



Università degli Studi di Napoli  
Federico II

Dipartimento di Ingegneria Elettrica e Tecnologie  
dell'Informazione

Corso di Laurea in Ingegneria dell'Automazione e Robotica

Field and Service Robotics

Homework 3

Marco Bartone P38000237

## Exercise 1

To address the first question, it is important to highlight that the number of degrees of freedom (DoF) varies depending on whether the drone is restricted to the ground or allowed to move freely in space. This robot has 8 degrees of freedom, each one associated with each propeller. Therefore, two scenarios were analyzed separately and the corresponding degrees of freedom and configuration spaces were determined for each case.

### Ground-Constrained case

When the drone is constrained to remain on the ground, it possesses exactly 8 degrees of freedom. Each propeller is modeled as a revolute joint, which corresponds to a configuration space of  $S^1$ . As a result, the total configuration space of the octocopter is equal to  $T^8$ .

### Free-Motion in Space

In this case the robot is capable of moving freely in space. Therefore, in addition to the 8 degrees of freedom associated with the propellers, we must consider 6 more degrees of freedom that define its spatial movement, expressed as  $\mathbb{R}^3 \times S^2 \times S^1 = \mathbb{R}^3 \times SO(3)$ . This results in a total configuration space of  $T^8 \times \mathbb{R}^3 \times SO(3)$ .

### Underactuation

From the provided image, it can be observed that all the propellers lies in the same plane. This flat configuration leads to an allocation matrix with the first two rows equal to zero. As a result, the matrix does not have full rank, and the system is therefore classified as underactuated.

### Allocation Matrix

In order to define the allocation matrix of the system, the body frame is considered according to the NED (North-East-Down) configuration, with its origin located at the drone's center of mass. Additionally, reference frames are assigned to each propeller, which differ from one another by an angle  $\alpha = 45^\circ$  with respect to the z-axis, due to the symmetry of the structure.

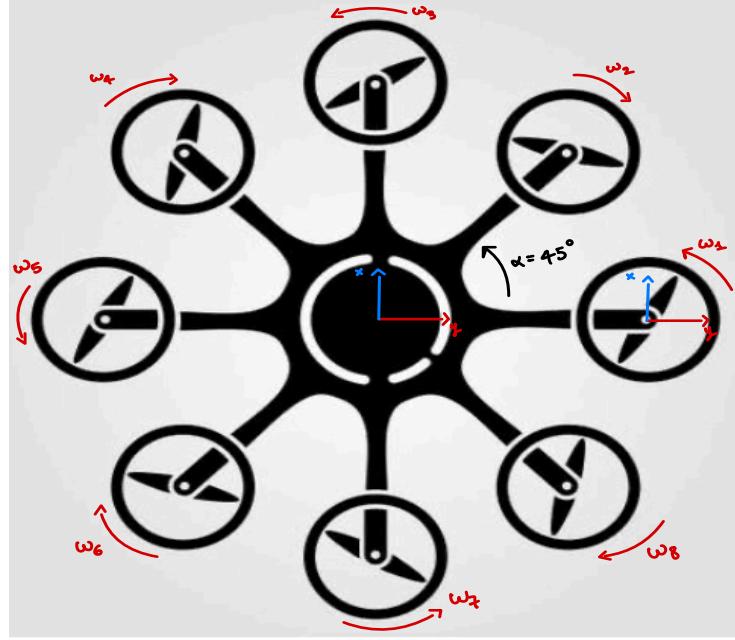


Figure 1: Top view with reference frames

The total thrust  $f^b$  and the drag moment  $\tau^b$  generated by the propellers with respect to the body frame are expressed as the net force and moment contributions from each individual rotor:

$$f^b = \sum_{i=1}^n c_f |\omega_i| \omega_i z_{P_i}^b$$

$$\tau^b = \sum_{i=1}^n |\omega_i| \omega_i (-k_i c_m z_{P_i}^b + c_f S(p_{P_i}^b) z_{P_i}^b)$$

In particular, the coefficient  $k_i = 1$  is assigned to the propellers rotating clockwise (descending chord), and  $k_i = -1$  to those rotating counterclockwise (ascending chord). As described in the previous subsection, due to the flat configuration of the propellers, the first two rows of the allocation matrix are zero. The third row, corresponding to the  $f_z$  component, consists of elements equal to  $c_f$ , since the direction of each propeller's axis  $z_{P_i}$  is opposite to that of the body frame.

The torque-related rows along the body axes are defined as follows:

- Along the  $x$ -axis: each component is given by the product of  $c_f$  and minus the  $y$ -component of  $p_{P_i}^b$ ;
- Along the  $y$ -axis: each entry is given by the product of  $c_f$  and the  $x$ -component of  $p_{P_i}^b$ ;
- Along the  $x$ -axis: the value is either  $c_m$  or  $-c_m$  depending on the sign of the corresponding  $k_i$  constant.

Once these considerations have been made, the allocation matrix can be computed, and its expression is given by:

$$G_q = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -c_f & -c_f \\ -lc_f s_{\alpha_1} & -lc_f s_{\alpha_2} & -lc_f s_{\alpha_3} & -lc_f s_{\alpha_4} & -lc_f s_{\alpha_5} & -lc_f s_{\alpha_6} & -lc_f s_{\alpha_7} & -lc_f s_{\alpha_8} \\ lc_f c_{\alpha_1} & lc_f c_{\alpha_2} & lc_f c_{\alpha_3} & lc_f c_{\alpha_4} & lc_f c_{\alpha_5} & lc_f c_{\alpha_6} & lc_f c_{\alpha_7} & lc_f c_{\alpha_8} \\ \pm c_m & \pm c_m \end{bmatrix}$$

By considering the values of the angles  $\alpha_i$ , as reported in the following table:

$i$	$\alpha_i$	$\cos \alpha_i$	$\sin \alpha_i$	Drag
1	$0^\circ$	1	0	$c_m$
2	$45^\circ$	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{2}}{2}$	$-c_m$
3	$90^\circ$	0	1	$c_m$
4	$135^\circ$	$-\frac{\sqrt{2}}{2}$	$\frac{\sqrt{2}}{2}$	$-c_m$
5	$180^\circ$	-1	0	$c_m$
6	$225^\circ$	$-\frac{\sqrt{2}}{2}$	$-\frac{\sqrt{2}}{2}$	$-c_m$
7	$270^\circ$	0	-1	$c_m$
8	$315^\circ$	$\frac{\sqrt{2}}{2}$	$-\frac{\sqrt{2}}{2}$	$-c_m$

Table 1: Table of angles

The allocation matrix then becomes:

$$G_q = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -c_f & -c_f \\ 0 & -lc_f \frac{\sqrt{2}}{2} & -lc_f & -lc_f \frac{\sqrt{2}}{2} & 0 & lc_f \frac{\sqrt{2}}{2} & lc_f & lc_f \frac{\sqrt{2}}{2} \\ lc_f & lc_f \frac{\sqrt{2}}{2} & 0 & -lc_f \frac{\sqrt{2}}{2} & -lc_f & -lc_f \frac{\sqrt{2}}{2} & 0 & lc_f \frac{\sqrt{2}}{2} \\ c_m & -c_m & c_m & -c_m & c_m & -c_m & c_m & -c_m \end{bmatrix}$$

## Exercise 2

### Ground Effect

The ground effect refers to the aerodynamic phenomenon that occurs when an unmanned aerial vehicle (UAV) operates in close proximity to the ground, resulting in an increased lift due to the generation of an upward force. In order to analyze this effect, the aerodynamic environment is assumed to be potential, with the fluid considered inviscid, incompressible, irrational and steady.

The ground effect is modeled through the method of images, which consists in introducing virtual rotors located symmetrically beneath the ground plane, at equal but opposite distances relative to the actual rotors. The downward airflow generated by the rotors is deflected upon interaction with the ground surface and redirected upwards, leading to an increase in the overall thrust.

The ground effect for a single rotor can be expressed as:

$$\frac{u_{T,ige}}{u_{T,oge}} = \frac{1}{1 - \left(\frac{\rho}{4z}\right)^2}$$

where  $u_{T,ige}$  denotes the thrust in the presence of ground effect and  $u_{T,oge}$  the thrust in free air, unaffected by ground proximity. The ground effect is considered negligible when the rotor operates at a height exceeding one rotor diameter above the ground. In configurations involving multiple rotors, the respective ground effects may combine, extending the phenomenon's influence to greater altitudes.

### Ceiling Effect

The ceiling effect arises when a multi rotor approaches an overhead surface (such as a bridge or tunnel) from below. As the UAV nears the ceiling, a notable increase in lift occurs due to the creation of a vacuum region above the propellers. This vacuum reduces aerodynamic drag, allowing the propellers to accelerate and thereby generate additional lift.

For a single rotor, the ceiling effect can be modeled:

$$\frac{u_{T,ice}}{u_{T,oce}} = \frac{1}{1 - \left(\frac{\rho}{z+k_2}\right)^2}$$

where  $u_{T,ice}$  represents the thrust under ceiling effect and  $u_{T,oce}$  the corresponding thrust in free air.

A significant difference between ground and ceiling effects is observed during hovering flight. In the case of ground effect, varying distances between individual rotors and the ground may lead to asymmetric thrust distribution, which introduces a disturbance moment  $M_{ge}$ . This moment can contribute positively to the vehicle's stability. In contrast, under ceiling effect conditions, a disturbance moment is generated that tends to amplify attitude instabilities, thus reducing the overall stability of the multi-rotor system during hover.

## Exercise 3

The estimator operates by comparing the predicted and actual evolution of the generalized momentum. A linear low-pass filter of the form:

$$G(s) = \frac{k_0^r}{(s + k_0)^r}$$

is implemented as a recursive integrator chain of order  $r = 1$ . The final estimate of the external wrench corresponds to the last state of this cascade.

At each iteration, the algorithm performs the following steps:

- Computation of the rotation matrix  $R_b$ , the transformation matrix  $Q(\eta)$ , its derivative  $\dot{Q}$  and the Coriolis matrix  $C(\eta, \dot{\eta})$ , based on the current orientation data;
- Calculation of the generalized momentum  $q$ , using measured linear and angular velocities;
- Evaluation of the known input component of the momentum derivative, including gravity compensation, applied thrust, and nominal rotation dynamics;
- Integration of the estimation dynamics using feedback correction term:

$$\gamma^{(1)} = k_1(q - \hat{q}), \quad \gamma^{(j)} = k_j y^{(j)}, \quad j = 2, \dots, r$$

- The final estimate of the external wrench is taken as  $\hat{w}_{ext} = \gamma^{(r)}$

The primary goal of the moment-based estimator is to approximate the external wrench  $w_{ext} = [f_e; \tau_e]$  acting on the UAV using its estimate  $\hat{w}_{ext} = [\hat{f}_e; \hat{\tau}_e]$ . This is achieved without augmenting the system model with additional state variables or external sensors, allowing disturbance estimation to be performed directly from onboard state measurements.

The following graphs illustrate the system response for various values of  $r$  and  $k_0$ , accompanied by analysis to determine the most suitable parameter selection:

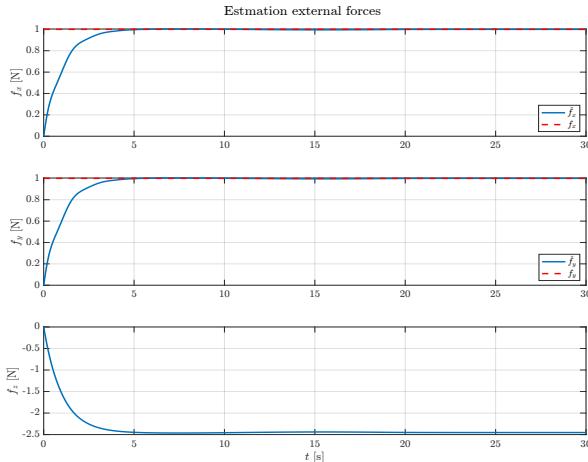


Figure 2: External forces:  $r = 1, k_0 = 1$

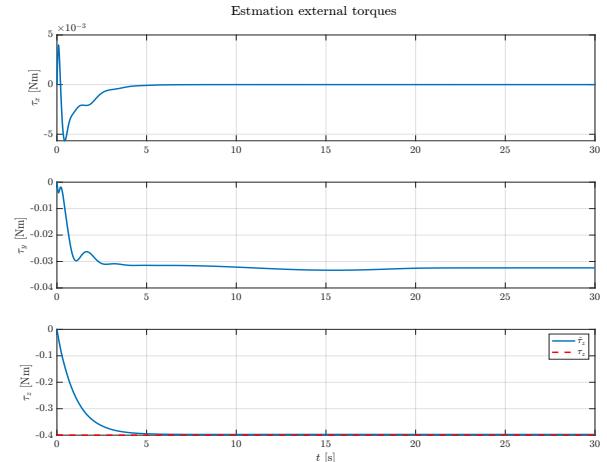
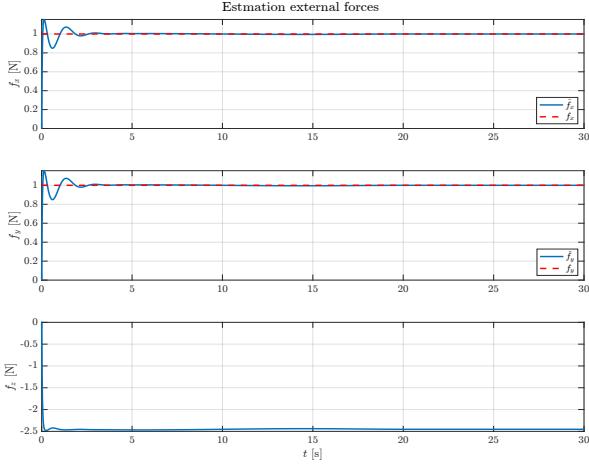
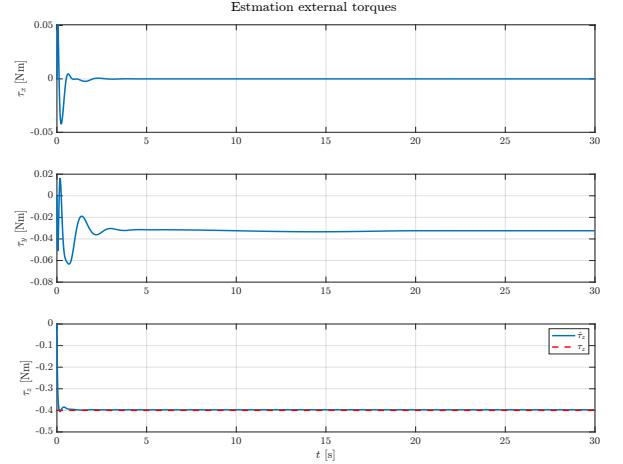
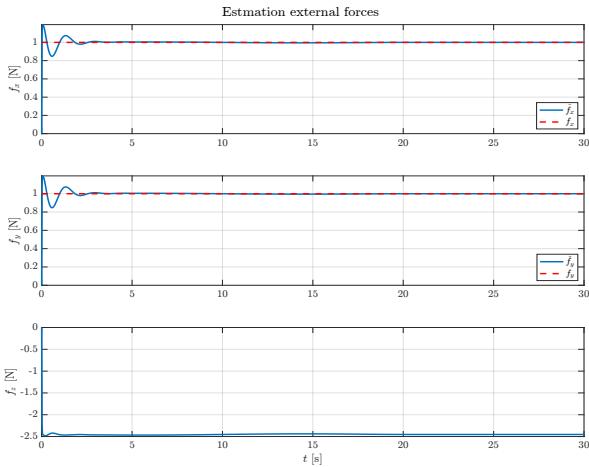
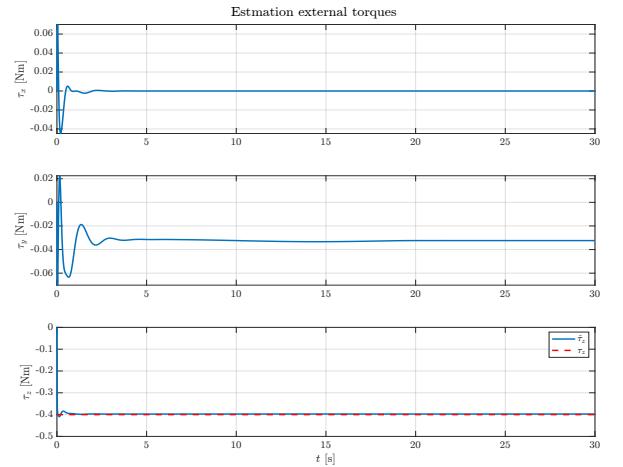
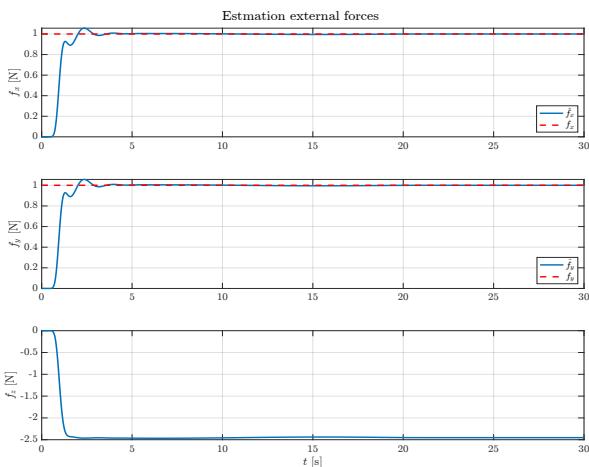
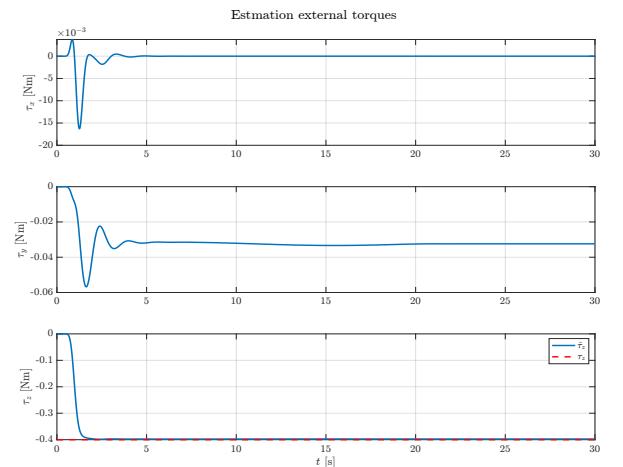
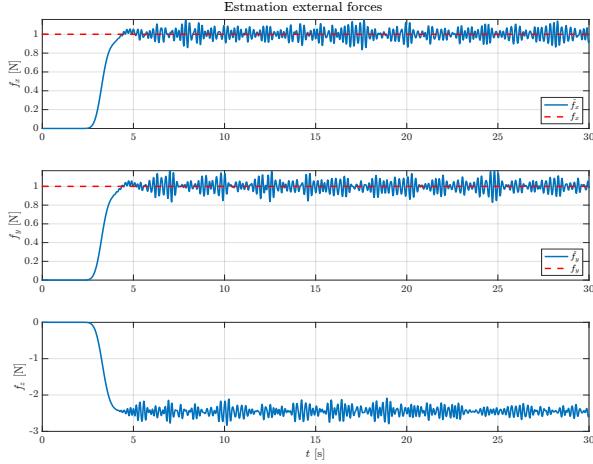
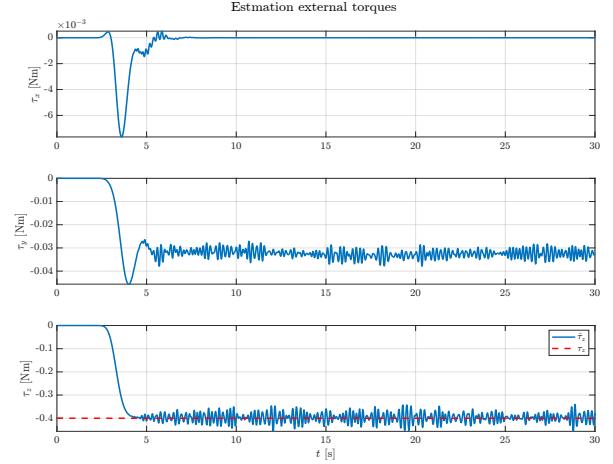


Figure 3: External torques:  $r = 1, k_0 = 1$

Figure 4: External forces:  $r = 1, k_0 = 30$ Figure 5: External torques:  $r = 1, k_0 = 30$ Figure 6: External forces:  $r = 1, k_0 = 100$ Figure 7: External torques:  $r = 1, k_0 = 100$ Figure 8: External forces:  $r = 30, k_0 = 30$ Figure 9: External torques:  $r = 30, k_0 = 30$

Figure 10: External forces:  $r = 100, k_0 = 30$ Figure 11: External torques:  $r = 100, k_0 = 30$ 

Analysis of the graphs indicates that variations in  $k_0$  alone lead to nearly identical settling times across all examined cases. However, increasing the pole (i.e., selecting a faster pole) results in a higher overshoot. Therefore, a suitable compromise to avoid excessive overshoot to be  $k_0 = 30$ . With regard to the estimator order, increasing the order does not yield significant improvements compared to the case where  $r=1$ . Furthermore,, for values starting from  $r = 86$  onward, the system becomes unstable (to better illustrate the instability, the case with  $r = 100$  is shown in 10 and 11). It is also noted that higher values of  $r$  introduce greater delay due to the presence of additional integrators. Given that the disturbances to be estimated are constant, a first-order estimator ( $r = 1$ ) is considered sufficient and thus represents the optimal choice.

The evaluation of the actual mass  $m_r$  is based on the UAV's equilibrium condition. Two formulations are considered:

$$mge_3 - u_T R_b e_3 + \hat{f}_z = 0$$

$$m_r ge_3 - u_T R_b e_3 = 0$$

The first equation describes the equilibrium under the assumption of an estimated mass that differs from the actual mass, which results in the presence of an external disturbance. The second equation represents the ideal case where the real mass is used, thereby eliminating any disturbance. By combining these two expressions, the relationship between the estimated mass and the real mass is obtained as follows:

$$m_r = m + \frac{\hat{f}_z}{g} = 1.25 \text{ kg}$$

## Exercise 4

In this section, the implementation of geometric control algorithm for a UAV is addressed. The main objective consists in completing both the inner and outer control loops on order to enable accurate tracking of the UAV's position and orientation. The flat outputs of the system—namely, the desired position and yaw angle—are provided by a trajectory planner. The planner outputs the desired body frame position  $p_{b,d}$ , desired linear velocity  $\dot{p}_{b,d}$ , desired linear acceleration  $\ddot{p}_{b,d}$ , and the direction of the desired body x-axis  $x_{b,d}$ , which is determined by the reference yaw angle.

Following the implementation of the required functions in MATLAB, control parameter tuning is performed, beginning with the inner loop (orientation control), followed by the outer loop (position control).

### Inner Loop

The inner loop controller is responsible for regulating the UAV's orientation to track the desired angular attitude. The orientation control begins with the computation of the desired  $x$ -axis of the body frame, which is derived from the reference yaw angle  $\phi_d$ :

$$x_{b,d} = \begin{bmatrix} \cos(\phi_d) \\ \sin(\phi_d) \\ 0 \end{bmatrix}$$

Specifically, the desired  $x_{b,d}$  is defined in the horizontal plane, with its components given by the cosine and sine of the yaw angle, and a zero vertical component. To ensure that the rotation matrix remains orthonormal, the desired  $y$ -axis,  $y_{b,d}$  is computed via cross product between the desired  $z$ -axis,  $z_{b,d}$  and the preliminary  $x$ -axis  $x_{b,d}$ . This vector is then normalized, and the  $x$ -axis is subsequently redefined through the cross product of  $y_{b,d}$  and  $z_{b,d}$ :

$$y_{b,d} = \frac{S(z_{b,d})x_{b,d}}{\|S(z_{b,d})x_{b,d}\|}$$

$$x_{b,d} = S(y_{b,d})z_{b,d}$$

Cross products are performed using the skew-symmetric matrix representation to maintain consistency with matrix operations. The desired rotation matrix  $R_{b,d}$  is then assembled by concatenating the orthonormal vectors  $x_{b,d}$ ,  $y_{b,d}$  and  $z_{b,d}$  as its columns:

$$R_{b,d} = \begin{bmatrix} S(y_{b,d})z_{b,d} & \underbrace{\frac{S(z_{b,d})x_{b,d}}{\|S(z_{b,d})x_{b,d}\|}}_{y_{b,d}} & z_{b,d} \end{bmatrix}$$

With this matrix, the orientation error  $e_R$  is calculated using the difference between the current and desired attitude, mapped through the vee operator to obtain a 3D vector. Additionally, the angular velocity error  $e_\omega$  is computed as the difference between the measured angular velocity and the desired angular velocity, transformed into the current body frame:

$$e_R = \frac{1}{2} (R_{b,d}^T R_b - R_b^T R_{b,d})^\vee$$

$$e_\omega = \omega_b^b - R_b^T R_{b,d} \omega_{b,d}^{b,d}$$

The control torque  $\tau^b$  is obtained by combining a proportional-derivative (PD) control law applied to the orientation and the angular velocity errors with dynamic compensation terms involving the inertia matrix and the skew-symmetric representation of the angular velocity. The rotational gain matrices  $K_R$  and  $K_\omega$  are selected to ensure fast and stable convergence of the attitude tracking:

$$\tau^b = -K_R e_R - K_\omega e_\omega + S(\omega_b^b) I_b \omega_b^b - I_b \left( S(\omega_b^b) R_b^T R_{b,d} \omega_{b,d}^{b,d} - R_b^T R_{b,d} \dot{\omega}_{b,d}^{b,d} \right)$$

## Outer Loop

The outer loop governs the translational motion of the UAV and is designed to ensure that the vehicle follows a desired trajectory in space, as defined by the reference position  $p_{b,d}$ , velocity  $\dot{p}_{b,d}$ , acceleration  $\ddot{p}_{b,d}$ , and the yaw angle  $\phi_d$ . The controller operates by computing the thrust force required to achieve the desired acceleration while compensating for gravitational effects and tracking errors. The translational errors in position and velocity are first evaluated by comparing the current and desired states:

$$e_p = p_b - p_{b,d}$$

$$\dot{e}_p = \dot{p}_b - \dot{p}_{b,d}$$

These errors are then used in a PD control formulation to compute a virtual force vector that, together with gravity and feedforward acceleration terms, determines the direction and magnitude of the desired thrust. The total thrust  $u_T$  is calculated as the projection of this control force onto the body frame's vertical axis  $e_3$ , transformed by the current rotation matrix  $R_b$ . To account for the typically faster dynamics along the vertical axis, the gain values associated with the z-component in the matrices,  $K_p$  and  $K_v$ , are set higher than those of the horizontal components:

$$u_T = -(-K_p e_p - K_v \dot{e}_p - mge_3 + m\ddot{p}_{b,d})^T R_b e_3$$

The desired direction of the body  $z$ -axis  $z_{b,d}$  is then determined by normalizing the net force vector obtained from the position controller. This vector provides the reference orientation needed to reconstruct the full desired rotation matrix used by the inner loop:

$$z_{b,d} = -\frac{-K_p e_p - K_v \dot{e}_p - mge_3 + m\ddot{p}_{b,d}}{\| -K_p e_p - K_v \dot{e}_p - mge_3 + m\ddot{p}_{b,d} \|}$$

The results of the implementation are presented below:

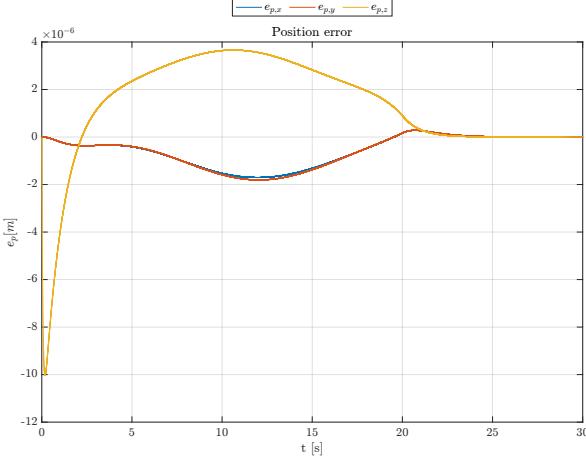


Figure 12: Position error  $e_p$

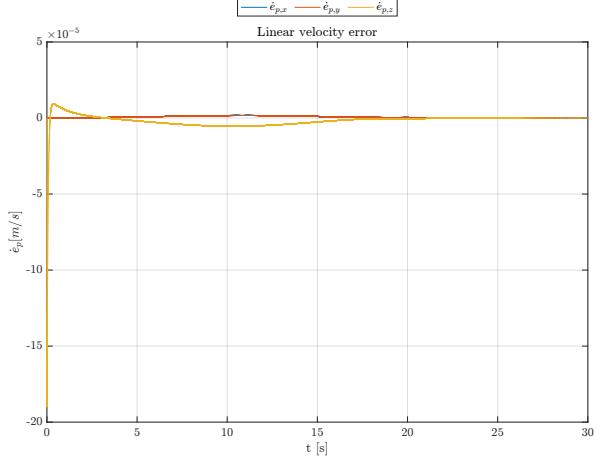


Figure 13: Linear velocity error  $\dot{e}_p$

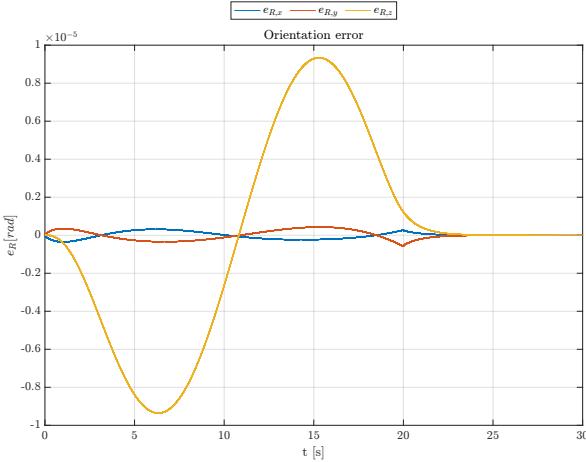


Figure 14: Orientation error  $e_R$

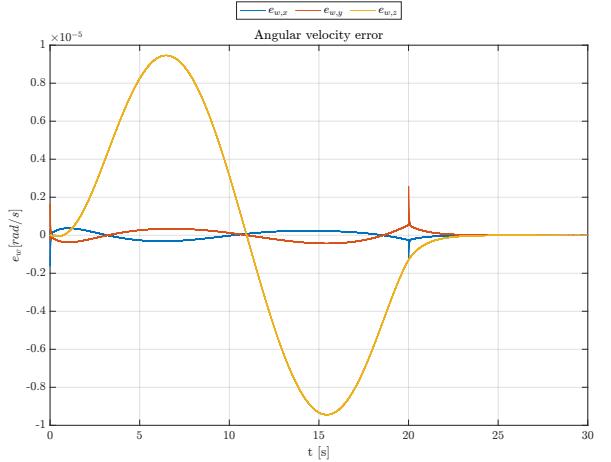


Figure 15: Angular velocity error  $e_\omega$

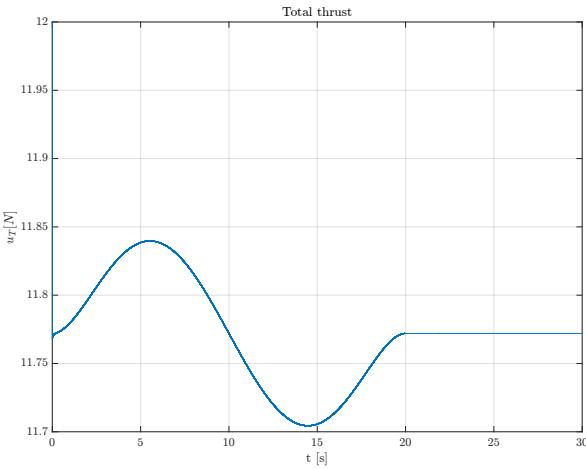


Figure 16: Total thrust  $u_T$

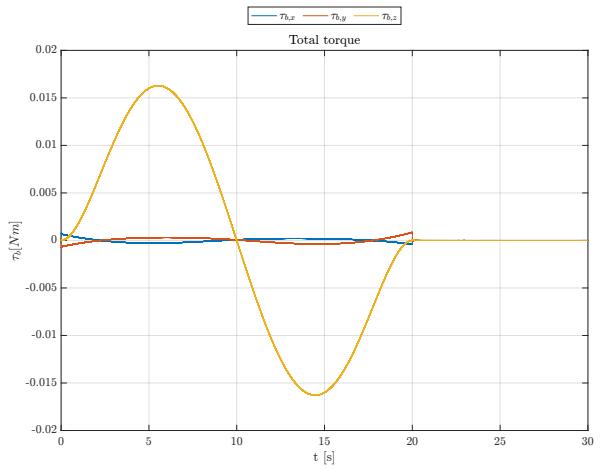


Figure 17: Total torque  $\tau^b$

## Exercise 5

In this implementation, a tilting control strategy based on the Voliro approach is adopted to enable full-actuation for the UAV platform. The primary goal is to design a controller that directly tracks the desired translational and rotational dynamic without extending the system's state with higher order derivatives (e.g., jerk or tilt angular velocities). The control design is formulated to compute the required body force  $f^b$  and torque  $\tau^b$  to achieve the desired behavior. These quantities are obtained through a combination of feedback linearization and a proportional-derivative (PD) control law, which operates on the errors in position, linear velocity, orientation and angular velocity:

$$f^b = R_b^{-1} (m(-K_p e_p - K_v \dot{e}_p + \ddot{p}_{b,d} - mge_3)$$

$$\tau^b = I_b (-K_R e_R - K_\omega e_\omega + \dot{\omega}_b^b) + S(\omega_b^b) I_b \omega_b^b$$

Once  $f^b$  and  $\tau^b$  are computed, the corresponding actuator commands must be determined. However, due to the UAV's tilting rotor configuration, the original nonlinear allocation problem is transformed into a linear one by introducing the following variable substitution:

$$u_{v,i} = c_f u_{\omega,i} \cos(\alpha_i), \quad u_{l,i} = c_f u_{\omega,i} \sin(\alpha_i)$$

These newly defined inputs  $u_{v,i}$  and  $u_{l,i}$  are then treated as control variables. The allocation matrix becomes constant and full-rank, allowing for a straightforward linear mapping between actuator commands and desired force/torque:

$$\begin{bmatrix} f_x \\ f_y \\ f_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ -1 & 0 & -1 & 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & -l & -\frac{c_m}{c_f} & 0 & 0 & l & \frac{c_m}{c_f} \\ l & \frac{c_m}{c_f} & 0 & 0 & -l & -\frac{c_m}{c_f} & 0 & 0 \\ -\frac{c_m}{c_f} & l & \frac{c_m}{c_f} & -l & -\frac{c_m}{c_f} & l & \frac{c_m}{c_f} & -l \end{bmatrix} \begin{bmatrix} u_{v,1} \\ u_{l,1} \\ u_{v,2} \\ u_{l,2} \\ u_{v,3} \\ u_{l,3} \\ u_{v,4} \\ u_{l,4} \end{bmatrix} = G_{q,static} u(\alpha, u_\omega)$$

The required actuator inputs are then retrieved by computing the pseudoinverse of the static allocation matrix:

$$u = G_{q,static}^\dagger \begin{bmatrix} f^b \\ \tau^b \end{bmatrix}$$

Finally, the physical inputs, namely the rotor speeds  $u_{\omega,i}$  and tilting angles  $\alpha_i$  are recovered from  $u_{v,i}$  and  $u_{l,i}$  as follows:

$$u_{\omega,i} = \frac{1}{c_f} \sqrt{u_{v,i}^2 + u_{l,i}^2}, \quad \alpha_i = \text{atan2}(u_{l,i} u_{v,i})$$

This control structure allows for precise trajectory tracking and attitude control in all directions, leveraging the capabilities of a fully actuated UAV while maintaining computational simplicity and system stability. The implementation results are illustrated in the following graphs:

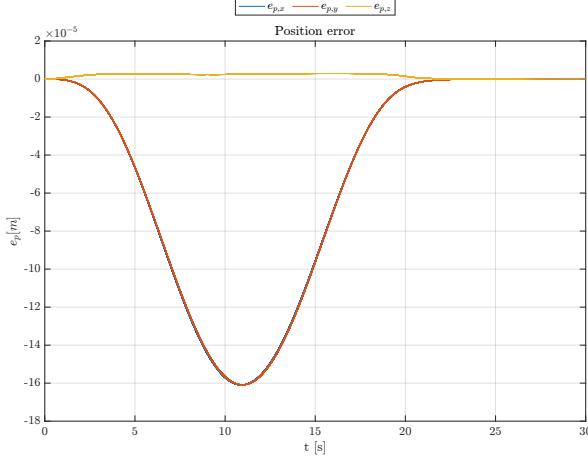


Figure 18: Position error  $e_p$

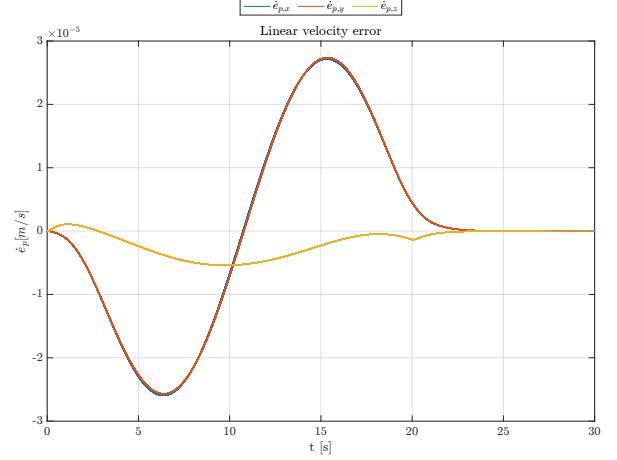


Figure 19: Linear velocity error  $\dot{e}_p$

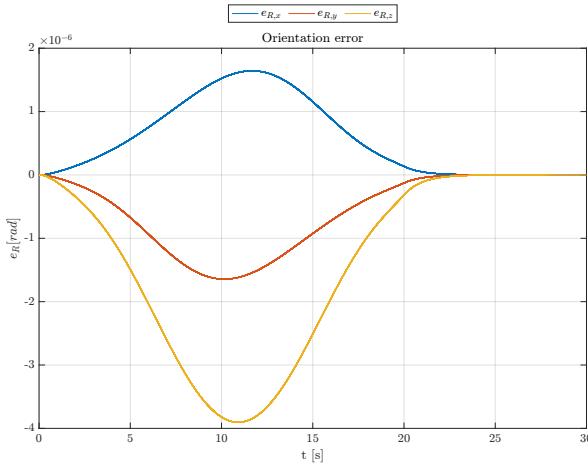


Figure 20: Orientation error  $e_R$

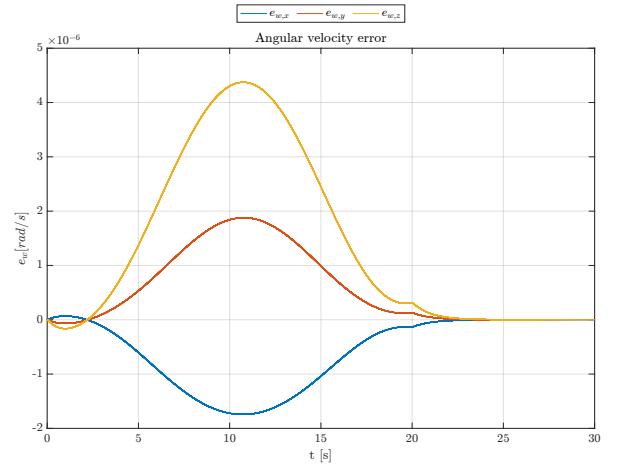


Figure 21: Angular velocity error  $e_\omega$

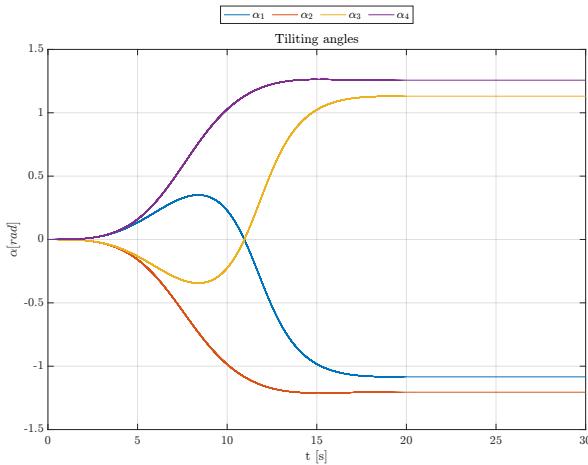


Figure 22: Tilting angles  $\alpha_i$

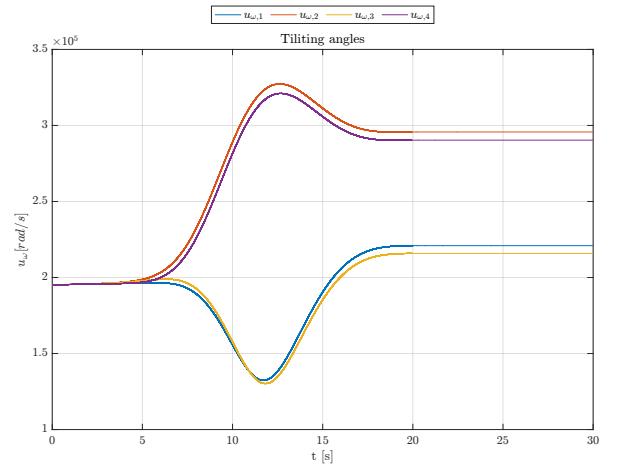


Figure 23: Rotor speed  $u_{\omega,i}$

The quadrotor adopts a coupled, asymmetric tilting configuration, resulting in a net force with vertical and horizontal components. The combination of uneven rotor speeds and non-uniform propeller inclinations generates asymmetric thrust, inducing both a horizontal force and a yaw torque. The resulting motion follows a diagonal linear trajectory inclined with respect to the vertical axis, characterized by ascent and lateral deviation. In the final stages of the simulation, the UAV hovers in the configuration it has reached, maintaining both position and attitude with high stability.

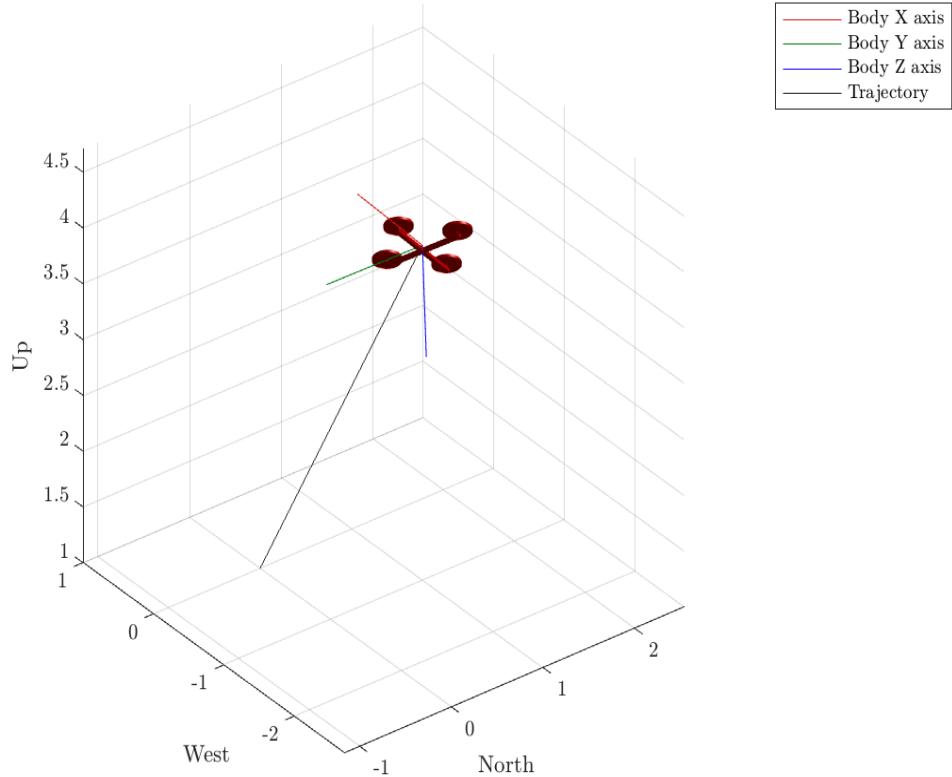


Figure 24: Quadrotor trajectory using Voliro approach