Chapter 1

Modelling, Planning and Control of an Aerial Manipulator

Corrado Verde **Buiano Giovanni**
matr. P38000139 **matr. P38000152**

During the last decade, the impact of aerial robotics in our daily lives has become more and more relevant. In fact, we can estimate that only in Europe the number of drones goes around 400.000 units, and this number is expected to increase in the upcoming years. One question arises: How did they became so popular? There could be plenty of answers for this question, but some these can easily explain this type of trend. Men have always been fascinated by the possibility of fly, not only for the possibility to speed up the journeys, but also to improve other human activities. Therefore, thanks to their versatility drones have found space in many types of applications which goes from entertainment to the I&M allowing humans to stay safe on the ground monitoring the drone which can perform risky tasks in the air.

1.1 Model and task specification

We will focus on a drone which has to deliver a package from an initial position to a final one, while recording the trajectory during the flight using a fixed camera. The drone use a 2 dof manipulator to perform an in hand delivery. The mechanical specification are the following: $m_b = 2kg$, $I_b = diag([1.24,\ 1.24,\ 2 * 1.24])$, $l = 0.30\ m$. Regarding the arm manipulator we have: $l_1 = 15\ cm$, $l_2 = 5\ cm$, $m_1 = m_2 = 0.5\ kg$, $I_{l_1} = 0.0011\ kg * m^2$, $I_{l_2} = 1.25 \times 10^{-4}\ kg * m^2$. We assume to neglect the mass and inertia of the arm motors, due to the low velocities of the manipulator. During the initial test for the controllers, the arm will not be considered, but it will be introduced once the estimator is presented.

1.2 Contributions from the authors

Regarding the planning, Corrado focused on the extension of BRRT to the 3D space, while Giovanni focused in application of DFS and the creation of the 3D map. Regarding the time law, both authors have collaborated to create functions able to create the 7-order polynomial and Via points in the trajectory. Regarding the controllers, both authors collaborated on the reimplementation of the geometrical controller, while Corrado focused on the Hierarchical and Giovanni on the passivity based. For the external disturbances simulation Corrado focused on the constant disturbances and Giovanni on the trapezoidal shape. The implementation of the online momentum based estimator was carried out by Giovanni for the Geometrical Controller while Corrado on the Hierarchical one. Regarding the implementation of the Arm manipulator, both authors gave their contribution in order to understand the theoretical aspects and possible implementation in the Simulink environment, in fact Corrado focused in the use of Newton-Euler formulation for the manipulator dynamics, while Giovanni focused on the sequence of transformation needed to project the force and momentum expressed in link frame into the body frame of the drone.

Chapter 2

Planning

2.1 3D scanning of the environment

To perform its task, the drone needs a path which connects the starting point to the goal point combined with an appropriate time law which guarantees bounded control input. Supposing that the drone moves in a neighbourhood, using the information given by satellite images, we can obtain a 3D image of the environment(Fig 2.1).

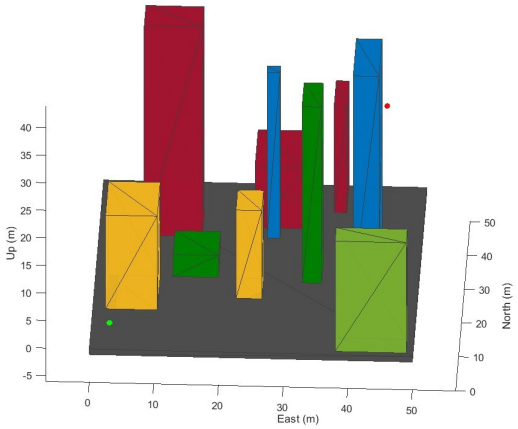


Figure 2.1: Map 3D of the environment

So we are considering an environment of dimension 50x50 m with variable size buildings with a maximum height of 39 m. If we suppose to use the NED configuration for both earth-fixed and body-frame, the initial configuration is $q_i = [3,\ 3,\ -3]$ and the final one is $q_f = [25,\ 45,\ -30]$. The problem is to find a geometric path connecting these two pint without hitting the obstacles. Using only the 3d map is not enough to evaluate e collision check algorithm, so in order to do that it is necessary to construct an occupancy 3D map. Using the 3d map, it is possible to simulate a 3D Lidar scanner whose output is the 3D occupancy grid map(Fig 2.2).

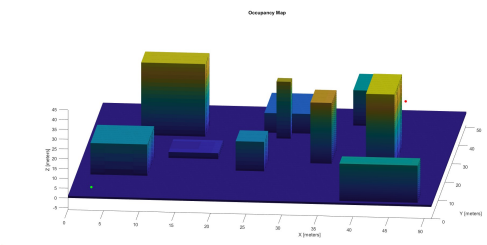


Figure 2.2: 3D occupancy grid map

2.2 BRRT

With the 3D occupancy map, it is possible to implement the BRRT algorithm which allows to find a path. the algorithm is the same seen in 2D but suitable changes have been made. In particular we have to cope for the additional dimension and also the robot is not considered anymore as a point but in a more appropriate way as a sphere of radius 0.5m . In this way, each samples of the rect individuate the position of the centre of the sphere which coincide with the centre of mass of the drone, and then the collision are checked between the 3D sphere object and the buildings. The algorithm is able to find a path with approximately 30 - 40 iterations and the result can be seen in figure 2.3.

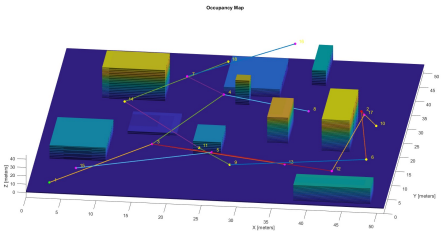
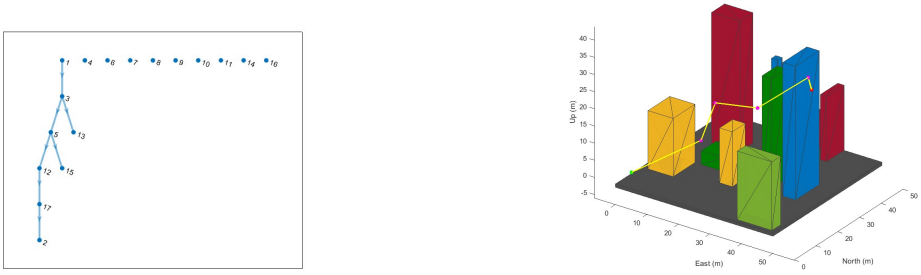


Figure 2.3: Full path

2.3 DFS

From figure 2.3, we can notice that starting from the initial position there can be multiple path leading to different configuration, therefore in order to highlight the single path connecting the start to the goal we can use a *Graph Search Algorithm* for example the DFS. Knowing the connection between the nodes it is possible to retrive the adjacency matrix which will be given as input to the function together with the starting goal and the complete list of the nodes. As output, we obtain the DFS tree with the sequence of explored nodes and the ordered sequence of nodes from the starting one to the final one:



(a) DFS tree

(b) Path inside the 3D environment

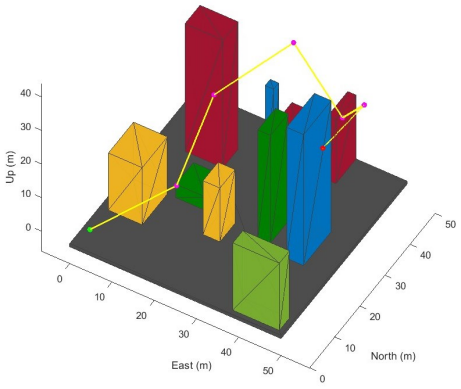
Figure 2.4

2.4 Defining the time law

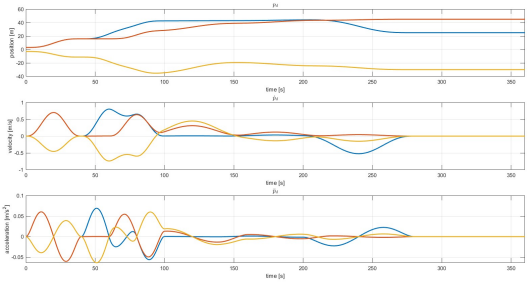
Among the different path which can be generated trough this procedure, we have chosen the one in figure 2.5a which allows the drone to perform the most panoramic flight during the task. Given the sequence of points of the path, we have to define a time law upon the geometric path, in order to define the references for our feedback controllers. It is known that the quadrotor is differentially flat system, and the flat outputs are x, y, z, ψ. In order to get bounded control input, it is preferable to choose intervals of time for each segment not too short, therefore the total time of simulation is 360 s where the first three segment have a duration of 40s, while the other three a duration of 80s. for what regards the heading ψ, the following values has been chosen:

$$\psi = \begin{bmatrix} 0 & -\frac{\pi}{6} & -\frac{\pi}{2} & 0 & \frac{\pi}{6} & \frac{\pi}{2} & \pi \end{bmatrix}$$

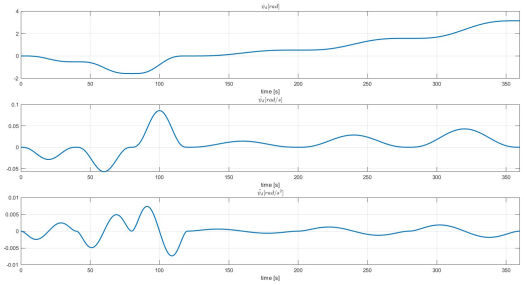
Then at the end of each segment, the drone will reach a particular heading, in order to capture with the fixed camera the trajectory. For the time law of each segment a seventh order polynomial has been choosen, which allow to impose not only the initial and final position and velocity, but also the acceleration and the jerk, which should be equal to zero to guarantee a trajectory as smooth as possible. For the parametric representation the arclength $s(t)$ as been choosen, varying from the value $s(t_i) = 0$ to $s(t_f) = \|p_f - p_i\|$. By appropriately connecting each segment of trajectory, we finally retrieve the references for the quadcoper. In addition, to guarantee that the drone does not stop in the middle point, it is possible to anticipate the generation of the following segment of trajectory, then the robot will not reach these points but will be very close to them becoming then Via points. The plots of the references are shown.



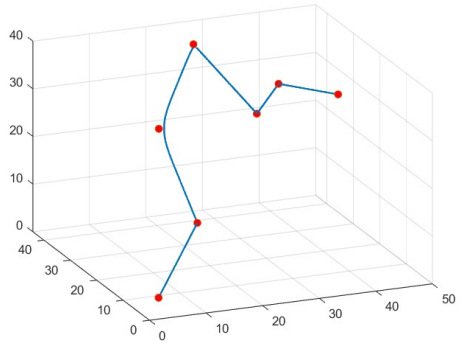
(a) Actual path used for the task



(b) p_d , $\dot{p_d}$ and $\ddot{p_d}$



(c) ψ_d , $\dot{\psi_d}$ and $\ddot{\psi_d}$



(d) Geometrical path

Figure 2.5

From figure 2.5d the via points are clearly visible in the trajectory, and also from the first plot in figure 2.5b, it is possible to notice that the drone reach the final position before the ending time as expected. From the velocity plot in 2.5b we can see that the maximum velocity reached goes around 0.7 $\frac{m}{s}$ and the the maximum acceleration is around 0.064 $\frac{m}{s^2}$ then the choice for the time intervals satisfy the constraint of bounded references. For what concerns the ψ_d we have that the velocities and the acceleration are also quite small, with maximum values of respectively 0.085 $\frac{rad}{s}$ and 0.0074 $\frac{rad}{s^2}$. These last considerations concludes the discussion regarding the planning which has involved many different aspects and techniques which are been carefully explored in order to get the better performances for the given task.

Chapter 3

Controller Development

In this chapter, once we have obtained the references defining the task, it is necessary to develop a feedback controller which ensures the perfect tracking of the references and asymptotic stability of the system. Three type of controllers will be examined highlighting pros and cons of each one.

3.1 Hierarchical controller

$$\mu_d = -K_p \begin{bmatrix} e_p \\ \dot{e}_p \end{bmatrix} + \ddot{p}_{b,d} \tag{3.1}$$

$$\tilde{\tau} = -K_e \begin{bmatrix} e_\eta \\ \dot{e}_\eta \end{bmatrix} + \ddot{\eta}_{b,d} \tag{3.2}$$

$$K_p = \begin{bmatrix} 100 & 0 & 0 & 100 & 0 & 0 \\ 0 & 100 & 0 & 0 & 100 & 0 \\ 0 & 0 & 100 & 0 & 0 & 100 \end{bmatrix} \qquad K_e = \begin{bmatrix} 50 & 0 & 0 & 50 & 0 & 0 \\ 0 & 50 & 0 & 0 & 50 & 0 \\ 0 & 0 & 50 & 0 & 0 & 50 \end{bmatrix}$$

From the results we can notice that the error in position is quite small with a maximum error around 1.75×10^{-3} (3.2a) and the drone perfectly follows the imposed trajectory (3.1). The linear velocity error has quite the same behaviour with an order of the error around 10^{-5} (3.2b). From the plot of e_{η_b} (3.3a) we can see that the error goes around 10^{-6} but some spike are present during this simulations. this is possible linked to the operation of derivative in order to obtain the references of Euler angles necessary to compute the control law, in particular the spike are present when there is a change of trajectory. Increasing the time interval of each segment bring to a smaller amplitude for the spikes. Regarding the control action, the total thrust (3.4a) has a quite smooth behaviour with a maximum value of 19.75 N which is necessary to lift up the drone in some phases of the trajectory, but as expected in the final instants of the simulation, at the equilibrium, the total thrust reach the initial value. For what concerns τ_b (3.4b), it is clearly visible that the values are quite small with a maximum value for the third component which goes around 0.018 Nm while the other component goes around the order 10^{-4} which are necessary to change the orientation of thrust vector.

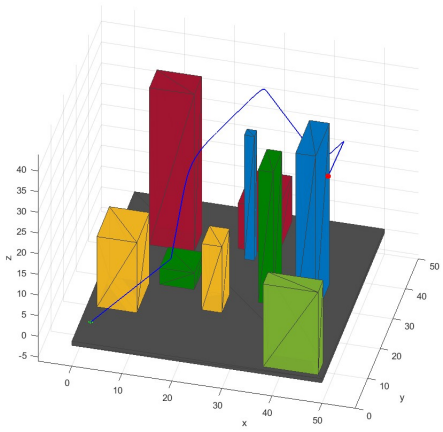
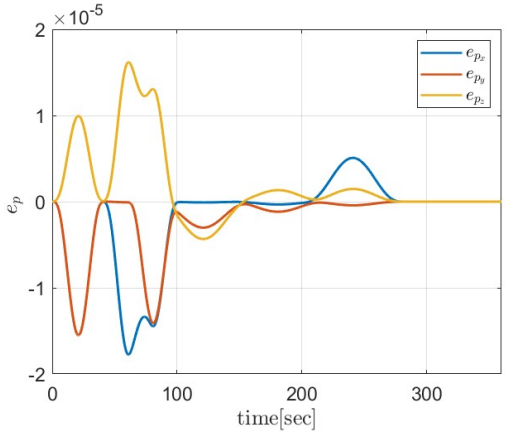
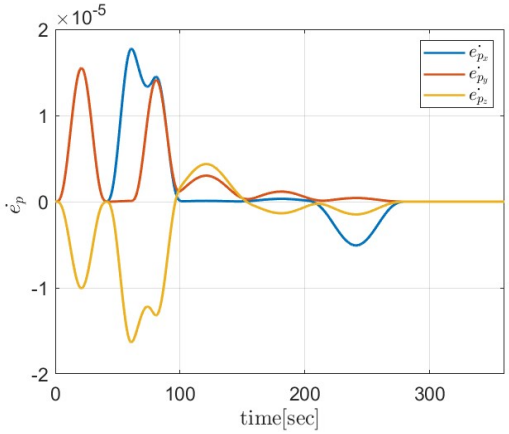


Figure 3.1: Trajectory

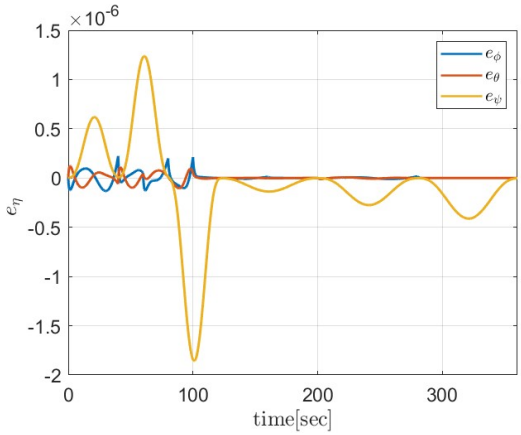


(a) Error in position

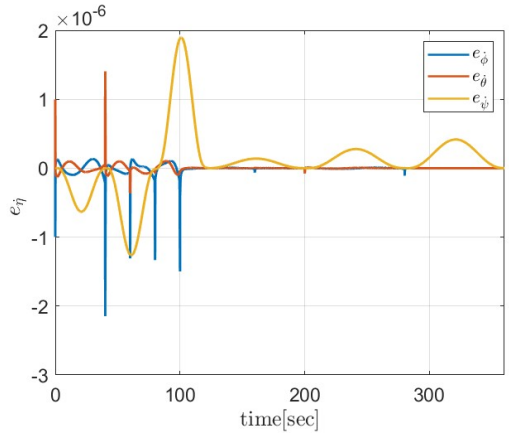


(b) Error in linear velocity

Figure 3.2

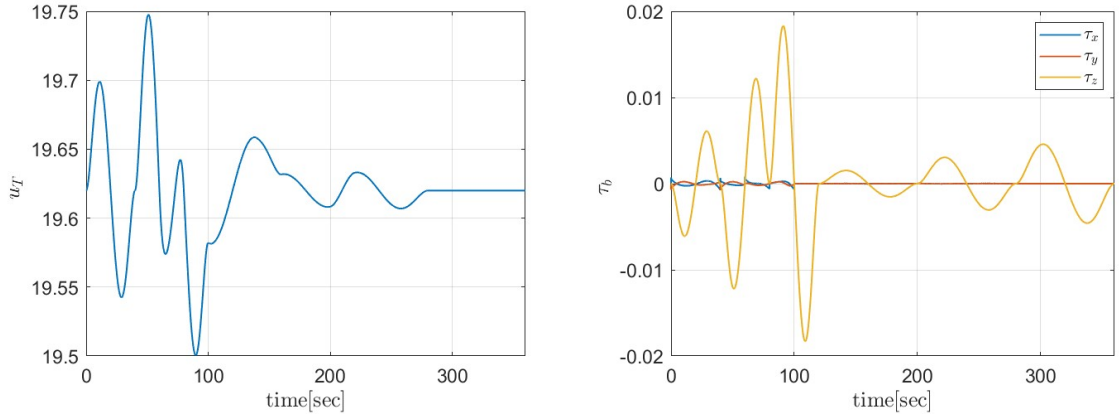


(a) Error in η



(b) Error in $\dot{\eta}$

Figure 3.3



(a) Total thrust

(b) τ_b

Figure 3.4

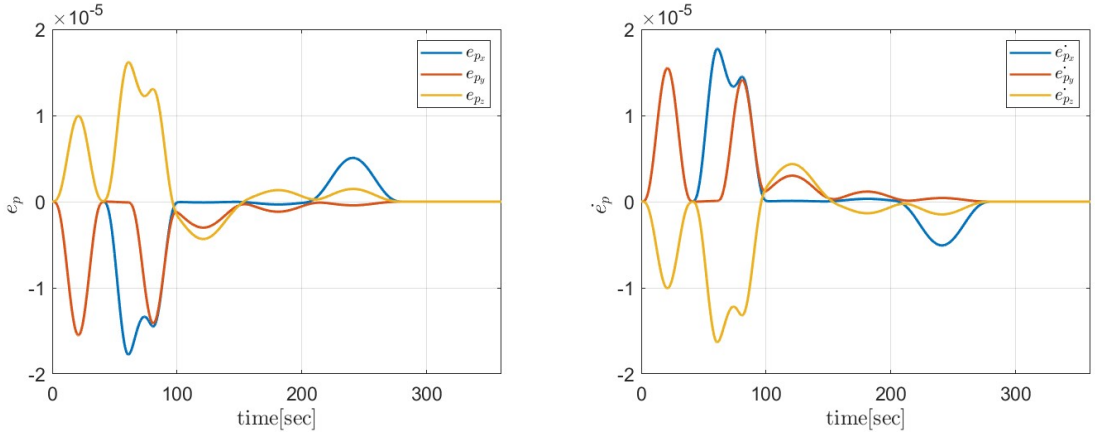
3.2 Passivity Based Controller

$$\tau^b = Q^{-T}\left(M\left(\eta_b\right)\ddot{\eta}_r + C\left(\eta_b, \dot{\eta}_b\right)\dot{\eta}_r - D_o v_\eta - K_o e_\eta\right) \tag{3.3}$$

For the simulations, the gains has been chosen as:

$$\sigma = 30 \quad \nu = 10 \quad D_o = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix} \quad K_o = \nu D_o$$

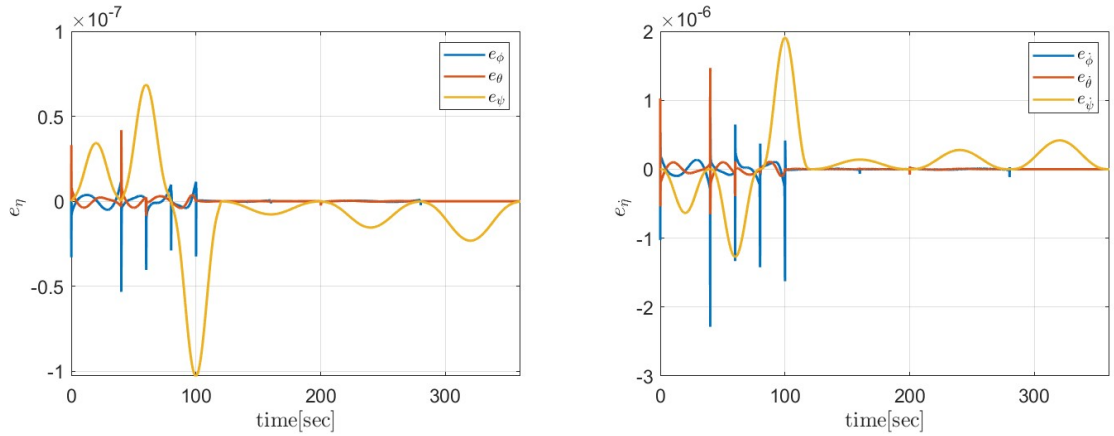
Regarding the results of the simulations, there are no great differences between the two controllers, only for the angular part (fig 3.6a) it is possible to notice a lower order of error, which goes around 10^{-7}.



(a) Error in position

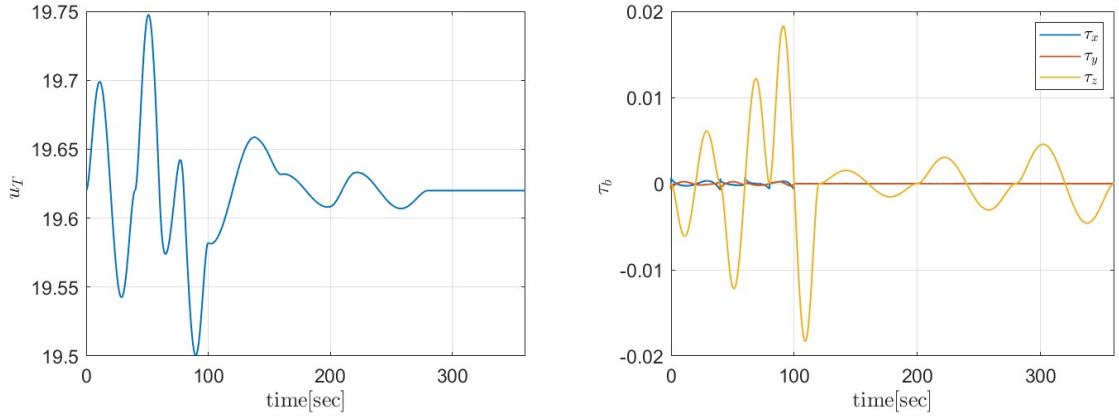
(b) Error in linear velocity

Figure 3.5



(a) error in η (b) error in $\dot{\eta}$

Figure 3.6



(a) Total thrust (b) τ_b

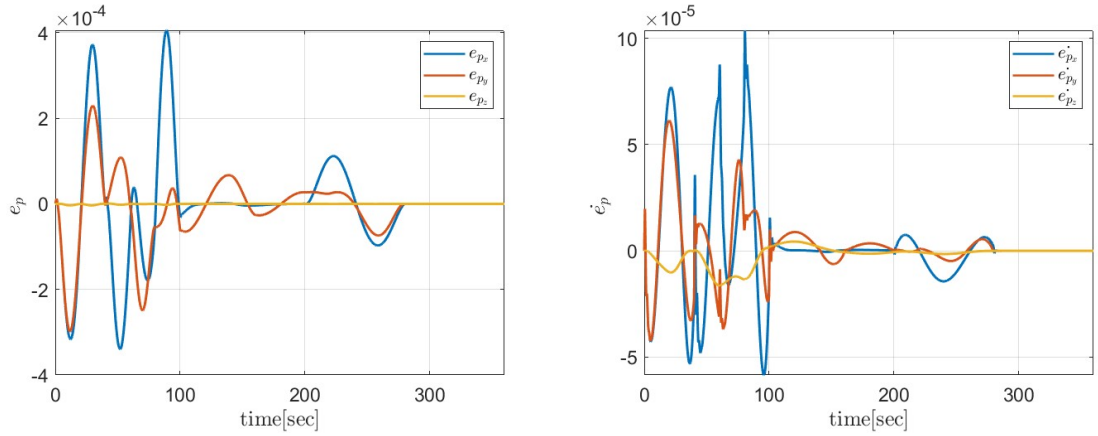
Figure 3.7

3.3 Geometrical Controller

$$\tau^b = -K_R e_R - K_\omega e_\omega + S\left(\omega_b^b\right) I_b \omega_b^b - I_b\left(S\left(\omega_b^b\right) R_b^T R_{b,d} \omega_{b,d}^{b,d} - R_b^T R_{b,d} \dot{\omega}_{b,d}^{b,d}\right) \tag{3.4}$$

The gains of the controller has been chosen as:

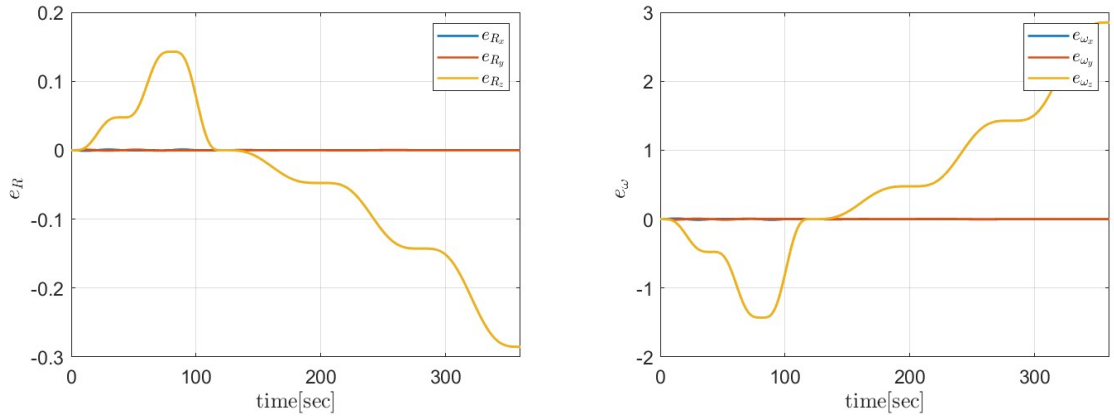
$$K_p = \begin{bmatrix} 40 & 0 & 0 \\ 0 & 40 & 0 \\ 0 & 0 & 40 \end{bmatrix} \quad K_v = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad K_R = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{bmatrix} \quad K_\omega = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix}$$



(a) Error in position

(b) Error in linear velocity

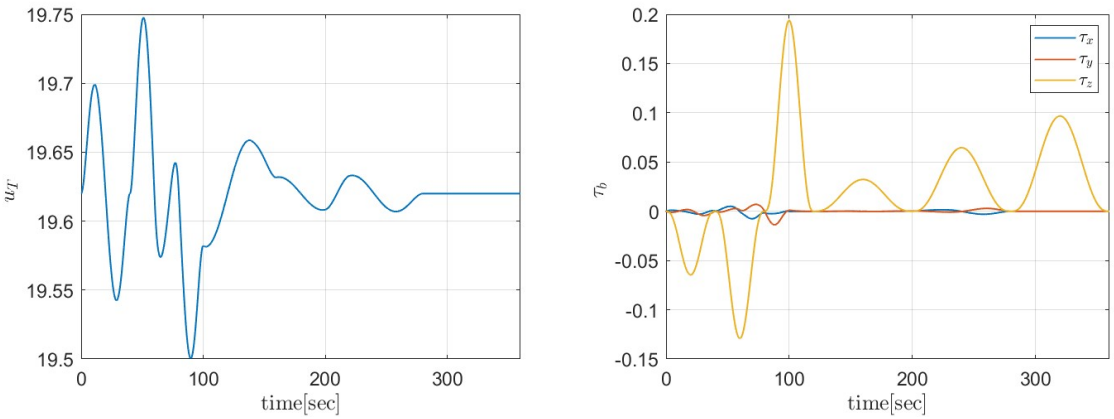
Figure 3.8



(a) Error in R

(b) Error in ω

Figure 3.9



(a) Total thrust

(b) τ_b

Figure 3.10

The controller is able to successfully track the references, with a behaviour of the total thrust similar to the one seen in the previous, while for the τ_b (fig 3.10b) the maximum peak is still 0.018 Nm but the shape for the third component is quite different from the one seen before, due to the different derivation of the control law. For the orientation (fig 3.9a), we can see that the error does not go asymptotically to zero, but remains bounded along the trajectory. The position error (fig 3.8a) has a maximum peak of 4×10^{-4}, but this could be linked to a different choice of the gains.

3.4 Comparison among the controllers

The developed controllers, in case of absence of disturbances, are able to track the reference planned in the previous chapter, with some differences between them. The first thing that we can notice is that the Hierarchical and the passivity based controllers have quite the same behaviour, being the control schemes very similar, with the only difference present in the angular part of the equation. In case of perfect known model, with an appropriate tuning of the parameters of the passivity based controller, which could be difficult due the great number of tunable parameters we are able to obtain great results, in some cases better then the hierarchical one. Another aspect which can be notice is the presence in both strategies of peaks in the errors for the angular quantities (fig 3.3b and 3.6b) , which have a really small amplitude and so they do not affect much the behaviour of τ_b. This may be due the presence of two numeric derivation in order to get the references for the Euler angles in particular ϕ and θ, which create this type of behaviour in the connection points of the trajectories. Instead, in the geometric controller, we can see that the behaviour is much smoother, being the overall scheme different from the one seen for the other two controllers. What we can see is that the error for the angular quantities does not go asymptotically to zero, but the error is still bounded and acceptable for our task. Regarding the behaviour for the total thrust, we can observe that it is quite identical for all the three schemes, being the derivation for the total thrust quite similar for all the schemes.

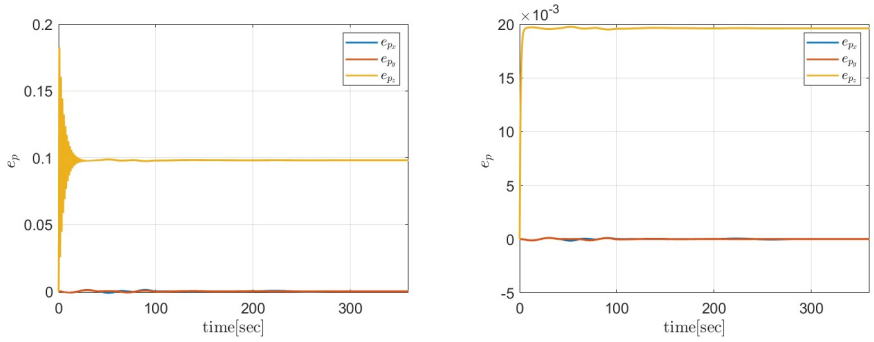
Chapter 4

Model Perturbation

In this chapter, we will discuss about the performances of the controllers in case of presence of unmodelled effect, such as model uncertainties and presence of external disturbances. Then we will introduce the momentum based estimator which will be tested on two type of controllers, the hierarchical and the geometrical controller, highlighting the different implementations of the estimator for both the schemes. For the hierarchical the gains have been lowered in order to not encounter problem in the computation of the numeric derivative.

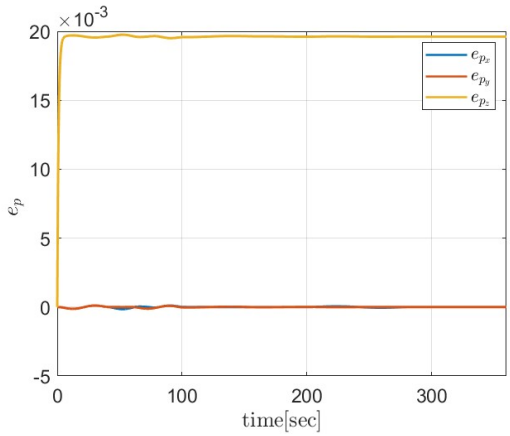
4.1 Parametric uncertainties

Assuming an uncertainty on the value of the mass and inertia matrix of about 20%, we want to test the robustness of the previously designed controllers. From the simulation, we can see that the main effect can bee seen on the plot regarding error in position (4.1) and (4.2a). In fact all three controllers present a constant error in position on the third component which is around 0.02 m for the hierarchical and the passivity based, while for the geometric controller is around 0.1 m. This is expected because a variation of the mass mainly effect the linear part of the model equation, as can be seen on the u_T, which should increase in order to compensate for the extra mass(figure (4.3b)). The effect on the variation on the inertia matrix can be seen in the plot of the angular quantities (4.2b) and (4.3a), in particular we can observe that the orientation error in the angular part is smaller than the one present in the hierarchical, because as expected the passivity based controller does not use FBL then it is more robust against model uncertainties.

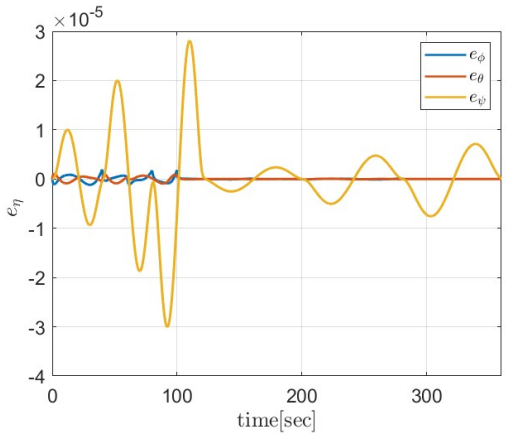


(a) Geometrical controller e_p (b) Hierarchical controller e_p

Figure 4.1: 20% of uncertainty on the value of the mass, control without estimator

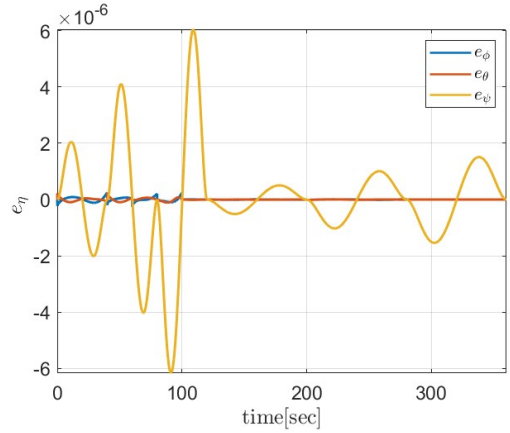


(a) Passivity based controller e_p

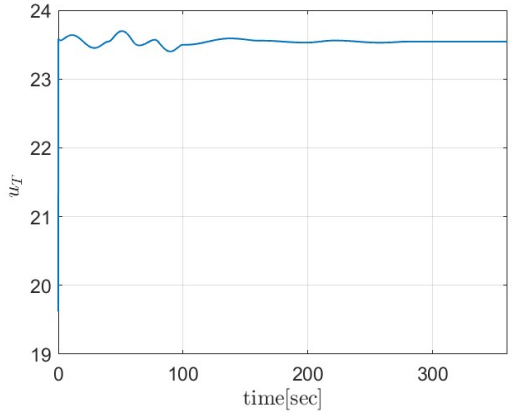


(b) Hierarchical controller e_η

Figure 4.2: 20% of uncertainty on the value of the mass, control without estimator



(a) Passivity based controller e_η



(b) Hierarchical controller u_T

Figure 4.3: 20% of uncertainty on the value of the mass, control without estimator

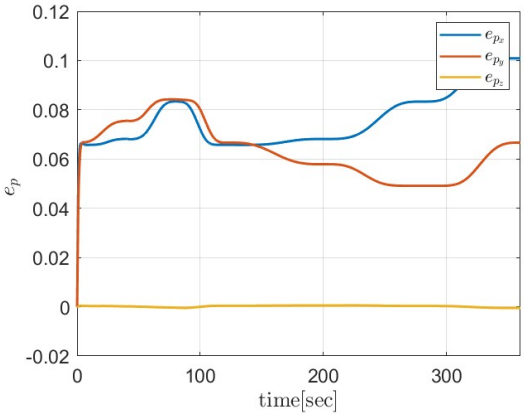
4.2 External Disturbances

In this section two type of disturbances will be examined, discussing the effect on the hierarchical controller and the geometrical controller. These type of disturbances can be seen as wind gust acting on the body, whose values are expressed in earth fixed frame. The disturbances are:

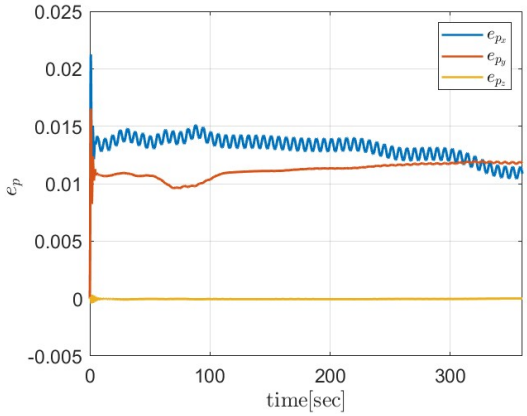
1. $f_e = \begin{bmatrix} 0.5N \\ 0.4N \\ 0 \end{bmatrix} \qquad \tau_e = \begin{bmatrix} 0 \\ 0.02Nm \\ 0 \end{bmatrix}$
2. trapezoidal force acting for 50 s on x axes and around the x axes

When we apply the constant disturbance in the hierarchical controller, we can notice that it will eventually lead to a constant error in position and orientation (look figure (4.4)), where it reach in the first one a maximum value of 0.1 which can lead to accidentally hit of obstacles. Also we can notice from the figure (4.5) that the initial value of τ_b reach some higher values, in order to recover the error. For the geometric controller, we can observe that a constant error is

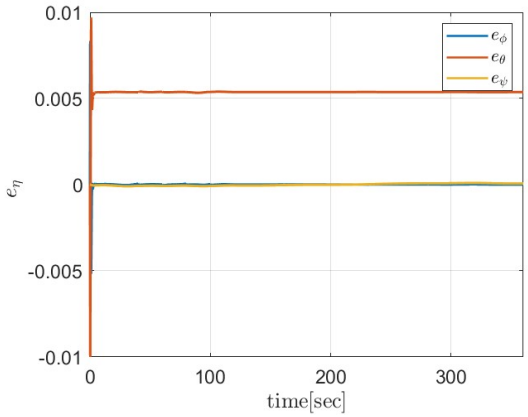
also present, but with lower value for the position part than the hierarchical, with a maximum value of 0.02 m, but we can notice an oscillatory behaviour in the response. The initial values of τ_b however are high and could not be physically feasible.



(a) Hierarchical controller e_p

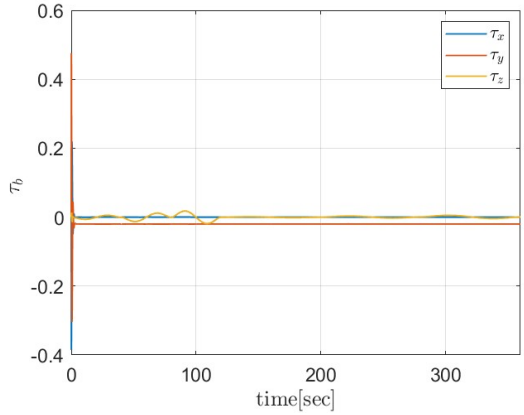


(b) Geometrical controller e_p

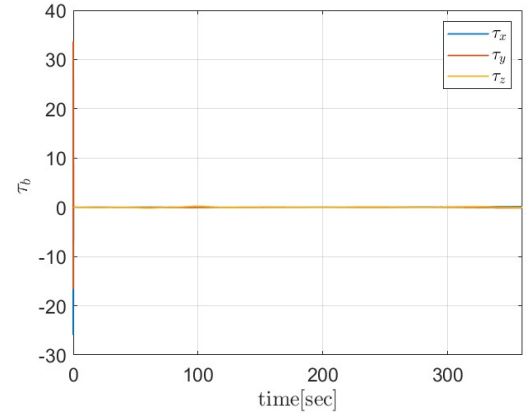


(c) Hierarchical controller e_η

Figure 4.4: External disturbances constant, control without estimator



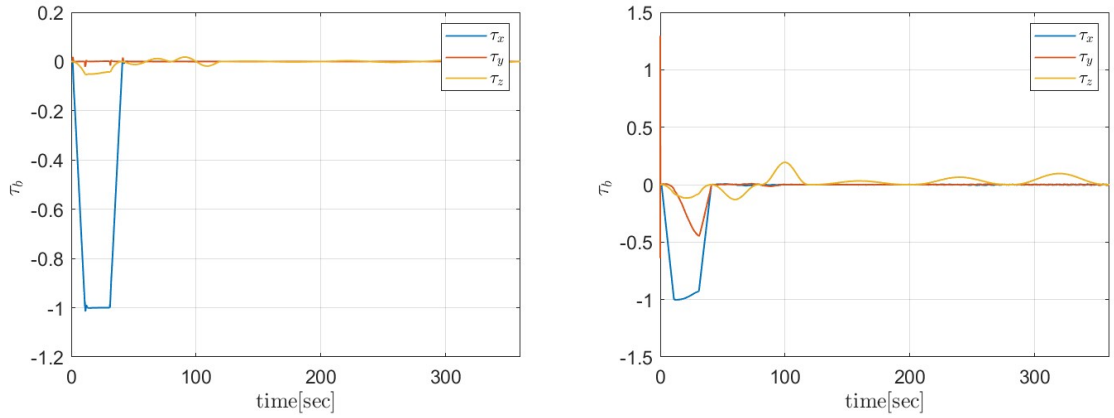
(a) Hierarchical controller τ_b



(b) Geometrical controller τ_b

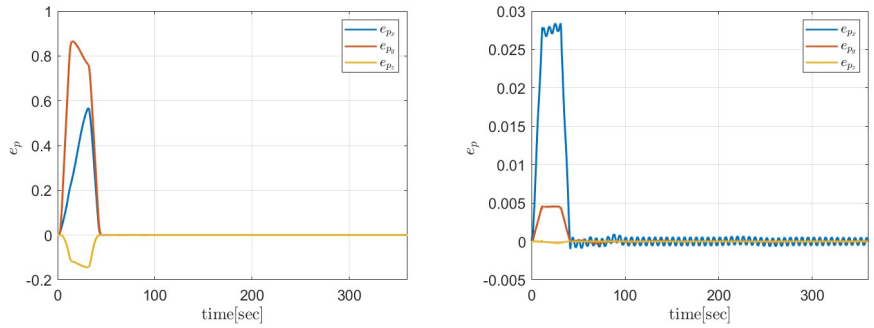
Figure 4.5: External disturbances constant, control without estimator

Applying the trapezoidal disturbance, we can see that in the hierarchical controller the error in position is quite big, being the amplitude of the disturbance greater, and it reach a maximum value of 0.8 m (looking the figure (4.7a) and (4.7b)). This could bring to a collision of the drone towards an object if it moves in narrow spaces, but in these case in the initial phase the drone is far from any building. We can see that the control action τ_x as the same shape of the disturbance looking the figure (4.6), trying to compensate for it, and the same shape can be seen also in the geometrical controller. For the geometrical controller the maximum amplitude for the position error is lower, reaching a value of 0.028 m, but with the same oscillatory behaviour seen with the constant disturbance(figure (4.7b)).

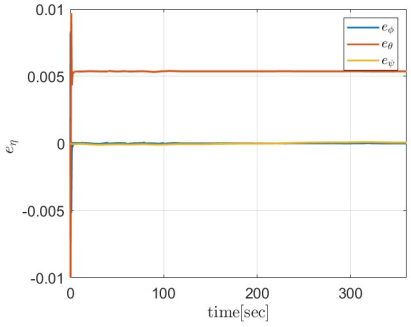


(a) Hierarchical controller τ_b (b) Geometrical controller τ_b

Figure 4.6: External disturbances trapezoidal, control without estimator



(a) Hierarchical controller e_p (b) Geometrical controller e_p



(c) Hierarchical controller e_η

Figure 4.7: External disturbances trapezoidal, control without estimator

4.3 Momentum Based Estimator

$G(s) \in \mathbb{C}^{6\times 6}$ is a diagonal matrix of transfer functions defined in equation (4.1).

$$G_i(s) = \frac{k_o^r}{(s+b)^r}, i = 1, \cdots, 6 \qquad b = 50 \geq 0 \qquad K_o = b \tag{4.1}$$

We can implement the filter using the following recursive formula that is constructed for the RPY dynamic model and in this case is implemented online but off course making sure that the estimator is the fastest system in the control loops.

$$\begin{bmatrix} \hat{f_e}(0) & \hat{\tau_e}(0) \end{bmatrix}^T = 0 \text{ and } q(0) = 0 \tag{4.2}$$

$$q(k+1) = \begin{bmatrix} mI_3 & O_3 \\ O_3 & M\left(\eta_b(k+1)\right) \end{bmatrix} \begin{bmatrix} \dot{p}_b(k+1) \\ \dot{\eta}_b(k+1) \end{bmatrix} \tag{4.3}$$

$$\gamma_1(k+1) = \gamma_1(k) + K_1 \left(q(k+1) - q(k) - T_s \begin{bmatrix} \hat{f}_e(k) \\ \hat{\tau}_e(k) \end{bmatrix} - T_s \begin{bmatrix} mge_3 - u_T(k) R_b(k) e_3 \\ C^T\left(\eta_b(k), \dot{\eta}_b(k)\right)\dot{\eta}_b(k) + Q^T\left(\eta_b(k)\right)\tau^b(k) \end{bmatrix} \right) \tag{4.4}$$

$$\gamma_i(k+1) = \gamma_i(k) + K_i T_s \left[-\begin{bmatrix} \hat{f}_e(k) \\ \hat{\tau}_e(k) \end{bmatrix} + \gamma_{i-1}(k) \right], \quad i = 2, \cdots, r \tag{4.5}$$

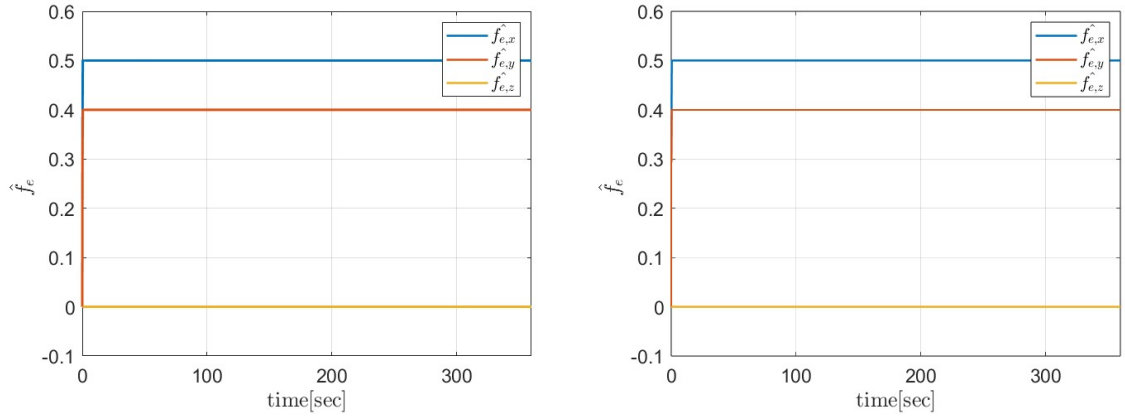
To have the recursive formula for the free coordinate model we have to change the equation (4.4) and (4.3) with (4.6) and (4.7).

$$q(k+1) = \begin{bmatrix} mI_3 & O_3 \\ O_3 & I_b \end{bmatrix} \begin{bmatrix} \dot{p}_b(k+1) \\ w_b^b(k+1) \end{bmatrix} \tag{4.6}$$

$$\gamma_1(k+1) = \gamma_1(k) + K_1 \left(q(k+1) - q(k) - T_s \begin{bmatrix} \hat{f}_e(k) \\ \hat{\tau}_e(k) \end{bmatrix} - T_s \begin{bmatrix} mge_3 - u_T(k) R_b(k) e_3 \\ -S(w_b^b(k)) I_b w_b^b(k) + \tau^b(k) \end{bmatrix} \right) \tag{4.7}$$

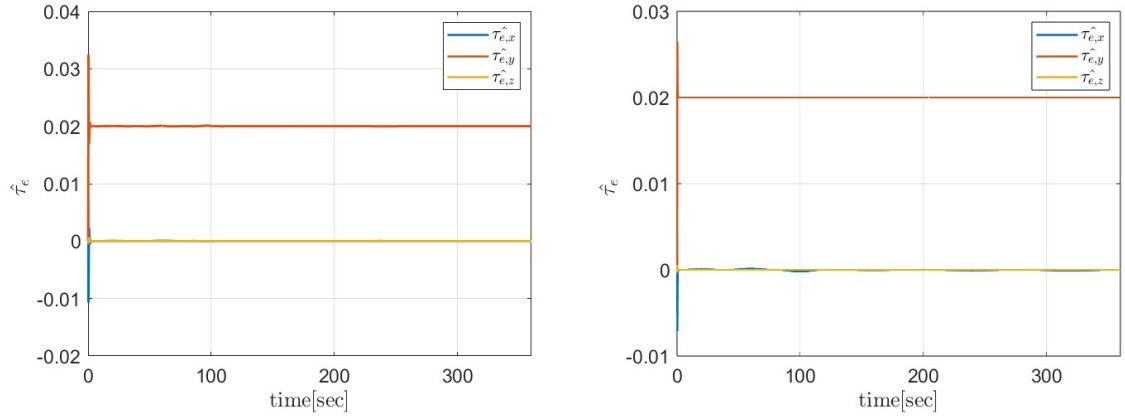
4.4 Comparison among the controllers using estimator

For the simulations, we have used an estimator of order 7 with $b = 50$. In the first case of constant disturbances we can see that the estimator is able to correctly estimate the external disturbances, with sufficiently fast dynamic(fig 4.8 and 4.9). When we try to apply the external estimated wrench in order to compensate the unmodelled disturbances, we can see that the error in position in the hierarchical controller (fig 4.10a), after an initial peak of 18×10^{-3}, goes asymptotically to zero, and the same happens for the error in orientation. Regarding the control action τ_b the maximum peak goes around 1.5 Nm (fig 4.11a). For the geometric controller being the τ_b too high (fig 4.5b), a saturation from -0.5 Nm to 0.5 Nm has been added (fig 4.11c), but we can see that the error in position does not converge to zero but the oscillatory behaviour is not present anymore and the overall amplitude is decreased, with a bounded value around 2×10^{-3} (fig 4.10b).



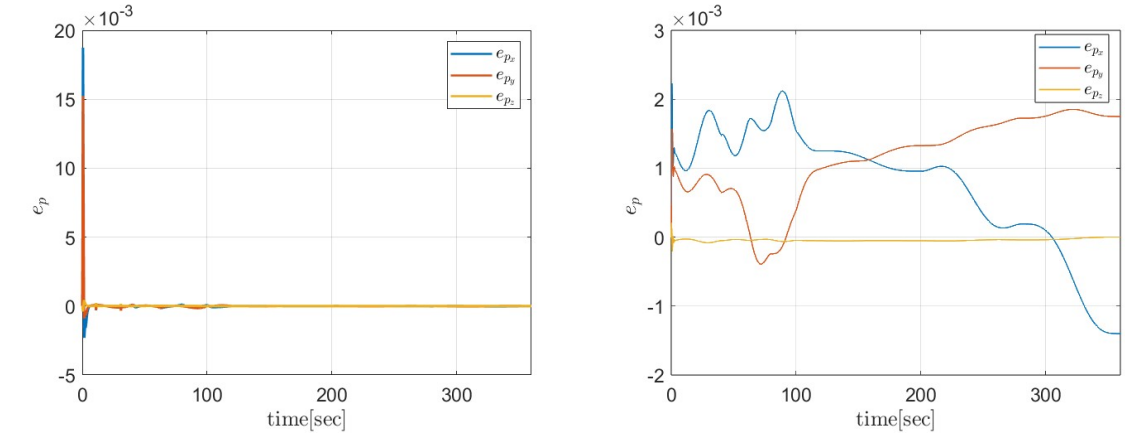
(a) Hierarchical controller $\hat{f}_e$ (b) Geometrical controller $\hat{f}_e$

Figure 4.8: External disturbances constant, controller with estimator



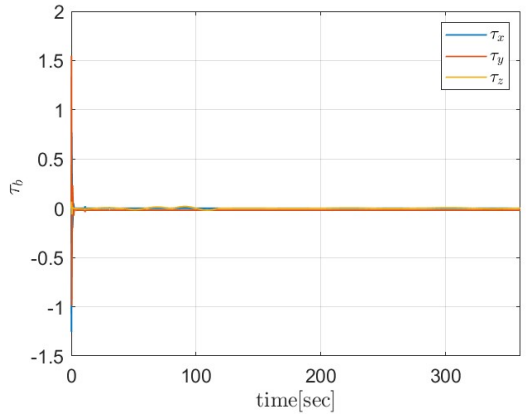
(a) Hierarchical controller $\hat{\tau}_e$ (b) Geometrical controller $\hat{\tau}_e$

Figure 4.9: External disturbances constant, controller with estimator

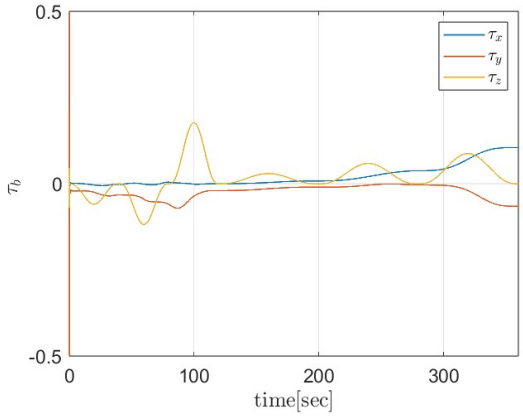


(a) Hierarchical controller e_p (b) Geometrical controller e_p

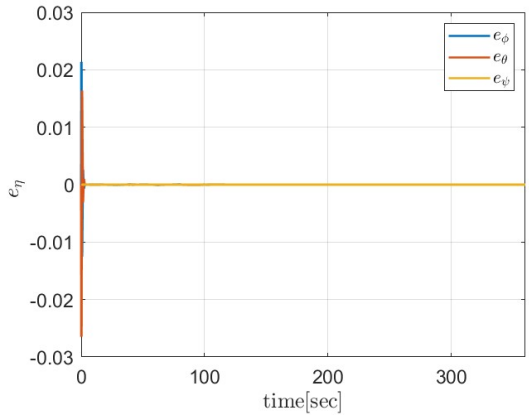
Figure 4.10: External disturbances constant, controller with estimator



(a) Hierarchical controller τ_b



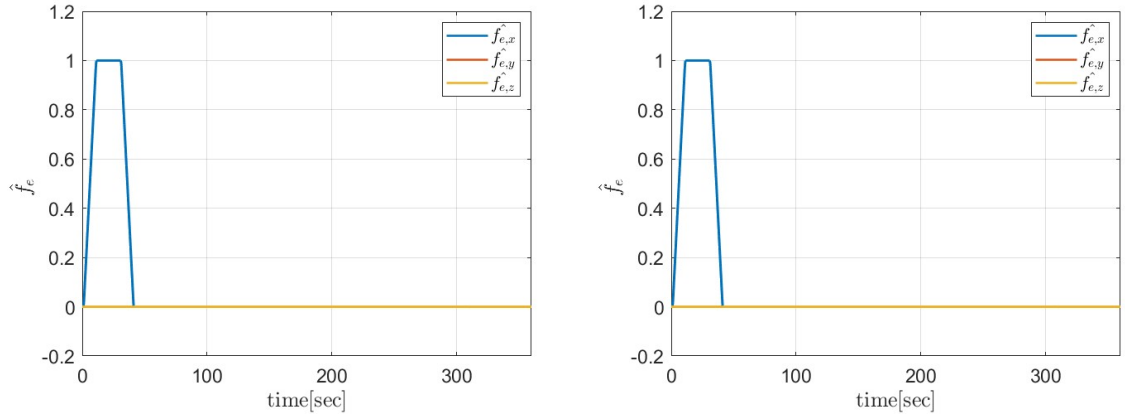
(b) Geometrical controller τ_b



(c) Hierarchical controller e_η

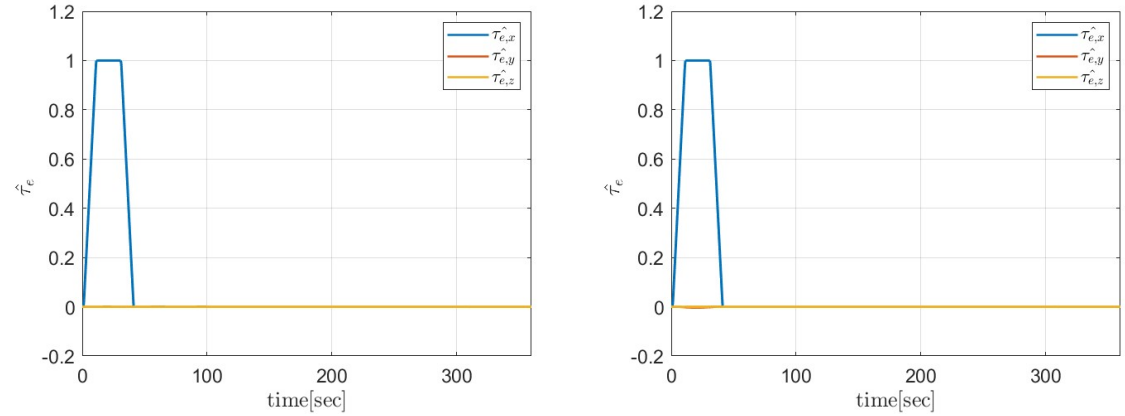
Figure 4.11: External disturbances constant, controller with estimator

For the trapezoidal shape, we can observe that the estimators in both controllers correctly estimate the external trapezoidal disturbances. In the hierarchical controller, we can see that the maximum peak reached in the error in position is now around 0.012 m, smaller then the one seen without compensation (looking figures (4.7a) and (4.14a)), while for the error in the angular part we only have a maximum peak around 4×10^{-4}(looking figure(4.15c)). The position error for the geometrical controller with the compensation does not present the oscillatory behaviour, with a maximum peak around 2.8×10^{-3}, which is lower than the one seen before(looking figures (4.14b) and (4.7b)). How we can see in the figures (4.15a) and (4.15b) the control action for both controller maintain the same shape for τ_x but we can see additional terms in the other component given by the compensation effect.



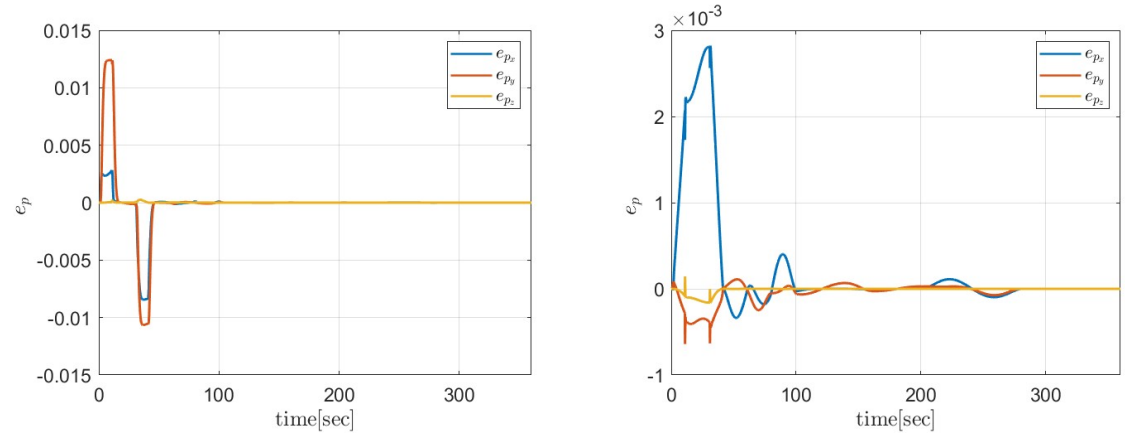
(a) Hierarchical controller $\hat{f}_e$ (b) Geometrical controller $\hat{f}_e$

Figure 4.12: External disturbances trapezoidal controller with estimator



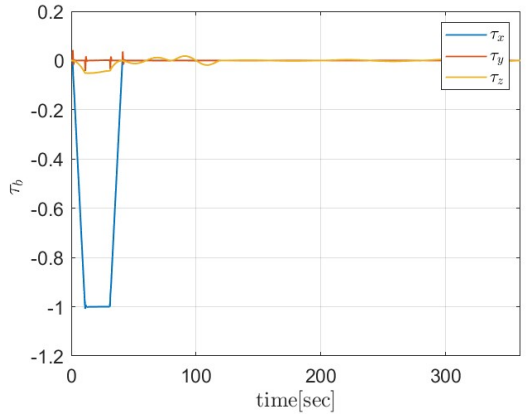
(a) Hierarchical controller $\hat{\tau}_e$ (b) Geometrical controller $\hat{\tau}_e$

Figure 4.13: External disturbances trapezoidal, controller with estimator

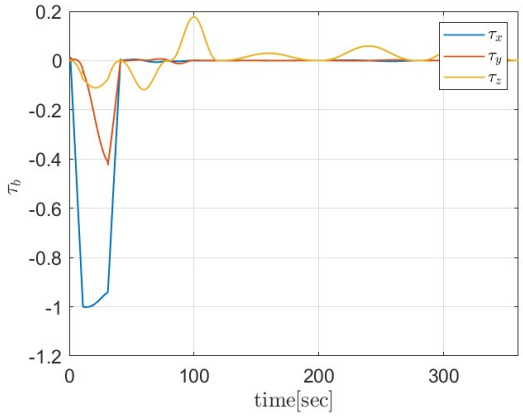


(a) Hierarchical controller e_p (b) Geometrical controller e_p

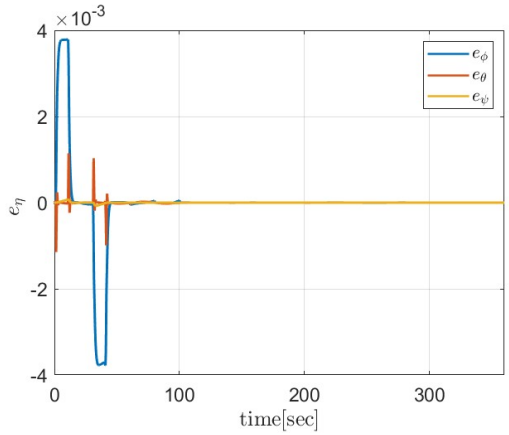
Figure 4.14: External disturbances trapezoidal, controller with estimator



(a) Hierarchical controller τ_b



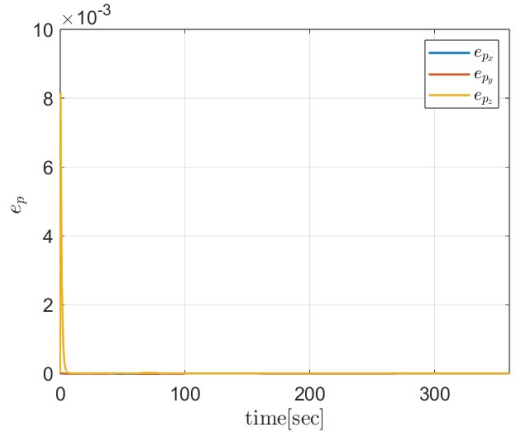
(b) Geometrical controller τ_b



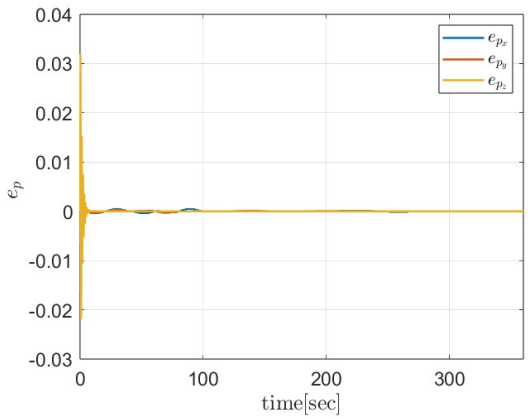
(c) Hierarchical controller e_η

Figure 4.15: External disturbances trapezoidal, controller with estimator

Other than the external disturbances, the momentum based estimator can compensate for an addiction unmodelled load. In fact from figures 4.17a and 4.17b we can observe that the the effect of the additional mass is seen as an extra force acting along the z axes, which is estimated and compensated, guaranteeing the convergence to zero of the position error (Fig 4.16).

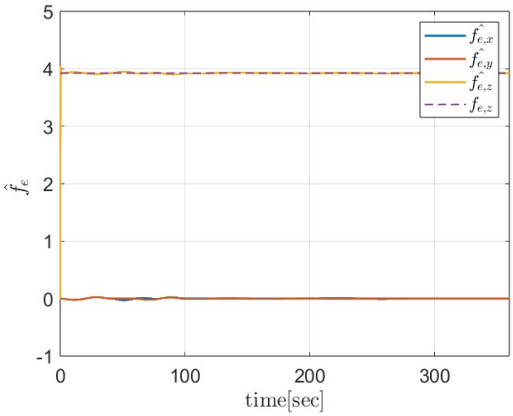


(a) e_p with compensation of mass Hierarchical

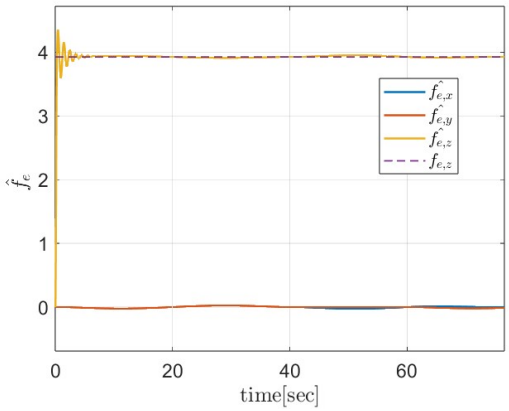


(b) e_p with compensation of mass geometrical

Figure 4.16

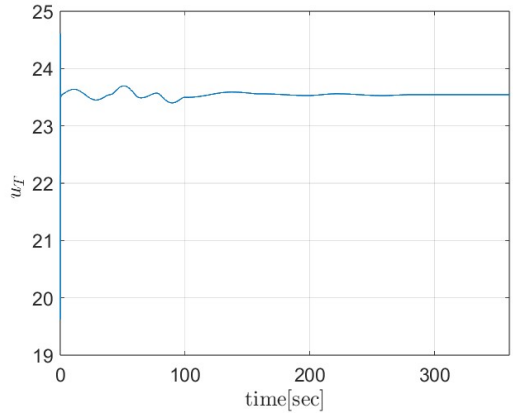


(a) Force estimation Hierarchical

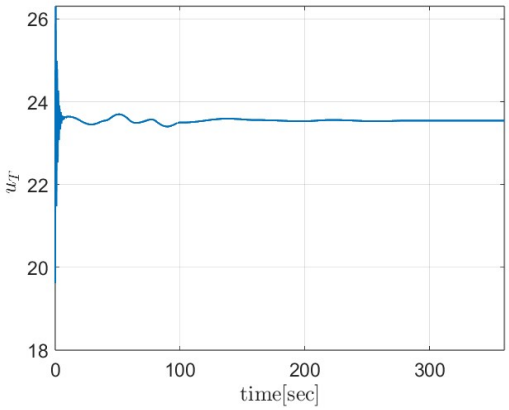


(b) Force estimation Geometrical

Figure 4.17



(a) u_T Hierarchical



(b) u_T Geometrical

Figure 4.18

Chapter 5

Aerial manipulator controller

In the previous chapter we have discussed the possibility to use an estimator to compensate for the external disturbances, but it can be easily implemented to compensate the presence of arm manipulator. In fact, there are many solutions in literature regarding the control of an aerial manipulator, for example we can consider the possibility to derivate the dynamic equations of the whole system, drone plus arm, using lagrangian formulation which lead to model of order 6+2 in this case. But in that case we should use a different approach to control this system. Instead, another solution consists in the independent control of the two system, using the typical control approach for both system, and then study the interactions which born among the two systems and compensate for those . In this section we will discuss about this last solution, explaining all the assumption made and discussing the achieved results.

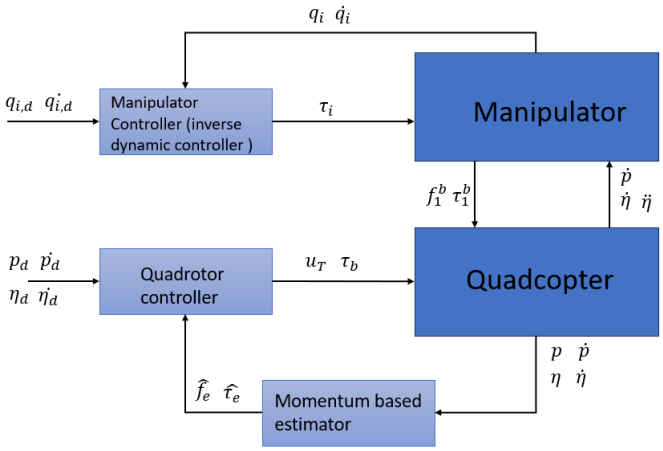


Figure 5.1: Aerial manipulator control scheme

5.1 Decentralized approach

As said in the first chapter, our drone must deliver a package, in particular near an high building, for example outside a balcony. Therefore the use of a little arm manipulator guarantee a more precise delivery operation. For the simulation, some assumptions has been made, in particular we imagine that the first link of the manipulator has been attached precisely in the centre of mass of the drone and that the velocities of the drone are not too fast, so they can be neglected inside internal controller for the manipulator. We can assume that the drone must deliver a light package around 0.012 kg, considering the weight of the manipulator itself. For the control of the manipulator, a classic inverse dynamic control scheme has been choosen, considering a well known model for the arm and also the mass of the package i known, so it can be compensated inside the controller. During the interval in which the drone must perform his trajectory, the arm is blocked in a rest position, then when the robot reach the final position, the robot arm torward the delivery position performing a line trajectory for the end effector of duration 10s. Also in this case a seventh order polynomial time law has been chosen, to guarantee smoothness also in the accelerations. In order to project the interaction of the arm on the body of the drone, it is useful to consider for the arm the Newton-Euler dynamic model which allows, using the

recursive formulation, to find the force and moment which a link impress to the previous one. In this case we are interested in the force that the first link can impress to the previous one (link 0), which coincide with the base of the manipulator.

$$\begin{aligned}\boldsymbol{f}_i^i =& \boldsymbol{R}_{i+1}^i \boldsymbol{f}_{i+1}^{i+1} + m_i \ddot{\boldsymbol{p}}_{C_i}^i \\ \boldsymbol{\mu}_i^i =& -\boldsymbol{f}_i^i \times \left(\boldsymbol{r}_{i-1,i}^i + \boldsymbol{r}_{i,C_i}^i\right) + \boldsymbol{R}_{i+1}^i \boldsymbol{\mu}_{i+1}^{i+1} \\ &+ \boldsymbol{R}_{i+1}^i \boldsymbol{f}_{i+1}^{i+1} \times \boldsymbol{r}_{i,C_i}^i + \overline{\boldsymbol{I}}_i^i \dot{\boldsymbol{\omega}}_i^i + \boldsymbol{\omega}_i^i \times \left(\overline{\boldsymbol{I}}_i^i \boldsymbol{\omega}_i^i\right) \\ &+ k_{r,i+1} \ddot{q}_{i+1} I_{m_{i+1}} z_{m_{i+1}}^i + k_{r,i+1} \dot{q}_{i+1} I_{m_{i+1}} \omega_i^i \times \boldsymbol{z}_{m_{i+1}}^i\end{aligned} \tag{5.1}$$

So practically if in the equation (5.1) we substitute $i = 0$ then we have the following external disturbances on the "link 0", which coincides unless the orientation with the body frame .

$$\boldsymbol{f}_{0,e} = \boldsymbol{R}_1^0 \boldsymbol{f}_1^1 \qquad \boldsymbol{\mu}_{0,e} = \boldsymbol{R}_1^0 \boldsymbol{\mu}_1^1 + \boldsymbol{R}_1^0 \boldsymbol{f}_1^1 \times \boldsymbol{r}_{0,C_0} \tag{5.2}$$

r_{0,C_i} is the vector from the origin of the frame 0 to centre of mass C_0 but is equal to zero because the frame 0 coincides with the body frame and this one is centred in the centre of mass of the drone(so practically O_b coincides with C_b). This wrench $\begin{bmatrix} f_1^1 & \mu_1^1 \end{bmatrix}^T$ (i.e. the wrench exerted by "link 1" on the "link 0") is expressed in the frame of the first link, so in order to project this wrench on the body of the drone we have to premultiply these wrench with the matrix R_1^b, which can be decomposed in the product $R_0^b\ R_1^0$ where R_1^0 can be found from the direct kinematics of the arm, while the rotation R_0^b is a fixed rotation along the x axes of -$pi/2$ (5.2).

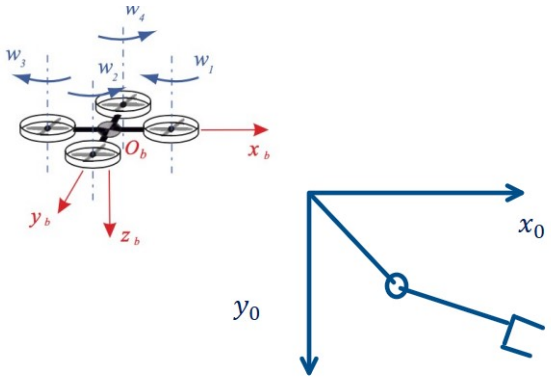
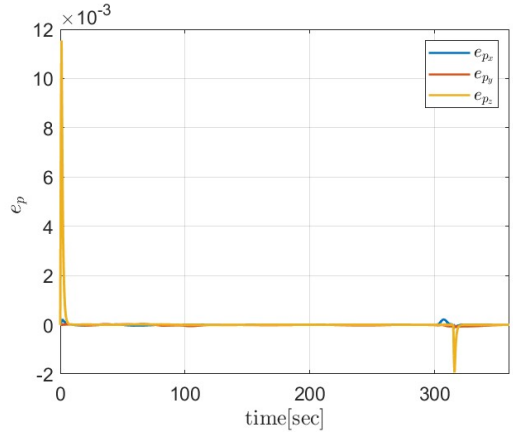


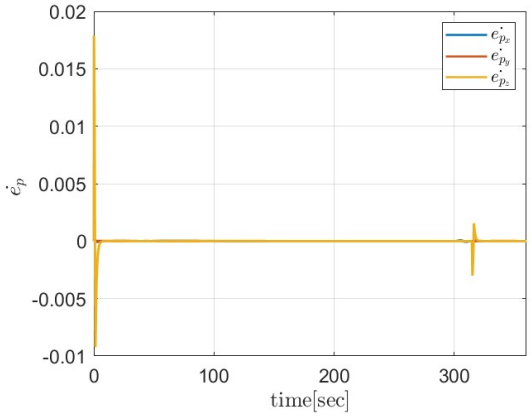
Figure 5.2: Aerial manipulator scheme

Then in order to correctly use the estimator these wrenches are projected to earth fixed frame using the matrix R_b which can be found from the dynamic model of the drone. In practice our manipulator starts from a position of rest so the external perturbations on the drone are an external force given by the weight force terms of the manipulator plus the mass and a τ_e. This configuration remains until simulation time equal to 305 and can be seen from the figure (5.5b) and (5.5a) how an additional mass leads to an increase in τ_b and u_T, after 10 second the additional mass is subtracted in the manipulator model, in fact from this moment the external force calculated by the momentum-based estimator decreases(look figure (5.6c),(5.6a) and (5.6b)) as well as τ_b and u_T. Looking the figure (5.3) and (5.4) we can note that the errors remain within an acceptable order of magnitude (remember that we take a sphere of radius 50 cm to create the path) and we also have in this case some overshooting of the error at the

beginning when we compensate the external wrench of the manipulator plus mass, and when the additional mass is delivered.

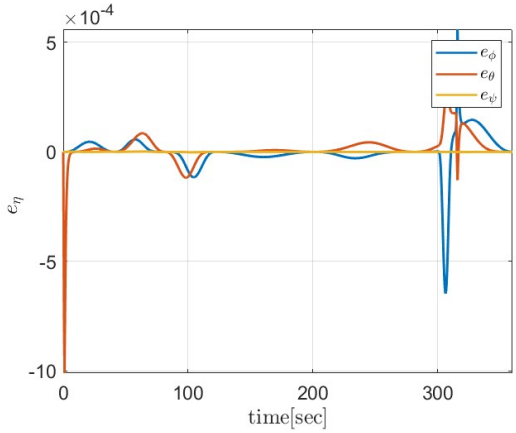


(a) e_p with arm manipulator

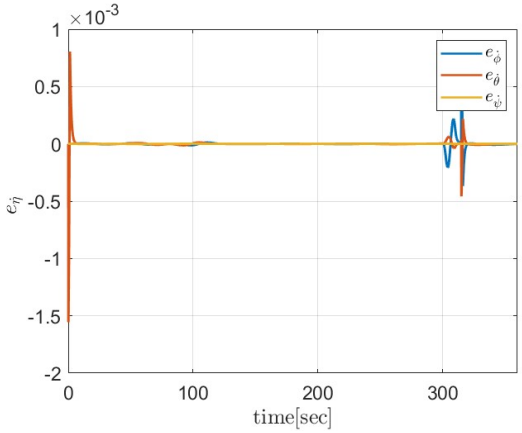


(b) $\dot{e}_p$ with arm manipulator

Figure 5.3



(a) e_η with arm manipulator

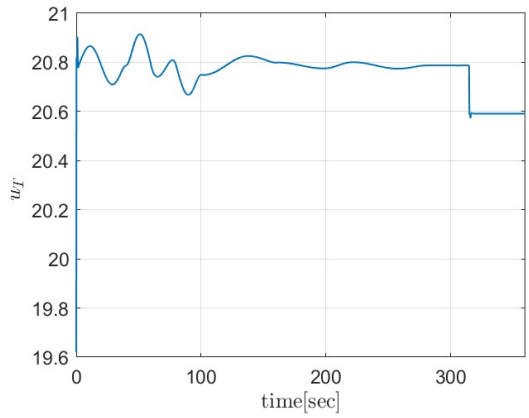


(b) $\dot{e}_\eta$ with arm manipulator

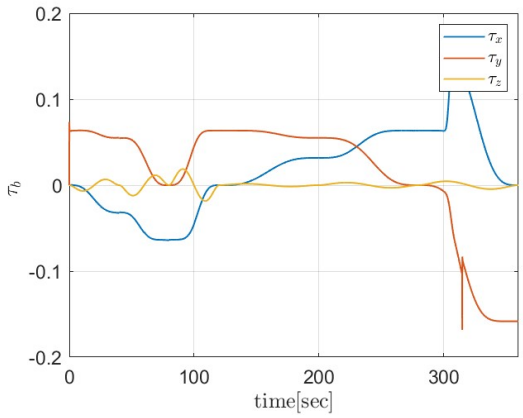
Figure 5.4

5.2 Bibiliography

B.Siciliano A.Ollero. Aerial Robotic Manipulation. Springer, 2019.
L.Villani G.Oriolo B.Siciliano, L.Sciavicco. Robotica: Modellistica, Pianificazione e Controllo. McGraw-Hill Libri Italia srl, 2010.
F.Ruggiero V.Lippiello. Cartesian impedance control of a uav with a robotic arm. International IFAC Symposium on Robot Contro, 2012.
J.A. Acosta Yarai E. Tlatelpa-Osorio, H. Rodr´ıguez-Cortes. A descentralized approach for the aerial manipulator trajectory tracking. International Conference on Unmanned Aircraft Systems (ICUAS), 2020.

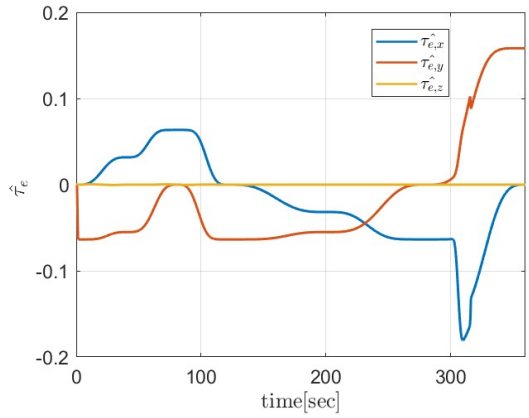


(a) u_T with arm manipulator

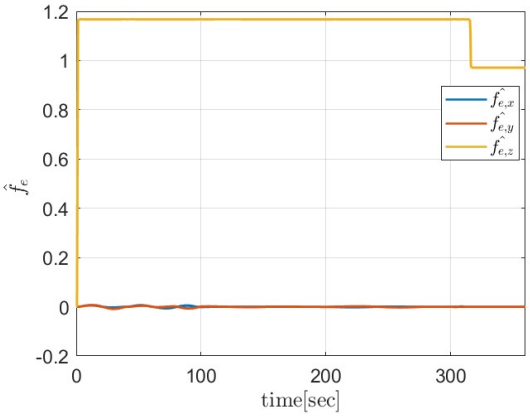


(b) τ_b with arm manipulator

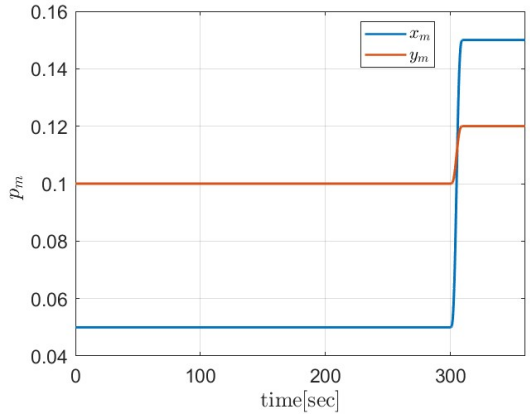
Figure 5.5



(a) $\hat{\tau}_e$ with arm manipulator



(b) $\hat{f}_e$ with arm manipulator



(c) manipulator trajectory

Figure 5.6