POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Systematic tuning of hierarchical attitude controllers with application to UAVs

Tesi di Laurea Magistrale in
Aeronautical Engineering - Ingegneria Aeronautica

Author: **Michele Labori**

Student ID: 962857
Advisor: Prof. Davide Invernizzi
Academic Year: 2022-23

Abstract

Attitude control is a fundamental challenge in the field of aerospace engineering and robotics. It involves the ability to accurately and precisely manipulate the orientation of a UAV quadrotor, spacecraft, satellite, or any other rigid body in three-dimensional space. This thesis focuses on the development and synthesis of nonlinear attitude controllers for rigid bodies and for the quadrotor Unmanned Aerial Vehicles (UAVs), using hierarchical control strategy. In the first part of the thesisa quick overview on the state-of-the-art regarding nonlinear attitude control is presented, then, the dynamics of a rigid body and multirotors are described with a focus, in this last, on how the attitude dynamic can affects the position dynamic. The second part of the thesis presents the design process for the attitude controllore, specifically a hierarchical control approach based on a coordinate-free formulation that makes of use of rotation matrices to describe the attitude is considered and illustrated in detail. The proposed control architecture includes a feedforward term, which is computed using a geometric command filtering approach that avoids analytical computations and improves tracking performance. The third part of the thesis presents a general of systematic tuning approach for the gains of the linearized hierarchical control architecture, using the mixed-sensitivity H_∞ synthesis. The fourth part of the thesis proposes the use of a geodesic attitude control law as a solution to mitigate the effects of the saturation of the propellers. The benefits of the proposed attitude control design are evaluated first in a numerical example for an ideal rigid body and, then, in a simulator of the with the quadrotor dynamic model of a quadrotor UAV implemented in MATLAB/Simulink. Finally, flight test experiments are carried out in the Flying Arena for Rotorcraft Technologies (FlyART) of the Aerospace Systems and Control Laboratory (ASCL) of Politecnico di Milano to the performance of the geodesic controller with the one obtained with a popular nonlinear control for multirotors in real case scenarios.

Keywords: Unmanned aerial vehicle, attitude control, geometric command filtering, hierarchical control, H_∞ synthesis approach, geodesic control

Sommario

Il controllo dell'assetto è una sfida fondamentale nel campo dell'ingegneria aerospaziale e della robotica. Coinvolge la capacità di manipolare con precision l'orientamento di un quadrirotore UAV, di un'astronave, di un satellite o di qualsiasi altro corpo rigido nello spazio tridimensionale. Questa tesi si concentra sullo sviluppo e sulla sintesi di controllori non lineari di assetto per corpi rigidi e per i quadrirotori UAV, utilizzando una strategia di controllo gerarchico. Nella prima parte della tesi viene presentata una panoramica veloce dello stato dell'arte riguardo al controllo non lineare dell'assetto, successivamente vengono descritte le dinamiche di un corpo rigido e dei multirotori ,con un focus di quest'ultimo su come la dinamica dell'assetto possa influenzare la dinamica della posizione. La seconda parte della tesi presenta il processo di progettazione per il controllo dell'assetto, in particolare viene considerato e illustrato dettagliatamente un approccio di controllo gerarchico basato su una formulazione senza coordinate che fa uso di matrici di rotazione per descrivere l'assetto. L'architettura di controllo proposta include un termine di feedforward, che viene calcolato utilizzando un approccio di filtraggio dei comandi geometrici che evita calcoli analitici e migliora le prestazioni di tracking. La terza parte della tesi presenta un approccio generale di sintesi sistematica per i guadagni dell'architettura di controllo gerarchico linearizzato, utilizzando la sintesi H_∞ di sensibilità mista. La quarta parte della tesi propone l'utilizzo di una legge di controllo dell'assetto geodetica come soluzione per mitigare gli effetti della saturazione delle eliche. I vantaggi del design di controllo dell'assetto proposto vengono valutati prima in un esempio numerico per un corpo rigido ideale e successivamente in un simulatore del modello dinamico del quadricottero UAV implementato in MATLAB/Simulink. Infine, vengono effettuati esperimenti di test in volo nell'Arena di Volo per Tecnologie Rotorcraft (FlyART) del Laboratorio di Sistemi Aerospaziali e Controllo (ASCL) del Politecnico di Milano per valutare le prestazioni del controllore geodetico rispetto a quello ottenuto con un popolare controllo non lineare per i multirotori in scenari reali.

Parole chiave: Drone UAV, controllo d'assetto, controllo gerarchico, filtro di comando geometrico, sintesi H_∞, controllo geodetico

Contents

1 | Introduction

1.1. Thesis overview and objectives

This thesis work deals with the attitude control problem, with specific focus on nonlinear attitude control and its application to multirotor UAV. The attitude control problem is very well known, yet it is still an active research topic, which has been addressed in various ways. In this work, a hierarchical architecture based on an inner-outer loop paradigm is considered to simplify the control design, by splitting the problem in kinematic and dynamic attitude control. Nonlinear PID-like controllers are used at the inner loop level to control the angular velocity. Two alternative nonlinear solutions borrowed from the literature are instread exploited in the outer loop for kinematic control. Both the considered strategies are coordinate-free: the attitude configuration is described using rotation matrices, which uniquely and globally identify the orientation of the rigid body in space, thereby avoiding potential issues associated with minimal parametrization (singularities) or with quaternions (ambiguity due to the double coverage). Specifically, the first solution is based on a widely adopted nonlinear stabilizer for full attitude control. The second solution is based on a geodesic feedback stabilizer, modified from the literature, which prioritizes reduced attitude control, where the objective is to control the direction of one axis along a great circle, over full attitude control.

One of the objective of this work is to exploit the latter solution for attitude control in vectored-thrust UAVs, such as quadrotors, where the stabilization of the thrust axis direction takes precedence over heading direction control. Given that the proposed control laws are nonlinear and depend upon several gains, another objective is to derive a systematic tuning method to achieve a desired level of performance.

1.2. Structure of the thesis

This thesis is organized as follows:

- In Chapter 2 an overview of the attitude control problem is presented, together

with the state-of-the-art, the mathematical modeling of the rigid body attitude dynamics and of vectored-thrust UAVs dynamics, and the problem statement for both the models.

- In Chapter 3, after the procedure of linearization of the equation presented in Chapter 2, the general formulation for the H_∞ synthesis approach is presented, where a main result of this work is also presented; finally, the tuning formulation approach is proposed also for the UAV model.
- In Chapter 4 an alternative outer loop stabilizer for the attitude control problem is presented, which is the geodesic feedback controller; it is first presented as a general formulation, inspired by [14], and, then, specialized for the UAV and rigid body models.
- In Chapter 5, the two different stabilizers are tested to solve the attitude control problem for a rigid body and to see the main advantages and differences between the two architectures.
- In Chapter 6, the tuning of the UAV controller is presented, and then simulations and experiments are presented both for the stabilizers, in order to compare the performance.

1.3. Contributions

The thesis makes several contributions. Firstly, it involves the development of a hierarchical control law for rigid bodies, which draws inspiration from [11]. The focus is here shifted to the application of this control law to UAVs. Specifically, the hierarchical architecture is implemented within the position control architecture for trajectory tracking in vectored-thrust UAVs by including, a nonlinear geometric filter, as described in [17], to address the computation of the time derivative of the desired attitude, a crucial component to improve the tracking performance.

To tune the gains of the control law, the thesis employs the H_∞ synthesis approach proposed in [12]. The nonlinear equations defining the closed-loop system of the tracking error dynamics are first linearized. This leads to a formulation that is different from the classic one used in H_∞ synthesis, where the exogenous signal is typically chosen as the reference command and the performance output as the tracking error. A proposition demonstrating the equivalence between the error and classic formulation when considering the initial conditions as the exogenous signal instead of the reference signal is derived. The method is initially introduced for the ideal rigid body problem and then adapted to

address the UAV case.

In the final part of the thesis, a geodesic control law, inspired by [14], is implemented within the proposed hierarchical architecture. The control law proposed in [14] has then been modified to incorporate error-dependent variable gains. This choice allows us to effectively prioritize reduced attitude stabilization when the reduced attitude error is large but to recover design performance when close to the desired trajectory, unlike the constant gains considered in [14]. Simulation and experimental results confirm the benefit of the geodesic stabilizer against a popular nonlinear stabilizer in reducing directionality windup issues associated with propellers saturation.

2 | Attitude control problem

This chapter reviews the state-of-the-art in the attitude control problem by discussing some existing solutions for the problem. For a detailed problem analysis, the reader can refer to [7].

Then, Section 2.2 shows first the mathematical modeling for the attitude of a generic rigid body, then the UAV mathematical model is presented, where it is underlined the importance of the attitude control for the position control.

In the end, the baseline controller is presented, which is similar to the controller depicted in work is the hierarchical control law strategy (refer to [11] for more details), also here it is possible to see first the control law for a generic rigid body and the UAV quadrotor.

2.1. State-of-the-art review

The attitude control problem for rigid bodies has been thoroughly investigated since the 1950s due to the various applications, such for example in aerospace and surveillance system. Two different subproblems can be considered:

- Full attitude control problem: A body-fixed frame is aligned with a reference frame.
- Reduced attitude control problem: Point a body-fixed vector in a specified direction in a reference frame.

Since the position dynamics of a quadrocopter can be controlled by changing the direction of the thrust axis, an attitude controller that solves the reduced attitude control problem would be sufficient to control its position.

However, many applications also demand controlling the attitude (like the yaw orientation). For this reason, various full attitude control techniques have been developed, for tracking control and attitude stabilization, with different attitude representations.

In particular, in [17] a quaternion-based command-filtered backstepping technique is proposed, where quaternions were used to represent the attitude of the vehicle, in order to

ensure global attitude tracking without singularities; while [16] presents a PID controller for trajectory tracking control, considering Euler angles as attitude representation; however, these two representations have some problem, the former is not unique while the latter is not globally defined, see [7] for more details.

A novel approach is developed in [14], where the attitude is represented by a rotation matrix, which globally and uniquely defines the orientation of a rigid body. In this work, the idea is to decouple attitude and reduced control; in particular, the reduced attitude control is steered along a geodesic path on an n-1 sphere, while the whole attitude is stabilized along SO(n); the two control laws are then coupled together in order to give a control law solving the full attitude problem.

The paper [5] introduces a model-based PD control law for attitude tracking, where the attitude error is decoupled in a reduced attitude error (misalignment with the thrust direction) and the yaw error. It is also introduced here a control allocation strategy in order to prioritize differently the correction of the reduced dynamics over the yaw error.

In [11] has been proposed a robust attitude tracking for a fully actuated rigid body, where it is exploited a non-linear control law with a hierarchical structure, which the inner loop deals with the angular velocity tracking, while the outer loop solves the attitude tracking. A similar work has been proposed [12], where the control law is much similar to [11], but introduces a systematic way to tune the gains, using the H_∞ synthesis on a linearized-closed loop system.

In the end, most recent works like [10] deals with a geometric adaptive position tracking control system for a quadrotor, here the attitude control is designed in such a way that the thrust direction is commanded independently from the yawing direction, which is irrelevant for the position tracking. Furthermore, this control law is augmented with an adaptive control terms which are used to mitigate effect of disturbances.

2.2. Mathematical Modeling

In this thesis work, the attitude control problem will be shown with an application for the UAV quadrotor, so this section must first introduce the general attitude dynamic motion and then specialize for UAV.

2.2.1. Rigid Body Motion

In order to define the equation of motion of a rigid body, consider an inertial frame and a body-fixed frame. The configuration of the rigid body is described by the orientation

of the body fixed frame concerning the inertial frame, which is represented, globally and uniquely, by the rotation matrix $R(t) := [b_1, b_2, b_3] \in SO(3)$, which is an orthogonal matrix and that satisfies:

$$R^T R = I_3, \tag{2.1}$$

where I_3 is the (3×3) identity matrix and b_i are the body axes resolved in the inertial frame. The rotation matrix also corresponds to a rotation of an angle $\theta \in \mathbb{R}$ around an axis $\hat{n}$, and it is defined as:

$$R_n(\theta) = e^{\theta \hat{n}}, \tag{2.2}$$

$\hat{n} \in \mathbb{S}^3$, this set is defined such that $\mathbb{S}^3 := \{\mathbf{x} \in \mathbb{R}^{n+1} \mid \|\mathbf{x}\| = 1\}$.

The attitude motion of the rigid body is described by Equation (2.3) :

$$\begin{cases} \dot{R} = RS(\omega) & \text{(2.3a)} \\ J\dot{\omega} = -S(\omega)J\omega + \tau_c + \tau_e, & \text{(2.3b)} \end{cases}$$

where $J = J^T \in \mathbb{R}^{3\times3}_{\geq 0}$ is the inertia matrix expressed in the body frame, $\omega \in \mathbb{R}^3$ is the body angular velocity, $\tau_c \in \mathbb{R}^3$ is the control torque exerted by the actuators and $\tau_e \in \mathbb{R}^3$ accounts for unknown exogenous effects. The map S is the map between $\mathbb{R}^3$ and $so(3)$, where it is defined as equation (2.4):

$$so(3) := \{\Omega \in \mathbb{R}^{3\times3} : \Omega = -\Omega^T\} \tag{2.4}$$

It is important to note that Equation (2.3a) cannot be integrated using standard integration techniques. To see more details about this issue, see Appendix A.1.

2.2.2. Quadrotor UAVs

A quadrotor consists of 4 rotors attached to a rigid cross airframe in a coplanar fashion, as shown in Figure 2.1.

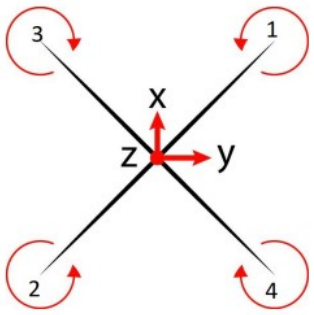

Figure 2.1: Representation of quadrotor with body-axis convention

A quadrotor is an underactuated platform because the number of inputs is 4 while the degrees of freedom are 6 (position and attitude). There are no force components that can be actively generated in the plane of the rotors; the translational degrees of freedom in the horizontal plane can be controlled through the attitude dynamics. In this particular case, the previous frame definitions are redefined as follows:

- $I = \{e_1^I, e_2^I, e_3^I\}$ is an inertial reference frame with e_3^I.
- $B = \{e_1^B, e_2^B, e_3^B\}$ is the airframe whose center coincides with the center of mass of the quadrotor such that e_3^B is in the opposite direction of thrust generation.

The attitude dynamic of a quadrotor UAV is defined as follows:

$$\begin{cases} \dot{R} = RS(\omega) & \text{(2.5a)} \\ \mathbf{G}(s)\omega = \tau_c + \tau_e, & \text{(2.5b)} \end{cases}$$

where $\mathbf{G}(s)$ is matrix of transfer function, defined as follows:

$$\mathbf{G}(s) = \begin{bmatrix} G_{roll}(s) & 0 & 0 \\ 0 & G_{pitch}(s) & 0 \\ 0 & 0 & G_{yaw}(s) \end{bmatrix}. \tag{2.6}$$

Where each element of the diagonal is a black-box model that links the control input (τ_c) with the output, which is the angular velocity (ω), which is identified with the PBSID

subspace model identification algorithm; the choice behind this identification is that the dynamics between τ_c and ω is not so simple as in the rigid body, because the quadrotor keeps into account also external effects such as aerodynamic disturbances or unmodeled dynamics. It would not be easy to resume all into a single analytical model.

The position dynamics is expressed in the inertial frame, and it is given by:

$$\begin{cases} \dot{p} = v & \text{(2.7a)} \\ m\dot{v} = mge_3 + f, & \text{(2.7b)} \end{cases}$$

where $p \in \mathbb{R}^3$ is the inertial position of the quadrotor, $v \in \mathbb{R}^3$ is the inertial velocity, m is the mass of the quadrotor and $f \in \mathbb{R}^3$ is the total force applied to the quadrotor; to see more in detail, see [6]. The force applied by the only propellers can be estimated as follows:

$$f_p = R\left(-\sum_{i=1}^{4}(T_i e_3) - c_d \sum_{i=1}^{4} \sqrt{T_i} P_{e3} v_{p_i}\right), \tag{2.8}$$

where:

$$P_{e3} = \mathbb{I}_3 - e_3 e_3^T \tag{2.9a}$$

$$e_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T. \tag{2.9b}$$

Vector $v_{p_i} \in \mathbb{R}^3$ represents the relative velocity of the hub of the i-th propeller with respect the wind velocity $v_W \in \mathbb{R}^3$:

$$v_{p_i} = R^T v_r + \omega \times d_{h_i} \tag{2.10}$$

with $v_r = v - v_W$ being the relative velocity of the center of mass and d_{h_i} being the position vector of the i-th hub with respect to the center of mass of the quadrotor. $T_i \geq 0$ is the component of the propeller force in the orthogonal direction of the rotor plane. The second term in the equation (2.8) is called $H - force$ and is a drag parasitic force due to the difference of relative air velocity between the advancing and retreating blade during the motion in a specific direction.

The expression of f_p can be simplified by recognizing that

$$T_c = \sum_{i=1}^{4} T_i \geq 0 \tag{2.11}$$

T_c is the total thrust delivered by the propellers.

Assume that $T_i \approx \frac{mg}{4}$ and that the rotors lie in the $\{b_1,b_2\}$ plane, the total force delivered by the propellers is:

$$f_p = -T_c R e_3 - R C_p R^T v_r \tag{2.12}$$

where $C_p = diag(C_{px}, C_{py}, C_{pz})$ is a damping diagonal matrix.

The first term represents the force that can be used to control the acceleration of the quadrotor in the $b_3 = Re_3$ direction (thrust axis) while the second term represents the drag, linear in velocity.

The total force applied to the quadrotor is:

$$f = f_p + f_a + f_e \tag{2.13}$$

where f_a is the parasitic drag associated with the structure of the main body:

$$f_a = -c_a |v_r| v_r \tag{2.14}$$

and f_e includes possible couplings with the attitude dynamics due to mismatches in the position of the center of mass with respect to the origin of the airframe and due to unmodeled aerodynamic effects, these last two terms can be considered as second-order effects, this force is treated as an exogenous disturbance.

Substituting (2.13) in (2.5), the equations turn out to be:

$$\begin{cases} \dot{p} = v & \text{(2.15a)} \\ m\dot{v} = mge_3 - T_c R e_3 - R C_p R^T v_r - c_a |v_r| v_r + f_e & \text{(2.15b)} \end{cases}$$

The position dynamics is underactuated because such dynamics only has one input, the total thrust T_c, but the thrust axis can be aligned in any desired direction by controlling

the direction of the thrust axis and exploiting the full actuation of the attitude dynamics. By modulating the thrust and changing the quadrotor attitude (i.e., changing b_3), the delivered force $T_c b_3$ can track any desired force vector.

The control strategy consists of:

- introduce in equation 2.15 a virtual force f_d to control position dynamics:

$$m\dot{v} = mge_3 - RC_pR^T v_r - c_a|v_r|v_r + f_e + f_d + (-f_d - T_cRe_3); \tag{2.16}$$

- Selecting $T_c = |f_d|$ and assigning as reference attitude to the attitude controller a rotation matrix R_d that has the third axis aligned with f_d, *i.e.*,

$$R_d e_3 = b_{d3} = -\frac{f_d}{|f_d|}. \tag{2.17}$$

The term $\Delta f = -f_d - T_cRe_3$ represents the mismatch between the desired virtual force and the actual force.

Under the assumption that the attitude controller ensures $b_3 \to b_{d3}$ sufficiently fast, for position control one can refer to the following model:

$$m\dot{v} = mge_3 - RC_pR^T v_r - c_a|v_r|v_r + f_e + f_d + \Delta f \tag{2.18}$$

Neglecting wind effect or collapsing it into the exogenous force, the above equation can be rewritten in this compact form:

$$m\dot{v} = mge_3 - RC_pR^T v - c_a|v|v + f_d + d(t) \tag{2.19}$$

The disturbance $d(t)$ is an exogenous term that is assumed to be bounded.

2.3. Hierarchical approach to attitude control

In this section, the main attitude control architecture is presented as baseline control, both for the rigid body and the UAV dynamics.

2.3.1. Rigid Body Attitude Control

This generic attitude control law aims for Equation (2.3) to guarantee attitude tracking in the presence of disturbances. The generic control law can be defined as static and dynamic, the latter will be better exploited during the section. For this problem, the following assumption is made to guarantee the smoothness and boundedness of the desired trajectory:

Proposition 2.1. *The desired trajectory* $t \to (R_d(t), \omega_d(t)) \in SO(3) \times \mathbb{R}^3$ *satisfies the following equation:*

$$\dot{R}_d(t) = R_d(t) S(\omega_d(t)). \tag{2.20}$$

Equation (2.20) *must hold* $\forall t > 0$ *and* $t \to \omega_d(t)$ *is continuously differentiable and uniformly bounded.*

To write the control law, let the attitude and the angular velocity error coordinates be defined as:

$$\begin{cases} R_e = R_d^T R & \text{(2.21a)} \\ \omega_e = \omega_v - \omega & \text{(2.21b)} \end{cases}$$

Then, consider the general control law formulation as follows:

$$\begin{cases} \tau_c = S(\omega) J \omega + J \dot{\omega}_v(R_e, \omega_e, t) + \gamma_\omega(x_c, \omega_e, \omega_v) & \text{(2.22a)} \\ \dot{x}_c = \gamma_c(x_c, \omega_e, \omega_v(R_e, t)). & \text{(2.22b)} \end{cases}$$

$$\begin{cases} \omega_v(R_e, t) = \gamma_R(R_e) + R_e^T \omega_d(t) & \text{(2.23a)} \\ \dot{\omega}_v(R_e, \omega_e, t) = \dot{\gamma}_R(R_e) - S(\gamma_R(R_e) - \omega_e) R_e^T \omega_d + R_e^T \dot{\omega}_d(t) & \text{(2.23b)} \end{cases}$$

In (2.22) $x_c \in \mathbb{R}^{n_c}$ represents the controller state and $\gamma_R(\cdot) : SO(3) \to \mathbb{R}^3$, while $\gamma_\omega(\cdot,\cdot,\cdot) : \mathbb{R}^{n_c} \times \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^3$, $\gamma_c(\cdot,\cdot,\cdot) : \mathbb{R}^{n_c} \times \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^3$ are continuous stabilizers to be defined. The discussion and computation about the structure of the outer loop are done in A.3.

The inner loop (Equations (2.22a) and (2.22b)) is in charge of assigning a suitable control torque τ_c to track the angular velocity reference ω_v provided by the outer loop (Equations (2.23a) and (2.23b)). Equation (2.22b) is used to resume a behaviour similar to a PID controller.

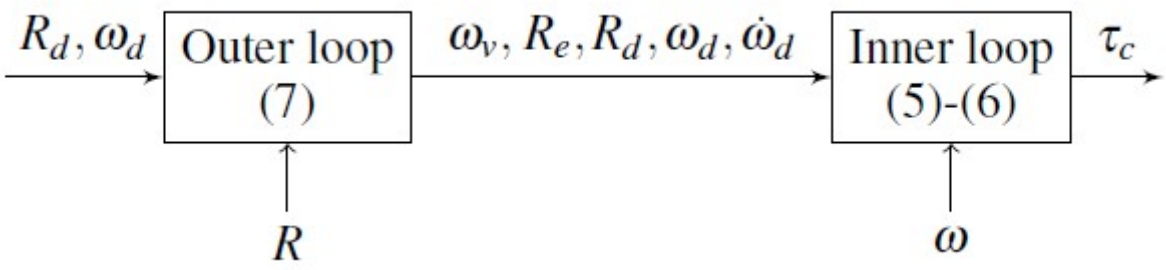

Figure 2.2: Sketch of the hierarchical controller for rigid body

Considering Equations (2.3) and (2.22a), using the definitions given by (2.21), the closed-loop dynamic errors are the following:

$$\begin{cases} \dot{R}_e = R_e S(\gamma_R(R_e) - \omega_e) & \text{(2.24a)} \\ J\dot{\omega}_e = -\gamma_\omega(x_c, \omega_e, \omega_v(R_e, t)) - \tau_e & \text{(2.24b)} \\ \dot{x}_c = \gamma_c(x_c, \omega_e, \omega_v(R_e, t)) & \text{(2.24c)} \end{cases}$$

Moreover, it is possible, by following the procedure presented in the paper [11], to retrieve, from the hierarchical control law, a cascade structure like this:

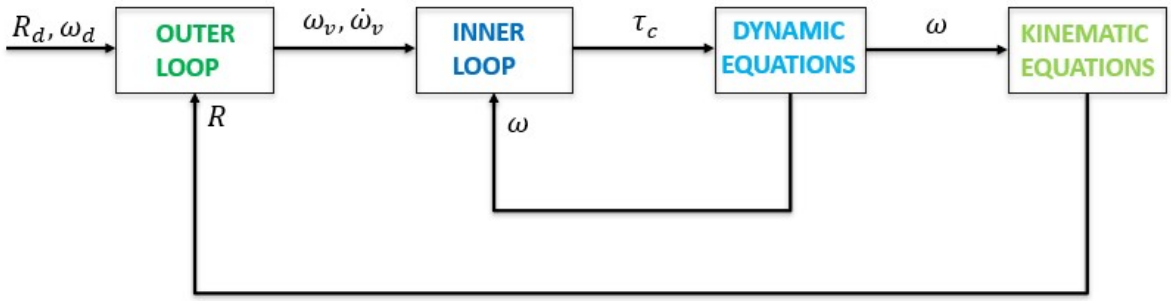

Figure 2.3: Closed-loop error dynamics: feedback interconnection

2.3.2. Outer Loop Control

This subsection presents the properties of the attitude stabilizer γ_R and how it has been chosen for this thesis work.

The attitude stabilizer $R_e \to \gamma_R(R_e)$ is at least C^2, uniformly bounded and such that:

- $R_e = I_3$ is a locally asymptotically stable equilibrium point for $\dot{R}_e = R_e S(\gamma_R(R_e))$.

- Equation (2.24a) is aISS (Appendix A.2) with respect to ω_e.

Once the assumptions are made, it is presented sample selections for the outer and inner loop stabilizers, respectively γ_R for the outer loop, γ_ω and γ_c for the inner loop.

Proposition 2.2. *Given $\gamma_R(R_e) = -S^{-1}(skew(K_r R_e))$, see paper [11], where $K_r \in \mathbb{R}^{3\times3}$ is a symmetric matrix with distinct eigenvalues satisfying $tr(K_r)I_3 - K_r \in \mathbb{R}^{3\times3}_{>0}$, the assumption 2.3.2 is satisfied.*

In order to compute $\dot{\omega}_v$, one needs to compute the time derivative of γ_R, which is given as follows:

$$\dot{\gamma}_R(R_e) = -S^{-1}(skew(K_r \dot{R}_e)). \tag{2.25}$$

By resuming the Equation (2.24a) it is possible to write:

$$\dot{\gamma}_R(R_e) = -S^{-1}(skew(K_r R_e S(\gamma_R(R_e) - \omega_e))) \tag{2.26}$$

The element inside the parenthesis can be rewritten as:

$$skew(K_r R_e S(\gamma_R(R_e) - \omega_e)) = \frac{(K_r R_e S(\gamma_R(R_e) - \omega_e) - (K_r R_e S(\gamma_R(R_e) - \omega_e)^T}{2} \tag{2.27a}$$

$$\frac{1}{2}((K_r R_e S(\gamma_R(R_e) - \omega_e) - S(\gamma_R(R_e) - \omega_e)^T (K_r R_e)^T) \tag{2.27b}$$

$$\frac{1}{2}(((K_r R_e) S(\gamma_R(R_e) - \omega_e) + S(\gamma_R(R_e) - \omega_e)(K_r R_e)^T) \tag{2.27c}$$

Retrieving the following property: $A^T S(x) + S(x)A = S((tr(A)I_3 - A)x)$, it is possible to rewrite the equation (2.27) as:

$$\frac{1}{2}((K_r R_e) S(\gamma_R(R_e) - \omega_e) + S(\gamma_R(R_e) - \omega_e)(K_r R_e)^T) = \tag{2.28a}$$

$$= \frac{1}{2}(S(tr(R_e^T K_r)I_3 - (R_e^T K_r))(\gamma_R(R_e) - \omega_e)) \tag{2.28b}$$

Finally by substituting the Equation (2.28) inside the Equation (2.26), it is possible to establish that:

$$-S^{-1}(skew(K_r R_e S(\gamma_R(R_e) - \omega_e))) = \tag{2.29a}$$

$$= -\frac{1}{2}\mathcal{S}^{-1}\mathcal{S}((tr(R_e^T K_r)I_3 - (R_e^T K_r))(\gamma_R(R_e) - \omega_e)) \tag{2.29b}$$

The term defined in Equation (2.25) is redefined such as:

$$\dot{\gamma}_R(R_e) = -\frac{1}{2}(tr(R_e^T K_r)I_3 - (R_e^T K_r))(\gamma_R(R_e) - \omega_e). \tag{2.30}$$

Here it is a sketch from Simulink of the outer loop that has been developed so far.

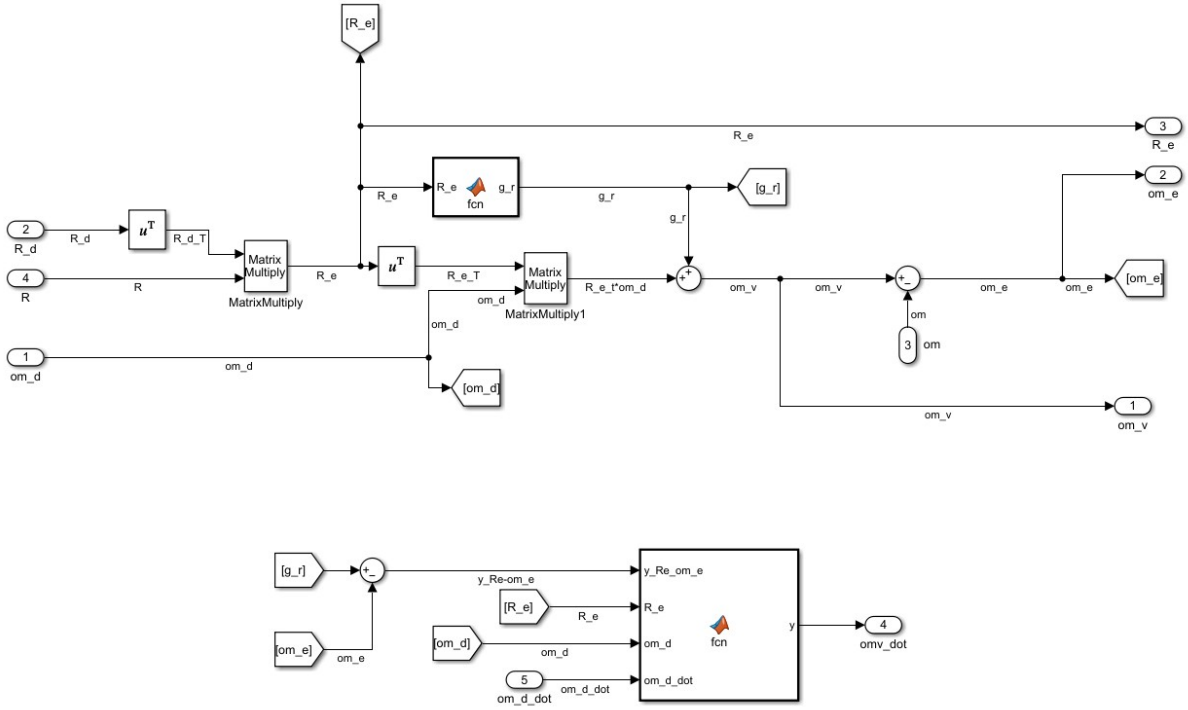

Figure 2.4: Outer loop controller: Computation of ω_v and $\dot{\omega}_v$

2.3.3. Inner loop control

After describing the outer control law, it is necessary to introduce the same idea for the inner loop controller, particularly for the stabilizers γ_c and γ_ω.

Proposition 2.3. *Let the inner loop stabilizer functions, see paper [11], for the Equations*

(2.22a) *and 2.22b:*

$$\begin{cases} \gamma_c(x_c, \omega_e, \omega_v) = A_c x_c + B_c \omega_e & (2.31a) \\ \gamma_\omega(x_c, \omega_e, \omega_v) = C_c x_c + (D_c + K_\omega)\omega_e + S(\omega_e)J(\omega_v - \omega_e). & (2.31b) \end{cases}$$

Suppose that $G_\omega(s) = C_c(sI_{nc} - A_c)^{-1}B_c + D_c$ *is a* $n_c \times n_c$ *positive real transfer function, where* (A_c, B_c) *and* (A_c, C_c) *are controllable and observable pairs, respectively. Then, for any* $K_\omega \in \mathbb{R}^{3\times3}_{>0}$ *and any* $\omega_M > 0$*, the couple* $(\omega_e, x_c) = (0, 0)$ *is a GAS equilibrium.*

By substituting the Equations (2.31b) inside the (2.22a), it is possible to write the control torque as:

$$\tau_c^{prop} = S(\omega_v)J\omega_v + J\dot{\omega}_v + C_c x_c + (D_c + K_\omega)\omega_e. \tag{2.32}$$

The first term of the equation (2.32) does not cancel the gyroscopic term $-S(\omega)J\omega$: this may be helpful in practical implementations to reduce the risk of actuators saturation, e.g., when the desired angular velocity ω_d is much smaller than the initial angular velocity.

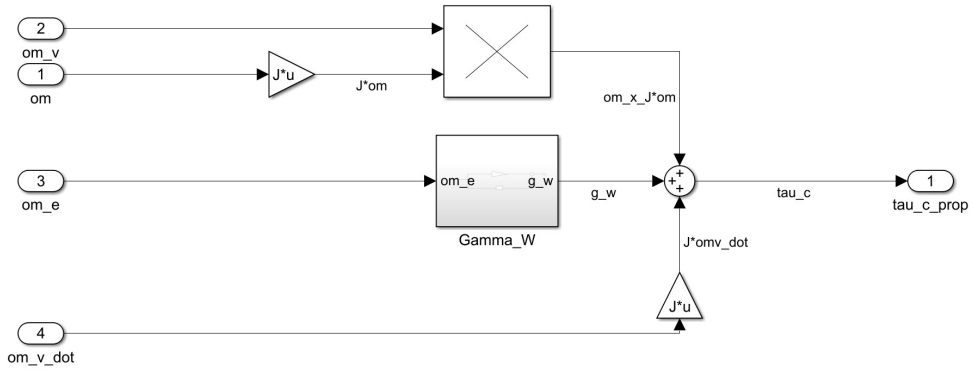

Figure 2.5: Inner loop controller: Computation of τ_c

In Chapter 5, a confrontation is presented between the control torque computed as (2.32) (called ″Proposed controller″) within the other 3 control torque variants, which are the following:

- **Feedback Linearizing**

  This control torque differs from the ″Proposed″ one (2.32) from the fact that the first term cancels the gyroscopic term, so the definition is such as follows:

$$\tau_c^{fl} = S(\omega)J\omega_v + J\dot{\omega}_v + C_c x_c + (D_c + K_\omega)\omega_e. \tag{2.33}$$

- **Saturated**

  This control torque takes into account the effect of possible saturation of the actuators, and it is defined as:

$$\tau_c^{sat} = \sigma_M(\tau_c^{prop}). \tag{2.34}$$

  where $\sigma_M(x) = (sat_M(x_1), sat_M(x_2), sat_M(x_3))$ for $x = (x1, x2, x3)$, where $sat_M(\cdot) = min(max(\cdot, -M), M)$ for $M \in \mathbb{R}_{>0}$.

- **PD-like**

  This control torque architecture has a PD-like behavior it is defined as follows:

$$\tau_c^{pd} = S(R_e^T\omega_d)JR_e^T\omega_d + JR_e^T\dot{\omega}_d + \gamma_R(R_e) + K_\omega(R_e^T\omega_d - \omega). \tag{2.35}$$

2.3.4. UAV Attitude Control

This subsection analyzes a possible implementation of the hierarchical control law adapted to the ANT-X quadrotor.

The main two differences between this control law and the one developed in Subsection 2.3.1 are the Inner loop control and the fact that only ω_v is computed in the outer loop regulator; the reason behind this choice will be explained in detail in Subsection 2.3.5.

The outer loop regulator is defined into Equation 2.23a with the choice of γ_R reported in equation (2.2).

The inner loop control law is a 2-DOF PID controller, defined as follows:

$$T_c(s) = \left(K_p + \frac{K_i}{s}\right)(\Omega_v(s) - \Omega(s)) + \frac{K_d s}{T_f s + 1}(-\Omega(s)), \tag{2.36}$$

where $T_c(s)$, $\Omega_v(s)$ and $\Omega(s)$ are respectively the Laplace transform related to the τ_c, ω_v and ω; K_p,K_i and K_d are tuning diagonal 3-by-3 matrixes, while T_f is the time bandwidth of the filter for the derivative action.

This simple architecture is easy to implement into multirotor control dynamics. It al-

lows for a good performance level, concerning the typical angular velocity dynamic that characterize the multirotors.

Here is a simulink sketch of the UAV inner loop regulator.

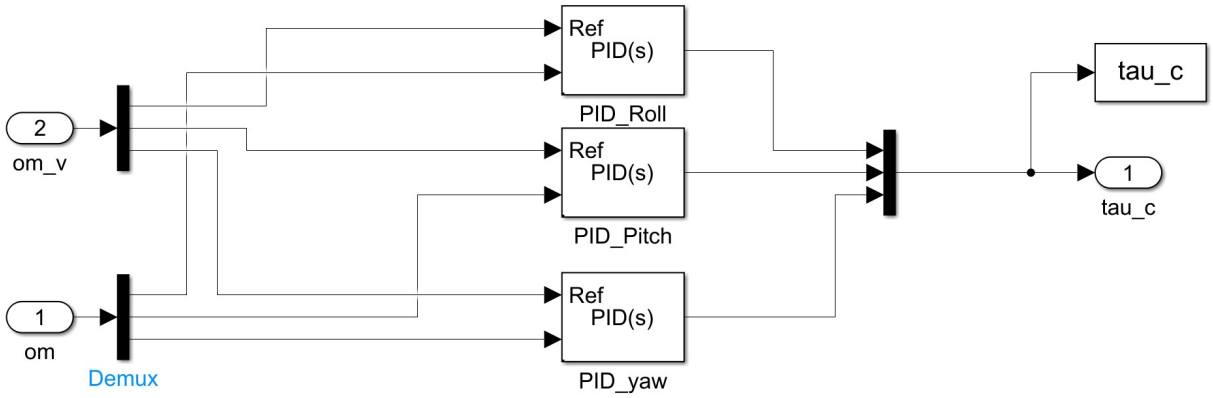

Figure 2.6: UAV Inner loop controller: Computation of τ_c

2.3.5. Command Filtering

This subsection wants to introduce one of the issue of using the architecture defined for the rigid body inside the UAV Quadrotor and a possible solution. This procedure has been followed also in [17] and [12].

In the rigid body context, it is possible to describe the control law by using a smooth function of ω_d and starting from the desired angular velocity it is possible to calculate R_d by using the Assumption 2.1 and $\dot{\omega}_d$ can be computed analytically.

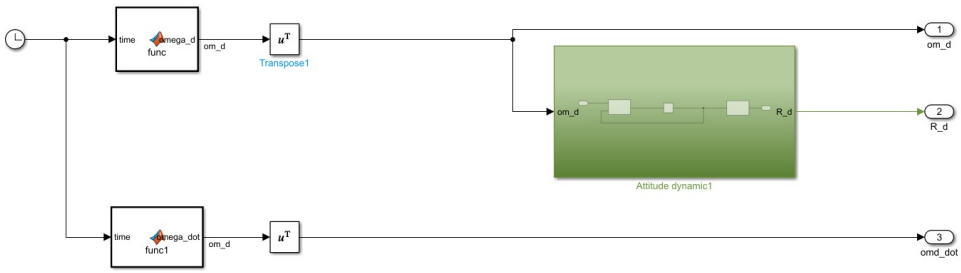

Figure 2.7: Definition of the desired input (Rigid Body): Computation of ω_d, $\dot{\omega}_d$ and R_d

R_d is the desired rotation matrix and, for a UAV quadrotor, it depends strongly on the desired force f_d, as explained in 2.2.2, so the computation of the time derivative of ω_d is possible, but this computation can be convoluted if the position controller's expression is too complicated and it depends upon the formulation of the controller.

A possible solution, inspired by [12], developed in this work is a non-linear second-order filter that considers the desired attitude R_d and where the output is ω_d.

Let R_d^f and ω_d^f be, respectively, the filtered desired attitude and angular velocity, then it is possible to define:

$$\begin{cases} \dot{R}_d^f = R_d^f S(\omega_d^f) & \text{(2.37a)} \\ \dot{\omega}_d^f = -\omega_n^2 S^{-1}(skew(R_e)) - 2\xi\omega_n\omega_d^f & \text{(2.37b)} \end{cases}$$

where:

$$R_e^f = R_d^T R_d^f \tag{2.38}$$

ω_n is the natural frequency of the filter, while the ξ is the damping ratio; this last parameter is fixed to 1 in order not to obtain a large overshoot around the natural frequency; this value will last for all the simulations and experiment; while the ω_n must be chosen to make a trade-off following this two reasonings:

- ω_n needs to be sufficiently fast to make the computation of the derivative, this information comes from the velocity of the position control to track correctly the desired attitude.
- Increasing the value of ω_n leads to absorbing a high-frequency signal inside the computation of the derivative, which leads to a dirty signal.

Once the ω_d is computed, then it is used to calculate ω_v (Equation 2.23a).

The Simulink model corresponding to (2.37) is reported in Figure 2.7:

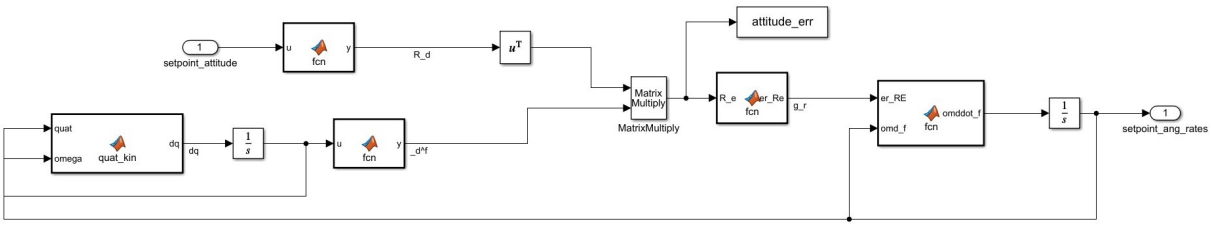

Figure 2.8: Command Filter: Computation of ω_d^f

The geometric filter presented in Figure 2.7 can track sufficiently well ω_d, but not its derivative $\dot{\omega}_d$; this is because the desired acceleration would be discontinuous, and this is the reason behind the choice to use a PID-2 inside the UAV inner-loop regulator; another

solution for this issue could be using a third-order filter, for more details see [9], instead of using the one considered for in the rigid body (Equations 2.23a and 2.23b).

3 | Systematic tuning of nonlinear attitude controllers

This chapter just presents a systematic approach to tune the controller's gain; the proposed method is a slight modification of the formulation seen in [12]. In order to develop the tuning technique, linearization of the closed-loop system is needed; this is what has been done in the first subsection, where the system of equations seen in Chapter 2 are linearized around a certain equilibrium; then, a general formulation is proposed for the tuning of the gains leveraging the H_∞ control framework for the model of a rigid body (2.3). In the end of the chapter, the synthesis approach is derived for the UAV Quadrotor, this approach aims to find the optimal gain, that subsequently, will be used to test the non-linear control law in a dedicated numerical simulator and, then, the UAV application.

3.1. Linearization of the hierarchical controller

3.1.1. Hypotesis and linearization of the attitude motion

Let $R \in \mathrm{SO}(3)$ and $\omega \in \mathbb{R}^3$ be, respectively, the rotation matrix and the angular velocity of a generic rigid body, then the hypothesis for the linearization are as follows:

Proposition 3.1. *Assume that as point of equilibrium is:*

$$\begin{cases} R = I_3 & \text{(3.1a)} \\ \omega = 0. & \text{(3.1b)} \end{cases}$$

Then, using the Rodrigues' formula the linearized attitude motion is such that $\omega \approx 0$ and the rotation matrix (defined in Equation 2.3) around the equilibrium can be rewritten such as:

$$R_u(\theta) = I_3 + sin(\theta)\ S(\hat{n}) + (1 - cos(\theta))\ S(\hat{n})^2 \approx \tag{3.2}$$

$$\approx I_3 + \theta\ S(\hat{n}) = I_3 + S(\hat{\theta}). \tag{3.3}$$

$\hat{\theta} \in \mathbb{R}^3$ is the vector of small rotation angles about $R = I_3$. This assumption is useful for the quadrotor, while the rigid body can be linearized for a generic couple of R_d and ω_d.

In this subsection, the linearization is applied to the Equations (2.3), (2.21) and for the dynamic equation defined in Proposition 2.1, as follows:

- **Kynematic equation** 2.3a:

$$\dot{(I_3 + S(\hat{\theta}))} = (I_3 + S(\hat{\theta}))S(\omega) \tag{3.4}$$

$$\dot{(\cancelto{0}{I_3} + S(\hat{\theta}))} = S(\omega) + \cancelto{\mathbf{0}}{S(\hat{\theta})S(\omega)} \tag{3.5}$$

$$S(\dot{\hat{\theta}}) = S(\omega) \tag{3.6}$$

$$\cancel{S^{-1}}S(\dot{\hat{\theta}}) = \cancel{S^{-1}}S(\omega) \tag{3.7}$$

$$\dot{\hat{\theta}} = \omega. \tag{3.8}$$

  So, as a result of the linearization, the equation (2.3a), under the assumption of small attitude motion is given by three independent differential equations corresponding to planar rotation about each axis.

- **Dynamic Equation** 2.3b: Equation (2.3*b*) is very straightforward to treat because the only non-linear term present is the gyroscopic term, an infinitesimal second-order term ($O(\omega^2)$):

$$J\dot{\omega} = \cancelto{0}{-S(\omega)J\omega} + \tau_c + \tau_e = \tau_c + \tau_e. \tag{3.9}$$

- **Attitude and angular velocity error**: Before starting with the linearization, let's define θ_d as the desired small rotation angle about $R_d = I_3$. The angular velocity

error is already linear, while the attitude error can be exploited as:

$$I_3 + S(\hat{\theta}_e) = (I_3 + S(\hat{\theta}_d)^T)(I_3 + S(\hat{\theta})) \tag{3.10}$$

$$\cancel{I_3} + S(\hat{\theta}_e) = \cancel{I_3} + S(\hat{\theta}_d)^T + S(\hat{\theta}_e) + \cancel{S(\hat{\theta}_d)^T S(\hat{\theta})} \tag{3.11}$$

$$S(\hat{\theta}_e) = S(\hat{\theta}_d)^T + S(\hat{\theta}) \tag{3.12}$$

$$S(\hat{\theta}_e) = -S(\hat{\theta}_d) + S(\hat{\theta}) \tag{3.13}$$

$$S(\hat{\theta}_e) = S(\hat{\theta}_d - \hat{\theta}) \tag{3.14}$$

$$\cancel{S^{-1}} S(\hat{\theta}_e) = -\cancel{S^{-1}} S(\hat{\theta}_d - \hat{\theta}) \tag{3.15}$$

$$\hat{\theta}_e = -(\hat{\theta}_d - \hat{\theta}) = -\bar{\theta}_e. \tag{3.16}$$

This definition holds only when it is considered small angles and small desired angle; in general, even if θ_e is small, the attitude and the desired attitude can be very far from the identity. The linearized attitude motion is written as:

$$\begin{cases} \dot{\hat{\theta}} = \omega, & \text{(3.17a)} \\ J\dot{\omega} = \tau_c + \tau_e. & \text{(3.17b)} \end{cases}$$

For the UAV Quadrotor (Equation 2.5), the linearization can be simply written as:

$$\begin{cases} \dot{\hat{\theta}} = \omega, & \text{(3.18a)} \\ \mathbf{G}(s)\omega = \tau_c + \tau_e. & \text{(3.18b)} \end{cases}$$

The linearized attitude and angular velocity errors are defined as:

$$\begin{cases} \hat{\theta}_e = -(\hat{\theta}_d - \hat{\theta}) = -\bar{\theta}_e, & \text{(3.19)} \\ \omega_e = \omega_v - \omega. & \text{(3.20)} \end{cases}$$

Thanks to the above derivations, it is possible now to linearize the control law in the function of the variables defined in 3.19 and 3.20.

3.1.2. Linearization of the control law and the closed-loop error dynamic

Once the attitude motion and attitude error coordinates are linearized (Subsection 3.1.1), it is time to linearize the Equations 2.23a and 2.23b (outer loop stabilizer) with the choice

made in 2.2 and Equations 2.22a and 2.22b (inner loop stabilizer) with the choice made in 2.3.

In this section, the linearization of the closed-loop dynamic is developed (Equation 2.24), which will be essential to demonstrate a particular result that will be presented in Subsection 3.2.1.

- **Linearization of ω_v:** By looking at Equations (2.23a), the formula is composed of two terms, so the linearization will be computed singularly for each term:

  1. **First term:** The first step is to linearize the term inside the brackets of 2.2:

$$skew(K_R R_e) = \frac{(K_R R_e) - (K_R R_e)^T}{2} \tag{3.21}$$

$$skew(K_R R_e) = \frac{K_R R_e - R_e^T K_R}{2} \tag{3.22}$$

$$skew(K_R R_e) = \frac{K_R(I_3 + S(\hat{\theta}_e)) - (I_3 + S(\hat{\theta}_e))^T K_R}{2} \tag{3.23}$$

$$skew(K_R R_e) = \frac{K_R + K_R S(\hat{\theta}_e) - (K_R + S(\hat{\theta}_e)^T K_R)}{2} \tag{3.24}$$

$$skew(K_R R_e) = \frac{\cancel{K_R} + K_R S(\hat{\theta}_e) - \cancel{K_R} - S(\hat{\theta}_e)^T K_R}{2} \tag{3.25}$$

$$skew(K_R R_e) = \frac{K_R S(\hat{\theta}_e) + S(\hat{\theta}_e) K_R}{2}. \tag{3.26}$$

  At this point, it is useful to exploit this property, $A^T S(x) + S(x)A = S((tr(A)I_3 - A)x)$, to rewrite the equation 3.21 in the following way:

$$skew(K_R R_e) = \frac{K_R S(\hat{\theta}_e) + S(\hat{\theta}_e) K_R}{2} = \frac{S((tr(K_R)I_3 - K_R)\hat{\theta}_e)}{2}. \tag{3.27}$$

  Now, it is possible to substitute the last equation inside the Equation presented in (2.2):

$$\gamma_R(R_e) = -\frac{1}{2}\cancel{S^{-1}}\cancel{S}((tr(K_R)I_3 - K_R)\hat{\theta}_e \tag{3.28a}$$

$$\gamma_R(R_e) = -\frac{1}{2}((tr(K_R)I_3 - K_R)\hat{\theta}_e \tag{3.28b}$$

$$\gamma_R(R_e) = -\bar{K}_R\hat{\theta}_e = \bar{K}_R\bar{\theta}_e. \tag{3.28c}$$

  The matrix $\bar{K}_R$ is defined as:

$$\bar{K}_R = \begin{bmatrix} \frac{K_{R_2}+K_{R_3}}{2} & 0 & 0 \\ 0 & \frac{K_{R_1}+K_{R_3}}{2} & 0 \\ 0 & 0 & \frac{K_{R_1}+K_{R_2}}{2}. \end{bmatrix}$$

2. **Second term:** The second term is straightforward with respect to the previous one, indeed:

$$R_e^T \omega_d = (I_3 + S(\hat{\theta}_e))^T \omega_d = \omega_d + \cancelto{\mathbf{0}}{S(\hat{\theta}_e)^T \omega_d} = \omega_d. \tag{3.29}$$

These results can also be obtained, without the assumption of small attitude motion, by requiring that the error angle and the desired angle are parallel, for example, in the case of a single-axis maneuver.

So, once all the terms are linearized, it is possible to write the combined linearized version of the first equation of (2.23a):

$$\omega_v = -\bar{K}_R \hat{\theta}_e + \omega_d = \bar{K}_R \bar{\theta}_e + \omega_d. \tag{3.30}$$

As it is possible to notice, ω_v in the linearized version becomes a Proportional Controller perturbed by the desired angular velocity ω_d.

- **Linearization of $\dot{\omega}_v$:** Recalling Equation (2.23b), it is possible to see that it is defined by three terms, and, also, in this case, they will develop singularly:

1. **First Term:** To develop this term, can be used an approach similar to the linearization made in (3.21), with the help of the Equation (2.24a), so the first passage is to develop what is inside the round brackets:

$$skew(K_R \dot{R}_e) = \frac{(K_R \dot{R}_e) - (K_R \dot{R}_e)^T}{2} \tag{3.31}$$

$$skew(K_R \dot{R}_e) = \frac{K_R \dot{R}_e - \dot{R}_e^T K_R}{2} \tag{3.32}$$

$$skew(K_R \dot{R}_e) = \frac{K_R R_e S(\gamma_R - \omega_e) - (R_e S(\gamma_R - \omega_e))^T K_R}{2} \tag{3.33}$$

$$skew(K_R \dot{R}_e) = \frac{K_R R_e S(\gamma_R - \omega_e) - S(\gamma_R - \omega_e)^T R_e^T K_R}{2} \tag{3.34}$$

$$skew(K_R \dot{R}_e) = \frac{K_R R_e S(\gamma_R - \omega_e) + S(\gamma_R - \omega_e) R_e^T K_R}{2}. \tag{3.35}$$

By using the same property used before, during the linearization of ω_v, $(A^T S(x)+$

$S(x)A = S((tr(A)I_3 - A)x))$, it is possible to obtain:

$$\frac{K_R R_e S(\gamma_R - \omega_e) + S(\gamma_R - \omega_e) R_e^T K_R}{2} = \frac{S((tr(R_e^T K_R) I_3 - R_e^T K_R)(\gamma_R - \omega_e))}{2}. \tag{3.36}$$

Now, it is possible to put this expression in the definition of $\dot{\gamma}_R$:

$$\dot{\gamma}_R(R_e) = -\frac{1}{2}\cancel{S^{-1}} S((tr(R_e^T K_R) I_3 - R_e^T K_R)(\gamma_R - \omega_e)) \tag{3.37}$$

$$\dot{\gamma}_R(R_e) = -\frac{1}{2}(tr(R_e^T K_R) I_3 - R_e^T K_R)(\gamma_R - \omega_e). \tag{3.38}$$

The term $(\gamma_R - \omega_e)$ can be developed and written as:

$$\gamma_R - \omega_e = \cancel{\omega_v} - R_e^T \omega_d - \cancel{\omega_v} + \omega \tag{3.39}$$

$$\gamma_R - \omega_e = \omega - R_e^T \omega_d \tag{3.40}$$

$$\gamma_R - \omega_e = \omega - (I_3 + S(\hat{\theta}_e)^T)\omega_d \tag{3.41}$$

$$\gamma_R - \omega_e = \omega - \omega_d - S(\hat{\theta}_e)^T \omega_d \tag{3.42}$$

$$\gamma_R - \omega_e = \omega - \omega_d + \cancel{S(\hat{\theta}_e)\omega_d}^{\,0} \tag{3.43}$$

$$\gamma_R - \omega_e = \omega - \omega_d = -(\omega_d - \omega) = -\bar{\omega}_e. \tag{3.44}$$

Once the term $(\gamma_R - \omega_e)$ is linearized, the term (3.37) can be further computed:

$$\dot{\gamma}_R = -\frac{1}{2}(tr(R_e^T K_R) I_3 - R_e^T K_R)\ (\gamma_R - \omega_e) \tag{3.45}$$

$$\dot{\gamma}_R = \frac{1}{2}(tr(R_e^T K_R) I_3 - R_e^T K_R)\ \bar{\omega}_e \tag{3.46}$$

$$\dot{\gamma}_R = \frac{1}{2}(tr((I_3 + S(\hat{\theta}_e))^T K_R) I_3 - (I_3 + S(\hat{\theta}_e))^T K_R)\ \bar{\omega}_e \tag{3.47}$$

$$\dot{\gamma}_R = \frac{1}{2}(tr(K_R + S(\hat{\theta}_e)^T K_R) I_3 - K_R + S(\hat{\theta}_e)^T K_R)\ \bar{\omega}_e \tag{3.48}$$

$$\dot{\gamma}_R = \frac{1}{2}(tr(K_R + \cancel{S(\hat{\theta}_e)^T K_R}^{\,0}) I_3 - K_R + \cancel{S(\hat{\theta}_e)^T K_R}^{\,0})\ \bar{\omega}_e \tag{3.49}$$

$$\dot{\gamma}_R = \frac{1}{2}(tr(K_R) I_3 - K_R)\ \bar{\omega}_e \tag{3.50}$$

$$\dot{\gamma}_R = \bar{K}_R\ \bar{\omega}_e. \tag{3.51}$$

The term $S(\hat{\theta}_e)$, in Equation (3.48) is small in confront to the constant term K_R, so it is assumed to be null.

2. **Second Term:** This term does not require any calculation because, when it is considered small angles and small attitude angles, it becomes a third-order infinitesimal term, and it is negligible with respect to the first two terms:

$$S(\gamma_R(R_e) - \omega_e)\ R_e^T\ \omega_d \approx 0. \tag{3.52}$$

3. **Third Term:** The last term has a similar procedure to the one shown in Equation 3.29:

$$R_e^T\ \dot{\omega}_d = (I_3 + S(\hat{\theta}_e))^T\ \dot{\omega}_d = \dot{\omega}_d + \cancelto{0}{S(\hat{\theta}_e)^T \omega_d} = \dot{\omega}_d. \tag{3.53}$$

This approximation can be obtained when $\dot{\omega}_d$ is small or when θ_e and $\dot{\omega}_d$ are parallel with each other.

So, the resulting equations coming from this procedure of linearization of the outer-loop stabilizer control are:

$$\begin{cases} \omega_v = \bar{K}_R\ \bar{\theta}_e + \omega_d & \text{(3.54a)} \\ \dot{\omega}_v = \bar{K}_R\ \bar{\omega}_e + \dot{\omega}_d. & \text{(3.54b)} \end{cases}$$

As it is possible to see, the equation (3.54a) is the time derivative of the equation (3.54b), indeed $\dot{\omega}_v$ could have been calculated by differentiation in time of ω_v (Equation (3.54a)).

The inner loop stabilizer is defined by the equations (2.22a) and (2.22b), with the use of the sample stabilizers defined in Proposition 2.3.

As it is possible to notice, this procedure is very straightforward because the only non-linearity coming from this expression is the gyroscopic term, so:

$$\begin{cases} \dot{x}_c = A_c x_c + B_c \omega_e, & \text{(3.55a)} \\ \tau_c = \cancelto{0}{S(\omega_v) J \omega} + J\dot{\omega}_v + C_c x_c + (D_c + K_\omega)\omega_e = J\dot{\omega}_v + C_c x_c + (D_c + K_\omega)\ \omega_e. & \text{(3.55b)} \end{cases}$$

In the linearized version, the inner loop stabilizer assumes the form of a PI-like controller with a feed-forward effect (ω_v).

Now, the last thing to do is to linearize the closed-error dynamics, in particular the Equations 2.24, using the choice seen in Propositions 2.2 and 2.3.

- **Attitude error dynamic**: This is the computation for the Equation 2.24a, and,

as the first passage, let's substitute the linearized rotation error matrix and the Equation 3.28c:

$$(I_3 + \dot{S}(\hat{\theta}_e)) = (I_3 + S(\hat{\theta}_e))S(-\bar{K}_R\hat{\theta}_e - \omega_e) \tag{3.56a}$$
$$S(\dot{\hat{\theta}}_e) = S(-\bar{K}_R\hat{\theta}_e - \omega_e) + \cancel{S(\hat{\theta}_e)S(-\bar{K}_R\hat{\theta}_e - \omega_e)} \tag{3.56b}$$
$$\not{S}(\dot{\hat{\theta}}_e) = \not{S}(-\bar{K}_R\hat{\theta}_e - \omega_e) \tag{3.56c}$$
$$\dot{\hat{\theta}}_e = -\bar{K}_R\hat{\theta}_e - \omega_e \tag{3.56d}$$
$$-\dot{\bar{\theta}}_e = \bar{K}_R\bar{\theta}_e - \omega_e \tag{3.56e}$$
$$\dot{\bar{\theta}}_e = -\bar{K}_R\bar{\theta}_e + \omega_e. \tag{3.56f}$$

So, the Equation (2.24a) is a first-order linear system, with an input ω_e. It must be underlined that in Equation 3.56b, the second term is canceled since it is an infinitesimal of greater order with respect to the other term.

- **Angular velocity error dynamic**: The linearization of the Equation (2.24b) simply results in cancelling the gyroscopic term, so:

$$J\dot{\omega}_e = -C_c x_c - (D_c + K_\omega)\omega_e - \cancel{S(\omega_e)J(\omega_v - \omega_e)} - \tau_e \tag{3.57a}$$
$$J\dot{\omega}_e = -C_c x_c - (D_c + K_\omega)\omega_e - \tau_e. \tag{3.57b}$$

So, the resulting closed-loop linear dynamic equations are the following:

$$\begin{cases} \dot{\bar{\theta}}_e = -\bar{K}_R\bar{\theta}_e + \omega_e & (3.58a) \\ J\dot{\omega}_e = -C_c x_c - (D_c + K_\omega)\omega_e - \tau_e & (3.58b) \\ \dot{x}_c = A_c x_c + B_c\omega_e. & (3.58c) \end{cases}$$

3.1.3. Linearization of the command filter

The linearization of the geometric command filter, defined in 2.37, is given by the following procedure:

- **First equation**: Assuming a small desired attitude motion (with the same reasoning used in Proposition 3.1), the procedure is like the one presented in equations (3.10), so the result is:

$$\dot{\theta}_d^f = \omega_d^f. \tag{3.59}$$

- **Second equation**: The Equation 2.37b is developed similarly to the procedure presented for Equations 3.21; given the first term of the equation as:

$$-\omega_n^2 S^{-1}(skew(\tilde{R}_e^f)), \tag{3.60}$$

The term inside the brackets can be rewritten as:

$$skew(R_e^f) = \frac{R_e^f - R_e^{fT}}{2} \tag{3.61}$$

$$skew(R_e^f) = \frac{(I_3 + S(\hat{\theta}_e^f)) - (I_3 + S(\hat{\theta}_e^f))^T}{2} \tag{3.62}$$

$$skew(R_e^f) = \frac{I_3 + S(\hat{\theta}_e^f) - (I_3 + S(\hat{\theta}_e^f)^T)}{2} \tag{3.63}$$

$$skew(R_e^f) = \frac{\cancel{I_3} + S(\hat{\theta}_e^f) - \cancel{I_3} - S(\hat{\theta}_e^f)^T}{2} \tag{3.64}$$

$$skew(R_e^f) = \frac{S(\hat{\theta}_e^f) + S(\hat{\theta}_e^f)}{2} = S(\hat{\theta}_e^f). \tag{3.65}$$

By inserting the last term of Equation (3.27) inside the Equation (3.21), it is possible to obtain:

$$-\omega_n^2 \cancel{S^{-1}} \cancel{S}(\hat{\theta}_e^f) = -\omega_n^2 \hat{\theta}_e^f. \tag{3.66}$$

$\hat{\theta}_e^f$ can be easily computed by following the same procedure shown in Equation (3.10), using the Definition (2.38), the result is:

$$\hat{\theta}_e^f = -(\hat{\theta}_d - \hat{\theta}_d^f) = -\bar{\theta}_e^f. \tag{3.67}$$

Therefore, it is possible to write the entire system as follows:

$$\begin{cases} \dot{\theta}_d^f = \omega_d^f & \text{(3.68a)} \\ \dot{\omega}_d^f = -\omega_n^2(\hat{\theta}_d - \hat{\theta}_d^f) - 2\xi\omega_n\omega_d^f. & \text{(3.68b)} \end{cases}$$

It is possible to notice that the above Equation can also be seen as:

$$F_\omega(s) = \frac{\omega_d^f}{\theta_d} = \frac{s\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}. \tag{3.69}$$

Equation 3.69 is a derivator cascaded with a second-order filter, so its task is to make the derivative of the desired angle, while the second-order filter reduces the high-frequency disturbances to make a less noisy computation of the derivative; although, it can be an upper limit for the signal velocity the can be derived correctly.

3.2. H_∞ synthesis systematic approach

In this section, the general formulation for the H_∞ is presented and, then, it will be shown that, starting from the error dynamic, the synthesis problem can be formulated using a classic approach. Finally, the synthesis approach will be applicated to the case of UAV quadrotor, with some differences from the rigid body framework.

3.2.1. General approach and formulation for a rigid body

The main goal of the linearized version of the generic control law presented in 3.1 is to track properly the desired setpoint, so the H_∞ approach (inspired by [12]) is, firstly, used to make the error dynamic asymptotically stable; then, it is possible to add requirements on the performances and control effort moderation, in particular, in the case of transient response, starting from the non-null initial condition. Along each axis, the linearized dynamics of a rigid body can be written in Laplace form as follows:

$$G(s) = \frac{Q(s)}{\tau_C(s)} = \frac{1}{Js} \tag{3.70}$$

For the tuning process, assume the following setup:

- Zero desired angular velocity and acceleration
- A disturbance $d(t)$ is added after the kinematic Equation 3.18a, such that:

$$\bar{\theta}_{e,pt} = \bar{\theta}_d - (\bar{\theta} + d) \tag{3.71a}$$
$$\bar{\theta}_{e,pt} = (\bar{\theta}_d - \bar{\theta}) - d \tag{3.71b}$$
$$\bar{\theta}_{e,pt} = \bar{\theta}_e - d \tag{3.71c}$$

where $\bar{\theta}_{e,pt}$ is the angle of the rigid body perturbed by the presence of the distur-

bance.

This disturbance propagates along the control system by affecting also the angular velocity error ($\omega_{e,pt}$), which is defined as:

$$\omega_{e,pt} = \omega_{v,pt} - \omega \tag{3.72a}$$
$$\omega_{e,pt} = \bar{K}_R \bar{\theta}_{e,pt} - \omega \tag{3.72b}$$
$$\omega_{e,pt} = \bar{K}_R (\bar{\theta}_e - d) - \omega \tag{3.72c}$$
$$\omega_{e,pt} = (\bar{K}_R \bar{\theta}_e - \omega) - \bar{K}_R d \tag{3.72d}$$
$$\omega_{e,pt} = \omega_e - \bar{K}_R d \tag{3.72e}$$

So, the linearized Equations for the control law for the plant dynamics 3.17 becomes:

$$\begin{cases} \omega_v = \bar{K}_R \, \bar{\theta}_{e,pt} = \bar{K}_R \, (\bar{\theta}_e - d) & (3.73a) \\ \dot{\omega}_v = \bar{K}_R \, \bar{\omega}_e = -\bar{K}_R \, \omega & (3.73b) \\ \tau_c = J\dot{\omega}_v + C_c x_c + (D_c + K_\omega) \, \omega_{e,pt} & (3.73c) \\ \dot{x}_c = A_c \, x_c + B_c \, \omega_{e,pt} & (3.73d) \end{cases}$$

Assume, now,that the inner-loop controller corresponds to a PI-like control law, namely, that the matrices in 3.73c-3.73c are chosen as follows:

- $A_c = D_c = \mathbf{0}$
- $B_c = I_3$
- $C_c = K_I$

So, in the end, the Equation 3.73, becomes:

$$\begin{cases} \omega_v = \bar{K}_R \, (\bar{\theta}_e - d) & (3.74a) \\ \dot{\omega}_v = -\bar{K}_R \, \omega & (3.74b) \\ \tau_c = J\dot{\omega}_v + K_I x_c + K_\omega \, \omega_{e,pt} & (3.74c) \\ \dot{x}_c = \omega_{e,pt} & (3.74d) \end{cases}$$

The Simulink model for the closed-loop linearized equation are depicted in 3.1,

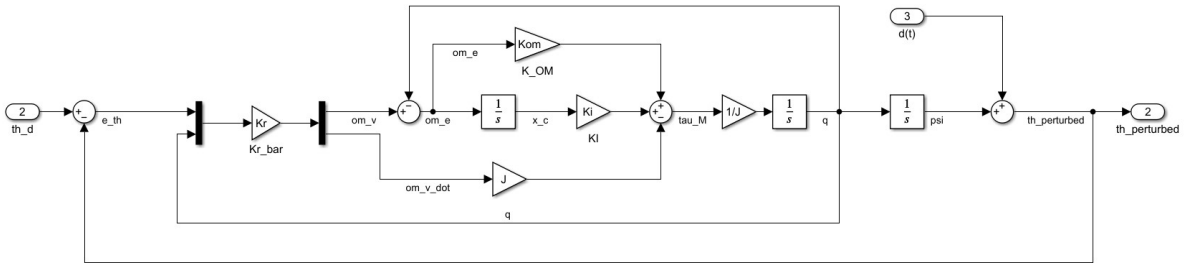

Figure 3.1: Linearized Control Architecture with plant dynamics

While the equations for the system in Figure 3.1 are,

$$\begin{cases} \omega_v = \bar{K}_R\, \bar{\theta}_{e,pt} & \text{(3.75a)} \\ J\dot{\omega} = -J(\bar{K}_R\, \omega) + K_I x_c + K_\omega\, \omega_{e,pt} & \text{(3.75b)} \\ \dot{\theta} = \omega & \text{(3.75c)} \\ \dot{x}_c = \omega_{e,pt} & \text{(3.75d)} \end{cases}$$

Control requirements for systematic H_∞ synthesis are stated in the form of weighting functions in the frequency domain. The following two requirements are typically taken into account:

- **Performance**: It is defined as the transfer function between d and $\bar{\theta}_e$, this transfer function is called "Sensitivity function" and links the error (in output) with respect to disturbances or setpoint angle (in input). For low-frequency signals in input, this transfer functions allows to reject the error efficiently.

  The weighting function is defined with the MatLab command "*makeweight*", with 4 inputs: DC,HF, [FREQ,MAG]; defined as:

  - DC: The value assumed by the transfer function at 0 rad/s ($|W(j0)| = DC$).
  - HF: The high-frequency magnitude of the transfer function ($|W(j\infty)| = HF$), and it represents also a constraint on the maximum peak for the transfer function.
  - $[FREQ, MAG]$: It's the magnitude of the transfer function for a certain frequency ($|W(jFREQ)| = MAG$). In this tuning, MAG is fixed to -3 dB, thanks to this assumption FREQ assumes the meaning of the desired sensitivity bandwidth.

- **Control Effort**: It is defined as the transfer function between d and τ_c and it is called "Control sensitivity function" and it is set in order to limit the high-frequency control action beyond the actuator capability.

  The weighting function is defined as equal as in the performance case, but here the MAG is not fixed to -3 dB; instead of a weighting function, it is possible to use also a straight horizontal line.

For a generic axis, define W_S and W_R respectively as the weighting function for performance and control effort and consider the following vector of tunable parameters:

$$\rho = [\bar{K}_R, K_I, K_\omega]^T \tag{3.76}$$

Let $S(s,\rho)$ and $R(s,\rho)$ be respectively the sensitivity and control sensitivity transfer function for the architecture, in Figure 3.1. Let the cost related to performance and control requirements be defined as:

$$\begin{cases} J_S(\rho) = \|W_S^{-1}(s)S(s,\rho)\|_\infty & \text{(3.77a)} \\ J_R(\rho) = \|W_R^{-1}(s)R(s,\rho)\|_\infty & \text{(3.77b)} \end{cases}$$

Then, it is possible to state the synthesis problem as an optimization problem:

$$\rho^* = \arg\min_\rho J_S(\rho) \tag{3.78}$$

subject to

$$J_R(\rho) \leq 1 \tag{3.79}$$

where ρ^* is the optimal value of the controller gain vector; it is possible to perform the formulation (3.78) with the MATLAB function "*Systune*".

This is a general procedure for the computation of the desired gains for a linear control architecture that acts on a rigid body.

The classic formulation described above (in which the linearization is computed about the zero equilibrium and feedforward terms are set to zero) can be used to shape the transient response also for the linearized closed-loop error dynamics, where the desired performance is specified in terms of convergence of the tracking errors to zero. Specifically, the response to non-null initial conditions on the error dynamics is shown to be equivalent

to the response to constant reference signals in the classic formulation.

Now, let us focus the analysis on the closed loop error dynamics developed in Section 3.2.1, in particular, consider Equation 3.58, by imposing the same parameter definition made for the system in figure 3.1, with the only difference that here it is assumed that also $\tau_e = 0$; it is possible to write:

$$\begin{cases} \dot{\bar{\theta}}_e = -\bar{K}_R \bar{\theta}_{e,pt} + \omega_{e,pt} & \text{(3.80a)} \\ J\dot{\omega}_e = -K_I x_c - K_\omega \omega_{e,pt} & \text{(3.80b)} \\ \dot{x}_c = \omega_{e,pt} & \text{(3.80c)} \end{cases}$$

It is possible now to write 3.80 as:

$$\begin{cases} \dot{\bar{\theta}}_e = -\bar{K}_R(\bar{\theta}_e - d) + \omega_e - \bar{K}_R d & \text{(3.81a)} \\ J\dot{\omega}_e = -K_I x_c - K_\omega(\omega_e - \bar{K}_R d) & \text{(3.81b)} \\ \dot{x}_c = \omega_e - \bar{K}_R d & \text{(3.81c)} \end{cases}$$

with the additional equation:

$$\bar{\tau}_c = -K_I x_c - K_\omega(\omega_e - \bar{K}_R d) + J\dot{\omega}_v = -K_I x_c - K_\omega(\omega_e - \bar{K}_R d) - J(\bar{K}_R \omega_e) \qquad \text{(3.82)}$$

This is the closed-loop error dynamics affected by the disturbance; The Simulink plant of the Equation 3.81 is shown in Figure 3.2:

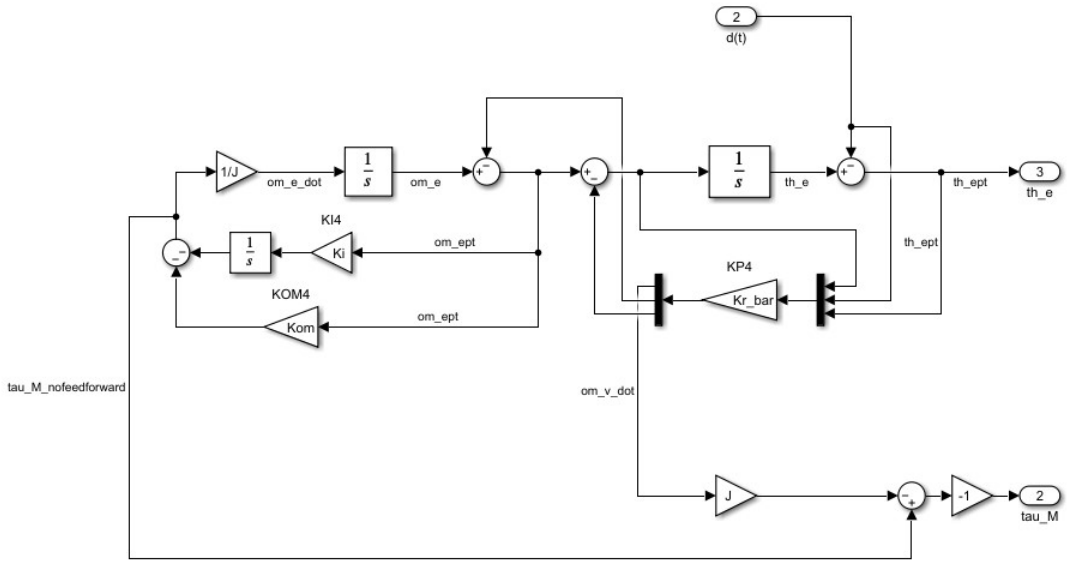

Figure 3.2: Linearized closed-loop error plant affected by disturbance

Given the two architectures developed so far, it is possible to state the following result:

Proposition 3.2. *Considering the two architectures* (3.74) *and* (3.81)*, the sensitivity functions* ($d \to \theta_e$) *and the control sensitivity functions* ($d \to \tau_c$ *in the first architecture*)*, are equivalent.*

Moreover, in the case of a step-wise disturbance input d *(or equivalently a step-wise reference* θ_d*), the time-response response obtained with the classic formulation is equivalent to considering non-null initial condition of the error angle* θ_e *in the error formulation.*

Proof. To demonstrate Proposition 3.2, we show that the classic formulation is equivalent to the error formulation.

So let us start from the classic formulation presented in Equation (3.74) applied to the rigid body formulation 3.17 (with τ_e null) which it is possible to write as:

$$\begin{cases} \omega_v = \bar{K}_R \, \bar{\theta}_{e,pt} & \text{(3.83a)} \\ \dot{\omega}_v = -\bar{K}_R \, \omega & \text{(3.83b)} \\ J\dot{\omega} = J\dot{\omega}_v + K_I x_c + K_\omega \, \omega_{e,pt} & \text{(3.83c)} \\ \dot{\theta} = \omega & \text{(3.83d)} \\ \dot{x}_c = \omega_{e,pt} & \text{(3.83e)} \end{cases}$$

- **Dynamic equation**: Starting from equation (3.83c), it is possible to rewrite the equation as:

$$J\dot{\omega} = J\dot{\omega}_v + K_I x_c + K_\omega \ \omega_{e,pt} \tag{3.84a}$$
$$J(\dot{\omega} - \dot{\omega}_v) = K_I x_c + K_\omega \ \omega_{e,pt} \tag{3.84b}$$

  And, resuming the definition (2.21b), it is possible to compute:

$$-J\dot{\omega}_e = K_I x_c + K_\omega \ \omega_{e,pt} \tag{3.85a}$$
$$J\dot{\omega}_e = -K_I x_c - K_\omega \ \omega_{e,pt} \tag{3.85b}$$

- **Kinematic equation**: Regarding the equation (3.83c),

$$\dot{\theta} = \omega \tag{3.86a}$$
$$\dot{\theta}_d - \dot{\theta} = \dot{\theta}_d - \omega \tag{3.86b}$$

  Using definition (3.19) and (2.21b), it is possible to rewrite:

$$\dot{\bar{\theta}}_e = \omega_d - \omega \tag{3.87a}$$
$$\dot{\bar{\theta}}_e = \omega_d - \omega_v + \omega_{e,pt} \tag{3.87b}$$

  ω_d is assumed to be null (there is no feedforward term during the synthesis approach) and, thanks to (3.83),

$$\dot{\bar{\theta}}_e = -\omega_v + \omega_{e,pt} \tag{3.88a}$$
$$\dot{\bar{\theta}}_e = -\bar{K}_R \bar{\theta}_{e,pt} + \omega_{e,pt} \tag{3.88b}$$

It is possible to note that, equation (3.88) and (3.85) is equivalent to the error formulation, presented in (3.58).

The next step, to better show the meaning of the external disturbance $d(t)$, is to assume that the initial condition of the error states is not null:

$$\begin{cases} \bar{\theta}_{e,0} \neq 0 & \text{(3.89a)} \\ \omega_{e,0} \neq 0 & \text{(3.89b)} \end{cases}$$

Then, it is possible to write the error equations (3.88) and (3.85) in Laplace form:

$$\begin{cases} s\bar{\Theta}_e - \bar{\Theta}_{e,0} = -\bar{K}_R\bar{\Theta}_{e,pt} + \Omega_{e,pt} & \text{(3.90a)} \\ sJ\Omega_e - J\Omega_{e,0} = -K_I X_c - K_\omega\ \Omega_{e,pt} & \text{(3.90b)} \end{cases}$$

The initial error angular velocity can be computed as:

$$\Omega_{e,0} = \Omega_{v,0} - \Omega_0 \quad \text{(3.91a)}$$
$$\Omega_{e,0} = \bar{K}_R\bar{\Theta}_{e,0} \quad \text{(3.91b)}$$

The initial angular velocity (Ω_0) is assumed to be null, then Equation (3.91) can be substituted in (3.90):

$$\begin{cases} s\bar{\Theta}_e - \bar{\Theta}_{e,0} = -\bar{K}_R\bar{\Theta}_{e,pt} + \Omega_{e,pt} & \text{(3.92a)} \\ J(s\Omega_e - \bar{K}_R\bar{\Theta}_{e,0}) = -K_I X_c - K_\omega\ \Omega_{e,pt} & \text{(3.92b)} \end{cases}$$

If $\Theta_{e,0}$ is assumed to be an external constant signal N, equations (3.92) become:

$$\begin{cases} s\bar{\Theta}_e - N = -\bar{K}_R\bar{\Theta}_{e,pt} + \Omega_{e,pt} & \text{(3.93a)} \\ J(s\Omega_e - \bar{K}_R\, N) = -K_I X_c - K_\omega\ \Omega_{e,pt} & \text{(3.93b)} \end{cases}$$

Coming back to the time domain, making the inverse Laplace transform,

$$\begin{cases} \dot{\bar{\theta}}_e - n\,\delta(t) = -\bar{K}_R\bar{\theta}_e + \omega_e & \text{(3.94a)} \\ J(\dot{\omega}_e - \bar{K}_R n\,\delta(t)) = -K_I x_c - K_\omega\ \omega_e & \text{(3.94b)} \end{cases}$$

$\delta(t)$ is known as the dirac delta and it is defined such that:

$$\delta(t) := \begin{cases} +\infty, & t = 0 \\ 0, & \forall t \in \mathbb{R}, \neq 0 \end{cases} \quad \text{(3.95)}$$

By resuming the definitions 3.71a and 3.72a, it is possible to compute the time derivative:

$$\begin{cases} \dot{\bar{\theta}}_{e,pt} = \dot{\bar{\theta}}_e - \dot{d} & \text{(3.96a)} \\ J\dot{\omega}_{e,pt} = J(\dot{\omega}_e - \bar{K}_R\dot{d}) & \text{(3.96b)} \end{cases}$$

By comparing the right-hand side of the equations 3.96 with the left-hand side of equations 3.94, it is possible to establish that the external disturbance d is linked with n,

$$\dot{d} = n\,\delta(t) \tag{3.97}$$

Referring to the paper [8], it is possible to determine that:

$$d(t) := \begin{cases} n, & t \geq 0 \\ 0, & t < 0 \end{cases} \tag{3.98}$$

So, the amplitude n of an external disturbance d is equivalent to the non-null initial condition of the angle error.

□

Based on these results, the standard tuning formulation in equation (3.78) will lead to the same optimal gains for both architectures.

3.2.2. UAV Quadrotor formulation

After discussing the general approach for a rigid body, in this section the approach for the quadrotor UAV is presented; first of all, let's assume that for the H_∞ approach, the desired angular velocity (ω_d) is assumed to be null (such as in Subsection 3.2.1), then it is possible to write the linearized control law for the Equation (3.18), as:

$$\begin{cases} \omega_v = \bar{K}_R\,\bar{\theta}_e & \text{(3.99a)} \\ T_c(s) = \left(K_p + \dfrac{K_i}{s}\right)(\Omega_v(s) - \Omega(s)) + \dfrac{K_d s}{T_f s + 1}(-\Omega(s)) & \text{(3.99b)} \end{cases}$$

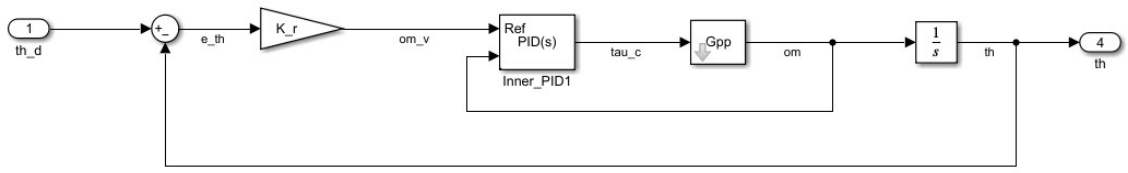

Figure 3.3: Linearized control law for the UAV quadrotor

Looking at the figure 3.3, it is possible to note that the control architecture has a cascade structure; relying on this property, the H_∞ synthesis approach can be subdivided into two steps:

1. **Tuning of the inner loop**: Let's consider the inner loop, made by Equations (3.18a) and (2.36)

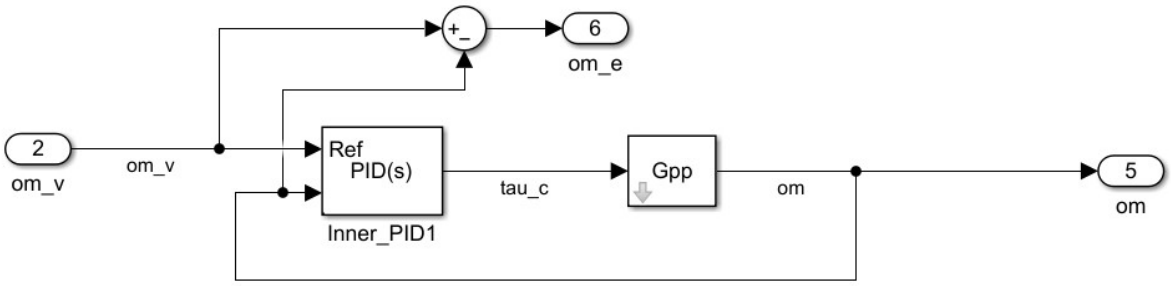

Figure 3.4: Linearized inner loop for the UAV quadrotor

Let's consider a weighting function $W_{S,inner}$ for the sensitivity function, defined equal as in Subsection 3.2.1, but this time, relating $\omega_v \to \omega_e$ and consider the following vector of gain parameters:

$$\rho_{inner} = [K_p, K_i, K_d, T_f]^T. \tag{3.100}$$

It is possible to define the inner loop performance cost $J_{S,inner}$:

$$J_{S,inner}(s, \rho_{inner}) = \|W_{S,inner}(s) S_{inner}(\rho_{inner}, s)\|_\infty \tag{3.101}$$

where S_{inner} is the sensitivity transfer function with input ω_v and ω_e as output. Under these definitions, it is possible to state the optimization problem for the inner loop control as follows:

$$\text{Find} \quad \rho^*_{inner} \quad \text{such that} \quad J_{S,inner} \leq 1 \tag{3.102}$$

In Section 6.1, it is possible to see that the control sensitivity is attenuated even if there is no control effort limitation.

Once the procedure is done, it is possible to find the optimal inner loop gain (ρ^*_{inner}).

2. **Tuning of the outer loop**: When the tuning for the inner loop is done, it is possible to focus only on the outer loop, thanks to the following statement.

Proposition 3.3. *Considering a cascade structure like in Figure 3.4, if the inner loop is faster than the outer loop (empirically, at least 10 times faster), the outer loop tuning can be considered independent from the inner loop and the Figure 3.4 becomes:*

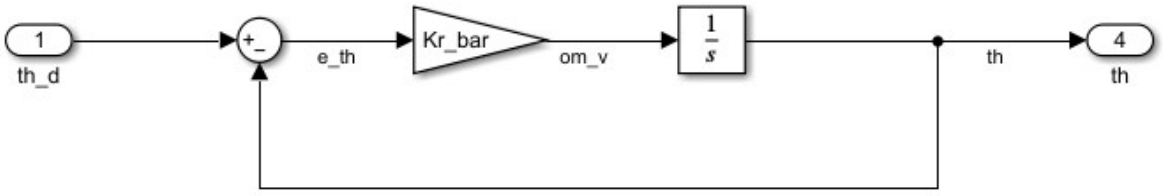

Figure 3.5: Linearized outer loop for the UAV quadrotor

The idea behind Proposition 3.3 is that when the dynamic of the outer loop starts, the dynamic of the inner loop should be at steady-state, in order to establish that:

$$\omega_v \approx \omega \tag{3.103}$$

By making this assumption, the tuning can be easily made because it consists of selecting an appropriate value for $\bar{K}_r$, the sensitivity function here is:

$$S_{outer}(s) = \frac{e_\theta}{\theta_d} = \frac{1}{1 + \frac{\bar{K}_r}{s}} = \frac{s}{s + \bar{K}_r} \tag{3.104a}$$

So, as it is possible to see, the choice of $\bar{K}_r$ leads to the bandwidth rejection of the error on the angle; the higher it is, the faster it is, but pay attention that it must

not be too higher, because if the rejection bandwidth becomes too similar with the inner loop bandwidth, the Preposition 3.1 is no more valid.

Once the gains are found, the gains are entered into the simulator, then, the simulator will be tested and validated first on the linearized control system (under also disturbances and with the use of the linearized command filter) and, then, in the non-linear UAV simulator (with also the use of the non-linear command filter).

Note that the gain $\bar{K}_R$ must be processed, because the actual value that is required to give is the inverse operation of Matrix 1:

$$\begin{bmatrix} K_{R1} \\ K_{R2} \\ K_{R3} \end{bmatrix} = \begin{bmatrix} -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} \bar{K}_{R1} \\ \bar{K}_{R2} \\ \bar{K}_{R3} \end{bmatrix}$$

4 | Hierarchical attitude control with geodesic feedback

This section presents a different approach than the one shown in Chapter 2 to solve the attitude control problem. Specifically, the geodesic approach, originally presented in [14], is adapted to the hierarchical architecture developed in this work and exploited within outer loop. The approach is termed "geodesic" because it steers the reduced attitude along a geodesic path on the 2-sphere while the full attitude is stabilized on SO(3). The steps to implement the geodesic approach in the hierarchical attitude controller used in this work are reported. The considered design is appealing for UAV attitude control as it allows prioritizing thrust-axis control over yaw direction control, thereby avoiding the risk of propellers saturation due to the poor yaw-torque generation mechanism in quadrotor UAV.

4.1. Definition of the problem

4.1.1. Full and Reduced attitude control

The kinematics of a rigid body has already been discussed in Subsection 2.2.1. Here we report the kinematics expressed in an inertial fixed-frame, because the geodesic feedback law (see [14]) is defined in that frame.

Let ω_i and $\omega_{d,i}$ be the angular velocity and the desired angular velocity of the rigid body defined in an inertial fixed frame. The attitude and the desired attitude kinematics can be formulated as follows:

$$\begin{cases} \dot{R} = S(\omega_i)R & \text{(4.1a)} \\ \dot{R}_d = S(\omega_{d,i})R_d & \text{(4.1b)} \end{cases}$$

Notice that Equations (4.1) are different than (2.3); in particular, the right-hand terms are switched.

The rotation matrix error is defined in (2.1), and its dynamics is:

$$\dot{R}_e = \dot{R}_d^T R + R_d^T \dot{R} \tag{4.2a}$$
$$\dot{R}_e = (S(\omega_{d,i})R_d)^T R + R_d^T (S(\omega_i)R) \tag{4.2b}$$
$$\dot{R}_e = R_d^T S(\omega_{d,i})^T R + R_d^T S(\omega_i)R \tag{4.2c}$$
$$\dot{R}_e = -R_d^T S(\omega_{d,i})R + R_d^T S(\omega_i)R \tag{4.2d}$$
$$\dot{R}_e = R_d^T S(\omega_i - \omega_{d,i})R \tag{4.2e}$$
$$\dot{R}_e = (R_d^T S(\omega_i - \omega_{d,i})R_d)R_e \tag{4.2f}$$
$$\dot{R}_e = UR_e \tag{4.2g}$$

Assuming $U \in so(3)$ as a virtual input, the attitude control objective formulated at the kinematic level is to stabilize R_e at I_3. In order to introduce the geodesic feedback approach, we first have to introduce the reduced attitude control problem where the objective is to stabilize a desired pointing direction rather than the full 3D attitude.

Let $e_1 \in \mathbb{S}^2$ be a vector expressed in the body-fixed frame on a 3-dimensional rigid body. The reduced attitude vector $r \in \mathbb{S}^2$ is defined as the inertial frame coordinates of e_1, written as follows:

$$r = Re_1. \tag{4.3}$$

The dynamics of r along 4.2a can be computed as:

$$\dot{r} = \dot{R}e_1 = S(\omega_i)Re_1 = S(\omega_i)r \tag{4.4}$$

These definitions (4.3 and 4.4) can also be extended to the desired reduced attitude ($r_d \in \mathbb{S}^2$); it is possible to define the reduced attitude error as:

$$r_e = R_d^T r = R_d^T Re_1 = R_e e_1 \tag{4.5}$$

Moreover, the dynamics of r_e reads:

$$\dot{r}_e = \dot{R}_e e_1 = UR_e e_1 = Ur_e, \tag{4.6}$$

where U is defined as in 4.2g, with the goal is to control r_e, to stabilize it at e_1, so that r will be aligned with r_d.

4.2. Geodesic feedback law formulation

4.2.1. Rigid body kinematics

The first thing that can be noticed is that $\dot{r}_e \perp r_e$, as can be easily shown:

$$\dot{r}_e^T r_e = (U r_e)^T r_e \tag{4.7a}$$
$$\dot{r}_e^T r_e = r_e^T U^T r_e \tag{4.7b}$$
$$\dot{r}_e^T r_e = -r_e^T U r_e \tag{4.7c}$$

The last Equation (4.7c) is a emisymmetric (or alternant) bilinear form $\phi(v, w) : \mathbb{S}^2 \times \mathbb{S}^2 \to \mathbb{R}$ and, its main property is:

$$\forall v \in V : \phi(v, v) = 0. \tag{4.8}$$

Therefore, Equation 4.7c is null and quantities such as $U \in so(3)$ has more than enough degrees of freedom to fully actuate r_e. In order to cast the control design in simpler form, one can express U in function of $u \in \mathbb{R}^3$ (the only non-null components of U), as follows:

$$U = u r_e^T - (u r_e^T)^T \tag{4.9}$$

Then, by using the identity $uv^T w = v^T w u$ for $u, v, w \in \mathbb{R}^3$, it is possible to rewrite Equation 4.6, such that:

$$\dot{r}_e = U r_e \tag{4.10a}$$
$$\dot{r}_e = (u r_e^T - (u r_e^T)^T) r_e \tag{4.10b}$$
$$\dot{r}_e = u \cancel{r_e^T r_e} - (r_e u^T) r_e \tag{4.10c}$$
$$\dot{r}_e = u - r_e r_e^T u \tag{4.10d}$$
$$\dot{r}_e = (I_3 - r_e r_e^T) u. \tag{4.10e}$$

Since u is arbitrary, r_e can be actuated in any direction along its tangent plane, defined as:

$$T_{r_e} S^2 := \{ v \in \mathbb{R}^3 \mid r_e^T v = 0 \, r_e \in \mathbb{S}^2 \}. \tag{4.11}$$

The reduced attitude stabilization problem aims to design a control input u to stabilize e_1. In case of a constant $v \in \mathbb{S}^3$, if $u = v$, then the dynamics defined in (4.10e) moves r_e in the steepest descent direction of the geodesic distance $\theta(v, r_e) = arccos(v^T r_e)$,

$$\arg\min_{u\in S^3}\dot{\theta} = \arg\max_{u\in S^3}\frac{d}{dt}(v^T r_e) = \arg\max_{u\in S^3}(v^T(I_3 - r_e r_e^T)u) = v \tag{4.12}$$

When the system is controlled along a path of minimum length in the state space (a great circle in this case), u said to be geodesic.

When moving to the full attitude stabilization, U is designed such that I_3 is an almost globally asymptotically stable equilibrium (defined in A.2) of the full attitude R_e, while the reduced attitude r_e moves toward e_1 along a great circle.

Proposition 4.1. *Let the feedback law $U : SO(3) \to so(3)$ be defined as:*

$$U = PR_e^T - R_eP + k\left[R_eQ(R_e^T - R_e)QR_e^T\right] \tag{4.13}$$

where $P \in \{A \in \mathbb{R}^{3\times3} \mid A^2 = A, A^T = A\}$, where P is a constant orthogonal projection, $k \in (0,\infty)$ and $Q = I_3 - P$.

The first skew-symmetric difference in equation (4.13) is designed to steer R_eP to P, while, when $\|R_eP - P\|_2$ is small, the second difference kicks in to steer R_eQ to Q; in simple terms, there is a stabilization on $\mathbb{S}^2$ followed by stabilization on SO(3), the two terms are fused into one smooth function. A slight modification of Equation (4.13) is introduced in Subsection 4.2.2. k presents a trade-off between the reduced and full attitude convergence rates. Q is also an orthogonal projection matrix and, moreover, P and Q are such that:

$$\begin{cases} P + Q = I_3 & \text{(4.14a)} \\ PQ = \mathbf{0}. & \text{(4.14b)} \end{cases}$$

Consider the closed-loop dynamics 4.2 with U selected as in Equation 4.13:

$$\dot{R}_e = UR = P - R_ePR_e + k\left[R_eQ(R_e^T - R_e)Q\right]. \tag{4.15}$$

Selecting $P = e_1e_1^T$, and multiplying the Equation 4.16 by e_1 , it is possible to write:

$$\begin{aligned}
\dot{R}_e e_1 &= (P - R_ePR_e + k\left[R_eQ(R_e^T - R_e)Q\right])e_1 && \text{(4.16a)}\\
\dot{r}_e &= (e_1e_1^T - R_e(e_1e_1^T)R_e + k\left[R_eQ(R_e^T - R_e)Q\right])e_1 && \text{(4.16b)}\\
\dot{r}_e &= e_1\cancel{e_1^Te_1} - R_e(e_1e_1^T)R_ee_1 + k\left[R_eQ(R_e^T - R_e)Q\right]Pe_1 && \text{(4.16c)}\\
\dot{r}_e &= e_1 - r_ee_1^Tr_e + \cancel{k\left[R_eQ(R_e^T - R_e)QP\right]e_1} && \text{(4.16d)}\\
\dot{r}_e &= e_1 - r_ee_1^Tr_e = e_1 - (e_1^Tr_e)r_e && \text{(4.16e)}
\end{aligned}$$

It is possible to note that Equation 4.16e is equal to Equation 4.10e with $u = e_1$; so, this feedback law results in $r_e \in \mathbb{S}^2$ moving towards e_1 along a great circle. As mentioned in [14], I_3 is an almost globally asymptotically stable equilibrium of the closed-loop dynamics generated by 4.13. The rate of convergence is locally exponential. The set of initial conditions from which convergence to the identity matrix fails is meager in $SO(3)$.

4.2.2. Geodesic law for hierarchical control

As the geodesic controller in equation 4.13 is defined in the inertial frame, the expression must be converted in body-frame components before implementing it in the hierarchical architecture proposed in equations 2.22 and 2.23.

Starting from

Proposition 4.2. *The geodesic feedback law 4.13 has the following expression in the body-fixed frame:*

$$S(\gamma_{R,geo}) = R_e^T U R_e \tag{4.17a}$$
$$S(\gamma_{R,geo}) = R_e^T (P R_e^T - R_e P + k \left[R_e Q (R_e^T - R_e) Q R_e^T\right]) R_e \tag{4.17b}$$
$$S(\gamma_{R,geo}) = R_e^T P \cancel{R_e^T R_e} - \cancel{R_e^T R_e} P R_e + k \left[\cancel{R_e^T R_e} Q (R_e^T - R_e) Q \cancel{R_e^T R_e}\right]) \tag{4.17c}$$
$$S(\gamma_{R,geo}) = R_e^T P - P R_e + k \left[Q (R_e^T - R_e) Q\right]. \tag{4.17d}$$

The Proposition 4.2 can be demonstrate starting from Equation 4.2g, let us assume that ω_i is controllable, then assign,

$$\omega_i = \omega_{d,i} + R_d\, u. \tag{4.18}$$

The equation 4.18 can be written in body-fixed frame:

$$\omega_b = R^T \omega_i \tag{4.19a}$$
$$\omega_b = R^T \omega_{d,i} + R^T R_d\, u \tag{4.19b}$$
$$\omega_b = \omega_{d,b} + R_e^T\, u. \tag{4.19c}$$

This Equation 4.19 implies Equation 4.17.

Equations 4.17 are a general formulation for the geodesic term in the body-fixed frame; in our work, this term has been used for the rigid body and, then, for the UAV Quadrotor

is introduced this architecture in order to overcome the saturation of the actuators, the control used is the following:

$$S(\gamma_{R,geo}) = k_2(R_e^T P - P R_e) + k(\Psi)\left[Q(R_e^T - R_e)Q\right], \tag{4.20}$$

$k_2 \in (0, \infty)$ is a gain that weighs the effect of the reduced attitude control, while $k(\Psi)$ is a parameter variant gain that accounts for the fact that for great angle θ (which is the angle between r_e and the axis e_1) the gain must be small and vice versa. This effect makes the reduced and whole attitude act separately in the angle function. The gain is defined as follows:

$$k(\Psi) = k_{max}(1-\Psi)^n + k_{min}\Psi^n \tag{4.21}$$

k_{min} and k_{max} are defined, respectively, as the minimum and the maximum value that $k(\Psi)$ can assume; this choice has been made in order to make the second difference less important when the angle are low and vice-versa, prioritizing the reduced attitude stabilization, when θ is low; moreover, Ψ is in the function of the cosine of the angle θ and not of the θ, because θ has a not well-defined derivative, for some configurations, which will be important when it is considered the term ω_v. Ψ is defined as,

$$\Psi = \frac{1 - cos(\theta)}{2} \tag{4.22a}$$

$$cos(\theta) = r_e^T e_1 \tag{4.22b}$$

Note that to implement the control law within the hierarchical design presented in Chapter 2, the term $\dot{\gamma}_{R,geo}$ in equation 2.21b must also be evaluated. To this end, the time derivative of $\gamma_{R,geo}$ can be computed as follows:

$$S(\dot{\gamma}_{R,geo}) = k_2(\dot{R}_e^T P - P\dot{R}_e) + \left[k(\Psi)Q(R_e^T - R_e)Q + \dot{k}(\Psi)Q(\dot{R}_e^T - \dot{R}_e)Q\right] \tag{4.23}$$

where,

$$\dot{k}(\Psi) = \left[-n(1-\Psi)^{n-1}k_{max} + n\Psi^{n-1}k_{min}\right]\dot{\Psi} \tag{4.24a}$$

$$\dot{\Psi} = \frac{sin(\theta)}{2} = \dot{r}_e^T e_1 \tag{4.24b}$$

with $\dot{r}_e$ defined as in Equation (4.10).

5 | Numerical results

This chapter presents numerical results corresponding to the implementation of the proposed hierachical architecture for attitude control of an ideal rigid-body. The first section presents all the simulation scenario and data, while the second section deals with the implementation of the hierarchical control the law discussed in Chapters 2 and 4.

5.1. Simulation example data

Let us consider a generic rigid body, fully actuated, whose dynamics is given in equations (2.3), where the inertia matrix is fixed to $J = diag([1, 2, 3])$ and the control disturbance τ_e is defined as a time-varying perturbation with the following profile:

$$\tau_e = \begin{cases} [1, 1, 1]^T, & t < 15 \\ [3, 3, 3]^T, & t \geq 15 \end{cases} \tag{5.1}$$

The desired attitude motion is characterized by a polynomial spin-up maneuver with a steady-state angular velocity $\omega_{dsteady} = [0.5, 0.5, 0.5]^T$ rad/s, like the one shown in the following figure:

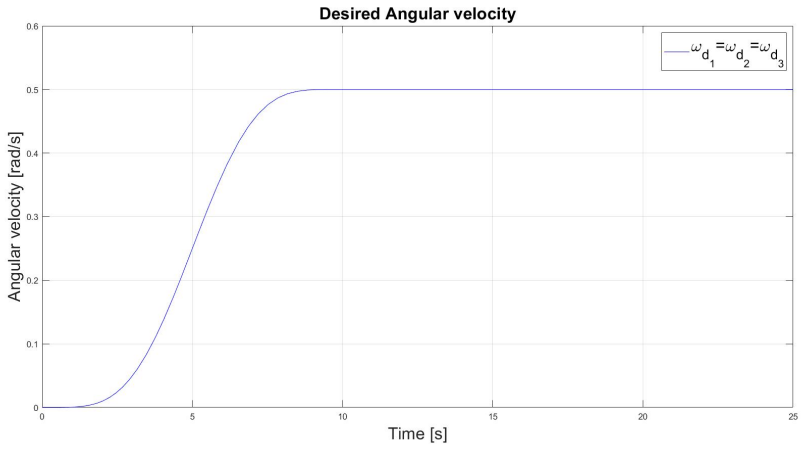

Figure 5.1: Spin up motion: desired angular velocity ω_d

The initial condition for the desired attitude, attitude and body-angular velocity are chosen as follows:

$$\begin{cases} R_d(0) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ R(0) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \\ \omega(0) = [3, 3, 3]^T \end{cases} \tag{5.2}$$

5.2. Hierachical control law

Once the general data are defined, the performance obtained by the controller presented in Chapter 2 will be shown in this section; in particular, a comparison is presented among several variants (proposed, feedback-linearizing, saturated, and PD-like), given, respectively, in Equations (2.32), (2.33), (2.34) and (2.35), will be presented.

The gains for the first 3 types of hiearchical controllers are set as:

	K_R	K_i	K_ω
roll	1	1.11	3.33
pitch	1.001	1.665	1.665
yaw	0.999	3.33	3.33

Table 5.1: Gains used for the first 3 variants

While for the PD-like controller (equation (2.35)), which is not based on a hierarchical architecture, the gains are selected as:

	K_R	K_i	K_ω
roll	25	1.11	10
pitch	12.5	1.665	20
yaw	0	3.33	30

Table 5.2: Gains used for PD-like control

Specifically, the gains of the inner loop stabilizer (K_ω and K_i) are chosen to have the linearized inner loop error dynamics (Equations (3.58)) behaving like first order systems with bandwidth $10\,rad/s$. The outer loop gains are tuned to have a $1\,rad/s$ bandwidth for the linearized outer loop error dynamics ((3.58)).

The values for the PD-like controller are chosen to get a similar transient response with respect to the other three-cascade methods; the constant used for the saturation is set to $M = 5.6Nm$.

Once all the tuning parameters are fixed, the results are shown using tracking error and control effort indexes, in particular:

- **Attitude tracking error**: This index corresponds to a non linear error for the angle displacement, it is a way to describe the attitude error in a scaled parameter. The index is computed using the following formula:

$$\|R_e\|_{SO(3)} = \frac{1}{4}\|R - I_3\|_F \in [0, 1], \tag{5.3}$$

  where $\|A\|_F$ is denoted as the Frobenius norm of a generic matrix A, defined as:

$$\|A\|_F = \sqrt{tr(A^T A)}. \tag{5.4}$$

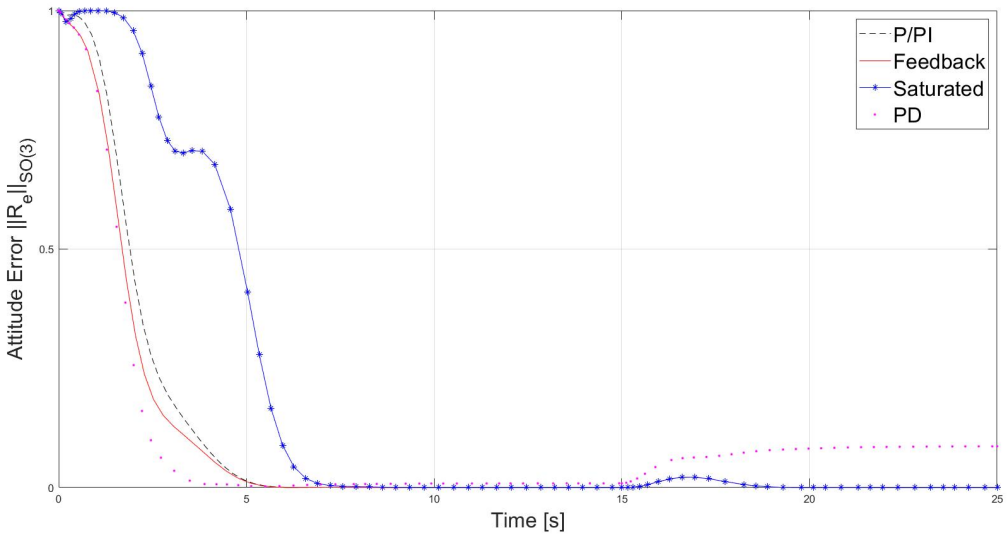

Figure 5.2: Attitude error $\|R_e\|_{SO(3)}$

It is possible to see that the feedback linearizing control (red one) has the fastest

response in terms of attitude error tracking, while the saturated response is the slowest (blue one). All controllers, except for the PD-like one, promptly react to the presence of constant disturbances due to the presence of integral states.

The PD-like control torque is easier and computationally convenient to write (because there is no need to compute ω_v); but, on the other hand, it is not capable to counter the effect of a disturbance at 15 s, the control torque at the beginning is very high, and it is needed to use higher tuning parameters to get the same response of the other architecture.

- **Angular velocity error index**: This element is different from the ω_e defined in Equation (2.21b), indeed, this index is defined as:

$$\|e_\omega\| = \|\omega - R_e^T \omega_d\|, \tag{5.5}$$

$\|\cdot\|$ is the Euclidean norm, where for a generic $x \in \mathbb{R}^n$:

$$\|x\| = \sqrt{x_1^2 + \ldots + x_n^2}. \tag{5.6}$$

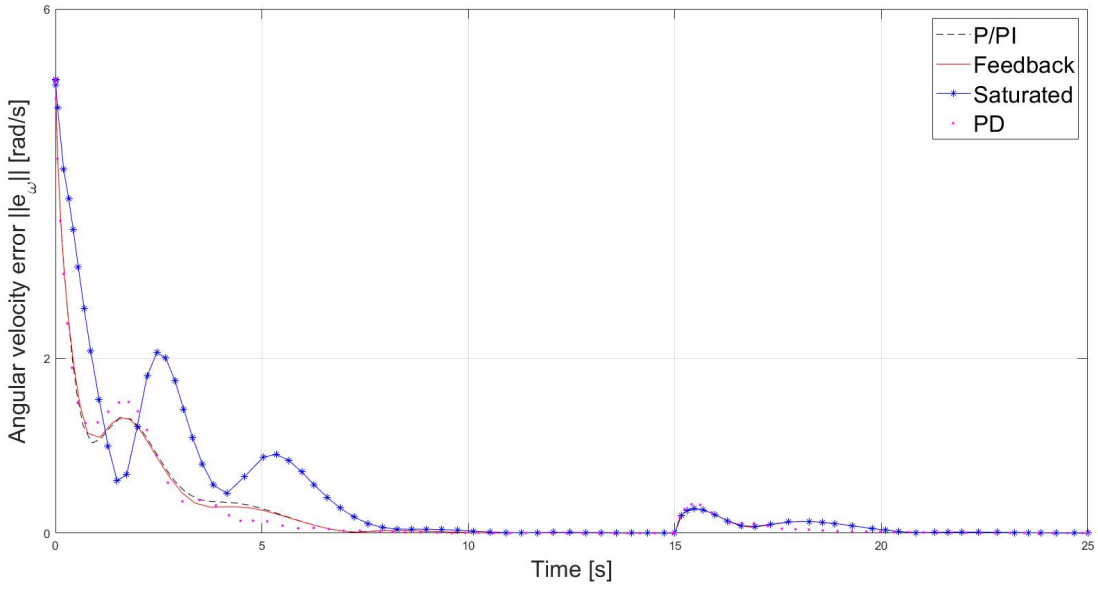

Figure 5.3: Angular velocity error index $\|e_\omega\|$

- **Control infinity norm**: This index chooses, among the three control input, the maximum value, in absolute value; it is defined as:

$$\|\tau_c\|_\infty = max_{i\in 1,\dots n}(|x_i|) \tag{5.7}$$

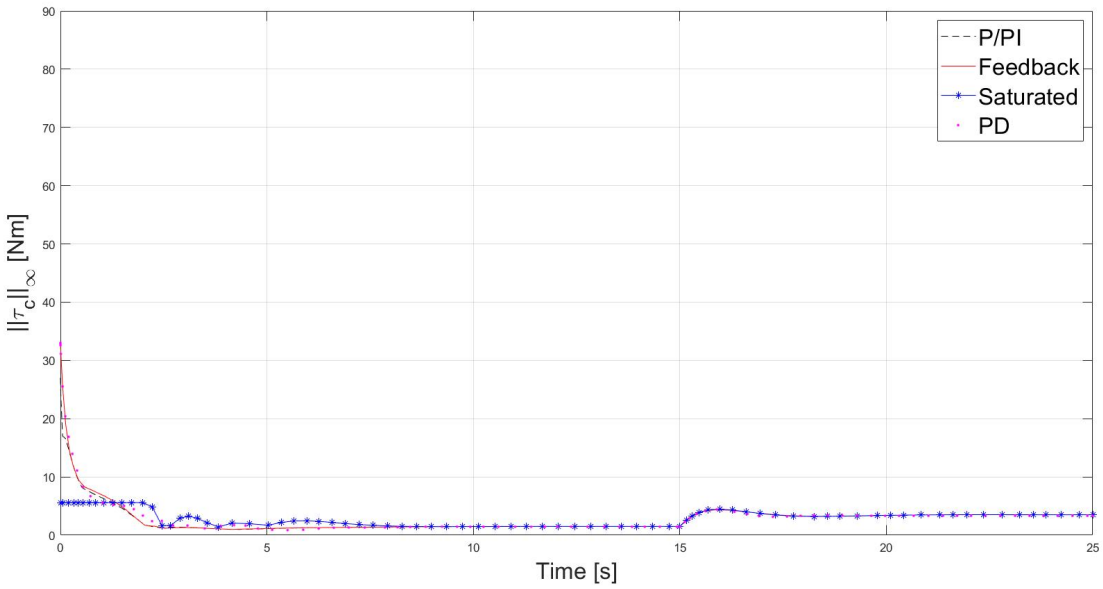

Figure 5.4: Infinity norm of the control torque τ_c

The original controller tries to reach the same performances of the feedback linearizing but with much less control effort. Their effect is very similar when the angle is very small.

The feedback linearizing control reaches the higher peak value at the beginning among all the other architectures.

5.3. Geodesic Feedback law

The last part of this chapter shows how the geodesic feedback control law can be used to stabilize the rigid body while prioritizing thrust-axis control; first of all, let's define the data used for the outer loop (seen in Chapter 4):

$$\begin{cases} k_{max} = 1 \\ k_{min} = 0.01 \\ k_2 = 2 \end{cases} \tag{5.8}$$

The third axis, namely, the $e_3 = [0, 0, 1]^T$, is chosen as the axis for the reduced attitude stabilization, so matrix P in equation (5.9) is chosen as:

$$P = e_3^T e_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.9}$$

The PI inner-loop controller presented in equation (2.32) is used with the same gains reported in Table 5.1 from the previous example.

The choice of the gains has been made via a trial & error approach, in particular, the gain k_2 is greater than k, in order to prioritize the reduced attitude kinematic, moreover, k is variable from a lower value k_{min} near to 0, in order to get rid of the second difference.

The first result that is shown, in Figure 5.5, concerns the behavior of the gain k with respect to as a function of θ, the angle between the desired and current third axis, like in figure 5.5:

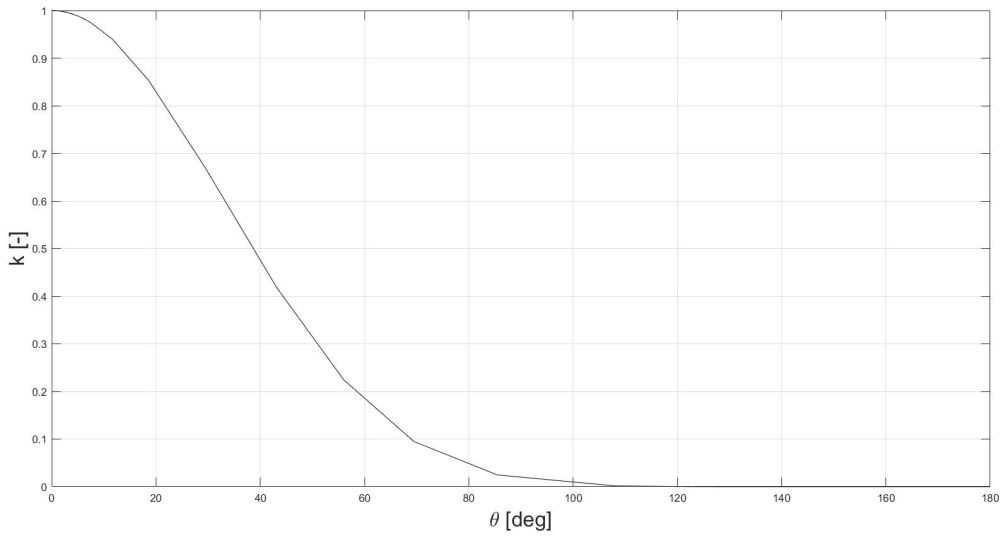

Figure 5.5: Gain k(θ)

As it is possible to see, the gain is near k_{min} when the angle between e_3 and r_e is significant (from around 110 degrees up to 180 degrees).

A comparison between the classic formulation (presented in Equation (4.13), choosing $k = k_{max}$) and the formulation with the variable k is presented.

The index error and the cumulative distances are computed both, for $i = 1, 2, 3$:

$$\begin{cases} e_{\theta,i} = arccos(R_{e,ii}) \\ d_{\theta,i} = \int_0^t \|\dot{R}_e e_i\|_2 \, d\tau \end{cases} \tag{5.10}$$

The plots for the classic formulation are presented, respectively:

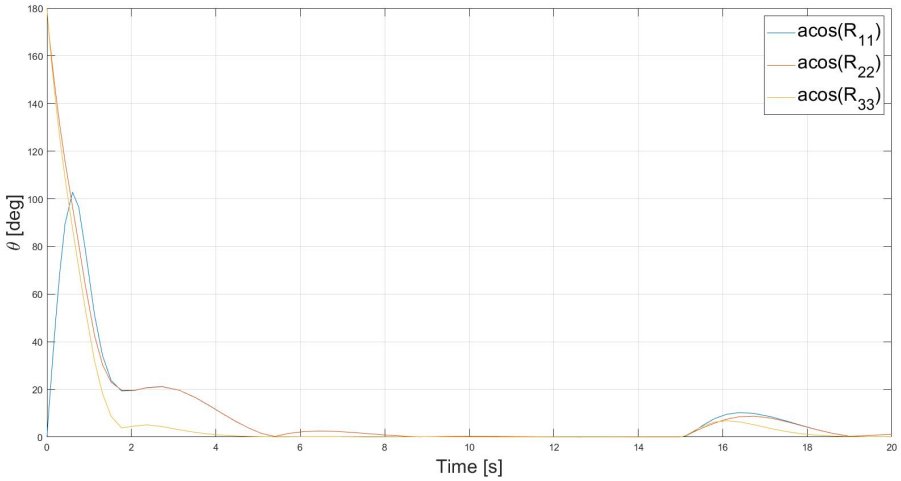

Figure 5.6: Error $arccos(R_{e,ii})$: classic formulation

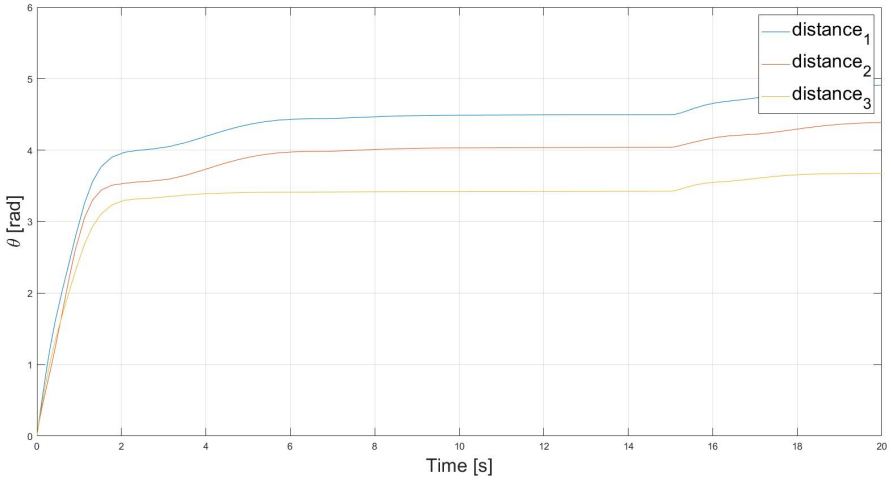

Figure 5.7: Traveled distance: classic formulation

While for the modified formulation:

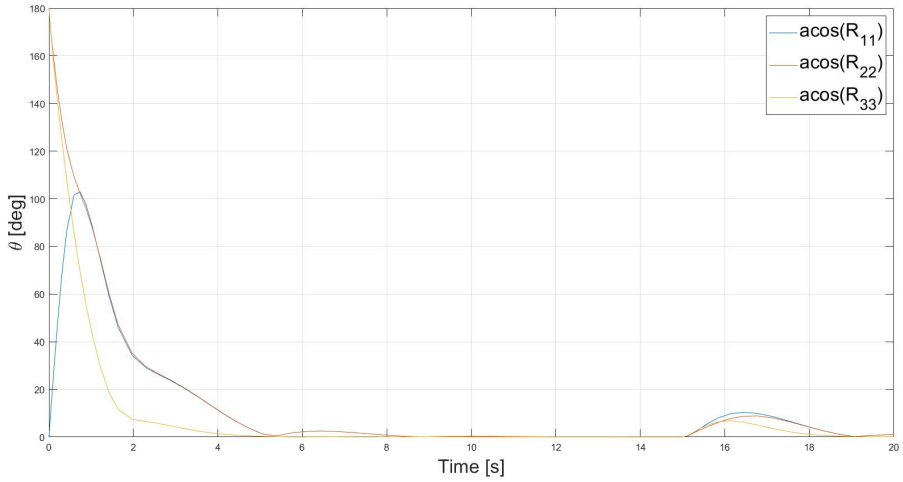

Figure 5.8: Error $arccos(R_{e,ii})$: variable k

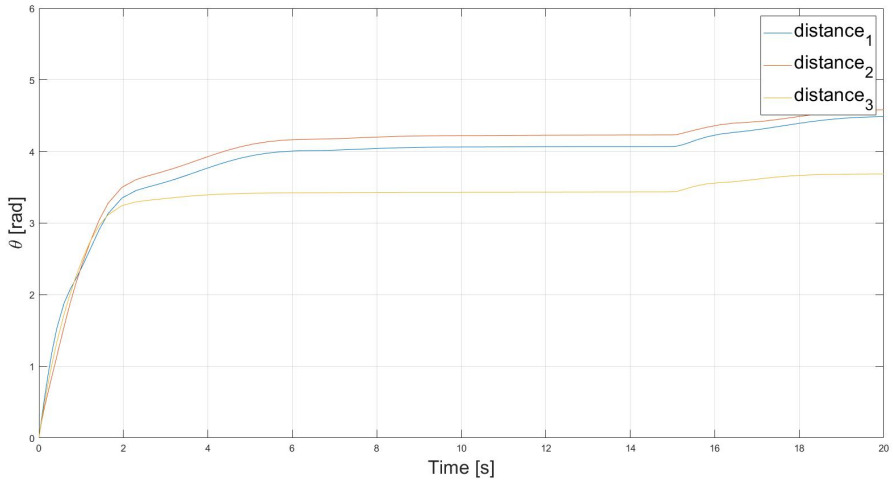

Figure 5.9: Traveled distance: variable k

It is possible to see, for both the formulations, that among all the angle errors, the $arccos(R_{e,33})$ converges to zero faster than the others, thanks to the geodesic effect of the control law; it can also be underlined, in Figure 5.7 and 5.9 (by looking at $d_{\theta,3}$), that it has the minimum cumulative distances.

The $d_{\theta,1}$ is higher in the case of classic formulation and, vice versa for the $d_{\theta,2}$. The control system behaves very efficiently under a control torque disturbance, rejecting it in around 2-3 seconds.

It is advantageous to see a comparison, using the indexes introduced in 5.2:

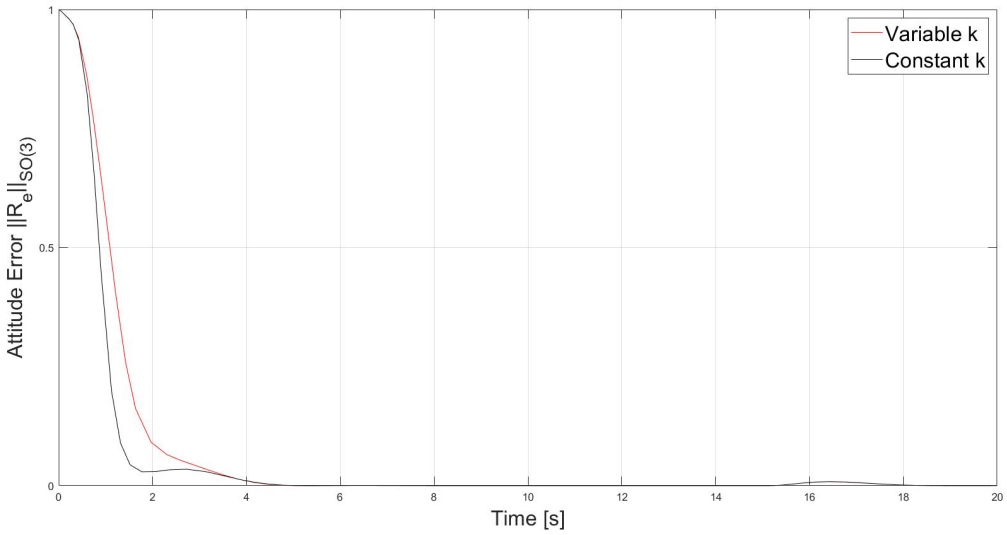

Figure 5.10: Attitude error: Classic vs Variable k

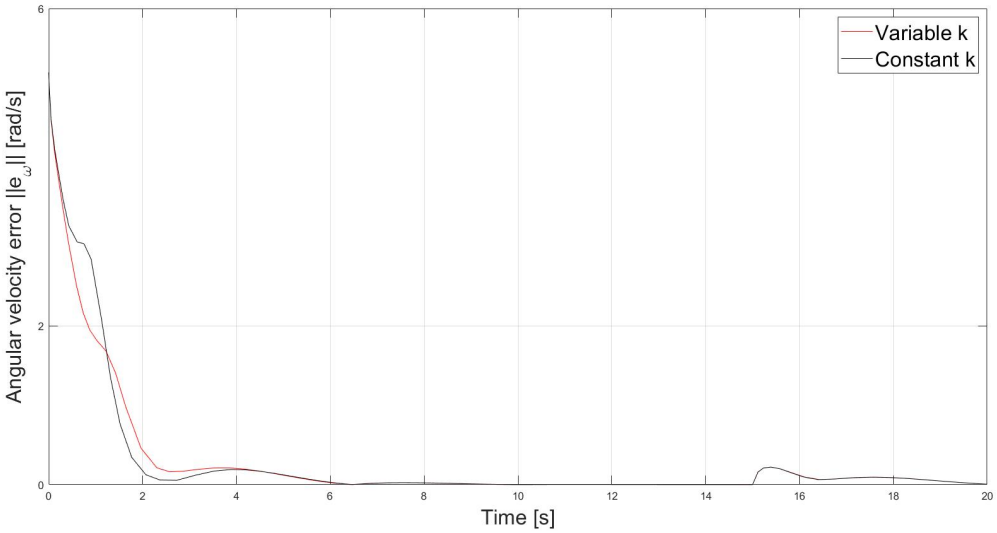

Figure 5.11: Angular velocity error index: Classic vs Variable k

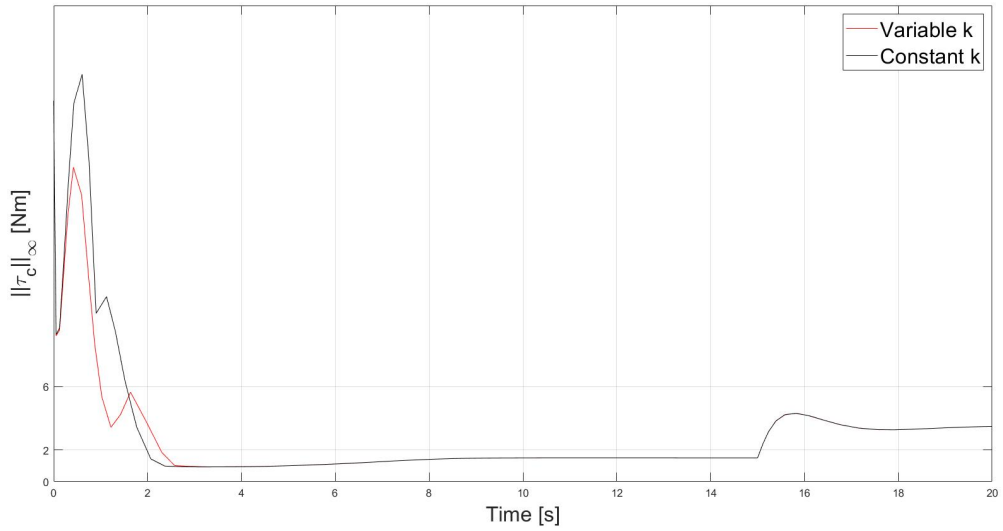

Figure 5.12: Infinity norm of the control torque: Classic vs Variable k

The advantage of the classic formulation over the modified one is that it produces a little bit faster transient response for the attitude error and angular velocity error; but, on the other hand, the major disadvantage comes from the control effort being higher than the classic one.

Finally, a comparison between the geodesic (with variable k) and the hierarchical controller architecture is shown as follows:

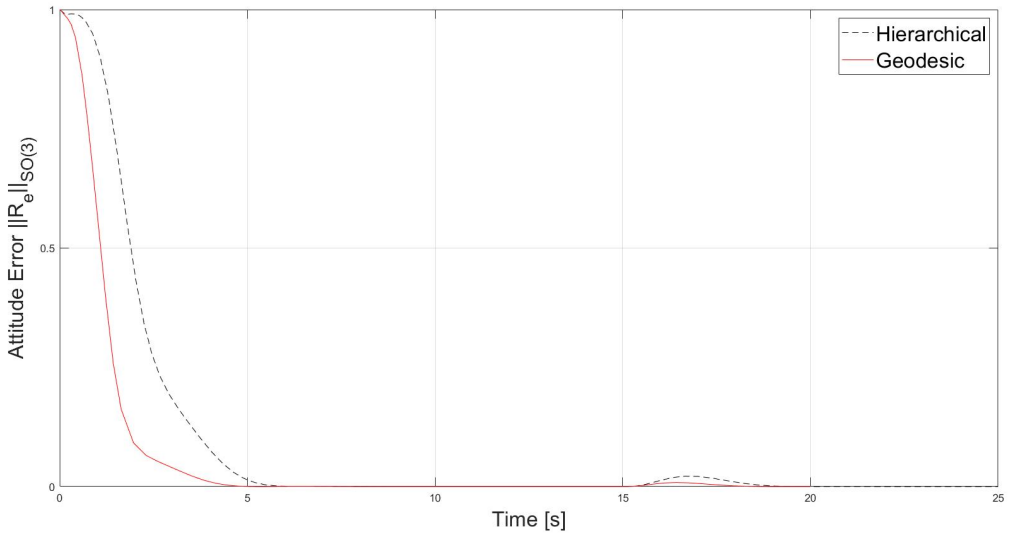

Figure 5.13: Attitude error: Geodesic vs Hierarchical

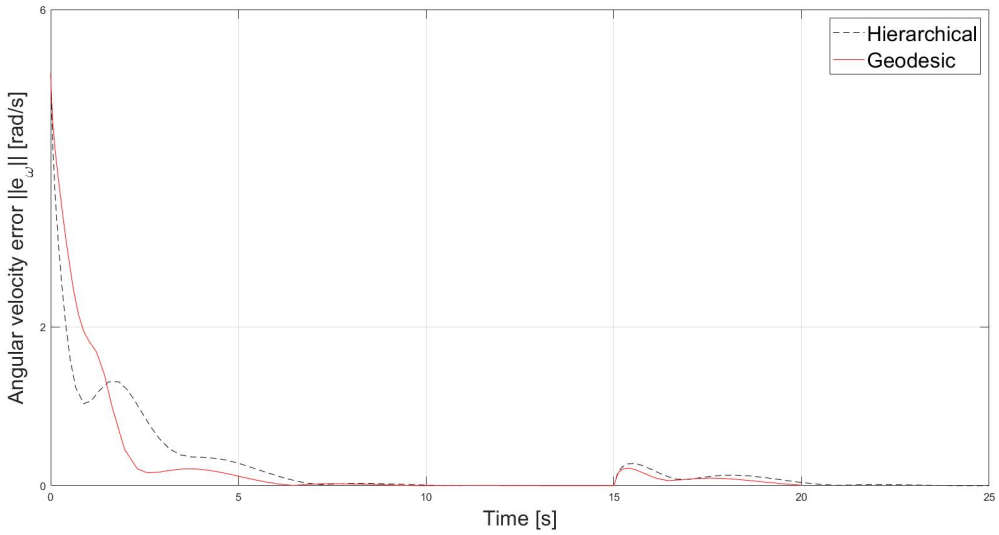

Figure 5.14: Angular velocity error index: Geodesic vs Hierarchical

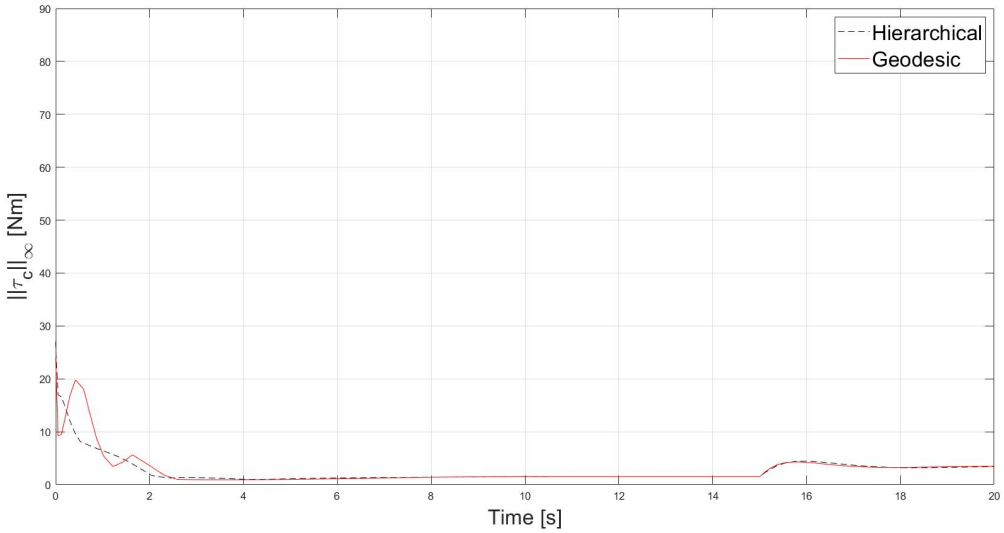

Figure 5.15: Infinity norm of the control torque: Geodesic vs Hierarchical

The geodesic control scheme reacts very faster to attitude tracking than the hierarchical control law, also for the angular velocity error, although the difference is low; however, the geodesic feedback requires more control effort than the hierarchical control law. The geodesic feedback reacts better to external disturbances than the hierarchical architecture.

6 | UAV Quadrotor: Simulations and Experiments

This chapter aims to present the numerical and experimental validation of the two architectures developed in Chapters 2 and 4. First, the tuning of the gains for the UAV is introduced with the method presented in Chapter 3 and this gains are tested preliminarily on a linearized system. Then, using the UAV simulator scheme on Simulink, the controllers are tested and simulated on the Simulink model and view the first advantages. Finally, experiments are carried out for both the controllers, in order to validate the results discovered in the simulations.

6.1. Tuning of the gains

The first step is the tuning of the gains of the control law presented in section 2.3 using the procedure outlined in Subsection 3.2.2; then, the results and plots of the behavior of the linearized control law will be presented for a preliminary performance assessment.

- **Tuning of the inner loop**: The linearized inner loop is depicted in Figure 3.4; the goal is to find the gains of the 2-dof PID to correctly track the response between ω_v and ω_e.

  The choice of the weighting functions is made relying on a pre-existing controller; for each axis, starting from the sensitivity function bandwidth and its peak of resonance, an improvement of the performance is made, keeping into account to not stress too much the actuators. The goal of the tuning is to get better response in terms of peak of the sensitivity transfer function and control effort, by making them as lower as possible; also, if this means to give up some performance in terms of time response. Indeed, the time response of the system is lower than the pre-existing controller, but the velocity of response can be recovered using the feedforward term.

  A weighting function for control moderation is not useful for the tuning, because by choosing carefully the parameters introduced in table 6.1, the control effort (reg-

ulated by the transfer function between ω_v and τ_c) is inside the constraint (by constraint, it is referred to the equation introduced in (3.79)), so it is not taken into account in the formulation. For completeness, in Figure 6.4, 6.5 and 6.6 are shown the control effort moderation, using the gains that will be depicted in table 6.2.

The parameters for the weighting functions according to the input arguments of the makeweight routing in MATLAB are reported in table 6.1:

	DC	[FREQ,MAG]	HF
roll	0.0001	[21,1]	1.7
pitch	0.0001	[21,1]	1.7
yaw	0.0001	[13,1]	1.5

Table 6.1: Weighting functions parameter for each axis

With these weighting functions, using the formulation used in Subsection 3.2.2, it is possible to obtain the following gains :

	K_p	K_i	K_d	T_f
roll	0.046	0.038	10^{-3}	108.76
pitch	0.046	0.024	10^{-3}	104.67
yaw	0.077	0.32	$1.5*10^{-3}$	99.82

Table 6.2: Gain tuned for inner-loop PID

The following figures allow to see the comparison between the weighting function and the tuned sensitivity function for each axis:

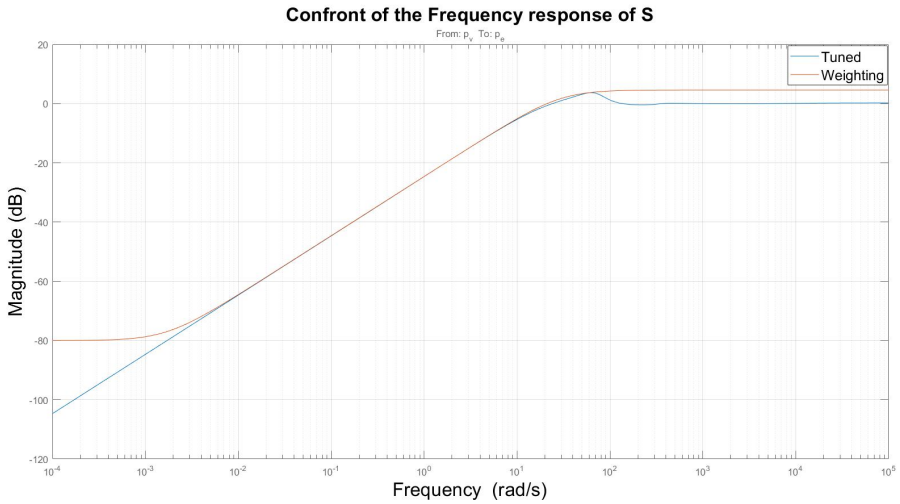

Figure 6.1: Confrontation weighting function vs tuned sensitivity function: roll axis

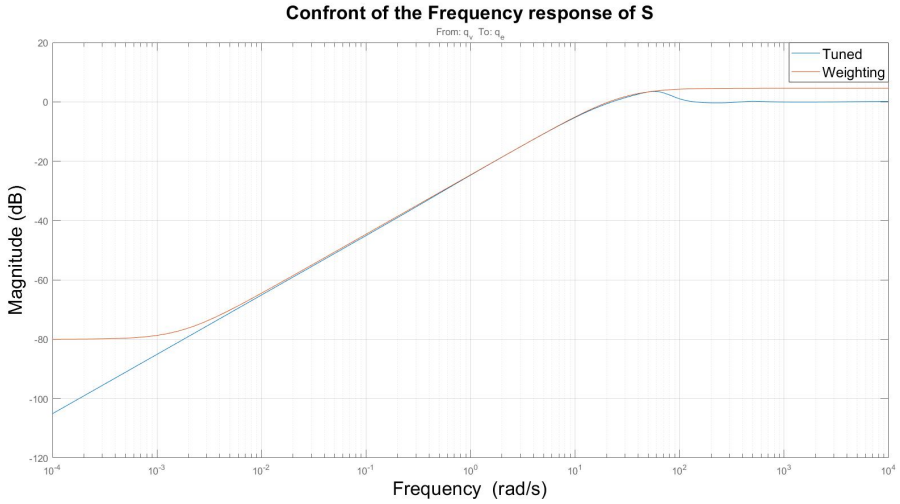

Figure 6.2: Confrontation weighting function vs tuned sensitivity function: pitch axis

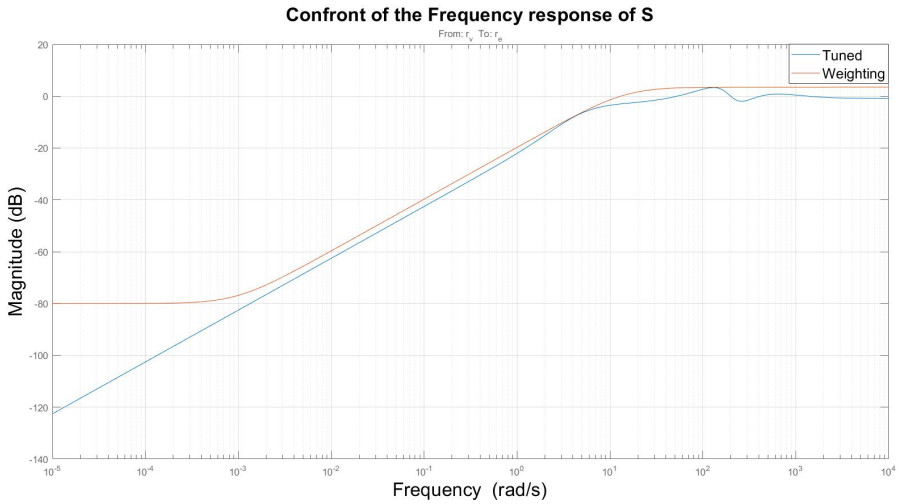

Figure 6.3: Confrontation weighting function vs tuned sensitivity function: yaw axis

In the following plots, the control effort transfer function (from ω_v to τ_c) is shown, and it is possible to see that, even if without control moderation requirements, the H_∞ tuning produces a response that it does not require so much control effort.

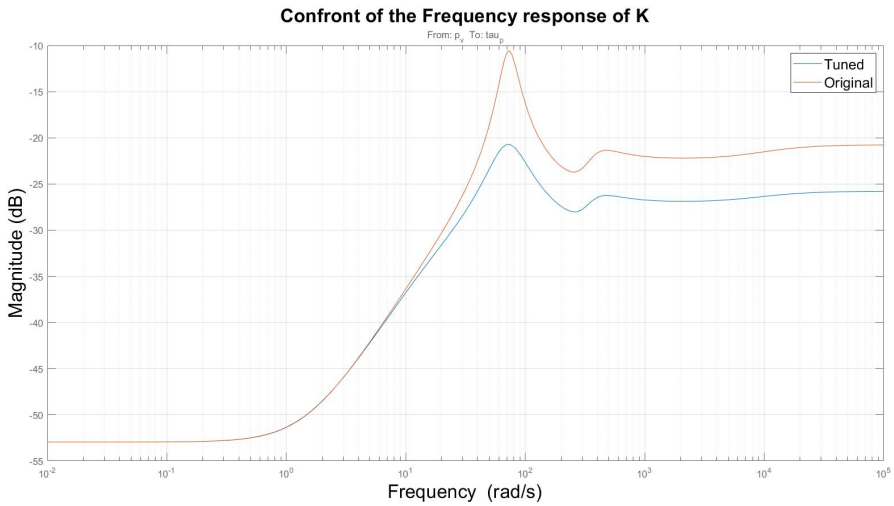

Figure 6.4: Control effort moderation: roll axis

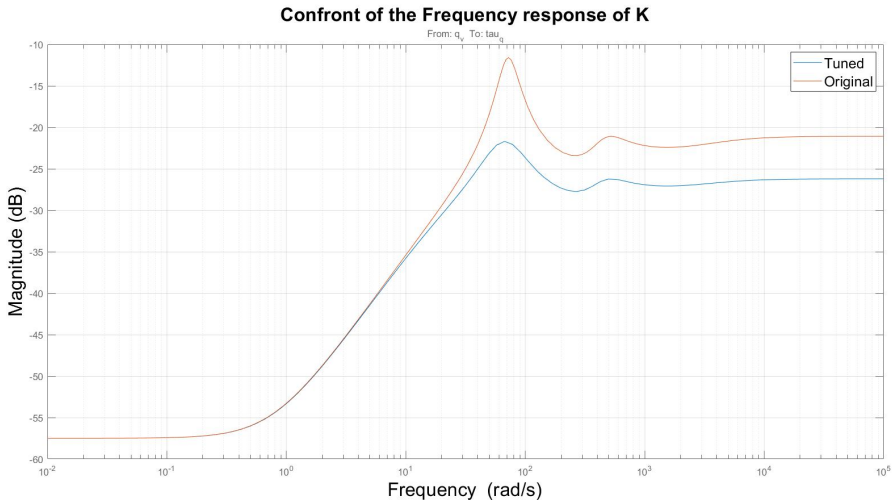

Figure 6.5: Control effort moderation: pitch axis

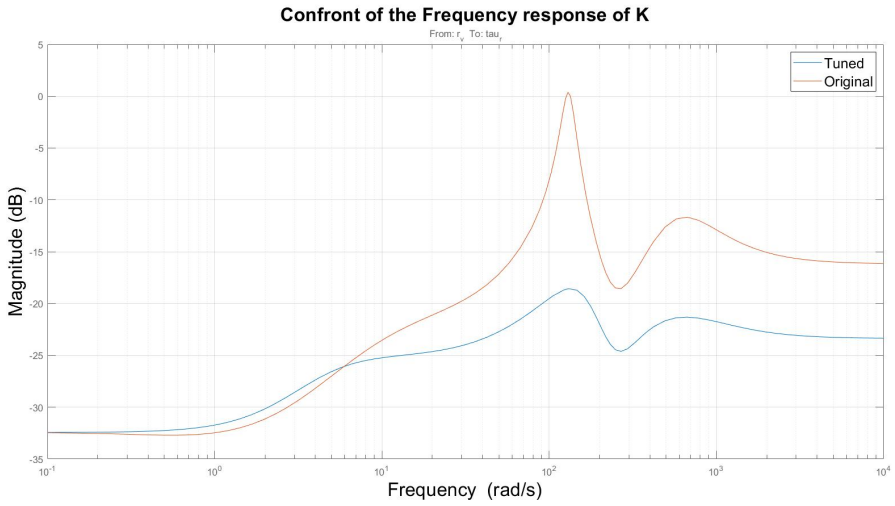

Figure 6.6: Control effort moderation: yaw axis

- **Tuning of the outer loop** The gains of the outer loop control are selected by hand, leveraging the simple scheme seen in Figure (3.5)

	$\bar{K}_r$
roll	10
pitch	10
yaw	3

Table 6.3: Gain for outer-loop controller

Remember that these gains are to be referred for the linearized plant when the non-linear plant gains to be used are the ones outcoming from the Equation 3.2.2

	K_r
roll	3
pitch	3
yaw	17

Table 6.4: Gain for outer-loop controller (Non-linear plant)

6.2. Simulations

6.2.1. Linearized model

Once the gains are defined, the next step is to simulate a response from the entire linearized system; in particular, a sine-wave response is simulated by selecting the following:

$$\begin{cases} \theta_{des} = cos(\omega_d t) & \text{(6.1a)} \\ \omega_{des} = \dfrac{d}{dt}(\theta_{des}) = -\omega_d sin(\omega_d t) & \text{(6.1b)} \end{cases}$$

Where $\omega_d = 10\, rad/s$ is the same for all three axes, the value of the frequency response sine-wave has been chosen to test the rigid body with a stressful maneuver. The Simulink scheme for the simulation is the following:

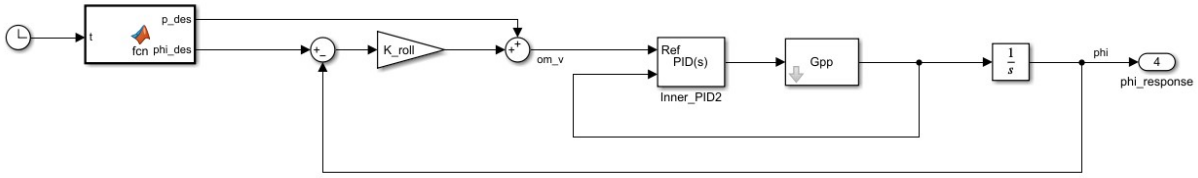

Figure 6.7: Linearized scheme of UAV quadrotor

The main goal of this simulation is to see the sine-wave response of the angle and its comparison with a case where ω_{des} is null (making the feedforward term equal to zero) and the case where ω_{des} is defined as in Equations (6.1), for the three axes.

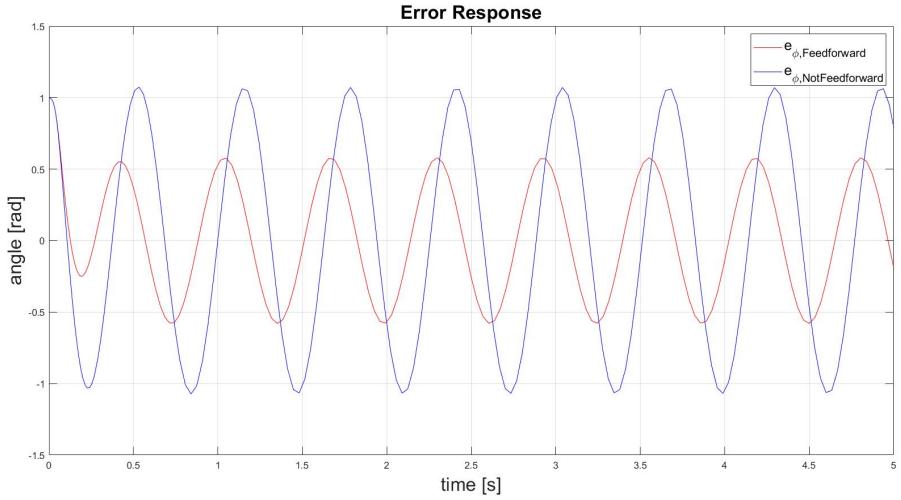

Figure 6.8: Error response: Roll axis

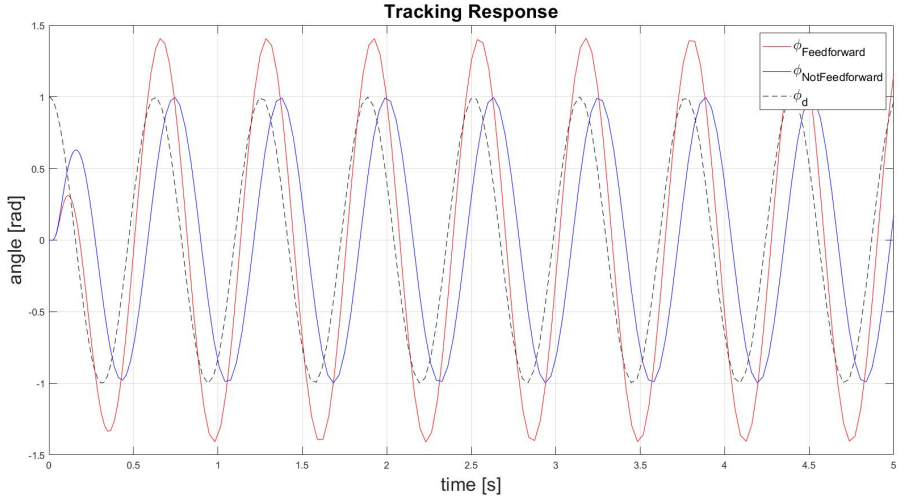

Figure 6.9: Tracking response: Roll axis

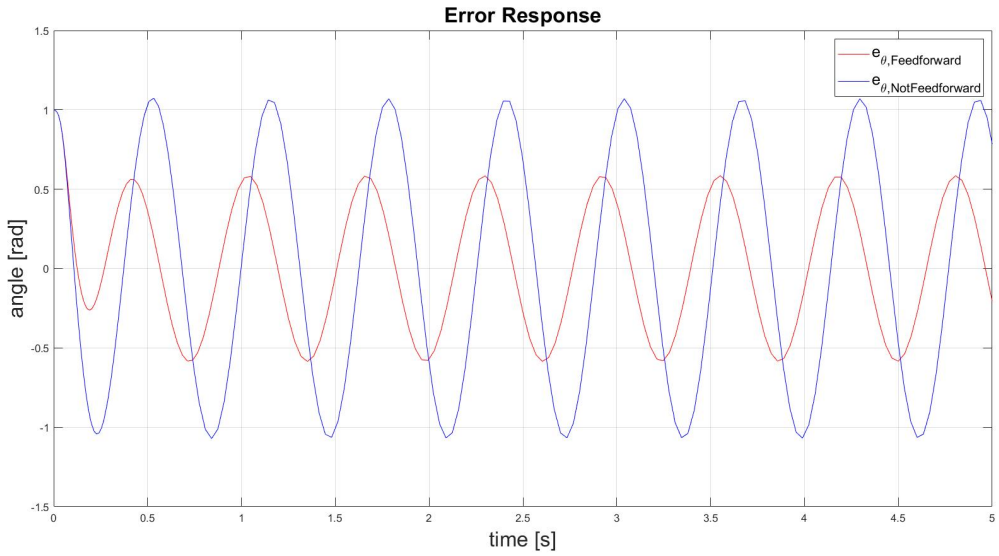

Figure 6.10: Error response: Pitch axis

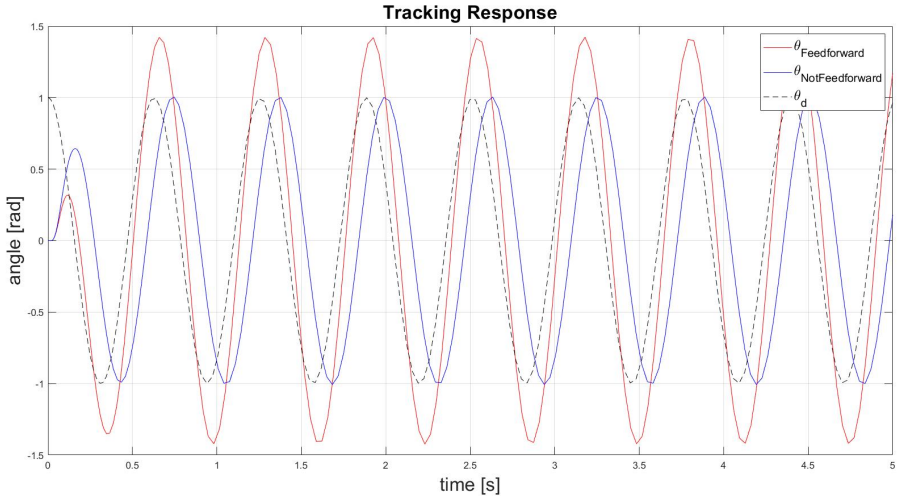

Figure 6.11: Tracking response: Pitch axis

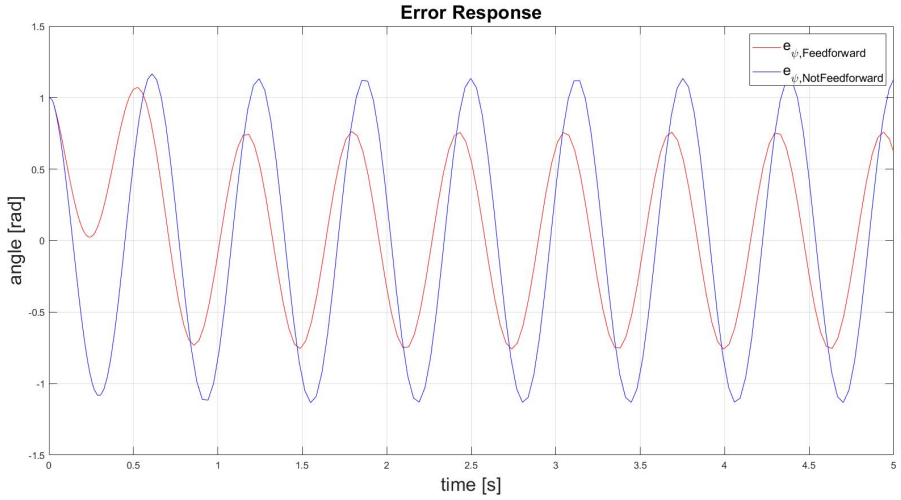

Figure 6.12: Error response: Yaw axis

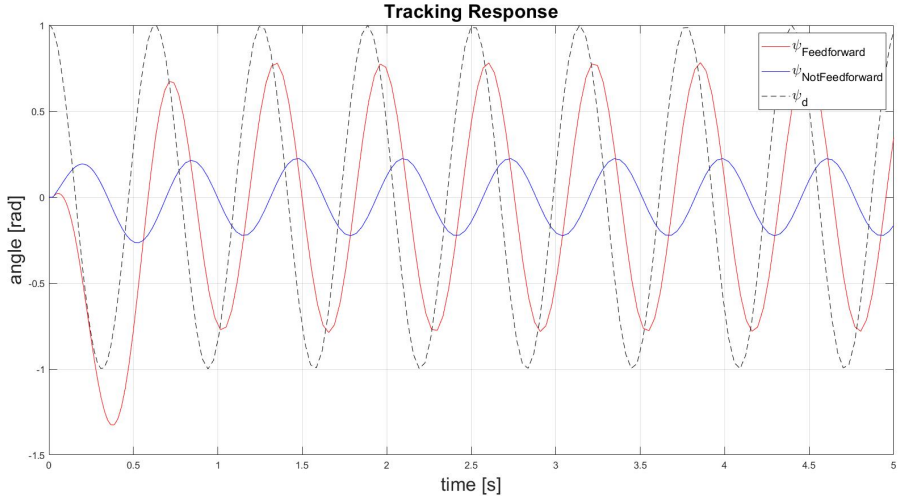

Figure 6.13: Tracking response: Yaw axis

For all three axes, although the system is linearized, it is possible to preliminary determine that the feedforward term is capable of reducing the amplitude of error by better tracking the setpoint.

6.2.2. UAV software simulator

The UAV simulator is composed of four different subsystems; the first one at the top-left (Figure 6.14), is a Matlab function that produces the desired setpoint (whether for position and velocity or the attitude and angular velocity); the second one, below the first

one (Figure 6.14), produces the state of the UAV (NED position, NED velocity, attitude and angular velocity), adding a white noise to them, in order to reproduce a realistic signal coming from the sensors.

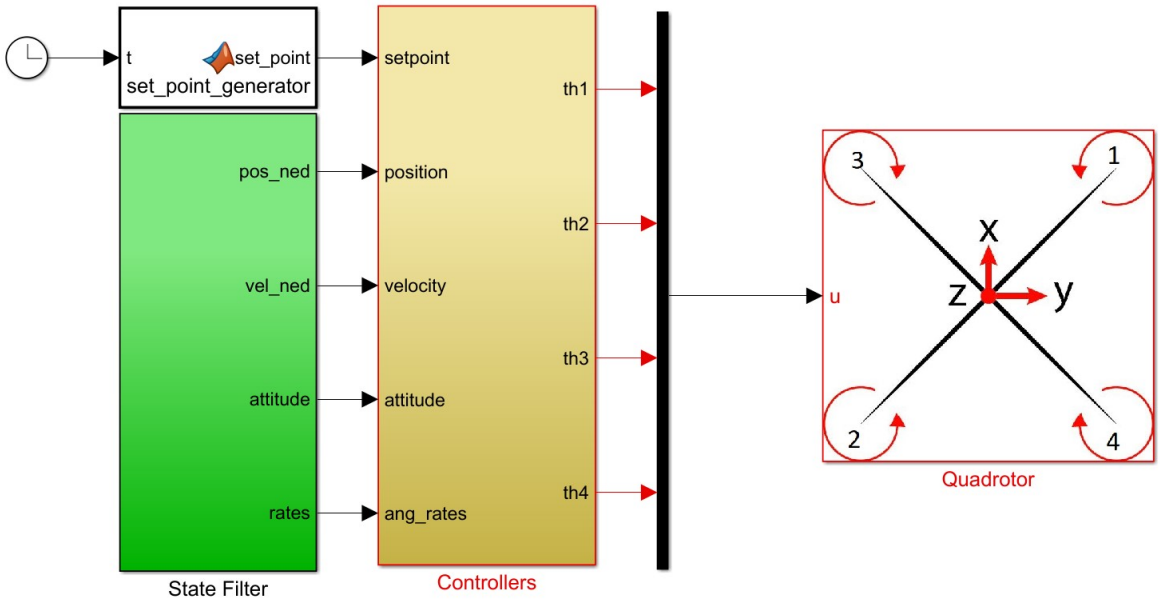

Figure 6.14: ANT-X UAV: Simulink scheme

The third component is the block in the center, at the right of the first two (6.14), and it is devoted to taking the setpoint signals and the state as inputs and producing the required thrust for the propellers; to do this, referring to Figure 6.15, the signals first enter in the blue block, which is a position controller, where an augmented adaptive control law is implemented to produce the required force and the desired attitude, in particular an MRAC law and the L_1 is implemented; for more details, see [13].

Then, the desired attitude enters inside the grey box (Figure 6.15) in order to compute the desired angular velocity, using the model developed in Subsection 2.3.5. The angular velocity, attitude, and their corresponding desired signals enter the lower box inside Figure 6.15, which is the attitude controller, where the thesis work is more concentrated; here, the attitude controller produces the desired torque to be applied.

Finally, the desired force and torque enter inside the orange box (Figure 6.15). The thrust percentage is obtained thanks to a control allocation, together with an adding of the saturation to the signal.

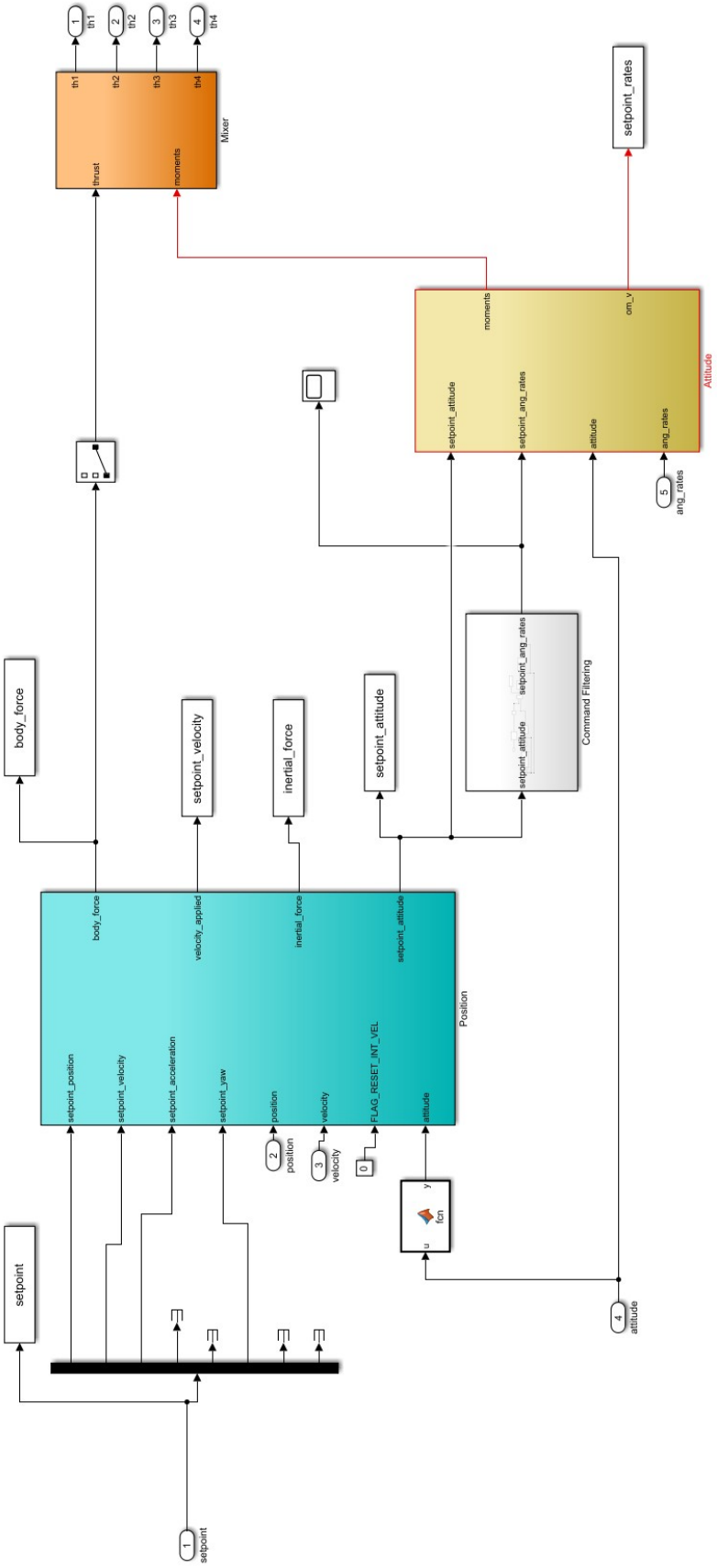

Figure 6.15: ANT-X UAV Quadrotor Controllers: Simulink scheme

Finally, after the percentage of thrust is obtained by the controller, they are fed into the last subsystem on the right (Figure 6.14), which is the UAV physical system, where the percentage of thrust is processed and translated as propellers force and torque.

From the propeller's force; the position, the velocity, and the acceleration (both in body and NED references) are obtained. While, from the torque of the propeller, the angular velocity and the attitude are computed. Both the computation is made according to the mathematical model presented in Subsection 2.2.2.

Figure 6.16 presents the quadrotor mathematical model, exploited in 2.2.2 and, at the bottom of the simulink scheme, when all the states are computed, they are fed into the first subsystem in green (Figure 6.14), so making the UAV loop end.

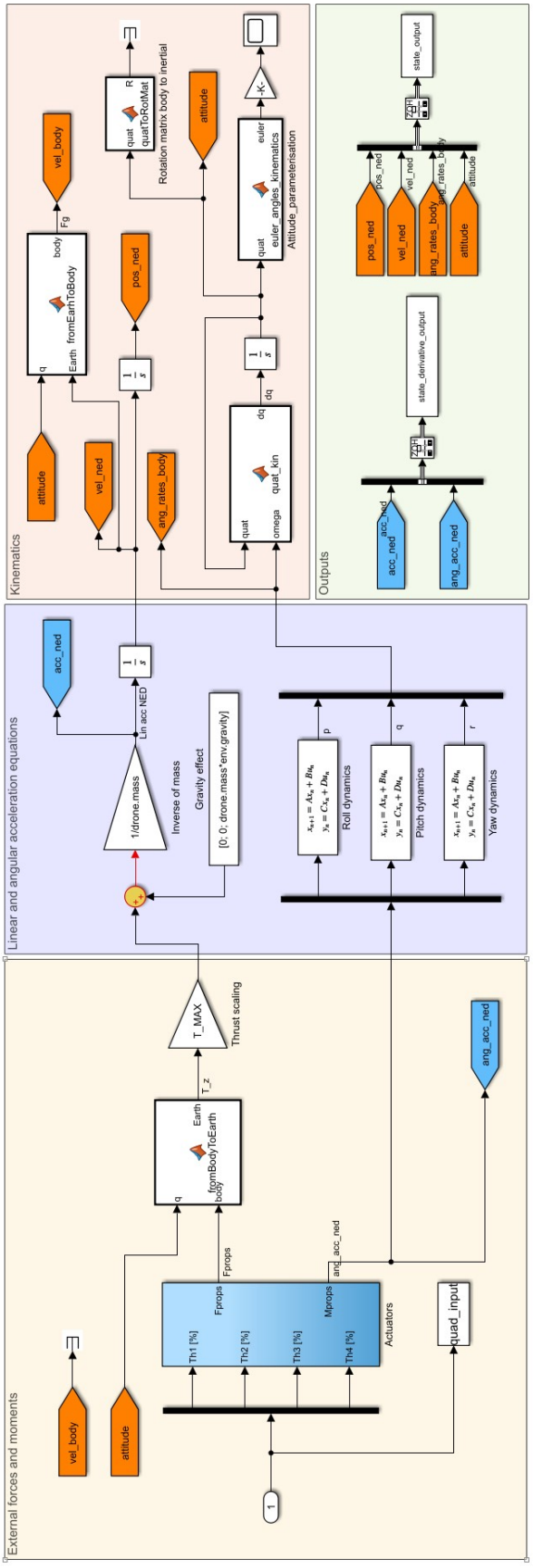

Figure 6.16: ANT-X UAV Quadrotor: Simulink scheme

In this subsection, after dealing with the linearized model of the UAV quadrotor, it is time to look at the entire nonlinear plant and its behavior. Since the system is now more complex than before, an analytical angular velocity is no longer used, it is then computed using the filtered derivative presented in Subsection 2.3.5; natural frequency (ω_n) for the filter, used in Equation (2.37), is set to $20\,rad/s$, both for hierarchical and geodesic law.

6.2.3. Hierachical control law

Regarding the Hierarchical control law architecture developed in Section 2.3, the objective is to test the performance of the UAV in tracking a circular trajectory at an altitude of $-1.5\,m$ for a small maneuver ($1.5\,rad/s$) and a high-velocity maneuver ($2.5\,rad/s$) and compare the performance with and without the feedforward term (equation (2.23a)).

The simulation plots report the position trajectory in the 2D plane and the attitude response and corresponding error. Two case are considered, correspoding to circular trajectories at $1.5\,rad/s$ and $2.5\,rad/s$. For each one, the result with and without the feedforward term are reported for comparison.

1. **Circular trajectory at 1.5 rad/s without feedforward term**

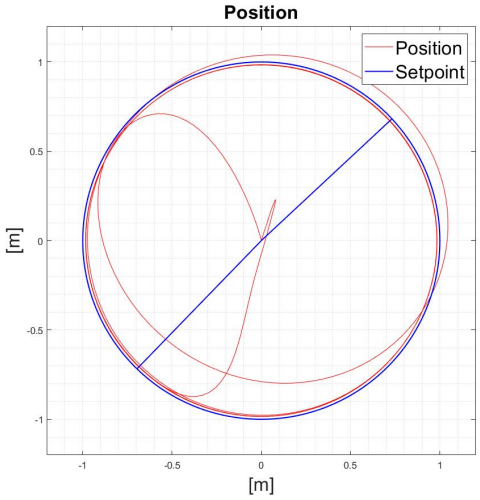

Figure 6.17: 2-D trajectory: 1.5 rad/s, no feedforward

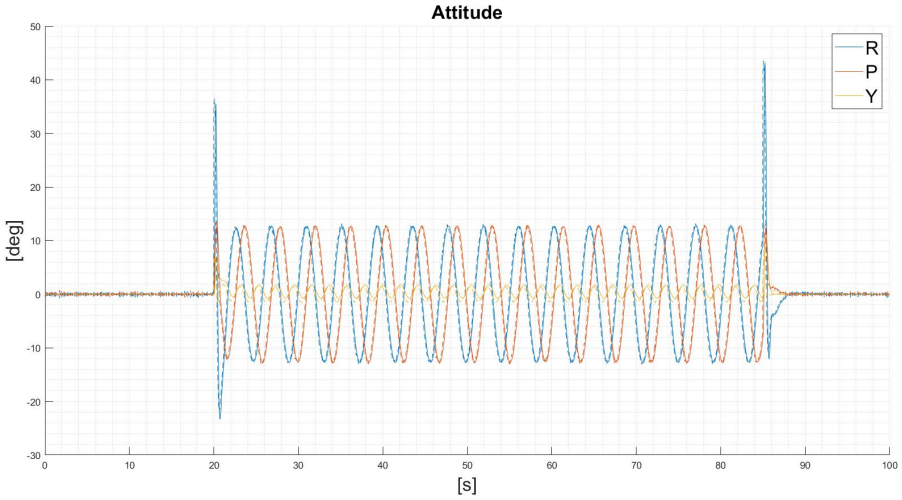

Figure 6.18: Attitude response: 1.5 rad/s, no feedforward

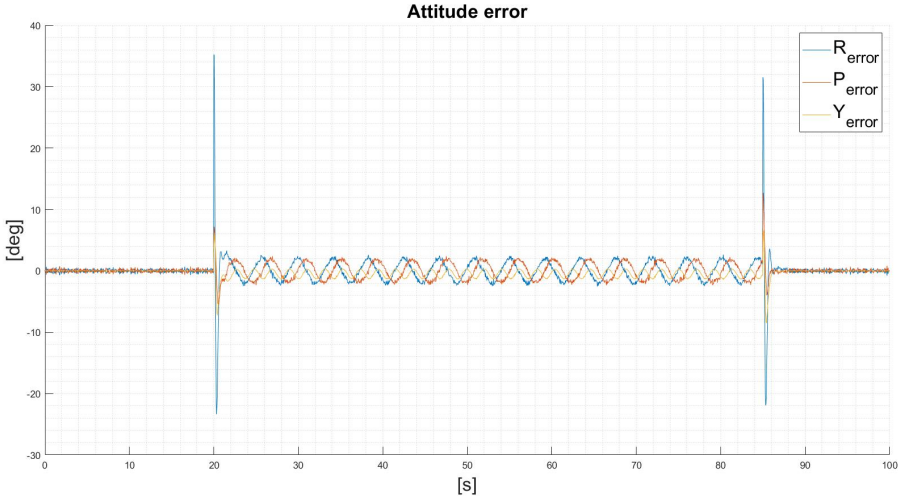

Figure 6.19: Attitude error response: 1.5 rad/s, no feedforward

2. **Circular trajectory at 1.5 rad/s with feedforward term**

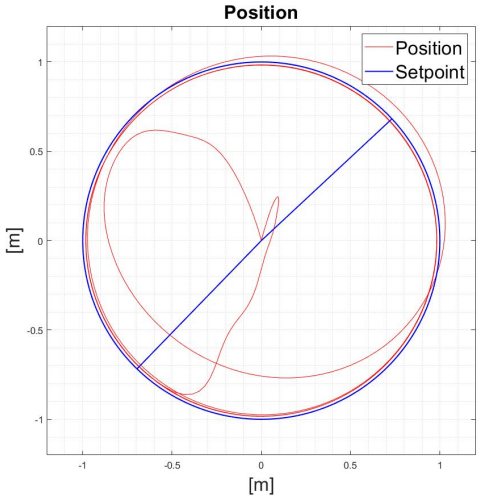

Figure 6.20: 2-D trajectory: 1.5 rad/s, feedforward

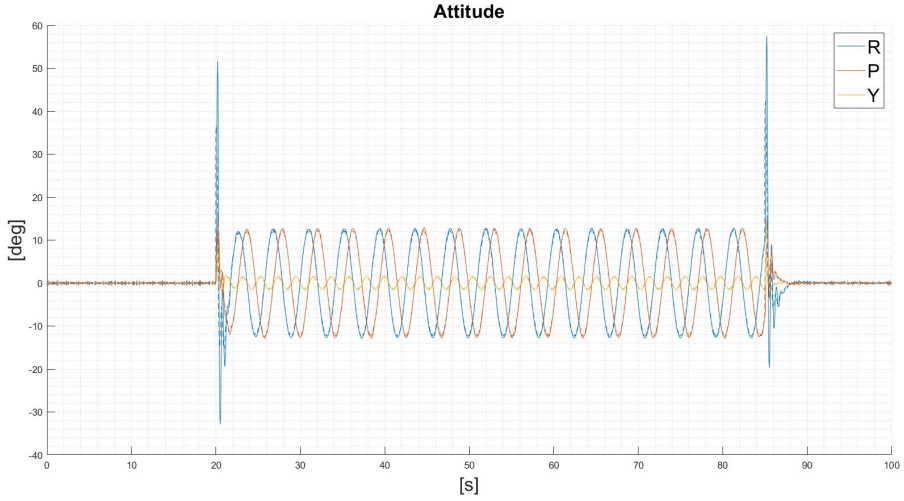

Figure 6.21: Attitude response: 1.5 rad/s, feedforward

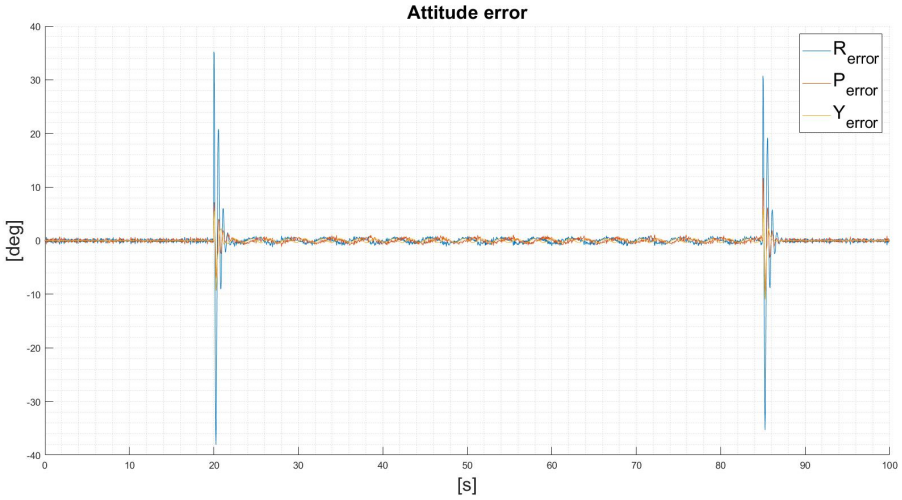

Figure 6.22: Attitude error response: 1.5 rad/s, feedforward

3. **Circular trajectory at 2.5 rad/s without feedforward term**

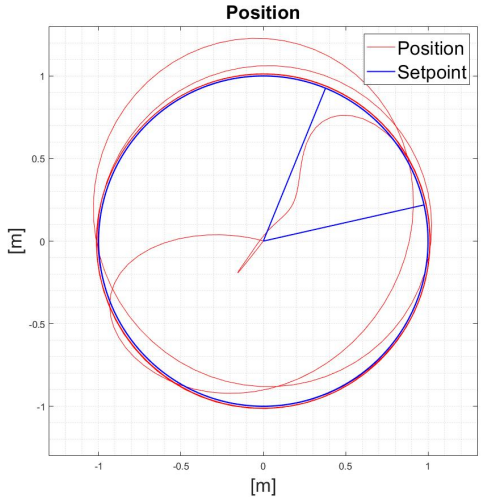

Figure 6.23: 2-D trajectory: 2.5 rad/s, no feedforward

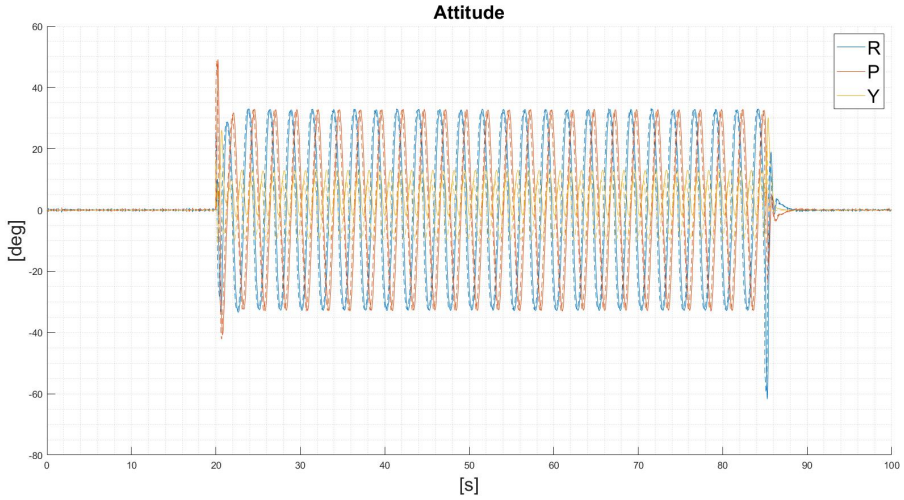

Figure 6.24: Attitude response: 2.5 rad/s, no feedforward

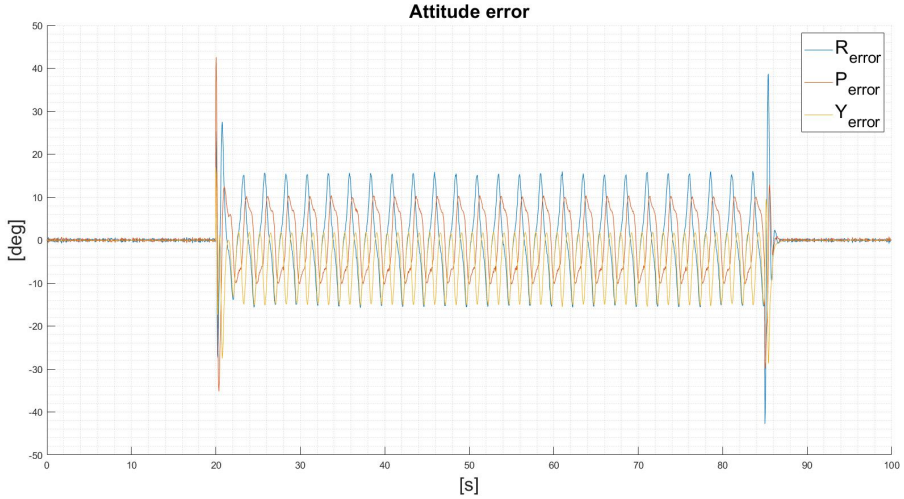

Figure 6.25: Attitude error response: 2.5 rad/s, no feedforward

4. **Circular trajectory at 2.5 rad/s with feedforward term**

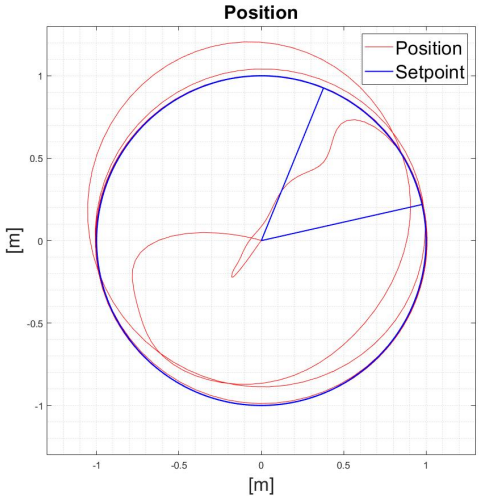

Figure 6.26: 2-D trajectory: 2.5 rad/s, feedforward

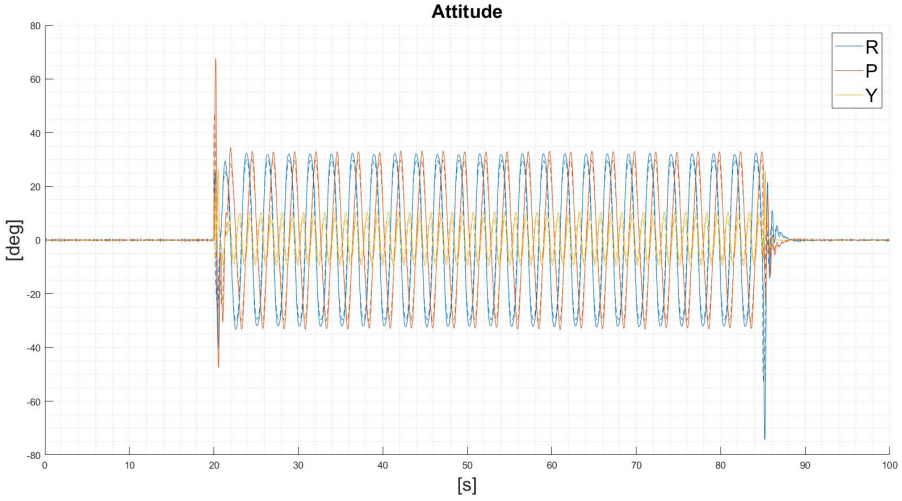

Figure 6.27: Attitude response: 2.5 rad/s, feedforward

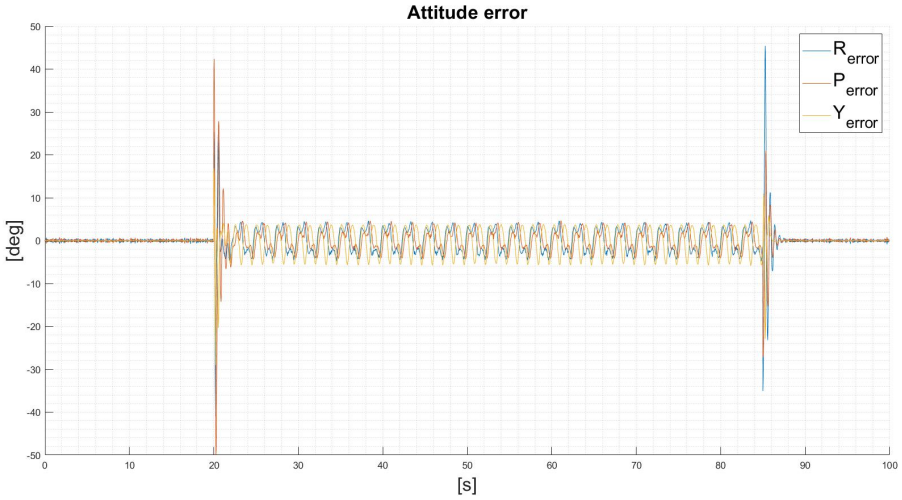

Figure 6.28: Attitude error response: 2.5 rad/s, feedforward

Once the simulation is done, it could be advantageous to compute the root mean square and the mean value of the norm of the angle error, to see the improvement of the performance, thanks to the modification of the attitude controller, using the following formulas:

$$\begin{cases} \mu_{norm,error} = \dfrac{1}{N}\displaystyle\sum_{i=1}^{N}|e_i| & (6.2) \\ RMS_{norm,error} = \sqrt{\dfrac{1}{N}\displaystyle\sum_{i=1}^{N}|e_i|^2} & (6.3) \end{cases}$$

The tables are shown of $\mu_{norm,error}$ and $RMS_{norm,error}$ for the four simulations are:

	$\Omega = 1.5\,rad/s$	$\Omega = 2.5\,rad/s$
feedforward	0.83	3.61
not feedforward	1.70	9.1

Table 6.5: Mean of the angle norm error [cm]

	$\Omega = 1.5\,rad/s$	$\Omega = 2.5\,rad/s$
feedforward	2.90	5.83
not feedforward	2.90	12.26

Table 6.6: RMS of the angle norm error [cm]

So, thanks to the performance index determined in Tables 6.5 and 6.6, it is possible to demonstrate that the feedforward helps improving significantly the tracking performance, independently of the choice to use feedforward or not, the increase of Ω leads to a decrease of the tracking performance, as expected.

6.2.4. Geodesic control law

This subsection deals with the application of the geodesic control law to the UAV simulator; the main goal here is to see the comparison between the performance of Hierarchical control law shown in 6.2.3 and the Geodesic control, including in both the feedforward term.

Another goal of this subsection is to underline that the geodesic feedback law has been introduced to avoid the propeller's saturation, which can cause windup effects that lead to a loss of performance.

When the geodesic term is considered, the feedforward term will act only for the first two angular velocity terms, while the third is considered to be zero; this is because the third axis must acts only as the geodesic controller, without any other input (in this case, the feedforward term).

The gains here are set as follows:

$$\begin{cases} k_{max} = 3 \\ k_{min} = 0.01 \\ k_2 = 5 \end{cases} \tag{6.4}$$

The natural frequency ω_n is now set to $30\,rad/s$.

The simulation consists of a trajectory where all the desired positions and angles are null, also used for the experimental and simulation validation in the Paper [15], except for:

$$\begin{cases} y_d(t) = 4\, step(t-1)\, m \\ z_d(t) = -2\, step(t-1)\, m \\ \psi_d(t) = \hat{\psi}\, step(t-1)\, deg \end{cases} \tag{6.5}$$

$\hat{\psi}$ first be set to $30\, deg$, and it is tested both using hierarchical and geodesic; then, the same test will be done, but using a larger angle, to better emphasize saturation effects, setting $\hat{\psi}$ equal to $120\, deg$.

1. **Hierarchical controller, yaw angle equal to 30°**

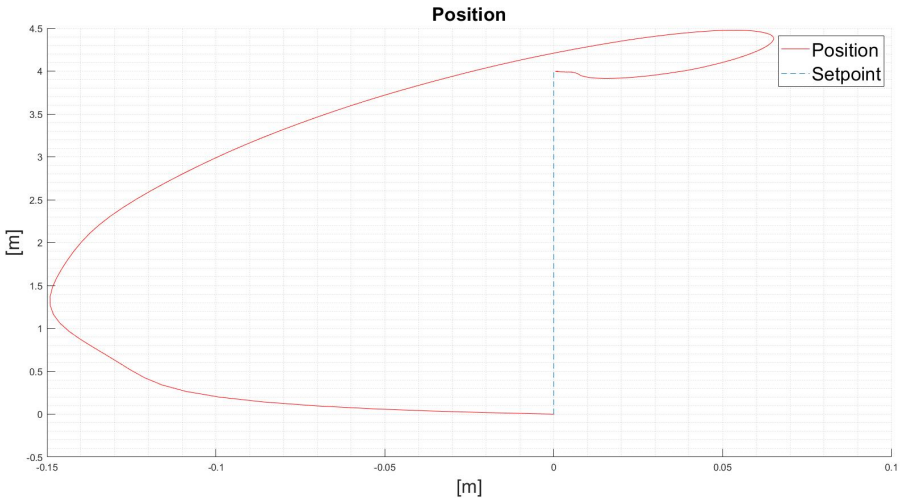

Figure 6.29: Trajectory (x-y plane): Hierarchical, $\hat{\psi} = 30\, deg$

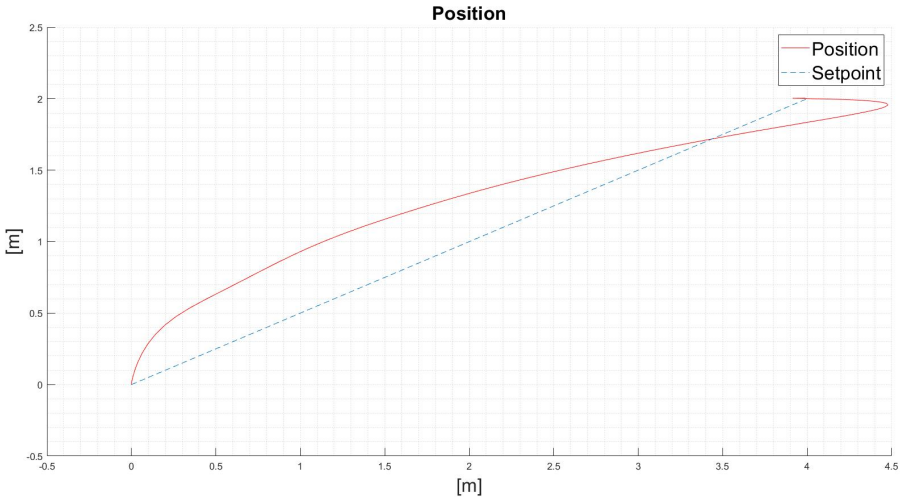

Figure 6.30: Trajectory (y-z plane): Hierarchical, $\hat{\psi} = 30\, deg$

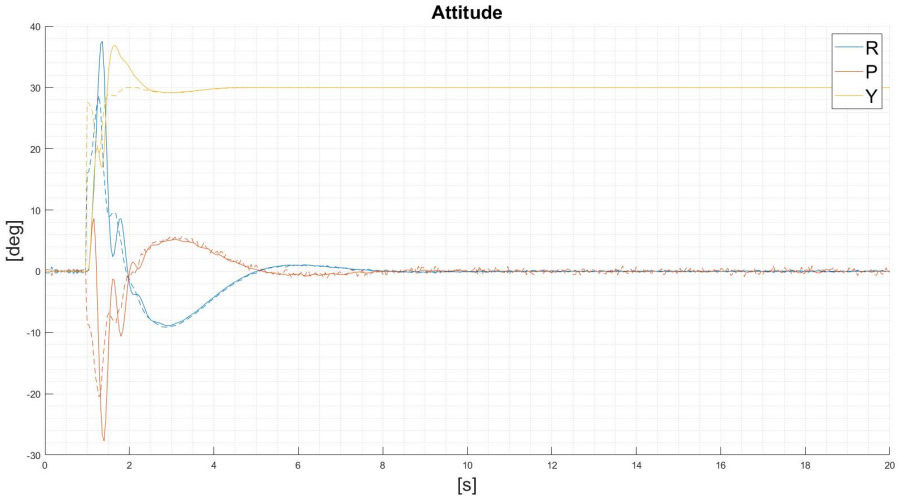

Figure 6.31: Attitude response: Hierarchical controller, $\hat{\psi} = 30\,deg$

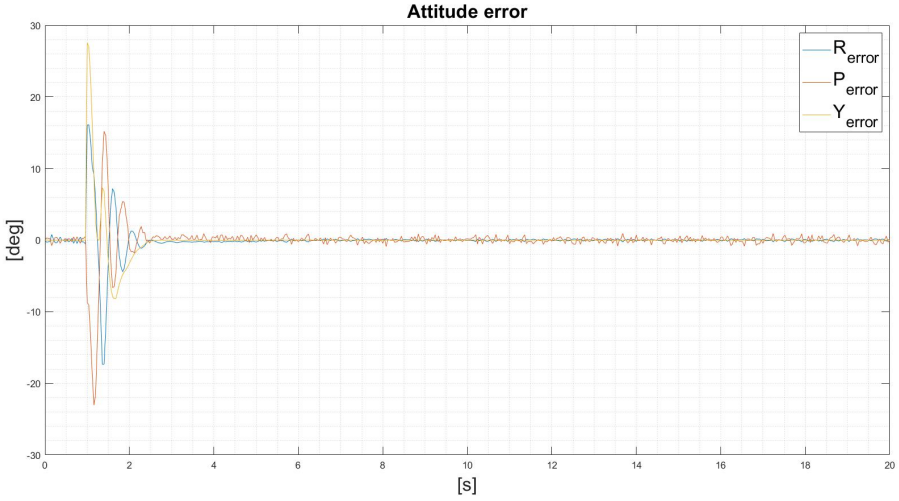

Figure 6.32: Attitude error response: Hierarchical controller, $\hat{\psi} = 30\,deg$

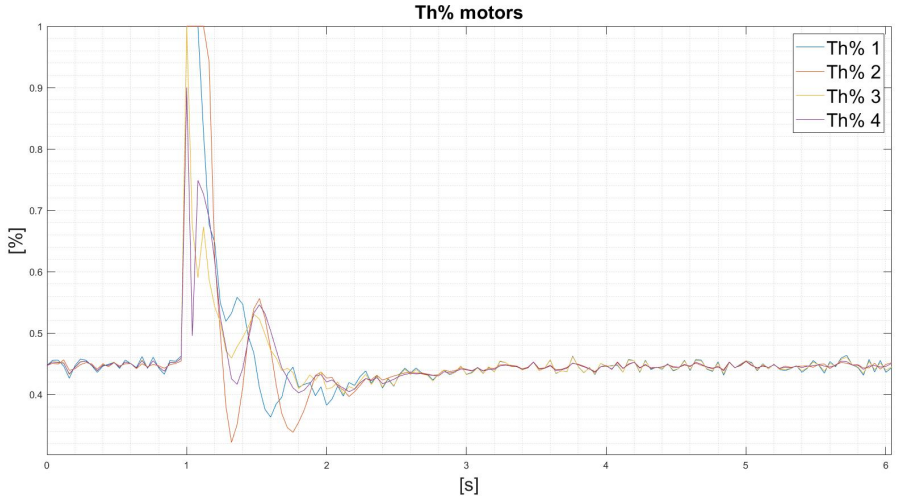

Figure 6.33: Propeller's thrust: Hierarchical controller, $\hat{\psi} = 30\,deg$

2. **Geodesic controller, yaw angle equal to 30°**

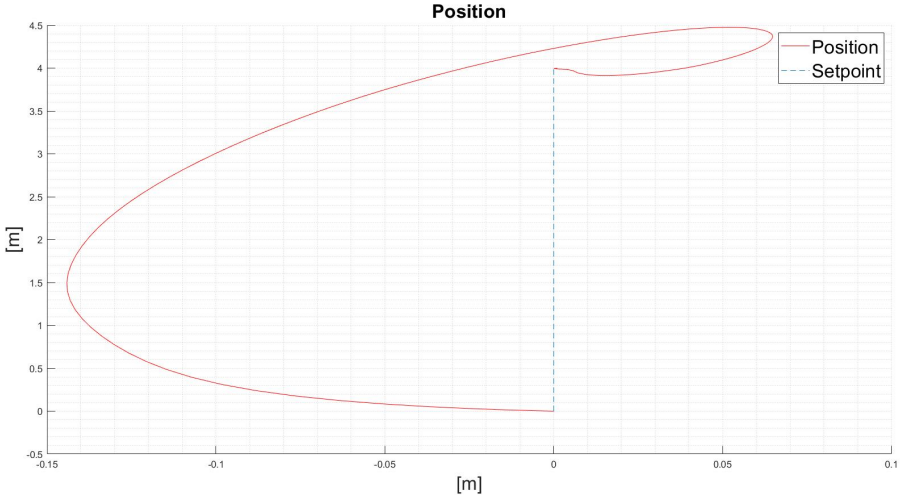

Figure 6.34: Trajectory (x-y plane): Geodesic, $\psi = 30\,deg$

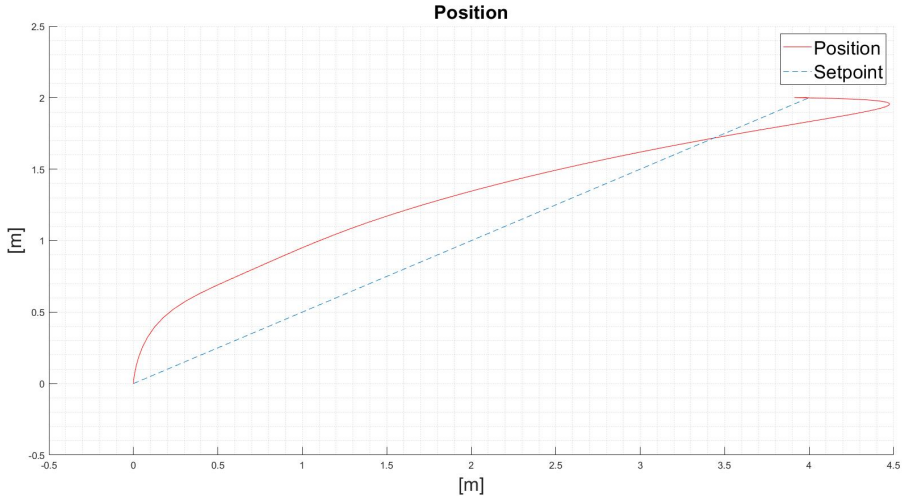

Figure 6.35: Trajectory (y-z plane): Geodesic, $\psi = 30\,deg$

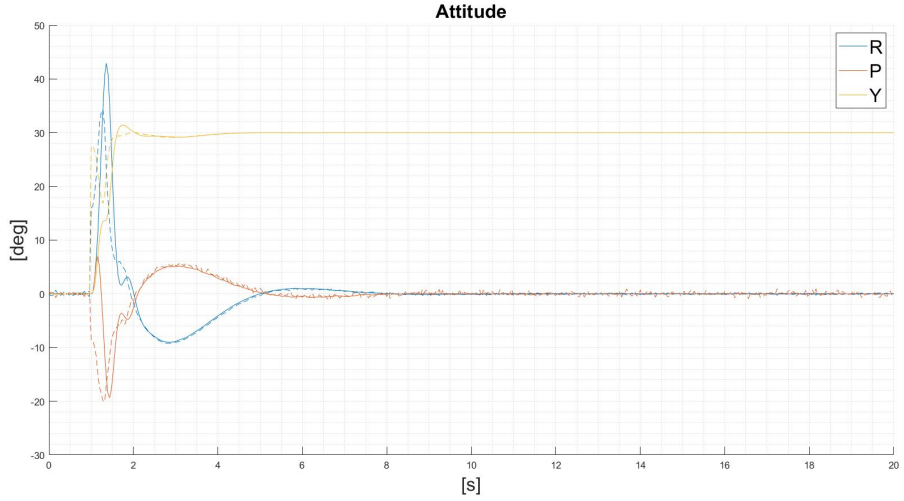

Figure 6.36: Attitude response: Geodesic controller, $\hat{\psi} = 30\,deg$

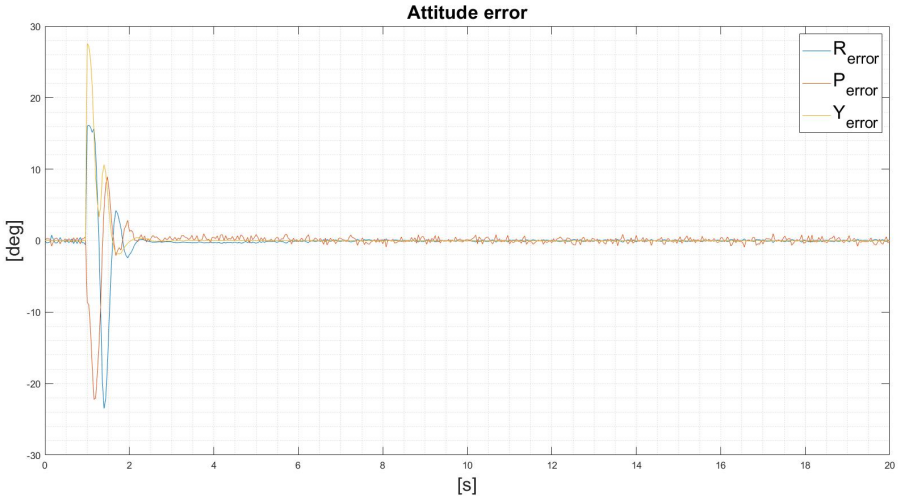

Figure 6.37: Attitude error response: Geodesic controller, $\hat{\psi} = 30\,deg$

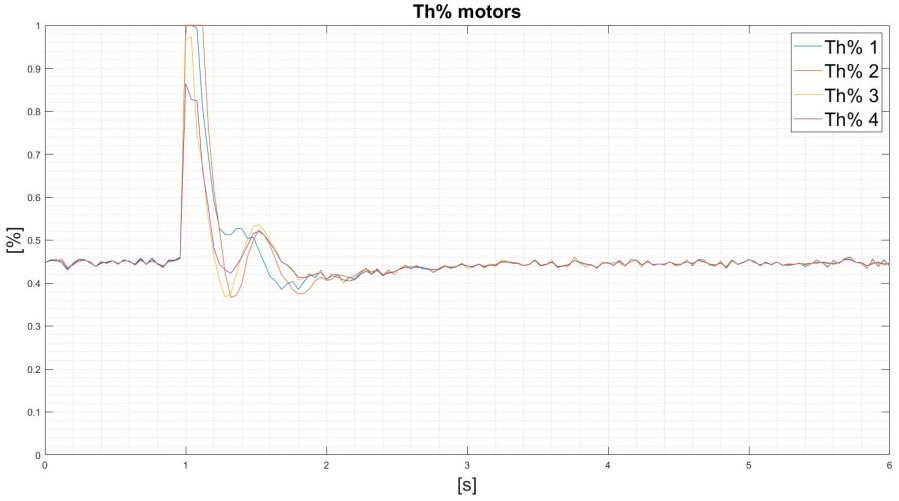

Figure 6.38: Propeller's thrust: Geodesic controller, $\hat{\psi} = 30\,deg$

3. **Hierarchical controller, yaw angle equal to 120°**

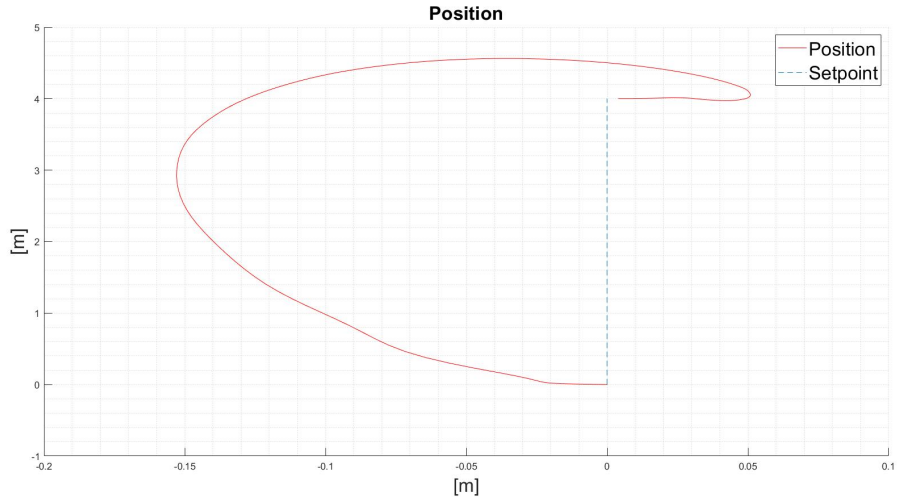

Figure 6.39: Trajectory (x-y plane): Hierarchical controller, $\hat{\psi} = 120\,deg$

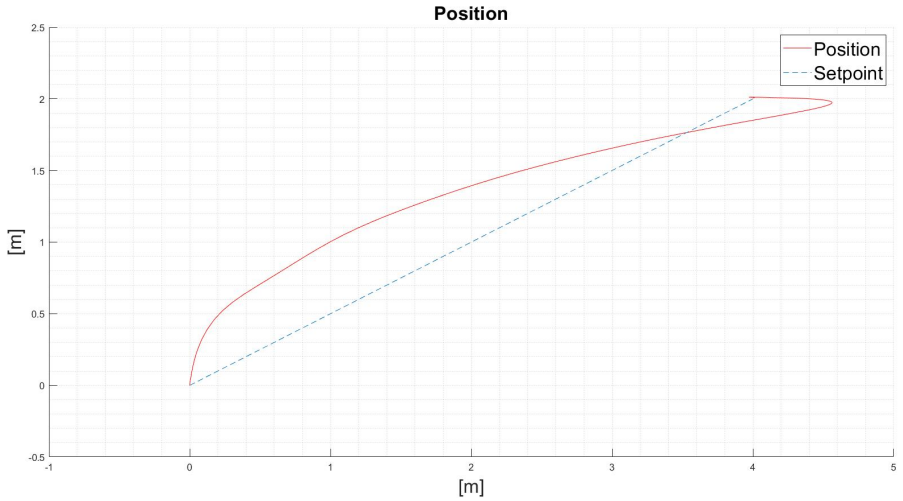

Figure 6.40: Trajectory (y-z plane): Hierarchical controller, $\hat{\psi} = 120\,deg$

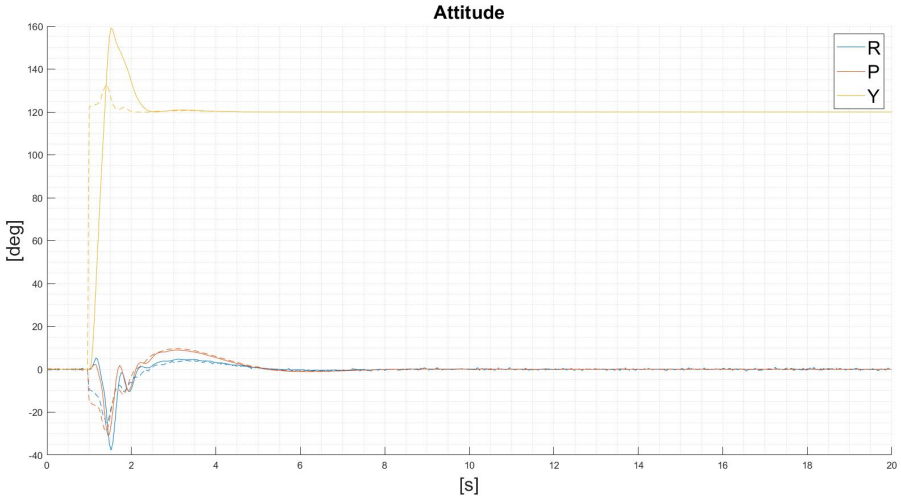

Figure 6.41: Attitude response: Hierarchical controller, $\hat{\psi} = 120deg$

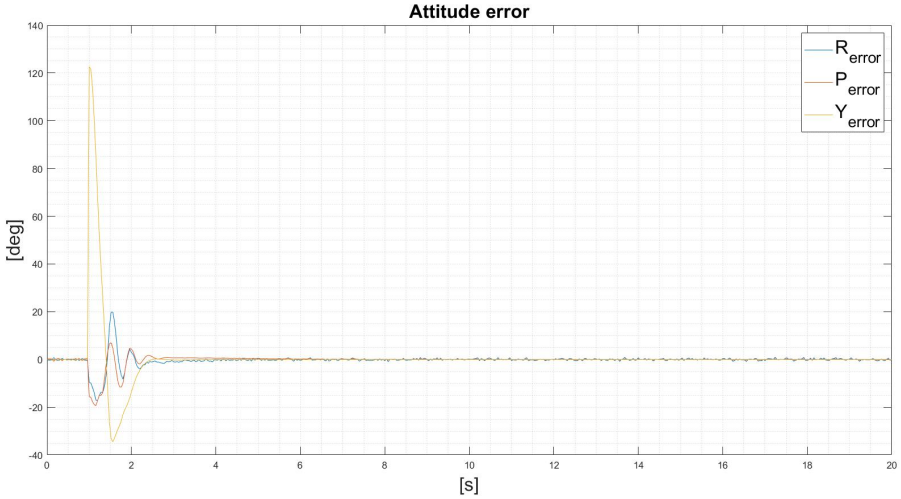

Figure 6.42: Attitude error response: Hierarchical controller, $\hat{\psi} = 120deg$

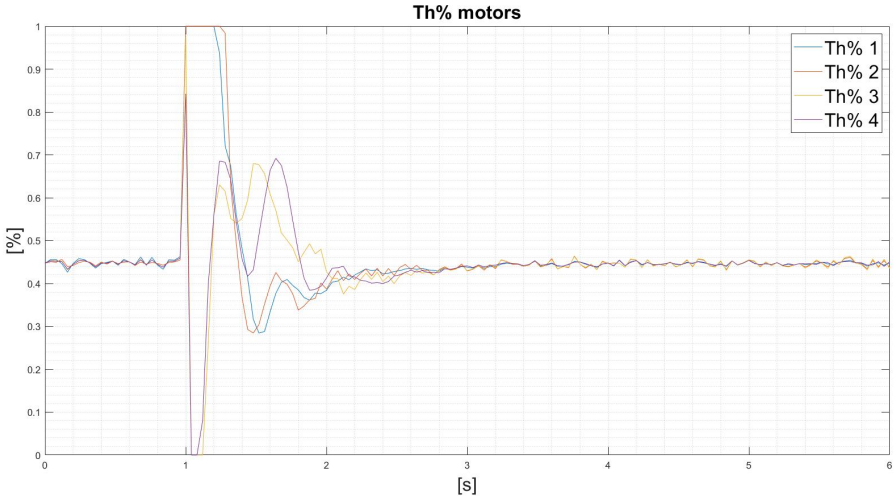

Figure 6.43: Propeller's thrust: Hierarchical controller, $\hat{\psi} = 120\,deg$

4. **Geodesic controller, yaw angle equal to 120°**

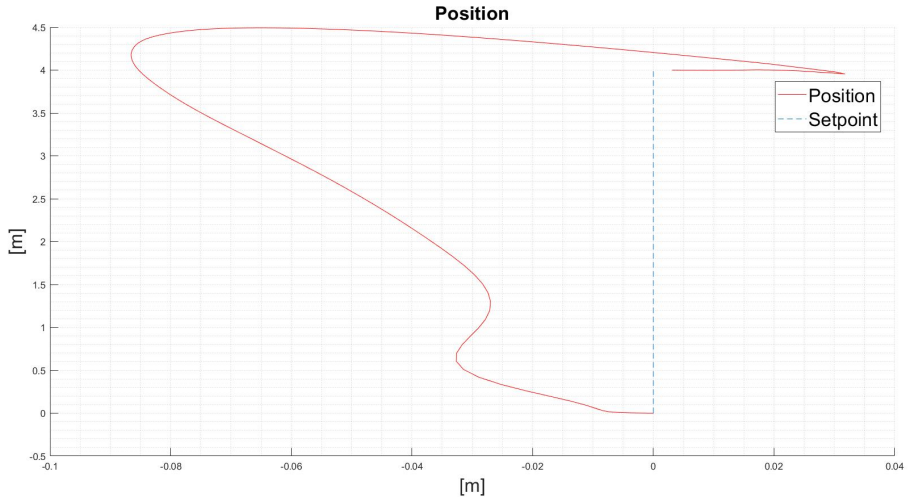

Figure 6.44: Trajectory (x-y plane): Geodesic controller, $\hat{\psi} = 120\,deg$

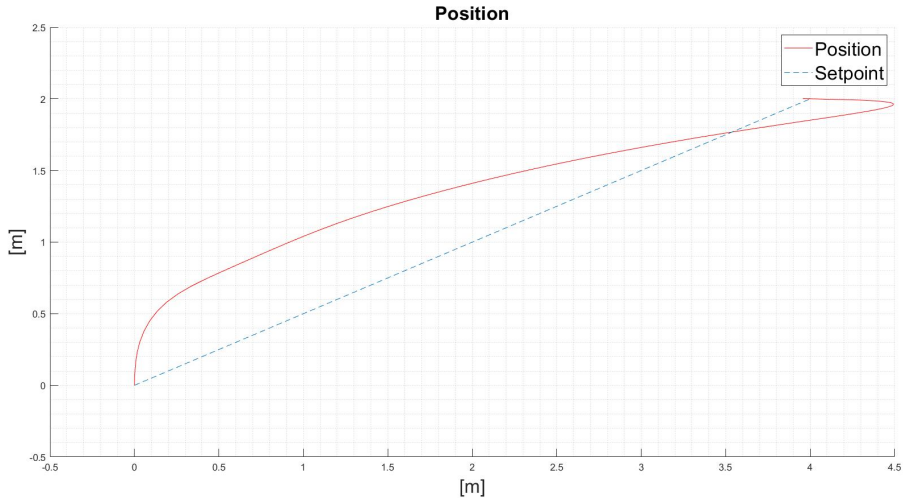

Figure 6.45: Trajectory (y-z plane): Geodesic controller, $\hat{\psi} = 120\,deg$

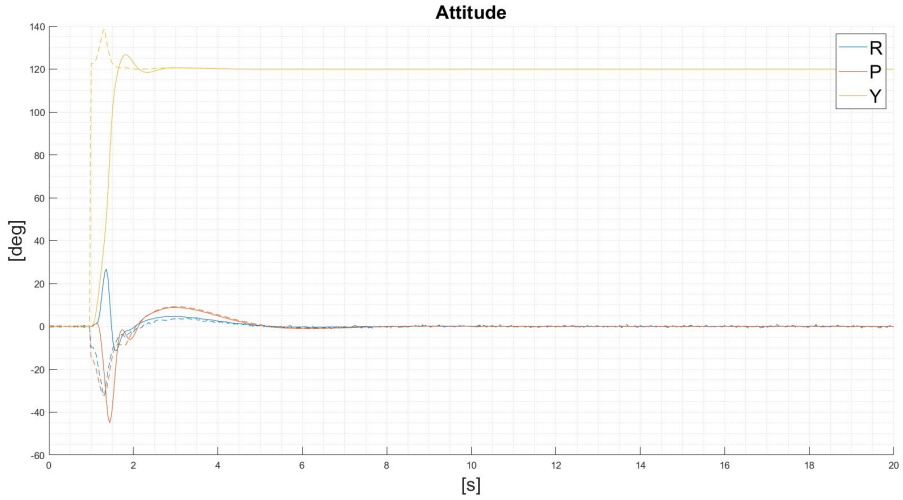

Figure 6.46: Attitude response: Geodesic controller, $\hat{\psi} = 120deg$

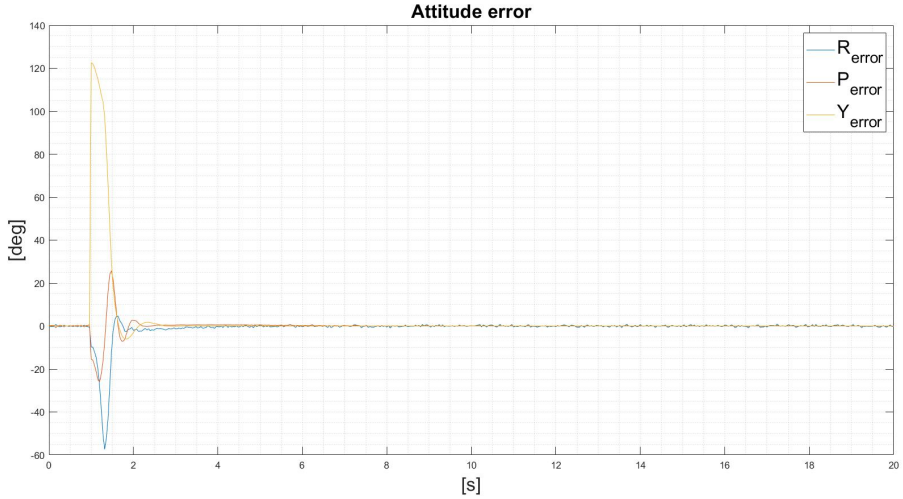

Figure 6.47: Attitude error response: Geodesic controller, $\hat{\psi} = 120deg$

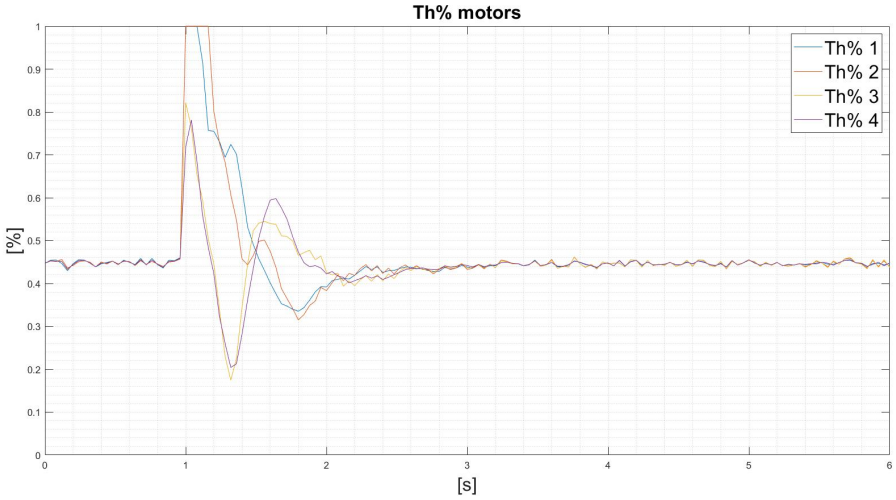

Figure 6.48: Propeller's thrust: Geodesic controller, $\hat{\psi} = 120\,deg$

Also, for geodesic feedback law, it is possible to retrieve some performance index; in particular, for the yaw-step response, it can be helpful to confront the percentage of overshoot, defined as follows:

$$overshoot_{percentage} = \frac{\psi_{max} - \hat{\psi}}{\hat{\psi}} * 100\,[\%] \tag{6.6}$$

The table is shown as follows:

	$\hat{\psi} = 30\,deg$	$\hat{\psi} = 120\,deg$
Hierarchical	22.92	32.64
Geodesic	4.64	5.57

Table 6.7: Overshoot percentage of yaw step response

The second performance index computed is the maximum displacement of the norm of x_{NED} concerning its reference, which is considered null, so the formula is defined as:

$$\Delta x_{max} = max(|x_{NED}|) * 100\,[cm] \tag{6.7}$$

	$\hat{\psi} = 30\,deg$	$\hat{\psi} = 120\,deg$
Hierarchical	14.9	15.3
Geodesic	14.4	8.65

Table 6.8: Maximum displacement of the x_{NED} coordinate

The geodesic law allows decreasing the overshoot percentage of the yaw step response significantly, concerning hierarchical architecture.

Another benefit of the simulation is that the maximum displacement of the x_{NED} is decreased using a geodesic architecture.

Also here, when a high maneuver is requested, the index performance increases, leading to a performance deterioration.

Regarding the thrust percentage, in the first experiment, both the architectures have the same amount of time of saturation of the propellers, but in hierarchical control, there are more motors in saturation than the geodesic feedback. This effect is amplified in the second experiment, where it is possible to see that the amount of time the motors stay in saturation is in the case of the hierarchical control law.

6.3. Experiments

6.3.1. Experimental equipment and setup

Figure 6.49: ANT-X UAV (source [1])

All experiments have been done in the Flying Arena for Rotorcraft Technologies (Fly-ART) of Politecnico di Milano. This structure features an indoor facility with a flight volume of 6x12x4m and is equipped with a 3D motion capture system (OptiTrack). The quadrotor on which all the control algorithms have been tested is the ANT-X UAV (figure 6.49) and its main characteristics are summarized in Table 6.9. The Flight Control Unit (FCU) is a Pixfalcon board (in figure 6.49), an open autopilot shield that can be used for multirotor and fixed-wing aircraft. It has a three-axis accelerometer, a three-axis gyroscope, a magnetometer, and a barometer. The firmware running on the board is the open-source software PX4 Pro Autopilot [3]. This firmware features attitude and position controllers and estimators, and it has been customized by the ANT-X rapid prototyping system for multirotor control [1] to allow replacing the baseline attitude and position controllers with user-defined ones. In addition to the FCU, a companion computer (figure 6.51) is also installed onboard: it is used to exchange information with the ground control station and to receive data from the motion capture system.

ANT-X main specifications		
Take Off Weight (TOW)	0.310	[kg]
Frame size	0.2 × 0.2 × 0.04	[m]
Rotors configuration	X	[-]
Propellers No	4	[-]
Blades № per rotor	3	[-]
Propeller diameter	3	[inch]
Propeller max thrust $\tau_{max,i}$	1.7	[N]
Saturation limit	70	[%]
Battery LiPo	950	[mAh]
Flight time (hovering)	6.5	[min]

Table 6.9: ANT-X main parameters

Figure 6.50: Pixfalcon FCU (source [3])

Figure 6.51: NanoPi NEO Air companion (source [2])

6.3.2. Hierarchical control law

The first experiment considered is a circular trajectory at $2\,rad/s$, considering first the control law with the feedforward effect and, then without it, with a natural frequency of the command filter ω_n equal to $20\,rad/s$. Regarding the position controller, the solution presented in [13] is implemented. The same gains used in the simulations are considered here.

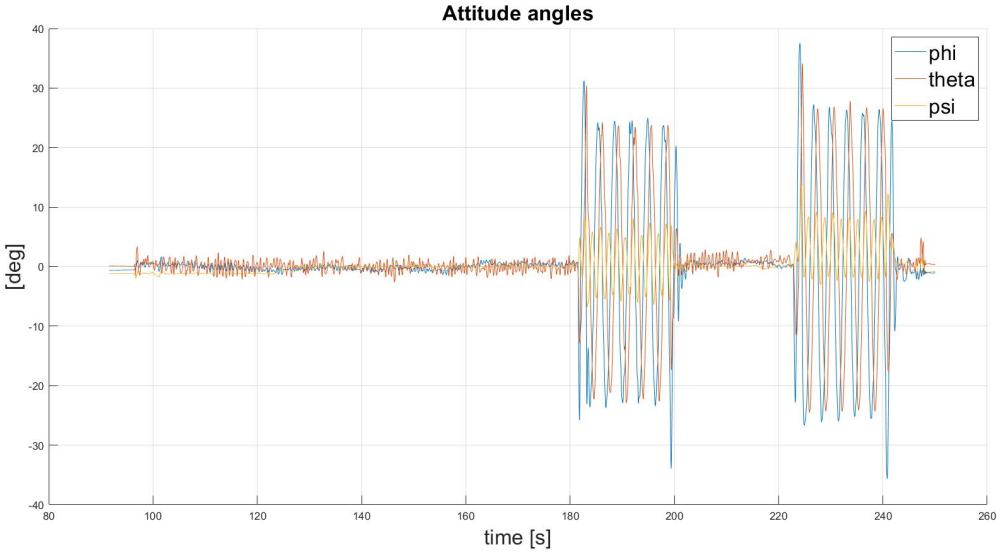

Figure 6.52: First Experiment Hierarchical (Attitude angle response)

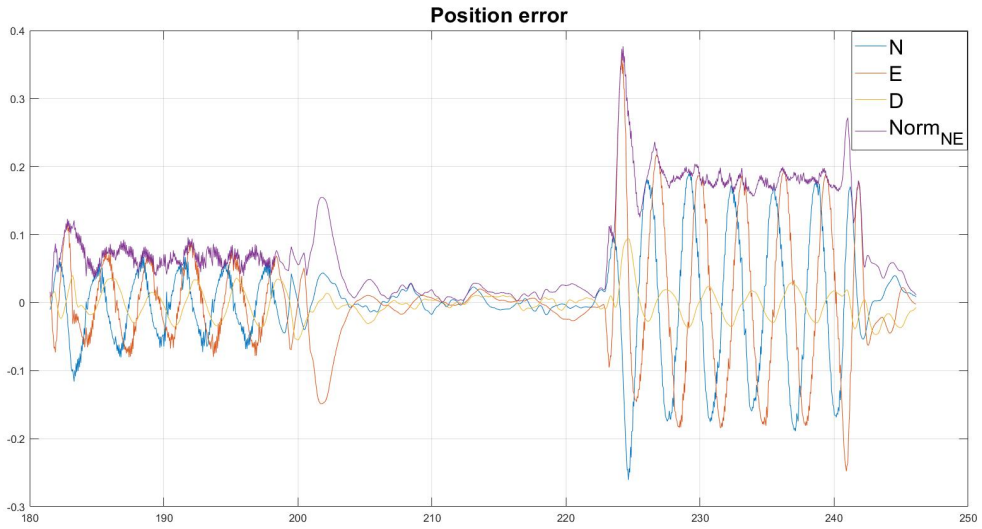

Figure 6.53: First Experiment Hierarchical (position error)

The Hierarchical control law has a lower position error in the first part of the test than in the second part, where the feedforward action is not considered. It is also possible to compute the RMS of the norm of the position error, and the results are, for this experiment:

$$\begin{cases} RMS_{feedforward} = 7.07m \\ RMS_{no,feedforward} = 19.26m \end{cases} \tag{6.8}$$

In the second experiment, the UAV is tested with a faster maneuver, a circular trajectory where this time $\Omega = 2.5rad/s$ and the natural frequency of the command filter has been set to $\omega_n = 30rad/s$. This time the first trajectory is made without a feedforward term, subsequently, for the second part, it has been added.

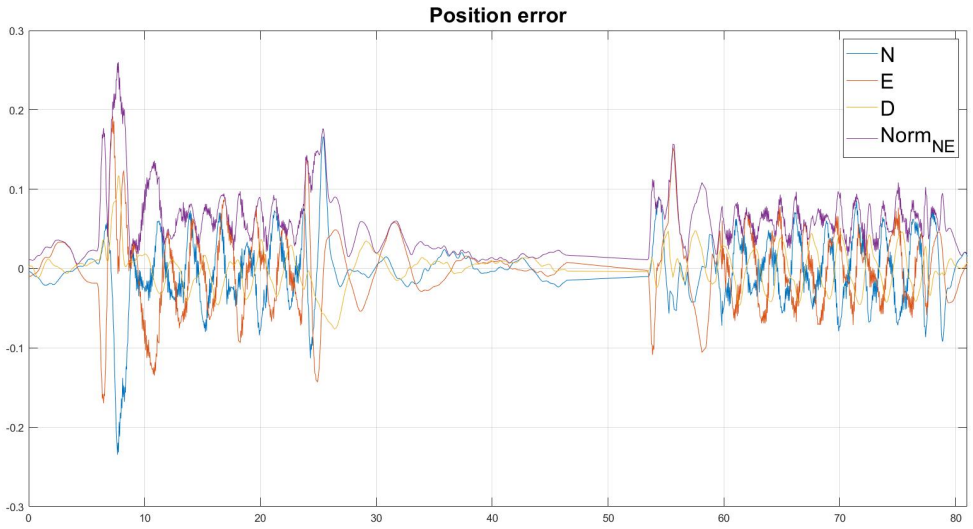

Figure 6.54: Second Experiment Hierarchical (Attitude angle response)

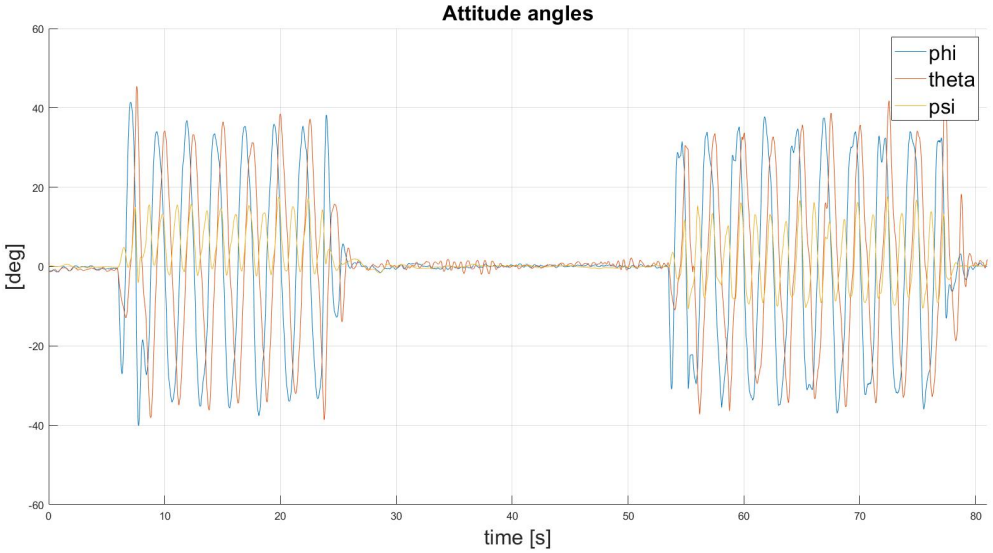

Figure 6.55: Second Experiment Hierarchical (position error)

The UAV can track the circular trajectory well; also, when the angular rate of the trajectory is higher, the feedforward term is helpful in making the position error smaller. Also, in this case, indeed, it is possible to compute the RMS and see that:

$$\begin{cases} RMS_{feedforward} = 6.52\,cm \\ RMS_{no,feedforward} = 9.76\,cm \end{cases} \tag{6.9}$$

6.3.3. Geodesic control law

The experiments of this subsection confront the geodesic and hierarchical control law using two different yaw angles (first $90\,deg$ and, then, $135\,deg$). The feedforward term is always non-null and the filter frequency ω_n has been set to $30\,rad/s$. The first experiment is a trajectory similar to the one presented in 6.2.4, but with the differences that the saturation has been fixed to $70\,\%$ of the thrust percentage and with the initial position is:

$$x_0 = [0, -2, -0.5] \tag{6.10}$$

and then, the final position will become:

$$x_{fin} = [0, 2, -2.5] \tag{6.11}$$

This experiment has been repeated two times, first using hierarchical control law and then geodesic law.

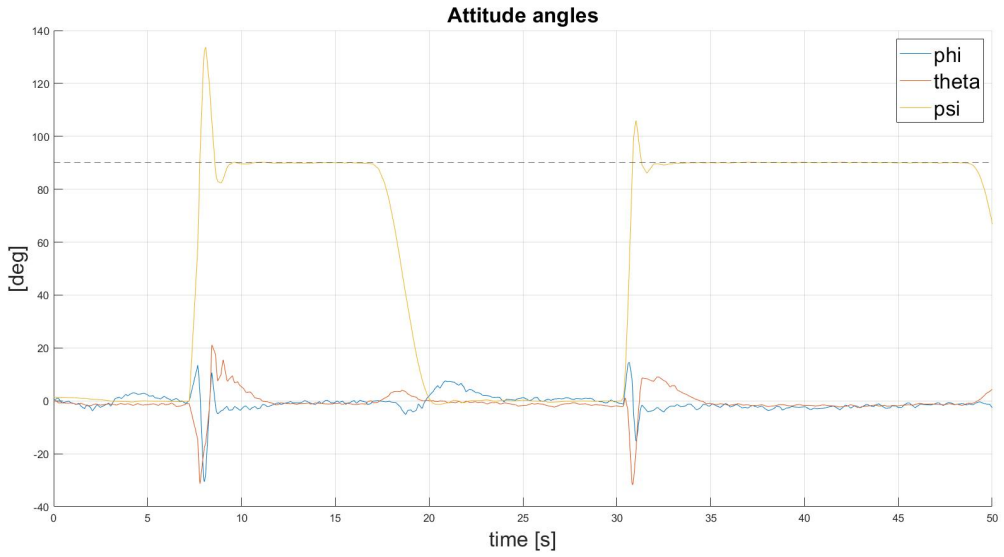

Figure 6.56: First Experiment Geodesic vs. Hierarchical (Attitude angle response)

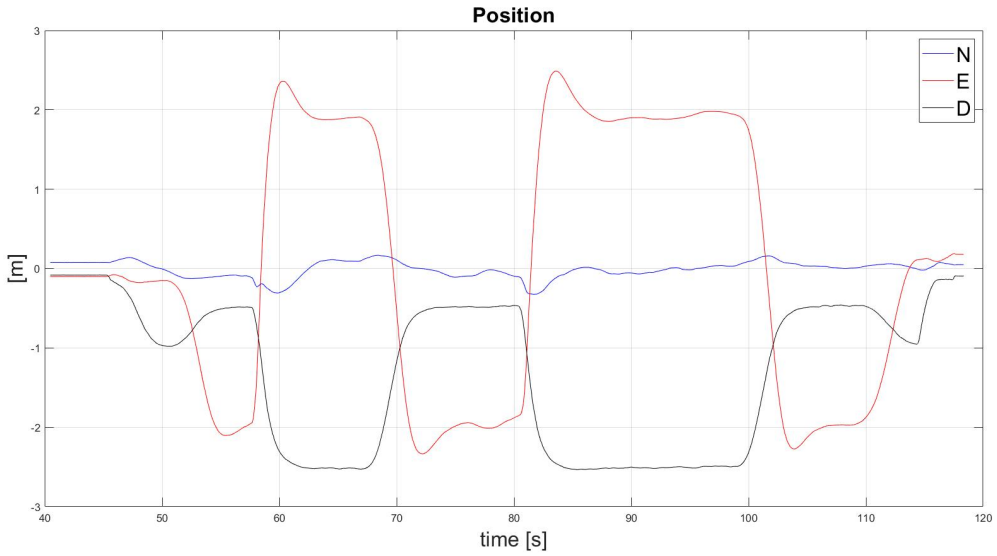

Figure 6.57: First Experiment Geodesic vs. Hierarchical (position error)

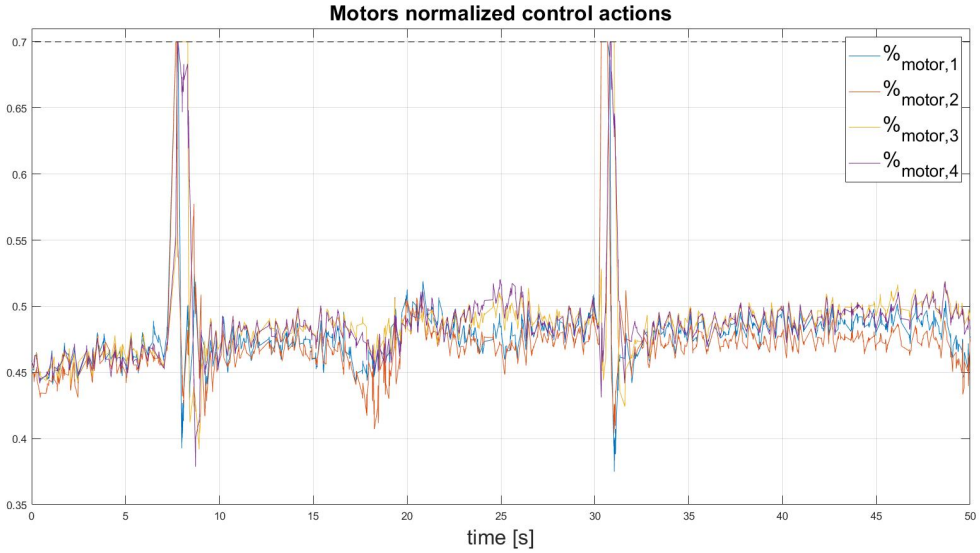

Figure 6.58: First Experiment Geodesic vs. Hierarchical (thrust propellers percentage)

The second experiment tested the UAV in the same way but setting a larger yaw angle ($135\,deg$) and the saturation has been fixed to 100 %; the plots are the following:

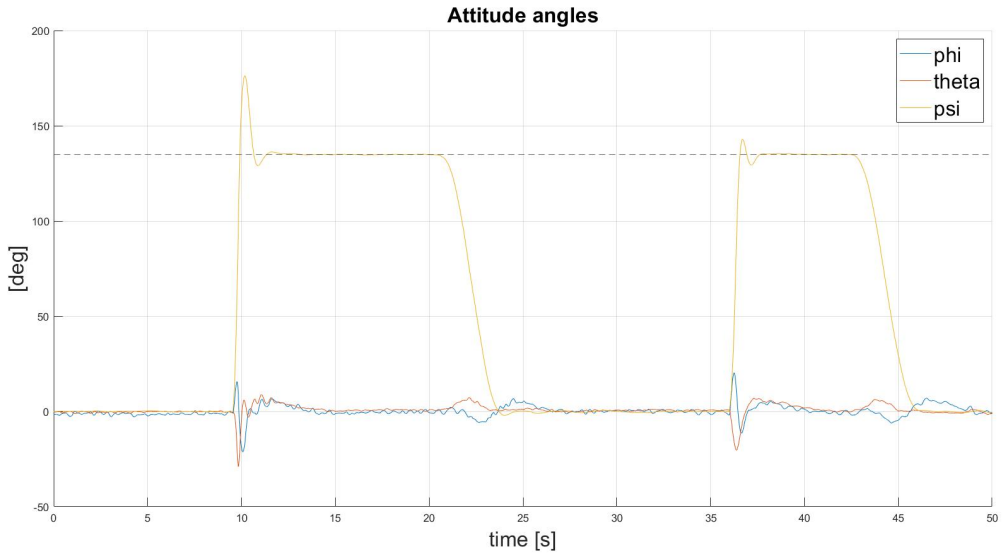

Figure 6.59: Second Experiment Geodesic vs. Hierarchical (Attitude angle response)

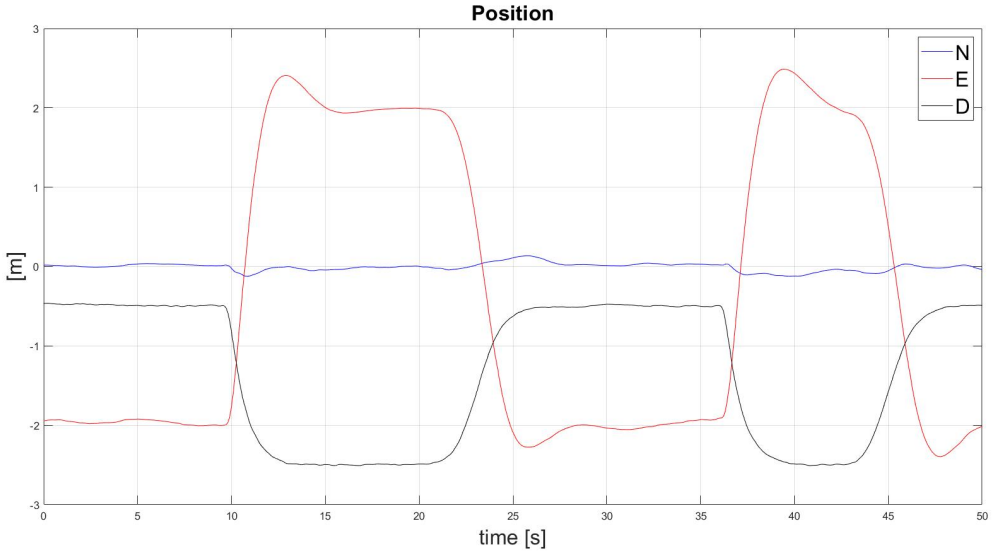

Figure 6.60: Second Experiment Geodesic vs. Hierarchical (position error)

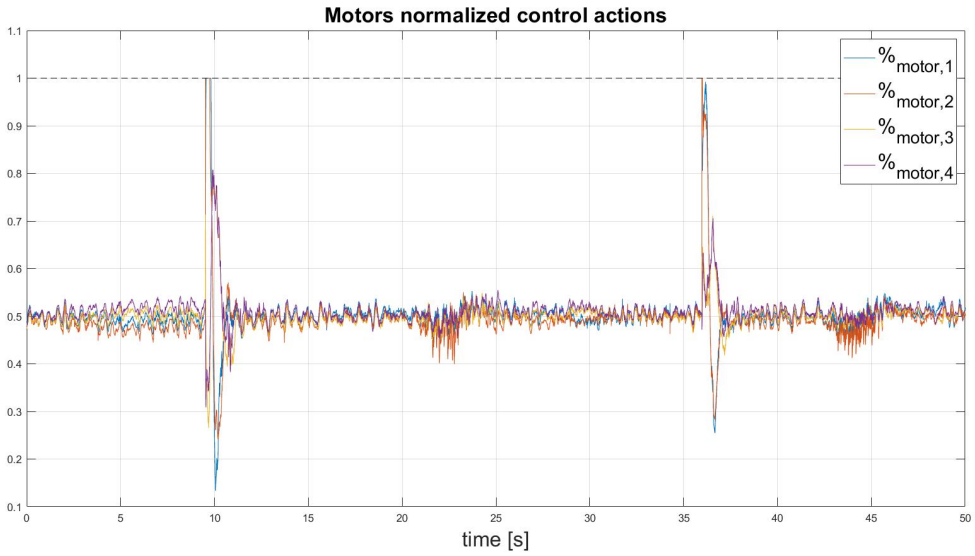

Figure 6.61: Second Experiment Geodesic vs. Hierarchical (thrust propellers percentage)

The main difference in this experiment is the overshoot of the yaw-angle response; the second architecture (geodesic law) is better in terms of response with respect to the first one (hierarchical law); this result can also be seen in the case where there a significant yaw angle maneuver.

However, during the simulation, the beneficial effect of displacement of x_{NED} is lost, this could be explained by the fact that the simulator doesn't account for some dynamics effect that in reality incomes.

Regarding the thrust percentage of propellers; for the first experiment, both the architectures are under the effect of saturation, but the geodesic law let the propellers stay in this condition for less time than the hierarchical controller. In second experiment, if the saturation is fixed to its normal value (100 %), it is possible to see that the geodesic control law is not saturated, while the hierarchical control it is, this affects directly the overshoot of the yaw-angle, which is greater when the saturation incomes.

7 | Conclusions

The thesis makes several contributions. Firstly, it involves the development of a hierarchical control law for rigid bodies, which draws inspiration from [11]. The focus is here shifted to the application of this control law to UAVs. Specifically, the hierarchical architecture is implemented within the position control architecture for trajectory tracking in vectored-thrust UAVs by including a nonlinear geometric filter, as described in [17], to address the computation of the time derivative of the desired attitude, a crucial component to improve the tracking performance.

To tune the gains of the control law, the thesis employs the H_∞ synthesis approach proposed in [12]. The nonlinear equations defining the closed-loop system of the tracking error dynamics are first linearized. This leads to a formulation that is different from the classic one used in H_∞ synthesis, where the exogenous signal is typically chosen as the reference command and the performance output as the tracking error. A proposition demonstrating the equivalence between the error and classic formulation when considering the initial conditions as the exogenous signal instead of the reference signal is derived. The method is initially introduced for the ideal rigid body problem and then adapted to address the UAV case.

In the final part of the thesis, a geodesic control law inspired by [14], is implemented within the proposed hierarchical architecture. The control law proposed in [14] has then been modified to incorporate error-dependent variable gains. This choice allows us to effectively prioritize reduced attitude stabilization when the reduced attitude error is large but to recover design performance when close to the desired trajectory, unlike the constant gains considered in [14]. Simulation and experimental results confirm the benefit of the geodesic stabilizer against a popular nonlinear stabilizer in reducing directionality windup issues associated with propellers saturation.

Bibliography

[1] Ant-x website., . https://antx.it/.

[2] Friendly arm wiki., . https://wiki.friendlyelec.com/wiki/index.php/Main_Page.

[3] Px4 autopilot user guide., . https://docs.px4.io/main/en/.

[4] D. Angeli. An almost global notion of input-to-state stability. *IEEE Transactions on Automatic Control*, 49(6):866–874, 2004. doi: 10.1109/TAC.2004.829594.

[5] D. Brescianini and R. D'Andrea. Tilt-prioritized quadrocopter attitude control. *IEEE Transactions on Control Systems Technology*, 28(2):376–387, 2018.

[6] P.-J. Bristeau, P. Martin, E. Salaün, and N. Petit. The role of propeller aerodynamics in the model of a quadrotor uav. In *2009 European control conference (ECC)*, pages 683–688. IEEE, 2009.

[7] N. A. Chaturvedi, A. K. Sanyal, and N. H. McClamroch. Rigid-body attitude control. *IEEE control systems magazine*, 31(3):30–51, 2011.

[8] C. di Fisica matematica 2. La funzione (generalizzata) delta di dirac.

[9] T. I. Fossen. Handbook of marine craft hydrodynamics and motion control.

[10] K. Gamagedara and T. Lee. Geometric adaptive controls of a quadrotor unmanned aerial vehicle with decoupled attitude dynamics. *Journal of Dynamic Systems, Measurement, and Control*, 144(3), 2022.

[11] D. Invernizzi, M. Lovera, and L. Zaccarian. Hierarchical dynamic control for robust attitude tracking. *IFAC-PapersOnLine*, 53(2):6171–6176, 2020.

[12] D. Invernizzi, S. Panza, and M. Lovera. Robust tuning of geometric attitude controllers for multirotor unmanned aerial vehicles. *Journal of Guidance, Control, and Dynamics*, 43(7):1332–1343, 2020.

[13] S. Liu. Adaptive control for high- performance trajectory tracking in quadrotor uavs.

[14] J. Markdahl, J. Hoppe, L. Wang, and X. Hu. A geodesic feedback law to decouple the full and reduced attitude. *Systems & Control Letters*, 102:32–41, 2017.

[15] F. Marzagalli, P. Ghignoni, G. Gozzini, and D. Invernizzi. Experimental validation of an anti-windup design trading off position and heading direction control performance for quadrotor uavs. *IFAC-PapersOnLine*, 55(22):117–122, 2022. ISSN 2405-8963. doi: https://doi.org/10.1016/j.ifacol.2023.03.020. URL `https://www.sciencedirect.com/science/article/pii/S2405896323002781`. 22nd IFAC Symposium on Automatic Control in Aerospace ACA 2022.

[16] B. Pratama, A. Muis, A. Subiantoro, M. Djemai, and R. B. Atitallah. Quadcopter trajectory tracking and attitude control based on euler angle limitation. In *2018 6th International Conference on Control Engineering & Information Technology (CEIT)*, pages 1–6. IEEE, 2018.

[17] S. Zhao, W. Dong, and J. A. Farrell. Quaternion-based trajectory tracking control of vtol-uavs using command filtered backstepping. In *2013 American Control Conference*, pages 1018–1023. IEEE, 2013.

A | Appendix A

A.1. Numerical Integration of non-Euclidean elements

This appendix deals with the problem of solving a numerical integration for a non-Euclidean variable, but first, it is useful to look at what happens in a normal numerical integration is used to solve Equation (2.3a), for example, using a Forward Euler:

$$R(k+1) = R(k) + R(k)S(\omega(k)) \quad \text{for k=0,1,...n-1} \tag{A.1}$$

Assume that at k=0 and impose that $R(0) \in SO(3)$, taking Equation A.1, it is possible to write $R(1)$, but there is no guarantee that $R(1) \in SO(3)$, because there is a sum with another element, in particular, there is no assurance that $R(1)$ has determinant equal to 1.

One way to solve this issue is to solve an equivalent equation in another parametrization and then return to the attitude parametrization. In particular:

- **Integration of quaternion kynematic**:

  Once the quaternion is obtained, the integration can be done by using the following equation:

$$\begin{cases} \dot{q} = \frac{1}{2}W(\omega)q = \frac{1}{2}\begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} q, & \text{(A.2a)} \\ \|q\| = 1, & \text{(A.2b)} \end{cases}$$

- **Parametrization from quaternion to rotation matrix**:

  After solving equations (A.2a) and (A.2b), the last passage is to come back to the rotation matrix framework; the rotation matrix $R(q)$ is obtained by relying on this formula:

$$R(q) = (q_4^2 - \|\mathbf{q}_v\|)\ I_3 - 2q_4\ S(\mathbf{q}_v) + 2\mathbf{q}_v\mathbf{q}_v^T, \tag{A.3}$$

$\mathbf{q}_v$ is the vectorial part of the quaternion and q_4 is the scalar part.

The simulink model of the equations (A.2a) and (A.2b), used to the equation 2.3a.

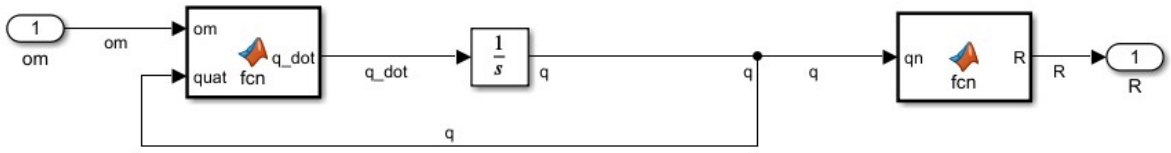

Figure A.1: Numerical Integration of Attitude equation

A.2. Almost Input-State Stability

The definition of almost Input-State stability is presented in this appendix, and how this is used in Proposition 2.3.2, refer to [4] for more detail.

Let $x \in \mathbb{R}^n$, $R \in SO(3)$ and define the class $\mathbf{K}$ as the class of function which are defined such as $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, continuous, zero at zero and strictly increasing.

Consider the cascade interconnection made by:

$$\begin{cases} \dot{x} = f(x) & \text{(A.4a)} \\ \dot{R} = Q(R) + G(R, x) & \text{(A.4b)} \end{cases}$$

where $f : \mathbb{R}^n \times R_{\geq 0} \to \mathbb{R}^n$, $Q : SO(3) \to TSO(3)$ and $Q : SO(3) \times \mathbb{R}^n \to TSO(3)$,TSO(3) refers as the tangent space to SO(3), and they are such that $f(0) = 0$, $Q(I_3) = 0$ and $G(I_3, 0) = 0$.

Suppose that the equilibrium $R = I_3$ is almost global input-state stable (aISS) for $\dot{R} = Q(R) + G(R, d)$, with respect to d, which means that:

- $R = I_3$ is locally asymptotically stable for $d = 0$.
- It exists a $\gamma \in \mathbf{K}$, such that for each essentially bounded and measurable $d : \mathbb{R} \to \mathbb{R}^n$, there exists a zero volume set $\mathbf{B}_d \subset SO(3)$ such that, for all $R(t_0) \in SO(3) \backslash \mathbf{B}_d$,

$$\lim_{t \to +\infty} sup \quad \|R(t)\|_{SO(3)} \leq \gamma(\|d\|_\infty) \tag{A.5}$$

Then, if the equilibrium $x = 0$ is GAS for Equation A.4a, then $(R, x) = (I_3, 0)$ is almost globally asymptotically stable.

A.3. Mathematical derivation for the outer-loop controller

This appendix shows the derivation of Equation (2.23). Assuming the dynamic of the attitude and desired attitude (defined respectively as (2.20) and (2.5a)), let us first compute the time derivative of R_e:

$$\dot{R}_e = \dot{R}_d^T R + R_d^T \dot{R} \tag{A.6a}$$
$$\dot{R}_e = S(\omega_d)^T R_d^T R + R_d^T R S(\omega) \tag{A.6b}$$
$$\dot{R}_e = -S(\omega_d) R_e + R_e S(\omega) \tag{A.6c}$$
$$\dot{R}_e = R_e(S(\omega) - R_e^T S(\omega_d) R_e) \tag{A.6d}$$

So, the error dynamic can be written as in Equation A.6d.

By, using the property $S(Rx) = RS(x)R^T$, $\forall x \in \mathbb{R}^3$, it is possible to write:

$$\dot{R}_e = R_e(S(\omega) - S(R_e^T \omega_d)) = R_e S(\omega - R_e^T \omega_d) \tag{A.7}$$

At this point, a fictitious angular velocity (ω_v) is needed inside the Equation A.7:

$$\dot{R}_e = R_e S(\omega - R_e^T \omega_d + \omega_v - \omega_v) \tag{A.8}$$

The fictitious angular velocity will be used in order to get rid of the term $R_e^T \omega_d$ inside the brackets and, at the same time, stabilize the error attitude matrix. So, the next step is to put Equation (2.23a) inside Equation (A.8), such as follows

$$\dot{R}_e = R_e S(\omega - R_e^T \omega_d - \omega_v + \gamma_R + R_e^T \omega_d) = \tag{A.9a}$$
$$\dot{R}_e = R_e S(\omega - \cancel{R_e^T \omega_d} - \omega_v + \gamma_R + \cancel{R_e^T \omega_d}) = \tag{A.9b}$$
$$\dot{R}_e = R_e S(\omega - \omega_v + \gamma_R) = \tag{A.9c}$$
$$\dot{R}_e = R_e S(\gamma_R - \omega_e) \tag{A.9d}$$

It is possible to note that Equation (A.9d) coincides with the expression in Equation

(2.24a).

The last step is to compute the time derivative of ω_v.

$$\dot{\omega}_v = \dot{\gamma}_R + \dot{R}_e^T \omega_d + R_e^T \dot{\omega}_d \tag{A.10}$$

By substituting the A.9d inside A.10, one obtains:

$$\dot{\omega}_v = \dot{\gamma}_R + S(\gamma_R - \omega_e)^T R_e^T \omega_d + R_e^T \dot{\omega}_d \tag{A.11a}$$

$$\dot{\omega}_v = \dot{\gamma}_R - S(\gamma_R - \omega_e) R_e^T \omega_d + R_e^T \dot{\omega}_d \tag{A.11b}$$

In the procedure, the equations are defined in a body-fixed frame, while, in Paper [14], the equations are written in an inertial reference, so $\gamma_{R,i}$ must be rotated in the right frame.

List of Figures

List of Tables

Acknowledgements

Mi è doveroso dedicare questo spazio della mia tesi a tutte le persone che mi hanno supportato nel mio percorso di crescita universitaria e professionale. Ringrazio infinitamente i miei genitori, mia sorella Sofia, la mia fidanzata Alessia e la sua famiglia e tutti i miei più cari amici, Miguel, Brambo, Davide, Momo, Antonio, Fede, Manu, Alex e Gregorio, che mi hanno sempre motivato a dare il meglio, che hanno condiviso con me gioie e dolori di questo percorso universitario. Infine, un sentito ringraziamento va al mio relatore Davide Invernizzi che mi ha seguito, con disponibilità e gentilezza, in ogni step della realizzazione dell'elaborato, fin dalla scelta dell'argomento.