Università degli Studi di Napoli Federico II

Scuola Politecnica e delle Scienze di Base

Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione

Corso di Laurea in Ingegneria dell'Automazione e Robotica

FIELD AND SERVICE ROBOTICS

Homework 3 Report

Student:
Emanuela Varone

Matricola:
P38000284

Contents

1 The octocopter system with co-planar propellers

The goal of this section is to analyze the degrees of freedom and actuation of an octocopter, by deriving its allocation matrix.

1.1 DOFs, Configuration Space Topology and Underactuation

Consider the octocopter in Fig.[1], composed by a rigid body and eight co-planar propellers.

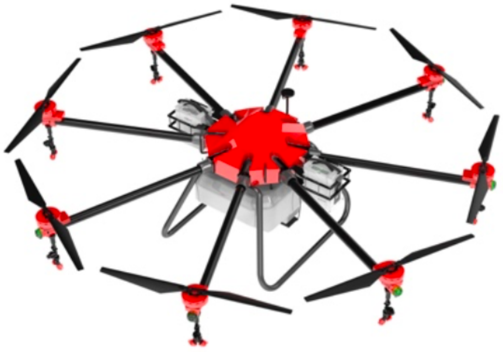


Figure 1: Octocopter

In order to determine analytically the number of DOFs, Grubler's formula was employed:

$$DoFs = m(N - 1 - J) + \sum_{i=1}^{J} f_i \tag{1}$$

It is necessary to distinguish the following two cases:

- **Body fixed on the ground:**

  Due to the spatial mechanism, m = 6. Considering that N = 8 + 1 = 9 (propellers plus the ground), J = 8 (rotational joints) and $\sum_{i=1}^{J} f_i = 8$ (revolute joint gives 1 DoF), the application of Grubler's formula (Eq.1) leads to a number of DoFs of the Octocopter equal to 8. The configuration space topology is:

  $$\textbf{Topology:} \quad S^1 \times \cdots \times S^1 = T^8.$$

- **Flying body:**

  While the UAV is flying, his number of DoFs increases to:

  $$DoFs = 8 + 6 = 14$$

  since, in this condition, the octocopter acquires 6 DoFs (3 translations + 3 rotations) of a planar rigid body. The configuration space topology is:

  $$\textbf{Topology:} \quad R^3 \times S^2 \times S^1 \times T^8.$$

In a flat configuration, while flying, the system is characterized by a number of DoFs equal to 6 and consists of 8 actuators. Although the greater number of actuators than the number of generalized variables might suggest that the system is fully actuated, however, it is possible to prove that the octocopter turns out to be underactuated. In fact, this configuration creates a fundamental limitation: can only apply three rotations and one translation along the vertical direction of the body frame. The allocation matrix for the UAV with the propellers in a flat configuration is $G_q \in \mathbb{R}^{4 \times n}$ (for the octocopter: $G_q \in \mathbb{R}^{4 \times 8}$). The octocopter needs to control 6 DoFs to move completely in 3D (3 for position and 3 for orientation), but with the current configuration it can only directly control 4, resulting in an **underactuated system** ($rank(G_d) < dim(q) = 6$). Having 8 co-planar propellers offers redundancy in the speed distribution between them, but does not solve the underactuation problem since they do not give the possibility to have more real inputs to the system than 4. To achieve complete control of all 6 DoFs, it would be necessary to have either non-coplanar propellers (tilted configurations) or motorised propellers (actively tilted configurations). Only then the allocation matrix would belong to $\mathbb{R}^{6 \times 8}$, allowing full control of three-dimensional motion without the need to tilt the entire drone.

1.2 Allocation Matrix

In the octocopter the control input forces and torques are related to the squared rotor rotational speeds, ω_i^2, through the constant matrix $G_q \in \mathbb{R}^{4 \times 8}$, called **allocation matrix**, as:

$$\begin{bmatrix} u_T \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = G_q \begin{bmatrix} \omega_1^2 \\ \vdots \\ \omega_8^2 \end{bmatrix} \tag{2}$$

In order to derive the allocation matrix, the NED frame was chosen (see Fig.[2]), in which, accordingly to the convention: the x-axis points to the North and is aligned with the arm associated with the first propeller, the y-axis points to the Est and z-axis points Down. Starting from the x-axis, as suggested, the propellers were labeled counter-clockwise.

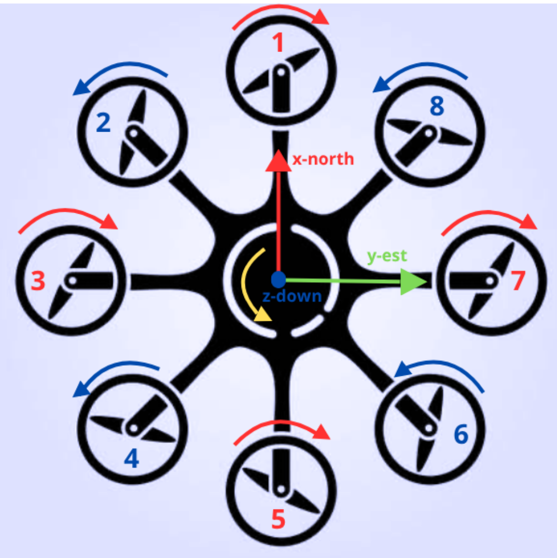


Figure 2: Octocopter NED Frame

As can be seen, since the 8 propellers are equidistant, the angle between two consecutive arms is equal to 45°. In general, the thrust force, T_i, generated by a rotor in free air, far from obstacles with the rotor inflow and outflow air, can be modeled with:

$$T_i = c_T \omega_i^2 \tag{3}$$

where c_T is a constant that can be determined for each rotor combination from static thrust tests. Hence, the total thrust is given by:

$$u_T = \sum_{i=1}^{8} T_i = \sum_{i=1}^{8} c_T \omega_i^2 \tag{4}$$

The reaction torque due to rotor drag acting on the airframe generated by a hovering rotor in free air may also be modeled as:

$$Q_i = c_Q \omega_i^2 \tag{5}$$

where the drag factor parameter c_Q can be determined from static tests. By assuming that the distances of the octocopter center of mass from the center of mass of each single motor are equal to l, the input control torques will depend on the thrust force, T_i, and drag torque, Q_i, generated by each rotor and can be computed as follows:

$$\tau_x = lT_2 \sin(45°) + lT_3 + lT_4 \sin(45°) - lT_6 \sin(45°) - lT_7 - lT_8 \sin(45°) \tag{6}$$

$$\tau_y = lT_1 + lT_2 \cos(45°) - lT_4 \cos(45°) - lT_5 - lT_6 \cos(45°) + lT_8 \cos(45°) \tag{7}$$

$$\tau_z = -Q_1 + Q_2 - Q_3 + Q_4 - Q_5 + Q_6 - Q_7 + Q_8 \tag{8}$$

Substituting Eq.(3) in (6) and (7), Eq.(5) in (8) and by replacing Eqs.(4)-(6)-(7)-(8) in (2) leads to the **Allocation Matrix** of the form:

$$G_q = \begin{bmatrix} -c_T & -c_T & -c_T & -c_T & -c_T & -c_T & -c_T & -c_T \\ 0 & k & lc_T & k & 0 & -k & -lc_T & -k \\ lc_T & k & 0 & -k & -lc_T & -k & 0 & k \\ -c_Q & c_Q & -c_Q & c_Q & -c_Q & c_Q & -c_Q & c_Q \end{bmatrix} \tag{9}$$

with $k = \frac{\sqrt{2}}{2} l c_T$, by recalling that $\sin(45°) = \cos(45°) = \frac{\sqrt{2}}{2}$.

Note: in Eqs.(3)-(4)-(5)-(6)-(7)-(8) was adopted the same notation used in *Aerial Robotic Manipulation* (Ollero, Siciliano).

2 Aerodynamic effects

UAVs and UAMs frequently operate in applications that require flying in close proximity to different structures, objects and obstacles. These operational conditions constrain the airflow produced by the rotor, creating complex aerodynamic phenomena that significantly impact vehicle performance and stability. Two distinct aerodynamic effects emerge from these constrained flight operations: the ground effect and the ceiling effect. Understanding the fundamental differences between these aerodynamic phenomena becomes crucial for developing effective control strategies and ensuring safe operation of rotorcraft systems in complex environments. While both effects result from the interaction between rotor-generated airflow and nearby surfaces, they exhibit markedly different physical mechanisms, mathematical relationships and operational implications.

2.1 Ground Effect: Fundamental Mechanism and Characteristics

2.1.1 Definition and Operational Conditions

Ground effect occurs when a UAV flies closes to the ground, typically within one rotor diameter from the ground surface. The ground surface acts as a physical barrier that prevents the free expansion of the rotor's downward flow, creating a cushioning effect of compressed air between the drone and the surface. This aerodynamic phenomenon fundamentally enhances lift and reduces induced drag, since, being the downward airflow restricted, there is an increase in pressure beneath the drone and therefore lift.

2.1.2 Mathematical Modeling Approach

In order to model this effect, the **method of images** is employed, which consists in placing a "virtual rotor" on the opposite side to the surface and at the same distance. Consequently, if the real rotors are placed at height h, the virtual ones will be positioned underground at negative altitude $-h$. This mathematical approach allows for precise calculation of the aerodynamic interactions between the real rotor system and its virtual counterpart, providing accurate predictions of performance changes in ground effect conditions.

2.1.3 Thrust Enhancement Formula

The ratio between the thrust inside the ground effect (T_{IGE}) and the thrust outside the ground effect (T_{OGE}), derived from power preserving theory and from aerodynamic considerations, is given by:

$$\frac{T_{IGE}}{T_{OGE}} = \frac{1}{1 - \left(\frac{\rho}{4z}\right)^2} \tag{10}$$

where ρ is the radius of the propeller and z the height of the propeller from the ground. This mathematical relationship reveals a critical operational characteristic: when the rotor operates at an altitude greater than two times its radius, the lift gain from ground effect vanishes completely. This finding establishes clear operational boundaries for when ground effect benefits can be expected and utilized.

2.2 Ceiling Effect: Mechanism and Aerodynamic Characteristics

2.2.1 Physical Phenomenon Description

Ceiling effect represents the complementary aerodynamic phenomenon to ground effect, occurring when a UAV needs to approach some objects from below. In this condition, the thrust increases significantly closes to the ceil, pushing the rotor even closer to it. Then, it is important to take into account such an effect to avoid crashes.

2.2.2 Mathematical Representation

The increased thrust is a consequence of the increase of the propeller rotational velocity, caused by the vacuum effect, which decreases the propeller drag when it is very close to the ceil, so that the propeller rotates faster. This effect can be approximated by equation:

$$\frac{T_{\text{ICE}}}{T_{\text{OCE}}} = \frac{1}{1 - \frac{1}{k_1}\left(\frac{\rho}{z+k_2}\right)^2} \tag{11}$$

where T_{ICE} represents the thrust inside the ceiling effect, T_{OCE} denotes the thrust outside the ceiling effect, k_1 and k_2 are parameters to tune. This mathematical formulation demonstrates how the ceiling effect's magnitude becomes more pronounced as the aircraft approaches the overhead surface.

2.3 Detailed Comparative Analysis

2.3.1 Fundamental Mechanism Differences

The primary distinction between ground effect and ceiling effect lies in their respective thrust enhancement mechanisms. Ceiling effect is characterized by increased thrust, due to the restriction of airflow, limiting frictional resistance and allowing the propellers to rotate more efficiently. On the other hand, ground effect induces more lift due to the accumulation of expelled air at high speed between the rotor and the ground. This fundamental difference in physical mechanisms leads to distinct operational characteristics and control implications for each phenomenon.

2.3.2 Thrust Gain Variation Analysis

Ground effect and ceiling effect exhibit different behaviors in thrust gain variation as aircraft distance from constraining surfaces changes. The thrust gain changes between the two effects when considering the propellers at the same altitude, demonstrating that these phenomena cannot be treated as simple inverses of each other. The variation patterns reveal that ground effect tends to provide more predictable and stable thrust enhancement, while ceiling effect may exhibit more complex behavior patterns depending on the specific geometric configuration and operational parameters.

2.4 Stability Implications and Control Considerations

2.4.1 Ground Effect Stability Characteristics

When in the presence of ground effect a hovering multirotor is subject to attitude perturbation, a stabilizing moment M_{ge} appears due to differences in rotor-to-ground distances. This stabilizing moment acts to restore the aircraft to its equilibrium attitude configuration, providing inherent stability benefits during ground effect operations. The stabilizing nature of ground effect makes it generally beneficial for aircraft control, as it tends to resist attitude disturbances and maintain stable flight conditions. This characteristic makes ground effect particularly advantageous during critical flight phases such as takeoff and landing operations.

2.4.2 Ceiling Effect Instability Concerns

In ceiling effect conditions, an unstable moment appears that increases the attitude perturbation. This destabilizing characteristic makes ceiling effect potentially more challenging from a control perspective, requiring more sophisticated and responsive control systems to maintain stable flight. The unstable nature of ceiling effect means that aircraft operating under ceiling constraints require enhanced control authority and faster response times to prevent attitude divergence and maintain controlled flight conditions.

2.4.3 Complex Operational Scenarios

In situations, like grasping objects, only a portion of the rotors are subjected to ground or ceiling effect. When only a portion of rotors is subjected to ground effect, a moment M_{ge} will exist, that prevents the grasp of the object. Conversely, if only part of the rotors is subjected to ceiling effect, the moment M_{ge} will facilitate the grasp of the object.

2.5 Multi-Rotor Configuration Considerations

The analysis reveals that, in the case of multiple rotors, the presence of multiple coplanar rotors introduces additional complexity with respect to the single-rotor scenario, since the airflow from the different rotors may interfere with each other. Tipically, the airflow in the middle of the space between the rotors is reverted after reaching the ground. This airflow hits the multirotor frame producing an extra force. The same phenomenon which appears for a multirotor operating in ground effect appears when a multirotor is flying close to the ceiling. The flow field generated around each rotor is affected also by the other rotors and the multirotor frame. Thus, the ceiling effect for a full multirotor produces a different effect than adding the effects of each individual rotor due to the coupling between the flow of the propellers.

3 Momentum-based Estimator

The **momentum-based estimator** is employed to compensate for unwanted aerodynamic effects and the arm dynamics. Consider the RPY quadrotor dynamic model (12) and the generalized momentum vector $q \in \mathbb{R}^6$ (13):

$$\begin{cases} m\ddot{p}_b = mge_3 - u_T R_b e_3 + f_e \\ M(\eta_b)\ddot{\eta}_b = -C(\eta_b, \dot{\eta}_b)\dot{\eta}_b + Q^T(\eta_b)\tau^b + \tau_e \end{cases} \tag{12}$$

$$q = \begin{bmatrix} mI_3 & O_3 \\ O_3 & M(\eta_b) \end{bmatrix} \begin{bmatrix} \dot{p}_b \\ \dot{\eta}_b \end{bmatrix} \tag{13}$$

In (12) the external wrench, $[f_e \quad \tau_e]^T$, is representative of unmodelled dynamics, disturbances and model uncertainty. The objective is to estimate the wrench $[f_e \quad \tau_e]^T$ through $[\hat{f}_e \quad \hat{\tau}_e]^T$. In particular, the goal of the proposed estimator is to achieve a linear relationship between the estimated external wrench and its time derivatives with the real one. To this end the Laplace domain can be exploited, in which holds:

$$L\begin{bmatrix} \hat{f}_e \\ \hat{\tau}_e \end{bmatrix} = G(s)L\begin{bmatrix} f_e \\ \tau_e \end{bmatrix} \tag{14}$$

where $G(s) \in \mathbb{C}^{6\times 6}$ is a diagonal matrix of transfer functions whole i-th element is in the form:

$$\mathbf{G}_i(\mathbf{s}) = \frac{c_0}{s^r + c_{r-1}s^{r-1} + \cdots + c_1 s + c_0}, \quad i = 1, \ldots, 6 \tag{15}$$

where $c_i > 0$ are coefficients assembling a Hurwitz polynomial and $r \geq 1$ is the desired degree of the estimator. Hence, each component of the estimated external wrench is a filtered estimate of the real one. To get (15) component-wise in (14), the following recursive estimator is designed in the discrete time with a sample time $Ts = 1ms$:

$$\gamma_1(k+1) = \gamma_1(k) + K_1\left(q(k+1) - q(k) - T_s\begin{bmatrix} \hat{f}_c(k) \\ \hat{\tau}_c(k) \end{bmatrix} - T_s\begin{bmatrix} mge_3 - u_T(k)R_b(k)e_3 \\ C^T(\eta_b(k), \dot{\eta}_b(k)) + Q^T(\eta_b(k))\tau^b(k) \end{bmatrix}\right) \tag{16}$$

$$\gamma_i(k+1) = \gamma_i(k) + K_i T_s\left(-\begin{bmatrix} \hat{f}_c(k) \\ \hat{\tau}_c(k) \end{bmatrix} + \gamma_{i-1}(k)\right) \quad i = 2, ..., r \tag{17}$$

where $K_i \in \mathbb{R}^{6\times 6}$ are positive definite diagonal matrices, which are directly related to the coefficients of the transfer function (15) via the relation:

$$\prod_{i=j+1}^{r} \mathbf{K}_i = c_j \quad \text{for } j = 0, \ldots, r-1 \tag{18}$$

To simplify the design and calibration of the estimator consider the transfer function of the form:

$$\mathbf{G}_i(\mathbf{s}) = \frac{k_0^r}{(s + p_{ole})^r}, \quad i = 1, \ldots, 6 \tag{19}$$

By setting $p_{ole} = k_0$ and ensuring $p_{ole} \geq 0$ (necessary condition for stability), it is possible to obtain an effective formulation in which the entire behaviour of the estimator is governed by two parameters: the order r and the value of k_0.

Assuming as initial conditions:

$$q = 0 \qquad \begin{bmatrix} \hat{f}_e \\ \hat{\tau}_e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{20}$$

means that at the initial instant the quadcopter has zero speed, is stationary, and the estimation starts before its takeoff.

3.1 Comparison of the estimation results with different values of r

By setting $k_0 = 20$ and assuming that the following disturbances are applied during the flight:

- $1N$ along the x- and y-axis of the world frame
- $-0.4Nm$ around the yaw axis

below the estimation results will be compared with different values of r.
(**NOTE:** The results discussed in the following refer to POINT3_HW3.mlx in HW3_FSR_Emanuela_Varone/POINT_3 folder.)

- **Low Values of r=[1,15]:**

As can be seen by observing Figs.[3]-[4]-[5]:

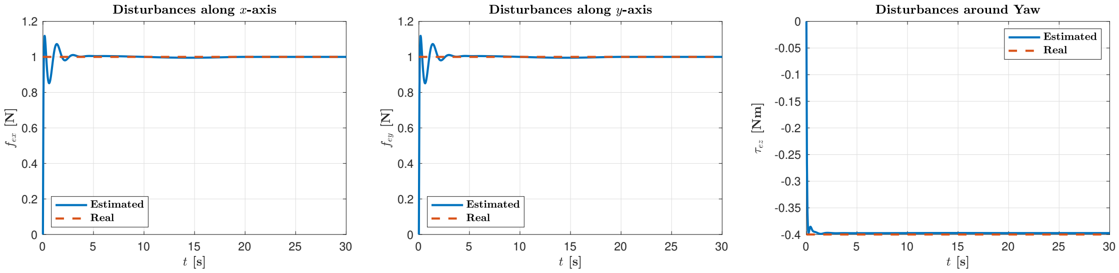


Figure 3: r=1

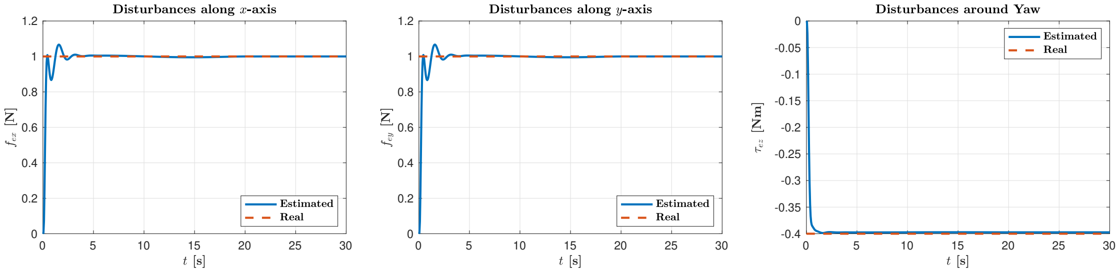


Figure 4: r=5

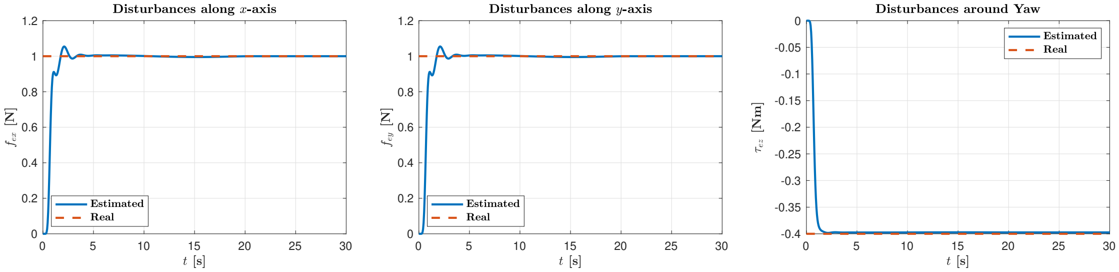


Figure 5: r=15

For $r = 1$ (minimum value), it can be observed that the estimated disturbances converge approximately toward 1 N along the x- and y-axes, but with non-negligible oscillations. The yaw torque disturbance estimate tends to approach -0.4 Nm, although it is affected by noise. For $r = 5$, a significant improvement is observed compared to the case with $r = 1$. The oscillations are noticeably reduced and the estimates converge more rapidly toward the target values. Although the signals still exhibit some noise, its amplitude is considerably lower. For $r = 15$, a further reduction in oscillations is achieved and the estimates of the disturbances appear more accurate and stable compared to previous cases.

- **Intermediate Values of r=[20,80]:**

As can be seen by observing Figs.[6]-[7]-[8], between $r = 20$ and $r = 40$, there is still a noticeable improvement, especially in noise reduction. From $r = 40$ to $r = 80$, the improvement is more marginal and mainly related to the reduction of residual noise, at the expense of a slight increase in response time. For $r > 40$, the further increase in r **does not bring significant** improvements in accuracy or stability of the estimate. The curves for $r = 50, 60, ..., 80$, are almost superimposable to those for $r = 40$, indicating that the system has reached a performance "plateau".

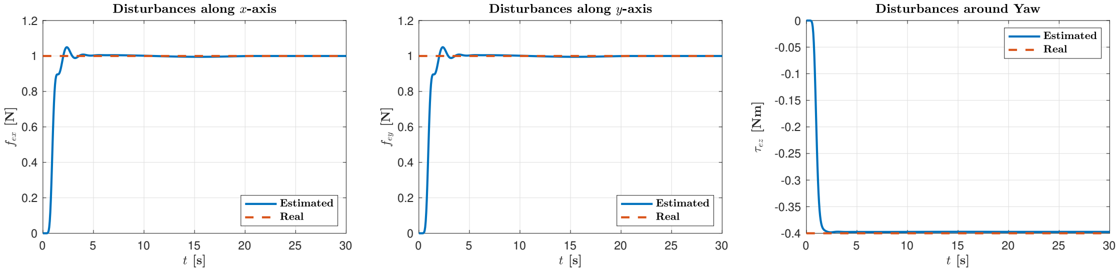


Figure 6: r=20

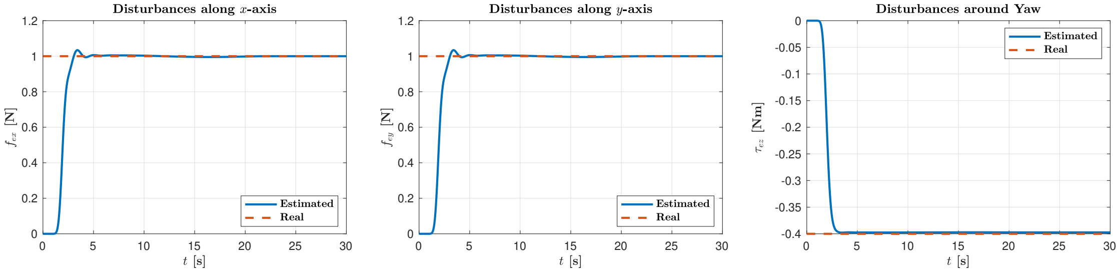


Figure 7: r=40

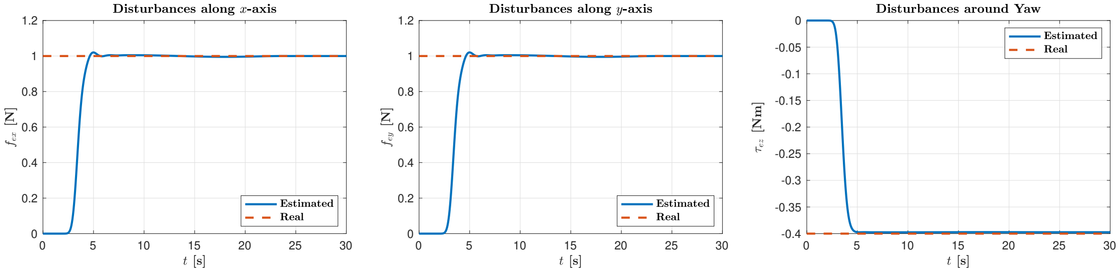


Figure 8: r=80

- **High Values of r=[100,150]:**

As can be seen by observing Figs.[9]-[10], for $r > 100$, not only is no improvement observed, but abnormal behaviour is noted with large oscillations and estimates becoming unstable and unreliable. This suggests that a too high value of r can lead to numerical instability problems in the estimator.

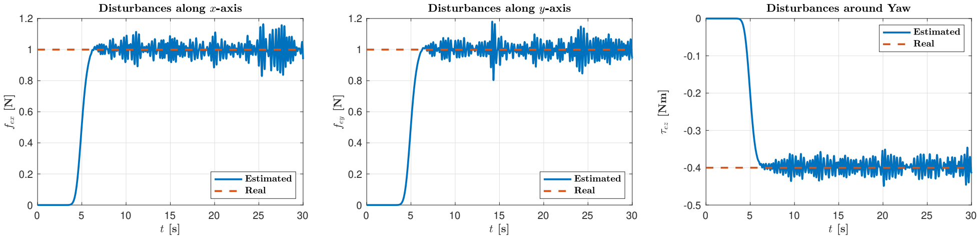


Figure 9: r=100

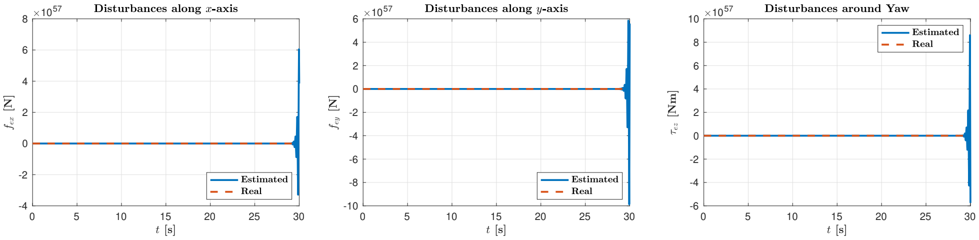


Figure 10: r=150

From the results obtained, it can be concluded that an r-value in the range of 40-60 represents a good compromise between estimation accuracy, response time and numerical stability.

3.2 Real mass of the UAV computation from the estimated disturbance along the z-axis

The real mass of the UAV was determined by analyzing the external disturbance estimated along the z-axis. Mathematically, the estimated disturbance along the z-axis can be expressed as:

$$\hat{f}_{e,z} = (m_{\text{real}} - m_{\text{nominal}}) \cdot g \tag{21}$$

From relation (21), it is possible to derive the actual mass:

$$m_{\text{real}} = m_{\text{nominal}} + \frac{\hat{f}_{e,z}}{g} \tag{22}$$

The implemented computation procedure consists of the following steps: first, the final estimated value of the disturbance along the z-axis, $\hat{f}_{e,z}$, is extracted. This value corresponds to extWrench_hat(3,end). Next, the estimated mass $\tilde{m}$ is

computed by dividing this disturbance by the gravitational acceleration. Finally, the actual mass is obtained by adding $\tilde{m}$ to the nominal mass.
The MATLAB implementation was as follows:

```matlab
% Compute the estimated mass from estimated z-axis disturbance
m_tilde = extWrench_hat(3,end) / g;

% Compute actual UAV mass
estimated_mass = m + m_tilde;
```

Listing 1: Estimation of actual mass from disturbance

Using this methodology, the estimated real mass of the quadcopter is $1.25kg$, 16.7% lower than the nominal value of $1.5kg$ used in the model. This discrepancy explains the presence of the persistent noise observed along the z-axis in the estimates provided by the momentum-based estimator. It is important to note that the accuracy of this estimate depends on the quality of the estimator used and the value of the r parameter chosen.

4 Geometric control of a quadrotor

The geometric controller consists of two main components: an outer position control loop and an inner attitude control loop. Through simulation, it is possible to demonstrate the effectiveness of the controller in tracking desired trajectories while maintaining stable flight dynamics. Considering the InitFcn callback in the geometric_control_template.slx file, it can be seen that the intended objective consists of tracking a trajectory from the initial position $\mathbf{p}_i = [0,\ 0,\ -1]\,\text{m}$, with $\psi_i = 0°$, to the final $\mathbf{p}_f = [1,\ 1,\ -4]\,\text{m}$, with $\psi_f = 20°$, over 20 seconds, with an additional 10-second steady-state hold.

4.1 Outer Loop Control

The outer loop generates the desired thrust vector based on position and velocity errors:

$$\mathbf{e}_p = \mathbf{p}_b - \mathbf{p}_{b,d}, \quad \dot{\mathbf{e}}_p = \dot{\mathbf{p}}_b - \dot{\mathbf{p}}_{b,d} \tag{23}$$

To ensure that the planned desired linear acceleration does not exceed the gravitational one, which would make the control problem infeasible, under the assumption:

$$|| - mg\mathbf{e}_3 + m\ddot{\mathbf{p}}_{\mathbf{b,d}}|| < \text{positive constant} \tag{24}$$

the control law for the desired thrust is:

$$u_T = -(-\mathbf{K}_p\mathbf{e}_p - \mathbf{K}_v\dot{\mathbf{e}}_p - mg\mathbf{e}_3 + m\ddot{\mathbf{p}}_{\mathbf{b,d}})\mathbf{R}_b\mathbf{e}_3 \tag{25}$$

Then, the desired z-axis, $z_{b,d}$, of the body frame is computed as:

$$\mathbf{z}_{b,d} = -\frac{-\mathbf{K}_p\mathbf{e}_p - \mathbf{K}_v\dot{\mathbf{e}}_p - mg\mathbf{e}_3 + m\ddot{\mathbf{p}}_{\mathbf{b,d}}}{||| - \mathbf{K}_p\mathbf{e}_p - \mathbf{K}_v\dot{\mathbf{e}}_p - mg\mathbf{e}_3 + m\ddot{\mathbf{p}}_{\mathbf{b,d}}||} \tag{26}$$

4.2 Inner Loop Control

Once the desired $z_{b,d}$ axis is defined, it is possible to construct the desired y-axis, $\mathbf{y}_{b,d}$:

$$\mathbf{y}_{b,d} = \frac{\mathbf{z}_{b,d} \times \mathbf{x}_{b,d}}{||\mathbf{z}_{b,d} \times \mathbf{x}_{b,d}||} \tag{27}$$

Next, to ensure orthogonality, the x-axis, $\mathbf{x}_{b,d}$, is computed as:

$$\mathbf{x}_{b,d} = S(y_{b,d})\mathbf{z}_{b,d} \tag{28}$$

Starting from $\mathbf{x}_{b,d}$, $\mathbf{y}_{b,d}$ and $z_{b,d}$, it is possible to evaluate the desired rotation matrix:

$$\mathbf{R}_{b,d} = [\mathbf{x}_{b,d} \quad \mathbf{y}_{b,d} \quad \mathbf{z}_{b,d}] \tag{29}$$

The rotational error in SO(3), e_R, and the angular velocity error in SO(3), e_ω, are defined, respectively, as:

$$\mathbf{e}_R = \frac{1}{2}\left(\mathbf{R}_{b,d}^\top\mathbf{R}_b - \mathbf{R}_b^\top\mathbf{R}_{b,d}\right)^\vee \tag{30}$$

$$\mathbf{e}_\omega = \boldsymbol{\omega}_b^b - \mathbf{R}_b^T\mathbf{R}_{b,d}\boldsymbol{\omega}_{b,d}^{b,d} \tag{31}$$

The torque control law is then:

$$\boldsymbol{\tau}^b = -\mathbf{K}_R\mathbf{e}_R - \mathbf{K}_\omega\mathbf{e}_\omega + \mathbf{S}(\boldsymbol{\omega}_b^b)\mathbf{I}_b\boldsymbol{\omega}_b^b - \mathbf{I}_b(\mathbf{S}(\boldsymbol{\omega}_b^b)\mathbf{R}_b^T\mathbf{R}_{b,d}\boldsymbol{\omega}_{b,d}^{b,d} - \mathbf{R}_b^T\mathbf{R}_{b,d}\dot{\boldsymbol{\omega}}_{b,d}^{b,d}) \tag{32}$$

4.3 Controller Performance Analysis

The gain optimization process followed an iterative approach based on system performance analysis. Gain tuning was performed through several simulations, with the aim of minimizing errors, while reducing oscillations and overshoots. After several optimization iterations, the final values of the gains were set as follows:

Outer Loop:

- Proportional gains: $K_p = \text{diag}([80, 80, 150])$
- Derivative gains: $K_v = \text{diag}([8, 8, 20])$

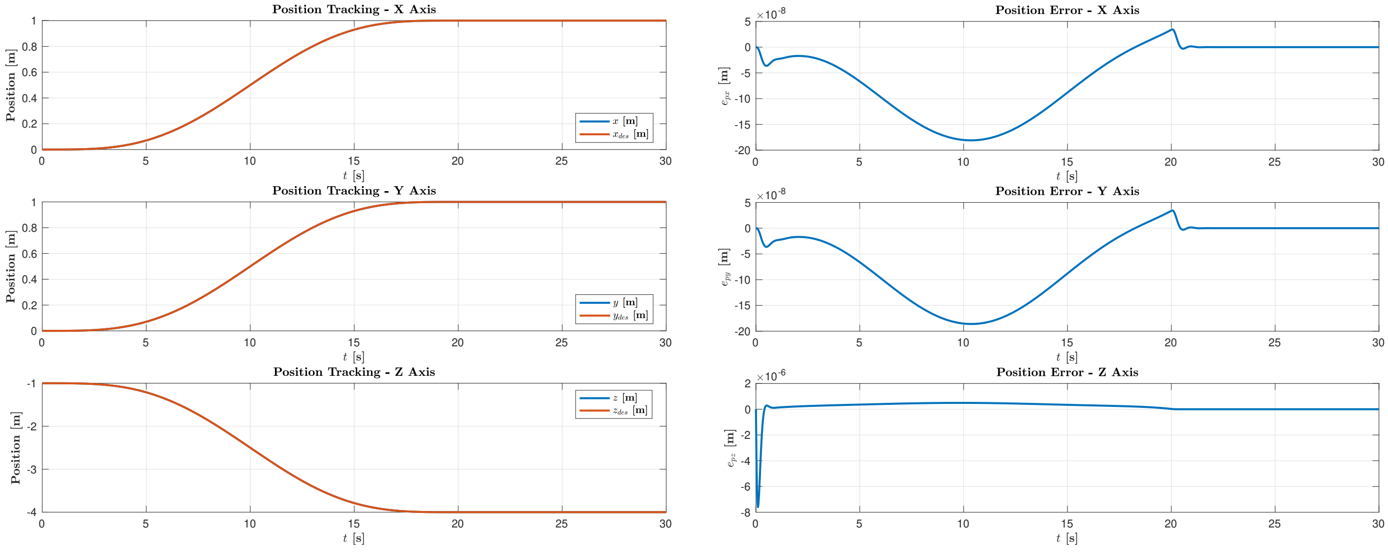
Inner Loop:

- Gains for rotational error in SO(3): $K_R = 60 \cdot \text{I}_3$
- Gains for angular velocity error in SO(3): $K_\omega = 20 \cdot \text{I}_3$

(**NOTE:** The results discussed in the following refer to geometric_control_template.slx file in HW3_FSR_Emanuela_Varone/POINT_4 folder).

4.3.1 Position Tracking and Errors

As shown in Fig.[11a], the system is capable of accurately following the reference trajectory along all axes. The tracking is particularly precise, with an almost perfect overlap between the actual and desired position curves. The tracking meets the objective exactly: the quadrotor starts $((x_i, y_i) = (0, 0))$ at $0m$, descends from $z_i = -1m$ to $z_f = -4m$ depth, following a smooth S-curve trajectory, reaching the target position of $(x_f, y_f, z_f) = (1, 1, -4)m$ around $t = 20s$, followed by a 10-second hold at the final position.



(a) Position tracking

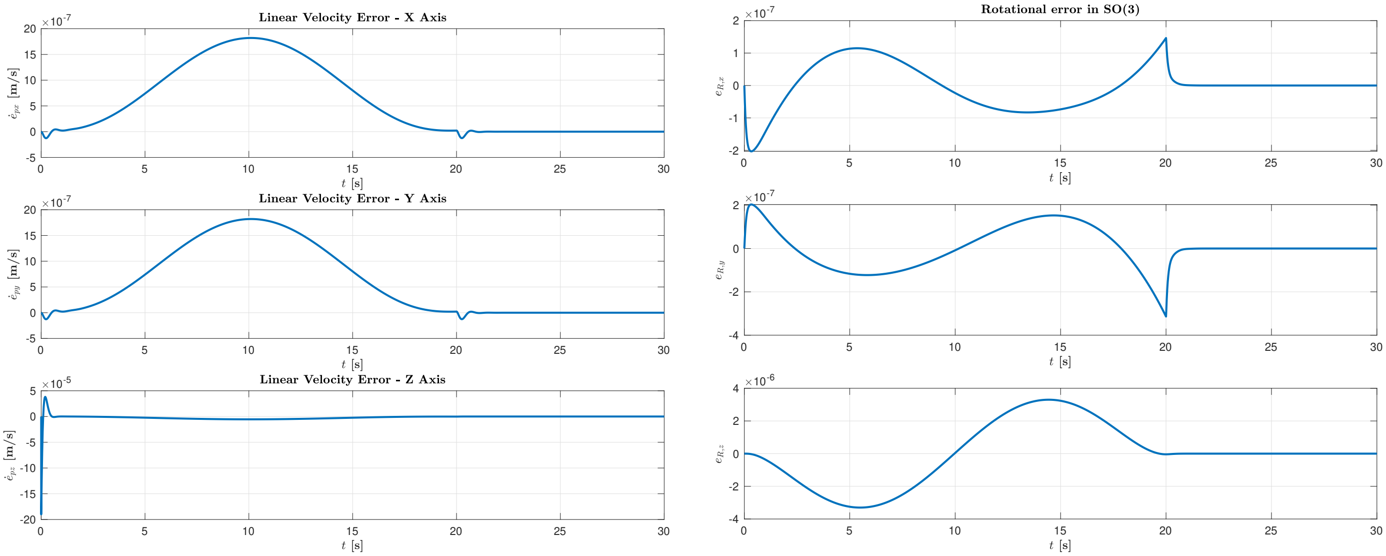
(b) Position errors

Figure 11: Position Tracking and Errors

The analysis of position errors (Fig.[11b]) reveals that: the position errors are extremely small, on the order of 10^{-8} m for the x and y axes and 10^{-6} m for the z axis. The e_{px} and e_{py} errors exhibit a sinusoidal pattern during the trajectory movement phase $(0 - 20s)$, with peak values reached around 10 seconds. A small transient peak appears around 20 seconds, when the trajectory reaches the final point. This peak is quickly damped, indicating good system robustness.

4.3.2 Linear Velocity Errors

As shown in Fig.[12a], the velocity errors follow a similar pattern to the position errors: remain on the order of 10^{-7} m/s for the x and y axes and 10^{-5} m/s for the z axis. The velocity errors exhibit a faster response than the position errors, anticipating them in time. The extremely small magnitude of these errors indicates outstanding linear velocity tracking performance.



(a) Linear Velocity Errors

(b) Rotational Error in SO(3)

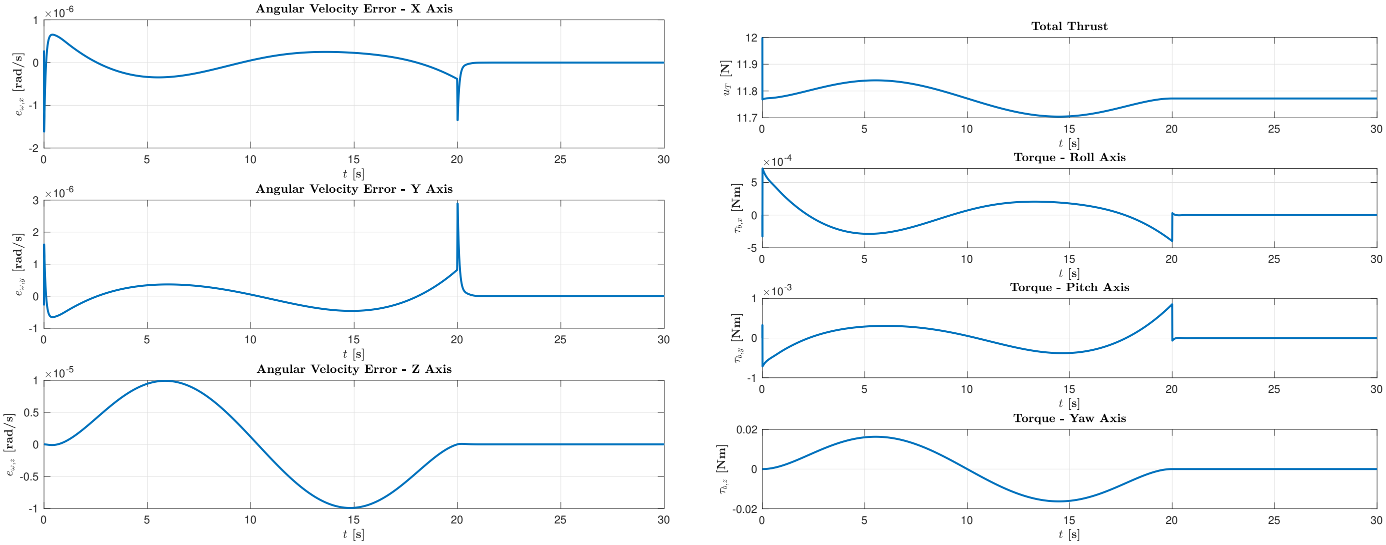
Figure 12

4.3.3 Rotational Error in SO(3)

Fig.[12b] shows the rotational error trend: all three components exhibit oscillatory behavior with extremely small magnitudes (10^{-7} to 10^{-6} range). Clear transients appear at $t = 20s$ when the trajectory completes. After $t = 20s$, all errors quickly converge to zero values, indicating perfect orientation tracking during the steady-state phase.

4.3.4 Angular Velocity Errors

Angular velocity errors (Fig.[13a]) are also impressively small (10^{-6} to 10^{-5} rad/s range): the $x-axis$ shows oscillatory behavior with a peak of about 7×10^{-7} rad/s, with a sharp transient at $t = 20s$. $y - axis$ exhibits similar oscillatory behavior to $x-axis$ but with a significant spike at $t = 20s$ reaching nearly $3\times10^{-6}rad/s$. $z-axis$ shows a full sinusoidal oscillation over the $20s$ trajectory, with amplitude around $1 \times 10^{-5}rad/s$ (larger than x and y axes errors).



(a) Angular Velocity Errors

(b) Control signals

Figure 13

4.3.5 Thrust and Torques

Consider Fig.[13b].

- The **top plot shows the total thrust** applied by the quadrotor's propellers throughout the trajectory: the thrust oscillates between approximately $11.7N$ and $11.85N$, representing a variation of only about $0.15N$ from the mean value. Such a small variation (only 1.3%) demonstrates that the controller is optimizing the use of energy, applying only the minimal variations necessary to execute the desired trajectory with precision. Around $t = 5s$, the thrust increases to approximately $11.83N$, as the quadrotor begins accelerating in the x and y directions while also descending. This slight increase compensates for the additional power needed during simultaneous multi-axis movement. Near $t = 15s$, the thrust decreases to its minimum of about $11.7N$, which corresponds to the point of maximum descent rate in the $z - axis$. This reduction makes sense physically as the quadrotor can use less thrust while descending. After $t = 20s$, the thrust stabilizes at approximately $11.77N$, which represents the exact thrust needed to hover at the final position ($-4m$ altitude).

- The **second plot shows the torque** $\tau_{b,x}$, which controls rotation around the $x - axis$. The torque takes very small values in the $10^{-4}Nm$ range, indicating minimal roll correction needed during the trajectory. At $t = 20s$, there's a sharp negative spike followed by an immediate correction to a steady value of approximately $0.310^{-4}Nm$. This rapid adjustment corresponds to the controller finalizing the desired orientation at the end point. These small roll torques help maintain the quadrotor's lateral stability while it moves in the x and y directions simultaneously.

- The **third plot shows the torque** $\tau_{b,y}$, controlling rotation around the $y - axis$. The magnitude is larger than roll torque, in the $10^{-3}Nm$ range, indicating more significant pitch adjustments during flight. The pitch torque pattern aligns with the forward acceleration and deceleration needed to trace the S-curve trajectory in both x and y, with higher values corresponding to the points requiring the most pitch attitude change.

- The **bottom plot shows the torque** $\tau_{b,z}$, which controls rotation around the $z - axis$. The magnitude is significantly larger than both roll and pitch, reaching values up to $0.018Nm$ (about 10 times larger than pitch torque). The sinusoidal pattern directly corresponds to the yaw rotation from $0°$ to $20°$ over the 20-second trajectory. The controller initially applies positive torque to accelerate, then negative torque to decelerate and precisely stop at the desired $20°$ orientation.

5 Tilting Quadrotor Controller Using the Voliro Approach

The control system for the tilting quadrotor aims to achieve precise trajectory tracking in both position and attitude while exploiting the unique capabilities of the tilting rotor configuration. Specifically, the control objectives include: **position tracking** (the quadrotor should follow a prescribed position trajectory with minimal error), **attitude tracking** (the quadrotor should track a desired orientation), **stability** (the system should remain stable throughout operation), **exploiting the full 6-DOF control capability** (unlike conventional quadrotors, tilting rotors allow for independent control of position and orientation).

5.1 Controller Implementation

The implemented controller follows the Voliro approach, which transforms the nonlinear allocation problem into a linear one through variable transformation. The controller implements a PD control scheme with feedforward terms, designed to handle both position and attitude tracking. The control architecture can be summarized as follows:

- **Error Computation:** position, velocity and attitude errors are computed based on the desired and current states.
- **Control Law:** a PD controller with feedforward terms generates the desired wrench (forces and torques):

  - Position control: $u_T = -K_p e_p - K_d \dot{e}_p + \ddot{p}_{des}$

  - Attitude control: $\tau = -K_r e_R - K_w e_\omega + \dot{\omega}_{des}$
- **Allocation Strategy:** the controller uses a pseudoinverse approach to allocate the desired wrench to the individual actuator commands (rotor speeds and tilt angles). The allocation matrix G_{static} maps the actuator commands to the wrench. The inverse dynamics approach computes the control inputs using $u = G^{\dagger}_{static}(v - b)$.
- **Command Computation:** the rotor speeds and tilt angles are calculated from the transformed variables:

  - Tilt angles: $\alpha_i = \text{atan2}(u_{l,i}, u_{v,i})$

  - Rotor speeds: $u_{\omega_i} = (\sqrt{u_{l,i}^2 + u_{v,i}^2})/c_f$
- The controller gains were tuned to achieve good tracking performance while maintaining stability:

  - Position gains: $K_p = \text{diag}([140, 140, 250])$, $K_d = \text{diag}([70, 70, 85])$

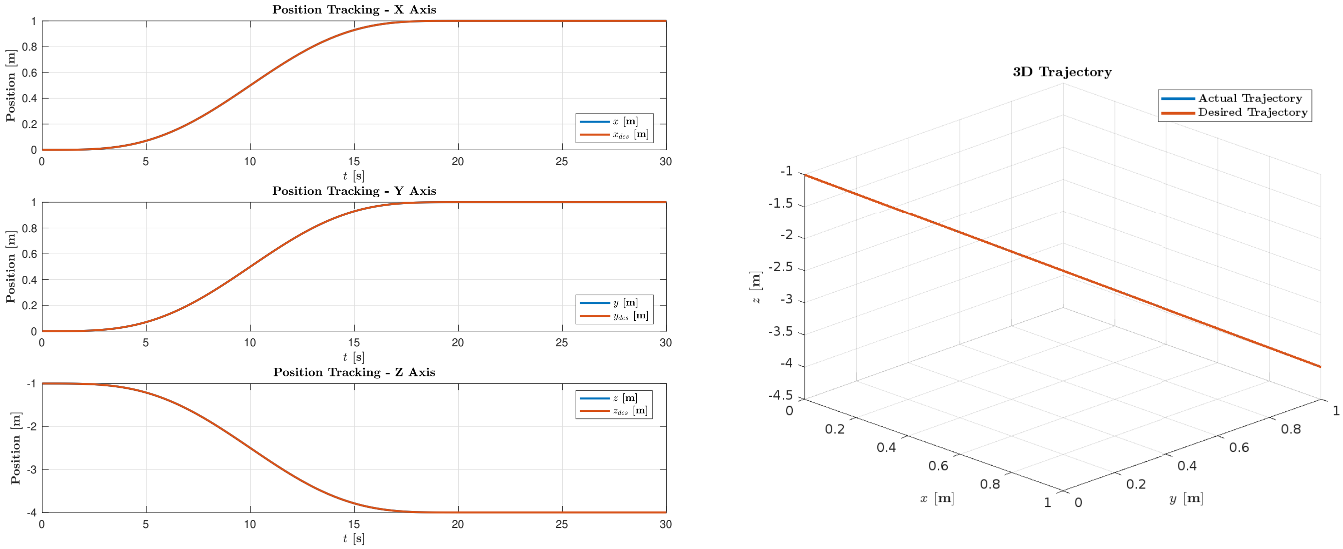
  - Attitude gains: $K_r = \text{diag}([200, 200, 350])$, $K_w = \text{diag}([120, 120, 160])$

5.2 Controller Performance Analysis

In this subsection the control performance of the tilting quadrotor using the Voliro approach will be analised. (**NOTE:** The results discussed in the following refer to tilting_control_voliro_template.slx file in HW3_FSR_Emanuela_Varone/POINT_5 folder.)

5.2.1 Position Tracking and Trajectory

The position tracking plots (Fig.[14a]) reveal the trajectory followed by the quadrotor: it moves from 0 to 1 meter along the x and y axes, following an S-shaped trajectory and descends from -1 to -4 meters. The actual position (blue line) perfectly overlaps with the desired position (orange line), indicating excellent tracking performance. Looking at the attached video (“UAV_trajectory.mp4”) in the folder HW3_FSR_Emanuela_Varone/Point_5/VIDEO, it can be noted that: initially positioned at coordinates $(0, 0, -1)m$, the quadrotor begins its motion after approximately 5 seconds of hover. The trajectory then unfolds as a simultaneous movement along all three spatial axes: in the horizontal plane, the quadrotor performs a diagonal movement, traveling 1 meter in the positive x-direction and 1 meter in the positive y-direction. Concurrently, the quadrotor descends 3 meters along the z-axis, moving from an altitude of -1 meter to a final altitude of -4 meters. This combined motion creates a diagonal descending trajectory (see Fig.[14b]) that brings the quadrotor from its starting point to the target position at $(1, 1, -4)$ meters. The movement profile follows smooth S-curves, indicating gradual acceleration and deceleration phases rather than abrupt changes in velocity. In the final seconds of the trajectory (after $t = 20s$), the quadrotor reaches its destination and enters a stable hovering phase. During this terminal phase, the quadrotor maintains its position at $(1, 1, -4)$ meters with remarkable precision (position errors less than 10^{-5} meters). This final hovering phase demonstrates the effectiveness of the controller in maintaining a stable position with tilted rotors, showcasing the Voliro approach’s capability to hold non-conventional flight configurations.



(a) Position tracking

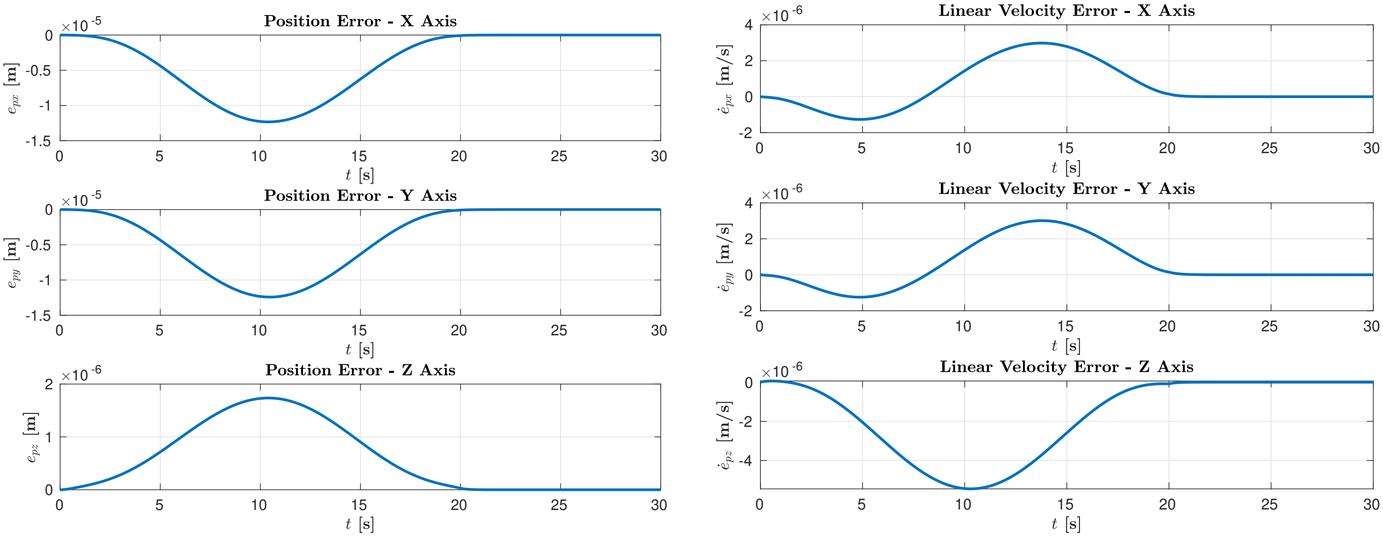
(b) Trajectory

Figure 14: Position Tracking and Trajectory

5.2.2 Position and Linear Velocity Errors

The position error plots (Fig.[15a]) reveal exceptional tracking accuracy across all three spatial dimensions, though with distinctly different behavioral patterns that provide crucial insights into the underlying trajectory structure and control effectiveness. The x and y position errors exhibit identical profiles, starting at zero and following precise parabolic trajectories. Both reach their minimum values of exactly $-1.25 \times 10^{-5}m$ at $t = 10s$, then gradually converge to zero. This perfect symmetry indicates the reference trajectory follows a diagonal path in the horizontal plane with a 45-degree orientation. The error magnitude suggests the quadrotor is commanded to move approximately 12.5 millimeters in each horizontal direction during the maneuvering phase. The $z - axis$ position error demonstrates fundamentally different dynamics, exhibiting a single positive peak of $1.7 \times 10^{-6}m$ at $t = 10s$. This peak occurs precisely when the horizontal errors reach their minima, indicating coordinated three-dimensional motion. The vertical error magnitude is approximately 7.4 times smaller than the horizontal errors, suggesting the reference trajectory includes a modest climbing component during the horizontal translation, creating a shallow helical or spiral path. The settling behavior provides quantitative insight into controller performance. All position errors converge to steady-state values within 20 seconds, achieving final tracking accuracies with essentially zero error.

The linear velocity error profiles (Fig.[15b]) provide critical information about the trajectory’s dynamic characteristics and the controller’s ability to manage velocity tracking during different flight phases. The x and y velocity errors maintain their symmetric relationship observed in position, with both channels exhibiting identical bell-shaped curves. The $z-axis$ velocity error shows distinctly different temporal behavior, with a negative peak of $-5\times10^{-6}m/s$ occurring at $t = 10s$. This negative peak coincides exactly with the positive position error peak, indicating the controller is commanding upward velocity to correct for the altitude deficit. All velocity errors approach zero after $t = 22s$.



(a) Position errors

(b) Linear Velocity Errors

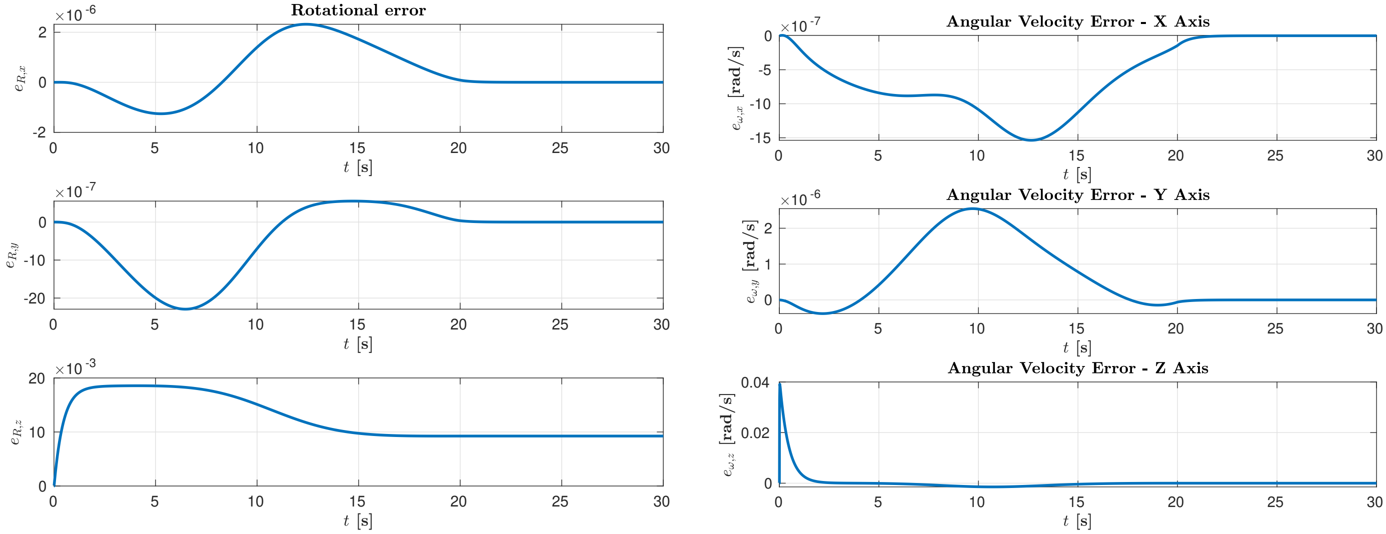
Figure 15

5.2.3 Rotational Error

The rotational error plots (Fig.[16a]) provide insight into orientation tracking. The $x-axis$ rotational error follows a pattern with negative values reaching a minimum around $t = 6s$ at approximately $-1.5\times10^{-6}rad$, followed by a positive peak of $2\times10^{-6}rad$ around $t = 13s$. The $y-axis$ rotational error exhibits a predominantly negative profile, starting at zero, decreasing to approximately $-2.2\times10^{-7}rad$ at $t = 6s$, then recovering with a slight positive overshoot around $t = 12s$ before settling. The smaller magnitude compared to the $x-axis$ error indicates better tracking performance in the y direction. The $z-axis$ rotational error shows the most significant deviation, starting from zero and rapidly increasing to approximately $1.8\times10^{-2}rad$ by $t = 3s$, then gradually decreasing to a steady-state value around $9\times10^{-3}rad$. This large persistent error is particularly concerning as it indicates poor yaw tracking performance.

5.2.4 Angular Velocity Errors

The angular velocity error plots (Fig.[16b]) show:
The $x-axis$ angular velocity error exhibits a distinctive pattern that reveals important characteristics of the control system's transient response. Initially starting at zero, the error decreases to approximately $-8\times10^{-7}rad/s$ around $t = 8s$, reaching its minimum of $-15\times10^{-7}rad/s$ at $t = 13s$ before gradually recovering to near-zero by $t = 20s$ and maintaining steady-state thereafter. This behavior indicates that the controller experiences temporary tracking difficulties during the maneuver execution phase. The gradual recovery pattern demonstrates that the controller corrects these errors. The $y-axis$ error presents a fundamentally different dynamic profile, exhibiting a bell-shaped curve that peaks at approximately $2.5\times10^{-6}rad/s$ around $t = 10s$. The symmetric nature of this error profile, with its gradual increase and decrease, indicates a well-damped response without oscillations. However, the magnitude being larger than the x-axis error suggests that the y-axis dynamics may be more challenging to control. The $z-axis$ error is characterized by a sharp initial spike reaching $0.04rad/s$ at $t = 1s$, before rapidly decaying to near-zero within 5 seconds. This large initial error, approximately two orders of magnitude greater than the x and y errors, indicates significant coupling between the yaw dynamics and the initial trajectory commands.



(a) Rotational Error

(b) Angular Velocity Errors

Figure 16

5.2.5 Tilt Angle Analysis

Consider (Fig.[17]), that shows the tilt angles of the four motors over time. Motors 1 and 2 exhibit similar negative tilt angle trajectories, both transitioning from 0 to approximately -1 rad over the first 15 seconds of simulation. Motor 1 shows a more gradual transition with an inflection point around $t = 8s$, while Motor 2 demonstrates a more linear decrease starting immediately. These negative tilt angles indicate that both motors are tilting to generate lateral force components. The coordination between these motors suggests they are working together to produce desired force vectors in the same general direction. In contrast, Motors 3 and 4 exhibit positive tilt angles, both reaching approximately 1 radian by $t = 15s$. Motor 3 shows an S-shaped trajectory with initial negative tilt before transitioning positive around $t = 10s$, while Motor 4 demonstrates a more monotonic increase from the beginning. The opposite tilt directions between motor pairs (1,2 vs 3,4) suggests a coordinated control strategy where opposing motors provide balancing forces and torques. This configuration is essential for maintaining equilibrium while generating net translational forces.

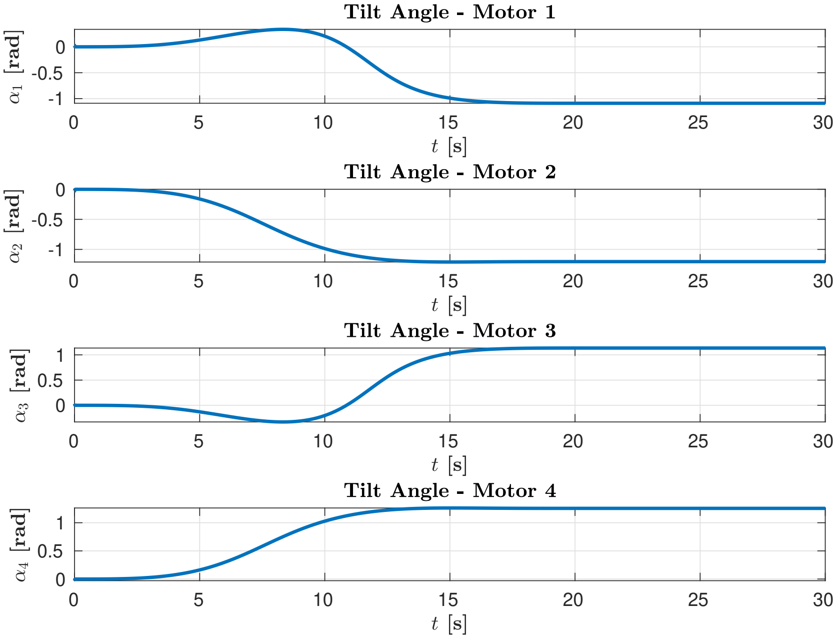


Figure 17: Tilt angles