



DIPARTIMENTO DI INGEGNERIA INFORMATICA, MODELLISTICA,
ELETTRONICA E SISTEMISTICA

Master Degree Robotics and Automation Engineering

Vehicles Control: Module 1

Robust, Optimal, and LPV Control of a Quad-Copter

Prof.
Alessandro Casavola
Gianfranco Gagliardi

Giuseppe Coppola
Mat. 263624

ACADEMIC YEAR 2025/26

Contents

| | | |
|-----------------|--|-----------|
| Contents | 1 | |
| 1 | Introduction | 4 |
| 2 | Quad-Copter Modelling | 5 |
| 2.1 | Structure | 5 |
| 2.2 | Quadcopter's Dynamic | 6 |
| 2.3 | State Space Representation | 8 |
| 2.4 | Simulink Modelling | 9 |
| 3 | Linearization and Analysis | 12 |
| 3.1 | Linear Approximation | 12 |
| 3.2 | Stability Analysis | 15 |
| 3.2.1 | Internal Stability | 16 |
| 3.2.2 | External Stability | 17 |
| 3.2.3 | Internal Stability Verification via LMIs | 17 |
| 3.3 | Structural Properties Analysis | 18 |
| 3.3.1 | Reachability and Controllability | 18 |
| 3.3.2 | Observability and Detectability | 19 |
| 4 | Control Design and Comparative Analysis | 20 |
| 4.1 | PID Control | 20 |
| 4.1.1 | Controller Design | 21 |
| | Altitude Controller | 21 |
| | Attitude Controller | 22 |
| | Position Controller | 22 |
| 4.1.2 | Results and Discussion | 24 |
| 4.2 | Optimal \mathcal{H}_∞ , \mathcal{H}_2 and \mathcal{L}_1 Control | 26 |
| 4.2.1 | External Disturbance Description | 26 |

| | |
|--|-----------|
| Mild Gust - Scenario 1 | 27 |
| Intense Gust - Scenario 2 | 27 |
| 4.2.2 State-Space Representation and Formulation | 28 |
| 4.2.3 \mathcal{H}_∞ Optimal Control | 29 |
| Results and Discussion | 30 |
| 4.2.4 \mathcal{H}_2 Optimal Control | 32 |
| Results and Discussion | 33 |
| 4.2.5 \mathcal{L}_1 Optimal Control | 35 |
| Results and Discussion | 35 |
| 4.2.6 Comparison of Controllers under Nominal Conditions | 37 |
| 4.2.7 Comparison of Controllers under Disturbance Conditions - Scenario 1 | 39 |
| \mathcal{H}_∞ Control - Scenario 1 | 39 |
| \mathcal{H}_2 Control - Scenario 1 | 40 |
| \mathcal{L}_1 Control - Scenario 1 | 41 |
| Controllers Comparison and Discussion - Scenario 1 | 42 |
| 4.2.8 Comparison of Controllers under Disturbance Conditions - Scenario 2 | 43 |
| \mathcal{H}_∞ Control - Scenario 2 | 44 |
| \mathcal{H}_2 Control - Scenario 2 | 44 |
| \mathcal{L}_1 Control - Scenario 2 | 45 |
| Controllers Comparison and Discussion - Scenario 2 | 46 |
| 5 Multi-Objective Control | 48 |
| 5.1 $\mathcal{H}_\infty/\mathcal{H}_2$ Control | 48 |
| 5.1.1 Scalarization Approach | 48 |
| Results and Discussion - Scenario 1 | 49 |
| Results and Discussion - Scenario 2 | 50 |
| 5.1.2 ϵ -Constrained Approach | 51 |
| Results and Discussion - Scenario 1 | 52 |
| Results and Discussion - Scenario 2 | 53 |
| 5.1.3 Controllers Comparison and Discussion | 54 |
| 5.2 \mathcal{H}_∞ Control with r -stability Constraints | 56 |
| Results and Discussion - Scenario 1 | 57 |
| Results and Discussion - Scenario 2 | 58 |
| 6 LPV Control and Gain Scheduling | 60 |
| 6.1 Implementation | 62 |

| | | |
|-------|---|-----------|
| 6.2 | Robust Control | 63 |
| 6.2.1 | \mathcal{H}_∞ Robust Control | 63 |
| 6.2.2 | \mathcal{H}_2 Robust Control | 64 |
| 6.2.3 | \mathcal{L}_1 Robust Control | 64 |
| 6.3 | Results and Discussion | 65 |
| 6.3.1 | \mathcal{H}_∞ Control - Scenario 1 | 65 |
| 6.3.2 | \mathcal{H}_∞ Control - Scenario 2 | 67 |
| 6.3.3 | \mathcal{H}_2 Control - Scenario 1 | 68 |
| 6.3.4 | \mathcal{H}_2 Control - Scenario 2 | 70 |
| 6.3.5 | \mathcal{L}_1 Control - Scenario 1 | 71 |
| 6.3.6 | \mathcal{L}_1 Control - Scenario 2 | 73 |
| 7 | Conclusions | 75 |

Chapter 1

Introduction

This report investigates the design and comparison of different control strategies for quadrotors, aiming to ensure stable flight even in the presence of external disturbances and uncertainties in system parameters. The work is organized around three main aspects:

- **Modeling:** Developing a dynamic model of the vehicle and linearizing it to enable stability analysis.
- **Controller Design:** Evaluating both classical PID control and advanced methods (\mathcal{H}_∞ , \mathcal{H}_2 , \mathcal{L}_1) to enhance disturbance rejection and overall performance.
- **Handling Uncertainties:** Implementing LPV controllers and Gain Scheduling to maintain consistent performance under varying physical conditions, such as changes in mass.

The ultimate goal is to identify the control strategy that achieves the best balance between dynamic accuracy, robustness, and actuator energy efficiency.

Chapter 2

Quad-Copter Modelling

The Quad-Copter is equipped with four arms and rotors. The most common arrangements of quadcopter are simmetrical cross (\times) or plus (+).

The positions and rotations of the quadcopter are represented by six variables of six degrees of freedom.

In this report, a cross (\times) quadcopter mathematical model is derived.

2.1 Structure

The rotors of the quadcopter are running in different pairs directions to eliminate the anti-torque effects and produce translational and rotational movement.

As in the Figure (2.1) rotor 1 and 3 are running in counter-clockwise direction, while rotor 2 and 4 are running in clockwise direction.

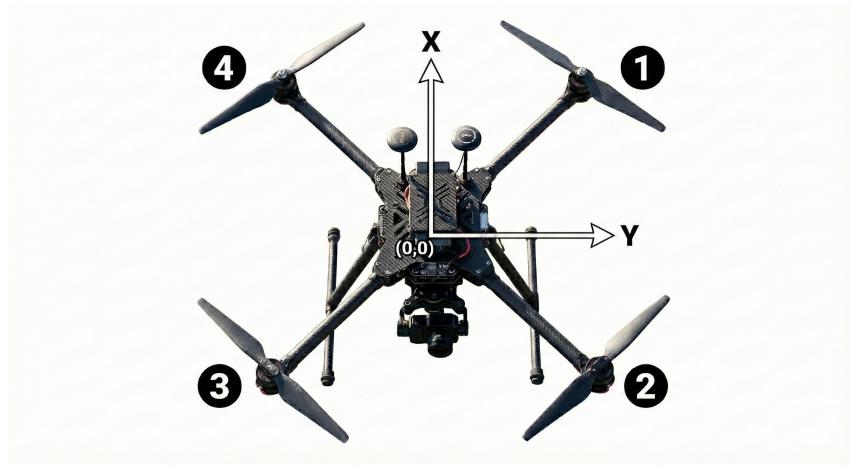


Figure 2.1: Structure of cross quadcopter

The motions and the movements of quadcopter are produced by the lifting force generated by the rotating rotors.

In order to simplify the mathematical model, several assumptions are made as follows:

- The quadcopter is assumed to be a rigid body.
- The structure of assumed to be asymmetrical with respect to x, y axis. Therefore, the degree-of-freedom of plus(+) structure can be used for cross (\times) structure.
- The rotors are considered as rigid, no blade flapping occurs.
- The rotors works under the same conditions at any time, in which the thrust coefficient and reaction are the same.

A motion is initiated by producing difference in the lift by manipulating the rotor velocity. The motions of quadcopter are:

1. **Hovering**: same speed for all rotors.
2. **Rolling**: Rotors 1&2 or Rotors 3&4.
3. **Pitching**: Rotors 1&4 or Rotors 2&3.
4. **Yawing**: Rotors 1&3 or Rotors 2&4.

2.2 Quadcopter's Dynamic

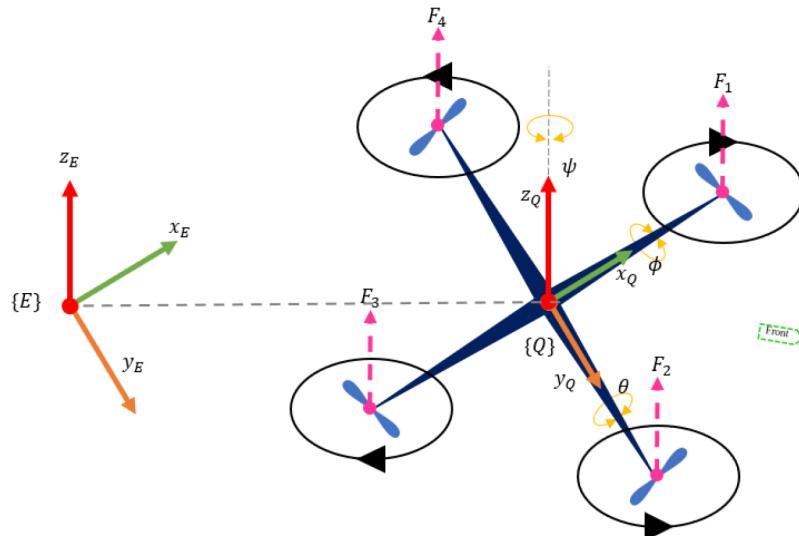


Figure 2.2: Frame of body-fixed and earth-fixed of quadcopter

The absolute position of the quadcopter are represented by (x, y, z) and the orientation are, respectively, described by **roll** (ϕ), **pitch** θ and **yaw** (ψ).

Based on Figure (2.2), the model is divided into position relative to inertial frame ξ and quadcopter attitude η :

$$\xi = [x \ y \ z]^T \in \mathbb{R}^3$$

$$\eta = [\phi \ \theta \ \psi]^T \in \mathbb{R}^3$$

The elementary rotations about x , y and z axes are defined using **Euler Angles**:

$$R_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \quad R_\theta = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad R_\psi = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

From this matrices, the rotation matrix of the body frame relative to the inertial frame is obtained. Let, for simplicity in the definition, $\cos(x) = cx$ and $\sin(x) = sx$:

$$R_T = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\psi s\phi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi s\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \quad (2.1)$$

Newton-Euler's is used to derive the dynamic of quadcopter as the model is assumed to be a rigid body:

$$m\ddot{\xi} = m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = -mgE_Z + u_1 R_T E_Z \quad (2.2)$$

where m is the **mass of the quadcopter** and g is the **gravitational coefficient** and it's defined in negative state as the direction of z -axis is upward. The z -axis vector matrix $E_Z = [0 \ 0 \ 1]^T$ whereas u_1 is the **total thrust force** generated by four rotors and satisfied as

$$u_1 = \sum_{i=0}^4 F_i \quad (2.3)$$

The final equation is:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \frac{u_1}{m} \begin{bmatrix} c\phi s\theta c\psi + s\phi s\psi \\ c\phi s\theta s\psi - s\phi c\psi \\ c\phi c\theta \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (2.4)$$

The **rotational subsystem** of the quadcopter dynamic is

$$I\ddot{\eta} = -\dot{\eta} \times I\dot{\eta} - \mathcal{J}_r (\dot{\eta} \times E_Z) \Omega_d + [u_2 \ u_3 \ u_4]^T \quad (2.5)$$

$$\begin{bmatrix} I_{xx}\ddot{\phi} \\ I_{yy}\ddot{\theta} \\ I_{zz}\ddot{\psi} \end{bmatrix} = \begin{bmatrix} (I_{yy} - I_{zz})\dot{\psi}\dot{\theta} - (\mathcal{J}_r\Omega_d)\dot{\theta} + lu_2 \\ (I_{zz} - I_{xx})\dot{\psi}\dot{\phi} - (\mathcal{J}_r\Omega_d)\dot{\phi} + lu_3 \\ (I_{xx} - I_{yy})\dot{\theta}\dot{\phi} + u_4 \end{bmatrix} \quad (2.6)$$

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ -b & b & b & -b \\ -b & b & -b & b \\ d & d & -d & -d \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \quad (2.7)$$

where l is the **lateral moment arm**, b is the **thrust coefficient**, d is the **drag coefficient**, I is defined as $\text{diag}(I_{xx}, I_{yy}, I_{zz})$, the **rotor inertia**, $\mathcal{J}_r = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4$ and the **rotor total speed** Ω_d .

Then , u_2 , u_3 and u_4 are the **total torque** related to quadcopter.

2.3 State Space Representation

Define the state vector $x \in \mathbb{R}^{12}$ corresponding to the 12 degree of freedom (position, orientation and their derivatives).

$$x = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}]^T = [x, \dot{x}, y, \dot{y}, z, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\phi}, \dot{\theta}, \dot{\psi}, \dot{\psi}]^T$$

Appllying the derivative, the state space representation (2.8) is obtained.

$$\left\{ \begin{array}{l} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{u_1}{m} (\cos(x_7) \sin(x_9) \cos(x_{11}) + \sin(x_7) \sin(x_{11})) \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = \frac{u_1}{m} (\cos(x_7) \sin(x_9) \sin(x_{11}) - \sin(x_7) \cos(x_{11})) \\ \dot{x}_5 = x_6 \\ \dot{x}_6 = \frac{u_1}{m} (\cos(x_7) \cos(x_9)) - g \\ \dot{x}_7 = x_8 \\ \dot{x}_8 = \frac{I_{yy}-I_{zz}}{I_{xx}} x_{10} x_{12} - \frac{\mathcal{J}_r \Omega_d}{I_{xx}} x_{10} + \frac{l}{I_{xx}} u_2 \\ \dot{x}_9 = x_{10} \\ \dot{x}_{10} = \frac{I_{zz}-I_{xx}}{I_{yy}} x_8 x_{12} + \frac{\mathcal{J}_r \Omega_d}{I_{yy}} x_8 + \frac{l}{I_{yy}} u_3 \\ \dot{x}_{11} = x_{12} \\ \dot{x}_{12} = \frac{I_{xx}-I_{yy}}{I_{zz}} x_8 x_{10} + \frac{1}{I_{xx}} u_4 \end{array} \right. \quad (2.8)$$

2.4 Simulink Modelling

Based on the model description, in this section, will be shown the simulation blocks of the dynamics of the quadcopter. The main block is built like:

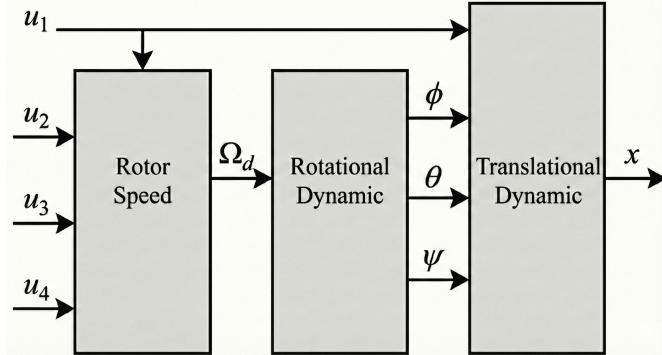


Figure 2.3: Blocks of the dynamic of quadcopter

Based on the equation (2.7), the simulation blocks of **Rotor Speed** is the following one:

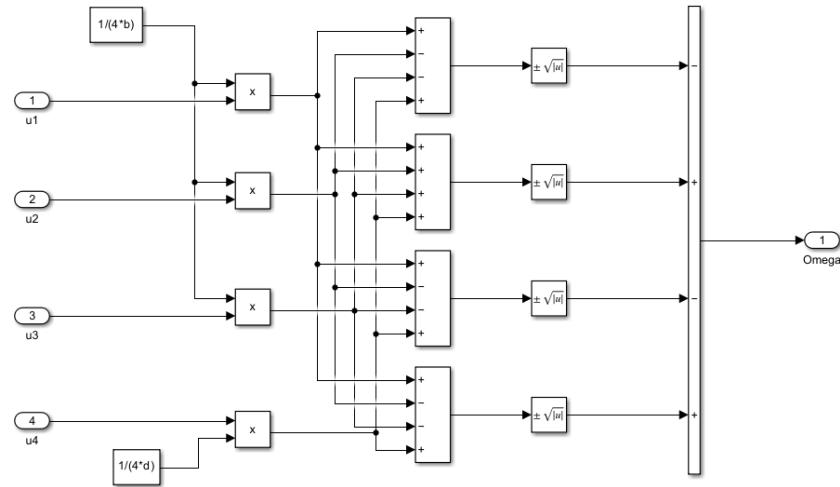


Figure 2.4: Simulation blocks of Rotor Speed

Based on the equation (2.6), the simulation blocks of **Rotational Dynamic** is the following one:

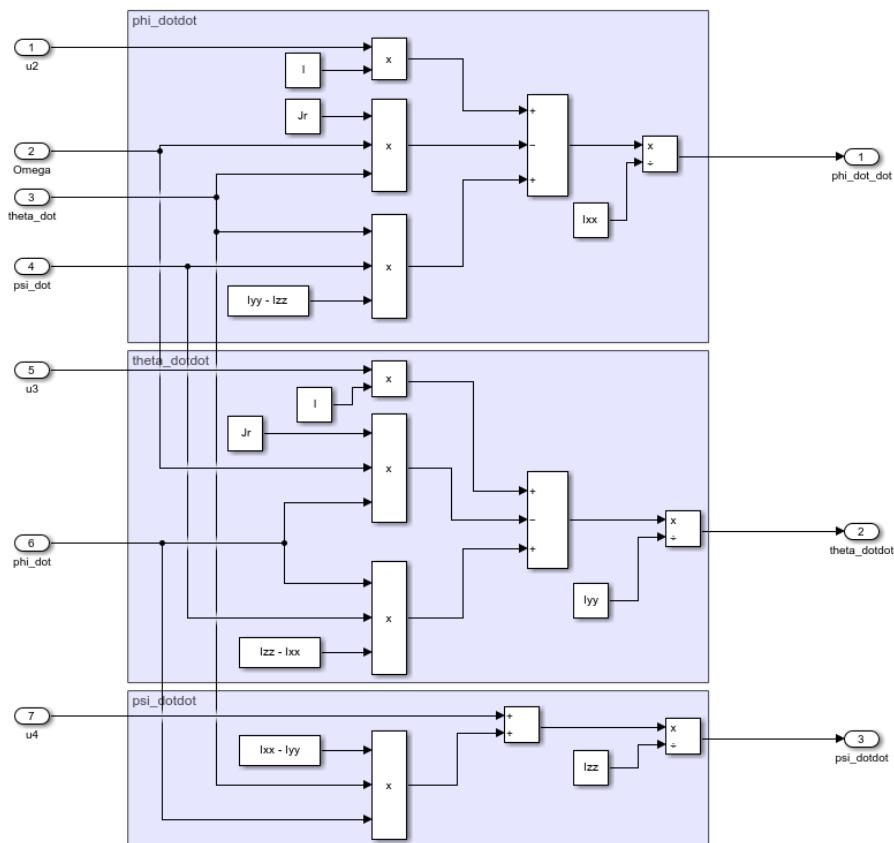


Figure 2.5: Simulation blocks of Rotational Dynamic

Based on the equation (2.6), the simulation blocks of **Translational Dynamic** is the following one:

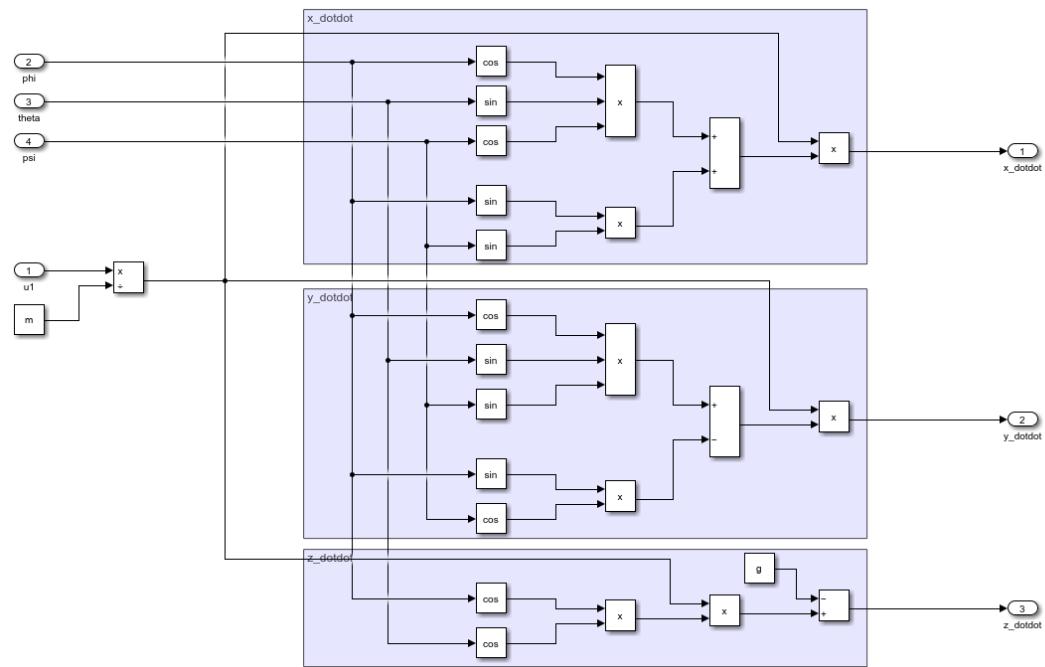


Figure 2.6: Simulation blocks of Translational Dynamic

Chapter 3

Linearization and Analysis

In this section, a linear approximation of the model will be defined and several properties will be investigated. The analysis will be carried out in detail by studying Lyapunov stability and examining the structural properties.

3.1 Linear Approximation

As a first step, to find the **equilibrium point** of the system, we must determine the state vector \mathbf{x}_{eq} and the control input vector \mathbf{u}_{eq} such that the system remains at rest or in steady state.

Mathematically, it is required to impose $\dot{\mathbf{x}} = \mathbf{0}$.

Examining the kinematic equations (where the derivative is simply a velocity):

$$\dot{x}_1 = x_2 = 0$$

$$\dot{x}_3 = x_4 = 0$$

$$\dot{x}_5 = x_6 = 0$$

$$\dot{x}_7 = x_8 = 0$$

$$\dot{x}_9 = x_{10} = 0$$

$$\dot{x}_{11} = x_{12} = 0$$

At equilibrium, all linear and angular velocities must be zero.

Next, by substituting into the rotational dynamics equations and set to zero:

$$\dot{x}_8 = 0 \quad \rightarrow \quad u_2 = 0$$

$$\dot{x}_{10} = 0 \quad \rightarrow \quad u_3 = 0$$

$$\dot{x}_{12} = 0 \quad \rightarrow \quad u_4 = 0$$

To maintain equilibrium, the net moment for roll, pitch and yaw must be zero.

Finally, by substituting the zero velocities into translational dynamics equations and set to zero:

$$\dot{x}_6 = 0 \quad \rightarrow \quad u_1 = mg$$

For a hovering equilibrium, It can be assumed that the quadcopter is level, meaning the roll (x_7) and pitch (x_9) angles are zero.

$$x_{eq} = \begin{bmatrix} \bar{x} & 0 & \bar{y} & 0 & \bar{z} & 0 & 0 & 0 & 0 & 0 & \bar{\psi} & 0 \end{bmatrix}^T$$

$$u_{eq} = \begin{bmatrix} mg & 0 & 0 & 0 \end{bmatrix}^T$$

where \bar{x} , \bar{y} , \bar{z} , and $\bar{\psi}$ can take any constant value.

For the sake of simplicity, they are assumed to be zero; consequently, the equilibrium point becomes

$$x_{eq} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \quad (3.1)$$

$$u_{eq} = \begin{bmatrix} mg & 0 & 0 & 0 \end{bmatrix}^T \quad (3.2)$$

The quadcopter dynamics derived in the report are *non-linear*, represented generally as $\dot{x} = f(x, u)$, where $x \in \mathbb{R}^{12}$ is the state vector and $u \in \mathbb{R}^4$ is the input vector.

To apply linear control techniques or analyze stability, we must approximate this non linear system as a linear system around specific operating point, known as equilibrium point.

Using Taylor series expansion truncated to the first order:

$$\dot{x} \approx f(x_{eq}, u_{eq}) + \frac{\partial f}{\partial x} \Big|_{x_{eq}, u_{eq}} (x - x_{eq}) + \frac{\partial f}{\partial u} \Big|_{x_{eq}, u_{eq}} (u - u_{eq})$$

Defining $\tilde{x} = x - x_{eq}$ and $\tilde{u} = u - u_{eq}$, it will be obtained the linear state-space form:

$$\dot{\tilde{x}} = A\tilde{x} + B\tilde{u} \quad (3.3)$$

where:

$$A = \frac{\partial f}{\partial x} \Big|_{x_{eq}, u_{eq}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & g & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -g & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{n \times n} = \mathbb{R}^{12 \times 12}$$

$$B = \frac{\partial f}{\partial u} \Big|_{x_{eq}, u_{eq}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{l}{I_{xx}} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{l}{I_{yy}} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix} \in \mathbb{R}^{n \times m} = \mathbb{R}^{12 \times 4}$$

Given the parameters used for the quadcopter model listed in Table (3.1), the obtained matrices will be:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9.81 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -9.81 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.4)$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1.25 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 19.145 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 19.145 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 35.286 \end{bmatrix} \quad (3.5)$$

| Parameter | Symbol | Value |
|----------------------------|-----------------|---|
| Quadrotor mass | m | 0.800 [kg] |
| Lateral motor arm | l | 0.300 [m] |
| Thrust coefficient | b | $1.9232 \cdot 10^{-5}$ [N·m ²] |
| Drag coefficient | d | $4.003 \cdot 10^{-7}$ [N·m·s ²] |
| Rolling moment of inertia | I_{xx} | 0.01567 [kg·m ²] |
| Pitching moment of inertia | I_{yy} | 0.01567 [kg·m ²] |
| Yawing moment of inertia | I_{zz} | 0.02834 [kg·m ²] |
| Gravity | g | 9.81 [g/s ²] |
| Rotor Inertia | \mathcal{J}_r | $6.01 \cdot 10^{-5}$ [kg·m ²] |

Table 3.1: Quadcopter Parameters

3.2 Stability Analysis

This section is devoted to the analysis of the internal and external stability of the system.

Internal stability describes the evolution of the system states when no external inputs are applied, and it determines whether the equilibrium point is asymptotically stable.

External stability focuses on the input–output behavior of the system and assesses its ability to produce bounded outputs in response to bounded inputs.

The study is conducted using the linearized state-space representation of the system.

3.2.1 Internal Stability

The internal stability of the system refers to the behavior of the state variables in the absence of external inputs. In other words, it characterizes whether the system's states remain bounded and converge to the equilibrium point over time.

For the linearized system represented in state-space form:

$$\dot{\tilde{x}}(t) = A\tilde{x}(t)$$

the system is **internally stable** if the solution $\tilde{x}(t)$ tends to zero as $t \rightarrow \infty$.

The internal stability analysis is carried out by examining the eigenvalues of the state matrix A .

In this case, **all eigenvalues of A are zero**, indicating that the system is marginally stable.

This means that, in the linear approximation, the state variables do **not diverge**, but they also do not asymptotically converge to the equilibrium point.

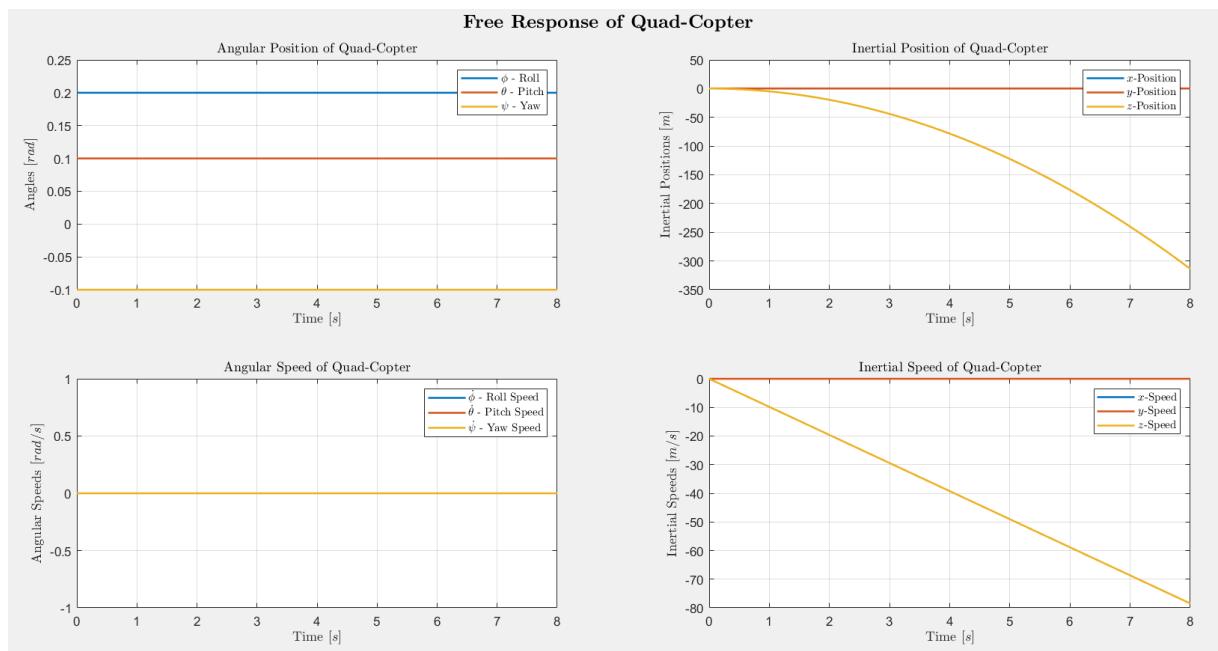


Figure 3.1: Free Evolution of the Quadcopter

The **free response** of the system confirms that:

- The angular positions (ϕ, θ, ψ) remain constant at their initial values , while the angular speeds remain at zero.
- The vertical dynamics illustrate the instability caused by the chain of integrators. The z -velocity decreases linearly, which in turn causes the z -position to diverge

parabolically. This unbounded behavior demonstrates that while the system does not explode exponentially, it drifts away from the equilibrium point indefinitely in the absence of control.

3.2.2 External Stability

External stability, also known as **BIBO (Bounded-Input Bounded-Output)** stability, concerns the relationship between the system inputs and outputs. A system is externally stable if every bounded input produces a bounded output.

For the linearized state space system:

$$\begin{cases} \dot{\tilde{x}}(t) = A\tilde{x}(t) + B\tilde{u}(t) \\ y(t) = C\tilde{x}(t) + D\tilde{u}(t) \end{cases}$$

the transfer function can be expressed as:

$$G(s) = C(sI - A)^{-1}B + D$$

The external stability analysis is performed by examining the poles of $G(s)$, which correspond to the eigenvalues of A in the SISO case.

In this case of the quadcopter, since all poles have zero real parts, the system is **marginally stable**.

3.2.3 Internal Stability Verification via LMIs

The stability of a linear time-invariant system can be analyzed using **Lyapunov's method**. A system $\dot{x} = Ax$ is *asymptotically stable* if there exists a positive definite matrix $P = P^T > 0$ such that the Lyapunov inequality is satisfied:

$$A^T P + PA < 0$$

This inequality guarantees that the Lyapunov function $V(x) = x^T Px$ decreases along system trajectories, proving that the equilibrium point is **asymptotically stable**.

Importantly, this condition can be reformulated as a **Linear Matrix Inequality** (LMI), which allows systematic numerical verification of stability for high-dimensional systems. The LMI formulation provides a computationally efficient approach to determine whether a positive definite P exists, even when analytical methods are not feasible.

The following MATLAB function implements this LMI-based stability check for a given state-space system:

```

1 P = sdpvar(n);
2 C1 = P >= 1e-7;
3 C2 = (A'*P + P*A <= -1e-7*eye(n));
4 con = C1 + C2;
5 opts = sdpsettings;
6 opts.solver='sedumi';
7 diag = optimize(con, [], opts);
8
9 if diag.problem == 0
10    disp('Stable System');
11    Psol = double(P);
12 else
13    disp('Unstable System');
14    Psol = [];
15 end

```

The result of the LMI-based check indicates that the system is **unstable**.

3.3 Structural Properties Analysis

The structural properties of a dynamical system provide fundamental insights into its inherent behavior, independent of specific numerical parameter values.

These properties include concepts such as **controllability**, **observability**, **reachability**, and **detectability**, which describe the system's ability to be controlled, monitored, and stabilized through its inputs and outputs.

Analyzing these properties is essential for understanding the capabilities and limitations of the system before designing controllers or observers. In this section, we systematically examine the structural characteristics of the system, highlighting how they influence its controllability, observability, and overall performance.

3.3.1 Reachability and Controllability

By definition, a state x_f is **reachable** from the initial state x_0 if there exist an input $u(t)$ defined over a finite time interval $[0, T]$ that drives the system from x_0 to x_f .

A system is said to be **controllable** if every state is reachable from the origin.

Reachability and controllability are closely related concepts: reachability focuses on individual states, while controllability considers the entire state space.

Moreover, for LTI systems, the two properties coincide.

To verify this properties, it is useful to analyze the controllability matrix \mathcal{R} defined as:

$$\mathcal{R} = \begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix} \in \mathbb{R}^{n \times nm}$$

The system is controllable if and only if \mathcal{R} has **full rank**. Controllability ensures that it is possible to drive the system from any initial state to any desired final state within finite time using appropriate inputs.

In the case of the quadcopter, since the controllability matrix has full rank, the system is **fully controllable**.

3.3.2 Observability and Detectability

Observability is a fundamental structural property that characterizes the ability to reconstruct the complete state of a system from its outputs.

A system is said to be **observable** if, for any initial state x_0 the output $y(t)$ over a finite time interval $[0, T]$ uniquely determine x_0 .

Detectability is a weaker property than observability. A system is detectable if all unobservable modes are stable, meaning that any state component that cannot be reconstructed from the outputs does not cause instability in the system.

For LTI systems, observability and detectability are structural properties that can be checked through the rank of the **observability matrix** \mathcal{O} defined as:

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \in \mathbb{R}^{pn \times n}$$

The system is observable if and only if \mathcal{O} has **full rank**.

In the case of the quadcopter, since the observability matrix has full rank, the system is **fully observable**.

Chapter 4

Control Design and Comparative Analysis

This chapter focuses on the design, implementation, and performance evaluation of several control strategies applied to the system under study. Both classical controllers, such as the PID, and modern robust and optimal controllers, including \mathcal{H}_∞ , \mathcal{H}_2 and \mathcal{L}_1 , are considered.

Each controller is developed and analyzed in terms of stability, tracking performance, and robustness. A comparative study is then performed to highlight the strengths and limitations of each method under different operating conditions.

4.1 PID Control

The PID controller is applied mainly in process control and industry, where it can be claimed to be one of the most favourable controllers.

The advantage of PID is that it is easy to implement and tune.

For the quadcopter, the PID controller is used to adjust the speed of all the rotors to achieve the desired orientation of the quadcopter. The PID controller worked as the corrector of the difference between the desired setpoint and measured point.

The main idea of the PID controller is:

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (4.1)$$

The error of the system is defined as $e(t) = x_d(t) - x(t)$ where $x_d(t)$ is the reference and $x(t)$ is the measured value.

4.1.1 Controller Design

The PID controller for the quadrotor is implemented in a modular block-wise approach. First, the altitude and attitude control loops are designed, followed by the position control loop.

The block diagram of the system with the PID controller is shown below.

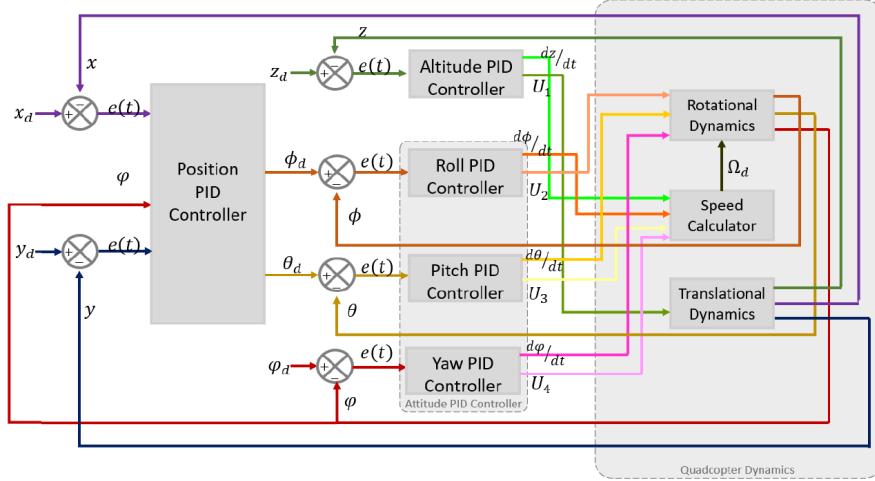


Figure 4.1: Block diagram of quadcopter with PID controller

Altitude Controller

The altitude dynamic is described in 2.4. To initiate a movement of quadcopter along z -axis, thrust force must be generated by all four rotors. The input u_1 is derived as:

$$u_{1PID} = K_p^z e(t)_z + K_i^z \int_0^t e(t)_z d(t) + K_d^z \dot{e}(t)_z \quad (4.2)$$

where

$$e(t)_z = z_d(t) - z(t)$$

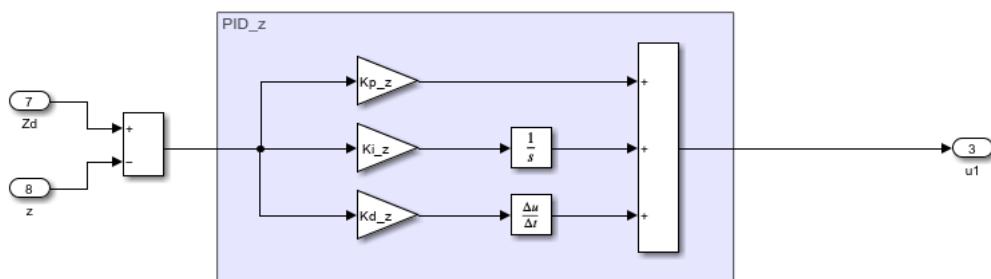


Figure 4.2: Simulink blocks of PID controller for z

Attitude Controller

The inputs u_2 , u_3 and u_4 are the control inputs of roll (ϕ), pitch (θ) and yaw (ψ), respectively.

$$u_{2,3,4PID} = K_p^a e(t)_a + K_i^a \int_0^t e(t)_a dt + K_d^a \dot{e}(t)_a \quad (4.3)$$

where

$$e(t)_a = a_d(t) - a(t)$$

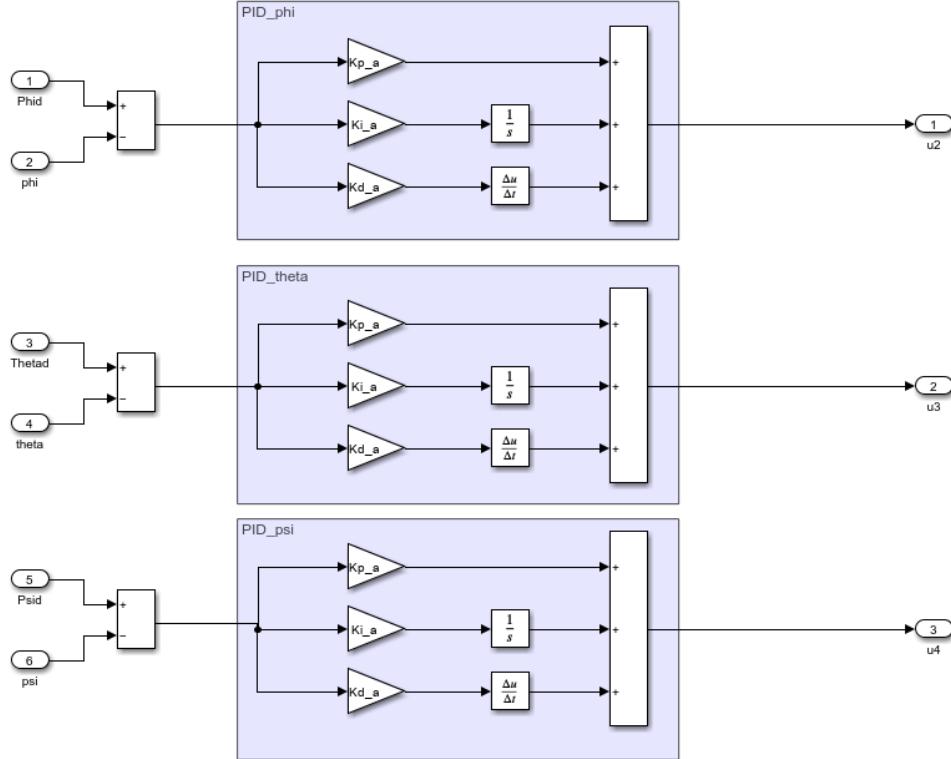


Figure 4.3: Simulink blocks of PID controller for attitude

Position Controller

To realize trajectory tracking of quadcopter, a manipulation of x and y positions need to be made with the roll and pitch angles as x and y positions cannot be controlled with u_1 . Quadcopter operates at hovering position which make the angle of roll and pitch are small-angle values. Therefore, the dynamics equations of the x and y positions are simplified based on the small angle assumptions ($s\phi_d \approx \phi_d$, $s\theta_d \approx \theta_d$, $c\phi_d = c\theta_d = 1$).

To derive equation of PID controller for positions:

$$\ddot{x} = \frac{u_1(\sin(\psi)\phi_d + \cos(\psi)\theta_d)}{m}$$

$$\ddot{y} = \frac{u_1(\sin(\psi)\theta_d - \cos(\psi)\phi_d)}{m}$$

In compact form:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \frac{u_1}{m} \begin{bmatrix} \sin(\psi) & \cos(\psi) \\ -\cos(\psi) & \sin(\psi) \end{bmatrix} \begin{bmatrix} \phi_d \\ \theta_d \end{bmatrix}$$

Then,

$$\phi_d = (u_x \sin(\psi_d) - u_y \cos(\psi_d)) \quad (4.4)$$

$$\theta_d = (u_x \cos(\psi_d) + u_y \sin(\psi_d)) \quad (4.5)$$

where ψ_d is the desired roll, and θ_d is the desired pitch. u_x and u_y are inputs control signal which both are then designed to be used in PID.

$$u_{x,y_{PID}} = K_p^p e(t)_p + K_i^p \int_0^t e(t)_p dt + K_d^p \dot{e}(t)_p \quad (4.6)$$

with the error

$$e(t)_p = p_d(t) - p(t)$$

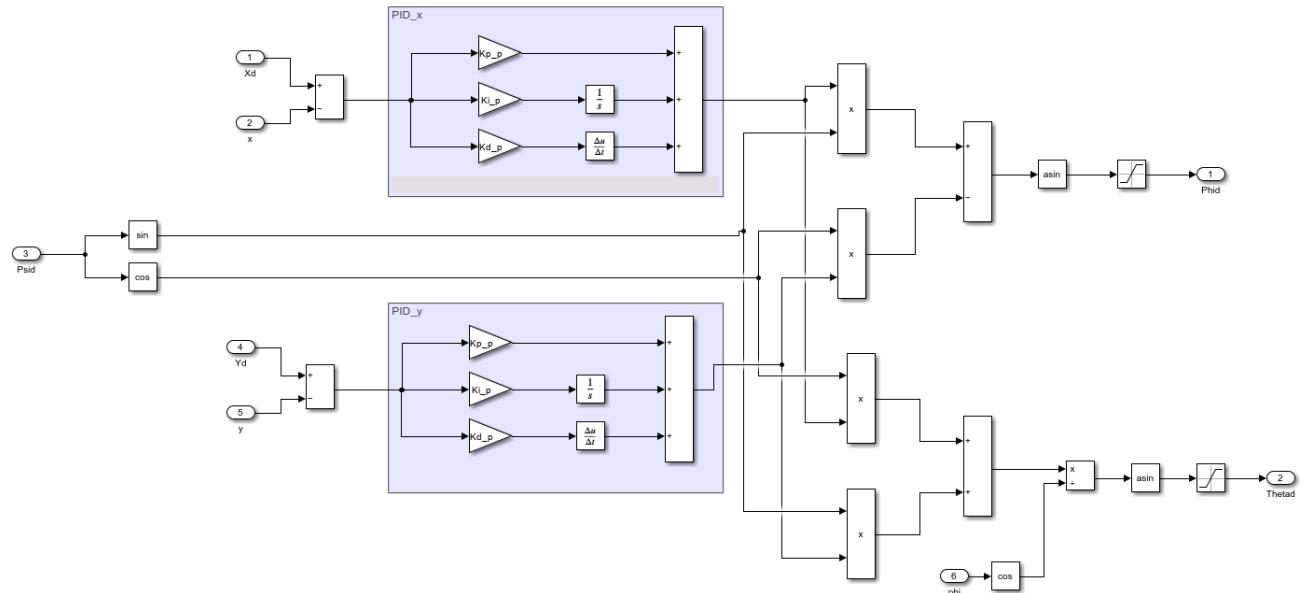


Figure 4.4: Simulink blocks of PID controller for x and y positions

Based on the illustration of PID controller of quadcopter in Figure (4.1), the complete model using PID controller is simulated as:

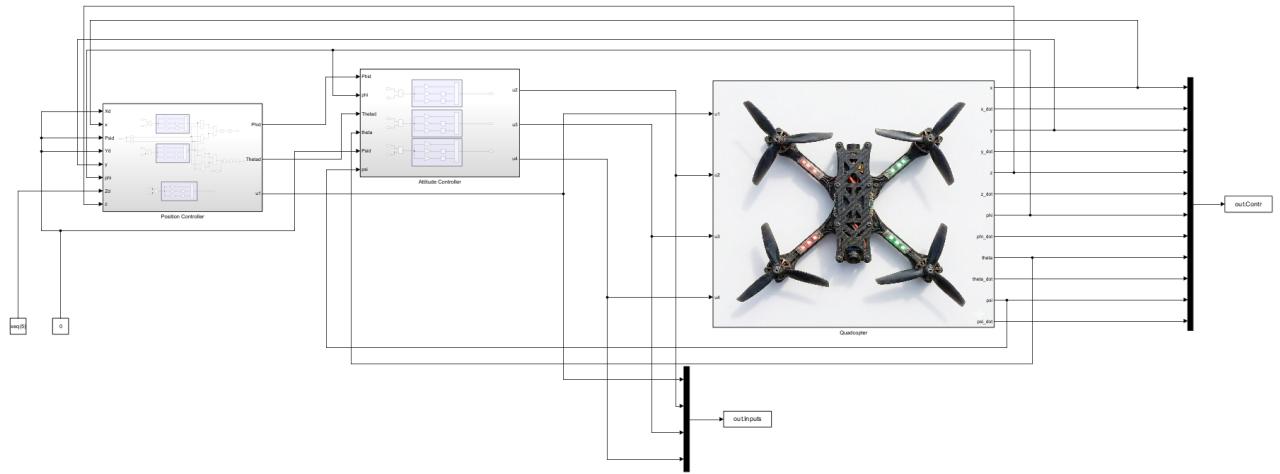


Figure 4.5: Simulink blocks of the Quadcopter with PID Controller

4.1.2 Results and Discussion

As a first step, the table below listed the gains values used.

| Gains | Position (x,y) | Altitude (z) | Attitude (ϕ,θ,ψ) |
|-------|--------------------|------------------|---------------------------------|
| K_p | 2.0 | 35.0 | 13.0 |
| K_i | 0.3 | 14.5 | 0.5 |
| K_d | 4.0 | 9.0 | 1.5 |

Table 4.1: PID gains values

Given the initial conditions

$$x_0 = \begin{bmatrix} 0.2 & 0 & 0.3 & 0.15 & 0.2 & 0 & 0.1 & 0 & -0.1 & 0 \end{bmatrix}^T$$

and a simulation time $T_{sim} = 8s$.

Figure (4.6) and (4.7) presents the results obtained from the simulation.

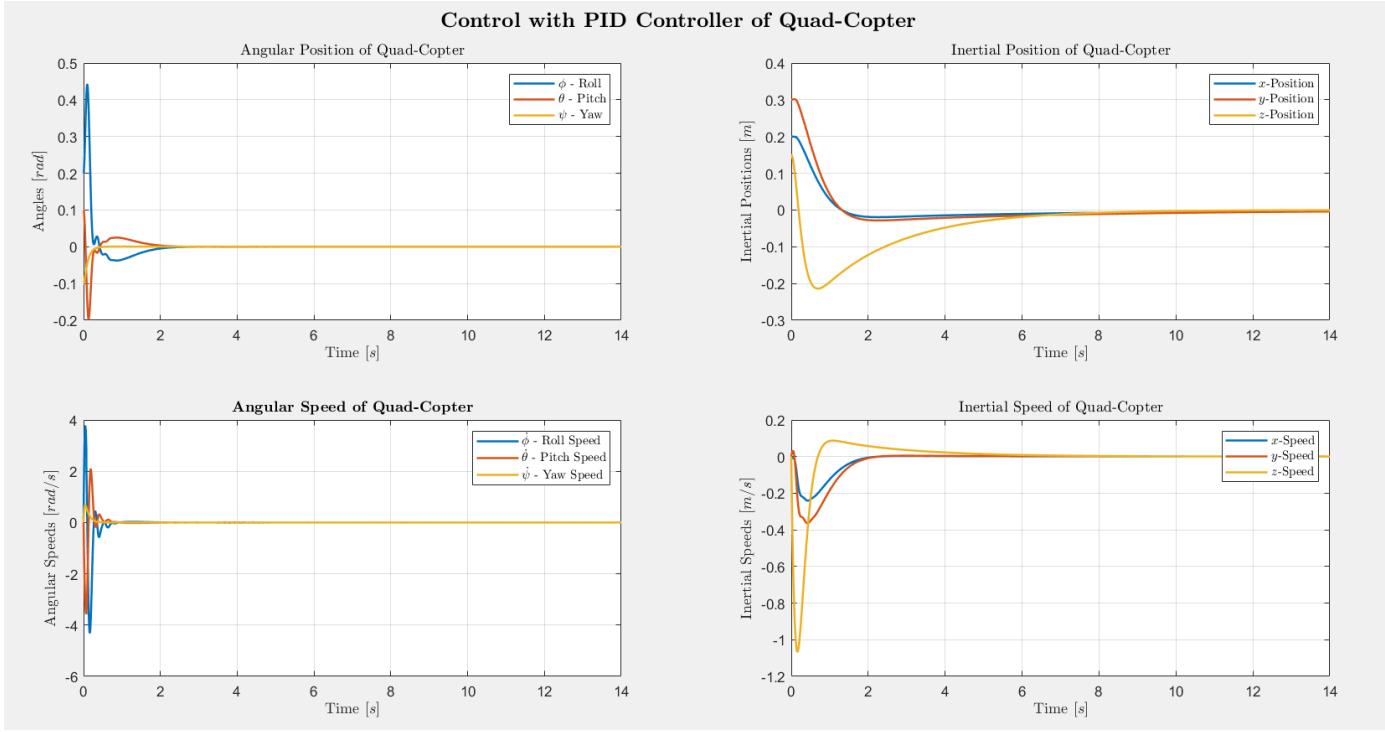


Figure 4.6: States evolution of the quadcopter with PID controller

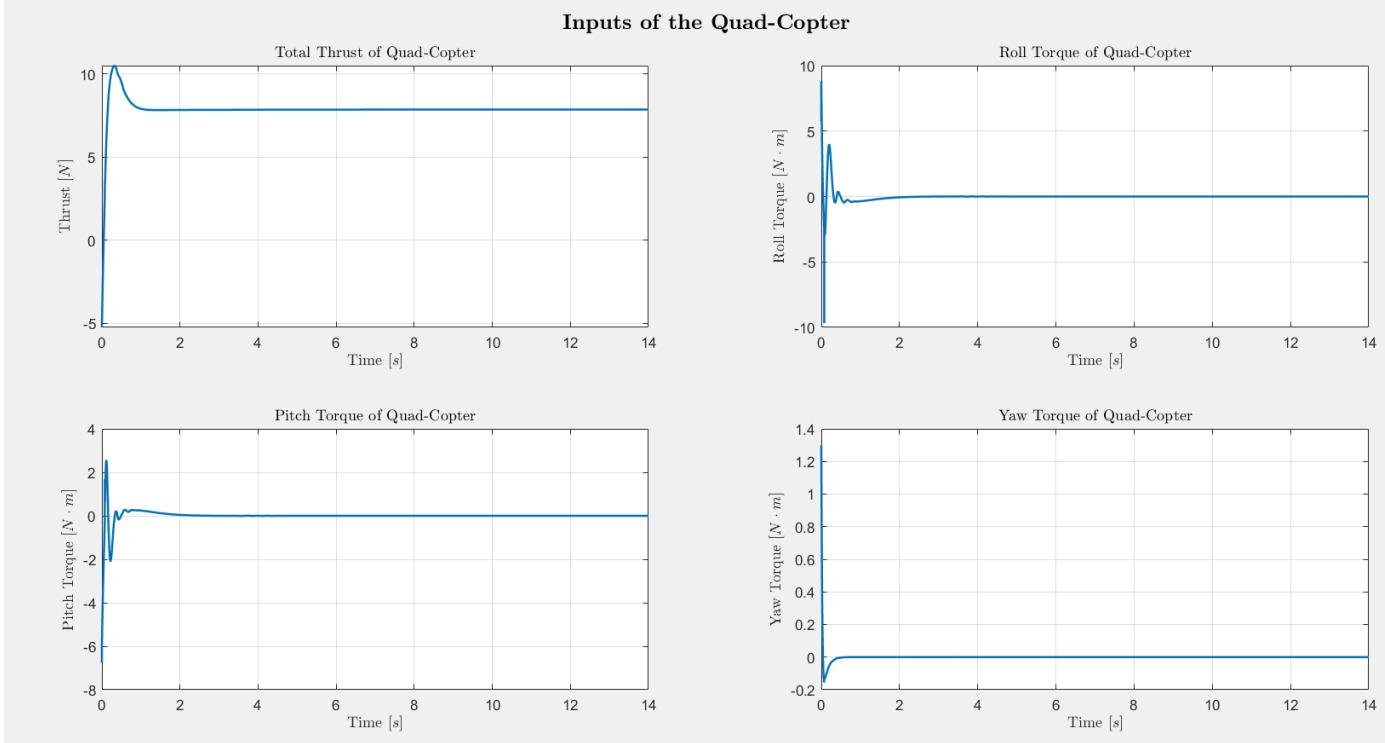


Figure 4.7: Inputs of the quadcopter with PID controller

The simulations confirm the PID controller's effectiveness in stabilizing the quadcopter. The control input plots (Fig. 4.7) show that thrust rapidly settles to the hovering value ($\approx m \cdot g$), while control torques converge to zero within 1.5 seconds. Regarding kinematics (Fig. 4.6), the system promptly corrects initial conditions: angular oscillations are damped within approximately 2 seconds, and the inertial position reaches the target

origin in about 4 seconds.

The system remains stable with zero steady-state error.

4.2 Optimal \mathcal{H}_∞ , \mathcal{H}_2 and \mathcal{L}_1 Control

The following **extended plant** description is considered:

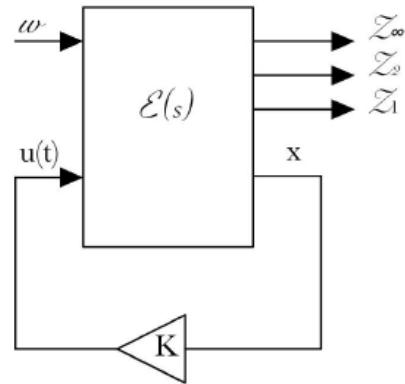


Figure 4.8: Extended Plant of the model

where

- ω is a vector that collect all exogenous signals acting on the model (noise, disturbance, ...).
- z_∞ , z_2 and z_1 is the vector of control objectives. Each component should be minimized by the choise of the controller.
- x represents the measured vector.
- $u(t)$ represents the vector of all commands to the system actuators.

4.2.1 External Disturbance Description



Figure 4.9: Real outdoor conditions

To evaluate the robustness of the control architecture, an external disturbance is introduced into the system dynamics.

The disturbance is modeled as a **wind gust** acting on the quadcopter's translational motion.

Mathematically, the wind is treated as an additive disturbance vector

$$\mathbf{d} = \begin{bmatrix} d_x & d_y & d_z \end{bmatrix}^T$$

acting on the linear acceleration states. Consequently, the translational equations of motion from the system model (2.8) are modified as follows:

$$\begin{cases} \dot{x}_2 = \frac{u_1}{m} (\cos(x_7) \sin(x_9) \cos(x_{11}) + \sin(x_7) \sin(x_{11})) + \mathbf{d}_x \\ \dot{x}_4 = \frac{u_1}{m} (\cos(x_7) \sin(x_9) \sin(x_{11}) - \sin(x_7) \cos(x_{11})) + \mathbf{d}_y \\ \dot{x}_6 = \frac{u_1}{m} (\cos(x_7) \cos(x_9)) - g + \mathbf{d}_z \end{cases}$$

In the simulations, two distinct wind gust scenarios are implemented.

Mild Gust - Scenario 1

In this first scenario, the system is subjected to an external disturbance designed to simulate a transient environmental perturbation of moderate intensity. As illustrated in the wind profile plot, the disturbance is modeled as a rectangular pulse.

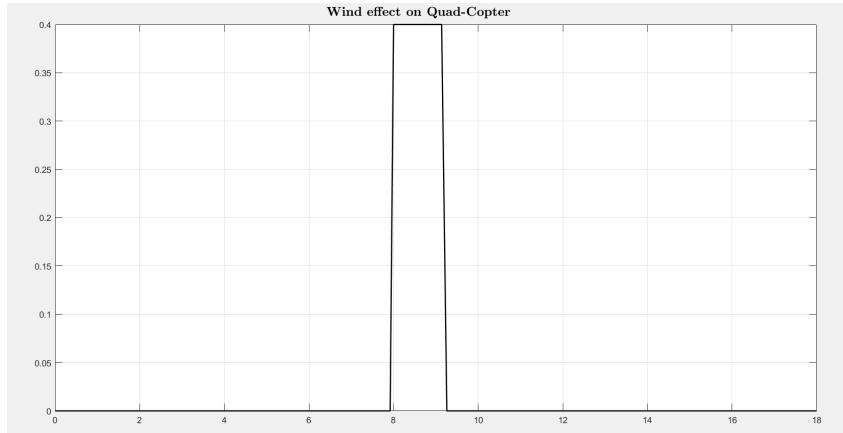


Figure 4.10: Mild Gust Profile

The disturbance is applied after the system has reached a steady state, allowing the analysis to isolate the wind effect from the initial step response.

Its duration is 1 second with a constant magnitude of 0.4.

Intense Gust - Scenario 2

The second scenario introduces more severe operating conditions to test the robustness of the control system. The perturbation simulates a stronger and more persistent wind gust, presenting a significant challenge to the aircraft's stability.

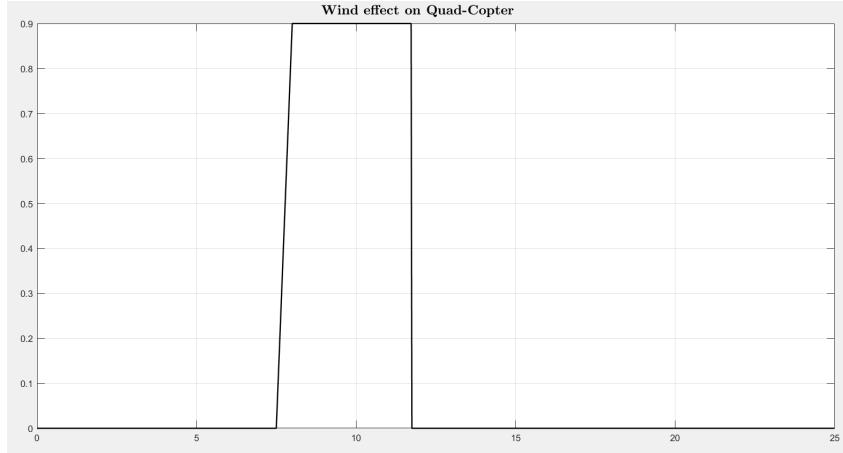


Figure 4.11: Mild Gust Profile

It starts at $t = 8$ seconds with a duration of 4 seconds and a magnitude of 0.9.

This extended duration allows for the observation of not only the transient response to the wind impact but also the controller's capacity to counteract a constant force over a prolonged period.

4.2.2 State-Space Representation and Formulation

The **Open-Loop System** is described by the following state-space equations:

$$\begin{cases} \dot{x} = Ax + B_1u + B_2\omega \\ z_\infty = C_1x + D_{11}\omega + D_{12}u \\ z_2 = C_2x + D_{22}u \\ z_1 = C_3x + D_{31}\omega + D_{32}u \end{cases} \quad (4.7)$$

The objective is to design a **Static State Feedback Controller** defined as

$$u = Kx \quad (4.8)$$

Where K is the gain matrix to be determined.

While the state matrix A and the control input matrix $B_1 = B$ are derived directly from the linearized system dynamics, the disturbance input matrix B_2 must be explicitly defined to map the exogenous inputs to the state derivatives.

$$B_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{m} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1.25 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.25 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.25 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{n \times n_w} = \mathbb{R}^{12 \times 6} \quad (4.9)$$

4.2.3 \mathcal{H}_∞ Optimal Control

The objective of the **\mathcal{H}_∞ control** is to design a controller that guarantees robust performance by minimizing the effect of external disturbances on the system output. This is achieved by minimizing the **H_∞ norm** of the closed-loop transfer function between exogenous inputs and controlled outputs.

For a linear time-invariant (LTI) system described by the transfer function $G(s)$ the H_∞ norm is defined as:

$$\|G\|_\infty = \sup_{\omega \in \mathbb{R}} \bar{\sigma}(G(j\omega))$$

where $\bar{\sigma}(\cdot)$ denotes the maximum singular value. This norm represents the worst-case gain of the system over all frequencies.

From the theory of **induced norms**, the H_∞ norm provides an upper bound on the input–output energy relation:

$$\|y\|_2 \leq \|G\|_\infty \|u\|_2$$

where u and y denote the input and output signals, respectively.

Consequently, minimizing the H_∞ norm limits the maximum energy amplification from input to output, ensuring effective attenuation of disturbances.

The \mathcal{H}_∞ control problem can be efficiently solved using Linear Matrix Inequalities (LMIs) by exploiting the Bounded Real Lemma. In particular, the existence of a symmetric

positive-definite matrix $P = P^T > 0$ satisfying suitable LMIs guarantees both the stability of the closed-loop system and the fulfillment of the performance constraint

$$\|G_{\omega z}\|_\infty < \gamma$$

The synthesis of the \mathcal{H}_∞ optimal controller consists of solving the following optimization problem

$$\begin{aligned} \begin{bmatrix} X^* & Y^* \end{bmatrix} &= \arg \min_{X, Y, \gamma} \quad \gamma \\ \text{s.t.} \quad & \begin{bmatrix} (AX + B_1Y) + (AX + B_1Y)^T & B_2 & (C_1X + D_{12}Y)^T \\ B_2^T & -\gamma I_{n_w} & D_{11}^T \\ (C_1X + D_{12}Y) & D_{11} & -\gamma I_{n_z} \end{bmatrix} < 0, \\ & X = X^T > 0. \end{aligned}$$

If a solution exists, it is unique, and the optimal controller is given by

$$K_\infty^* = Y^*(X^*)^{-1}$$

Results and Discussion

In order to first assess the nominal behavior of the \mathcal{H}_∞ controller, simulations are performed in the absence of external disturbances. This preliminary analysis allows the evaluation of the intrinsic closed-loop properties, such as stability and transient performance, without the influence of exogenous inputs.

The closed-loop system is initialized with non-zero initial conditions, while the disturbance input is set to zero for the entire simulation horizon. This choice enables a clear observation of the system dynamics under the sole action of the state-feedback controller. Considering

$$\dot{x}(t) = (A + B_1 K_\infty) x(t)$$

and its eigenvalues:

$$\lambda_{cl\infty} = \begin{bmatrix} -6.12 + e03 + 0.00i \\ -14.55 + 0.00i \\ -1.51 + 1.43i \\ -1.51 - 1.43i \\ -0.66 + 0.99i \\ -0.66 - 0.99i \\ -0.72 + 1.11i \\ -0.72 - 1.11i \\ -0.19 + 0.19i \\ -0.19 - 0.19i \\ -1.19 + 0.00i \\ -1.32 + 0.00i \end{bmatrix}$$

This confirms the asymptotic stability of the controlled system.

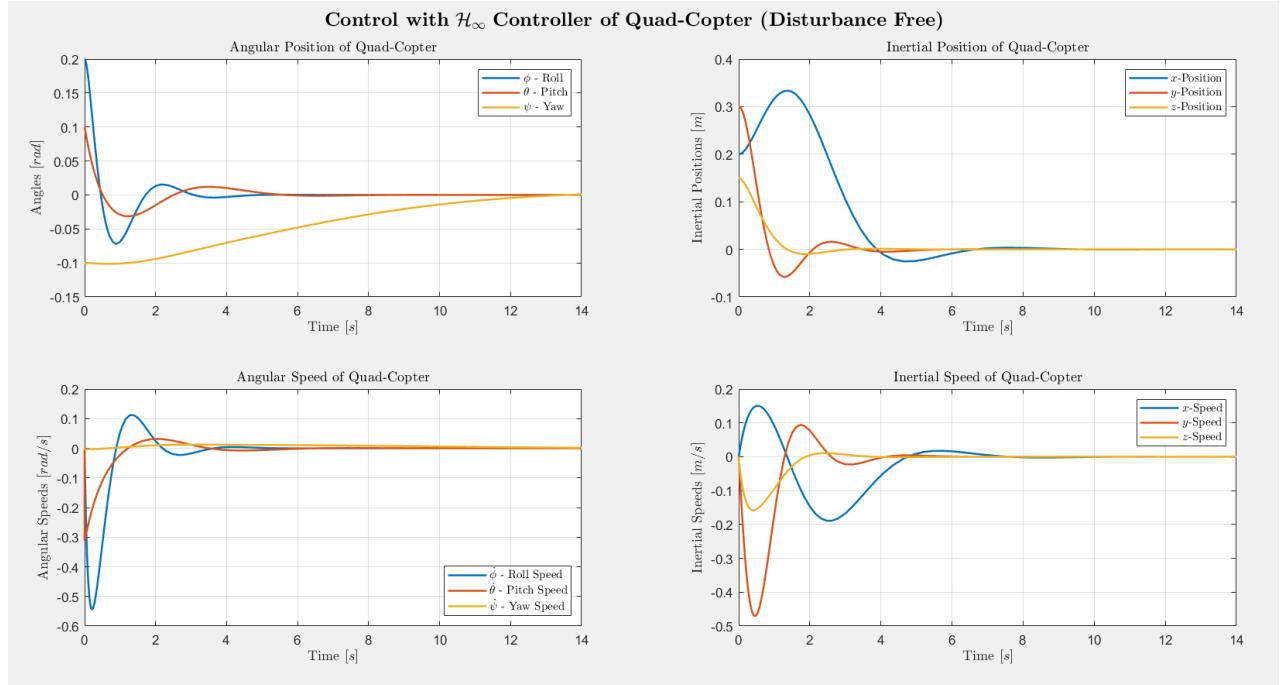


Figure 4.12: States evolution of the quadcopter with \mathcal{H}_∞ controller (Disturbance Free)

The simulation results are consistent with the theoretical stability properties of the closed-loop system and confirm the effectiveness of the \mathcal{H}_∞ controller under nominal conditions. From a physical standpoint, the observed closed-loop behavior is consistent with the dynamics of a linearized quadrotor model. The transient oscillations in attitude and position are caused by the coupling between rotational and translational dynamics and are well damped by the \mathcal{H}_∞ controller.

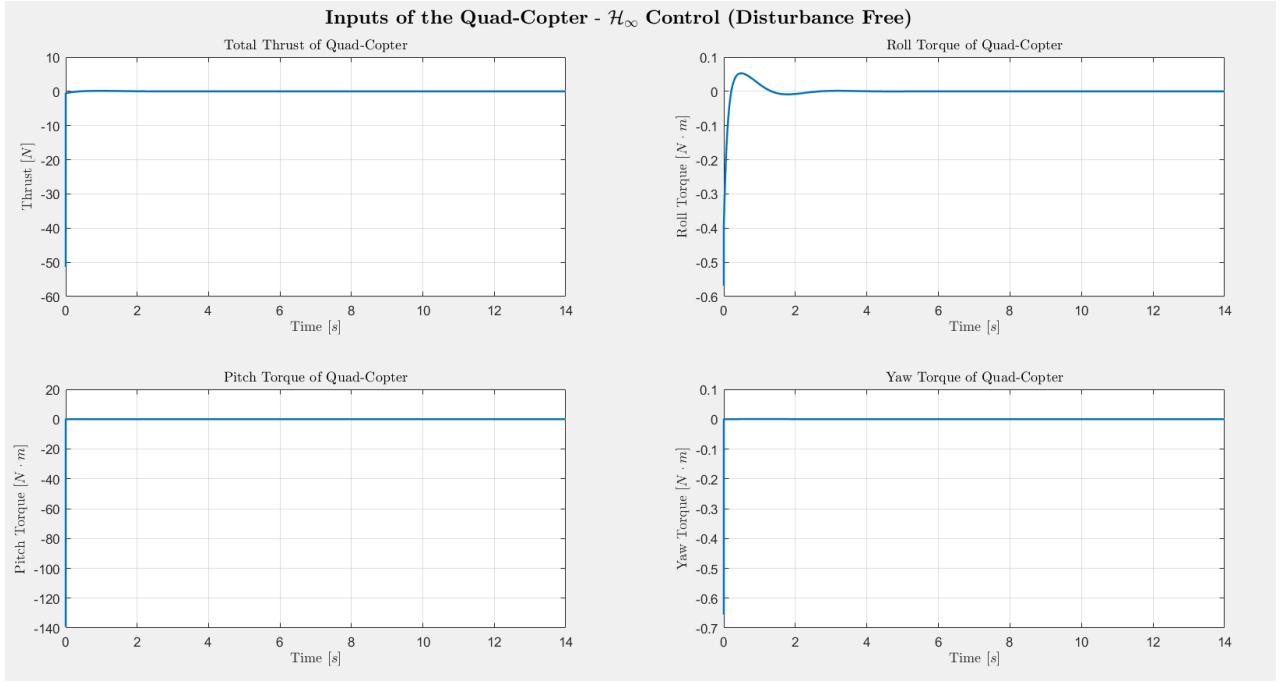


Figure 4.13: Inputs of the quadcopter with \mathcal{H}_∞ controller (Disturbance Free)

The absence of disturbances and actuator limitations leads to an idealized response; however, the results remain physically meaningful and provide a valid assessment of the nominal closed-loop performance.

4.2.4 \mathcal{H}_2 Optimal Control

The objective of **\mathcal{H}_2 control** is to design a controller that minimizes the **energy of the output** in response to stochastic or energy-limited disturbances. This is formalized through the input–output energy relation:

$$\|y\|_2 \leq \|G\|_2 \|w\|_2,$$

where w is the disturbance input, y is the performance output, and $\|G\|_2$ is the \mathcal{H}_2 norm of the system. Minimizing the \mathcal{H}_2 norm corresponds to *reducing the expected output energy* while respecting the input energy, which leads to improved performance without excessive control effort.

For a linear time-invariant (LTI) system described in state-space form, the synthesis of the \mathcal{H}_2 optimal controller can be formulated as an optimization problem using Linear Matrix Inequalities (LMIs):

$$\begin{bmatrix} X^* & Y^* \end{bmatrix} = \arg \min_{X, Y, Q} \text{tr}\{Q\}$$

s.t.

$$\begin{bmatrix} (AX + B_1Y) + (AX + B_1Y)^T & B_2 \\ B_2^T & -I_{n_w} \end{bmatrix} < 0,$$

$$\begin{bmatrix} Q & (C_2X + D_{22}Y) \\ (C_2X + D_{22}Y)^T & X \end{bmatrix} < 0,$$

$$X = X^T > 0, \quad Q = Q^T > 0.$$

where $\mathbf{Q} \in \mathbb{R}^{n_z \times n_z}$ is a symmetric positive-definite matrix related to the output energy. If a feasible solution exists, it is unique, and the corresponding optimal static state-feedback controller is recovered as:

$$K_2^* = Y^*(X^*)^{-1}.$$

Results and Discussion

Similar to \mathcal{H}_∞ , \mathcal{H}_2 simulations are purely nominal.

Considering the closed loop system:

$$\dot{\mathbf{x}}(t) = (A + B_1 K_2) \mathbf{x}(t)$$

and its eigenvalues

$$\lambda_{cl_2} = \begin{bmatrix} -5.30 + 9.48i \\ -5.30 - 9.48i \\ -8.94 + 3.97i \\ -8.94 - 3.97i \\ -7.87 + 7.26i \\ -7.87 - 7.26i \\ -7.54 + 5.74i \\ -7.54 - 5.74i \\ -2.84 + 2.84i \\ -2.84 - 2.84i \\ -2.86 + 2.89i \\ -2.86 - 2.89i \end{bmatrix}$$

This confirms, also in this case, the asymptotic stability of the controlled system.

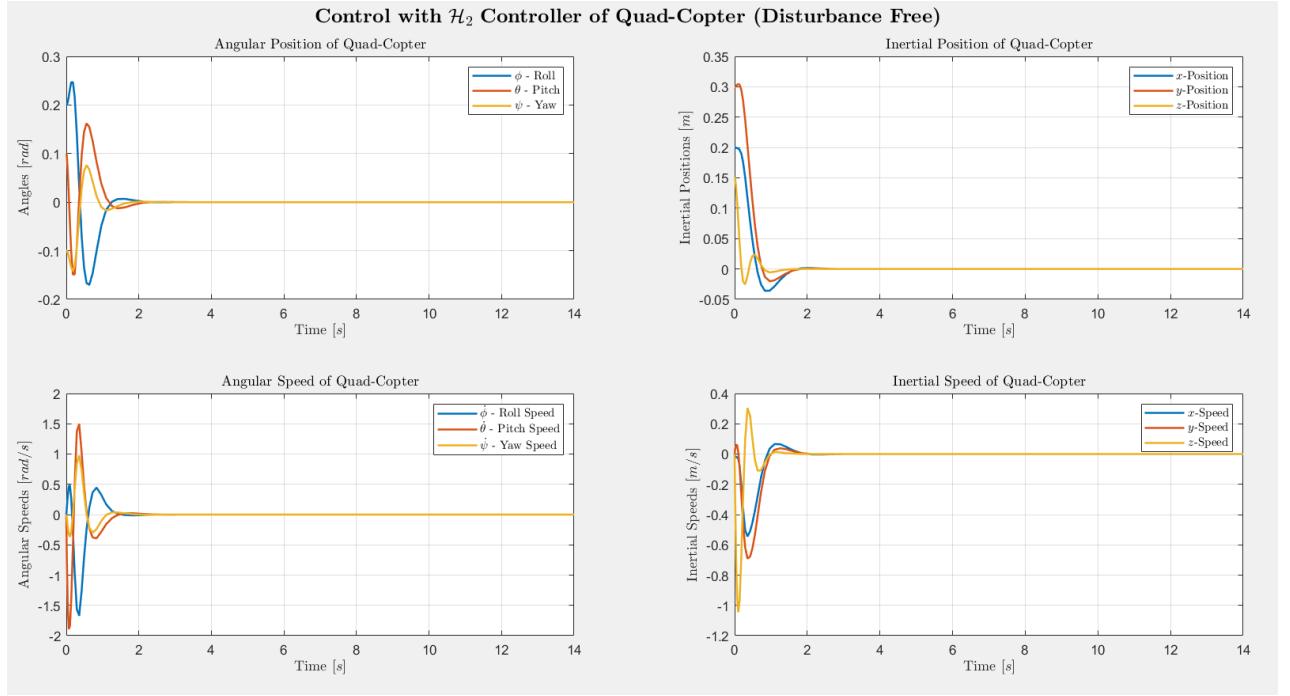


Figure 4.14: States evolution of the quadcopter with \mathcal{H}_2 controller (Disturbance Free)

The response analysis highlights an *underdamped behavior*, with initial oscillations that are fully attenuated within a settling time of approximately 2.5 seconds. The velocities, although exhibiting high peaks during the initial transient to correct the error, rapidly converge to zero, ensuring the asymptotic stability of the system.

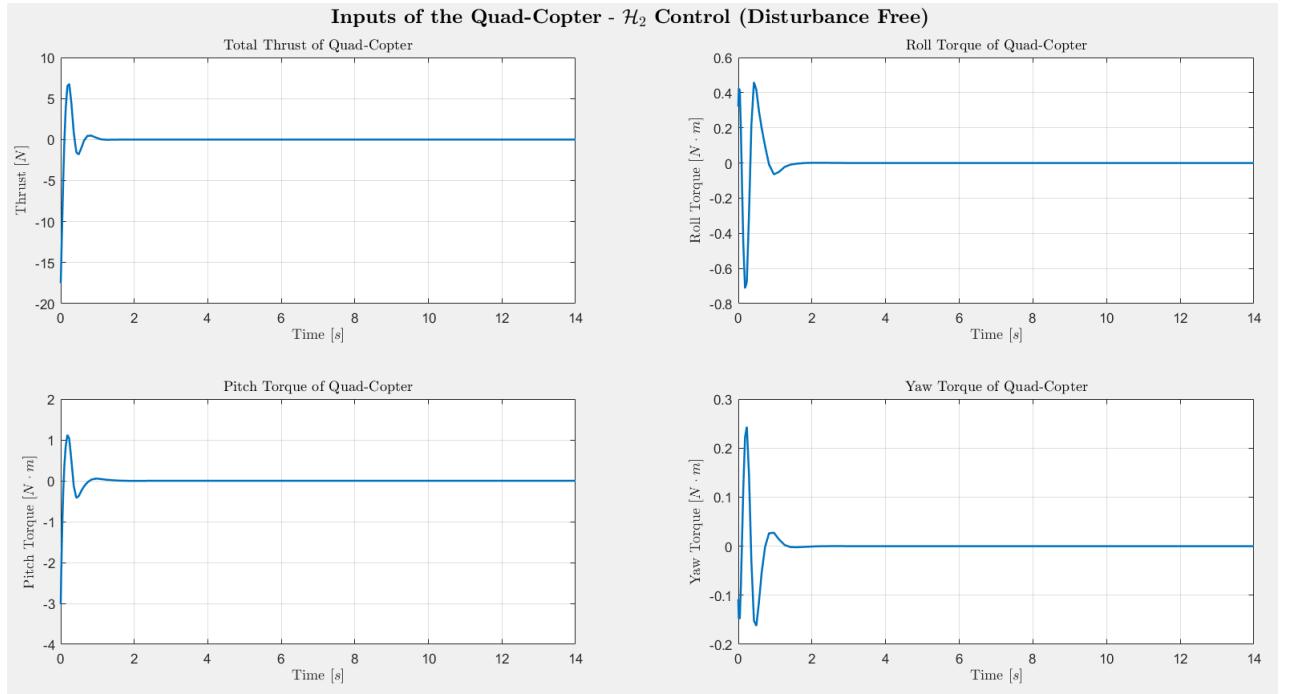


Figure 4.15: Inputs of the quadcopter with \mathcal{H}_2 controller (Disturbance Free)

4.2.5 \mathcal{L}_1 Optimal Control

The objective of \mathcal{L}_1 control is to achieve fast adaptation and robustness in the presence of uncertainties and external disturbances, while guaranteeing bounded control action and predictable transient performance. Unlike classical robust control techniques, \mathcal{L}_1 control explicitly separates the objectives of robustness and adaptation.

The key idea of \mathcal{L}_1 control is to limit the \mathcal{L}_1 norm of the closed-loop transfer function from disturbances to system outputs, thereby ensuring a bounded worst-case effect of uncertainties:

$$\|y\|_\infty \leq \|G\|_{\mathcal{L}_1} \|w\|_\infty.$$

This formulation allows the controller to achieve rapid adaptation without compromising robustness, since the control signal is filtered through a low-pass filter that prevents high-frequency oscillations and excessive control effort.

For a linear time-invariant (LTI) system described in state-space form, the synthesis of the \mathcal{L}_1 optimal controller can be formulated as an optimization problem using Linear Matrix Inequalities (LMIs):

$$\begin{aligned} \begin{bmatrix} X^* & Y^* \end{bmatrix} = \arg \min_{X, Y, \mu, \zeta} \quad & \zeta \\ \text{s.t.} \quad & \begin{bmatrix} (AX + B_1Y) + (AX + B_1Y)^T + \lambda X & B_2 \\ B_2^T & -\mu I_{n_w} \end{bmatrix} < 0, \\ & \begin{bmatrix} \lambda X & 0 & (C_3X + D_{32}Y)^T \\ 0 & (\zeta - \mu)I_{n_w} & D_{31}^T \\ (C_3X + D_{32}Y) & D_{31} & \zeta I_{n_z} \end{bmatrix} < 0, \\ & X = X^T > 0, \quad \mu > 0, \quad (\lambda > 0 \text{ fixed}) \end{aligned}$$

If a feasible solution exists, it is unique, and the corresponding optimal static state-feedback controller is recovered as:

$$K_1^* = Y^*(X^*)^{-1}.$$

Results and Discussion

Under nominal conditions, following the same approach used for the other controllers, the analysis is then performed for the \mathcal{L}_1 controller.

Considering the closed loop system and taking $\lambda = 0.5$ as the parameter for the LMI solution to compute K_1 :

$$\dot{x}(t) = (A + B_1 K_1)x(t)$$

and its eigenvalues

$$\lambda_{cl_1} = \begin{bmatrix} -5.19 + 8.04i \\ -5.19 - 8.04i \\ -5.59 + 8.01i \\ -5.59 - 8.01i \\ -10.87 + 0.00i \\ -8.85 + 1.51i \\ -8.85 - 1.51i \\ -6.88 + 0.76i \\ -6.88 - 0.76i \\ -5.82 + 0.46i \\ -5.82 - 0.46i \\ -4.98 + 0.00i \end{bmatrix}$$

This confirms the asymptotic stability of the controlled system.

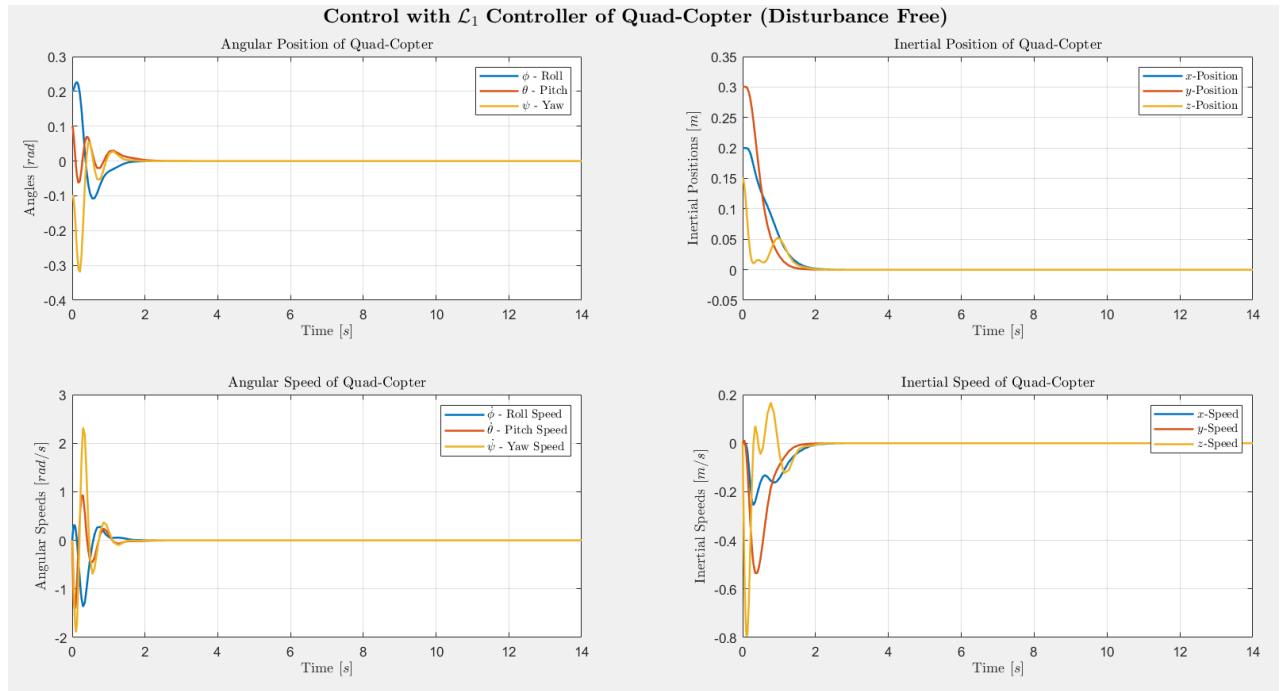


Figure 4.16: States evolution of the quadcopter with \mathcal{L}_1 controller (Disturbance Free)

The transient analysis highlights a fast and decisive response: the angular and linear velocities exhibit significant initial peaks (necessary to instantaneously compensate for the position error), which are quickly damped, ensuring stable asymptotic convergence without significant residual oscillations.

As shown in the Figure (4.17), the system reaches the rest condition (zero inputs) in less than 2 seconds, confirming the high responsiveness of the controller and its ability to damp oscillations in minimal time.

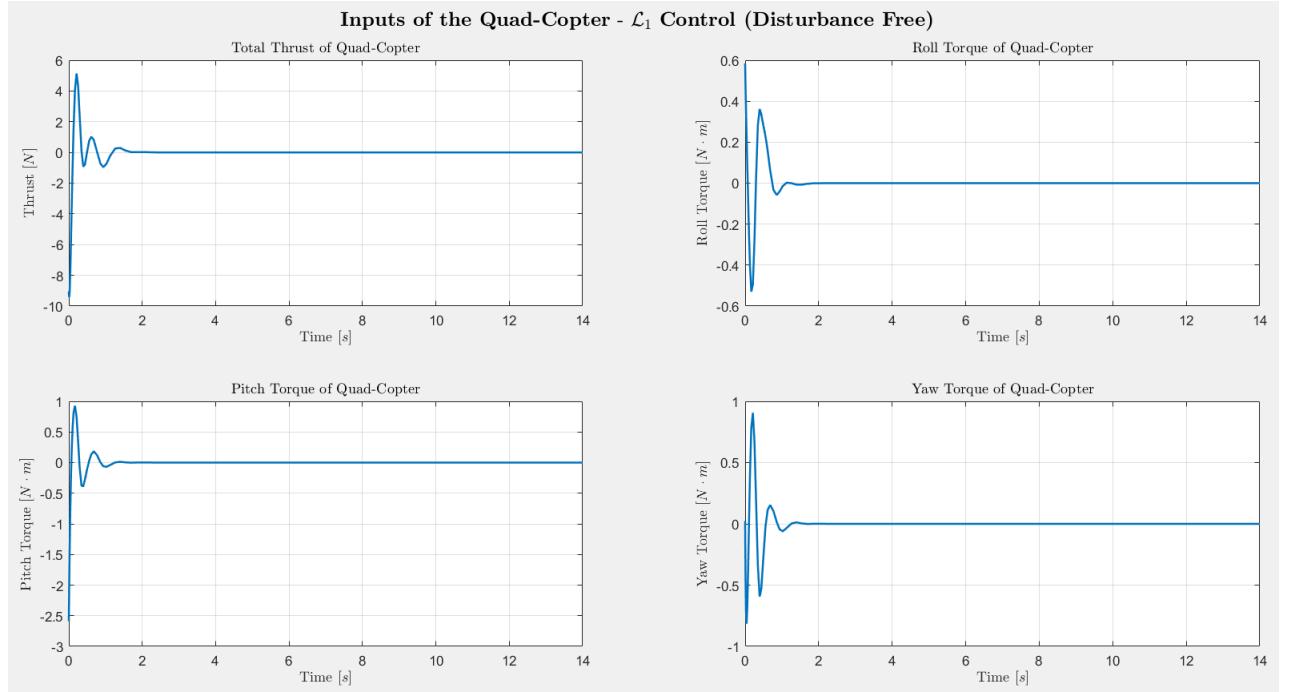


Figure 4.17: Inputs of the quadcopter with \mathcal{L}_1 controller (Disturbance Free)

4.2.6 Comparison of Controllers under Nominal Conditions

In this section, a comparative analysis of the **PID**, \mathcal{H}_∞ , \mathcal{H}_2 and \mathcal{L}_1 controllers is presented under nominal conditions, i.e., without external disturbances or model uncertainties.

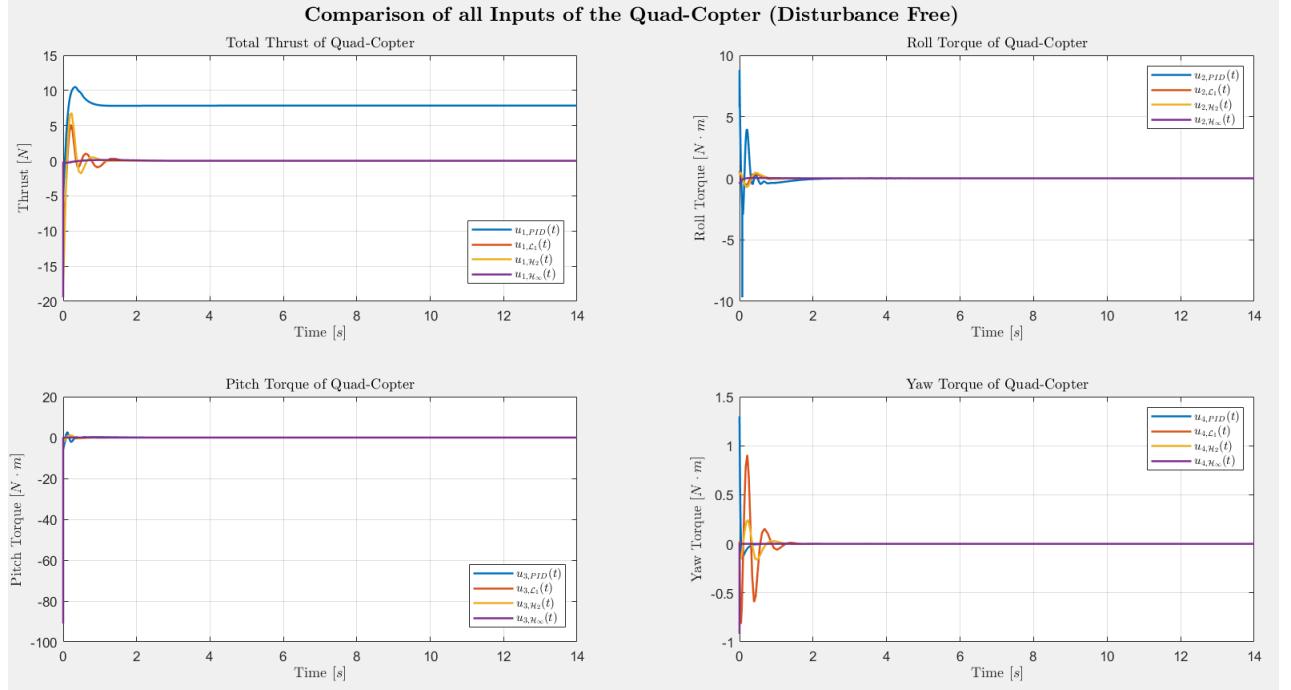


Figure 4.18: Comparative plot of control inputs generated by the controllers (Disturbance Free)

It is important to highlight the role of the feed-forward term (denoted in the control

scheme as $u_{eq}(1)$), which represents the open-loop component necessary to compensate for known system forces, in this case the gravitational force.

This design allows the feedback controller to operate in the linear regime, focusing exclusively on dynamic perturbations and trajectory corrections, while the static hovering condition is ensured a priori by the known feed-forward term ($u \approx m \cdot g$).

As a consequence The PID controller, which does not explicitly separate the equilibrium thrust from the feedback action, shows the total thrust including the weight compensation. Instead, the advanced controllers designed around the linearized deviation model, only regulate deviations from equilibrium, and their outputs converge to zero once the drone is stable. The actual thrust applied to the motors is still u_{eq} , but the plotted signals reflect only the corrective component.

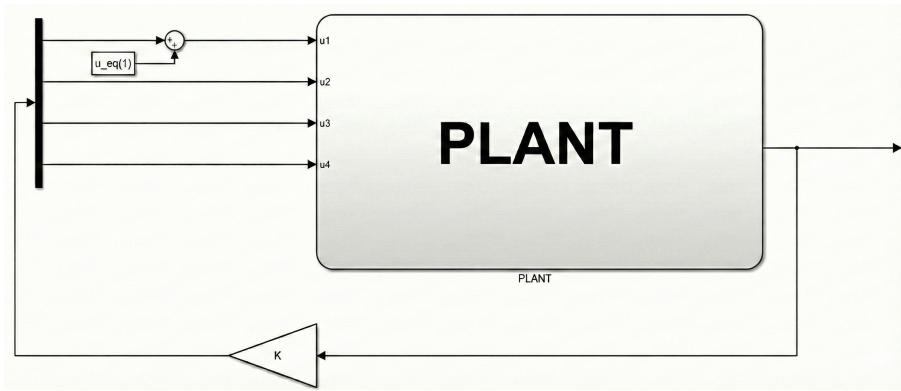


Figure 4.19: Control Scheme with feed-forward term

Figure (4.18) provides a comparative view of all control inputs, allowing an evaluation of control effort and regulation quality:

- **Total Thrust (u_1)**: in the PID controller it has an aggressive response and then settles to the hover value. In the \mathcal{H}_2 and \mathcal{L}_1 controller it oscillate around zero with initial peaks (smaller compared to PID).
- **Roll Torque (u_2) and Yaw Torque (u_4)**: in the \mathcal{L}_1 controller they have high reactivity but exhibits high-frequency oscillations during the initial phase. In the \mathcal{H}_2 they appear "softer" compared to the \mathcal{L}_1 controller, characterized by smoother trajectories. This reflects the inherent nature of \mathcal{H}_2 control, which minimizes the average energy of both the error and the control effort, thereby penalizing sudden peaks. Then, the PID controller shows longer and less clean ("noisy") transients, featuring larger overshoots before settling.
- **Pitch Torque (u_4)**: \mathcal{L}_1 and \mathcal{H}_2 maintains controlled activity with minimal oscillations, while PID peaks are higher.

In nearly all torque plots, the \mathcal{H}_∞ line remains almost flat or extremely close to zero compared to the other controllers.

This because the \mathcal{H}_∞ controller is explicitly designed for robustness (ensuring performance even in worst-case disturbance scenarios). In a "Disturbance-Free" (ideal) scenario like this, it tends to be highly conservative, applying only the minimum necessary intervention. While it yields a slower response, it guarantees that the system never drifts out of control.

4.2.7 Comparison of Controllers under Disturbance Conditions - Scenario 1

The analysis is conducted under **Scenario 1** (mild gust), where a low-intensity external disturbance is applied to the system. The purpose of this scenario for this controller and for all the others analyzed in the simulations, is not to evaluate recovery from large deviations, but rather to assess the disturbance rejection capability of the different controllers when the quadrotor is already operating at the equilibrium condition.

Accordingly, the system is initialized in steady-state conditions, with the initial state set to

$$x_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

corresponding to hovering at the equilibrium point. Under these conditions, the disturbance acts as a perturbation around the operating point, allowing a direct comparison of how the \mathcal{H}_∞ , \mathcal{H}_2 and \mathcal{L}_1 controllers attenuate its effects while preserving stability and nominal performance.

\mathcal{H}_∞ Control - Scenario 1

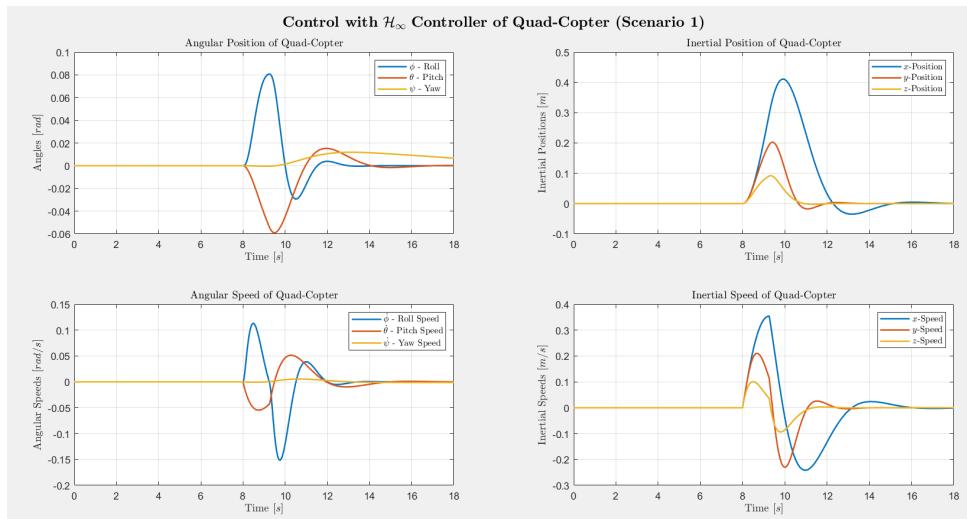


Figure 4.20: States evolution of the quadcopter with \mathcal{H}_∞ controller (Scenario 1)

The quadcopter is displaced from its initial position and tilted by the gust. The controller reacts by compensating for both the positional and attitude deviations, driving the system back to the initial reference position.

The control system successfully rejects the perturbation and restores stability. The drone returns to steady state within approximately 6 seconds, demonstrating effective disturbance rejection and closed-loop stability.

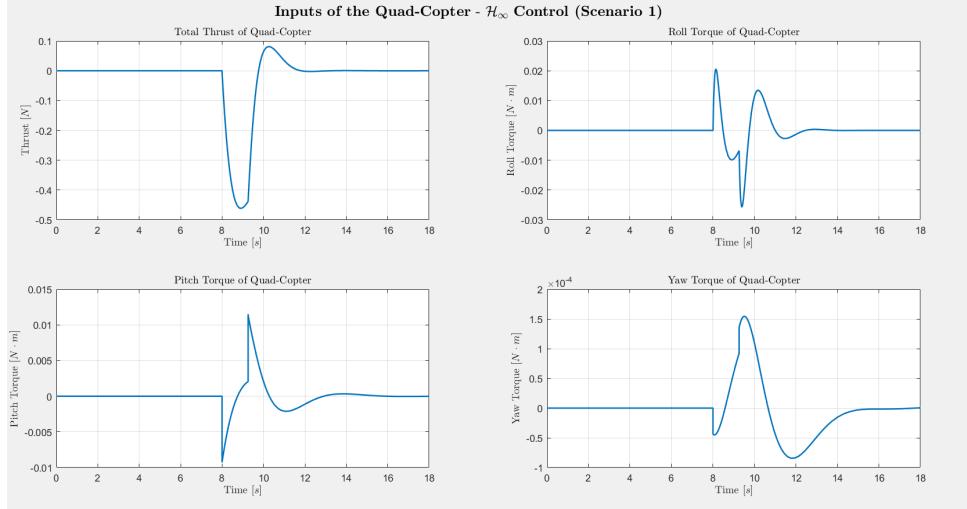


Figure 4.21: Inputs of the quadcopter with \mathcal{H}_∞ controller (Scenario 1)

\mathcal{H}_2 Control - Scenario 1

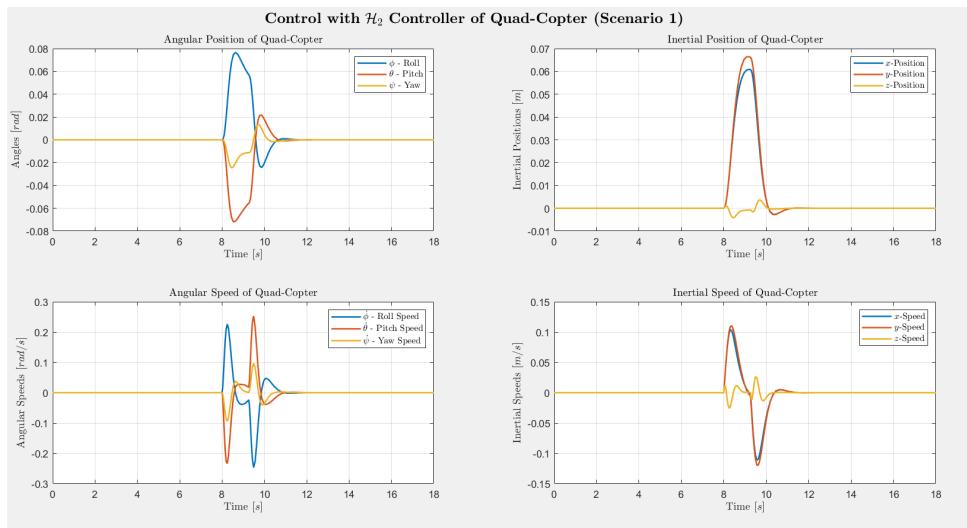


Figure 4.22: States evolution of the quadcopter with \mathcal{H}_2 controller (Scenario 1)

The motors respond in a highly aggressive and fast manner. Sharp and immediate peaks in both torque and thrust are observed, indicating a nervous but prompt actuator behavior. The displacement from the zero reference position remains minimal. Following the disturbance, the drone rapidly returns to a stationary condition, reaching steady state within

a very short time (by $t = 11\text{s}$).

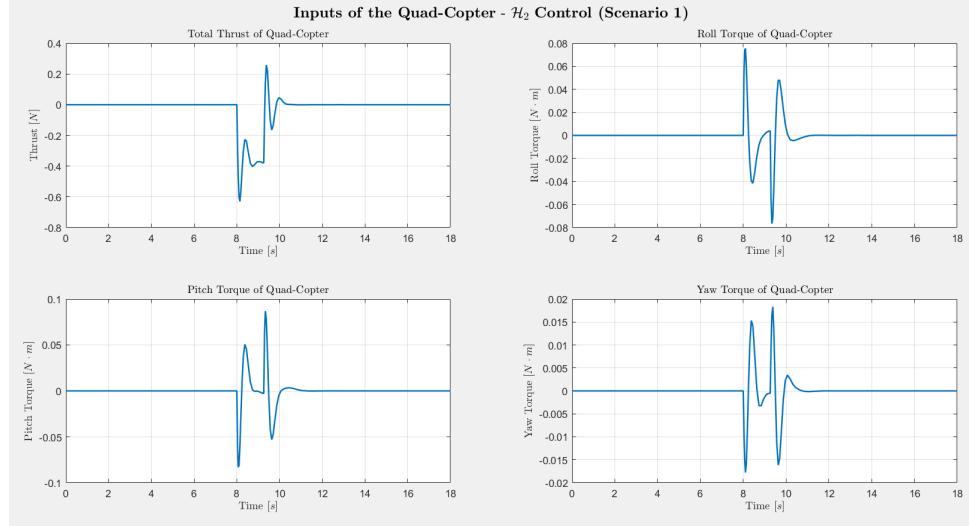


Figure 4.23: Inputs of the quadcopter with \mathcal{H}_2 controller (Scenario 1)

\mathcal{L}_1 Control - Scenario 1

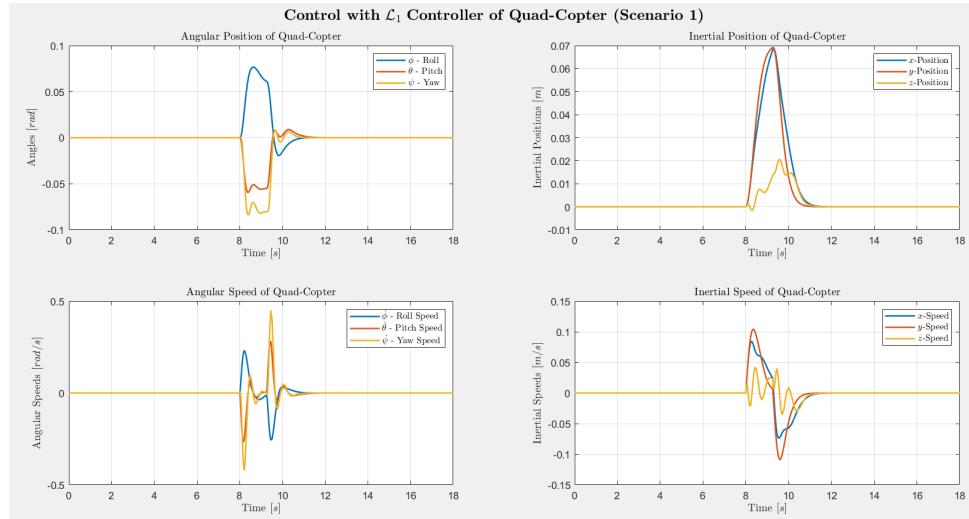


Figure 4.24: States evolution of the quadcopter with \mathcal{L}_1 controller (Scenario 1)

As shown in the Figure (4.25), the control commands exhibit high-frequency oscillations. The controller continuously and extremely rapidly adapts the motor inputs in order to cancel the disturbance in real time, highlighting an aggressive and highly responsive control action.

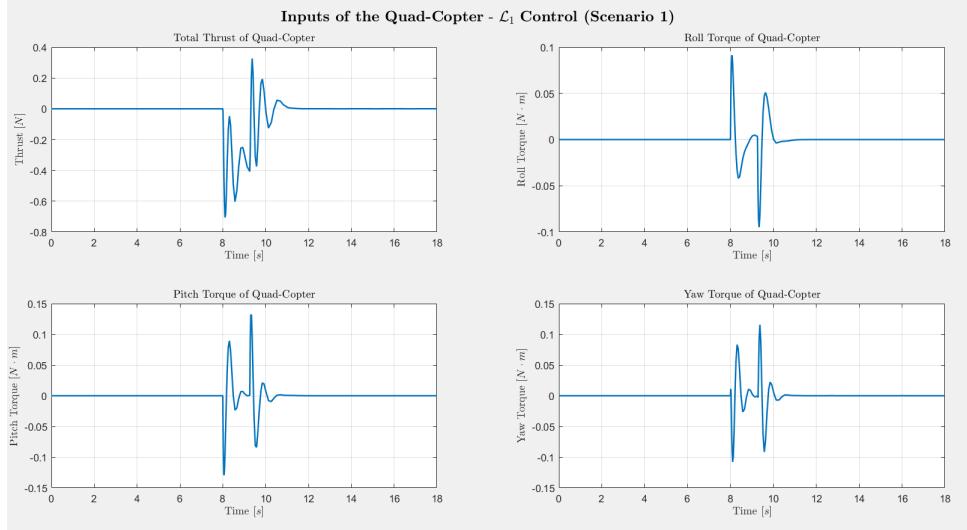


Figure 4.25: Inputs of the quadcopter with \mathcal{L}_1 controller (Scenario 1)

Controllers Comparison and Discussion - Scenario 1

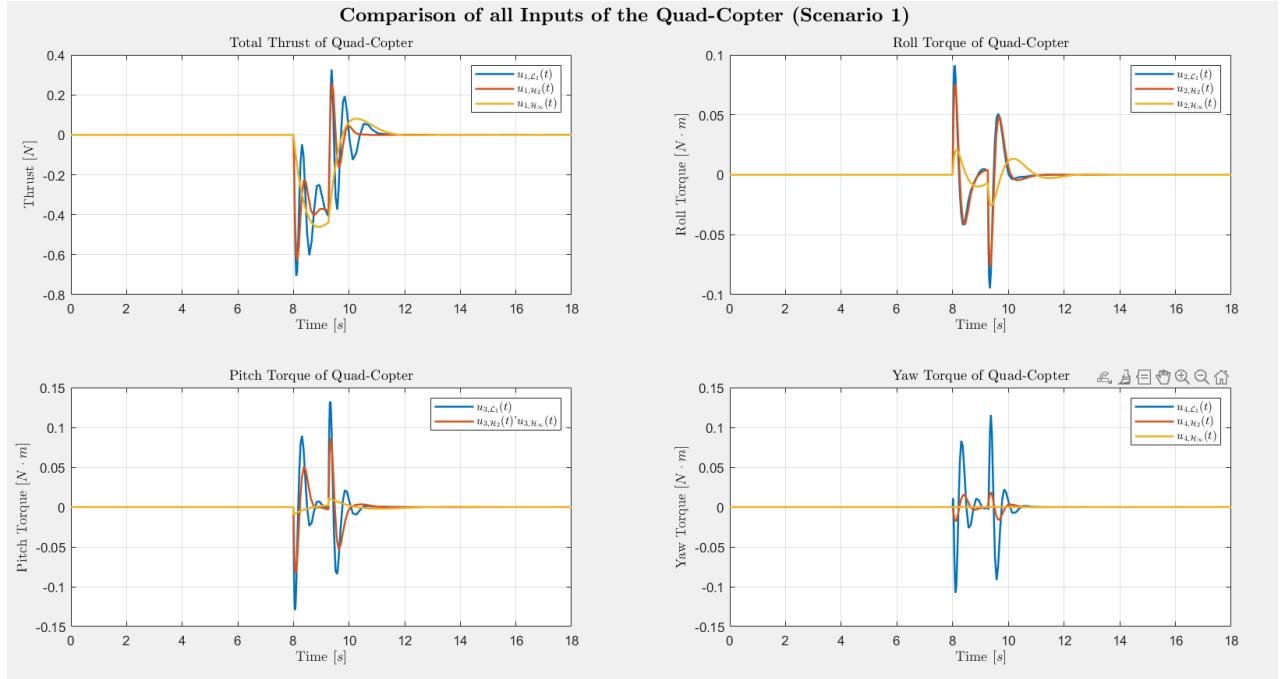


Figure 4.26: Comparative plot of control inputs generated by the controllers (Scenario 1)

The \mathcal{H}_{∞} controller is the least performant in terms of **precision**. It allows the quadcopter to deviate by approximately 40 cm along the x -axis before recovery. The correction action is slow and smooth, indicating a conservative control response.

The \mathcal{H}_2 delivers excellent performance. The position error is drastically reduced to approximately 6–7 cm. The recovery is fast and occurs without significant residual oscillations, indicating a well-damped and precise control behavior.

The \mathcal{L}_1 exhibits performance nearly identical to the \mathcal{H}_2 controller in terms of spatial error,

with a peak deviation of approximately 7 cm.

In terms of control inputs:

- \mathcal{H}_∞ requires the lowest control effort. The thrust and torque profiles are smooth and of low magnitude, indicating an efficient and non-aggressive actuation strategy.
- \mathcal{H}_2 represents a balanced compromise. It reacts with decisive peaks, yet the control signals remain clean, combining responsiveness with smooth actuation.
- \mathcal{L}_1 is the most aggressive and “nervous.” High-frequency oscillations are observed, indicating that the motors are continuously subjected to rapid command changes in order to maintain position.

Overall, in this case, the \mathcal{H}_2 controller is preferable in terms of position accuracy and error compared to the \mathcal{H}_∞ controller. Additionally, it is less demanding on the motors than the \mathcal{L}_1 controller, which exhibits high-frequency oscillations that can increase actuator wear.

The \mathcal{H}_∞ controller is not preferred in this case because it is mathematically designed for robustness.

Its primary objective is to ensure that the system remains stable even in the worst-case scenario or in the presence of uncertainties in the quadcopter model.

4.2.8 Comparison of Controllers under Disturbance Conditions

- Scenario 2

The analysis is now extended to Scenario 2 (intense gust), where a higher-intensity external disturbance is applied to the system. All other simulation conditions remain the same as in Scenario 1, allowing for a direct comparison of the controllers’ performance under more severe perturbations.

\mathcal{H}_∞ Control - Scenario 2

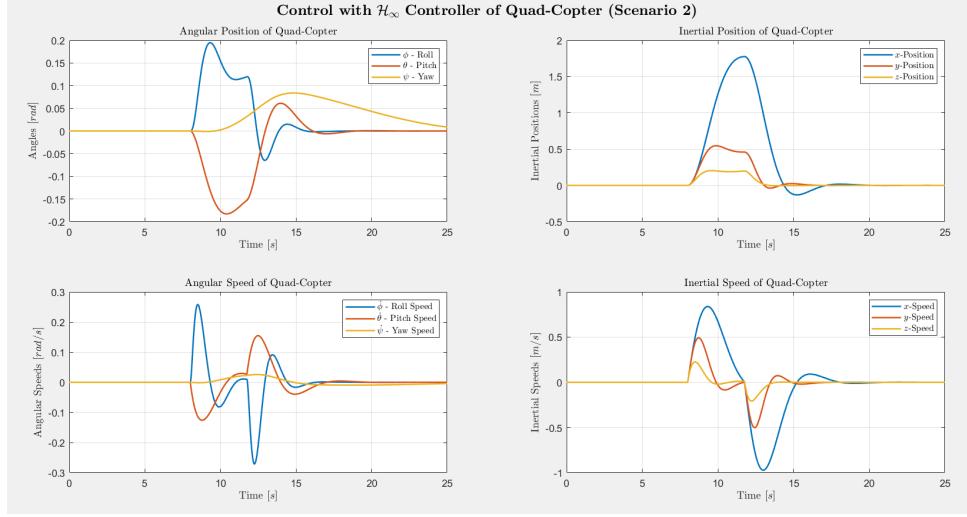


Figure 4.27: States evolution of the quadcopter with \mathcal{H}_∞ controller (Scenario 2)

In this scenario, a sustained gust pushes the drone significantly farther, reaching a displacement of nearly 1.8 m, compared to the instantaneous disturbance in the previous scenario.

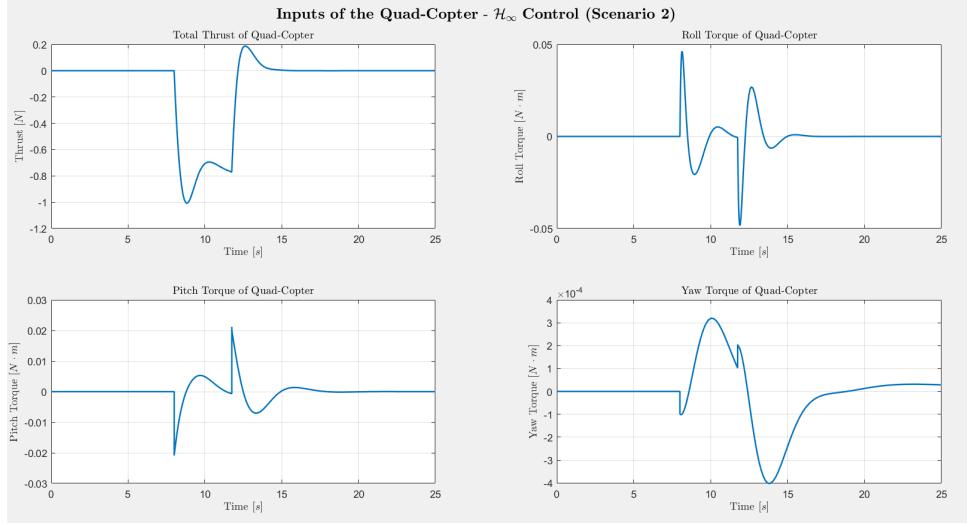


Figure 4.28: Inputs of the quadcopter with \mathcal{H}_∞ controller (Scenario 2)

\mathcal{H}_2 Control - Scenario 2

As shown in the Figures (4.29) and (4.30), the controller does not passively drift under the sustained gust; rather, it actively counteracts the disturbance by tilting, maintaining a negligible position error throughout the event.

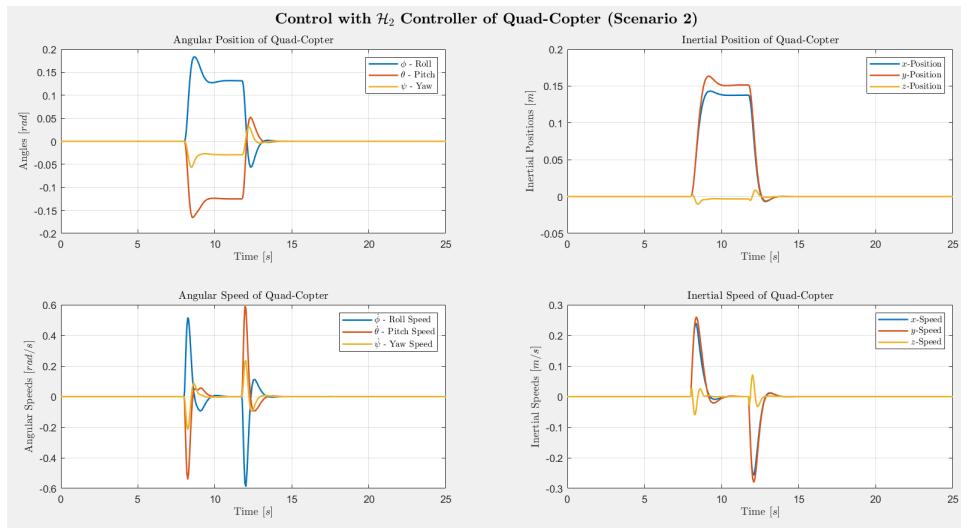


Figure 4.29: States evolution of the quadcopter with \mathcal{H}_2 controller (Scenario 2)

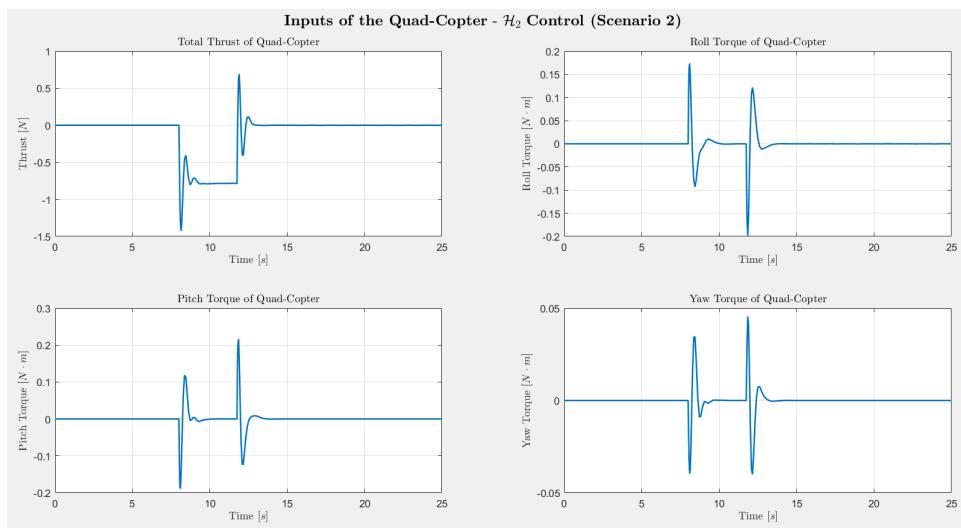


Figure 4.30: Inputs of the quadcopter with \mathcal{H}_2 controller (Scenario 2)

\mathcal{L}_1 Control - Scenario 2

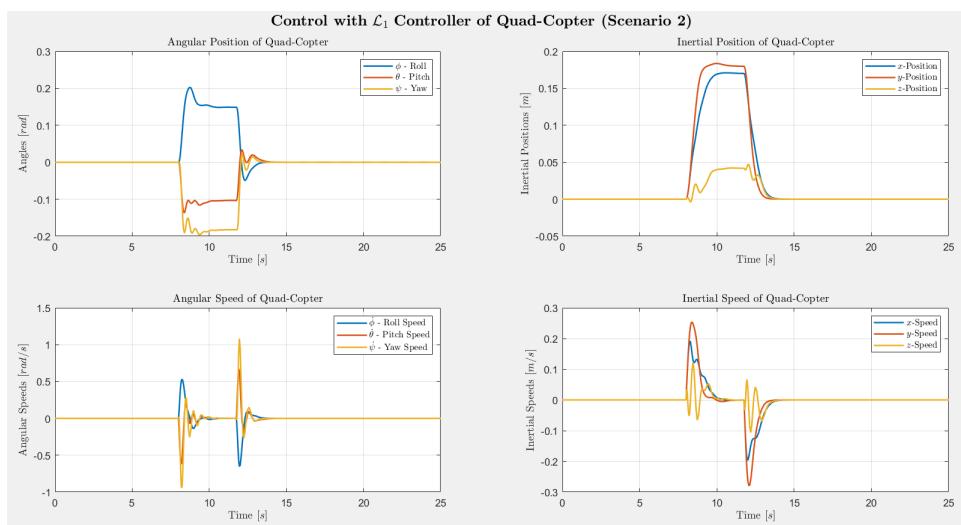


Figure 4.31: States evolution of the quadcopter with \mathcal{L}_1 controller (Scenario 2)

In this case as well, the controller neutralizes the continuous gust by rapidly and aggressively adjusting the commands, effectively compensating for the disturbance in real time.

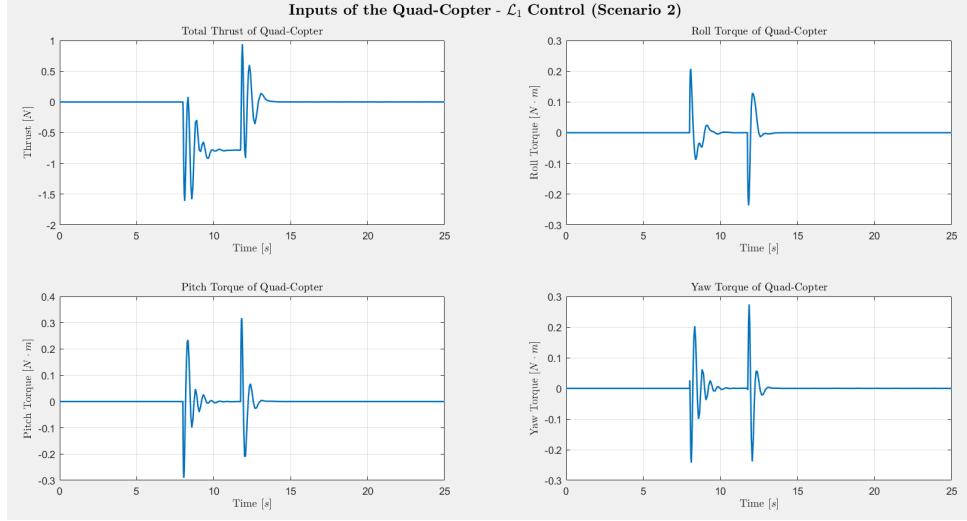


Figure 4.32: Inputs of the quadcopter with \mathcal{L}_1 controller (Scenario 2)

Controllers Comparison and Discussion - Scenario 2

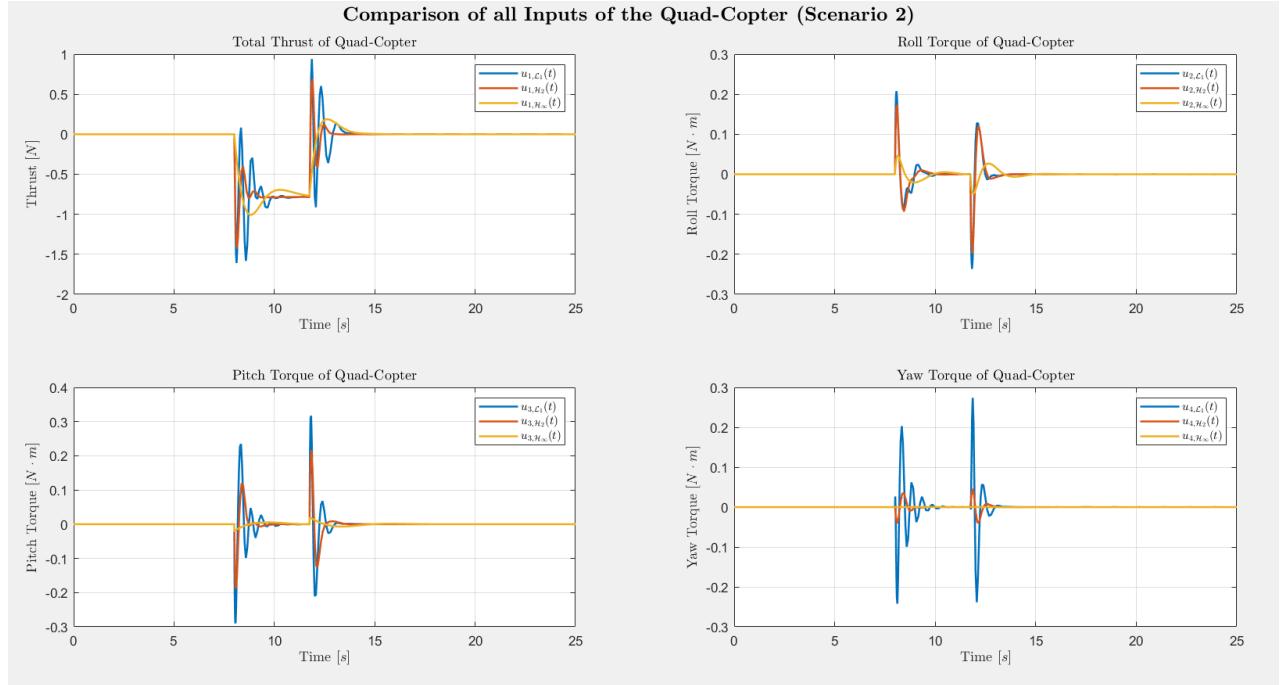


Figure 4.33: Comparative plot of control inputs generated by the controllers (Scenario 2)

The \mathcal{H}_∞ controller fails to maintain position, allowing the quadcopter to drift nearly 1.8 meters due to an overly “soft” response. In contrast, the other two controllers are able to hold the drone almost in place, limiting the position error to less than 20 cm.

The crucial difference lies in how the two controllers achieve this.

The \mathcal{H}_2 controller applies a steady and "smooth" thrust that perfectly balances the wind, whereas the \mathcal{L}_1 controller achieves the same result by "spraying" the motors with violent oscillations (chattering).

In conclusion, the \mathcal{H}_2 controller is the best in this scenario as well, providing the same millimeter-level precision as the \mathcal{L}_1 controller while maintaining smooth operation that is safe for the hardware.

Chapter 5

Multi-Objective Control

In the previous chapters, single-objective controllers were analyzed, focusing primarily on position tracking or disturbance rejection. However, in practical quadrotor applications, multiple objectives often need to be satisfied simultaneously, such as maintaining precise position, minimizing control effort, ensuring smooth actuator behavior, and guaranteeing robustness under uncertainties.

Multi-objective control strategies provide a systematic framework to balance these competing goals, allowing the designer to prioritize or trade off different performance criteria according to mission requirements.

5.1 $\mathcal{H}_\infty/\mathcal{H}_2$ Control

This first multi-objective control combines \mathcal{H}_∞ and \mathcal{H}_2 objectives aiming to balance robustness and performance simultaneously. Two design approaches are considered:

- **Scalarization:** the objectives are combined into a single cost function, weighted to reflect their relative importance.
- **ϵ -Constrained:** one objective is optimized while the other is treated as a constraint with a specified tolerance ϵ , allowing for precise control over the trade-off between performance and robustness.

5.1.1 Scalarization Approach

The objective of this approach is, given 2 scalars $a, b > 0$, to find:

$$K_{\mathcal{H}_\infty/\mathcal{H}_2}^* = \min_{K \in \mathbb{S}} a \|T_\infty\|_{\mathcal{H}_\infty} + b \|T_2\|_{\mathcal{H}_2}$$

The controller gain $K_{\mathcal{H}_\infty/\mathcal{H}_2}^*$ is computed by solving the following Linear Matrix Inequality (LMI) formulation:

$$\begin{aligned} \begin{bmatrix} X^* & Y^* \end{bmatrix} = \arg \min_{X, Y, Q} \quad & a\gamma + b\nu \quad (\text{given } a, b > 0) \\ \text{s.t.} \quad & \begin{bmatrix} (AX + B_1Y) + (AX + B_1Y)^T & B_2 & (C_1X + D_{12}Y)^T \\ B_2^T & -\gamma I_{n_w} & D_{11}^T \\ (C_1X + D_{12}Y) & D_{11} & -\gamma I_{n_z} \end{bmatrix} < 0, \\ & \begin{bmatrix} (AX + B_1Y) + (AX + B_1Y)^T & B_2 \\ B_2^T & -I_{n_w} \end{bmatrix} < 0, \\ & \begin{bmatrix} Q & (C_2X + D_{22}Y) \\ (C_2X + D_{22}Y)^T & X \end{bmatrix} < 0, \\ & X = X^T > 0, \quad Q = Q^T > 0, \quad \text{tr}\{Q\} < \nu^2. \end{aligned}$$

where

$$K_{\mathcal{H}_\infty/\mathcal{H}_2}^* = Y^* X^{*-1}$$

Results and Discussion - Scenario 1

Choosing the parameters $a = 5$ and $b = 3$.

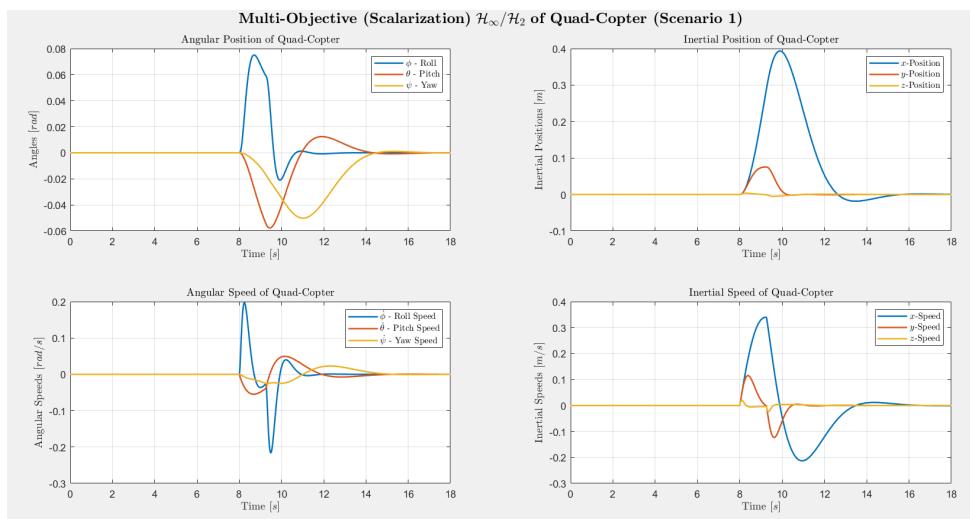


Figure 5.1: States evolution of the quadcopter with $\mathcal{H}_\infty/\mathcal{H}_2$ controller - Scalarization Approach (Scenario 1)

The system prioritizes robustness, allowing the drone to deviate by approximately 40 cm in order to keep the motor inputs smooth and free of aggressive peaks.

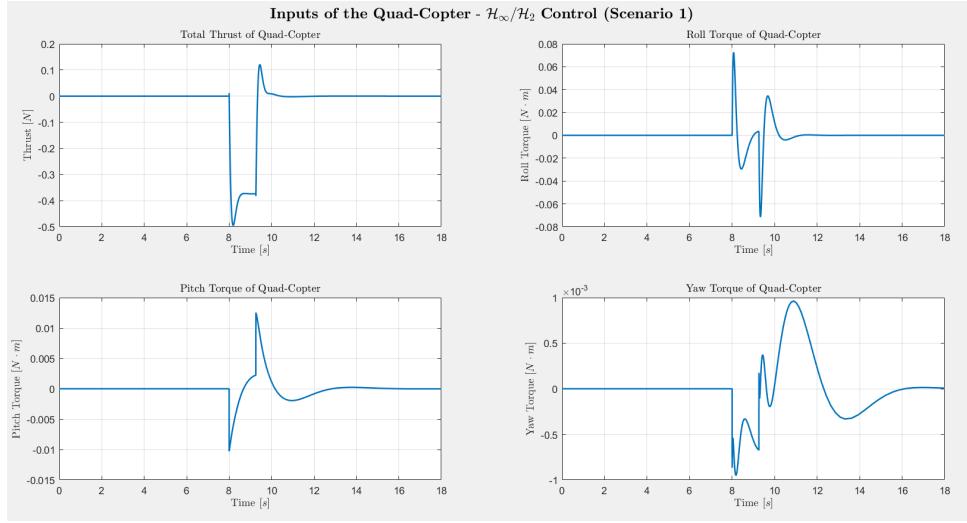


Figure 5.2: Inputs of the quadcopter with $\mathcal{H}_\infty/\mathcal{H}_2$ controller - Scalarization Approach (Scenario 1)

Results and Discussion - Scenario 2

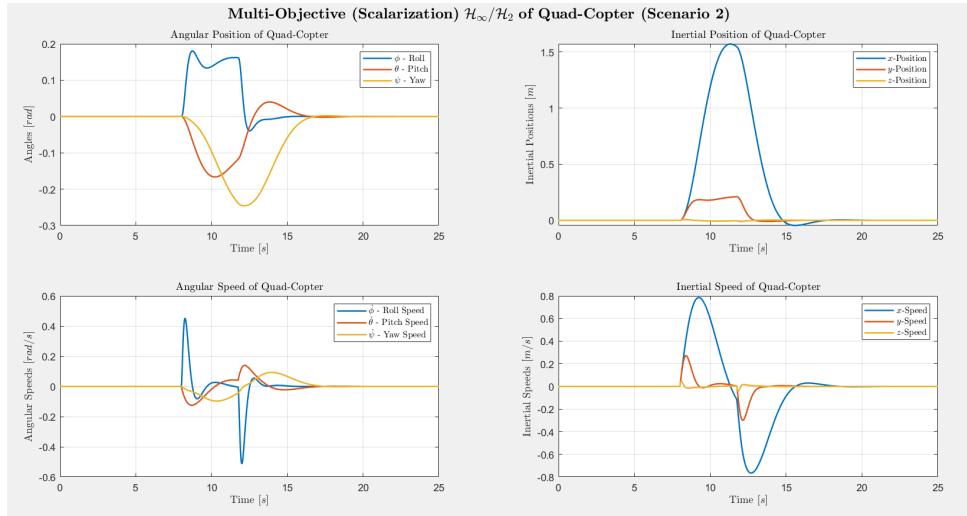


Figure 5.3: States evolution of the quadcopter with $\mathcal{H}_\infty/\mathcal{H}_2$ controller - Scalarization Approach (Scenario 2)

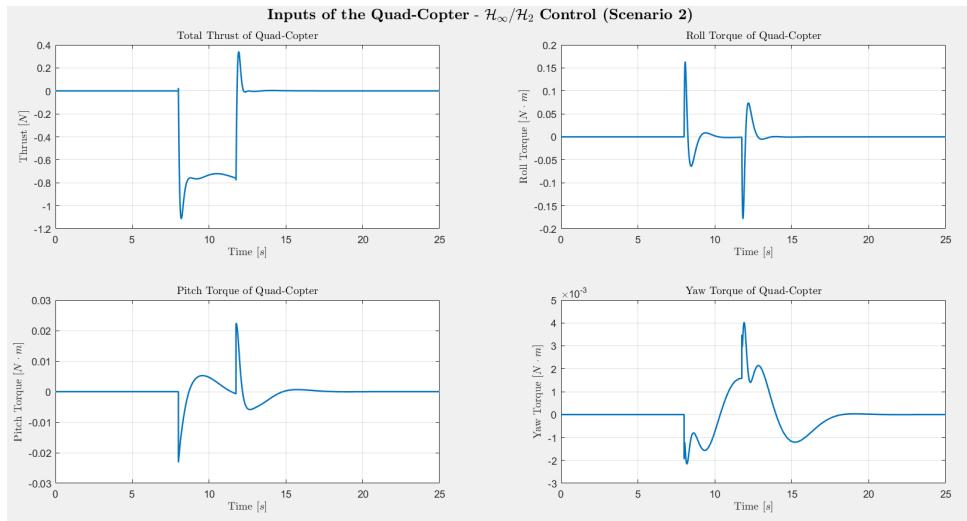


Figure 5.4: Inputs of the quadcopter with $\mathcal{H}_\infty/\mathcal{H}_2$ controller - Scalarization Approach (Scenario 2)

The system successfully converges, as confirmed by the closed-loop eigenvalues, which are:

$$\lambda_{\infty,2_{sc}} = \begin{bmatrix} -1.79e+03 + 0.00i \\ -45.01 + 0.00i \\ -8.22 + 7.87i \\ -8.22 - 7.87i \\ -10.75 + 3.11i \\ -10.75 - 3.11i \\ -0.82 + 1.00i \\ -0.82 - 1.00i \\ -1.46 + 0.02i \\ -1.46 - 0.02i \\ -2.61 + 2.87i \\ -2.61 - 2.87i \end{bmatrix}$$

5.1.2 ϵ -Constrained Approach

Given a scalar $a > 0$, the objective is to find:

$$K_{\mathcal{H}_\infty/\mathcal{H}_2}^* = \min_{K \in \mathbb{S}} \|T_\infty\|_{\mathcal{H}_\infty}$$

s. t. $\|T_2\|_{\mathcal{H}_2} < a$

The controller gain $K_{\mathcal{H}_\infty/\mathcal{H}_2}^*$ is computed by solving the following Linear Matrix Inequality (LMI) formulation:

$$\begin{aligned}
\begin{bmatrix} X^* & Y^* \end{bmatrix} = \arg \min_{X, Y, Q} \quad & \gamma \quad (\text{given } a > 0) \\
\text{s.t.} \quad & \begin{bmatrix} (AX + B_1Y) + (AX + B_1Y)^T & B_2 & (C_1X + D_{12}Y)^T \\ B_2^T & -\gamma I_{n_w} & D_{11}^T \\ (C_1X + D_{12}Y) & D_{11} & -\gamma I_{n_z} \end{bmatrix} < 0, \\
& \begin{bmatrix} (AX + B_1Y) + (AX + B_1Y)^T & B_2 \\ B_2^T & -I_{n_w} \end{bmatrix} < 0, \\
& \begin{bmatrix} Q & (C_2X + D_{22}Y) \\ (C_2X + D_{22}Y)^T & X \end{bmatrix} < 0, \\
& \text{tr}\{Q\} < a^2 \\
& X = X^T > 0, Q = Q^T > 0.
\end{aligned}$$

where

$$K_{\mathcal{H}_\infty/\mathcal{H}_2}^* = Y^* X^{*-1}$$

Results and Discussion - Scenario 1

Choosing the parameter $a = 5$, the closed-loop eigenvalues are:

$$\lambda_{\infty,2_{sc}} = \begin{bmatrix} -47.45 + 0.00i \\ -8.20 + 8.04i \\ -8.20 - 8.04i \\ -7.85 + 7.24i \\ -7.85 - 7.24i \\ -6.91 + 4.77i \\ -6.91 - 4.77i \\ -4.22 + 0.00i \\ -1.82 + 2.35i \\ -1.82 - 2.35i \\ -2.86 + 2.88i \\ -2.86 - 2.88i \end{bmatrix}$$

This indicates that the system is stable in closed-loop.

As shown in the Figure (5.5) and (5.6), the controller is able to limit the drone displacement to approximately 12 cm, providing a practical trade-off that approaches the

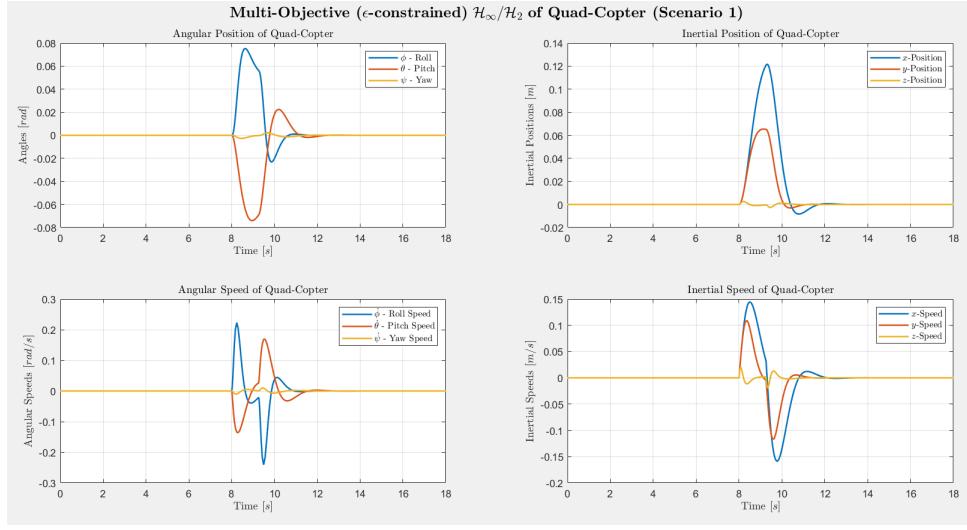


Figure 5.5: States evolution of the quadcopter with $\mathcal{H}_\infty/\mathcal{H}_2$ controller - ϵ -Constrained Approach (Scenario 1)

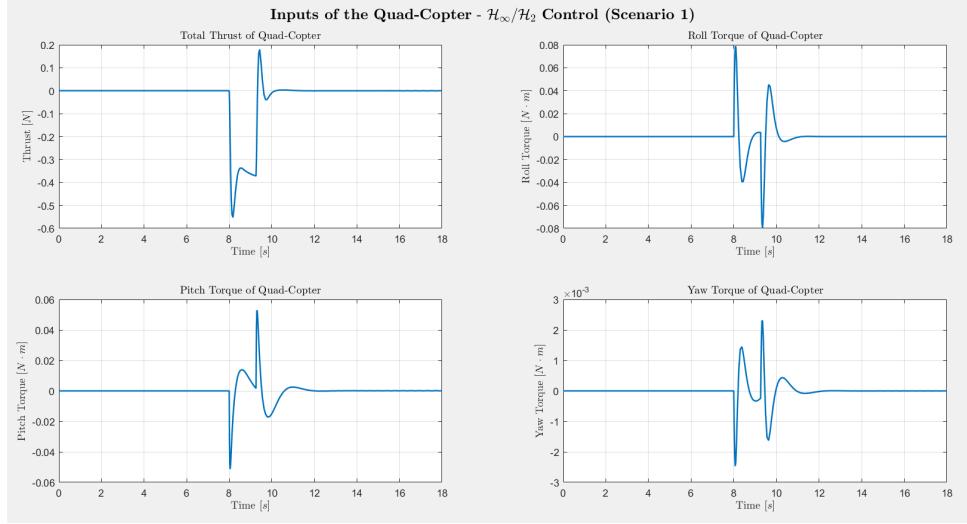


Figure 5.6: Inputs of the quadcopter with $\mathcal{H}_\infty/\mathcal{H}_2$ controller - ϵ -Constrained Approach (Scenario 1)

performance of the \mathcal{H}_2 controller while still enforcing robustness constraints on the control inputs.

Results and Discussion - Scenario 2

Figure (5.7) and (5.8) show that the controller is able to effectively counteract the drift, limiting the position error to approximately 30 cm by applying a steady and decisive torque.

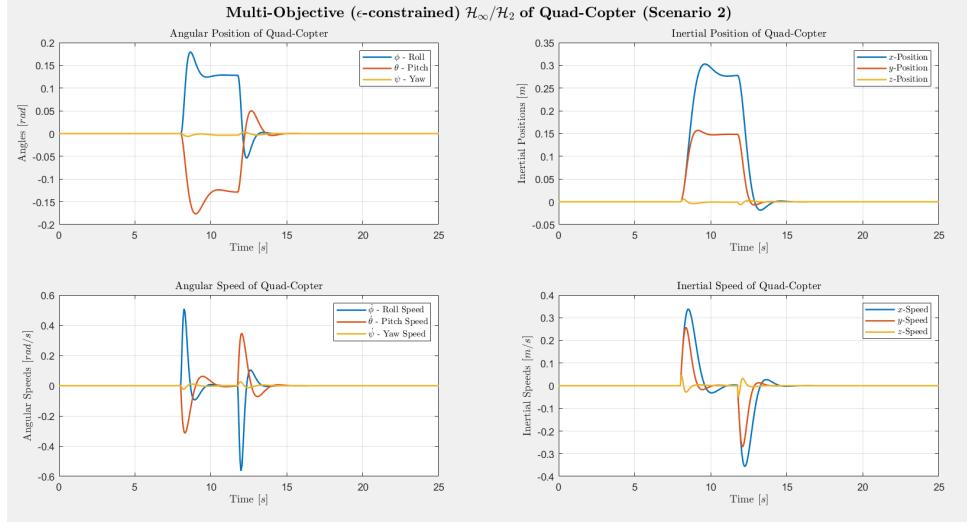


Figure 5.7: States evolution of the quadcopter with $\mathcal{H}_\infty/\mathcal{H}_2$ controller - ϵ -Constrained Approach (Scenario 2)

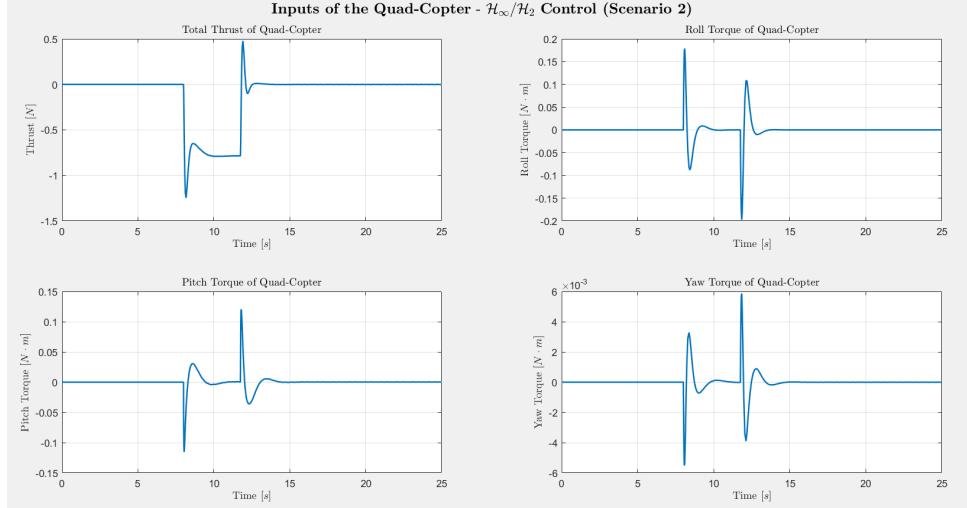


Figure 5.8: Inputs of the quadcopter with $\mathcal{H}_\infty/\mathcal{H}_2$ controller - ϵ -Constrained Approach (Scenario 2)

5.1.3 Controllers Comparison and Discussion

The scalarization approach results in a conservative controller. Its response closely follows that of the \mathcal{H}_∞ controller. In fact, under Scenario 2, the controller allows the drone to drift up to approximately 1.5 m, as the optimization excessively prioritizes stability and robustness over position accuracy.

In contrast, the ϵ -constrained approach proves to be significantly more effective and aggressive. It is able to limit the position error to approximately 30 cm in Scenario 2, approaching the performance of the \mathcal{H}_2 controller. This is achieved by applying the re-

quired control torque without introducing excessive peaks, unlike the behavior observed with the \mathcal{L}_1 controller.

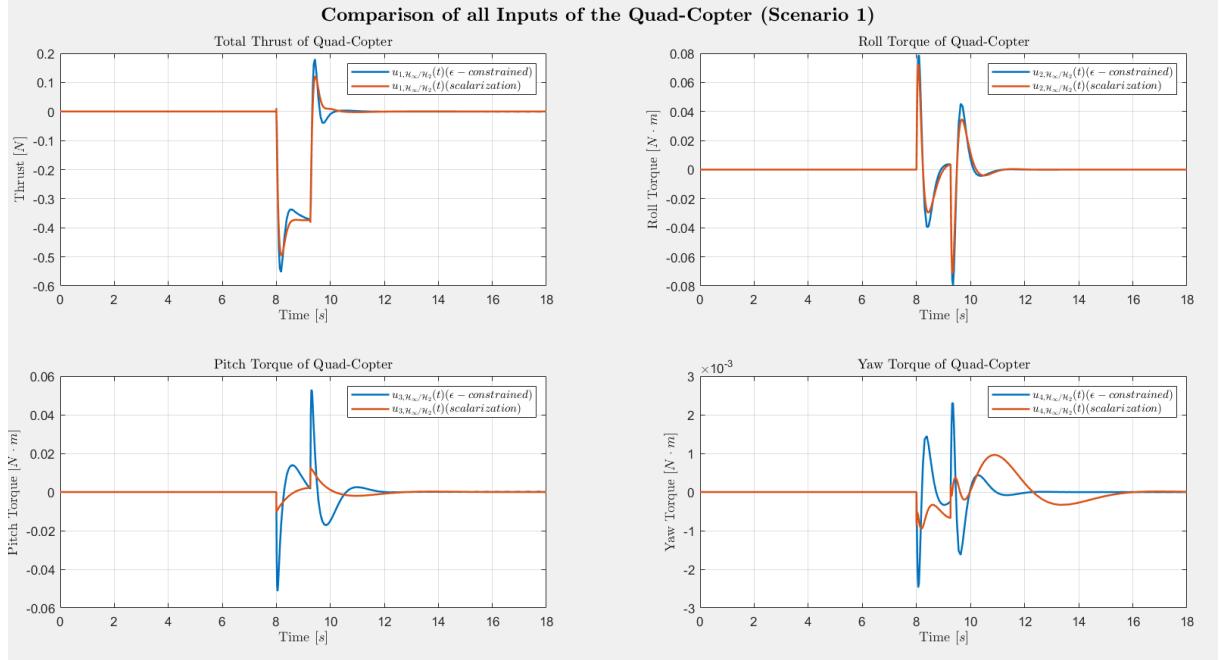


Figure 5.9: Comparative plot of control inputs generated by the controllers (Scenario 1)

Overall, the ϵ -constrained approach emerges as the most suitable solution, as it achieves near- \mathcal{H}_2 precision while preserving the robustness guarantees of the \mathcal{H}_∞ framework.

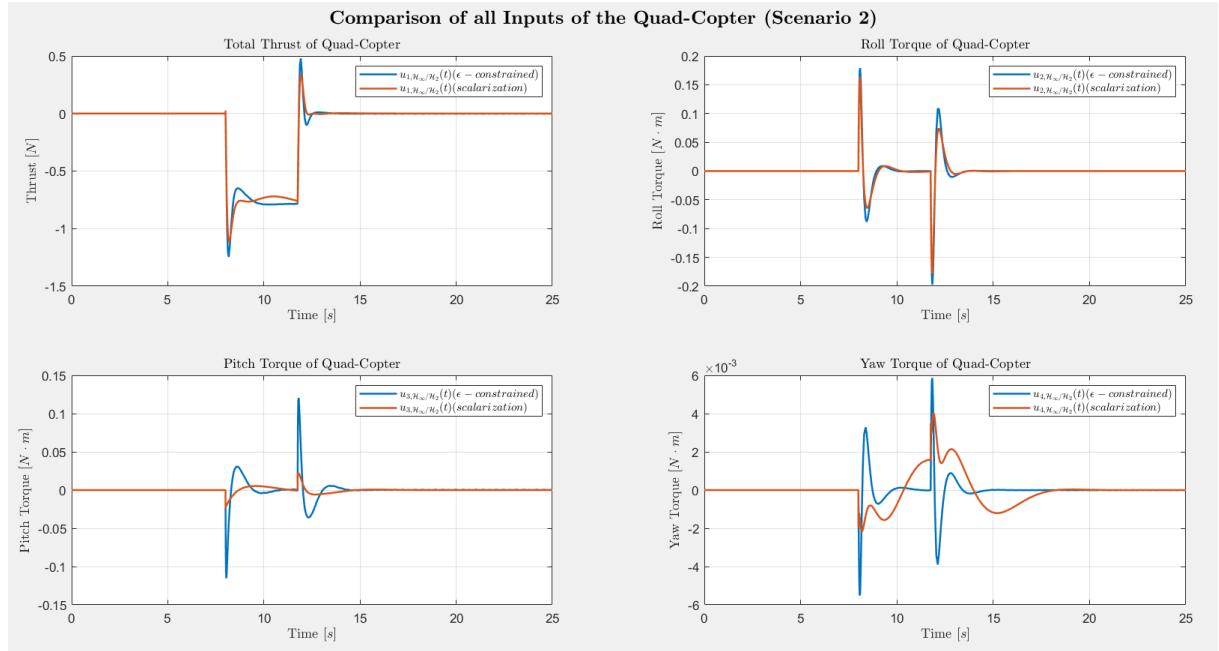


Figure 5.10: Comparative plot of control inputs generated by the controllers (Scenario 2)

5.2 \mathcal{H}_∞ Control with r -stability Constraints

In addition to the $\mathcal{H}_{\text{infinity}}$ constraint, specific regional stability constraints are introduced in order to shape the closed-loop dynamic behavior:

- A **decay rate** constraint shifts the eigenvalues to the left half-plane by at least α ensuring a minimum convergence speed.
- A **conic sector** constraint defined by the angle θ limits the oscillatory behavior of the closed-loop dynamics.
- A **disk** constraint with radius r bounds the eigenvalues within a specified region, preventing excessively fast or aggressive dynamics.

These constraints collectively ensure that the closed-loop system remains stable, robust, and well-damped, while avoiding overly aggressive control actions. The resulting controller prioritizes robustness and smooth actuator behavior, making it suitable for scenarios where stability guarantees are critical.

The controller gain

$$K_{\mathcal{H}_\infty/\mathcal{H}_r} = \min_{K \in RS} \|T_\infty\|_{\mathcal{H}_\infty}$$

is computed by solving a set of LMIs that jointly enforce \mathcal{H}_∞ performance, Lyapunov stability, and regional eigenvalue constraints on the closed-loop system.

Therefore, given r , θ and α

$$\begin{aligned} \begin{bmatrix} X^* & Y^* \end{bmatrix} &= \arg \min_{X, Y} \gamma \\ \text{s.t. } & \begin{bmatrix} (AX + B_1Y) + (AX + B_1Y)^T & B_2 & (C_1X + D_{12}Y)^T \\ B_2^T & -\gamma I_{n_w} & D_{11}^T \\ (C_1X + D_{12}Y) & D_{11} & -\gamma I_{n_z} \end{bmatrix} < 0, \\ & \begin{bmatrix} -rX & (AX + B_1Y) \\ (AX + B_1Y)^T & -rX \end{bmatrix} < 0, \\ & (AX + BY)^T + (AX + BY)^T + 2\alpha X < 0 \\ & \begin{bmatrix} \sin(\theta) ((A + BY) + (A + BY)^T) & \cos(\theta) ((A + BY) - (A + BY)^T) \\ \cos(\theta) ((A + BY) - (A + BY)^T) & \sin(\theta) ((A + BY) + (A + BY)^T) \end{bmatrix} < 0, \\ & X = X^T > 0, \gamma > 0. \end{aligned}$$

where

$$K_{\mathcal{H}_\infty/\mathcal{H}_r} = Y^* X^{*-1}$$

Results and Discussion - Scenario 1

Computing the closed-loop eigenvalues

$$\lambda_{\infty,r} = \begin{bmatrix} -19.10 + 0.00i \\ -11.95 + 0.00i \\ -2.04 + 1.02i \\ -2.04 - 1.02i \\ -0.95 + 0.67i \\ -0.95 - 0.67i \\ -0.55 + 0.15i \\ -0.55 - 0.15i \\ -2.62 + 0.00i \\ -2.28 + 0.00i \\ -1.72 + 0.00i \\ -1.54 + 0.00i \end{bmatrix}$$

It can be verified that they satisfy the imposed constraints and remain within the prescribed LMI region.

Choosing $\alpha = 0.5$, $\theta = \frac{\pi}{4}$ and $r = 20$:

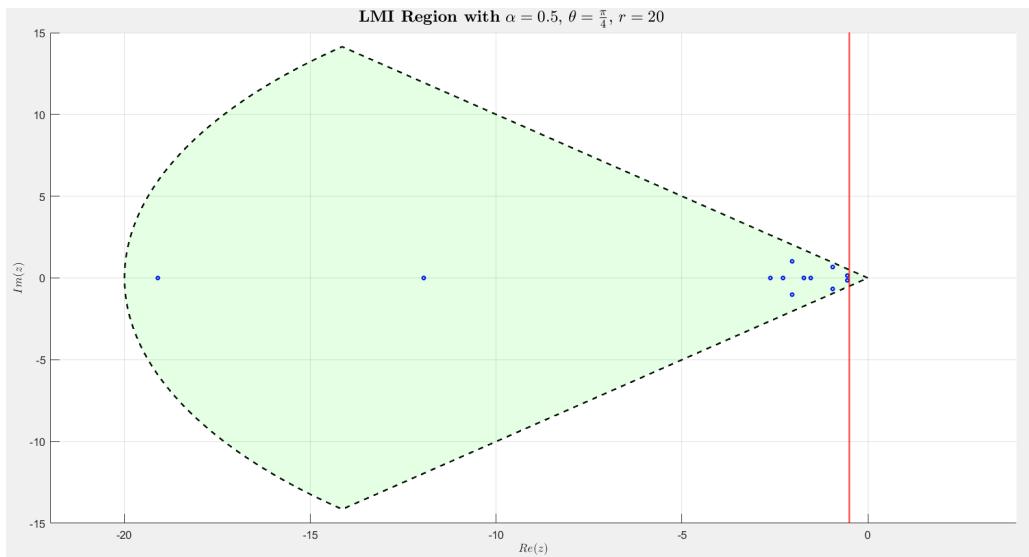


Figure 5.11: LMI region with closed loop eigenvalues

As shown in the Figure (5.12) and (5.13), due to these limitations, in order to maintain this “smoothness” and avoid abrupt motions, the system is unable to generate the aggressive force required to immediately counteract the wind, resulting in a large drift of approximately 45 cm.

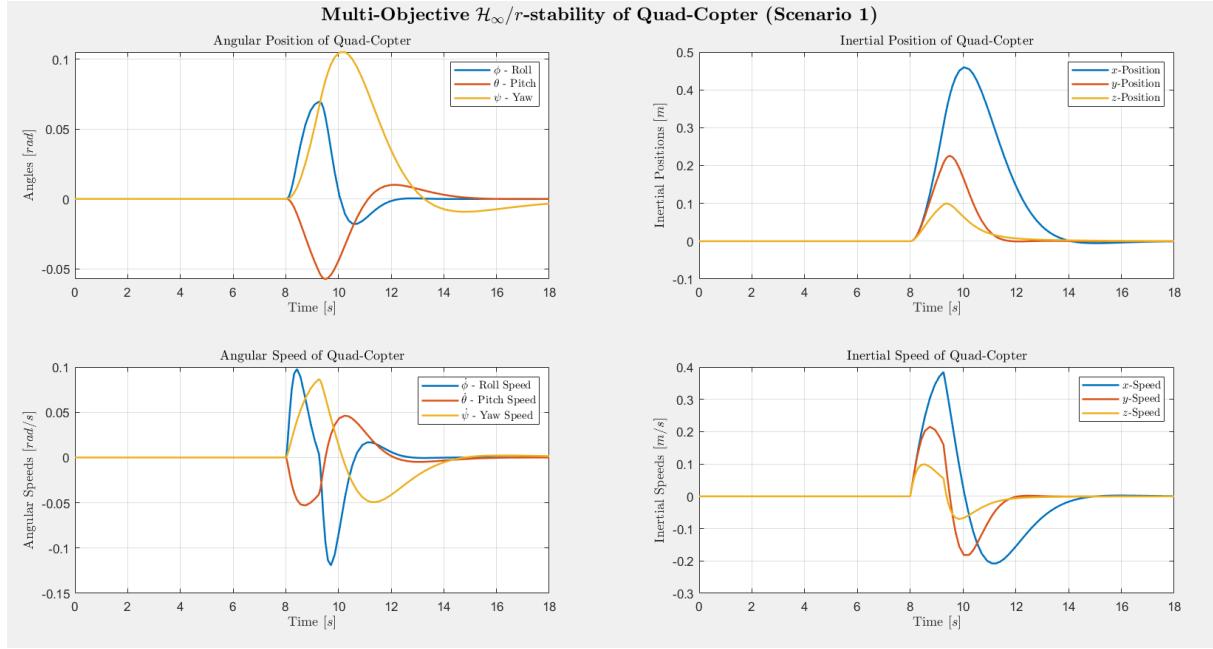


Figure 5.12: States evolution of the quadcopter with \mathcal{H}_∞/r -stability controller (Scenario 1)

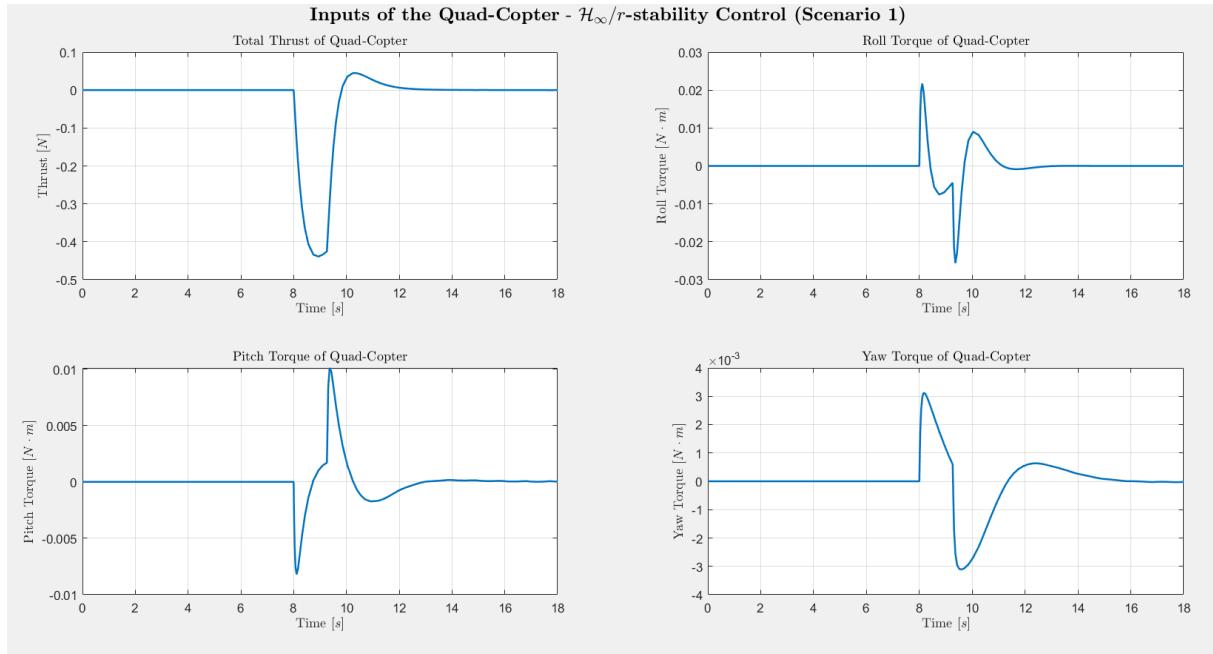


Figure 5.13: Inputs of the quadcopter with \mathcal{H}_∞/r -stability controller (Scenario 1)

Results and Discussion - Scenario 2

The strictness of the geometric constraints drastically limits performance.

The reaction speed limit ($r = 20$) prevents the motors from promptly counteracting the sustained thrust, resulting in a large drift of nearly 3 meters.

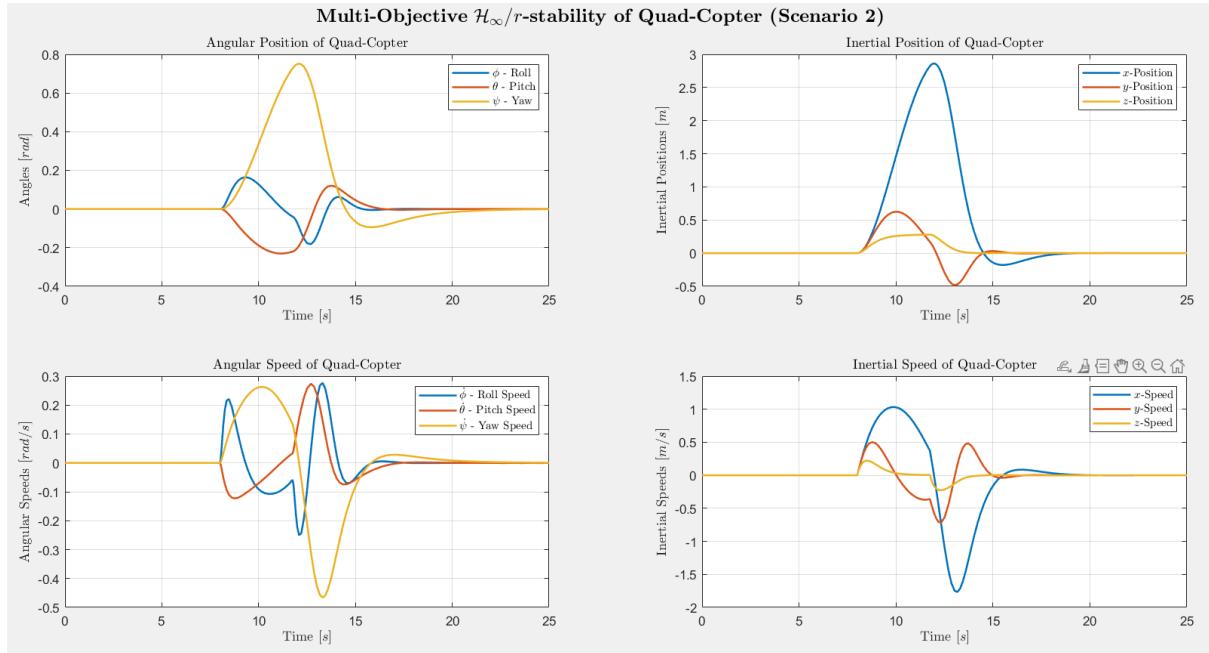


Figure 5.14: States evolution of the quadcopter with \mathcal{H}_∞/r -stability controller (Scenario 2)

Although the motion remains smooth and free of oscillations thanks to the angular constraint (θ), the controller is too “slow” and permissive, proving completely unable to handle a sustained environmental disturbance in order to maintain smooth control commands

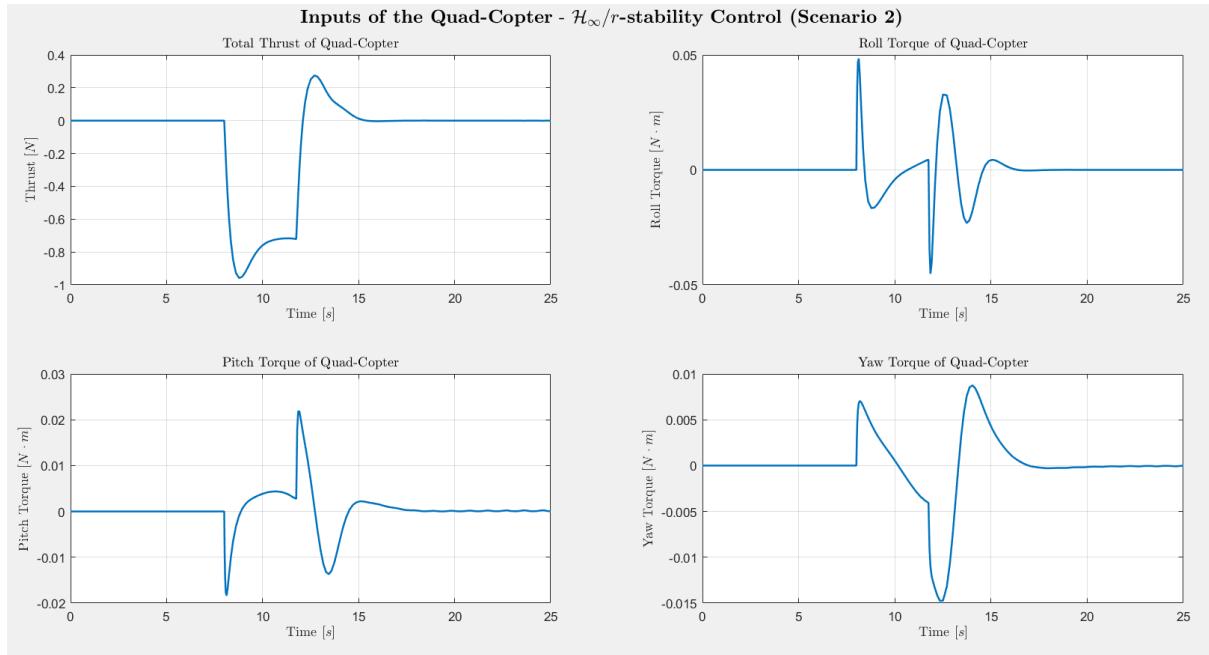


Figure 5.15: Inputs of the quadcopter with \mathcal{H}_∞/r -stability controller (Scenario 2)

Chapter 6

LPV Control and Gain Scheduling

In real aeronautical systems, the physical parameters of the mathematical model are not known with absolute accuracy and may vary over time due to operational and structural factors. These variations inevitably lead to discrepancies between the model adopted during the control design phase and the actual behavior of the real system, making it necessary to assess the robustness of the implemented control strategies.

In the case of a quadcopter, the mass represents one of the most critical parameters, as it directly affects the translational dynamics and the compensation of the gravitational force. In particular, the mass value may vary due to:

1. transportation of different payloads;
2. battery consumption during flight;
3. manufacturing tolerances and assembly variations;
4. approximations in the parameter identification process.

For these reasons, this chapter analyzes the behavior of the control system in the presence of parametric uncertainty on the mass, while keeping the controller structure designed on the nominal model unchanged. The objective is to evaluate the robustness and performance of the proposed control approaches, highlighting the effects of mass variations on the system dynamics and on the control action.

In particular, three control strategies are considered and compared:

- an \mathcal{H}_∞ controller, designed to guarantee robustness against model uncertainties and external disturbances;
- an \mathcal{H}_2 , aimed at optimizing performance in terms of energy and transient response;

- an \mathcal{L}_1 , characterized by strong uncertainty compensation capabilities at the cost of increased actuator activity.

Considering the following mass uncertainty interval:

$$m \in [m_{min} \quad m_{max}]$$

where $m_{min} = 0.7$ [kg] and $m_{max} = 1$ [kg], the system will be **Linear Parameter-Varying** (LPV), that is:

$$\begin{cases} \dot{x}(t) = A(m(t))x(t) + B_1(m(t))u(t) + B_2(m(t))\omega(t) \\ y(t) = C(m(t))x(t) + D(m(t))u(t) \end{cases}$$

where $m(t)$ is the mass that can be measured on-line.

In the case of the quadcopter, the system will be:

$$\begin{cases} \dot{x}(t) = Ax(t) + B_1(m(t))u(t) + B_2(m(t))\omega(t) \\ y(t) = Cx(t) \end{cases}$$

The mass of the quadcopter, which is the critical parameter affecting the system dynamics, was modeled as a time-varying signal in the Simulink environment.

At specific time instants, the mass changes abruptly to represent different payload conditions, as illustrated in Figure (6.1)

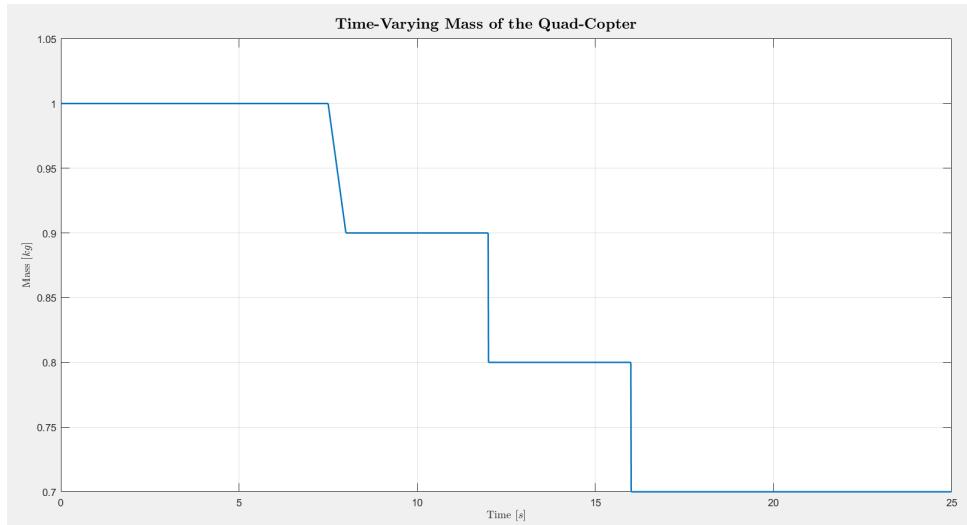


Figure 6.1: Time-Varying Mass signal

6.1 Implementation

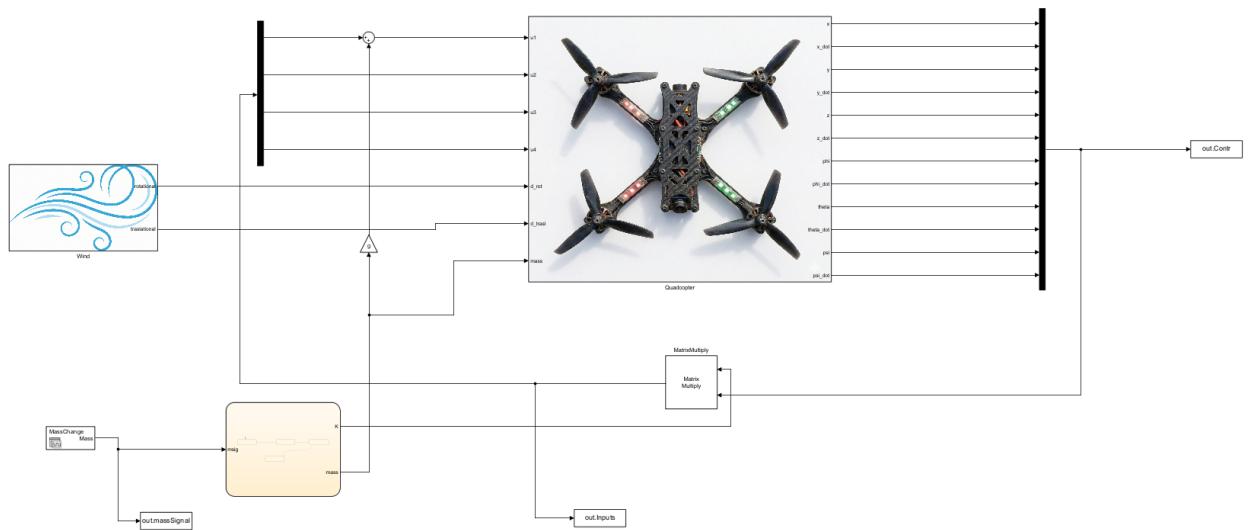


Figure 6.2: Simulink Block Diagram with Gain Scheduling

The logic for adapting the controller is implemented via **Stateflow Chart**.

As illustrated in the Figure (6.3), this block acts as a *finite state machine*.

This block continuously monitors the current mass signal; when the mass signal matches a predefined value (the label on the edges), the logic transitions to the corresponding state. As output, the scheduler updates the control gain matrix K_i to the precomputed optimal value. Each matrix gain K is computed with a robust approach:

- K_1 is computed with a value of $m(t) \in [1 \quad 0.9]$
- K_2 is computed with a value of $m(t) \in [0.9 \quad 0.8]$
- K_3 is computed with a value of $m(t) \in [0.8 \quad 0.7]$
- K_4 is computed with a value of $m(t)$ constant equal to 0.7

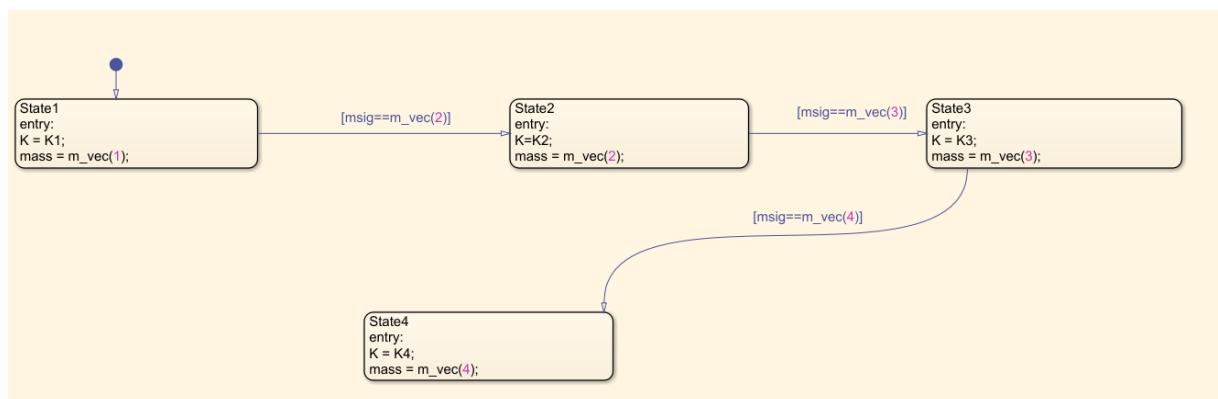


Figure 6.3: Flow Chart for Control Law Execution

6.2 Robust Control

Robust control methodologies are adopted to ensure stability and performance of the closed-loop system in the presence of model uncertainties and external disturbances. In particular, the controllers are designed to explicitly account for parametric variations in the system dynamics, with a focus on mass uncertainty, in order to guarantee reliable operation over a wide range of operating conditions.

Considering the following State Space System

$$\left\{ \begin{array}{l} \dot{x} = A(p)x + B_1(p)u + B_2(p)\omega \\ z_\infty = C_1(p)x + D_{11}(p)\omega + D_{12}(p)u \\ z_2 = C_2(p)x + D_{22}(p)u \\ z_1 = C_3(p)x + D_{31}(p)\omega + D_{32}(p)u \end{array} \right. \quad (6.1)$$

with the controller

$$u = Kx$$

The closed-loop system is given by:

$$\left\{ \begin{array}{l} \dot{x} = [A(p) + B_1(p)K]x + B_2(p)\omega \\ z_\infty = [C_1(p) + D_{12}(p)K]x + D_{11}(p)\omega \\ z_2 = [C_2(p) + D_{22}(p)K]x \\ z_1 = [C_3(p) + D_{32}(p)K]x + D_{31}(p)\omega \end{array} \right. \quad (6.2)$$

In the uncertainty case, the closed loop transfer matrix are not well defined because the depend also by p .

So it is considered the *worst-case* induced norms.

6.2.1 \mathcal{H}_∞ Robust Control

The robust synthesis problem is defined as

$$K_\infty^* = \arg \min_{K \in \mathbb{S}} \|T_\infty\|_{\mathcal{H}_\infty}^{WC}$$

The optimal controller is achieved by solving the following LMI:

$$\begin{aligned} \begin{bmatrix} X^* & Y^* \end{bmatrix} &= \arg \min_{X, Y, \gamma} \gamma \\ \text{s.t. } & \begin{bmatrix} (A_i X + B_{1,i} Y) + (A_i X + B_{1,i} Y)^T & B_{2,i} & (C_{1,i} X + D_{12,i} Y)^T \\ B_{2,i}^T & -\gamma I_{n_w} & D_{11,i}^T \\ (C_{1,i} X + D_{12,i} Y) & D_{11,i} & -\gamma I_{n_z} \end{bmatrix} < 0, \end{aligned}$$

$$X = X^T > 0 \quad \text{with } i = 1, \dots, l.$$

The optimal \mathcal{H}_∞ optimal controller gain is given by

$$K_\infty^* = Y^* X^{*-1}$$

6.2.2 \mathcal{H}_2 Robust Control

The robust synthesis problem is defined as

$$K_2^* = \arg \min_{K \in \mathbb{S}} \|T_2\|_{\mathcal{H}_2}^{WC}$$

The optimal controller is achieved by solving the following LMI:

$$\begin{aligned} \begin{bmatrix} X^* & Y^* \end{bmatrix} &= \arg \min_{X, Y, Q} \text{tr}\{Q\} \\ \text{s.t. } & \begin{bmatrix} (A_i X + B_{1,i} Y) + (A_i X + B_{1,i} Y)^T & B_{2,i} \\ B_{2,i}^T & -I_{n_w} \end{bmatrix} < 0, \\ & \begin{bmatrix} Q & (C_{2,i} X + D_{22,i} Y) \\ (C_{2,i} X + D_{22,i} Y)^T & X \end{bmatrix} < 0, \\ & X = X^T > 0, \quad Q = Q^T > 0 \quad \text{with } i = 1, \dots, l. \end{aligned}$$

The optimal \mathcal{H}_2 optimal controller gain is given by

$$K_2^* = Y^* X^{*-1}$$

6.2.3 \mathcal{L}_1 Robust Control

The robust synthesis problem is defined as

$$K_1^* = \arg \min_{K \in \mathbb{S}} \|T_1\|_{\mathcal{H}_1}^{WC}$$

The optimal controller is achieved by solving the following LMI:

$$\begin{aligned} \begin{bmatrix} X^* & Y^* \end{bmatrix} &= \arg \min_{X, Y, \mu, \zeta} \zeta \\ \text{s.t. } & \begin{bmatrix} (A_i X + B_{1,i} Y) + (A_i X + B_{1,i} Y)^T + \lambda X & B_{2,i} \\ B_{2,i}^T & -\mu I_{n_w} \end{bmatrix} < 0, \\ & \begin{bmatrix} \lambda X & 0 & (C_{3,i} X + D_{32,i} Y)^T \\ 0 & (\zeta - \mu) I_{n_w} & D_{31,i}^T \\ (C_{3,i} X + D_{32,i} Y) & D_{31,i} & \zeta I_{n_z} \end{bmatrix} < 0, \\ & X = X^T > 0, \quad \mu > 0, \quad (\lambda > 0 \text{ fixed and } i = 1, \dots, l) \end{aligned}$$

The optimal \mathcal{L}_1 optimal controller gain is given by

$$K_1^* = Y^* X^{*-1}$$

6.3 Results and Discussion

In these simulations, a comparison is carried out between a robust static controller and a robust dynamic controller, implemented according to a Stateflow-based logic.

Each controller in the dynamic scheme is designed using robust control techniques. In particular:

1. K_1 is computed is computed by considering the system matrices obtained from the linearized model evaluated at $\begin{bmatrix} 1 & 0.9 \end{bmatrix}$
2. K_2 is computed is computed by considering the system matrices obtained from the linearized model evaluated at $\begin{bmatrix} 0.9 & 0.8 \end{bmatrix}$
3. K_3 is computed is computed by considering the system matrices obtained from the linearized model evaluated at $\begin{bmatrix} 0.8 & 0.7 \end{bmatrix}$
4. K_4 is computed is computed using a common optimal controller by considering the system matrices obtained from the linearized model evaluated at 0.7

6.3.1 \mathcal{H}_∞ Control - Scenario 1

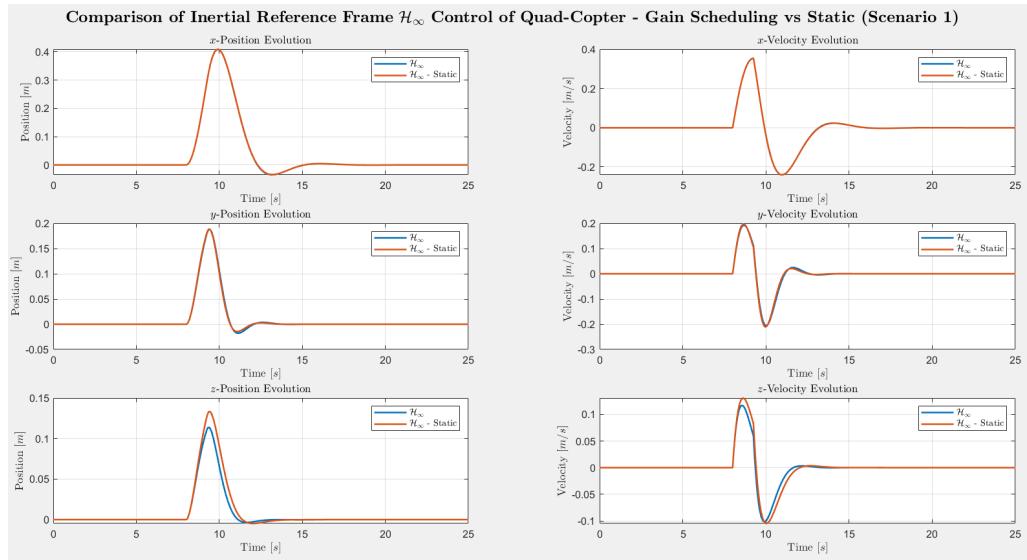


Figure 6.4: Controlled States in the Inertial Reference Frame (Scenario 1)

The dynamic controller exhibits superior performance compared to the static robust controller, achieving a shorter settling time and a reduced overshoot.

Furthermore, in the lateral dynamics, the dynamic control strategy effectively attenuates residual oscillations, resulting in a faster response and a significant reduction of the tracking error.

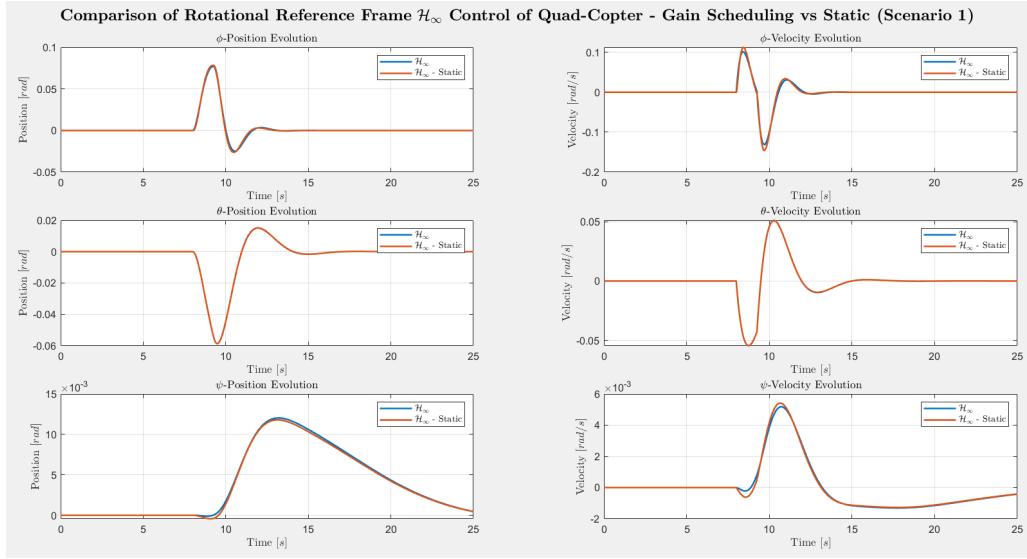


Figure 6.5: Controlled States in the Rotational Reference Frame (Scenario 1)

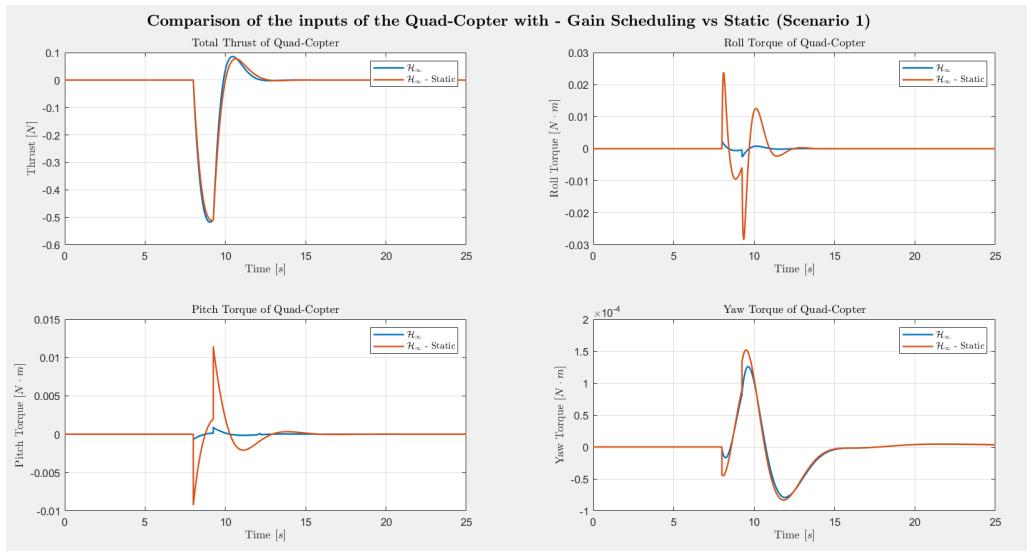


Figure 6.6: Inputs with Robust \mathcal{H}_∞ Controller (Scenario 1)

The dynamic controller optimizes the control effort, indicating a more efficient use of the actuators compared to the static robust controller.

6.3.2 \mathcal{H}_∞ Control - Scenario 2

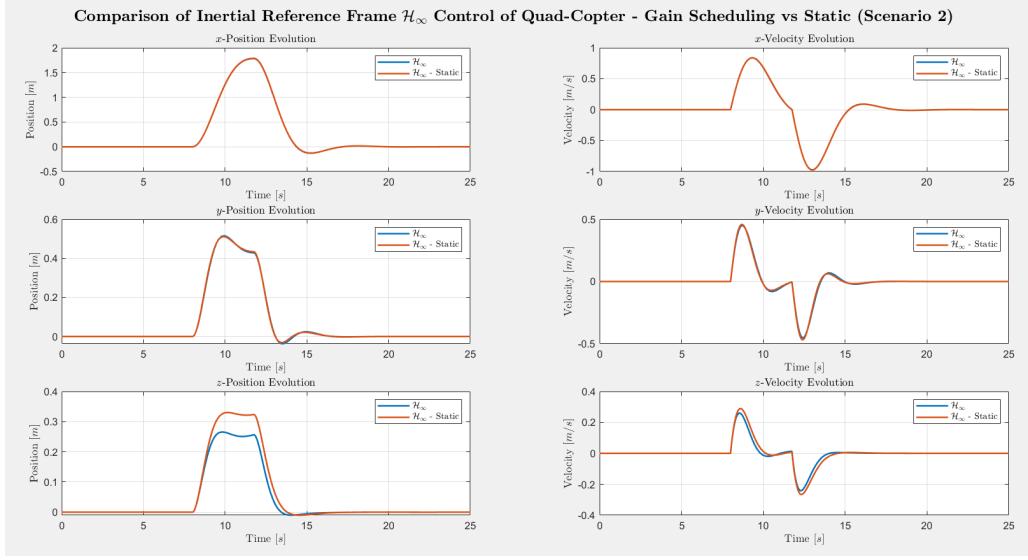


Figure 6.7: Controlled States in the Inertial Reference Frame (Scenario 2)

The comparison shows that the static controller is sufficiently robust for the considered scenario, achieving a performance comparable to that of the more complex gain-scheduled controller .

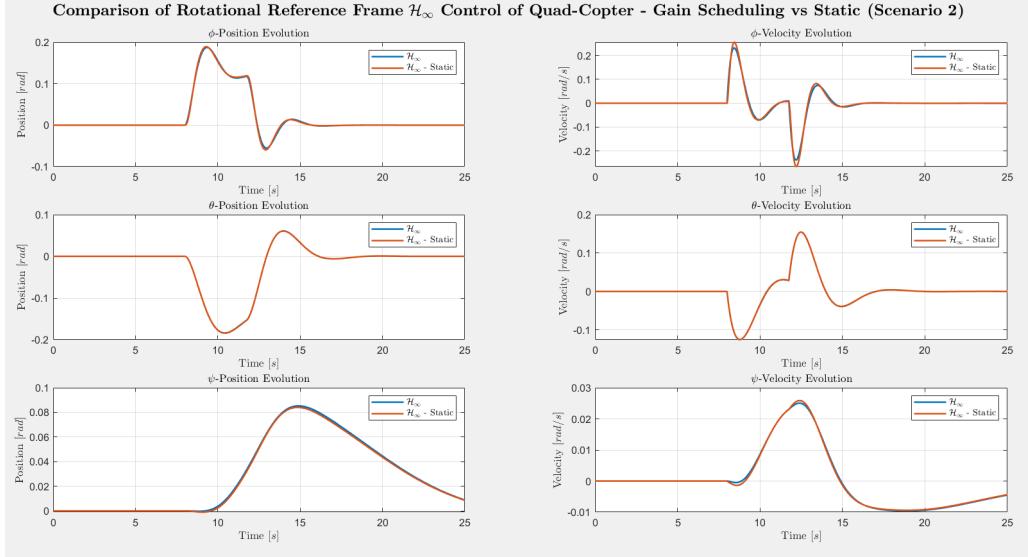


Figure 6.8: Controlled States in the Rotational Reference Frame (Scenario 2)

Moreover, in this specific case, the static controller appears preferable, as it avoids the torque peak observed in the control input around $t = 12$ s.

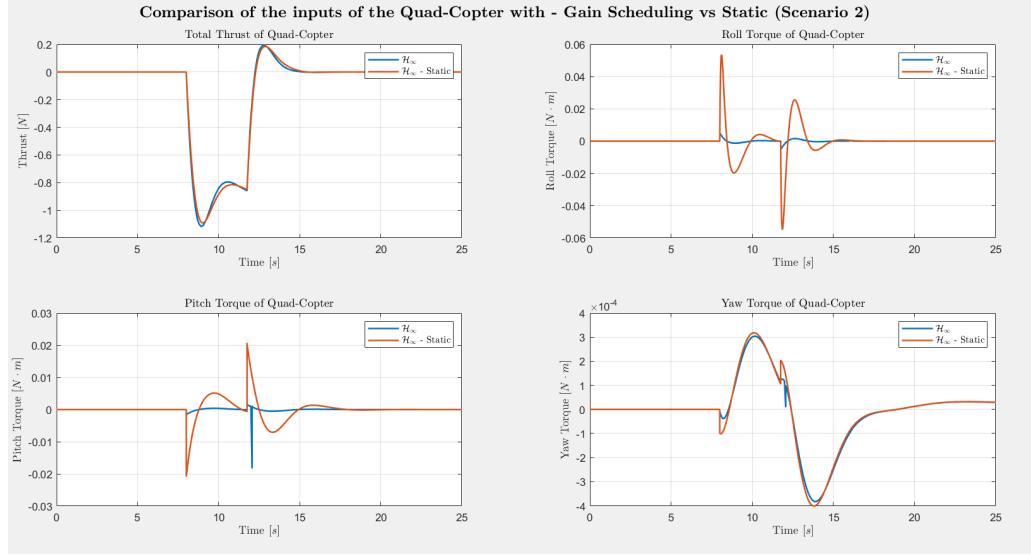


Figure 6.9: Inputs with Robust \mathcal{H}_∞ Controller (Scenario 2)

6.3.3 \mathcal{H}_2 Control - Scenario 1

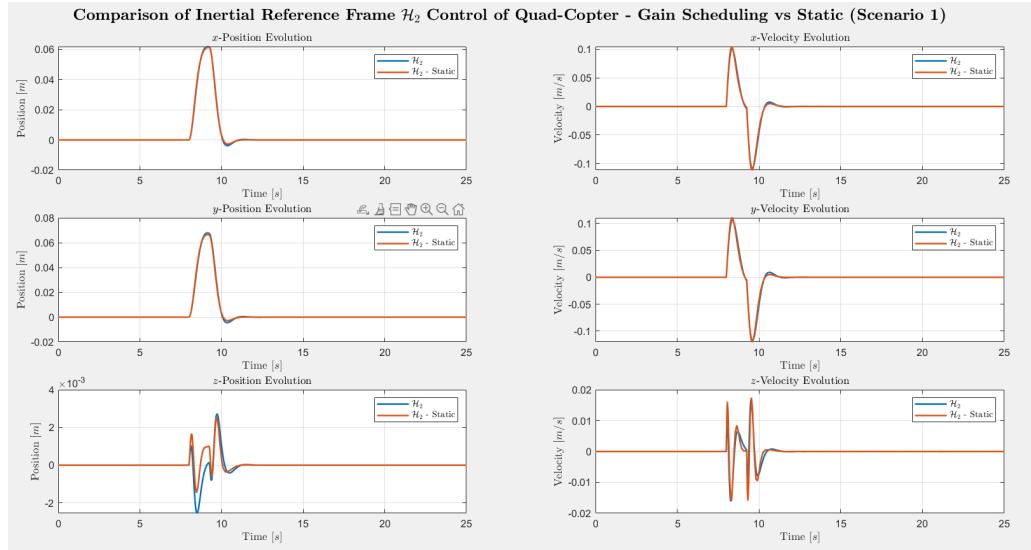


Figure 6.10: Controlled States in the Inertial Reference Frame (Scenario 1)

As shown in the Figure (6.12) the Gain Scheduling strategy is clearly superior. It achieves the desired motion with minimal energy expenditure and reduced mechanical stress, whereas the Static controller wastes resources through aggressive and oscillatory corrective actions.

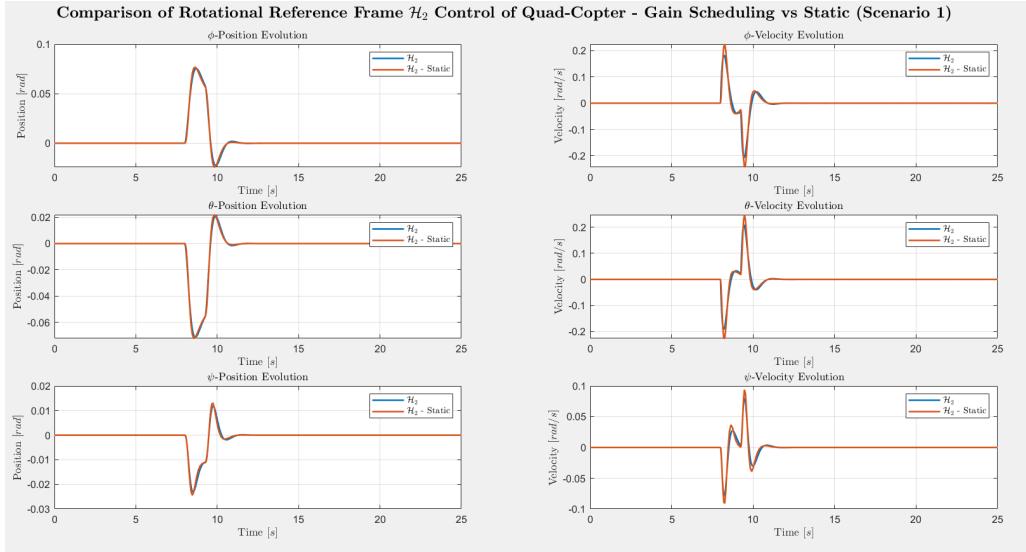


Figure 6.11: Controlled States in the Rotational Reference Frame (Scenario 1)

The Static controller exhibits strong oscillations and significant torque peaks in order to maintain control.

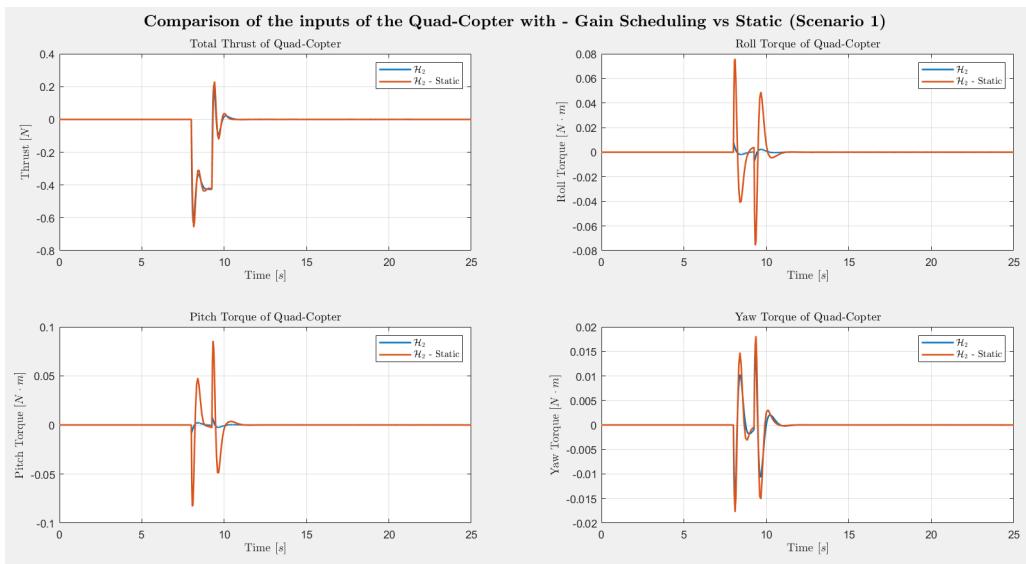


Figure 6.12: Inputs with Robust \mathcal{H}_2 Controller (Scenario 1)

6.3.4 \mathcal{H}_2 Control - Scenario 2

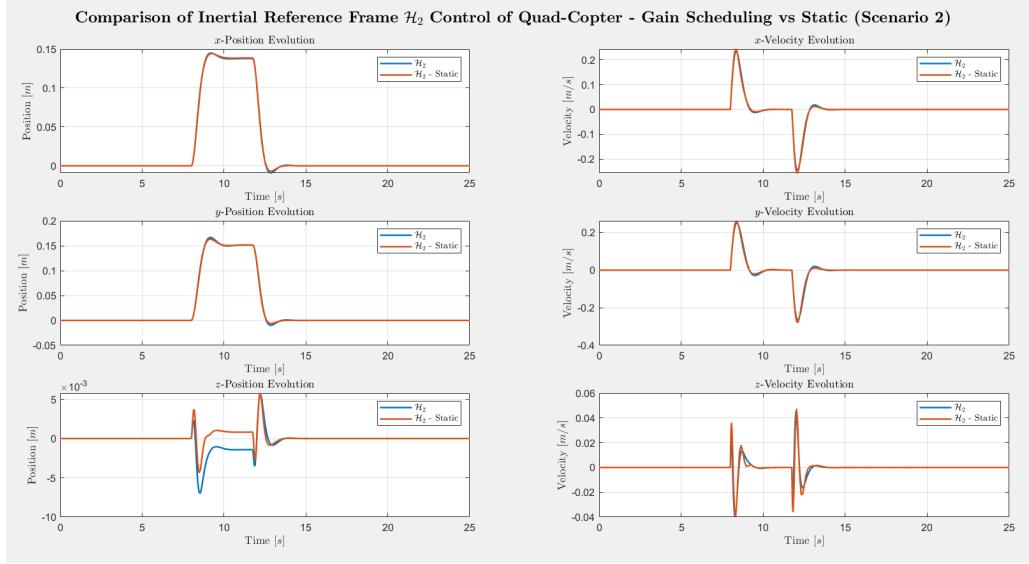


Figure 6.13: Controlled States in the Inertial Reference Frame (Scenario 2)

As in Scenario 1, the Gain Scheduling approach is clearly superior also in this scenario. Although both controllers achieve the same motion (Figures (6.13) and (6.14)), the Static controller does so at the cost of unacceptable actuator stress as shown in Figure (6.15).

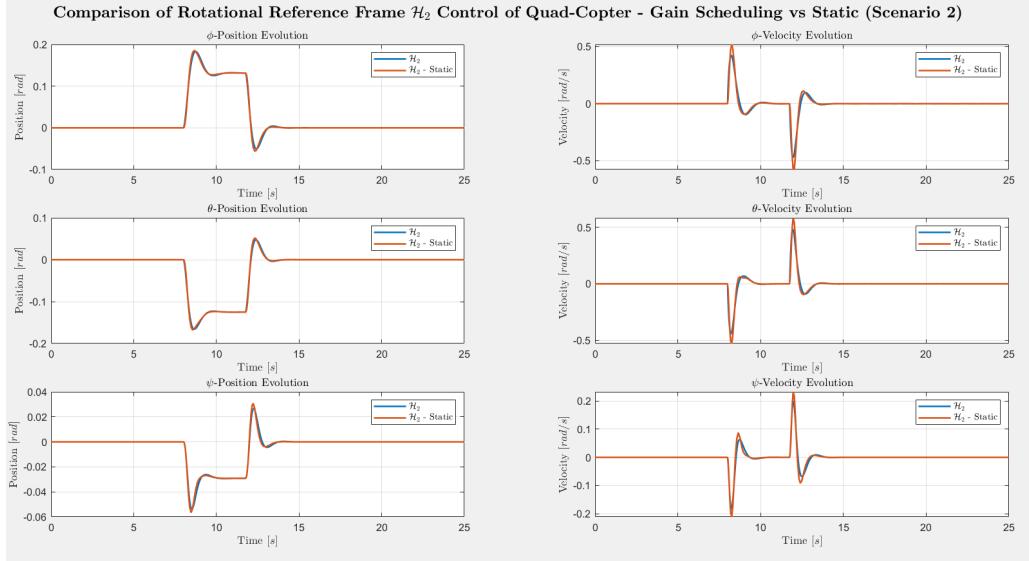


Figure 6.14: Controlled States in the Rotational Reference Frame (Scenario 2)

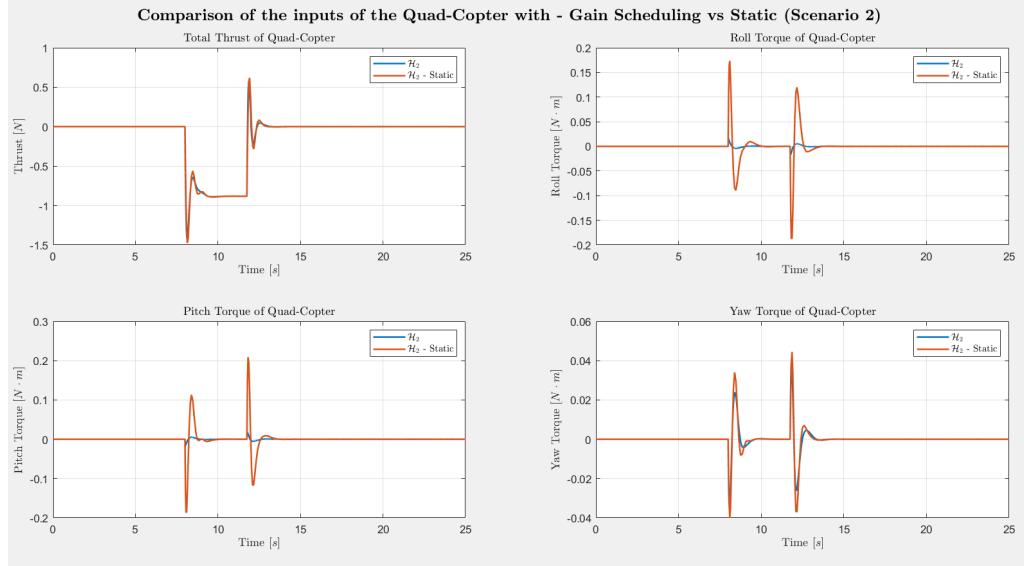


Figure 6.15: Inputs with Robust \mathcal{H}_2 Controller (Scenario 2)

6.3.5 \mathcal{L}_1 Control - Scenario 1

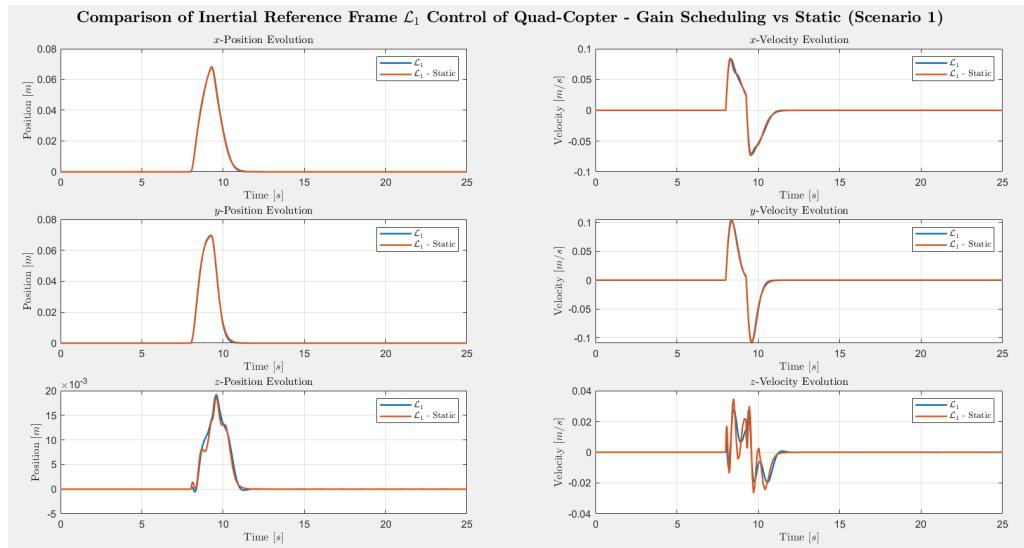


Figure 6.16: Controlled States in the Inertial Reference Frame (Scenario 1)

The Static controller suffers from severe torque peaks and pronounced oscillatory behavior in the interval between 8 and 10 seconds.

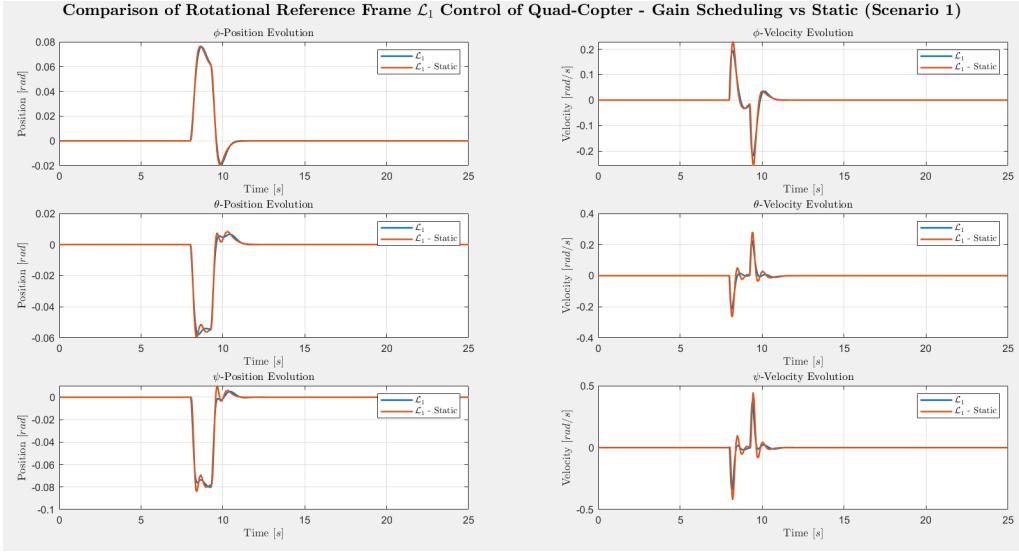


Figure 6.17: Controlled States in the Rotational Reference Frame (Scenario 1)

Also for this control task, the Gain Scheduling strategy is clearly preferable. Although the Static controller guarantees the same trajectory, it does so in a “rough” manner, heavily stressing the actuators through impulsive control commands that are completely avoided by the Gain Scheduling approach.

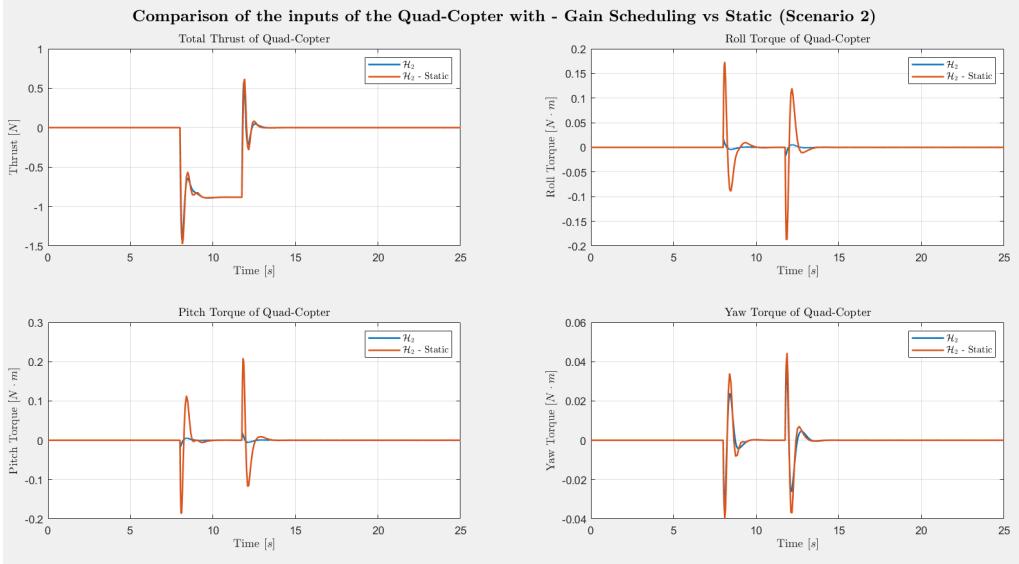


Figure 6.18: Inputs with Robust \mathcal{L}_1 Controller (Scenario 1)

6.3.6 \mathcal{L}_1 Control - Scenario 2

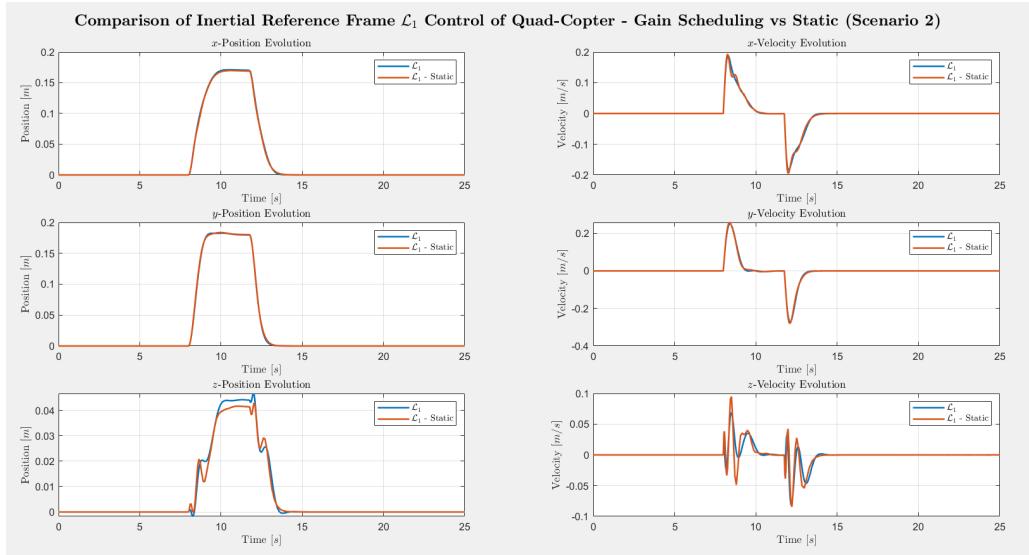


Figure 6.19: Controlled States in the Inertial Reference Frame (Scenario 2)

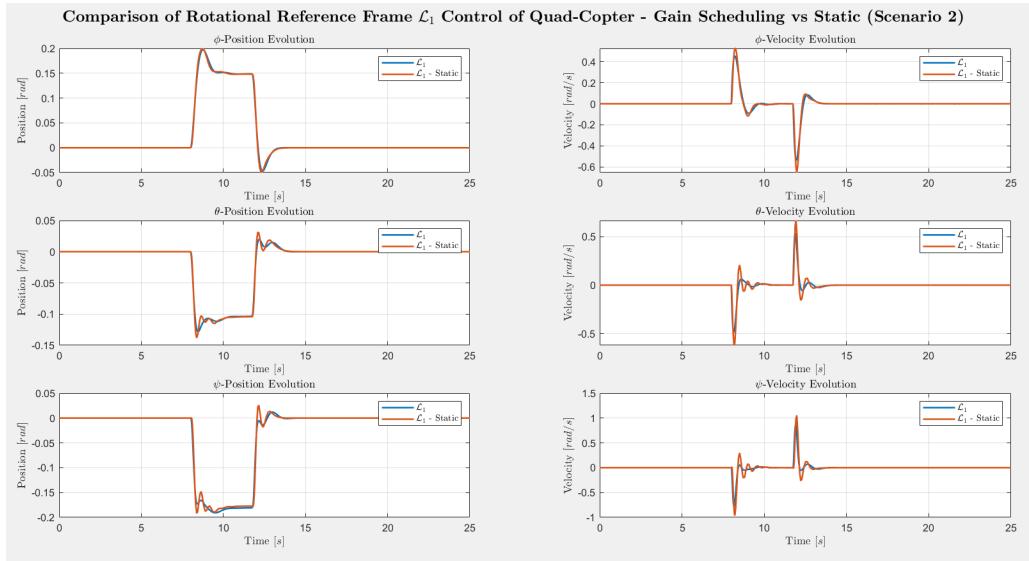


Figure 6.20: Controlled States in the Rotational Reference Frame (Scenario 2)

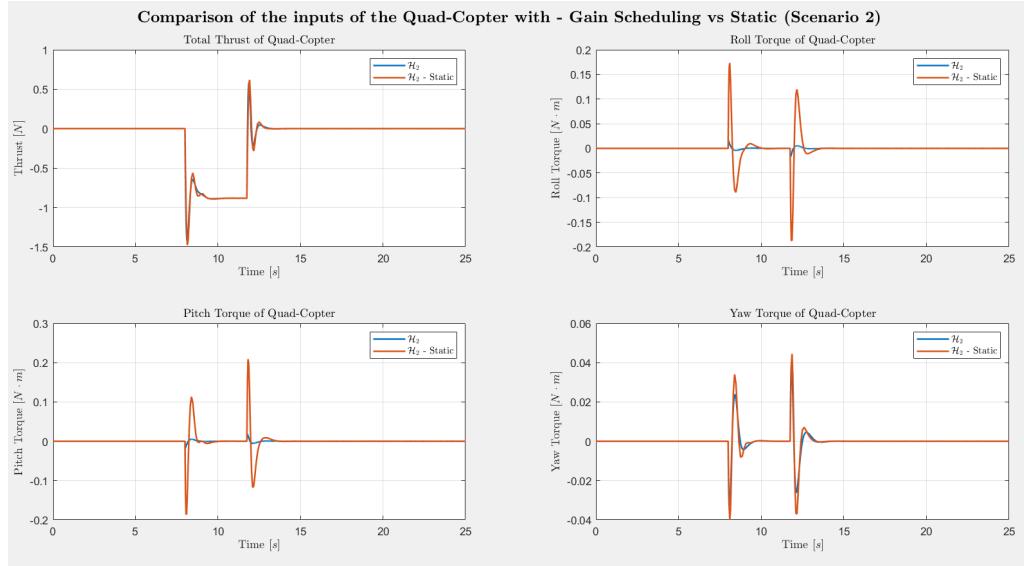


Figure 6.21: Inputs with Robust \mathcal{L}_1 Controller (Scenario 2)

Once again, it is confirmed that the Gain Scheduling approach provides superior performance, consistently outperforming the alternative control strategy.

Chapter 7

Conclusions

This work compared several control strategies for a quadcopter, ranging from classical and optimal controllers to robust and LPV-based approaches, under nominal conditions, external disturbances, and parametric uncertainty.

Under **nominal conditions**, the **static \mathcal{H}_2** controller emerges as the most balanced solution. It provides fast and well-damped responses while making the most efficient use of the actuators, with smooth control inputs and limited peaks. In contrast, the **PID controller**, although effective and simple, requires higher control effort and shows reduced efficiency.

In the presence of external disturbances, the **\mathcal{H}_2 controller** still guarantees good performance but loses robustness. The **\mathcal{H}_∞ controller** ensures the best disturbance rejection and stability margins, at the cost of more conservative behavior and increased actuator usage. The **\mathcal{L}_1 controller** achieves fast compensation but exhibits high-frequency control activity, leading to less efficient actuator utilization.

When **parametric uncertainty** is considered, robust and LPV controllers significantly improve reliability. **Robust \mathcal{H}_∞** provides the highest level of robustness, while **robust \mathcal{H}_2** and **LPV Gain Scheduling** offer a favorable trade-off between performance, robustness, and actuator efficiency when parameter variations are moderate or measurable.