FIELD AND SERVICE ROBOTICS

HOMEWORK 3

Student:
Paolo Maisto
P38000191

Anno Accademico 2023/2024

Answer 1:

Given the drone in the figure we can define the following parameters and then calculate the degrees of freedom:

- $m = 6$: it is a spatial mechanism;
- $N = 9$: 8 for propellers and 1 for ground;
- $J = 8$: 1 joint for each propeller;
- $\sum f_i = 8$: each propeller is a rotational joint with 1 DOF;

The robot's DOFs on the ground are:

$$\mathbf{DOFs} = \mathbf{m}(\mathbf{N} - \mathbf{J} - \mathbf{1}) + \sum \mathbf{f_i} = \mathbf{6}(\mathbf{9} - \mathbf{8} - \mathbf{1}) + \mathbf{8} = \mathbf{8}$$

Instead, when it flies, we have to add 6 DOFs given by the spatial rigid body:

$$\mathbf{DOFs} = \mathbf{8} + \mathbf{6} = \mathbf{14}$$

The configuration space of the robot on the ground is:

$$\mathbf{S^1} \times \mathbf{S^1} \times \mathbf{S^1} \times \mathbf{S^1} \times \mathbf{S^1} \times \mathbf{S^1} \times \mathbf{S^1} \times \mathbf{S^1} = \mathbf{T^8}$$

For a rigid body:

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

We can determine that the configuration space of this robot is:

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

The Octocopter under consideration is underactuated because it has 8 real inputs, that are the angular speeds for each propeller, while the number of relevant degrees of fredoom is 6, so it cannot move in all directions, which means that in each configuration there will be a certain acceleration in a certain direction that cannot be achivied.
Moreover for a UAV with the propellers arranged in a flat setup, the allocation matrix is $G_q \in \mathbb{R}^{4 \times n}$, where n denotes the number of propellers. In this case, we have that

$n > 4$ and the propellers are co-planar, so there is redundancy in distributing velocities among all propellers. However, the system remains underactuated since the n co-planar propellers do not offer more virtual inputs than 4.

At this point the track requires the computation of the allocation matrix, so various parameters have been introduced. The real input parameters for UAVs are the angular velocities ω_i of each propeller, while the control parameters are:

- The total thrust u_T, which represents the sum of the thrusts generated by all propellers, where each propeller produces a thrust $T_i > 0$:

$$u_T = \sum_{i=1}^{8} T_i = T_1 + T_2 + T_3 + T_4 + T_5 + T_6 + T_7 + T_8$$

- The control torque τ^b:

$$\tau^b = \begin{bmatrix} \tau_x & \tau_y & \tau_z \end{bmatrix}^T$$

In which the relationship between the thrust of individual propellers and the velocity is approximated by:

$$T_i = c_T \omega_i^2$$
$$Q_i = c_Q \omega_i^2$$

Where:

- $c_T > 0$ thrust constant;
- $c_Q > 0$ drag factor;

But it must be emphasized that this approximation is true if we are far from obstacles and considering a quasi-static analysis. Therefore, the sought relationship between the control parameters and the actual system parameters is established by the allocation matrix as shown below:

$$\begin{bmatrix} u_T \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \text{Allocation Matrix} \times \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \vdots \\ \omega_8^2 \end{bmatrix}$$

It is important to note that this relationship overlooks the interaction between various control inputs, rotor dynamics, and other minor aerodynamic effects.

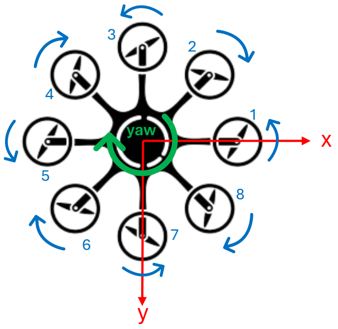


Figure 1: configuration of the drone.

In order to obtain the allocation matrix, first of all, the propellers have been numbered counterclockwise, as regards the body frame the NED convection was used in which the x axis is considered aligned with the axis of the first propeller as shown in the figure above (the axes of the body frame are represented in red, the numbering of the propellers and their direction of rotation in blue and the yaw in green).

Having defined the total thrust u_T and the control torque τ^b previously it remains to compute the individual components of the control torque as follows:

$$\begin{aligned}\tau_x = {} & lT_1\sin(0) + lT_2\sin(45) + lT_3\sin(90) + lT_4\sin(90+45) + lT_5\sin(180) \\ & + lT_6\sin(180+45) + lT_7\sin(270) + lT_8\sin(270+45)\end{aligned}$$

$$\begin{aligned}\tau_y = {} & lT_1\cos(0) + lT_2\cos(45) + lT_3\cos(90) + lT_4\cos(90+45) + lT_5\cos(180) \\ & + lT_6\cos(180+45) + lT_7\cos(270) + lT_8\cos(270+45))\end{aligned}$$

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

Where l represents the length of the bars, so we can then compute the allocation matrix:

$$\begin{bmatrix} c_T & c_T & c_T & c_T & c_T & c_T & c_T & c_T \\ lc_T\sin(0) & lc_T\sin(45) & lc_T\sin(90) & lc_T\sin(90+45) & lc_T\sin(180) & lc_T\sin(180+45) & lc_T\sin(270) & lc_T\sin(270+45) \\ lc_T\cos(0) & lc_T\cos(45) & lc_T\cos(90) & lc_T\cos(90+45) & lc_T\cos(180) & lc_T\cos(180+45) & lc_T\cos(270) & lc_T\cos(270+45) \\ Q_1 & -Q_2 & Q_3 & -Q_4 & Q_5 & -Q_6 & Q_7 & -Q_8 \end{bmatrix}$$

From which it is obtained that:

$$\begin{bmatrix} u_T \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} c_T & c_T & c_T & c_T & c_T & c_T & c_T & c_T \\ 0 & lc_T\sin(45) & lc_T & lc_T\sin(90+45) & 0 & lc_T\sin(180+45) & -lc_T & lc_T\sin(270+45) \\ lc_T & lc_T\cos(45) & 0 & lc_T\cos(90+45) & -lc_T & lc_T\cos(180+45) & 0 & lc_T\cos(270+45) \\ Q_1 & -Q_2 & Q_3 & -Q_4 & Q_5 & -Q_6 & Q_7 & -Q_8 \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \vdots \\ \omega_8^2 \end{bmatrix}$$

As you can see from the allocation matrix, negative signs are not defined in the various cases, as they will be returned directly from the sine and cosine of the respective angles.

Answer 2:

During the course the following controllers were presented: the hierarchical controller, the geometric controller and the passivity-based controller. The main characteristics and consequent advantages and disadvantages of each controller will be briefly presented below, to then describe and highlight the main common aspects and substantial differences.

- Starting from the hierarchical controller it is based on the dynamic RPY model of the UAV; therefore, from the outset it can be said that it presents the problems of singularity with regard to representation. This controller is based on simple control techniques, in fact it applies the linearization of the feedback on the angular part and then uses a PD controller with acceleration feedforward. In addition, this controller generates two control loops: the outer loop that deals with position control and the inner loop that deals with attitude control and applies feedback linearization. From the outer loop equations it is noted that the angular part is asymptotically stable; while for the linear part it is necessary to use perturbation theory to demonstrate simple stability. Summing up the main advantage of this control is first of all the use of very standard control techniques, which give the controller simplicity, as feedback linearization and PD controllers; therefore, by using RPY angles for angular error, it makes it easier to understand its physical meaning. While the main disadvantage is that since there is a singularity at the time when $\theta \neq \pm\frac{\pi}{2}$, you cannot use this controller for acrobatic flights as we are not sure that the following constraint $\theta \neq \pm\frac{\pi}{2}$ be observed throughout the flight. Similarly, an additional constraint must be imposed, that is, the third component of μ_d must be less than the modulus of the acceleration of gravity, which means that the UAV cannot accelerate to the ground more than the acceleration when it is in free fall; in the same way we want to never turn off all the propellers as it means that the drone crashes. A further disadvantage is that the linearization of the feedback imposes that the controller has problems of robustness, which in turn in order to properly operate requires perfect deletions, but to avoid this problem you can add integral actions (consequently, the addition of these integral actions may have an advantage, as they would increase the robustness of the system). A last disadvantage is that the flat outputs provide only the angles of roll and pitch, so the desired angular velocities and acceleration should be calculated numerically online, and then you need to add a second-order low-pass digital filter that slows down the controller.

- The geometric controller, however, unlike the previous controller, is based on the model of the UAV without coordinates and therefore does not suffer from any singularity. Always unlike the previous controller, e_R and e_ω errors are defined in $SO(3)$. As for the previous controller, two control loop are generated here: the inner loop that with a series of calculations allows to obtain the torque τ_b to adjust the orientation of the UAV; while the outer loop allows to obtain u_t and $z_{b,d}$. This controller ensures asymptotic stability of the system in closed circuit since it is designed to make the tracking error tend to zero when there are no orientation errors or when orientation errors are limited. Finally, here too, the constraint expressed as follows:

$$\| -mge_3 + m\ddot{p}_{b,d}\| < \text{positive constant}$$

  must be respected; this ensures that the planned desired linear acceleration does not exceed the acceleration of gravity.

In this controller we have several advantages. The first one is that not suffering from any singularity, so it is suitable for acrobatic flights; also it is possible to demonstrate exponential stability through the Lyapunov theory if the initial set-up error is less than 90 degree. Finally, taking the formula of the total thrust we find the scalar product between $z_{b,d}$ and z_b; which means that being possible to rewrite u_t proportional to the cosine between $z_{b,d}$ and z_b, it has a self-limiting effect compared to angular errors. The disadvantages, as mentioned above, are that it uses linearization of feedback and consequently presents problems of robustness; and the errors e_r and e_ω are difficult to read being defined in $SO(3)$.

- Finally, the passivity base controller is based on the dynamic RPY model of the UAV, therefore it suffers of problems of singularity of representation. It is the only controller not to use the linearization of the feedback in order to obtain a greater robustness, but consequently there is to consider that it does not have perfect deletions that characterize the linearization of the feedback. In this case there is the addition of external estimators in order to compensate unknown terms, such as external disturbances; it, however, not ensuring the asymptotic cancellation of errors, its presence is crucial because without it it would be difficult to compensate for the disturbances and optimize the gains of the controller for its optimal operation. In the passivity-based controller, the interconnection between linear and angular components depends on the current and desired speed, as well as the error of orientation is obtained through a positive coupling parameter σ. In addition, the limitation of errors can be demonstrated by Lyapunov and perturbation theory. Summing up the advantages are the lack of feedback linearization; the outer loop equations that are obtained can be seen, both for the linear and angular part, as a mass-damping-spring systems in which the spring part is fully programmable, the damping part is partially programmable and the inertia is fixed; which gives the possibility to easily adjust the bandwidth. Also the main advantage is that these equations establish a passive relationship between τ and v_η. Instead among the first disadvantages there is the problem of the singularity of representation in which also in this case assumes that $\theta \neq \pm\frac{\pi}{2}$. in addition, it may be difficult to calibrate all the existing parameters and constants, such as the coupling factor σ, which if adjusted correctly, as it affects the monitoring result, in fact a value too small can cause vibration, while for larger values the tracking improves, but can also lead to instability.

At this point presented the various controllers can be defined the main differences and similarities. First, while the geometric controller is based on the model of the UAV free of coordinates does not suffer from any singularity (so it is the only one suitable for acrobatic flight), the other two controllers suffer from singularity of representation, based on the dynamic RPY model of the UAV, which, however, in turn brings a further difference, that is, while for the geometric controller the errors defined in $SO(3)$ are more difficult to understand, unlike the other two controllers where using RPY angles for angular error allow easier reading of the latter. Moreover, about the passivity base controller it is the only one that does not have the feedback linearization, so it has greater robustness, but it is more difficult to develop. Finally, two aspects should be emphasized: that even if the estimator has not been presented and mentioned, it is possible to add it to all three controllers and not only to the passivity base one and robustness problems can be contained using the integral action.

Answer 3:

Successively will be briefly introduced the ground effect and the ceiling effect, emphasizing the advantages and disadvantages and finally show the differences between the two.

The ground effect manifests when a multirotor approaches the ground during flight. This phenomenon is caused by the interaction of the airflow produced by the rotors with the ground surface, resulting in several changes in flight dynamics:

- When the rotor gets close to the ground, the presence of the surface deflects the airflow radially and parallel to the ground, generating new velocity and pressure fields. This creates greater thrust compared to when the rotor is in free air, giving the impression that the ground "pushes up" the multirotor.

- In multirotors, the ground effect is more complex than in single rotors due to the interference between the airflows of various rotors. The airflow at the center of the space between the rotors can reverse direction after reaching the ground, hitting the multirotor frame and producing an additional force. This force must be considered in the total thrust calculation.

- A multirotor hovering close to the ground that experiences an attitude perturbation can generate a stabilizing moment. If the rotors are at different distances from the ground (as shown in the following figure), the differential thrust of the rotors creates a moment that tries to rotate the multirotor to counteract the disturbance, thus stabilizing it.

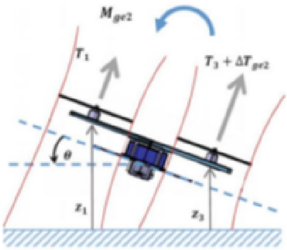


Figure 2: The hovering multirotor subject to an attitude perturbation.

- But the ground effect according to the various configurations acts differently, in fact when only some of the rotors are affected by the ground effect (as shown in the following figure), an asymmetric aerodynamic effect arises. This can create a moment that tries to rotate the multirotor, preventing stable object grasping.

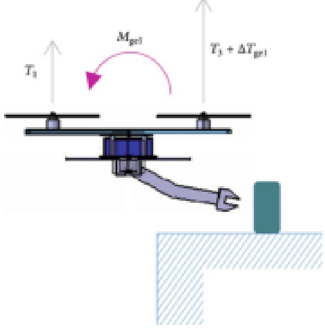


Figure 3: UAM with only a part of the rotors subjected to the ground effect.

While the ceiling effect occurs when a multirotor flies close to a surface above it, such as a ceiling or the underside of a bridge. This phenomenon significantly affects flight behavior:

- Similar to the ground effect, when a rotor approaches the ceiling, the thrust increases significantly. This is due to a vacuum effect that decreases the propeller's resistance when it is very close to the ceiling, allowing the rotor to spin faster and generate more thrust.
- In multirotors, the ceiling effect causes complex interactions between the airflow of different rotors and the multirotor frame. This creates behavior different from the simple sum of the effects of individual rotors, with the flow field influenced by mutual interference.
- The ceiling effect can offer significant advantages, such as increased stability during operations that involve contact with upper structures and an extended maximum flight time. The energy savings are particularly useful in UAV applications that require maintaining contact with the ceiling.
- It is crucial to consider the ceiling effect to avoid collisions with the upper surface. The increased thrust can push the multirotor even closer to the ceiling, increasing the risk of a crash.

Both the ground effect and the ceiling effect significantly alter the flight dynamics of multirotors, but they do so in distinct ways. The ground effect increases thrust when the rotors approach the ground, stabilizing the multirotor in the event of disturbances. The ceiling effect, on the other hand, increases thrust when the multirotor flies close to an upper surface, enhancing stability and reducing energy consumption. However, both effects require careful management to avoid potential collisions and fully exploit the operational benefits they offer.

Answer 4:

The *Es_4.m* file implements the trace's request for this point using the software Matlab. The initial step in implementing the order r estimator was to load all the parameters from the provided file and compute the necessary variables to build the estimator, including R_b (attitude), Q (matrix relating the derivatives of Euler angles to angular velocities), M (inertia matrix), q (generalized momentum), and C (matrix representing centrifugal and Coriolis forces). The formulae of the estimator used are the general expression of an order estimator r.

In order to obtain a reliable assessment of the external wrench, a Butterworth filter with a low-pass configuration was used. By utilizing the `butter` function, we acquire the coefficients essential for shaping the transfer function of the low-pass Butterworth analog filter with cutoff frequency ω_n, therefore the parameters on which we operate are r and ω_n. The Butterworth analog filter's transfer function is delineated through the coefficients of the polynomial numerator, denoted as a, and the polynomial denominator, denoted as c_j, as follows:

$$G(s) = \frac{A(s)}{C(s)} = \frac{a(1)s^n + a(2)s^{n-1} + \ldots + a(n+1)}{c_j(1)s^n + c_j(2)s^{n-1} + \ldots + c_j(n+1)}$$

Numerous tests were conducted, supposing the sampling time of the estimator is equal to 1 ms, with varying the ordere r of the estimator and the cutoff frequency ω_n. The results were compared with the disturbances applied during the flight required by the trace, from which we can observe that, just as required by the trace, the UAV is subjected to :

- A disturbance force of 0.5 N along the x-axis and y-axis of the world frame;
- A disturbance torque of 0.2 Nm around the yaw axis;

$$\begin{bmatrix} \hat{f}_e \\ \hat{\tau}_e \end{bmatrix} = \begin{bmatrix} \hat{f}_x \\ \hat{f}_y \\ \hat{f}_z \\ \hat{\tau}_x \\ \hat{\tau}_y \\ \hat{\tau}_z \end{bmatrix} = \begin{bmatrix} 0.5000 \\ 0.5000 \\ -2.4525 \\ 0 \\ 0.0081 \\ 0.1997 \end{bmatrix}$$

Upon examining the next graphs, show that: immediately after the application of a higher order estimator to the primary order estimator, there is no significant enhancement in the estimation of external disturbances. This phenomenon ensues due to the constancy of the applied disturbances, rendering a solitary integration action able in their quantification. In fact, the enhancement in estimation accuracy by increasing the estimator's order is minimal in this context because the difference between the final values of the estimates is very small. The figures demonstrate that as the estimator's order increases the response exhibit increasing delay. This is expected as it is a known effect of adding integrators, but it also suggests that using the lowest-order estimator that still provides a satisfactory estimate is likely prudent to avoid this delay.

Finally, the last six plots (figure 6) indicate that if the order becomes excessively high, the estimator start oscillating; this is likely due to the excessive delay. Therefore, we can conclude that there are no substantial benefits to increasing the estimator order beyond the first.

Below are presented the cases where $\omega_n = 1$ is fixed and r is increased gradually:

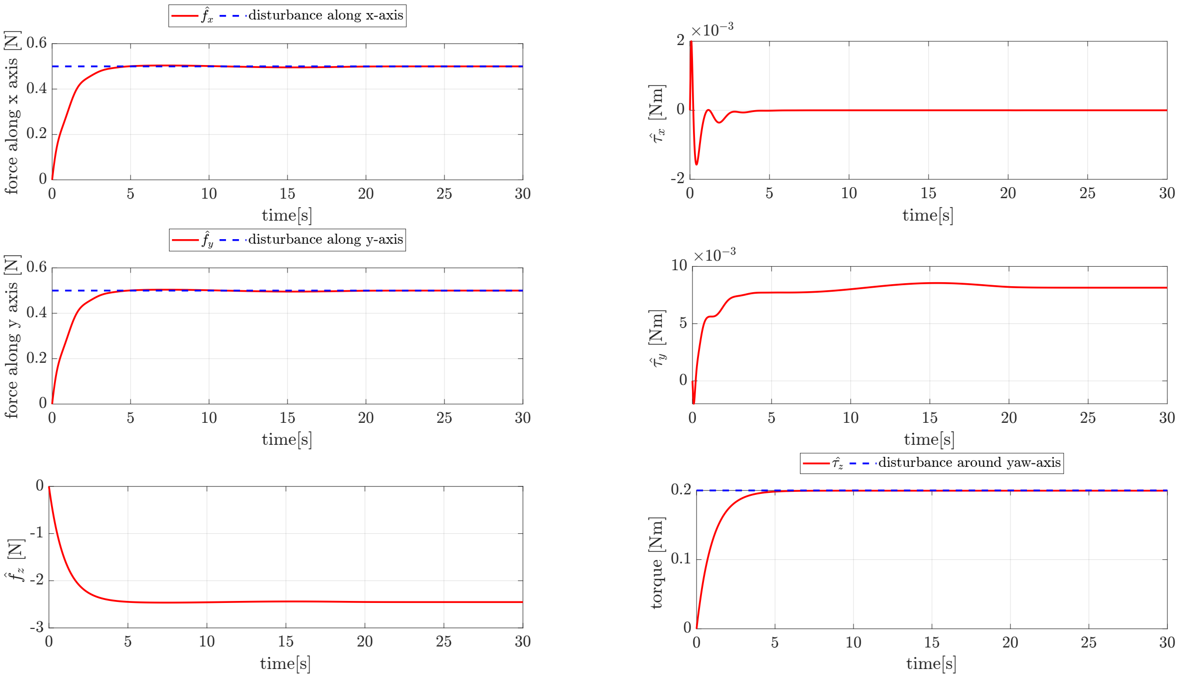


Figure 4: $r = 1$ and $\omega_n = 1$

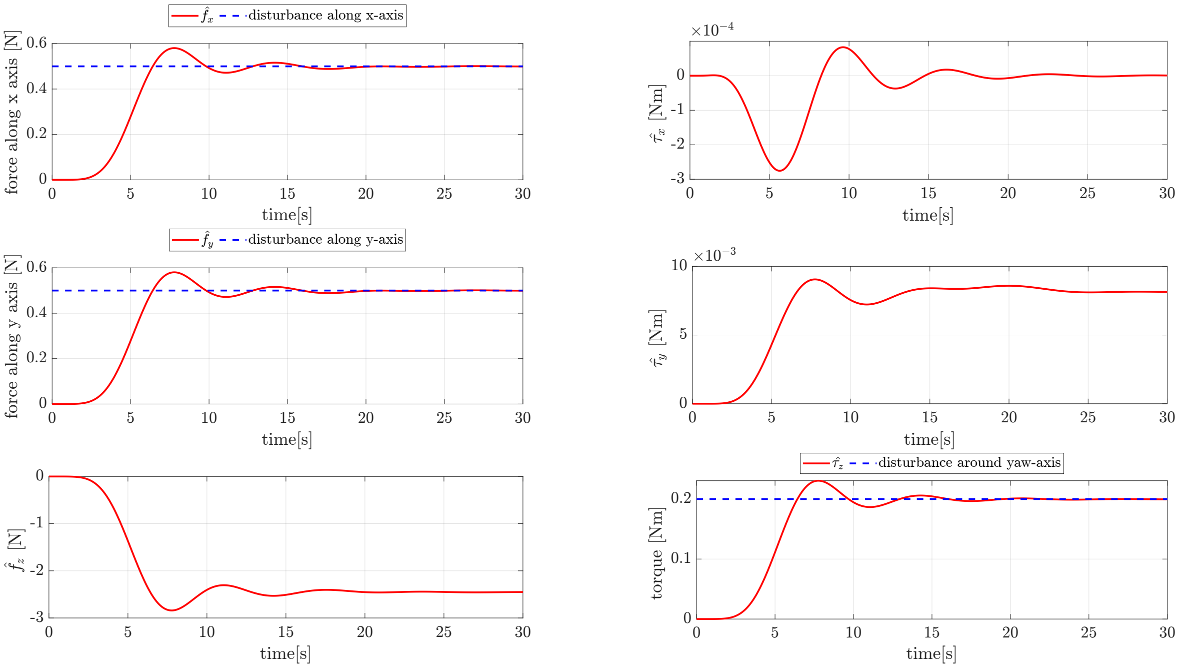


Figure 5: $r = 7$ and $\omega_n = 1$

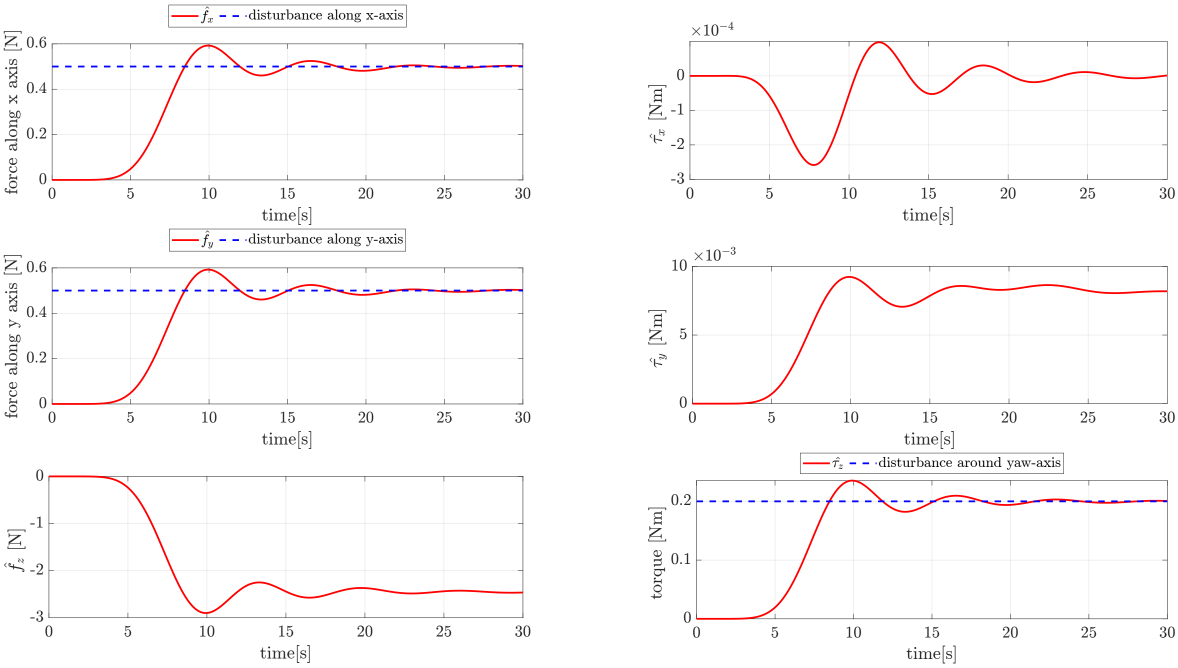


Figure 6: $r = 10$ and $\omega_n = 1$

On the other hand you can notice that, notably, adjusting the Butterworth filter's frequency ω_n enhances the controller's responsiveness, accelerating the convergence of external wrench estimates towards their steady-state values. However, this expedited response may lead to amplified overshoot tendencies. This means that looking at the following figures, in order of how they are arranged, increasing the frequency ω_n we will have a very fast transient, but this leads to a greater over-elongation; vice versa decreasing it you get a longer transient but less over-elongation, as can be seen from the first plots.

Vice versa below are presented the cases where $r = 1$ is placed and the frequency ω_n is increased gradually:

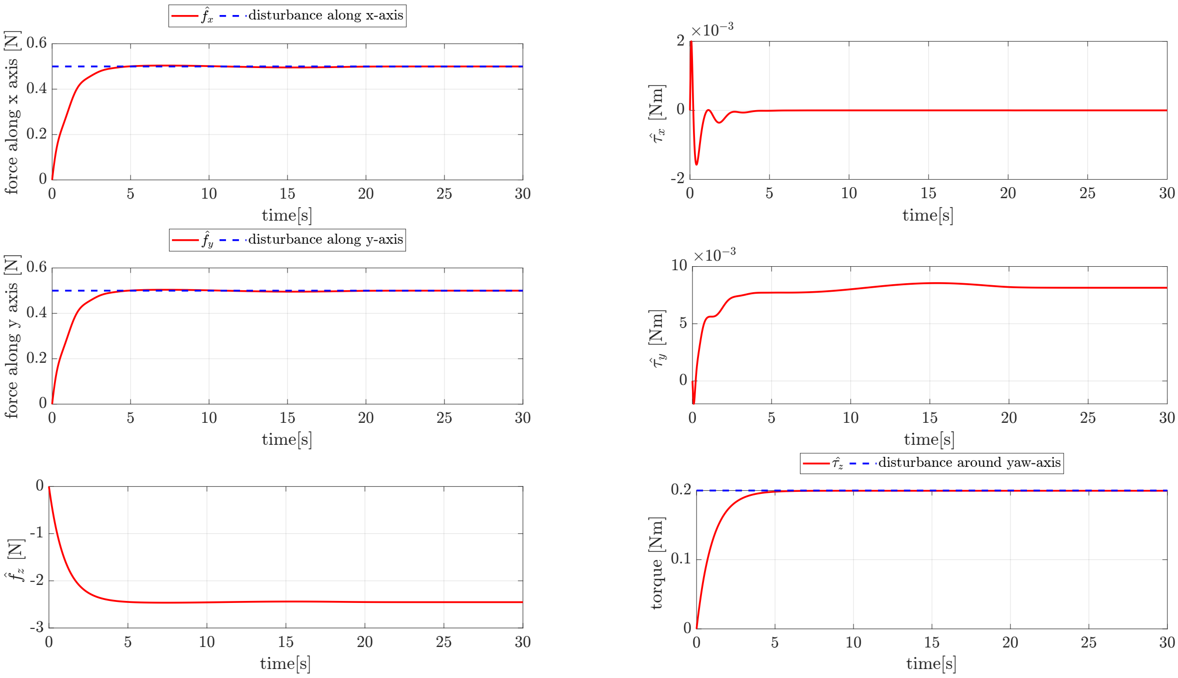


Figure 7: $r = 1$ and $\omega_n = 1$

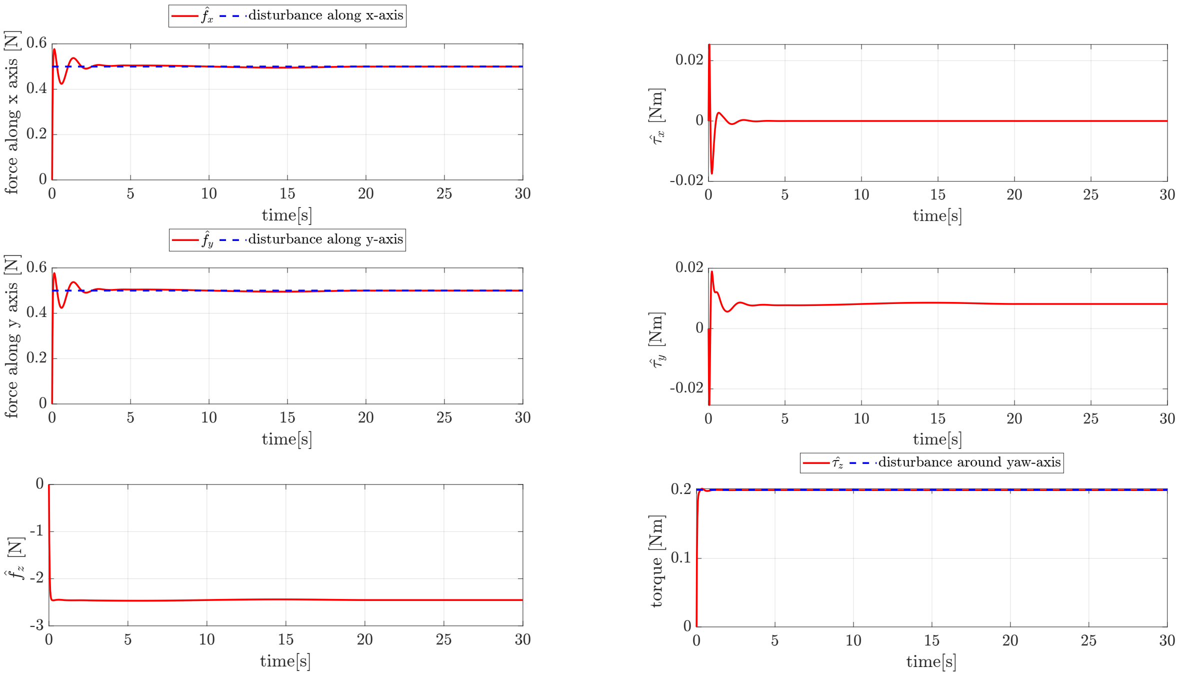


Figure 8: $r = 1$ and $\omega_n = 30$

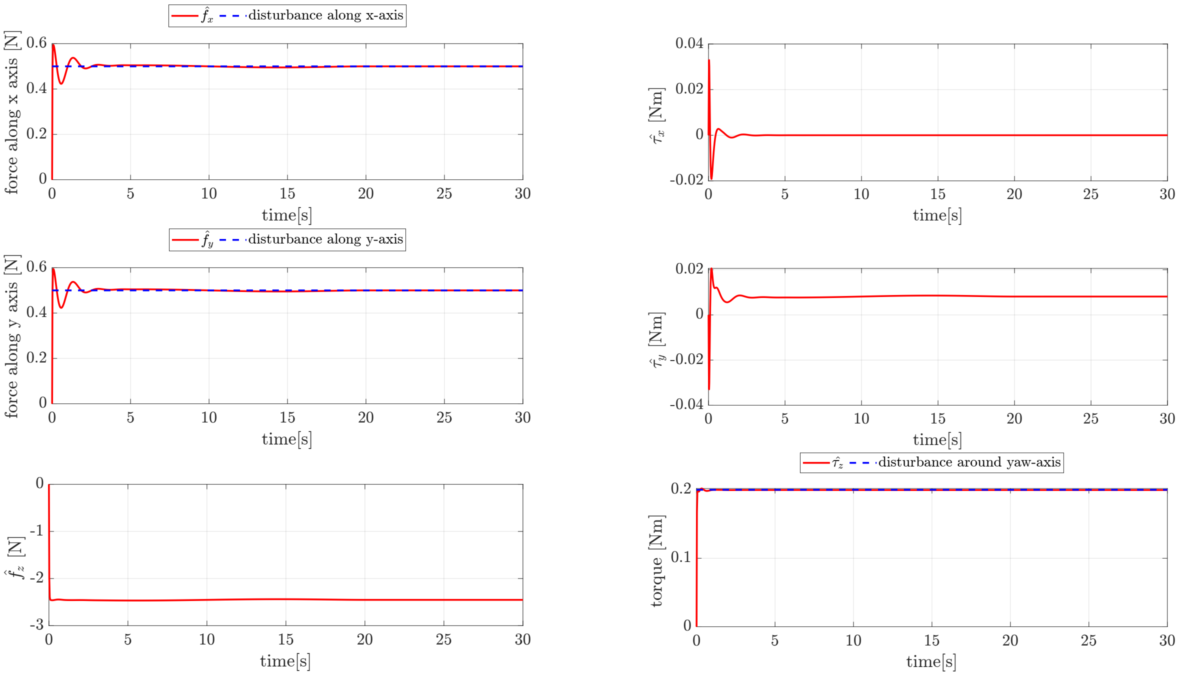


Figure 9: $r = 1$ and $\omega_n = 70$

At the end, the real mass of the UAV is computed from the estimated disturbance along the z-axis. The following formula is implemented in the MATLAB code to achieve our goal, which is composed of the following terms:

- The first component takes into account the contribution of the disturbance along the z-axis;
- The second component represents the assumed mass of the UAV;

$$m_{\text{real}} = \frac{\hat{f}_{ez}}{g} + m_{\text{est}} = 1.25\,\text{Kg}$$

Answer 5:

In the Simulink file *geometric_ control_ template.slx* provided to us, the geometric controller has been implemented in which, however, the internal part of the outer and inner loop must be completed and then the parameters tuned. As for the two loops, the formulas presented in the course were implemented. Specifically, the inner loop controller in order to determine τ_x, τ_y and τ_z employes the following formulas considering that $e_p = p_b - p_{b,d}$ and $\dot{e}_p = \dot{p}_b - \dot{p}_{b,d}$:

$$y_{b,d} = \frac{z_{b,d} \times x_{b,d}}{\|z_{b,d} \times x_{b,d}\|}$$

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

$$\omega_{b,d}^{b,d} = \left(R_{b,d}^T \dot{R}_{b,d}\right)^V$$

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

With

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

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

in order to avoid any representation singularity by defining angular errors in SO(3). But $z_{b,d}$ is obtained from the outer loop whose aim is also to calculate the total thrust u_T. Consequently the formulas implemented in the outer loop are the following:

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

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

Instead, as for the gains, several tests were carried out, so the trial and error procedure was used in order to calibrate them. Obviously, the goal that you want to achieve is to choose certain gains so that the inner loop is faster than the outer loop, that is, that the angular part arrives at convergence to 0 before the outer loop. As a result, the following gains were chosen:

- As regards the outer loop:
  - $K_p = diag([20, 20, 300])$;
  - $K_v = diag([15, 15, 45])$;
- As regards the inner loop:
  - $K_r = diag([30, 30, 400])$;
  - $K_w = diag([25, 25, 250])$;

Then the results obtained are reported.

The following graphs represent the position error on the left and the linear velocity error on the right:

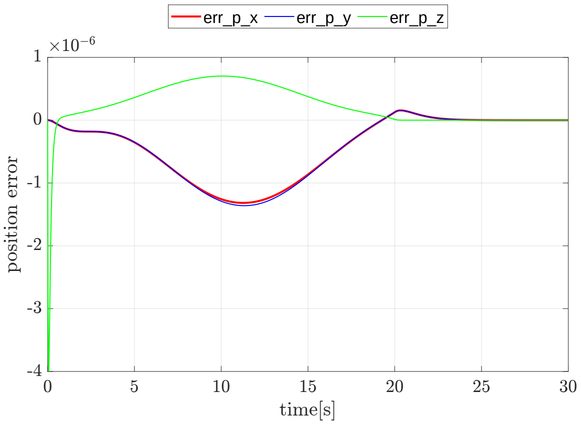


Figure 10: Position error

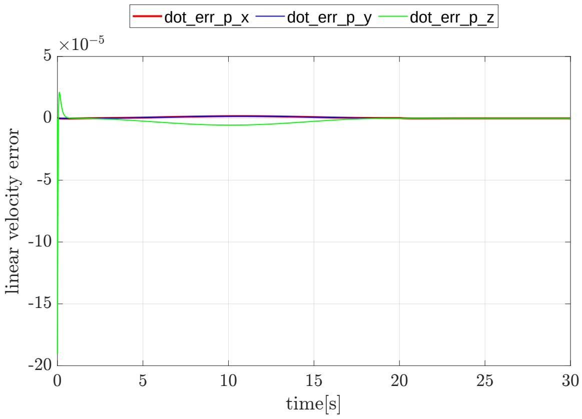


Figure 11: Linear velocity error

The figures above show that with the control gains we have chosen we are able to keep linear errors in the order of 10^{-6} and 10^{-5} respectively which makes them negligible for a UAV.

In this case, it is essential to report the attitude error, since as already mentioned in question 2, if this error is less than 90 degrees asymptotic stability can be demonstrated. So from the graphs shown, you can see how our goal is achieved because the inner loop (angular part given by the next 2 graphs) even if a little converges to 0 before the outer loop (the linear part, the first two graphs) around 22 seconds.

While the following graphs represent the attitude error on the left and the angular velocity error on the right:

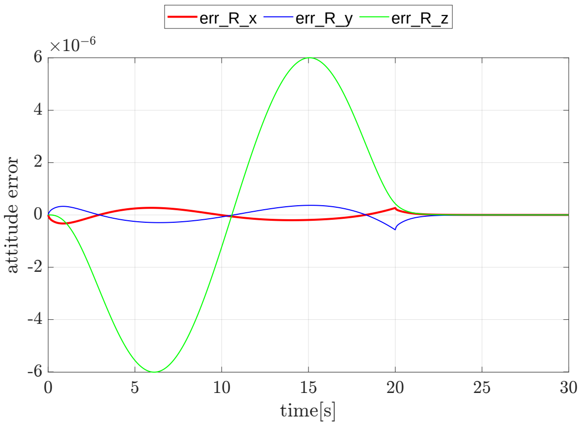


Figure 12: Attitude error

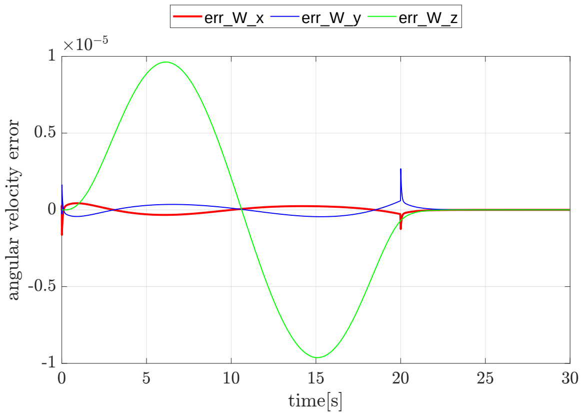


Figure 13: Angular velocity error

The figures above illustrate that the angular errors e_R and e_ω goes to 0 once the steady state has been achieved.

At the end, the following last two figures show the control values generated by the control system: the total thrust u_T exhibits a spike at the initial instance, followed by a gradual evolution and ultimately stabilizes at a value of $11.772N$, precisely the force required to sustain our UAV in flight; meanwhile the torques in τ_b have a regular evolution, having always very little values and converging to zero at steady state.

On the left the commanded total thrust is represented and on the right the commanded control torque:

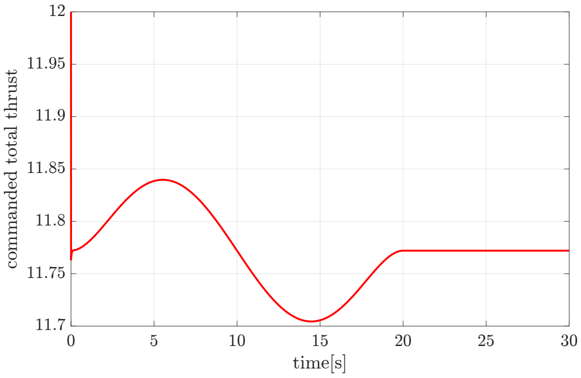


Figure 14: Total thrust

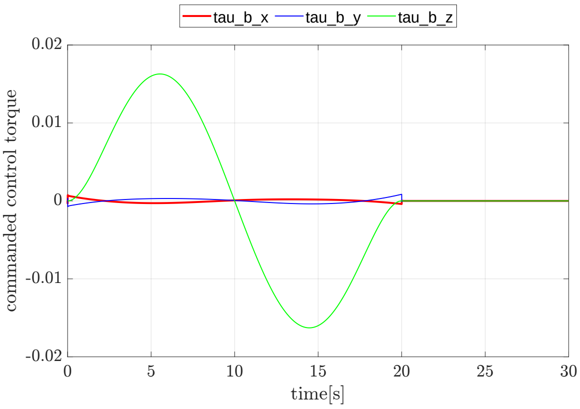
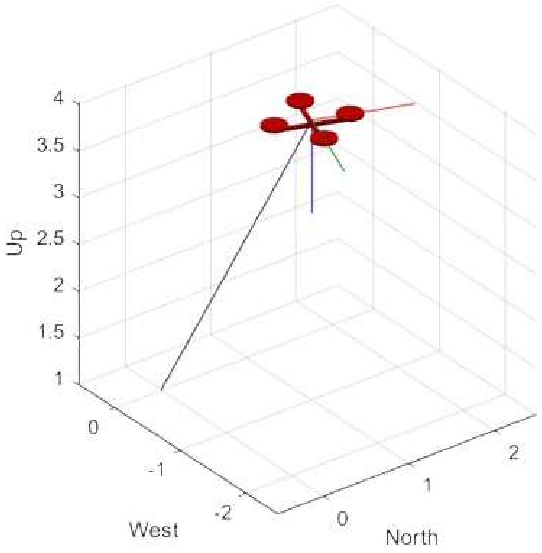


Figure 15: Control torque

Finally the trajectory of the UAV in 3D space is shown below:


Up
West
North