Università degli studi di
Napoli Federico II

Laurea magistrale in ingegneria dell'automazione e robotica

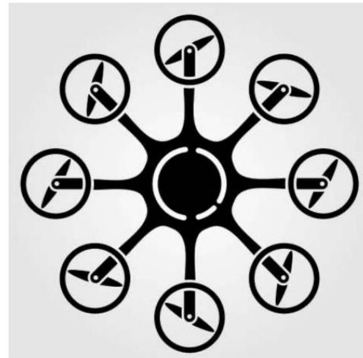
Field and service robotics

Homework 3

a.y. 2023/24

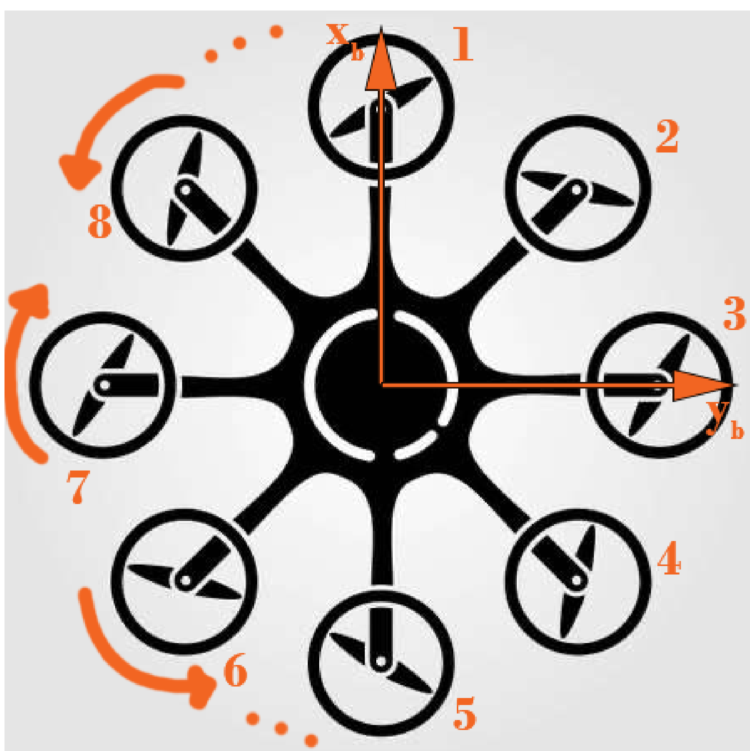
Professor: Fabio Ruggiero

Salvatore Del Peschio



1. **Given the octocopter in the figures above, write the number of degrees of freedom of the system, providing a formal description of the configuration space. Besides, is the system underactuated? Briefly motivate your answer. Finally, derive the allocation matrix for the drone, considering the top view of the octocopter given in the right image above. [Hint: put the body frame aligned with one arm of the octocopter and label the propellers (counter-)clockwise].**

The octocopter has 14 degrees of freedom while flying. Six coordinates represent the position and orientation of the drone in three-dimensional space, while the other eight represent the position of the actuators. For control purposes, we can consider only the six degrees of freedom that describe the pose of the drone in space. When the drone is fixed to the ground, it becomes an 11-DOF system if we consider its position and orientation on the ground (useful for describing the pose of the drone before liftoff or after landing), or an 8-DOF system if it is docked on a fixed base.



In the case of the robot flying, the configuration space is represented by the three-dimensional pose $\mathbb{R}^3 \times \mathbb{S}^2 \times \mathbb{S}^1$ and eight revolute joints representing the position of the propellers $\mathbb{T}^8$. Thus, we obtain

$$\mathbb{R}^3 \times \mathbb{S}^2 \times \mathbb{S}^1 \times \mathbb{T}^8$$

If we consider, as previously described, a landed or docked drone, we obtain $\mathbb{R}^2 \times \mathbb{S}^1 \times \mathbb{T}^8$ and $\mathbb{T}^8$, respectively. The octocopter shown above has a flat configuration; therefore, it is underactuated because this configuration can apply only three rotations and one translation along the vertical direction of the body frame. It cannot generate accelerations along the plane in which its propellers are arranged.
In order to find the allocation matrix we need to connect the control inputs (thrust u_T and torques τ_x, τ_y, τ_z) with the actual inputs of the system, in our case, the velocity ω_i of each propeller. This allow us to establish the following relation

$$\begin{bmatrix} u_T \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = G_q \begin{bmatrix} \omega_1^2 & \omega_2^2 & \omega_3^2 & \omega_4^2 & \omega_5^2 & \omega_6^2 & \omega_7^2 & \omega_8^2 \end{bmatrix}^T$$

Using the NED frame with a z-axis pointing inside the page, the total thrust is given by the sum of the thrust contributions from each propeller.

$$u_T = T_1 + T_2 + T_3 + T_4 + T_5 + T_6 + T_7 + T_8$$

For the torques, we obtain

$$\tau_x = -l\sin(45°)T_2 - lT_3 - l\sin(45°)T_4 + l\sin(45°)T_6 + lT_7 + l\sin(45°)T_8$$

$$\tau_y = lT_1 + l\sin(45°)T_2 - l\sin(45°)T_4 - lT_5 - l\sin(45°)T_6 + l\sin(45°)T_8$$

$$\tau_z = -Q_1 + Q_2 - Q_3 + Q_4 - Q_5 + Q_6 - Q_7 + Q_8$$

Recalling that the relations between the thrust of a single propeller and the velocity can be approximated as

$$T_i = c_T\omega_i^2$$

$$Q_i = c_Q\omega_i^2$$

The allocation matrix can be written as

$$G_q = \begin{bmatrix} c_T & c_T & c_T & c_T & c_T & c_T & c_T & c_T \\ 0 & -l\sin(45°)c_T & -lc_T & -l\sin(45°)c_T & 0 & l\sin(45°)c_T & lc_T & l\sin(45°)c_T \\ lc_T & l\sin(45°)c_T & 0 & -l\sin(45°)c_T & -lc_T & -l\sin(45°)c_T & 0 & l\sin(45°)c_T \\ -c_Q & c_Q & -c_Q & c_Q & -c_Q & c_Q & -c_Q & c_Q \end{bmatrix}$$

2. **Briefly and qualitatively describe the main differences between the hierarchical controller, the geometric controller and the passivity-based controller presented in the course. What are the main advantages and drawbacks of each control method?**

- **Hierarchical controller**: The controller's objective is to obtain two linear subsystems, one controlling the linear part and the other for the angular part. In fact the architecture consists of two loops: an outer loop for managing the linear dynamics and an inner loop for handling the angular dynamics. This configuration allows for time-scale separation, necessary by the fact that the angular dynamics are faster than the linear ones. The controller is built on the RPY dynamic model of the robot, with the angular part being feedback linearizable. However, this representation has a known singularity at $\theta = \pm\frac{\pi}{2}$ that can results in diverging control actions. Therefore, this controller is not suitable for acrobatic flights. This control strategy uses partial-feedback linearization of the dynamic model, where the fully actuated angular part is feedback linearized, but the underactuated linear part is not. Consequently, this can lead to robustness issues when the parameters are not well-known or change during the flight.
- **Geometric controller**: Starting from a coordinate-free dynamic model we obtain this controller. It represents orientation using rotation matrices avoiding singularity issues and enabling the controller to manage acrobatic maneuvers effectively. Similar to the hierarchical control approach, the geometric controller also employs feedback linearization, which can introduce robustness problems if the model parameters are not accurately known. The exponential stability of the closed-loop system can be proven if the initial attitude error is less than 90°.
- **Passivity-based controller**: This controller avoids feedback linearization of the angular part and offers robustness against model uncertainties by utilizing reference quantities ($\dot{\eta}_r, \ddot{\eta}_r, e_\eta, \dot{e}_\eta$ and ν_η) to design control actions for the angular part of the model. Nonetheless, It necessitates fine-tuning of the coupling factor that significantly impacts the tracking results.

3. **Briefly and qualitatively describe the differences between the ground effect and the ceiling effect.**

- **Ground effect**: is the effect that a UAV is subject to when it flies closes to the ground resulting in the drone pushed upwards. In order to analyze this effect we must adopt the so-called potential aerodynamic assumptions, treating the fluid as inviscid, incompressible, irrotational, and steady. The ground effect is modeled using the method of images envisioning virtual rotors below the ground surface at an equal and opposite distance to the real ones. The airflow redirected towards the ground is reverted and hits the frame, thereby increasing thrust. It can be shown that, in the case of a single rotor, the ground effect becomes negligible when it is more than one diameter off the ground. However, with multiple rotors, the effects accumulate and can be present even at higher altitude. The ground effect can influence heavily the behavior of the robot that can be subject to attitude perturbations when trying to tilt near the ground or approaching raised surface.
- **Ceiling effect**: is the effect that a UAV is subject to when it approach some objects from below. The lift of the propellers increase significantly close to the ceiling due to the vacuum effect that decreases the propeller drag so that rotate faster. This can enhance flight efficiency by reducing energy consumption because it is possible to develop the same thrust with less energy usage but must be taken into account to avoid crashes.

4. **Consider the workspace file attached as ws_homework_3_2024.mat. Within this file, you can find the values of a flight with a quadrotor with the commanded thrust (thrust) and torques (tau), the measured linear velocity (linear_vel), attitude expressed as Euler angles (attitude) and the time derivative of such angles (attitude_vel). The employed quadrotor has a supposed mass of** 1.5 kg**, and an inertia matrix referred to the body frame equal to** $\text{diag}\left([1.2416, 1.2416, 2^{*}1.2416]\right)$**. Implement yourself the momentum-based estimator of order** r **to estimate the external disturbances acting on the UAV during the flight. Suppose the sampling time of the estimator is equal to** 1 ms**. Compare the obtained estimation with the following disturbances applied during the flight:**

   - **0.5 N along the** x **and** y**-axis of the world frame;**
   - **0.2 Nm around the yaw axis.**

   Compare the estimation results with different values of r**. Try to answer the following questions.**

   - **From which value of** r **the estimation results do not improve too much?**
   - **Compute the real mass of the UAV from the estimated disturbance along the** z**-axis.**

Before the development of the estimator it should be noted that in its design the inverse Laplace operator is used with the hypotheses that $\begin{bmatrix} \hat{f}_e(0) \\ \hat{\tau}_e(0) \end{bmatrix}$ and $q(0)$, this means that the estimation should start before the UAV's take-off. To implement the momentum-based estimator of order r after the initialization of the parameters in the workspace we establish our r-order transfer function as

$$G_i(s) = \frac{(k_0)^r}{(s+c_0)^r} = \frac{k_0}{s^r + a_{r-1}s^{r-1} + ... + a_1 s + a_0} \quad i = 1, ..., 6$$

where $k_o = c_0$ to obtain a unit gain filter. After the extraction of the denominators coefficient from the transfer function we can obtain the values of K_i in the product

$$\prod_{i=j+1}^{r} K_i = c_j \quad j = 0, ..., r-1$$

The estimator has been implemented utilizing the following formulas to compute γ at every time step where the integral is carried out with the Euler method

$$\gamma_1(k+1) = \gamma_1(k) + K_1\left(q(k+1) - q(k) - T_s\left(\begin{bmatrix} \hat{f}_e(k) \\ \hat{\tau}_e(k) \end{bmatrix} + \begin{bmatrix} mge_3 - u^T(k)R_b(k)e_3 \\ C^T(k)\dot{\eta}_b(k) + Q^T(k)\tau^b(k)] \end{bmatrix}\right)\right)$$

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

Let us proceed to analyze the results with different values of r and c_0 As we expected the estimator has a faster transient with an higher c_0 converging in a shorter time to the correct values but it present a considerable overshoot.

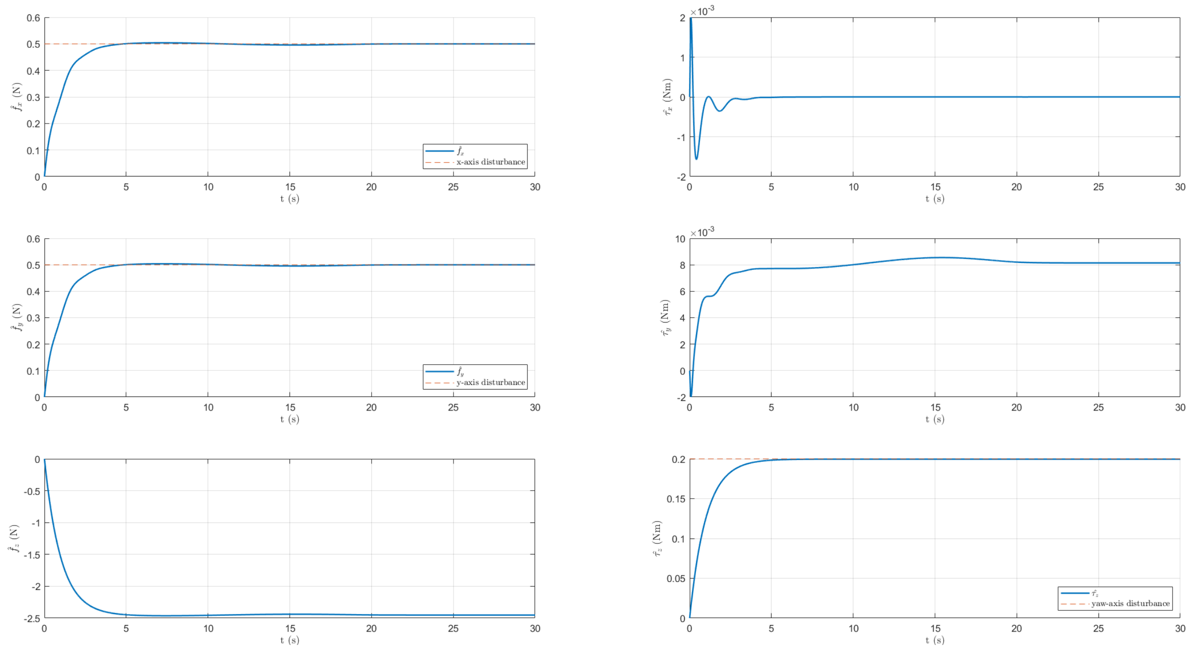


Figure 1: External wrench ($r = 1$, $c_0 = 1$)

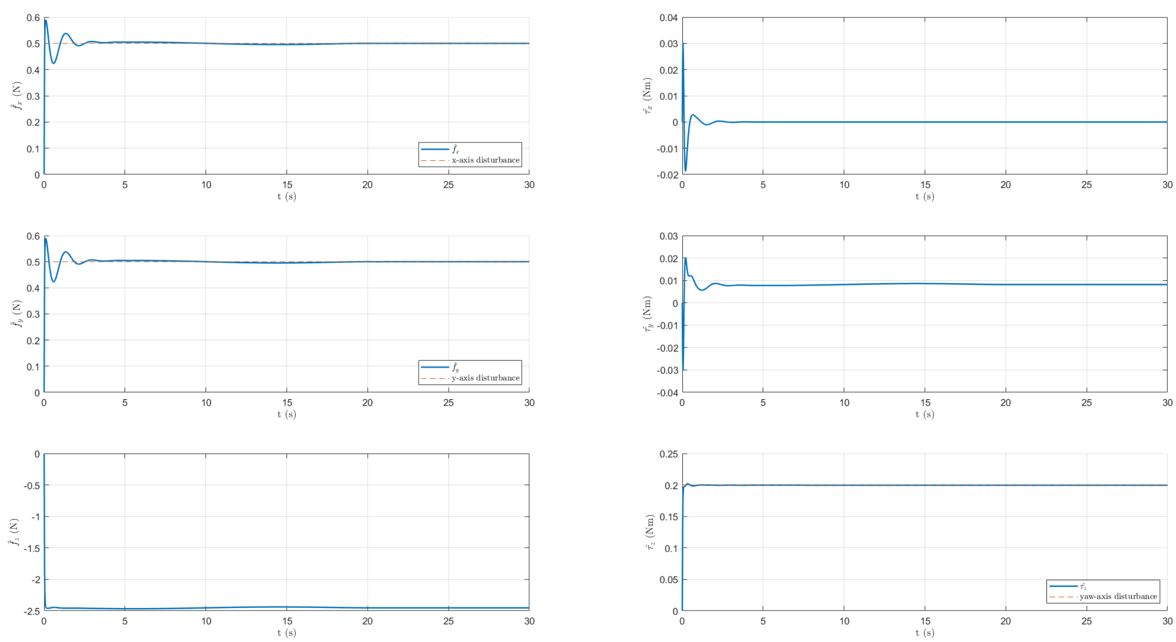


Figure 2: External wrench ($r = 1$, $c_0 = 50$)

Increasing the order of our estimator we are increasing the number of integrator. Therefore, we have an increasing delay in the transient of our estimator, this can be observed in the graphs with $c_0 = 1$. Too much delay must be avoided since the estimator should be the fastest subsystem, so it is best to use the minimum order that ensure the right estimation; since we have a constant disturbance, a first order filter should be sufficient. We can conclude that in this case a first order estimator can give us a good result and an higher value of r does not improve too much the behavior.

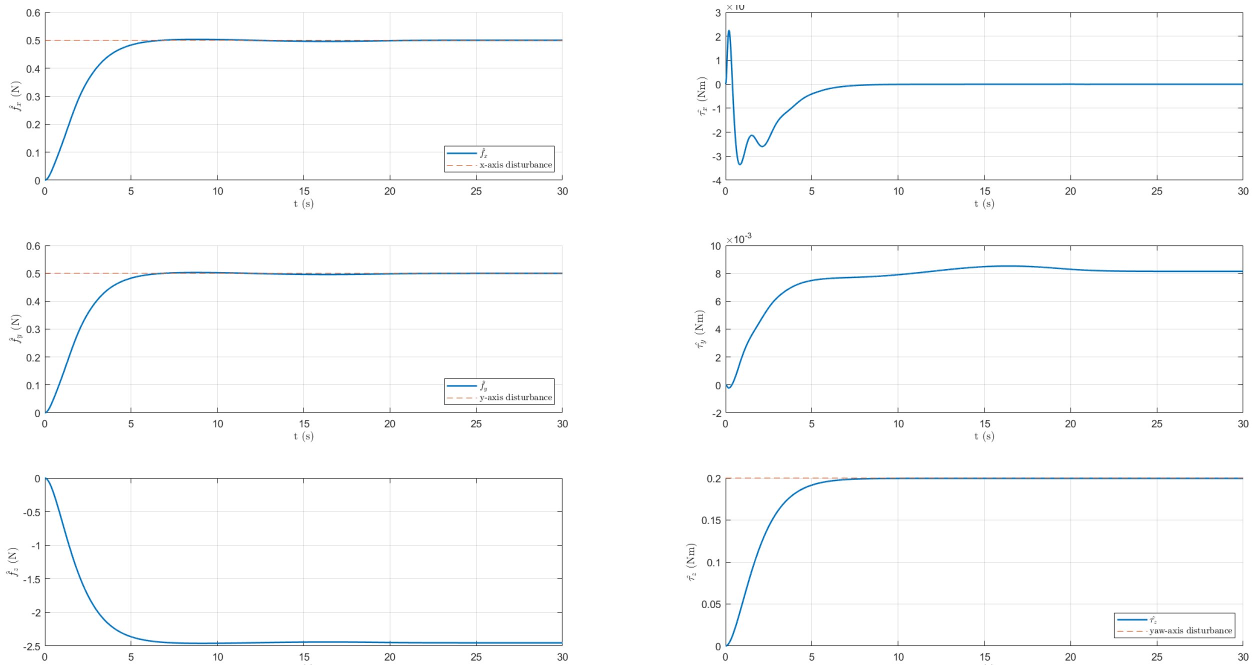


Figure 3: External wrench ($r = 2$, $c_0 = 1$)

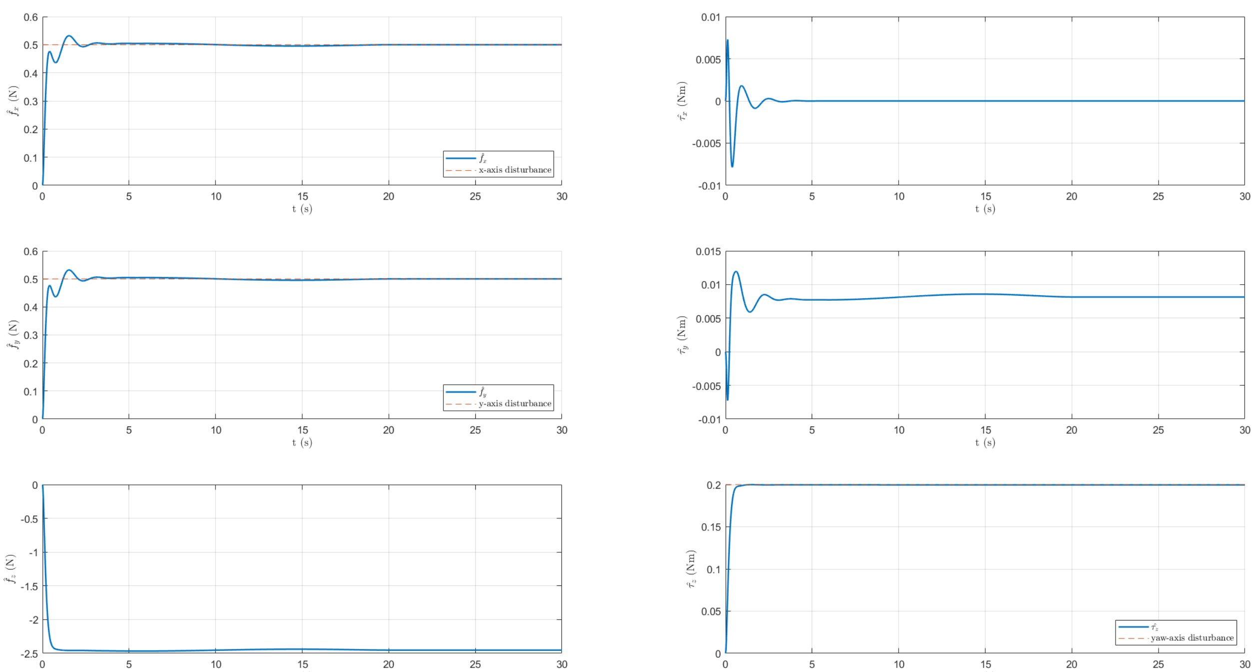


Figure 4: External wrench ($r = 2$, $c_0 = 10$)

To compute the real mass of the UAV we can consider that at the equilibrium we do not have disturbance force along the z-axis, so we could conclude that any disturbance is due to uncertainties in the mass. If $\overline{m} = 1.5$kg is the nominal mass, once obtained the value of $\hat{f}_z = \tilde{m}g$ we can compute the real mass as $m = \overline{m} - \tilde{m} = 1.25$kg.

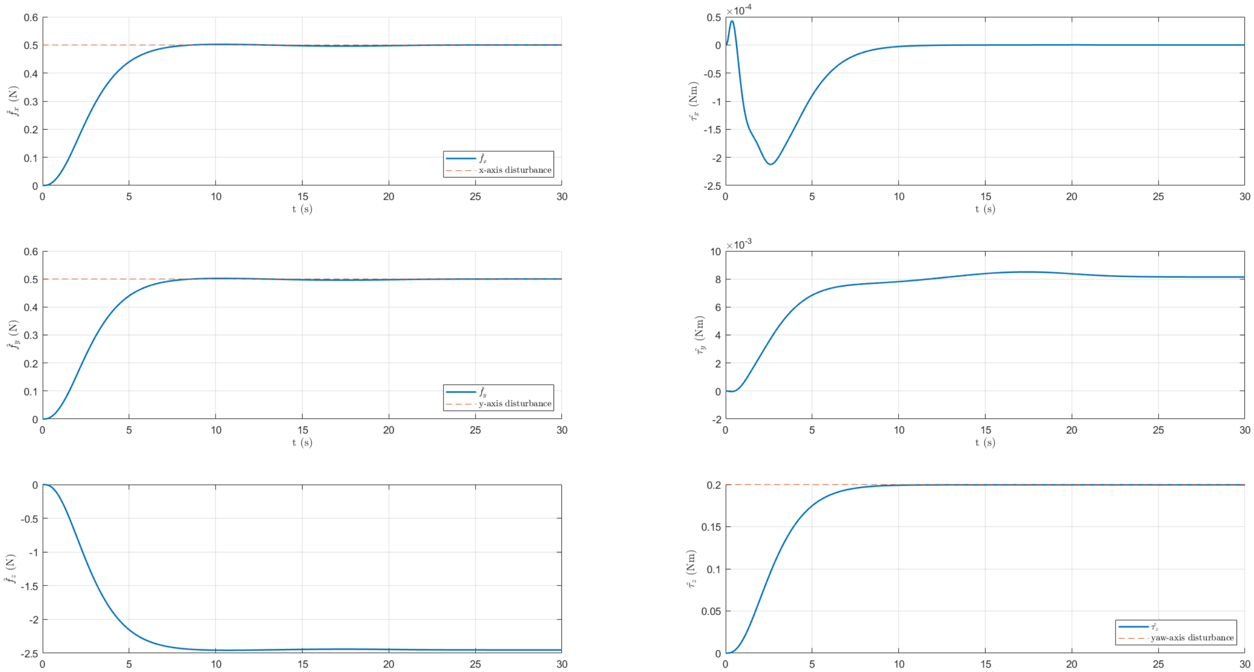


Figure 5: External wrench ($r = 3$, $c_0 = 1$)

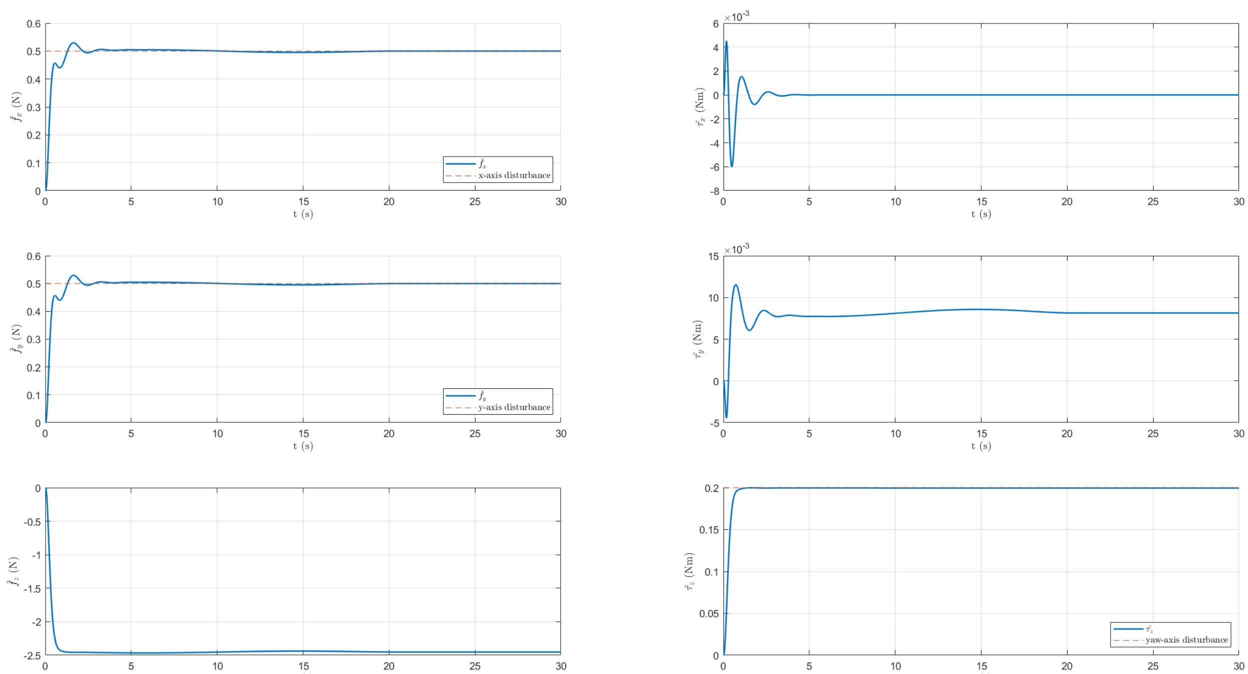


Figure 6: External wrench ($r = 3$, $c_0 = 10$)

5. **Consider the Simulink file attached as geometric_control_template.slx. Within this file, you can find a template to implement yourself the geometric control. You must fill in the inner and outer loops. Simulate the scheme and report the plots you believe are most interesting. You may add further scopes to the scheme to extract data you believe most interesting to show.**

To implement the geometric control we begin filling the outer loop, after the initialization of K_p and K_v we compute the position error and its time derivative as

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

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

then, we compute the total thrust u_T and the desired z-axis of the body frame $z_{b,d}$ as

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

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

Proceeding to the inner loop, first we initialize K_R and K_ω, subsequently $z_{b,d}$ has been used to obtain the desired rotation matrix $R_{b,d}$ with the following relations

$$R_{b,d} = \begin{bmatrix} S\left(y_{b,d}\right) z_{b,d} & y_{b,d} & z_{b,d} \end{bmatrix}$$

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

Using the desired angular velocity expressed in the body frame $\omega_{b,d}^{b,d}$ and obtaining from the sensors the readings necessary to extract the current rotation matrix R_b and current angular velocity ω_b^b we are able to compute the rotational error err_R and the angular velocity error err_ω both expressed in $SO(3)$

$$e_R = \frac{1}{2}\left(R_{b,d}^T R_b - R_b^T R_{b,d}\right)^\vee$$
$$e_\omega = \omega_b^b - R_b^T R_{b,d} \omega_{b,d}^{b,d}$$

Finally, we use the errors to attain the inner loop control which is used as input together with the total thrust to simulate the quadrotor dynamics

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

It is possible to examine the execution of the trajectory thanks to the UAV toolbox. Choosing the following gains

$$K_p = diag\begin{pmatrix} 100 & 100 & 1000 \end{pmatrix}$$

$$K_v = diag\begin{pmatrix} 10 & 10 & 50 \end{pmatrix}$$

$$K_R = diag\begin{pmatrix} 100 & 100 & 700 \end{pmatrix}$$

$$K_\omega = diag\begin{pmatrix} 10 & 10 & 30 \end{pmatrix}$$

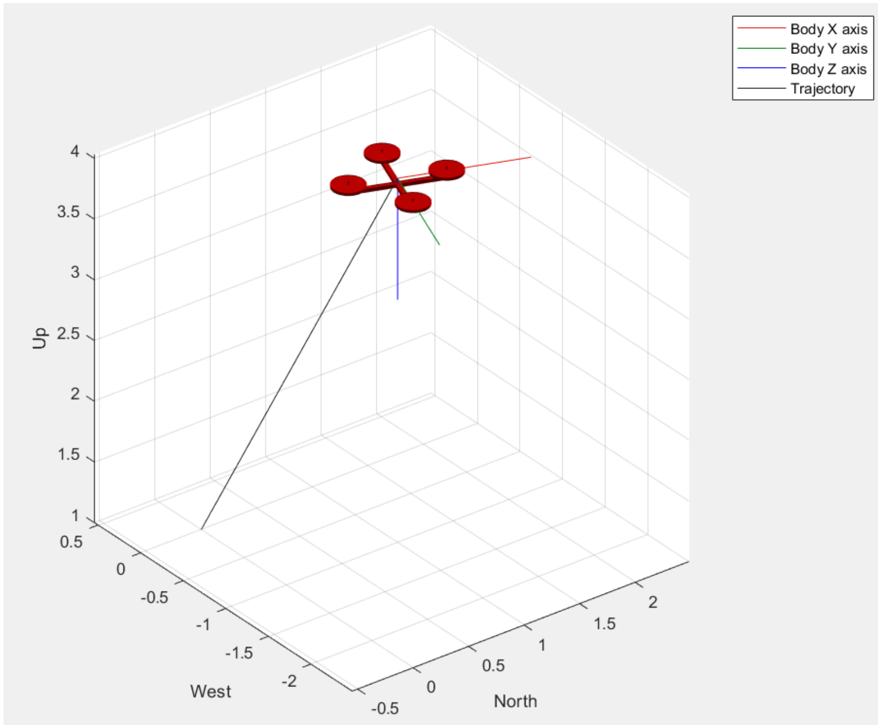
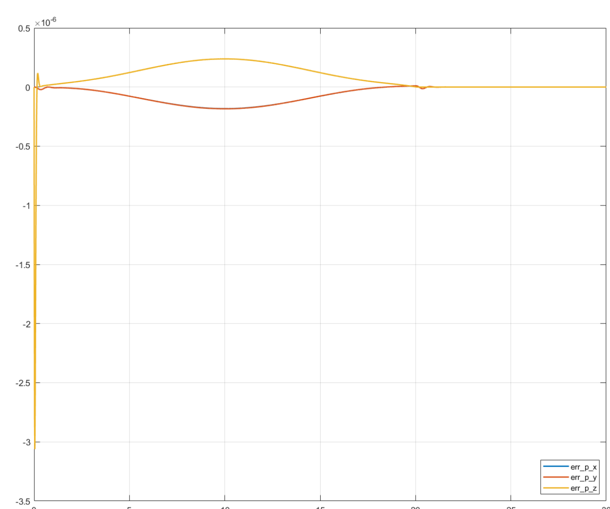
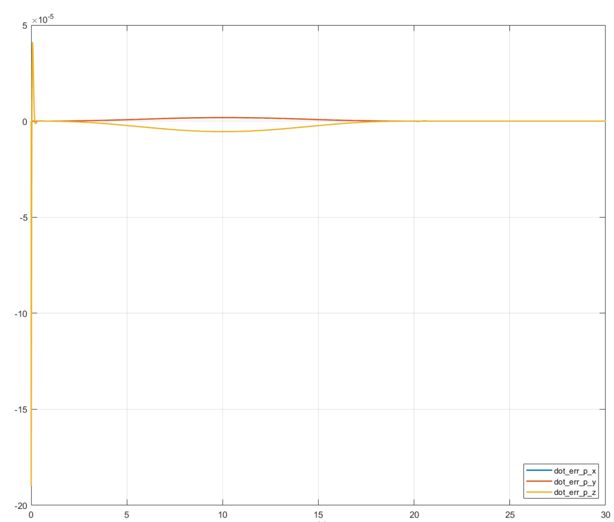


Figure 7: UAV simulator

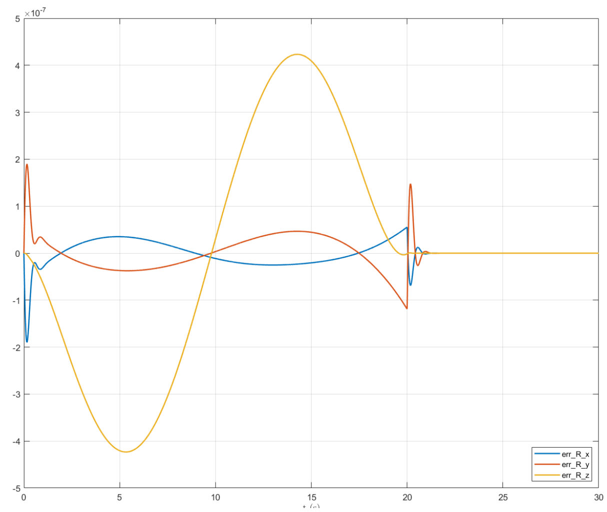
We can observe an error in the order of 10^{-6} for err_p and 10^{-5} for dot_err_p with a spike at the beginning, this is caused by the initial condition $u_T = 12$ that must be mitigate as it is shown in its plot, this initial thrust creates an initial acceleration useless in order to track the trajectory. Tuning the gains is it possible to mitigate the peak at the expense of higher errors during the flight.



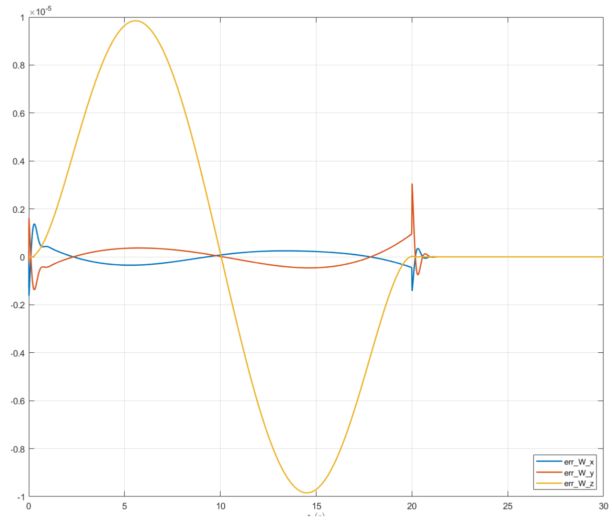
(a) err_p



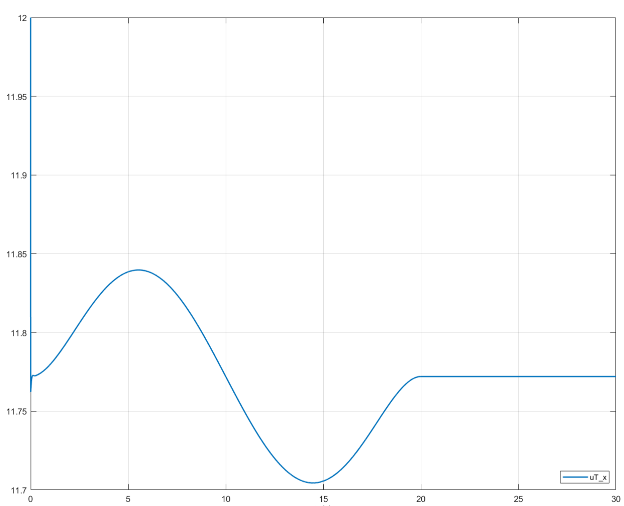
(b) dot_err_p



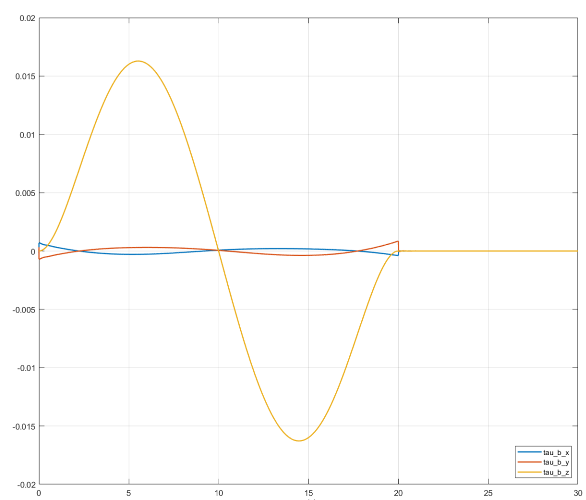
(a) err_R



(b) err_W



(a) u_T



(b) tau_b