} - k_q \mathbf{d}_* \mathbf{d}_*^\top \boldsymbol{\epsilon}_\Delta'$$

3.4. Improved HC Architecture

This section proposes the addition of an integral term to the reference force $\mathbf{f}_r$, introduced in (3.17). A step-by-step derivation of this modification is provided to demonstrate that the inclusion of such term does not affect the stability of the control system, and that the negativity of the previously discussed Lyapunov function is still guaranteed. The new expression of the reference force is thus presented as follows:

$$\mathbf{f}_r := mg\mathbf{e}_3 + m\ddot{\mathbf{p}}_r - k_{pp}\mathbf{e}_p - k_{pd}\mathbf{e}_v - k_{pi}\int \mathbf{e}_p\ dt \tag{3.50}$$

As for the previous considerations regarding the derivative of the mismatch $\mathbf{f}_\Delta$, they remain unchanged. In order to ensure the negativity of the Lyapunov function, the time derivative of this mismatch must always take the following form:

$$\dot{\mathbf{f}}_\Delta = -k_\Delta \mathbf{f}_\Delta$$

Therefore, the equation already derived in (3.48), and reported below, still holds:

$$\dot{\mathbf{f}}_{\Delta} = \boldsymbol{\nu} + \mathbf{R}(\mathbf{q}_d)\frac{(\mathbf{I}-\mathbf{P}_{\mathbf{d}})}{\|\mathbf{d}\|}\dot{\mathbf{d}}f_c - \dot{\mathbf{f}}_r := -k_{\Delta}\mathbf{f}_{\Delta} \tag{3.51}$$

What changes is precisely the derivative of $\mathbf{f}_r$ reported in this equation. The new expression of the reference force derivative is computed as follows:

$$\dot{\mathbf{f}}_r = -k_{pp}\dot{\mathbf{e}}_p - k_{pd}\dot{\mathbf{e}}_v + m\,\dddot{\mathbf{p}}_r - k_{pi}\mathbf{e}_p \tag{3.52}$$

Recalling that the reference force also appears in the velocity error dynamics $\mathbf{e}_v$ (see equation 3.29), the new expression of the velocity error dynamics becomes:

$$\dot{\mathbf{e}}_v = \frac{1}{m}[(\mathbf{R}(\mathbf{q}) - \mathbf{R}(\mathbf{q}_d))\mathbf{d}_* f - k_{pp}\mathbf{e}_p - k_{pd}\mathbf{e}_v + \mathbf{f}_{\Delta} - k_{pi}\int \mathbf{e}_p\ dt] \tag{3.53}$$

so the derivative of the reference force becomes:

$$\dot{\mathbf{f}}_r = \left(\frac{k_{pd}k_{pp}}{m} - k_{pi}\right)\mathbf{e}_p + \left(\frac{{k_{pd}}^2}{m} - k_{pp}\right)\mathbf{e}_v - \frac{k_{pd}}{m}\mathbf{f}_{\Delta} + m\,\dddot{\mathbf{p}}_r + \frac{k_{pd}k_{pi}}{m}\int \mathbf{e}_p\ dt \tag{3.54}$$

Therefore, by substituting equation (3.54) into equation (3.51), the ultimate definition of the parameter $\boldsymbol{\nu}$ is obtained, which is referred to as $\boldsymbol{\nu}_f$ for compactness:

$$\boldsymbol{\nu}_f = \boldsymbol{\nu} + \frac{k_{pd}k_{pi}}{m}\int \mathbf{e}_p\ dt - k_{pi}\mathbf{e}_p \tag{3.55}$$

in which $\boldsymbol{\nu}$ is the version defined in (3.49). For the sake of completeness, the controller states are reported below, which will be modified according to the new definition of the parameter $\boldsymbol{\nu}$:

$$\dot{f}_c = (\mathbf{R}(\mathbf{q}_d)\mathbf{d}_*)^{\top}\boldsymbol{\nu}_f$$

$$\boldsymbol{\omega}_d = \boldsymbol{\omega}_d^0 + \boldsymbol{\omega}_d' = \frac{1}{f_c}[\mathbf{d}_*]_{\times}\mathbf{R}(\mathbf{q}_d)^{\top}\boldsymbol{\nu}_f - k_q\mathbf{d}_*\mathbf{d}_*^{\top}\boldsymbol{\epsilon}_{\Delta}'$$

4. IMPROVED HC PERFORMACE ASSESSMENT

This chapter represents the core validation phase of the thesis, where the performance of the improved HC is thoroughly evaluated. The customized real-world hexarotor platform used to assess the control strategy is first introduced. To ensure an objective evaluation, alternative state-of-the-art control solutions are then presented, together with the Key Performance Indices (KPI) selected for the comparison. The chapter concludes with a detailed discussion and analysis of the results obtained from the various test campaigns conducted both in the MATLAB/Simulink[1] numerical simulation environment and in the Gazebo[2] physics-based simulation environment.

4.1. Reference Hexarotor Platform

The hexarotor used as a reference platform to validate the performance of the improved version of the Hierachical Controller is the custom-built system shown in Figure 4.1. This hexarotor, identified by the name HR01, features a star-shaped configuration, with six rotors symmetrically arranged around the center of mass on a carbon fiber frame **Tarot 680 Pro**, with an overall diameter of approximately 0.8 meters. The total weight, including all hardware components and sensors, amounts to approximately 3.5 kg. Propulsion is provided by six **Tarot 4008 6S** brushless motors rated at 380 kV, each paired with a 35 A **Holybro Tekko 32 ESC**, driving 13-inch carbon fiber propellers with a pitch of 5.5 inches. In addition, HR01 is equipped with a **Pixhawk 4 flight controller** and a **Raspberry Pi 4B**, housed in a lightweight aluminum case

---

[1] MATLAB® (MATrix LABoratory) is a programming environment and numerical computing language developed by MathWorks. It is used for numerical simulations, data analysis, and solving complex mathematical problems.
Simulink® is a graphical programming environment integrated in MATLAB®, specifically designed for modeling, analyzing, and simulating dynamical systems.

[2] Gazebo is an open-source 3D robotics simulator designed to accurately and efficiently simulate robots in complex indoor and outdoor environments. It provides a robust physics engine, high-quality graphics, and interfaces for sensor simulation.
It is widely used in research and development for testing robotic algorithms, enabling realistic simulation before real-world deployment.

with integrated fans. Onboard HR01 is also an **Intel RealSense Depth Camera D435**, capable of acquiring RGB images at a maximum resolution of 1920 × 1080 p, with a frame rate of 30 fps and a field of view (FOV) of 69° × 42°.

Figure 4.1: Star-shaped hexarotor used for validation, HR01

In this work, the tilting angles α_i and β_i with $i = 1, \dots, 6$ (see Fig. 3.2) of the i-th propeller are kept constant:

$$\alpha_i = (-1)^i \alpha \quad \text{and} \quad \beta_i = \beta$$

The values of α and β used are reported in Table 4.1. In the following, a table is provided that summarizes all the physical parameters of the hexarotor used in this study.

Parameter	**Definition**	**Value**	**Unit**
m	total mass of the hexarotor	3.5	$[kg]$
g	gravity constant	9.81	$[m/s^2]$
γ_i	angular direction of each hexarotor arm	$(i-1)\frac{\pi}{3}$	$[°]$
L_x	distance between $\mathbf{O_{P_i}}$ and $\mathbf{O_B}$	0.358	$[m]$
$\mathbf{J}$	inertia of hexarotor	$\begin{bmatrix} 0.155 & 0 & 0 \\ 0 & 0.147 & 0 \\ 0 & 0 & 0.251 \end{bmatrix}$	$[kg \cdot m^2]$
c_f	propeller thrust coefficient	0.0015	$[N/Hz^2]$
c_τ	propeller drag coefficient	$4.59 \cdot 10^{-5}$	$[Nm/Hz^2]$
u_{max}	max propellers speed	108	$[Hz]$
α	tilt angle about $\mathbf{O_{P_i}O_B}$	25	$[°]$
β	tilt angle about $\mathbf{Y_{P_i}}$	10	$[°]$

Table 4.1: Parameters of the hexarotor used in the study.

4.2. Benchmark for Validation

In order to evaluate the performance of the new HC controller version, the Geometric Controller (GC) [15] and the Flatness-based Controller (FC) [14] are described below. The Key Performance Indices used to make a comparision between the different control solutions are also presented.

4.2.1. Geometrical Controller (GC)

Geometric control offers an effective framework for managing multi-rotor UAVs, particularly those that are fully-actuated. By utilizing the geometry of the configuration space, this method enables direct regulation of both position and orientation on the manifold. This results in more intuitive and resilient performance when dealing with complex dynamics, nonlinear behavior, and large rotational movements, outperforming many conventional control strategies.

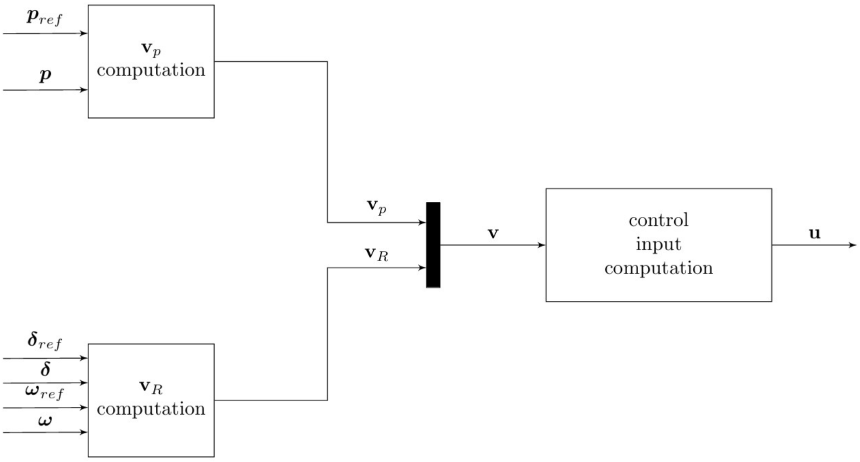


Figure 4.2: Geometrical Controller architecture.

Starting from the Hexarotor dynamic equations presented in Chapter 3, they can be rewritten as follows:

$$\begin{bmatrix} \ddot{\mathrm{p}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = \begin{bmatrix} -g\mathbf{e}_3 \\ \mathbf{J}^{-1}(-\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega}) \end{bmatrix} + \begin{bmatrix} \frac{1}{m}\mathbf{R}(\boldsymbol{\delta}) & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{J}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{F} \\ \mathbf{M} \end{bmatrix} = \mathbf{b} + \mathbf{K}_T\mathbf{u} \tag{4.1}$$

where $\mathbf{b} \in \mathbb{R}^6$ is the so-called *drift vector*, and $\mathbf{K}_T \in \mathbb{R}^{6\times6}$ is the *decoupling matrix.* Thanks to the platform's full actuation, this matrix is always invertible, thus allowing the control input to be chosen as:

$$\mathbf{u} = \mathbf{K}_T^{-1}\left(-\mathbf{b} + \begin{bmatrix}\ddot{\mathbf{p}} \\ \dot{\boldsymbol{\omega}}\end{bmatrix}\right) \tag{4.2}$$

Let us define $\mathbf{v} = \begin{bmatrix}\mathbf{v}_p \\ \mathbf{v}_R\end{bmatrix} = \begin{bmatrix}\ddot{\mathbf{p}} \\ \dot{\boldsymbol{\omega}}\end{bmatrix}$, to be consistent with the notation used in Figure (4.2), which illustrates the architecture of the Geometric Controller. The goal is to achieve convergence to zero of the position error, defined as:

$$\mathbf{e}_p = \mathbf{p} - \mathbf{p}_r \in \mathbb{R}^3 \tag{4.3}$$

where $\mathbf{p}$ is the actual position and $\mathbf{p}_r$ is the reference position.
Additionally, convergence to zero of the orientation error is desired, defined as:

$$\mathbf{e}_a = \frac{1}{2}[\mathbf{R}_{ref}^\top \mathbf{R}(\boldsymbol{\delta}) - \mathbf{R}(\boldsymbol{\delta})^\top \mathbf{R}_{ref}]_\text{v} \tag{4.4}$$

where $\mathbf{R}_{ref}$ is the reference orientation defined as a rotation matrix, and the operator $[\cdot]_\text{v}$ denotes the operation that maps a rotation matrix in $SO(3)$ to a vector in $\mathbb{R}^3$.
To ensure this, the first component of the vector $\mathbf{v}$, shown in Figure (4.2), can be chosen as:

$$\mathbf{v}_p = \ddot{\mathbf{p}}_r - \mathbf{K}_{p1}\mathbf{e}_p - \mathbf{K}_{p2}\dot{\mathbf{e}}_p - \mathbf{K}_{p3}\int \mathbf{e}_p \; dt \tag{4.5}$$

where $\mathbf{K}_{p1}$, $\mathbf{K}_{p2}$, and $\mathbf{K}_{p3} \in \mathbb{R}^{3\times3}$ are diagonal positive definite gain matrices.
The second component of $\mathbf{v}$ is chosen as:

$$\mathbf{v}_R = \dot{\boldsymbol{\omega}}_{ref} - \mathbf{K}_{R1}\mathbf{e}_a - \mathbf{K}_{R2}\dot{\mathbf{e}}_\omega - \mathbf{K}_{R3}\int \mathbf{e}_a \; dt \tag{4.6}$$

where $\mathbf{K}_{R1}$, $\mathbf{K}_{R2}$, and $\mathbf{K}_{R3} \in \mathbb{R}^{3\times3}$ are diagonal positive definite gain matrices, and $\dot{\mathbf{e}}_\omega$ is obtained from the following:

$$\mathbf{e}_\omega = \boldsymbol{\omega} - \mathbf{R}(\boldsymbol{\delta})^\top \mathbf{R}_{ref}\boldsymbol{\omega}_{ref} \tag{4.7}$$

which represents the angular velocity error.
Starting from the two components of $\mathbf{v}$, defined in equations (4.5) and (4.6), the full

vector can be rewritten as:

$$
\begin{aligned}
\mathbf{v} &= \begin{bmatrix} \ddot{\mathbf{p}}_r - \mathbf{K}_{p1}\dot{\mathbf{e}}_p - \mathbf{K}_{p2}\mathbf{e}_p - \mathbf{K}_{p3}\int \mathbf{e}_p\ dt \\ \dot{\boldsymbol{\omega}}_r - \mathbf{K}_{R1}\dot{\mathbf{e}}_\omega - \mathbf{K}_{R2}\mathbf{e}_a - \mathbf{K}_{R3}\int \mathbf{e}_a\ dt \end{bmatrix} = \\
&= \begin{bmatrix} \ddot{\mathbf{p}}_r \\ \dot{\boldsymbol{\omega}}_r \end{bmatrix} - \begin{bmatrix} \mathbf{K}_{p1} & \mathbf{K}_{p2} & \mathbf{K}_{p3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{K}_{R1} & \mathbf{K}_{R2} & \mathbf{K}_{R3} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{e}}_p \\ \mathbf{e}_p \\ \int \mathbf{e}_p\ dt \\ \dot{\mathbf{e}}_\omega \\ \mathbf{e}_a \\ \int \mathbf{e}_a\ dt \end{bmatrix} = \\
&= \mathbf{v}_r - \mathbf{K}_g\mathbf{e}
\end{aligned}
$$

Finally, the control input $\mathbf{u}$ can be computed from equation (4.2) as:

$$\mathbf{u} = \mathbf{K}_T^{-1}(-\mathbf{b} + \mathbf{v}_r - \mathbf{K}_g\mathbf{e}) \in \mathbb{R}^6 \tag{4.8}$$

where $\mathbf{v}_r \in \mathbb{R}^6$, $\mathbf{K}_g \in \mathbb{R}^{6\times18}$ and $\mathbf{e} \in \mathbb{R}^{18}$

4.2.2. Flatness-based Controller (FC)

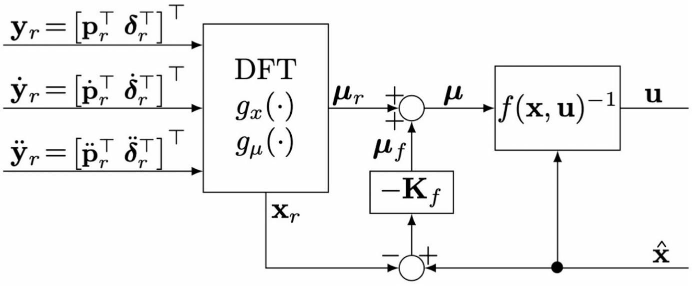


Figure 4.3: Flatness-based Controller architecture, from [14]

This control solution is based on the *flat differentially* property that a system can exhibit. A system is said to be *differentially flat* if it is possible to express its state and input as functions $g_x(\cdot)$, $g_\mu(\cdot)$, respectively, of a set of outputs and a finite number of their derivatives, named *flat outputs*. With a suitable choice of input, output, and state vectors, it is possible to demonstrate that a fully-actuated hexarotor can be modeled as a differentially flat system.

For a system that is differentially flat, denoting with $\mathbf{x} \in \mathbb{R}^n$, $\boldsymbol{\mu} \in \mathbb{R}^m$, $\mathbf{y} \in \mathbb{R}^m$ the state, input, and output vectors respectively, the following function can be identified:

$$\mathbf{x} = g_x(\mathbf{y}, \dot{\mathbf{y}}) \tag{4.9}$$

By choosing the state vector as $\mathbf{x}_p := [\mathbf{p} \quad \boldsymbol{\delta} \quad \mathbf{v} \quad \boldsymbol{\omega}]^\intercal \in \mathbb{R}^{12}$, the differential flatness of the tilted hexarotor is guaranteed and the model equations (described in Chapter 3) lead to the following linear state-space system:

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{x})\mathbf{x} + \mathbf{B}\boldsymbol{\mu} - \mathbf{g} \tag{4.10}$$

with matrices $\mathbf{A}(\mathbf{x}) \in \mathbb{R}^{12\times12}$, $\mathbf{B} \in \mathbb{R}^{12\times6}$ and $\mathbf{g} \in \mathbb{R}^{12}$ the gravity vector, defined as follows:

$$\mathbf{A}(\mathbf{x}) = \begin{bmatrix} \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{T}(\boldsymbol{\delta}) \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 \\ \frac{1}{m}\mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{J}^{-1} \end{bmatrix}, \quad \mathbf{g} = \begin{bmatrix} \mathbf{0}_3 \\ \mathbf{e}_3 \\ \mathbf{0}_3 \\ \mathbf{0}_3 \end{bmatrix} \tag{4.11}$$

In matrix $\mathbf{A}(\mathbf{x})$, the matrix $\mathbf{T}(\boldsymbol{\delta}) \in \mathbb{R}^3$, previously introduced in Chapter 3, depends on the Euler angles convention used. The inverse of $\mathbf{T}(\boldsymbol{\delta})$ links the angular velocity vector to the time derivative of the Euler angles:

$$\boldsymbol{\omega} = \mathbf{T}^{-1}(\boldsymbol{\delta})\dot{\boldsymbol{\delta}}, \quad \mathbf{T}^{-1}(\boldsymbol{\delta}) = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \cos\theta\sin\phi \\ 0 & -\sin\phi & \cos\theta\cos\phi \end{bmatrix} \tag{4.12}$$

The flat input $\boldsymbol{\mu}$ in (4.10) is connected to the input vector $\mathbf{u}$ as follows:

$$\boldsymbol{\mu} = f(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \mathbf{0} \\ -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} \end{bmatrix} + \begin{bmatrix} \mathbf{R}(\boldsymbol{\delta}) & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \mathbf{F} \\ \mathbf{M} \end{bmatrix} \mathbf{u} \tag{4.13}$$

Finally, the flat output vector can be defined by considering both the position and the orientation of the platform, selecting $\mathbf{y} = [\mathbf{y}_p \ \mathbf{y}_\delta] = [\mathbf{p}^\top \ \boldsymbol{\delta}^\top]^\top \in \mathbb{R}^6$, resulting:

$$\mathbf{y} = h(\mathbf{x}) = \mathbf{C}\mathbf{x} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_3 & \mathbf{0} \end{bmatrix} \mathbf{x} \tag{4.14}$$

The above considerations allow to verify the actual differential flatness of the system, since it is possible to express the state as follows:

$$\mathbf{x} = g_x(\mathbf{y}, \dot{\mathbf{y}}) = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_3 \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{y} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{W}(\boldsymbol{\delta})^{-1} \end{bmatrix} \dot{\mathbf{y}} \tag{4.15}$$

and the control input $\mathbf{u}$ in relation to the flat control input $\boldsymbol{\mu}$:

$$\mathbf{u} = \begin{bmatrix} \mathbf{F} \\ \mathbf{M} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{R}(\boldsymbol{\delta}) & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_3 \end{bmatrix}^{-1} \left(\boldsymbol{\mu} - \begin{bmatrix} \mathbf{0} \\ -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} \end{bmatrix} \right) = f(\mathbf{x}, \mathbf{u})^{-1} \tag{4.16}$$

Starting from the dynamic equations of the hexarotor model expressed in terms of rotation matrices, and omitting intermediate analytical steps, we obtain:

$$\mathbf{u} = \begin{bmatrix} \mathbf{F} \\ \mathbf{M} \end{bmatrix}^{-1} \begin{bmatrix} m\mathbf{R}(\boldsymbol{\delta})^{\top}(\ddot{\mathbf{p}} + g\mathbf{e}_3) \\ \mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} \end{bmatrix} \tag{4.17}$$

Considering this last equation and substituting into (4.13), the function $\boldsymbol{\mu} = g_{\mu}(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}})$ is defined as follows:

$$\begin{aligned} \boldsymbol{\mu} = g_{\mu}(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}}) &= \begin{bmatrix} \mathbf{0} \\ -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} \end{bmatrix} + \begin{bmatrix} \mathbf{R}(\boldsymbol{\delta}) & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \mathbf{F} \\ \mathbf{M} \end{bmatrix} \mathbf{u} = \\ &= \begin{bmatrix} \mathbf{0} \\ -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} \end{bmatrix} + \begin{bmatrix} \mathbf{R}(\boldsymbol{\delta}) & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \mathbf{F} \\ \mathbf{M} \end{bmatrix} \begin{bmatrix} \mathbf{F} \\ \mathbf{M} \end{bmatrix}^{-1} \begin{bmatrix} m\mathbf{R}(\boldsymbol{\delta})^{\top}(\ddot{\mathbf{p}} + g\mathbf{e}_3) \\ \mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} \end{bmatrix} = \\ &= \begin{bmatrix} \mathbf{0} \\ -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} \end{bmatrix} + \begin{bmatrix} m(\ddot{\mathbf{p}} + g\mathbf{e}_3) \\ \mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} \end{bmatrix} \end{aligned}$$

The FC Architecture is shown in Figure 4.3.
The *Differential Flatness Transformation* (DFT) block is responsible for computing $\boldsymbol{\mu}_r$ and the reference state $\mathbf{x}_r$, which is used to compute the feedback action $\boldsymbol{\mu}_f$. This feedback action is derived using an LQR approach on the linear system introduced in (4.10):

$$\boldsymbol{\mu}_f = -\mathbf{K}_f(\mathbf{x} - \mathbf{x}_r) \tag{4.18}$$

where $\mathbf{K}_f \in \mathbb{R}^{6\times 12}$ is the gain matrix. Finally, the resulting vector $\boldsymbol{\mu}$ is converted into the input vector $\mathbf{u}$ by exploiting Equation (4.16).

4.2.3. Key Performance Indices

To objectively assess the performance of the improved HC compared to the state-of-the-art GC and FC, the following performance indices are considered:

- *Position tracking error norm* $\|\mathbf{e}_p\| \in \mathbb{R}$
- *Attitude tracking error* $e_a = 2\arccos(\mathbf{q}^\top \mathbf{q}_r) \in \mathbb{R}$, computed as the Riemannian distance on $\mathbb{S}^3$
- *Control input norm* $\|\mathbf{u}\| \in \mathbb{R}$, which is used to tune the controllers by ensuring comparable energy consumption.
- *Mean Saturation Index* (MSI), defined as follows:

  $$MSI = \frac{1}{T}\int_0^T \frac{1}{n}\left(\sum_{i=1}^{n} \frac{u_i}{u_{\max}^2}\right)^2 dt$$

  This index is a scalar quantity ($0 < MSI < 1$) that quantifies, on average, how close the actuator commands are to their saturation limits. To provide a global measure for quick comparison, the average of the MSI values computed for each propeller is considered.
- *Root Mean Square Error* (RMSE) on each state variable computed as:

  $$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - {x_i}^{\text{ref}})^2} \in \mathbb{R}$$

  where x_i denotes the generic state variable on which the RMSE value is computed.

4.3. Numerical Validations

Figure 4.4: MATLAB/Simulink logo

In this section, the results of the tests carried out in the MATLAB/Simulink environment are presented and discussed. In detail, the Navigation Tasks selected for the evaluation are first introduced. The analysis then focuses on comparing the performance of the improved version of the Hierarchical Controller with the original architecture, before moving on to a comparison with the other control strategies described in Section 4.2.

4.3.1. Navigation Task Overview

With the aim of testing the performance of the improved Hierarchical Controller architecture, two different navigation tasks are considered. The first one (**NT01**) consist of tracking a circular trajectory with a radius of 2 meters at a constant altitude of 1 meter above the ground, which is used as a reference for position control. Simultaneously, the controller is required to stabilize the platform attitude by tracking step references for the three attitude angles. In particular, the reference ϕ_r follows the sequence $[-7°,\ 0°,\ 7°]$, the reference θ_r is $[0°,\ 3.5°,\ 7°]$, while the reference for ψ_r is designed to be more aggressive, varying in the range $[90°,\ 270°]$ with increasing steps of 45°. The NT01 implies the following initial condition for the UAV:

$$\begin{aligned}\mathbf{x}_0 &= \begin{bmatrix}\mathbf{p}^\top & \mathbf{v}^\top & \boldsymbol{\delta}^\top & \boldsymbol{\omega}^\top\end{bmatrix}^\top = \\ &= [2\ \ 0\ \ 1\ \ 0\ \ 0.6283\ \ 0\ \ -7\ \ 0\ \ 90\ \ 0\ \ 0\ \ 0]^\top\end{aligned}$$

The second navigation task (**NT02**) is designed by taking into account a potential real-world application. Considering a scenario in which the UAV must monitor a target using an onboard camera, the same circular position trajectory analyzed in the previous task is adopted, with an altitude of 1 meter. Regarding the angular variables, roll and pitch are kept constant at 0° and 7°, respectively, while yaw is time-varying. Specifically, the yaw angle follows a ramp with a constant slope equal to the angular velocity of the circular trajectory, so that the camera remains continuously oriented toward the object to be monitored. The initial conditions used for NT02 are set as follows:

$$\begin{aligned} \mathbf{x}_0 &= \left[\mathbf{p}^\top \quad \mathbf{v}^\top \quad \boldsymbol{\delta}^\top \quad \boldsymbol{\omega}^\top\right]^\top = \\ &= [2 \ \ 0 \ \ 1 \ \ 0 \ \ 0.6283 \ \ 0 \ \ 7 \ \ 0 \ \ 0 \ \ 0 \ \ 0 \ \ 0]^\top \end{aligned}$$

4.3.2. Original vs Improved HC Architecture

The first test conducted to validate the modifications and enhancements introduced in the architecture of the HC consists in comparing its performance with the original architecture, described in [14] and discussed in Section 3.2. This test was carried out considering **NT01** under ideal conditions. To ensure an objective comparison, the same gain values were used for both versions of the HC architectures, although it is important to note that the original version does not include an integral gain. The tuning parameters adopted are reported in Table 4.2

k_{pp}	k_{pd}	k_{ap}	k_{ad}	k_Δ	k_q	k_{pi}
264	25	25	10	6.5	200	200

Table 4.2: HC gains

Table 4.3 reports the Key Performance Indices (KPIs) obtained for the two different HC architectures using the implemented tuning.

	$\|\mathbf{u}\|$ $[Hz]$	$\|\mathbf{e}_p\|$ $[m]$	e_a [°]	MSI
$\mathbf{HC}_{old}$	104.9507	0.0405	0.2260	0.1491
$\mathbf{HC}_{new}$	104.9486	**0.0074**	0.2245	0.1491

Table 4.3: Performance indices, Improved vs Original HC

As can be observed, the energy consumption, synthesized in the norm of the control input, is identical for both controllers. This result is further confirmed by the identical MSI value obtained by the two control solutions. This highlights that the improvements introduced in the architecture did not affect the energy expenditure of the HC neither positively nor negatively. A rather interesting result is that the improved version of the HC controller exhibits a position error norm that is one order of magnitude lower than that of the previous version, confirming the effectiveness of the improvements introduced in its architecture. Regarding the attitude error, the two versions of the HC controller appear to exhibit similar behavior. Further insights are provided later through the analysis of the RMSE values for each state variable. Before that, the plots related to NT01 under ideal conditions for both versions of the HC architecture are presented (Figures 4.5 and 4.6).

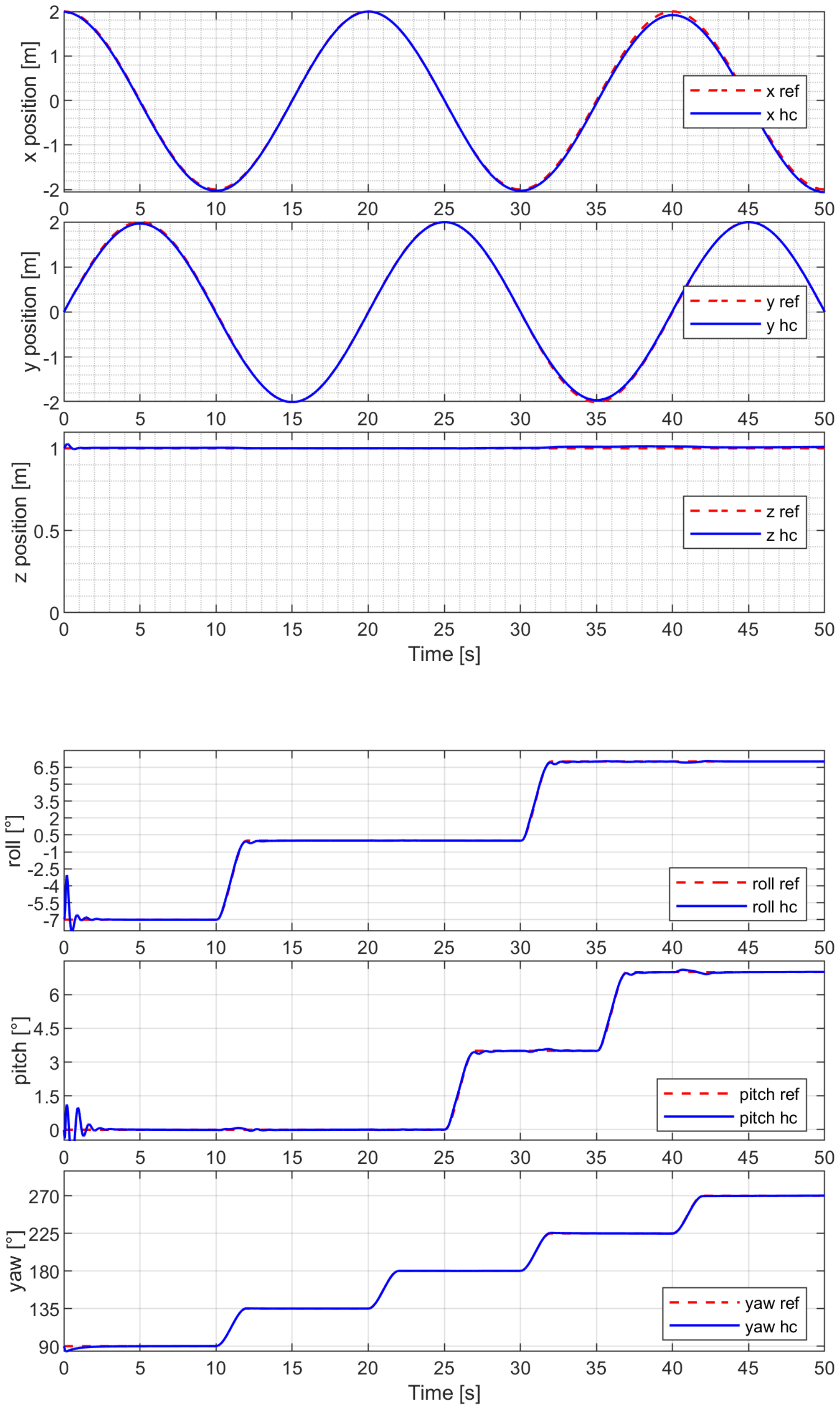


Figure 4.5: Position Tracking and Attitude Stabilization Original HC

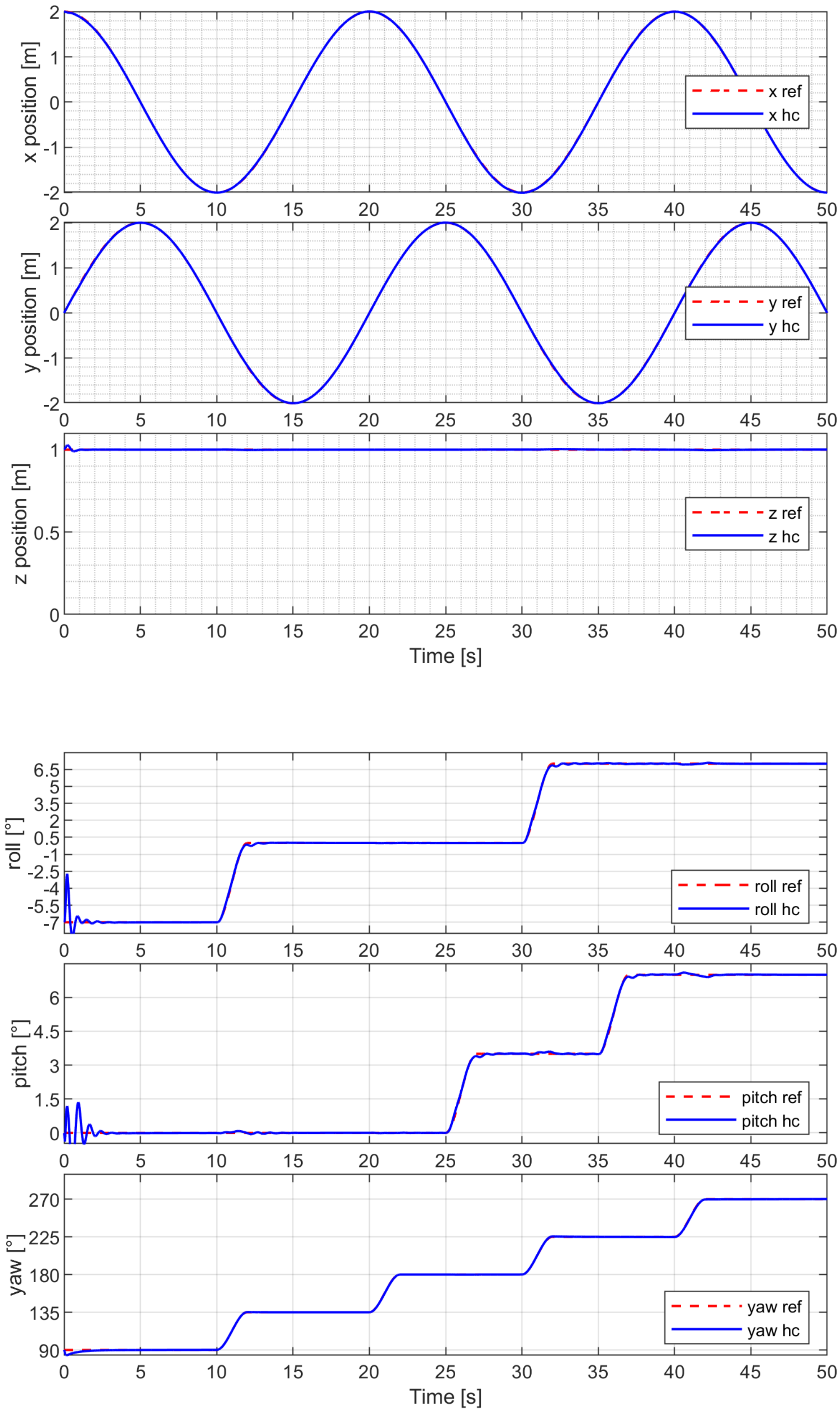


Figure 4.6: Position Tracking and Attitude Stabilization Improved HC

Table 4.4 reports the RMSE values for the position and orientation variables obtained using the two different HC architectures. The analysis of these results further confirms the improved position tracking performance achieved by the improved version of the HC controller. However, the attitude tracking behavior appears to be unaffected.

Controller	**RMSE**					
	x [m]	y [m]	z [m]	ϕ [°]	θ [°]	ψ [°]
HC_{old}	0.0428	0.0221	0.0060	0.2326	0.1327	0.6473
HC_{new}	**0.0063**	**0.0068**	0.0022	0.2558	0.1520	0.6487

Table 4.4: RMSE comparision, ideal conditions

After confirming the effectiveness of the improved version of the HC through comparison with the original version, an investigation was carried out on the robustness of the HC to parametric uncertainties. By increasing the mass by a given percentage (indicated as a subscript in Table 4.5) in the HR01 model implemented in Simulink, the variation in the RMSE values previously presented in Table 4.4 was analyzed. Furthermore, it is specified that from this point onward, the term HC will be used to refer exclusively to the improved version of the Hierarchical Controller.

Controller	**RMSE**					
	x [m]	y [m]	z [m]	ϕ [°]	θ [°]	ψ [°]
HC	0.0063	0.0068	0.0022	0.2558	0.1520	0.6487
$\text{HC}_{+10\%}$	0.0068	0.0073	0.0036	0.2032	0.1348	0.6396
$\text{HC}_{+15\%}$	0.0071	0.0076	0.0050	0.1834	0.1310	0.6359
$\text{HC}_{+20\%}$	0.0074	0.0079	0.0065	0.1671	0.1305	0.6328

Table 4.5: RMSE comparison with mass variation

The results obtained highlight the controller's robustness to uncertainties in model parameters, specifically mass. The analysis was extended up to a 20% increase in mass, and the controller's response remained almost invariant. The RMSE increased by 17.4% in x, 16.2% in y, and 66% in z, compared to the nominal mass case. Moreover, despite the increased uncertainty in the model's mass, the RMSE values achieved by the improved HC are consistently lower than those obtained by the original HC without mass uncertainty, as reported in Table 4.4.
Interestingly, the RMSE values for the angular variables decrease as the mass increases. This behavior could be due to the greater inertia introduced by the increased mass, which dampens angular oscillations and thus reduces the overall RMSE.

The same analysis was carried out by increasing the model's inertia, and the corresponding results are reported in Table 4.6.

Controller	**RMSE**					
	x [m]	y [m]	z [m]	ϕ [°]	θ [°]	ψ [°]
HC	0.0063	0.0068	0.0022	0.2558	0.1520	0.6487
$\text{HC}_{+10\%}$	0.0063	0.0068	0.0022	0.2556	0.1554	0.6281
$\text{HC}_{+15\%}$	0.0063	0.0068	0.0022	0.2555	0.1568	0.6192
$\text{HC}_{+20\%}$	0.0063	0.0068	0.0022	0.2554	0.1581	0.6112

Table 4.6: RMSE comparison with inertia variation

From the analysis of the RMSE values reported in Table 4.6, no significant changes in the RMSE values are detected. An interesting observation is that the RMSE values of the position variables are not affected at all by the inertia variation.
The robustness of the HC is subsequently tested by introducing external disturbances. In particular, Table 4.7 shows the variation in the RMSE values of the state variables due to the addition of a constant force, defined as follows:

$$\mathbf{F}_{dist} = \begin{bmatrix} 0.x \\ 0.x \\ 0 \end{bmatrix} \text{[N]} \tag{4.19}$$

A constant disturbance force was thus applied only along the x and y axes, defined in $\mathscr{F}_W$. The magnitude of this force, expressed in Newton, is reported in Table 4.7 as a subscript of "HC".

Controller	**RMSE**					
	x [m]	y [m]	z [m]	ϕ [°]	θ [°]	ψ [°]
HC	0.0063	0.0068	0.0022	0.2558	0.1520	0.6487
$\text{HC}_{0.1}$	0.0063	0.0069	0.0022	0.2967	0.2283	0.6504
$\text{HC}_{0.2}$	0.0063	0.0069	0.0022	0.4076	0.3713	0.6532
$\text{HC}_{0.3}$	0.0063	0.0069	0.0022	0.5475	0.5294	0.6570
$\text{HC}_{0.4}$	0.0063	0.0070	0.0022	0.6992	0.6925	0.6619
$\text{HC}_{0.5}$	0.0063	0.0071	0.0022	0.8563	0.8577	0.6678

Table 4.7: RMSE comparison with a constant force $\mathbf{F}_{dist} = [0.x, 0.x, 0]^\top$ applied

Analyzing the variation of RMSE values resulting from the application of the abovementioned disturbance force, it can be observed that there are no substantial differences

in position as the force magnitude increases. On the other hand, a lack of robustness is evident in the angular variables. Indeed, compared to the case with no disturbance force applied, a percentage increase of 237.7% on ϕ, 464.3% on θ, and 2.95% on ψ is observed. This behavior could be due to the intrinsic structure of HC, which initially tends to stabilize the UAV position and then proceeds to adjust its attitude. The presence of the disturbance force, causing the controller to operate more aggressively, prevents it from effectively managing both dynamics simultaneously.
The same test was performed by applying a constant disturbance torque, defined in $\mathscr{F}_B$:

$$\boldsymbol{\tau}_{dist} = \begin{bmatrix} 0.x \\ 0.x \\ 0.x \end{bmatrix} \text{[Nm]} \tag{4.20}$$

This disturbance torque was applied along all three axes with the same magnitude, which is reported, as done for the force, as a subscript of "HC" (see Table 4.8).

Controller	**RMSE**					
	x [m]	y [m]	z [m]	ϕ [°]	θ [°]	ψ [°]
HC	0.0063	0.0068	0.0022	0.2558	0.1520	0.6487
$\text{HC}_{0.5}$	0.0063	0.0068	0.0022	0.2574	0.1540	0.8042
$\text{HC}_{0.75}$	0.0063	0.0068	0.0022	0.2583	0.1551	0.9971
HC_1	0.0063	0.0068	0.0022	0.2592	0.1563	1.2239

Table 4.8: RMSE comparison with a constant torque $\boldsymbol{\tau}_{dist} = [0.x, 0.x, 0.x]^\top$ applied

Analyzing the RMSE variation in the state variables, the Hierarchical Controller does not appear to be affected by the presence of the disturbance torque. In fact, even when a constant torque of 1 Nm is applied to each axis, no substantial performance degradation is observed.

4.3.3. Improved HC vs GC and FC

After confirming the effectiveness of the improvements introduced in the HC architecture, the performance of the latter is compared with two other state-of-the-art control solutions, namely GC and FC, presented in Section 4.2. Specifically, three comparative tests were conducted:

- **Test 01**, based on NT01 under ideal conditions
- **Test 02**, based on NT01 with the addition of Gaussian noise (with zero mean and variance as illustrated in Table 4.12)
- **Test 03**, based on NT02 with the same Gaussian noise as in Test 02

The control solutions were tuned during Test 01 to achieve the same control input norm as the HC, thereby equalizing the energy consumption among the control strategies and enabling an objective comparison. The gains used for the alternative control solutions are reported in Table 4.9 (for the HC gains, refer to Table 4.2) and remain unchanged for Test 02 and Test 03.

K_{p1}	`diag([200 200 120])`
K_{p2}	`diag([10 5 10])`
K_{p3}	`diag([15 13 5])`
K_{R1}	`diag([1500 1500 1500])`
K_{R2}	`diag([6 6 6])`
K_{R3}	`diag([10 10 10])`

(a) GC gain matrices.

Q	$10^3 \cdot$ `diag([1 1 5 0.001 0.001 0.001 100 100 5 0.001 0.001 0.001])`
R	$10 \cdot$ `diag([2 2 2 2 2 2])`

(b) FC weight matrices.

Table 4.9: Controllers' gain matrices.

The first comparison performed among the different control solutions is based on NT01 under ideal conditions (Test 01). In this test, the KPIs obtained with the improved HC architecture are reported again and will be compared with the KPIs achieved using GC and FC. Table 4.10 shows the values of the performance indices obtained.

	$\|\|\mathbf{u}\|\|$ $[Hz]$	$\|\|\mathbf{e}_p\|\|$ $[m]$	e_a $[°]$	MSI
HC	104.9486	**0.0074**	**0.2245**	0.1491
FC	104.8850	**0.0045**	**0.0248**	0.1486
GC	104.8919	**0.0002**	**0.0263**	0.1486

Table 4.10: Performance indices, Test 01

Analyzing these indices, it becomes evident that under ideal conditions, no substantial differences are observed among the various control solutions. When examining the position error norm, the GC controller seems to achieve the best performance, followed by the FC and HC controllers, which show errors of the same order of magnitude. Regarding the attitude tracking error, the HC appears to exhibit an order of magnitude higher error, probably due to the initial settling phase appreciable in Figure 4.6. Further conclusions will be drawn from the analysis of the RMSE values reported in Table 4.11. Below, the plots related to the tracking of NT01 under ideal conditions for the GC and FC solutions are presented in Figures 4.7 and 4.8 (for the behavior of HC, refer to Fig. 4.6).

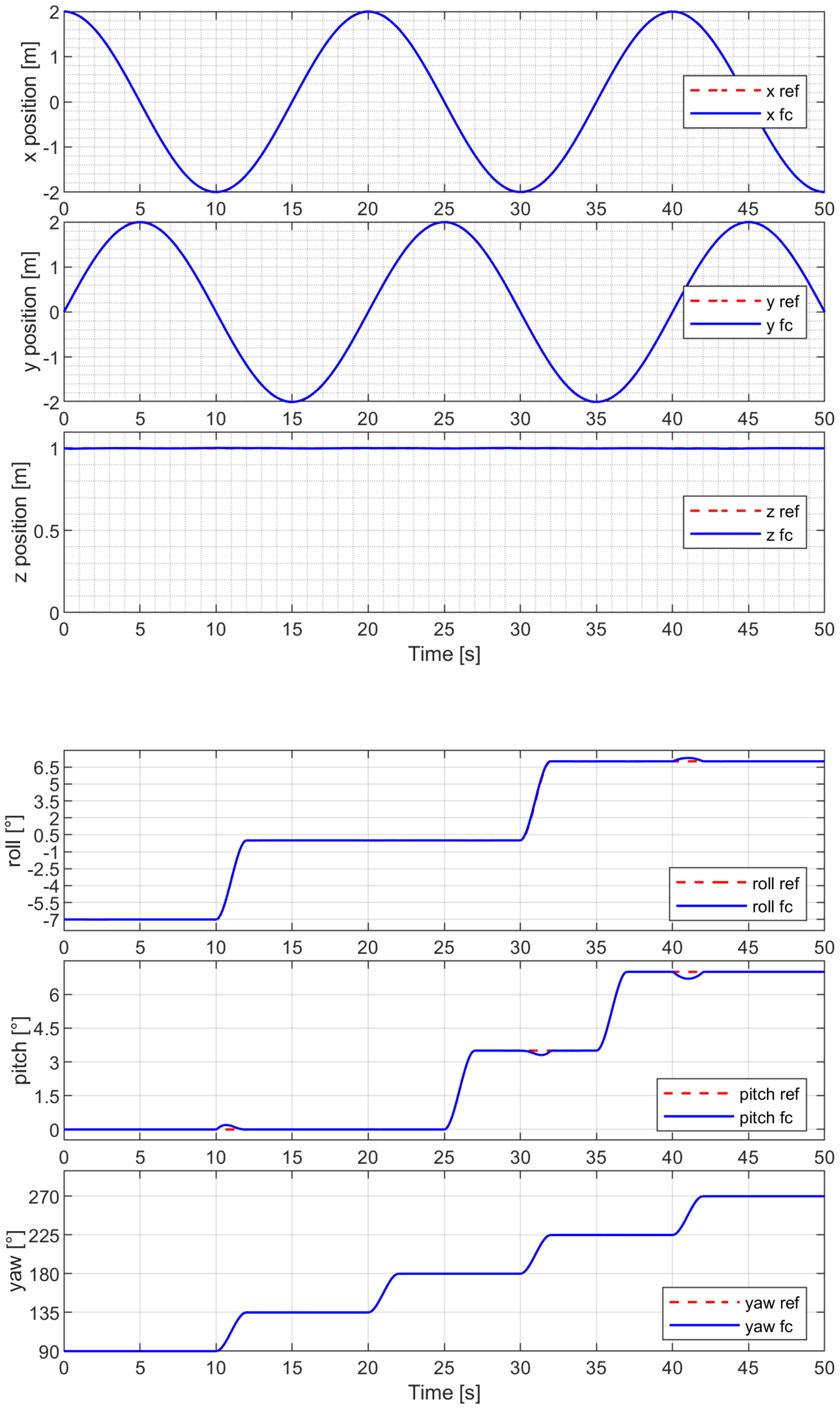


Figure 4.7: Position Tracking and Attitude Stabilization FC, Test 01

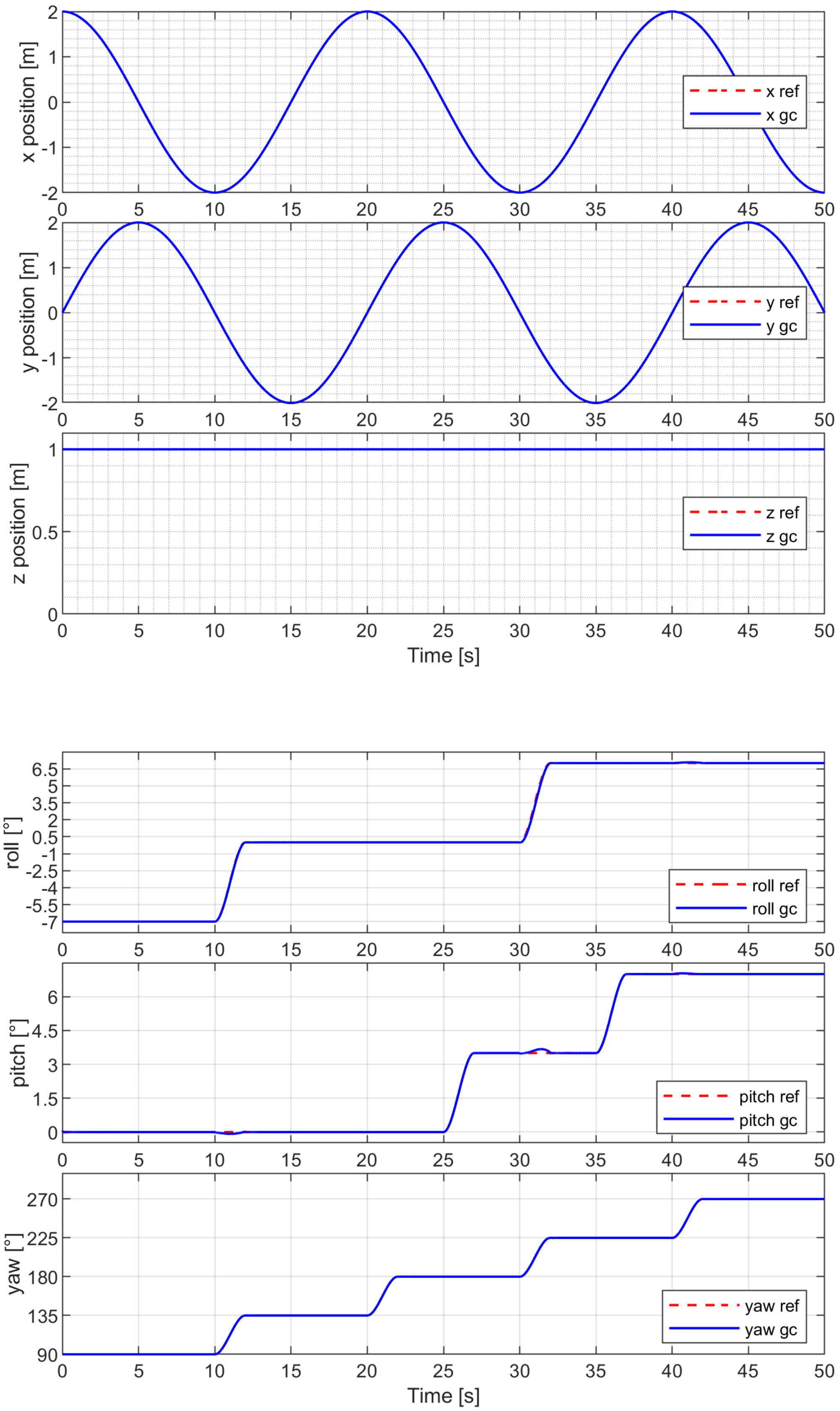


Figure 4.8: Position Tracking and Attitude Stabilization GC, Test 01

Controller	**RMSE**					
	x [m]	y [m]	z [m]	ϕ [°]	θ [°]	ψ [°]
HC	0.0063	0.0068	0.0022	0.2558	0.1520	0.6487
FC	0.0029	0.0033	0.0009	0.0494	0.0566	0.0149
GC	0.0001	0.0001	0	0.0382	0.0253	0.0411

Table 4.11: RMSE comparision, Test 01

The analysis of the RMSE values confirms that the Geometric Controller exhibits the best performance in position tracking, with RMSE values for the corresponding variables being practically negligible. The FC and HC solutions also demonstrate excellent position performance, with RMSE values for these variables on the order of millimeters. The comparison between HC and FC carried out in Test 01 confirms the conclusions drawn in [14], regarding the superior performance of FC compared to HC in the absence of disturbances, thus making it more suitable for indoor applications.
In order to assess the robustness of the different control strategies, from this point onward, every other numerical test conducted is performed with the addition of Gaussian noise (with zero mean and variance as illustrated in Table 4.12) on every state variable, as reported in the incipit of the section. The selected values mimic the performance of a MoCap system.

	1st component	**2nd component**	**3rd component**
$\mathbf{p}\,[m^2]$	$4.099 \cdot 10^{-7}$	$2.838 \cdot 10^{-7}$	$2.105 \cdot 10^{-8}$
$\boldsymbol{\delta}\,[deg^2]$	0.0012	0.0011	0.0011
$\mathbf{v}\,[(m/s)^2]$	$2.050 \cdot 10^{-6}$	$1.419 \cdot 10^{-6}$	$1.050 \cdot 10^{-7}$
$\boldsymbol{\omega}\,[(deg/s)^2]$	0.0024	0.0022	0.0022

Table 4.12: Noise variances used in Test 02 and Test 03

The second test (Test 02) conducted to compare HC, GC, and FC is also based on NT01. Table 4.13 reports the obtained KPIs, which will be related to those obtained during Test 01 in order to investigate how the addition of noise may have influenced them.

	$\|\mathbf{u}\|$ $[Hz]$	$\|\mathbf{e}_p\|$ $[m]$	e_a $[°]$	MSI
HC	105.2114	0.0078	0.5467	0.1507
FC	104.9782	0.0052	0.4146	0.1492
GC	107.1448	0.0184	0.4356	0.1622

Table 4.13: Performance indices, Test 02

As can be observed, the norm of the control input remains almost unchanged compared to Test 01, with a slight increase in the GC controller, which also exhibits a higher MSI, as expected. The FC controller achieves a lower position error norm compared to its competitors. A more detailed comparison is carried out later through the analysis of the RMSE values reported in Table 4.14. Regarding the attitude tracking error, the behavior of the three solutions appears to be comparable, with the HC controller being the least accurate, showing a difference of approximately 0.1° with respect to the value obtained during Test 01.
It is also worth noting that the GC controller, compared to the ideal case reported in Table 4.10, experienced a significant increase in the position error norm and attitude error, with a percentage increase of 9100% and 1556.65%, respectively. Further conclusions will be drawn from the analysis of the RMSE values reported in Table 4.14. Before that, the plots related to position and attitude tracking during Test 02 are presented in figures 4.9, 4.10, 4.11.

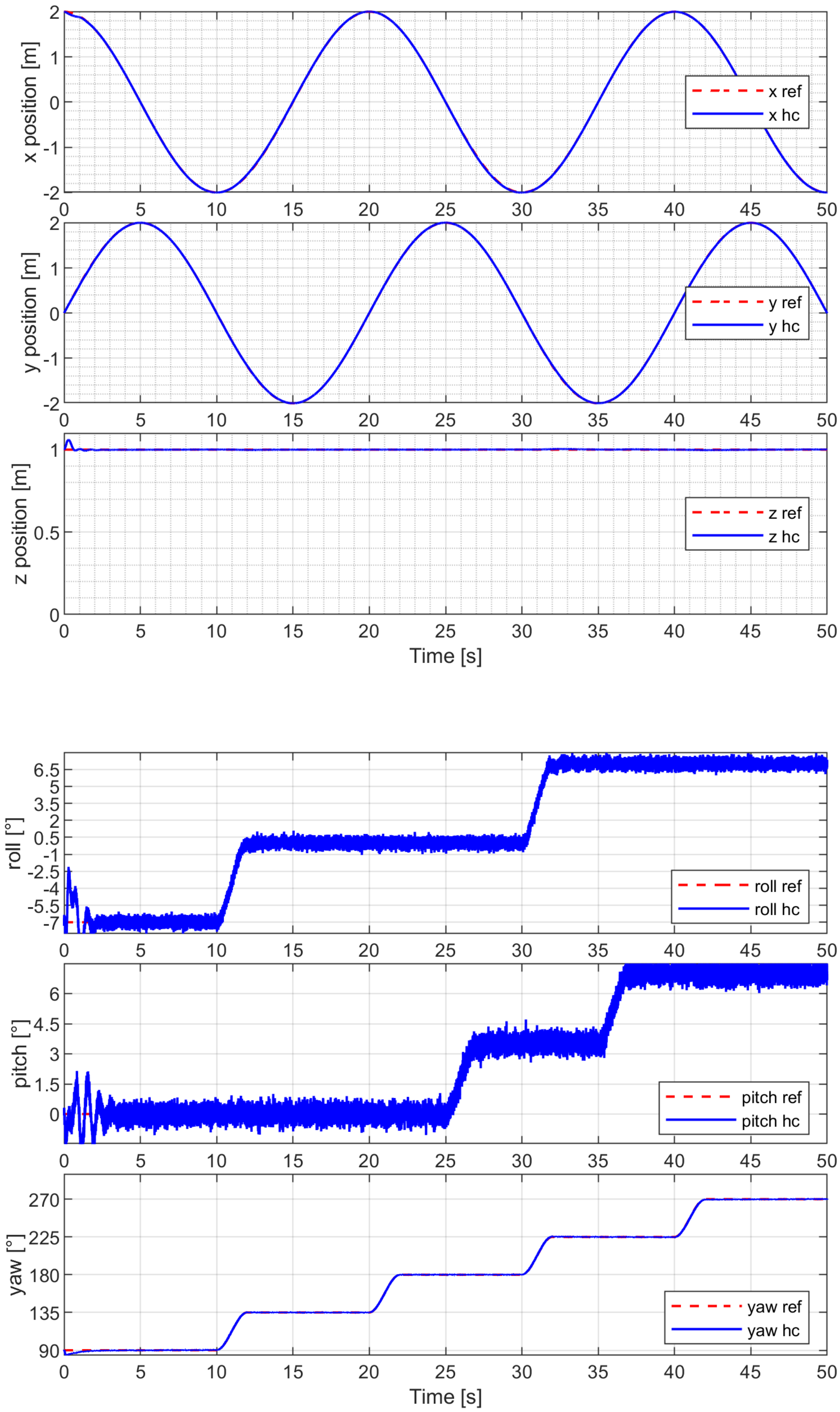


Figure 4.9: Position Tracking and Attitude Stabilization HC, Test 02

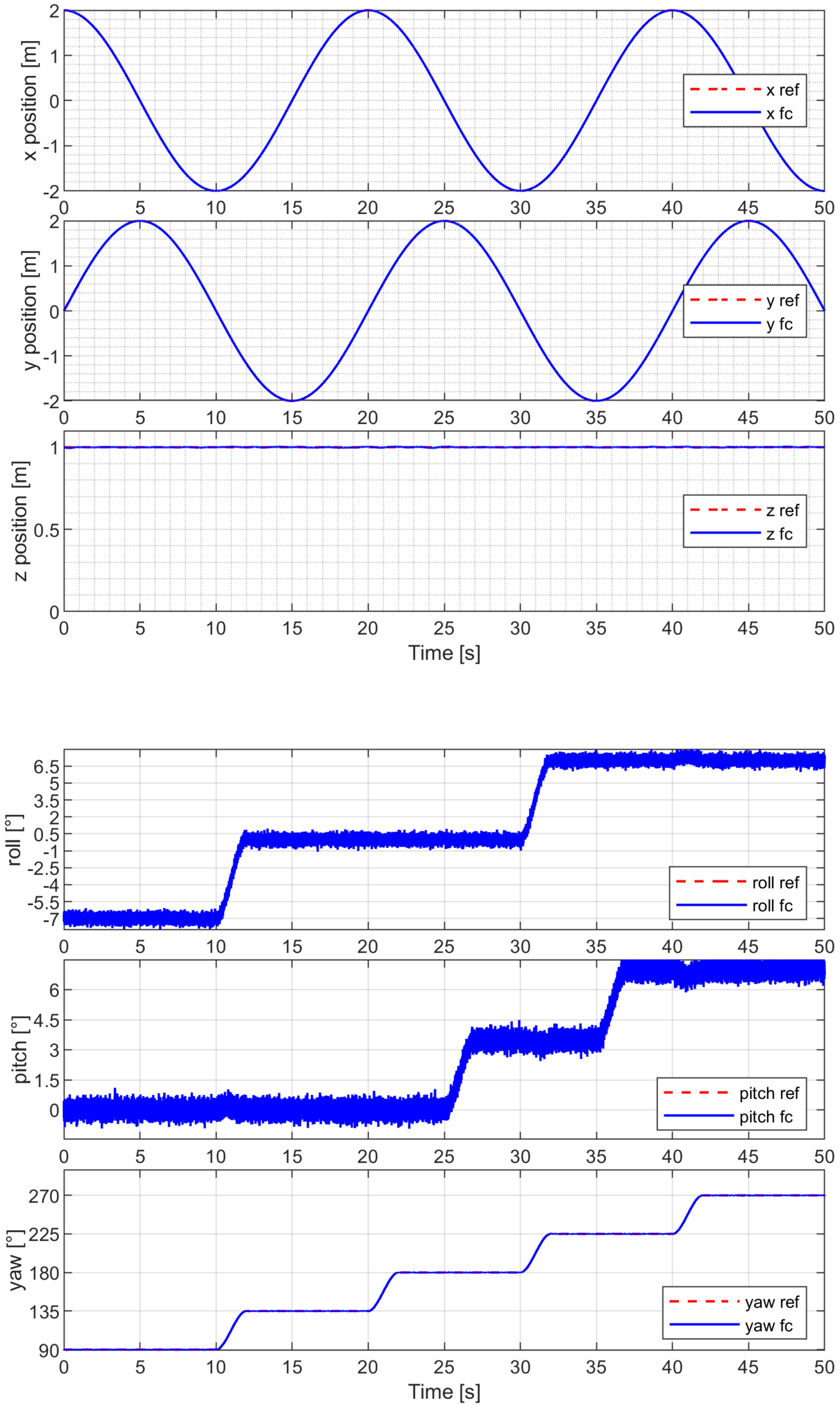


Figure 4.10: Position Tracking and Attitude Stabilization FC, Test 02

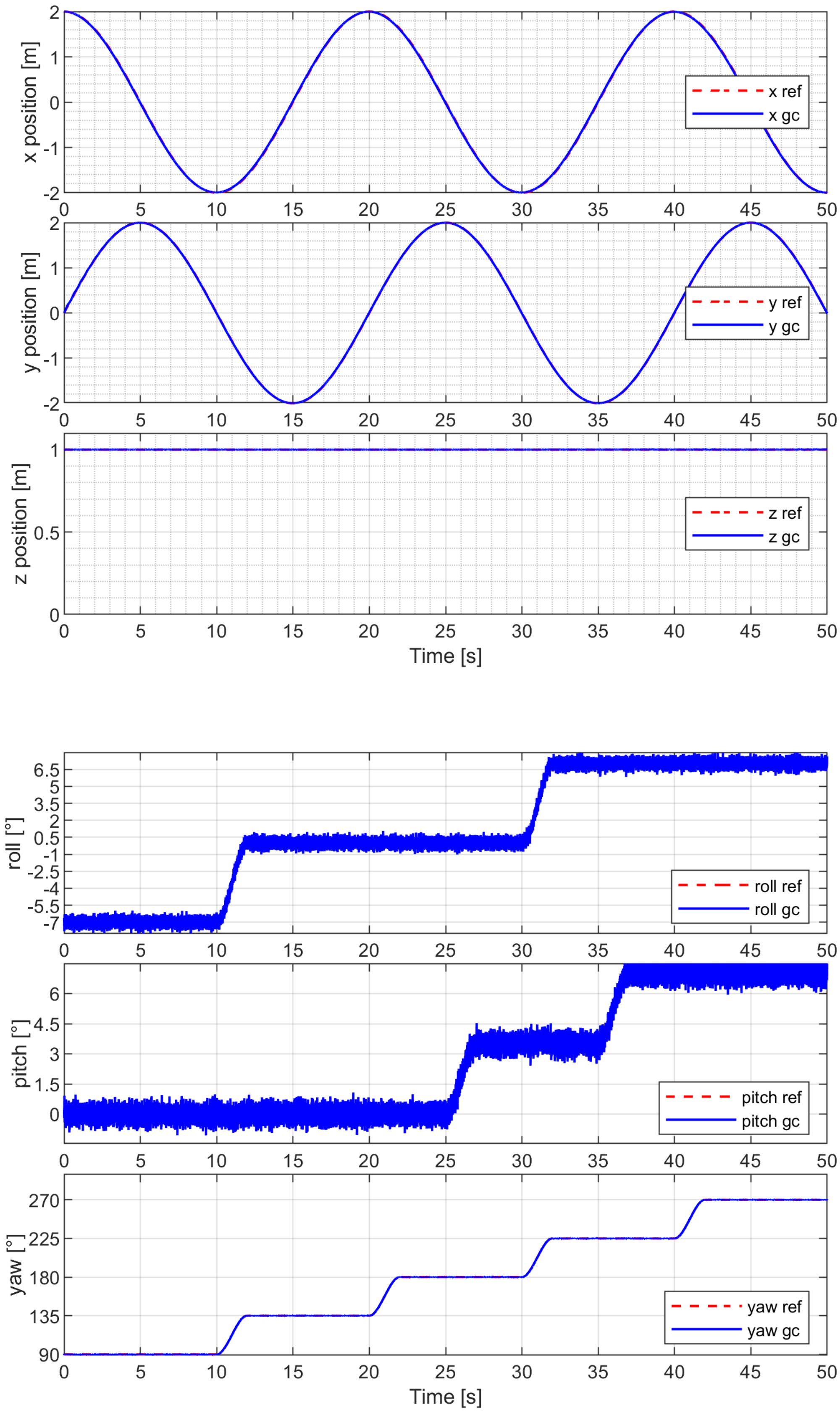


Figure 4.11: Position Tracking and Attitude Stabilization GC, Test 02

As shown in Table 4.14, all control solutions do not present significant issues in position tracking, with RMSE values for the position variables on the order of millimeters. However, it is worth noting that the RMSE values of the GC controller on the position variables drastically increase compared to the ideal case reported in Table 4.11. In fact, a percentage increase of 16600% on the x variable, 8700% on the y variable, and an absolute increase of 0.0006 meters on the z variable are recorded.

Controller	**RMSE**					
	x [m]	y [m]	z [m]	ϕ [°]	θ [°]	ψ [°]
HC	0.0087	0.0048	0.0045	0.4761	0.3242	0.6622
FC	0.0013	0.0014	0.0014	0.2689	0.2592	0.2543
GC	0.0167	0.0088	0.0006	0.2852	0.2719	0.2668

Table 4.14: RMSE comparision, Test 02

The situation differs when analyzing the angular errors. To ensure the most accurate comparison possible, the evolution of the angular errors is reported below. The analysis was conducted starting from 5 seconds onwards, thus excluding the initial settling phase of the angles toward the reference value observed with the HC, as illustrated in Figure 4.9.

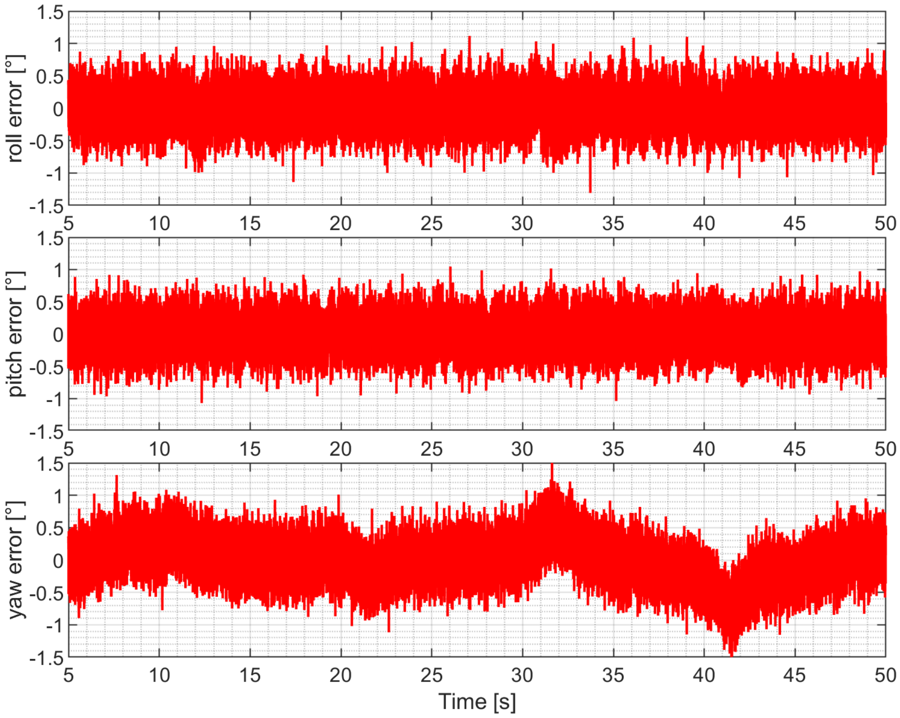


Figure 4.12: Attitude Error HC, Test 02

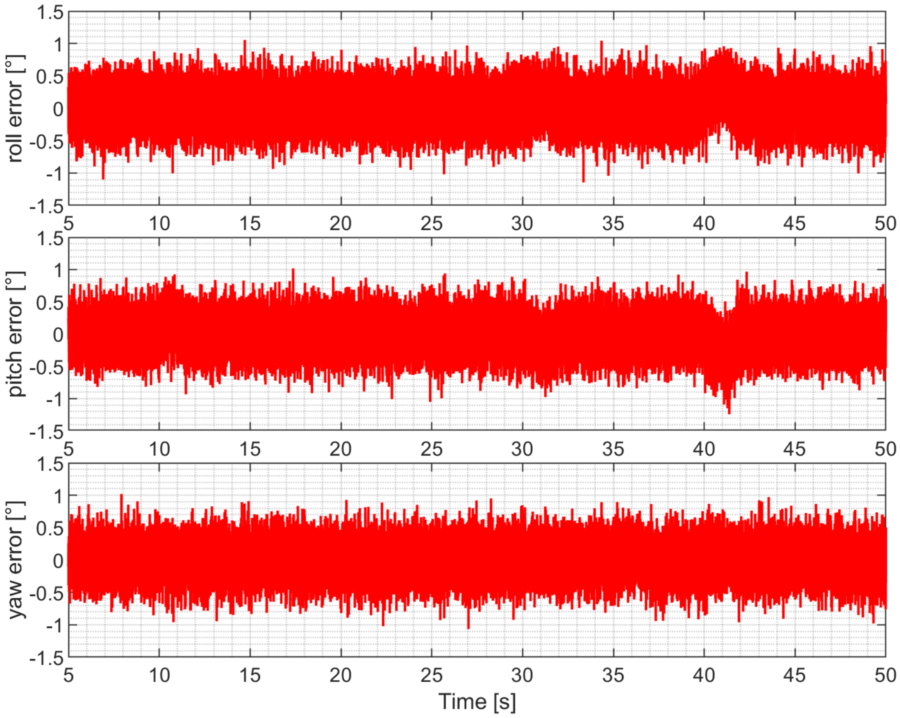


Figure 4.13: Attitude Error FC, Test 02

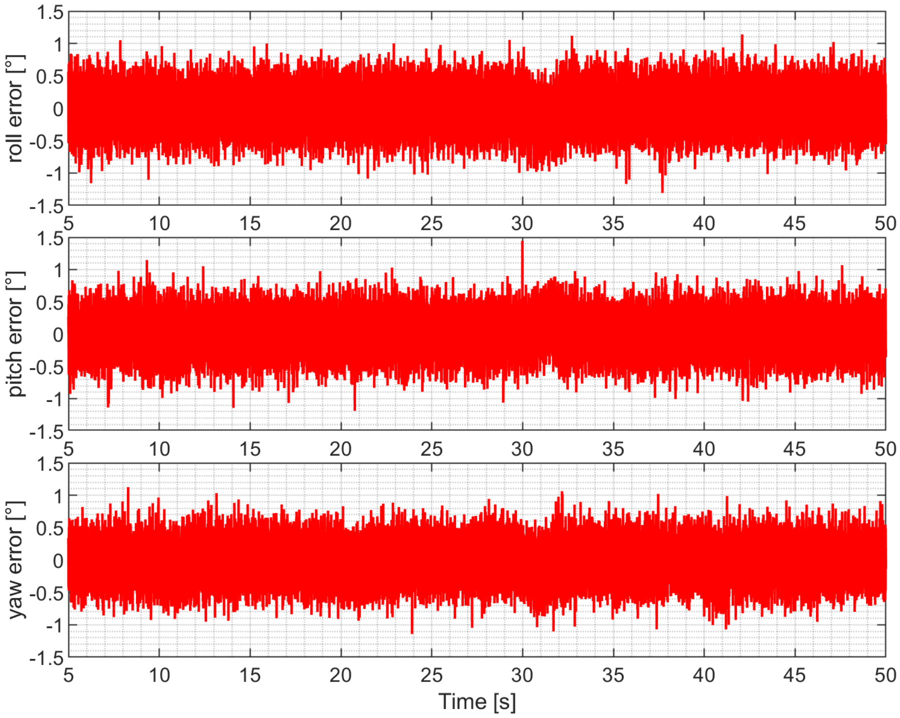


Figure 4.14: Attitude Error GC, Test 02

Observing Figures 4.12, 4.13, and 4.14, the angular error trends in all three control solutions remain within ±1°, indicating a similar behavior among the different control solutions.

The results obtained during Test 02 confirm the effectiveness of the improvements introduced in the Hierarchical Controller architecture, thus making HC a worthy competitor. Moreover, from Test 02, the Geometric Controller appeared to be the least robust to disturbances. For this reason, the decision was made to discard this control solution from this point onward and focus the comparison exclusively between FC and HC, which is still sufficient considering that the ultimate goal is to evaluate the performance of the improved HC architecture.

The numerical validation concludes with a final comparison test based on NT02, in order to assess the performance of each controller when subjected to a different reference, without modifying the gains (Test 03). Table 4.15 presents the resulting performance indices obtained during this simulation.

	$\|\mathbf{u}\|$ $[Hz]$	$\|\mathbf{e}_p\|$ $[m]$	e_a $[°]$	MSI
HC	106.5797	**0.0179**	0.4851	0.1583
FC	106.2551	**0.0023**	0.4284	0.1562

Table 4.15: Performance indices, Test 03

Analyzing these results, it can be observed that in terms of energy consumption, the two control solutions appear comparable, presenting similar values of control effort norm and MSI. Regarding the position error norm, the change in reference trajectory does not seem to significantly affect the FC controller, while the HC exhibits a value that is one order of magnitude higher than the one reported in Table 4.13, during Test 02. As for the attitude tracking error, the opposite behavior is observed. With this trajectory, the HC solution yields a slightly lower e_a value (by approximately two hundredths of a degree) compared to the result obtained with the trajectory discussed in Test 02. On the other hand, the FC controller shows a slight increase in this index, though still maintaining a lower value than HC. Further conclusions will be drawn from the RMSE values presented in Table 4.16. Figures 4.15 and 4.16 show the tracking plots for the proposed trajectory using both control solutions.

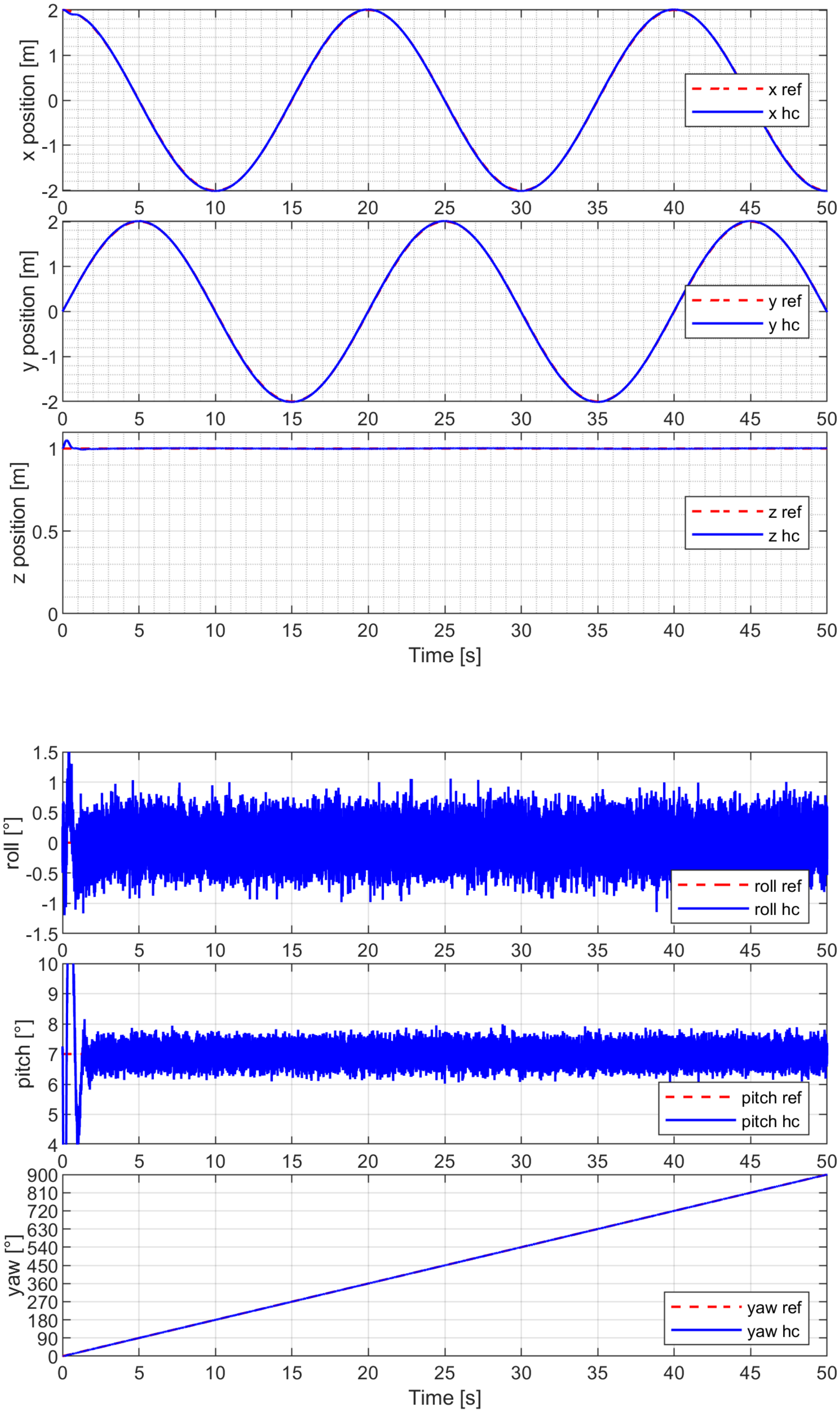


Figure 4.15: Position and Attitude Tracking HC, Test 03

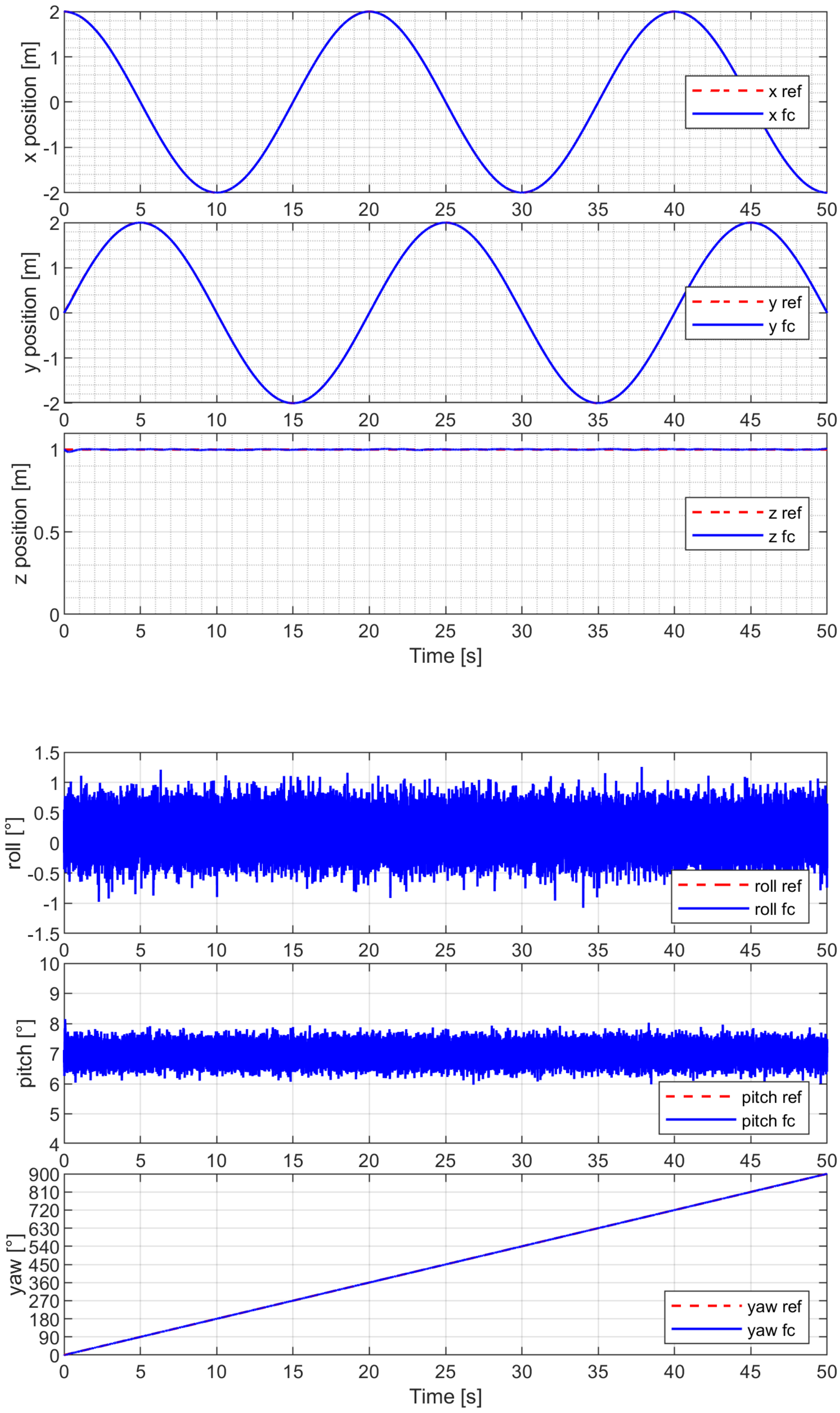


Figure 4.16: Position and Attitude Tracking FC, Test 03

As shown by the RMSE values on the position variables (Table 4.16), both control solutions exhibit satisfactory performance in position tracking, confirming the outcomes observed in Test 02. The FC controller demonstrates better behavior, with RMSE values on the position variables in the millimeter range, while the HC remains in the millimeter range only along the z-axis, reaching values greater than 1 cm on the x and y axes. The situation differs for the angular variables. The HC controller shows a lower RMSE on the roll angle compared to FC, but higher values on the pitch and yaw angles.

Controller	**RMSE**					
	x [m]	y [m]	z [m]	ϕ [°]	θ [°]	ψ [°]
HC	0.0135	0.0119	0.0039	0.2770	0.7597	0.3094
FC	0.0011	0.0020	0.0021	0.3122	0.2530	0.2608

Table 4.16: RMSE comparison, Test 03

Due to the high RMSE value on the pitch angle for the HC solution, the angular error trends for both control strategies are shown in Figures 4.17 and 4.18. Analyzing these trends, it becomes evident that the high pitch RMSE observed with the HC controller is mainly due to the initial settling phase to reach the desired reference value. In fact, once settled, the error remains within ±1°, showing a behavior very similar to that of the FC solution.

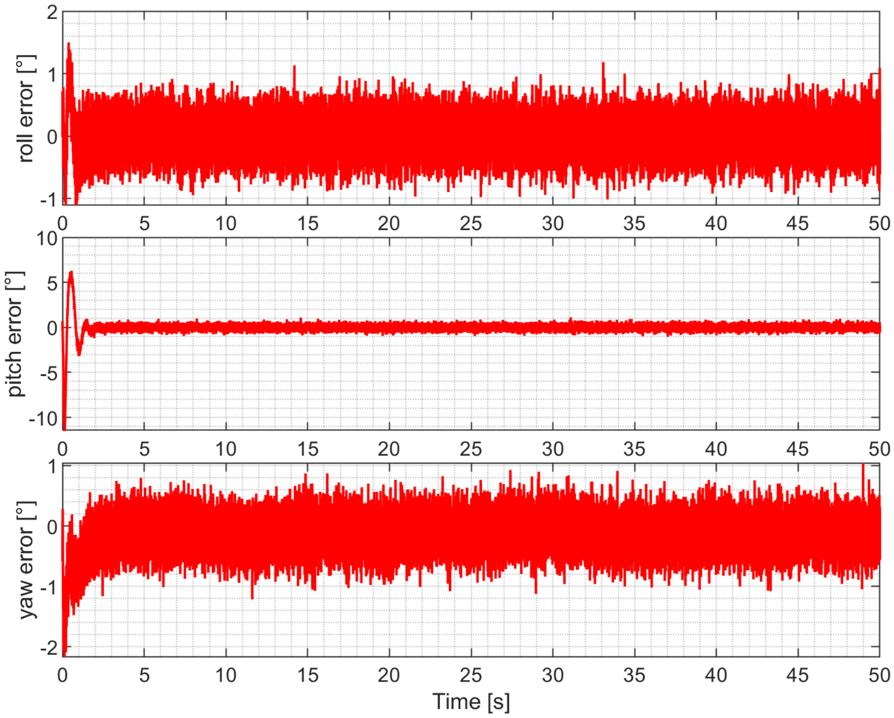


Figure 4.17: Attitude Error HC, Test 03

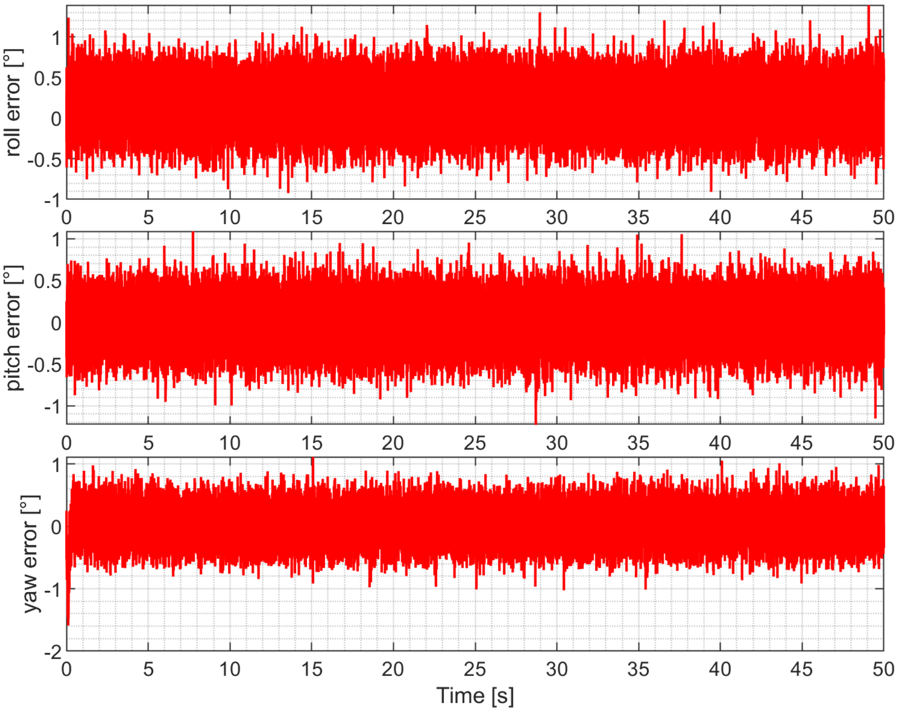


Figure 4.18: Attitude Error FC, Test 03

4.4. Physics-Based Validation

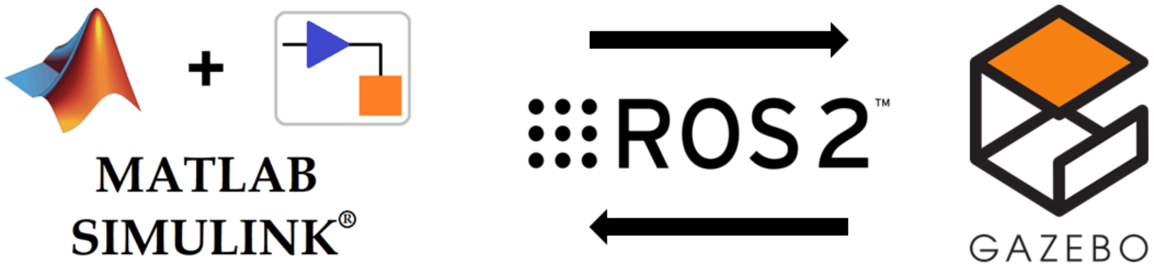


Figure 4.19: Physics-Based Validation Setup

The setup used to perform the physics-based validation of the HC is described in Figure 4.19. The fully-actuated hexarotor presented in Section 4.1 was modeled in Gazebo, as shown in Fig.4.20. The commands from the controller, which is still implemented in Simulink, were adapted to be properly received by the UAV in Gazebo, which expects to receive the angular velocities of the six propellers in radians per second. The control input is therefore transmitted, through a communication network implemented using ROS 2[3], to the UAV modeled in Gazebo, which in turn sends its odometry back to Simulink, again via ROS 2. This odometry data is used as feedback for the controller, with a communication frequency set to 100 Hz.

---

[3] ROS 2 (Robot Operating System 2) is an open-source middleware framework designed for the development of robotic software. It provides tools, libraries, and communication infrastructure to enable modular, distributed, and real-time control of robotic systems.

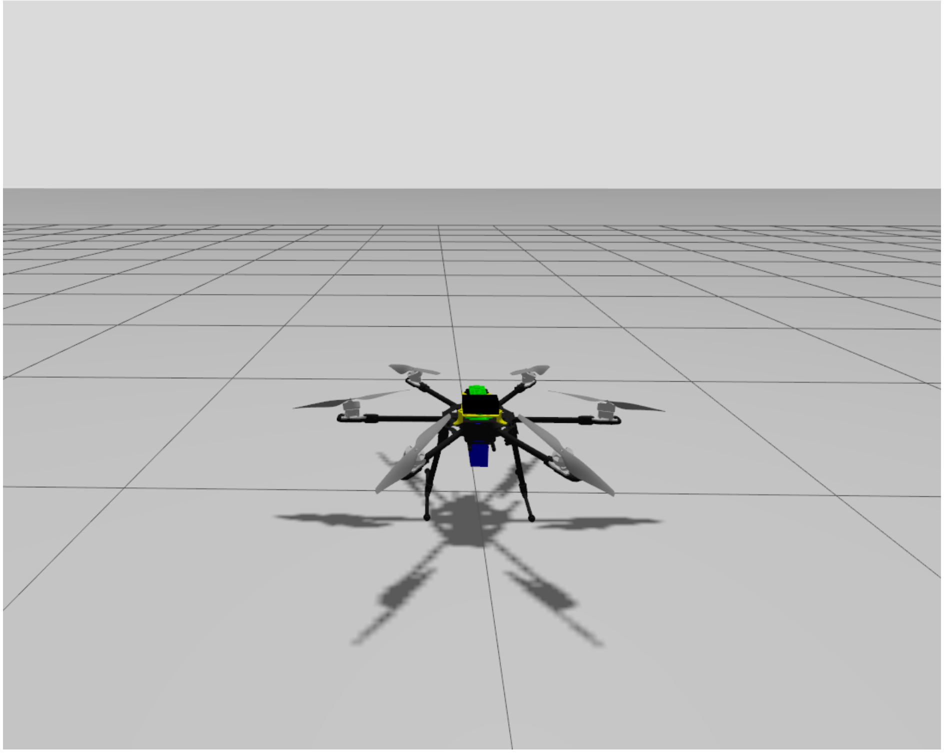


Figure 4.20: HR01 in Gazebo before take-off

4.4.1. Navigation Task Overview

This section describes the navigation task selected to perform the physics-based validation of the HC, referred to by the acronym NT03. Specifically, NT03 can be divided into a take-off maneuver (during which the hexarotor reaches an altitude of 2 meters) and the stabilization of the initial conditions to follow the second part, during which the tilted hexarotor is commanded to follow a circular trajectory with a radius of 2 meters. During this trajectory, the roll and pitch angles are varied according to the sequences $[-7°,\ 0°,\ 7°]$ and $[0°,\ 3.5°,\ 7°]$, respectively, while the yaw angle is kept constant at 0°. Going into further detail, the take-off and stabilization maneuver, which spans the time interval [0, 20] seconds, can be divided as follows:

- $\Delta t = [0, 5]$: the UAV is commanded to reach an altitude of 2 meters.
- $\Delta t = [5, 10]$: the UAV is required to achieve a roll angle of $-7°$.
- $\Delta t = [10, 20]$: the position along the y-axis is adjusted such that at 20 seconds the UAV attains the initial velocity required to execute the circular trajectory.

The choice of using a simplified navigation task compared to NT01 and NT02 is motivated by the physics-based setup, which is employed for the first time in this thesis

work and is still under development.
The initial conditions of the UAV in the Gazebo simulator were set as follows:

$$\begin{aligned}\mathbf{x}_0 &= \begin{bmatrix}\mathbf{p}^\top & \mathbf{v}^\top & \boldsymbol{\delta}^\top & \boldsymbol{\omega}^\top\end{bmatrix}^\top = \\ &= [2 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^\top\end{aligned}$$

4.4.2. Improved HC vs FC

To perform an objective comparison of the performance of the HC, the FC was also suitably adapted to enable a direct comparison. Both controllers were tuned to achieve almost the same average control norm, as previously done for the Numerical Validations discussed in Section 4.3. The corresponding gain values used in the two control strategies are reported in Table 4.17.

k_{pp}	k_{pd}	k_{ap}	k_{ad}	k_{Δ}	k_q	k_{pi}
20	3	20	1.5	2	1	1

(a) HC gains.

Q	$10 \cdot$ `diag([100 100 100 0.1 0.1 0.1 800 800 800 0.1 0.1 0.1])`
R	$10 \cdot$ `diag([1 1 1 1 1 1])`

(c) FC weight matrices.

Table 4.17: Controllers' gains.

The key performance indices obtained throughout the simulation of NT03 are reported in Table 4.18.

	$\|\mathbf{u}\|$ $[rad/s]$	$\|\mathbf{e}_p\|$ $[m]$	e_a [°]	MSI
HC	944.1305	0.1460	**0.4330**	0.3767
FC	940.4965	0.1247	**2.3096**	0.3730

Table 4.18: Performance Indices, physics-based Validation

The analysis of these values reveals that the control input norm is comparable between the two control strategies, as both were tuned with the specific aim of achieving this outcome. This observation is further confirmed by the nearly identical MSI values. The FC appears to perform slightly better in terms of position tracking, exhibiting a lower position error norm of 2 centimeters compared to the HC. The opposite is observed in terms of attitude stabilization: while the FC shows an attitude error of 2.3°, the HC maintains a significantly lower error of only 0.4°, remaining within acceptable bounds.

The tracking plots during NT03 are also reported. For clarity of presentation, the plots are organized according to the two phases of NT03 introduced in Section 4.4.1.

In Figure 4.21, observing the variable z, it is evident that the HC solution tends to stabilize more accurately on the 2 meter target compared to the FC solution. Conversely, along the x and y axes, the HC solution exhibits some deviation and oscillation.

Regarding the attitude behavior, there appears to be no comparison, as the HC solution tends to stabilize the UAV at the requested roll value despite some oscillations, whereas the FC solution develops a significant error relative to the reference, confirming the validity of the attitude error reported in Table 4.18.

Instead, Figure 4.22 depicts the tracking of the circular trajectory with orientation adjustments following the previously discussed maneuver. Only two periods of the trajectory are shown, corresponding to a total flight time of 40 seconds.

As previously highlighted, the FC solution fails to accurately track the roll and pitch reference angles, deviating on average by approximately 2° from the desired value, whereas the HC solution demonstrates better tracking performance. In terms of position, both solutions exhibit satisfactory tracking, with the only significant difference observed in the altitude z. Indeed, the FC solution is unable to maintain the 2-meter altitude reference, showing continuous altitude drops.

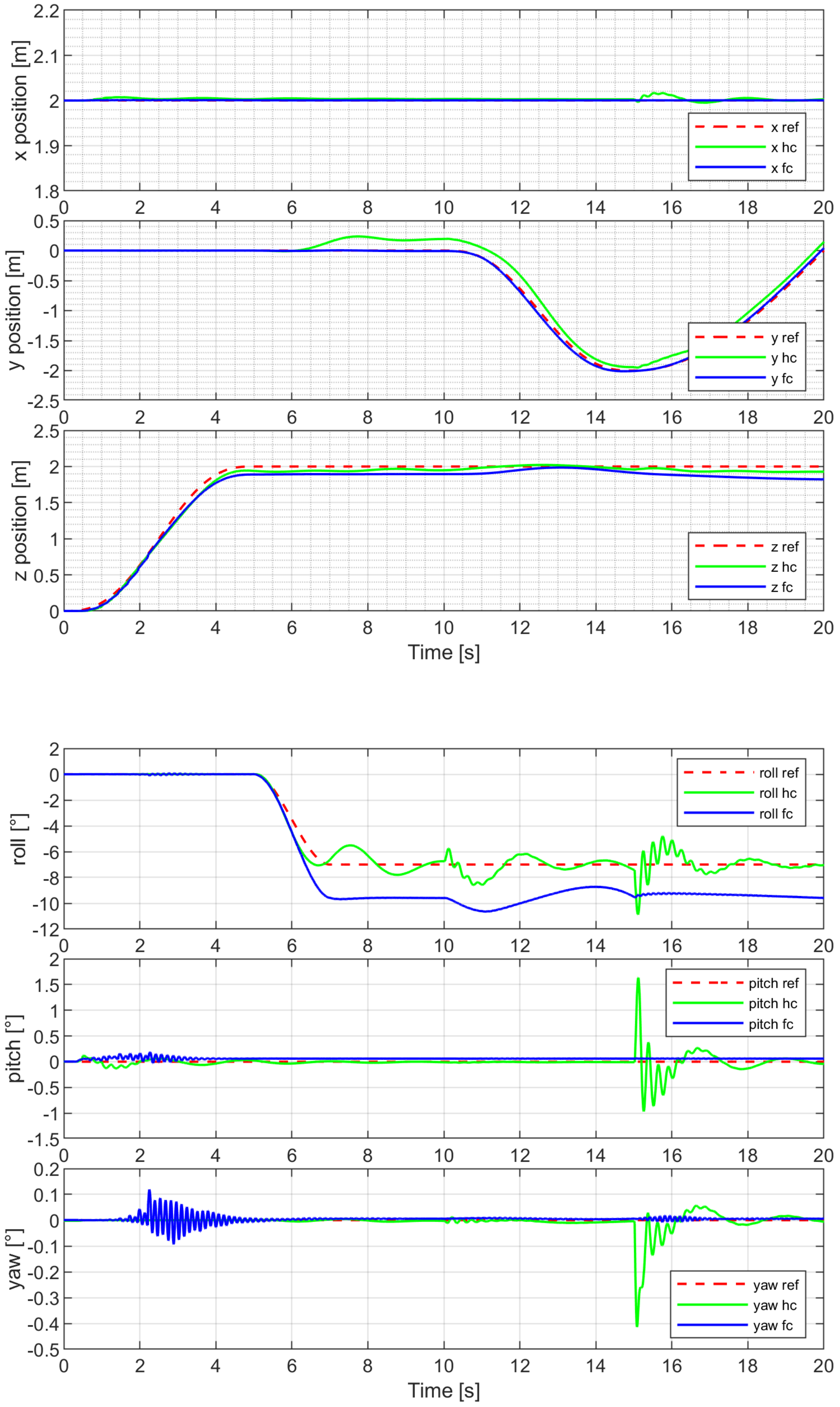


Figure 4.21: Take-off and initial conditions adjustment, HC vs FC

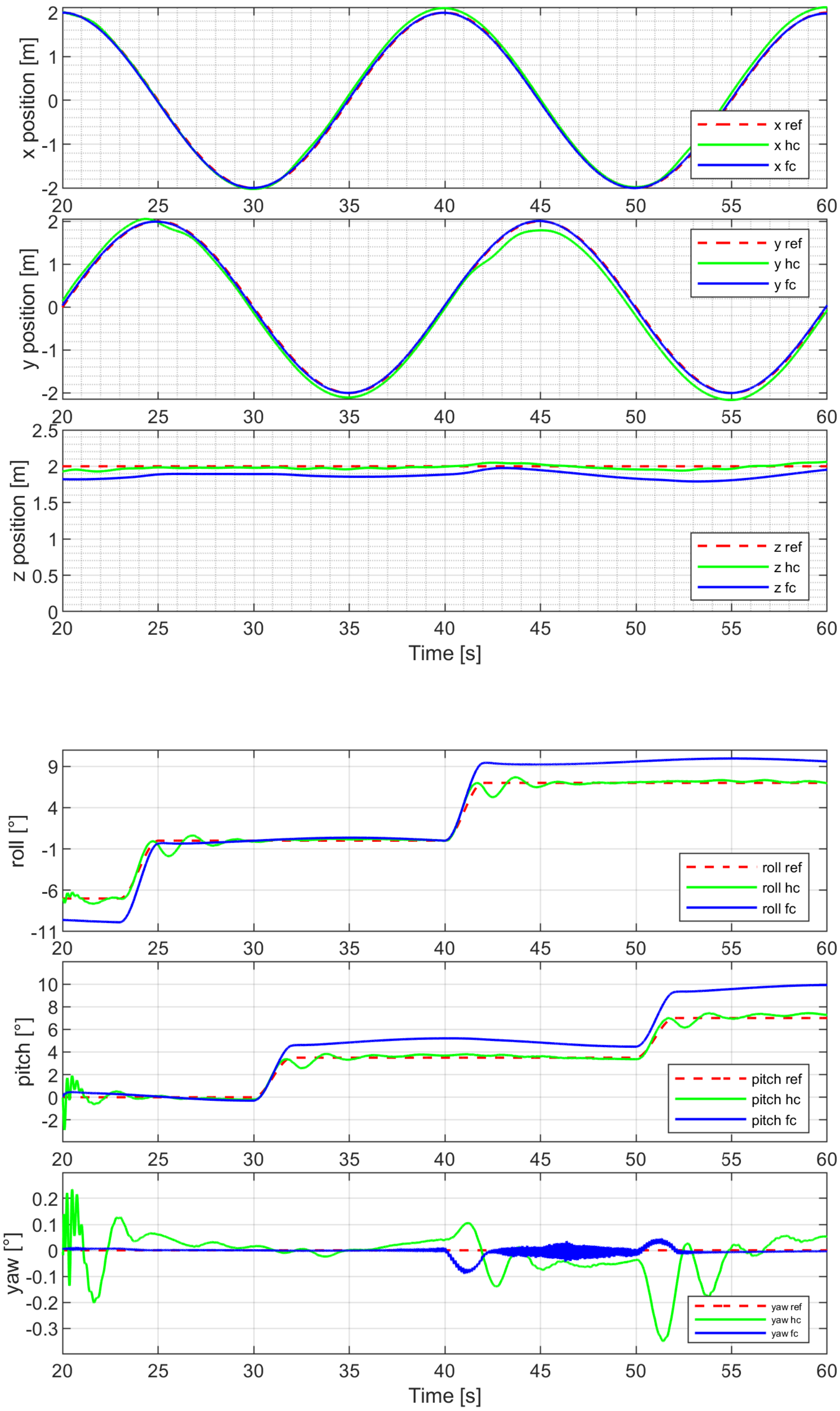


Figure 4.22: Position and Attitude Tracking, HC vs FC

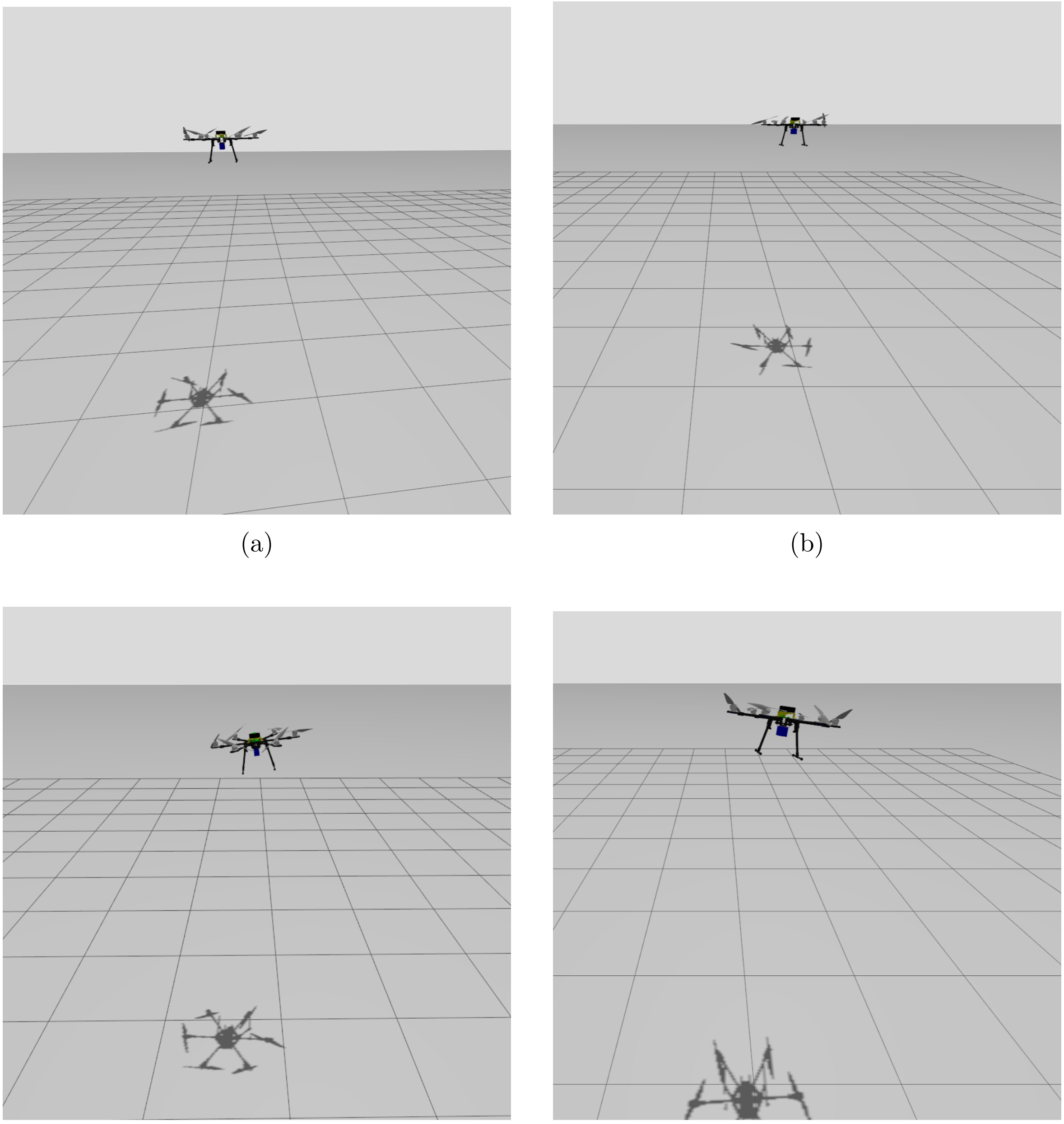
To complete this comparison, Table 4.19 reports the RMSE values for each state variable.

Controller	**RMSE**					
	x [m]	y [m]	z [m]	ϕ [°]	θ [°]	ψ [°]
HC	0.0727	0.1324	0.0431	0.4987	0.3060	0.0712
FC	0.0334	0.0304	0.1255	2.2385	1.8434	0.0429

Table 4.19: RMSE comparision in Gazebo environment

The analysis of the RMSE values for the position variables confirms that both solutions perform well in position tracking. Conversely, the RMSE values for the angular variables reinforce the superior effectiveness of the HC solution compared to the FC solution. This is likely due to the fact that, in a physics-based environment such as Gazebo, the linearity assumption on which the FC solution relies becomes invalid, thus degrading its performance. This further confirms the necessity of continuing to investigate nonlinear control solutions such as HC.

This validation concludes by presenting in Figure 4.23 snapshots of the HR01 flight in Gazebo, controlled by the HC controller.



(a) (b)

(c) (d)

Figure 4.23: Captured snapshots of HR01 following the circular trajectory using Hierarchical Controller. In Figures (c) and (d), the roll inclination of HR01 can be clearly observed.

5. DISCUSSION AND CONCLUSION

The work presented in this thesis focused on the analysis and enhancement of the Hierarchical Controller (HC) architecture, applied to a fully-actuated hexarotor with fixed tilted propellers. After revising and improving the architecture proposed in [14], an extensive comparative study was conducted, firstly between the improved HC controller and its original architecture, then with two other state-of-the-art control strategies: the Geometric Controller (GC) and the Flatness-based Controller (FC).

Numerical simulations in MATLAB/Simulink proved the improved HC architecture to be more effective than its original version. In particular, the correction of the derivative of the zero-moment direction and the inclusion of an integral term led to a significant improvement in position and orientation tracking, without increasing the average energy consumption.

The comparison with GC and FC highlighted that the HC solution is a valid alternative for control. Specifically, Test 01 revealed that, under ideal conditions, HC showed the worst behavior (see Table 4.10). However, when Gaussian noise was added in Test 02, HC demonstrated considerable robustness, exhibiting only moderate deviations in KPIs compared to Test 01. In contrast, GC turned out to be the least robust solution among the three (see Table 4.13).

In Test 03, in which only HC and FC were compared while changing the navigation task and keeping the gains unchanged, HC showed inferior performance compared to FC (see Table 4.15).

This behavior was overturned in the physics-based validation carried out in Gazebo, in which the HC controller demonstrated superior robustness with respect to the FC controller, as shown in Table 4.19, despite the latter exhibiting better performance in the numerical simulations. This result suggests that the HC controller is better suited to handle parametric uncertainties, external disturbances, and unmodeled dynamics present in more realistic environments, confirming HC as the best choice for outdoor applications.

Moreover, this outcome further confirms that control solutions which explicitly account for the systems nonlinearities (such as the HC controller) may be more appropriate in real-world scenarios, where the validity of linear assumptions (on which the FC approach is based) deteriorates.

To conclude, it is worth stating that, with a more accurate tuning process of the HC, it would certainly have been possible to achieve better tracking of the desired reference trajectory in the physics-based environment, albeit at the cost of increased control effort. However, the objective of this thesis was to demonstrate the effectiveness of the HC solution in comparison with other valid approaches. Therefore, for this purpose, the work conducted has proven to be more than adequate.

Finally, possible future developments are proposed. After having confirmed the effectiveness and usefulness of the setup used for physics-based simulationsrepresented in Figure 4.20 and described in Section 4.4, which is still under developmentit would be possible to further investigate the performance of HC by varying the navigation tasks and performing more precise gain tuning, with the ultimate goal of optimally controlling the Hexarotor HR01 described in Section 4.1. This would enable reaching the next step of this thesis work, namely the control of the UAV in the real world.

Bibliography

[1] Tká, M., and Mésáro, P. (2019). Utilizing drone technology in the civil engineering. SSP - Journal of Civil Engineering, 14(1), 2737. DOI: 10.1515/sspjce-2019-0003

[2] Calderan Pier: Droni DIY. Il manuale per hobbisti e maker, Apogeo, 2015

[3] F. Ruggiero, V. Lippiello, and A. Ollero, Aerial manipulation: A literature review, IEEE Robot. Autom. Lett., vol. 3, no. 3, pp. 19571964, July 2018. doi: 10.1109/LRA.2018.2808541.

[4] Rashad, R., Goerres, J., Aarts, R., Engelen, J. B. C., & Stramigioli, S. (2019). Fully Actuated Multirotor UAVs: A Literature Review. IEEE Robotics and Automation Magazine.

[5] Rashad, R., Kuipers, P., Engelen, J. B. C., & Stramigioli, S. (2018). Design and Control of a Fully-Actuated Hexrotor for Aerial Manipulation Applications. Journal of Mechanisms and Robotics, 10(4), 041007.

[6] Bodie, K., Brunner, M., Pantic, M., Walser, S., Pfändler, P., Angst, U., Siegwart, R., & Nieto, J. (2019). An Omnidirectional Aerial Manipulation Platform for Contact-Based Inspection. Robotics: Science and Systems

[7] M. Connolly, The use of multi-rotor remotely operated aerial vehicles (roavs) as a method of close visually inspecting (cvi) live and difficult to access assets on offshore platforms, in: Abu Dhabi International Petroleum Exhibition and Conference, Abu Dhabi, UAE, 2014.

[8] He, G., Jangir, Y., Geng, J., Mousaei, M., Bai, D., & Scherer, S. (2023). Image-based Visual Servo Control for Aerial Manipulation Using a Fully-Actuated UAV. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)

[9] F. G. Costa, J. Ueyama, T. Braun, G. Pessin, F. S. Osório, and P. A. Vargas, The use of unmanned aerial vehicles and wireless sensor network in agricultural applications, in: Proc. IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Munich, Germany, pp. 50455048, 2012.

[10] H. Almurib, P. Nathan, and T. Kumar, Control and path planning of quadrotor aerial vehicles for search and rescue, in: Proc. SICE Annual Conference, Tokyo, Japan, pp. 700705, 2011.

[11] R. Beard, T. McLain, D. Nelson D.B. Kingston, and D. Johanson, Decentralized cooperative aerial surveillance using fixed-wing miniature uavs, Proc. IEEE 94 (7) 13061324, 2006.

[12] Giulia Michieletto, Angelo Cenedese, Luca Zaccarian, Antonio Franchi. Nonlinear Control of Multi-Rotor Aerial Vehicles Based on the Zero-Moment Direction. IFAC World Congress 2017, Jul 2017, Toulouse, France.

[13] G. Michieletto, A. Cenedese, L. Zaccarian, and A. Franchi, Hierarchical nonlinear control for multi-rotor asymptotic stabilization based on zero-moment direction, *Automatica, vol. 117, p. 108991,* 2020.

[14] M. Perin, M. Bertoni, G. Michieletto, R. Oboe, and A. Cenedese, Trajectory track ing for tilted hexarotor with concurrent attitude regulation, 2024.

[15] Markus Ryll, Davide Bicego and Antonio Franchi, "Modeling and control of Fast-Hex: a fully-actuated by synchronized-tilting Hexarotor", 2016 IEEE International Conference on Intelligent Robots and Systems, October 09-14,2016, Daejeon,Korea

[16] J. Yoon and J. Doh, Optimal PID control for hovering stabilization of quadcopter using long short term memory, Advanced Engineering Informatics, vol. 53, p. 101679, 2022.

[17] O. A. Dhewa, A. Dharmawan, and T. K. Priyambodo, Model of Linear Quadratic Regulator (LQR) Control Method in Hovering State of Quadrotor, Journal of Telecommunication, Electronic and Computer Engineering, vol. 9, no. 3, pp. 135 143, 2017.

[18] Moali, O.; Mezghani, D.; Mami, A.; Oussar, A.; Nemra, A. "UAV Trajectory Tracking Using Proportional-Integral-Derivative Type-2 Fuzzy Logic Controller with Genetic Algorithm Parameter Tuning." Sensors 2024, 24, 6678.

[19] T. Ma and S. Wong, Trajectory Tracking Control for Quadrotor UAV, in Proc. IEEE Int. Conf. on Robotics and Biomimetics (ROBIO), Macau, China, 2017, pp. 1751 1756.

[20] Tomas Baca, Daniel Hert, Giuseppe Loianno, Martin Saska, Vijay Kumar, "Model predictive trajectory tracking and collision avoidance for reliable outdoor deployment of unmanned aerial vehicles", IEEE international Conference on Intelligent Robots and System, 2018

[21] S. Rajappa, M. Ryll, H. H. Bu lthoff, and A. Franchi, Modeling, control and design optimization for a fully-actuated hexarotor aerial vehicle with tilted propellers, 2015 IEEE International Conference on Robotics and Automation (ICRA), Washington State Convention Center, Seattle, Washington, May 26-30, 2015.