Politecnico di Milano
School of industrial and Information Engineering
Master of science in Aeronautical Engineering

Model-reference adaptive control of the vertical dynamics of a multirotor UAV

Advisor: Prof. Marco LOVERA
Co-Advisor: Eng. Mattia GIURATO

Thesis by:
William NGUEMEN DJONKOUA Matr. 894553

Academic Year 2019–2020

To my mother…

Acknowledgments

I would like to express my sincere gratitude to my supervisor, professor Marco Lovera, for allowing me to work on this wonderful project also for his helpful discussion, support and encouragement throughout this work.His vision and passion for research influenced my attitude for research work and spurred my creativity.I would also like to thank my co-Advisor ingineer Mattia Giurato for supporting me, helping me and make things easy for me when they seemed to be hard.

As the popular saying goes: you can choose your friends and relatives but not your family... Benjamin from a special, wonderful and adorable family who is THE DJONKOUA FAMILY.. you saw me being born, growing up, supervising, encouraging and supported. This scroll is just the result of your collective and individual efforts, I can never thank you enough, so please find in these words the THANKS of the youngest who becomes the result of your efforts.

Abstract

In the previous decade, a quick spread of UAV multicopter has been observed, whose application field is constantly expanding.Most of the time the focus has been on their autonomy, flight control and high maneuverability.Must of the times, phenomena happening during low altitude flight is usually ignored.So it is important for the vehicle which will operate in such condition to have a good controller who can compensate those effects.
The first step in control development is an adequate dynamic system modelling, which should involve a faithful mathematical representation of the mechanical system.In this thesis, in order to deal with unknown parameters and unknown disturbances affecting the vertical dynamic of the quadcopter, we designed a MRAC-observer baseline adaptive controller such a ways that beside an existing baseline control input, the adaptive control input can be activated and deactivated when needed.

This Thesis presents a detailed dynamic analytical model of the quad-rotor helicopter using the linear Taylor series approximation method.It also presents an MRAC observer baseline adaptive controller technique, used to control the dynamics vertical equations of motion of the drone. The designed control technique is then implemented, inside MATLAB/SIMULINK, and tested on the quadcopter simulator.

Contents

List of Figures

Introduction

This is the report of the application of model reference adaptive control to control the vertical dynamic equation of motion of a quadcopter.This chapter aims to give the reader an overview and introduction to this investigated problem.A brief overview of the disposition of the report is given.

Background

The use of unmanned areal vehicles (UAVs), or drones has many interesting applications. Beyond the uses within military applications, UAVs can perform search and rescue operations in hazardous environments, surveillance and inspections of hard to reach places (Waharte and Trigoni, 2010; Nikolic et al., 2013.). UAVs can even be used as lifted base (called here fixed UAV) allowing another UAV(called here rescued UAV) to land on up of the fixed UAV.In this particular mission, where the mass of the fixed UAV can vary and where the fixed UAV can face several disturbances as for example the force generated by the rescued UAV and the ground effect if the fixed UAV operates close to the ground, is then important to design a good controller capable of dealing with parameter uncertainties and unknown disturbances.

Goals

The main objective for this master's thesis was to control the vertical dynamic equations of motion of the UAV facing the problem like uncertainty parameters and unknown disturbances. To overcome this problem, we developed an model reference adaptive control observer based augmentation (MRAC-OBA).

Thesis structure

To facilitate the reading, the organization of the thesis structure is provided:

- Chapter 1: An overview of the model reference adaptive control theory is provided.

- Chapter 2: The mathematical description of the quadcopter under study is derived.

- Chapter 3: The application of the MRAC method to the linear dynamic equation of motion of the quadcopter.

- Chapter 4: Finally the implementation and simulation of the proposed controller is done using MATLAB/SIMULINK and the simulator model provided by the FLYART laboratory of politecnico di Milano.

Chapter 1

Model Reference Adaptive Control

An adaptive controller is capable of achieving good performance in the presence of significant parametric uncertainties, and even without the full knowledge of the plant[3].The Model Reference Adaptive Control (MRAC) was originally proposed by Whitaker et.al [4] in 1958,and this control method is still actively studied today.
MRAC has three major components: reference model, weight (gain) update law, and controller.As the figure 1.1 shows, the reference model specifies the desired behaviour of the closed-loop system.The output of the system to be controlled is compared to the output of the reference model.This comparison results in an error signal used in the weight update law.The controller employs the weight information from the weight update law to form the adaptive control signal.

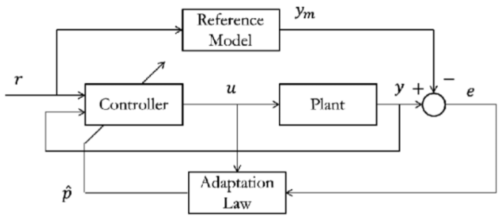


Figure 1.1: MRAC scheme

In this thesis instead of following the traditional MRAC theory, we used a modified version called here MRAC observer-based augmentation (MRAC-OBA). (see figure 1.2)

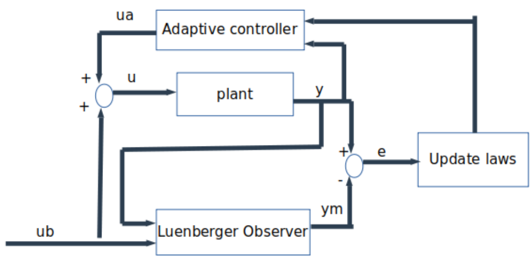


Figure 1.2: MRAC OBA scheme

This MRAC-OBA adds the adaptive control input(u_a) to the baseline control input (u_b), to form the total control input u, such that the u_a input can be activated when needed to face problems like parameters uncertainties,disturbances acting on the system.
Since we do not have access to the true state of the system,we will build an Luenberger observer to estimate them and use it as the reference model.

1.1 Uncertain system

Let's consider the following uncertain system.

$$\dot{x} = Ax + Bu + B_1 d + B_2 d_1 \tag{1.1}$$

$$y = Cx \tag{1.2}$$

Where $x \in R^n$ is the state vector,$y \in R^p$ is the system output.
$u \in R^m$ is the control input.$d \in R^r$ and $d_1 \in R^q$ are disturbances acting on the system.$A \in R^{nxn}$,$C \in R^{pxn}$,are known matrices. (A, C) being observable pair.
$B \in R^{nxm}$ is an unknown matrice and the pair (A, B) is assumed to be controllable.
$B_1 \in R^{nxr}$ and $B_2 \in R^{nxq}$ are matrices.
The multiplicative technique is used to model the unknown B matrix as follows

$$B = B_o \Lambda \tag{1.3}$$

Where $B_o \in R^{nxm}$ and Λ are respectively the nominal and uncertain part of matrix B.We assumed in this thesis that Λ is a quadratic diagonal positive and invertibile matrix.
Let's assume that B_2 is not proportional to B_o and that B_1 is proportional to B_o.

$$B_1 = B_o K_{b1}; \quad B_2 \neq B_o K_{b2} \tag{1.4}$$

Then:

$$\dot{x} = Ax + B_o\Lambda(u + \Lambda^{-1}K_{b1}d) + B_2d_1 \tag{1.5}$$

Since d enters in the control input while d_1 does not enter in the control input,d is called matched uncertainty and d_1 is called unmatched uncertainty.
To proceed we will define the total matched uncertainty as $d = \Lambda^{-1}K_{b1}d$.

From now on,we will neglect the unmatched uncertainty since in this thesis we are dealing with a system with only matched uncertainty.

$$\dot{x} = Ax + B_o\Lambda(u + d) \tag{1.6}$$

The total inputs is defined as follows:

$$u = u_a + u_b \tag{1.7}$$

where u_a and u_b are respectively the adaptive control inputs and the baseline control inputs.
Adding and subtracting BK_1x and BK_2u_b to the system (1.6) allows us to change the dynamics of the system as equation 1.8 shows:

$$\dot{x} = A_rx + B_ru_b + B_o\Lambda(u_a + W^Tf) \tag{1.8}$$

where:

$$A_r = A - B_oK_1 \tag{1.9}$$

$$B_r = B_oK_2 \tag{1.10}$$

$$W^T = \begin{bmatrix}\Lambda^{-1}K_1 & (I - \Lambda^{-1}K_2) & d\end{bmatrix};\ f = \begin{bmatrix} x \\ u_b \\ col \end{bmatrix} \tag{1.11}$$

and col is a column vector of elements all equals to 1.

Now let's design the adaptive control input in the following ways

$$u_a = -\hat{W}^Tf \tag{1.12}$$

where $\hat{W}$ is the estimate of W.

1.2 Reference system

Since we do not have access to the state of the system,we will design a Luenberger state observer and we will use it as our reference model.

$$\dot{\hat{x}} = A_r\hat{x} + B_rU_b + L(y - \hat{y}) \tag{1.13}$$

$$\hat{y} = C\hat{x} \tag{1.14}$$

Assuming that the pair (A_r, C) is observable,we can design the gain L in such a way that the matrix $(A_r - LC)$ has all its eigenvalues with negative real part.

1.3 Tracking error and update laws

Here,we want to design the update laws such that the dynamic equation of the state estimation error (equations 1.16) will be an asymptotic stable system.

Let introduce the state estimation error:

$$e = x - \hat{x} \tag{1.15}$$

Its time derivative is given by

$$\dot{e} = (A_r - LC)e - B_o\Lambda(\Delta W^T f) \tag{1.16}$$

with $\Delta W = \hat{W} - W$

The Lyapunov stability theory is used to design the update laws.Let define the positive definite Lyapunov's candidate function:

$$V(e, \hat{W}) = e^T Pe + trace(\Delta W^T \Gamma^{-1} \Delta W \Gamma) \tag{1.17}$$

where the matrix P,solution of the equation 1.21,must be symmetric positive definite matrix.
The derivative of $V(e, \hat{W})$ along the trajectories of system 1.16 is given by:

$$\dot{V}(e, \hat{W}) = -e^T Qe + 2trace(\Delta W^T (\Gamma^{-1}\Delta\dot{W} - fe^T PB_o)\Lambda) \tag{1.18}$$

If we design the update law as following,

$$\Gamma^{-1}\Delta\dot{W} - fe^T PB_o = 0 \implies \dot{\hat{W}} = \Delta\dot{W} = \Gamma fe^T PB_o \tag{1.19}$$

the derivative of the Lyapunov candidate becomes:

$$\dot{V} = -e^T Q e \tag{1.20}$$

where

$$Q = -(A_r - LC)^T P - P(A_r - LC) \tag{1.21}$$

In order to have asymptotic stability the chosen matrice Q,must be positive definite.

1.4 Controlled system

Finally,the MRAC-OBA gave to us the following controlled system:

$$\dot{x} = Ax + B_o \Lambda (u + d) \tag{1.22}$$

$$\dot{\hat{x}} = A_r \hat{x} + B_r u_b + L(y - \hat{y}) \tag{1.23}$$

$$\dot{e} = (A_r - LC)e - B_o \Lambda (\Delta W^T f) \tag{1.24}$$

$$\Delta \dot{W} = \dot{\hat{W}} = \Gamma f e^T P B \tag{1.25}$$

$$y = Cx; \hat{y} = C\hat{x} \tag{1.26}$$

$$e = x - \hat{x} \tag{1.27}$$

$$u_a = -\hat{W}^T f \tag{1.28}$$

$$u = u_a + u_b \tag{1.29}$$

Chapter 2

Quadcopter

In this chapter we will derive the equations of motion of the quadcopter. We will linearize them around the hovering equilibrium point.

2.1 Reference frames

Generally for the quadcopter, it is possible to use two reference frames: The Earth fixed frame and the body fixed frame.

2.1.1 Earth fixed frame

The hypotheses of flat and still Earth surface are made. For these reasons a fixed frame $F_E = \{O, N, E, D\}$ attached to the Earth can be considered as an inertial reference system. The origin can be an arbitrary fixed point on the Earth. The standard convention (Figure: 2.1) provides the N axis pointing North, the E axis East and the D axis aligned with the direction of gravity, pointing downward. This reference system is also known as the NED (meaning North-East-Down) frame.

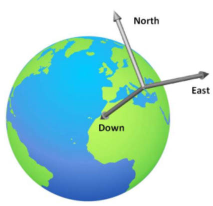


Figure 2.1: NED frame

2.1.2 Body fixed frame

Since it is easy to measure data from the body fixed frame, it is convenient to derive the equations of motion in that body frame $F_B = \{O_{ABC}, X_B, Y_B, Z_B\}$. This frame, in fact, has the origin in the center of gravity of the quadcopter and changes its orientation with it. The X_B axis is parallel to the longitudinal axis of the quadcopter, the Z_B axis lays in the plane of symmetry pointing downward and the Y_B axis is found according to the right-handed rule. Figure 2.2 shows the NED and the body fixed frame together.

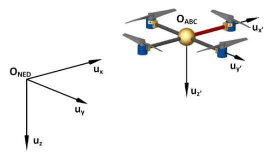


Figure 2.2: NED and body frame

2.2 Euler angles and three-dimensional rotations

One of the methods that allow to switch from a Cartesian coordinate system to another one is based on the definition of three independent parameters, able to describe the relative orientation of the two sets of reference axes. The Euler angles (ϕ, θ, ψ) are three independent angular quantities used to define the position of a generic reference frame $F_1 = (X_1, Y_1, Z_1)$ with respect to an inertial reference frame(or with respect to another set of three axes $F_2 = (X_2, Y_2, Z_2)$).

Problem: the components of a vector with respect to the fixed triad $F_1(X_1, Y_1, Z_1)$ with unit vectors $(\vec{i}, \vec{j}, \vec{k})$ must be processed in a second reference system $F_2(X_2, Y_2, Z_2)$, with unit vectors $(\vec{e_1}, \vec{e_2}, \vec{e_3})$, rotated with respect to frame F_1. In the present case the three rotations are applied to the triad F_1 to bring it to coincide with the triad F_2:

- First rotation (yaw angle ψ): Positive rotation around Z_1 in order to define a set of intermediate axes $F_1' = (X_1', Y_1', Z_1' = Z_1)$ and a rotation matrix R_ψ.

$$R_\psi = \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

- Second rotation (pitch angle θ) around the axis Y_1' of F_1' getting

another intermediate reference system
$F"_1 = (X"_1, Y"_1 = Y_1', Z"_1)$ and a rotation matrix R_θ.

$$R_\psi = \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \tag{2.2}$$

- Third rotation (rol angle ϕ): around the axis $X"_1$ of $F"_1$ to align F_1 with $F_2 = (X_2 = X"_1, Y_2, Z_2)$.The rotation matrix is given by R_ϕ.

$$R_\psi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\psi & -s_\psi \\ 0 & s_\psi & c_\psi \end{bmatrix} \tag{2.3}$$

By combining them, we can obtain the rotation matrix from body fixed frame to NED frame.

$$R_{\psi\theta\phi} = \begin{bmatrix} c_\psi c_\theta & s_\phi c_\psi s_\theta - s_\psi c_\phi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ s_\psi c_\theta & s_\psi s_\phi s_\theta + c_\psi c_\phi & c_\phi s_\theta s_\psi - c_\psi s_\phi \\ -s_\theta & s_\phi c_\theta & c_\theta c_\phi \end{bmatrix} \tag{2.4}$$

where: $R_{\psi\theta\phi} = R_\psi R_\theta R_\phi$ and c_x, s_x and t_x stand respectivelly for $\cos(x), \sin(x)$ and $\tan(x)$.
Consider a vector $\vec{V}$ expressed in NED frame by a vector column as:

$$V_{NED} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{2.5}$$

and the same vector $\vec{V}$ expressed in body frame by a column vector as:

$$V_B = \begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} \tag{2.6}$$

we can relate them using the rotation matrix as follows:

$$V_{NED} = R_{\psi\theta\phi} V_B \tag{2.7}$$

Let's introduce the time derivative of Euler angles

$$\dot{e}_{NED} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{2.8}$$

and the angular velocity of the body frame with respect to the NED frame, expressed in body fixed frame

$$\Omega_B = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2.9}$$

The time derivative of the Euler angles and the angular velocity are linked by a transformation matrix T as follows:

$$T = \begin{bmatrix} 1 & s_\phi t_\theta & t_\theta c_\phi \\ 0 & c_\phi & -s_\phi \\ 0 & \frac{s_\phi}{c\theta} & \frac{c_\phi}{c\theta} \end{bmatrix} \tag{2.10}$$

$$\dot{e}_{NED} = T\Omega_B \tag{2.11}$$

2.3 Flight dynamics equations

The dynamic equilibrium of an aircraft can be expressed by two vectorial equations:

$$F_a + F_r + F_i = 0 \tag{2.12}$$

$$M_a + M_r + M_i = 0 \tag{2.13}$$

where a,r and i indexes refer to applied, reaction and inertialess respectively. For an aircraft in flight, the reaction forces and moments are null. The inertial forces and moments can be defined in an inertial reference frame as:

$$F_i = -\frac{dQ}{dt} \tag{2.14}$$

$$M_i = -\frac{dK}{dt} - v_p \Lambda Q \tag{2.15}$$

where Q is the momentum, K is the moment associated with the momentum, P is the reference point and Λ is the cross product operator. Then, considering the reference point P coinciding with the center of gravity, the applied forces and moments are:

$$F_a = \frac{dQ}{dt} \tag{2.16}$$

$$M_a = \frac{dK}{dt} \tag{2.17}$$

Translational equations of motion

Assuming that the mass of the UAV is constant in time, using equations (2.16) and using Poisson's formulas, it is possible to write the linear motion equation:

$$m(V_G + \Omega_B \Lambda V_G) = F_g + F_{prop} \tag{2.18}$$

$$V_G = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{2.19}$$

where V_G is the velocity of the center of mass of the multirotor and the vectors F_g , F_{prop} represent, respectively the gravity force and the forces generated by the UAV propellers.

Angular equations of motion

The definition of the inertia matrix about the center of gravity is introduced:

$$J_G = \begin{bmatrix} J_{xx} & -J_{xy} & J_{-xz} \\ -J_{xy} & J_{yy} & -J_{yz} \\ -J_{xz} & J_{yz} & J_{zz} \end{bmatrix} \tag{2.20}$$

where:

$$J_{xx} = \int (y^2 + z^2)dm, \;\; J_{yy} = \int (x^2 + z^2)dm, \;\; J_{zz} = \int (y^2 + x^2)dm \tag{2.21}$$

$$J_{xy} = \int (xy)dm, \;\; J_{xz} = \int (xz)dm, \;\; J_{yz} = \int (yz)dm \tag{2.22}$$

We assumed that the body fixed frame is principal of inertia who means $J_{xy} = J_{xz} = J_{yz} = 0$. Considering now equation (2.17) , the angular motion equation can be obtained:

$$J_G \dot{\Omega}_B + \Omega_B \Lambda (J_G \Omega_B) = M_{prop} \tag{2.23}$$

where the moment of the gravity force is zero, since we are evaluating the moment with respect to the center of gravity, and M_{prop} is the propeller's torque.

2.4 Actuator model

The quadcopter can have different configurations, but in this thesis we focused on the X-configuration as Figure (2.3) shows.

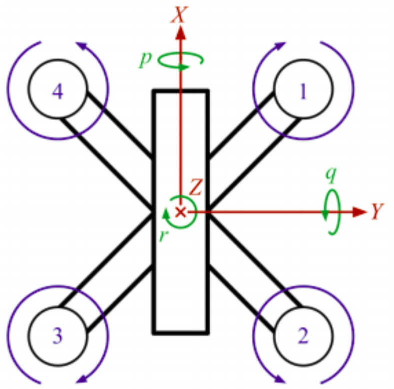


Figure 2.3: an illustration of the quadcopter's body-fixed coordinate system(red),the angular rates(green),along with the position and direction of each engine(blue)

The forces and moments are expressed in the body fixed frame.
We assume that each propeller generates a force in the direction of the z axis of the body fixed frame:

$$T_i = K_{T_i}\Omega_i^2, \ i = 1, 2, 3, 4 \tag{2.24}$$

and a torque, in the direction of the z axis of the body fixed frame

$$\tau_i = K_{b_i}\Omega_i^2, \ i = 1, 2, 3, 4 \tag{2.25}$$

where K_{T_i} , K_{b_i} and Ω_i are respectively thrust coefficient, torque coefficient and propeller angular speed.

Then:

$$F_{prop} = \begin{bmatrix} 0 \\ 0 \\ -T_1 - T_2 - T_3 - T_4 \end{bmatrix} \tag{2.26}$$

$$M_{prop} = \begin{bmatrix} (-T_1 - T_2 + T_3 + T_4)ls_\alpha \\ (T_1 - T_2 - T_3 + T_4)lc_\alpha \\ -\tau_1 + \tau_2 - \tau_3 + \tau_4 \end{bmatrix} \tag{2.27}$$

Where l stands for a distance between the propeller and the center of gravity of the quadcopter. α is half the angle between the propeller arm 1 and arm 4. Let's introduce the following quantities:

$$u_1 = -T_1 - T_2 - T_3 - T_4 \tag{2.28}$$

$$u_2 = (-T_1 - T_2 + T_3 + T_4) l s_\alpha \tag{2.29}$$

$$u_3 = (T_1 - T_2 - T_3 + T_4) l c_\alpha \tag{2.30}$$

$$u_4 = -\tau_1 + \tau_2 - \tau_3 + \tau_4 \tag{2.31}$$

2.5 Equations of motion

By taking advantage of everything we have said so far, we are able to derive the following nonlinear equations of motion:

$$\dot{u} = -g s_\theta - qw + rv \tag{2.32}$$

$$\dot{v} = g s_\phi c_\theta + pw - ru \tag{2.33}$$

$$\dot{w} = g c_\theta c_\phi - pv + qu + \frac{u_1}{m} \tag{2.34}$$

$$\dot{p} = \frac{1}{J_{xx}}(u_2 - qr(J_{zz} - J_{yy})) \tag{2.35}$$

$$\dot{q} = \frac{1}{J_{yy}}(u_3 - pr(J_{xx} - J_{zz})) \tag{2.36}$$

$$\dot{r} = \frac{1}{J_{zz}}(u_4 - pq(J_{yy} - J_{xx})) \tag{2.37}$$

where g is the gravity acceleration and m is the mass of the quadcopter.

By adding equations (2.7), (2.11) to those equation, the whole system becomes a system with 12 states:

$$x = [x, y, z, \psi, \theta, \phi, u, v, w, p, q, r]^T$$

and 4 controls input:

$$u = [u_1, u_2, u_3, u_4]^T$$

Equilibrium point:

In this thesis, we have taken the equilibrium point when the quadcopter is hovering:

$$x = 0,\ u_1 = -mg,\ u_2 = u_3 = u_4 = 0 \tag{2.38}$$

where x is the state of the system.

Linearized equations of motions

Due to the goal of this thesis, we only reported here the linear vertical dynamic equation of motion of the quadcopter linearized around the equilibrium point:

$$\dot{z} = w \tag{2.39}$$

$$\dot{w} = g + \frac{u_1}{m} \tag{2.40}$$

Chapter 3

Altitude dynamics

In this chapter, we will use the MRAC-OBA technique introduced in Chapter 1, to control the vertical motion of the quadcopter.

3.1 Vertical dynamics

The vertical dynamic equation of motion is defined by equation (3.1) and equation (3.2)

$$\dot{z} = w \tag{3.1}$$

$$\dot{w} = g + \frac{1}{m}(u_1 + d) \tag{3.2}$$

where d represents the total disturbance acting on the system.

State space model:

To facilitate the manipulation, we will rewrite the equations of motion of the system in the state space form.

$$\dot{x} = Ax + B(u_1 + d) + B_1 g \tag{3.3}$$

where:

$$x = \begin{bmatrix} z \\ w \end{bmatrix}, \; u_1 = u_a + u_b \tag{3.4}$$

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \; B = \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}, \; B_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{3.5}$$

and m is assumed to be the uncertain mass of the system. Equation (3.6) shows how the multiplicative technique is used to model the mass of the system:

$$m = m_{nom} + m_{unc} = m_{nom}\beta \tag{3.6}$$

$$\beta = 1 + \frac{m_{unc}}{m_{nom}} \tag{3.7}$$

$$\frac{1}{m} = \frac{1}{m_{nom}}\Lambda \tag{3.8}$$

$$\Lambda = \frac{1}{\beta} \tag{3.9}$$

where m_{nom} and m_{unc} are respectively the nominal and the uncertain part of the mass.
We notice that Λ is positive and invertible and also observe that:

$$B = B_o\Lambda, B_o = \begin{bmatrix} 0 \\ \dfrac{1}{m_{nom}} \end{bmatrix} \tag{3.10}$$

$$B_1 = B_o m_{nom} \tag{3.11}$$

Then system (3.3) becomes:

$$\dot{x} = Ax + B_o\Lambda(u_a + u_b + d + m_{nom}\Lambda^{-1}g) \tag{3.12}$$

Adding and subtracting B_oK_1x and $B_oK_2u_b$ to the system (3.12), allows us to change the dynamics of the system into equation (3.13)

$$\dot{x} = A_rx + B_ru_b + B_o\Lambda(u_a + W^Tf) \tag{3.13}$$

where:

$$A_r = A - B_oK_1 \tag{3.14}$$

$$B_r = B_oK_2 \tag{3.15}$$

$$W^T = \begin{bmatrix} m_{nom}\Lambda^{-1} & \Lambda^{-1}K_1 & (I - \Lambda^{-1}K_2) & d \end{bmatrix}; \; f = \begin{bmatrix} g \\ x \\ u_b \\ 1 \end{bmatrix} \tag{3.16}$$

Adaptive input

Let's design the adaptive control input as follows

$$u_a = -\hat{W}^Tf \tag{3.17}$$

where $\hat{W}$ is the estimate of W.

Equation (3.18) describes the Luenberger state observer which is used as the reference system:

$$\dot{\hat{x}} = A_r\hat{x} + B_r u_b + L(y - \hat{y}) \tag{3.18}$$

$$\hat{y} = C\hat{x} \tag{3.19}$$

By observing that the system (A, B_o) is controllable, the use of the pole placement technique allows us to design the matrix A_r in a way that the pair (A_r, C) will be observable, so that the same pole placement technique can be used to design the observer gain L. After giving different values to the parameter K_1 and giving different desired eigenvalues(pol) for the matrix $(A_r - LC)$, we observed that the observer was having a good state estimation which the following values:

$$K_1 = \begin{bmatrix} 60.4 & 21.14 \end{bmatrix} \tag{3.20}$$

$$A_r = \begin{bmatrix} 0 & 1 \\ -40 & -14 \end{bmatrix},\ B_r = \begin{bmatrix} 0 \\ \frac{1}{m_{nom}} \end{bmatrix},\ m_{nom} = 1.510kg \tag{3.21}$$

$$pol = \begin{bmatrix} -40 & -160 \end{bmatrix} \tag{3.22}$$

$$L = \begin{bmatrix} 40 & -40 \\ 1 & 146 \end{bmatrix} \tag{3.23}$$

the matrix B_r is given considering $K_2 = 1$.

<u>Tracking error and update laws:</u>

Let's introduce the state estimation error:

$$e = x - \hat{x} \tag{3.24}$$

Its time derivative is given by

$$\dot{e} = (A_r - LC)e - B_o\Lambda(\Delta W^T f) \tag{3.25}$$

with $\Delta W = \hat{W} - W$.

Now let's use the Lyapunov stability theory to design the update laws such a way that the system (3.25) is asymptotically stable. Define the positive definite Lyapunov's candidate function as

$$V(e, \hat{W}) = e^T P e + trace(\Delta W^T \Gamma^{-1} \Delta W \Gamma) \tag{3.26}$$

and its derivative along the trajectory of system (3.25) is given by:

$$\dot{V}(e, \hat{W}) = -e^T Qe + 2trace(\Delta W^T(\Gamma^{-1}\Delta\dot{W} - fe^T PB_o)\Lambda) \tag{3.27}$$

If we design the update law as follows

$$\Delta\dot{W} = \dot{\hat{W}} = \Gamma fe^T PB_o \tag{3.28}$$

the derivative of the Lyapunov candidate becomes:

$$\dot{V} = -e^T Qe, \tag{3.29}$$

where:

$$Q = -(A_r - LC)^T P - P(A_r - LC) \tag{3.30}$$

After taking Q as an identity matrix we solve the equation 3.30 for P. We found out that P is symmetric positive definite:

$$P = \begin{bmatrix} 0.0109 & 0.0015 \\ 0.0015 & 0.0035 \end{bmatrix} \tag{3.31}$$

So we can conclude that the system (3.25) is asymptotically stable.

3.2 Controlled system

The controlled system is given by the following equations:

$$\dot{x} = Ax + B_o\Lambda(u_1 + d + m_{nom}\Lambda^{-1}g) \tag{3.32}$$

$$\dot{\hat{x}} = A_r\hat{x} + B_r u_b + L(y - \hat{y}) \tag{3.33}$$

$$\dot{e} = (A_r - LC)e - B_o\Lambda(\Delta W^T f) \tag{3.34}$$

$$\Delta\dot{W} = \dot{\hat{W}} = \Gamma fe^T PB_o \tag{3.35}$$

$$u_a = -\hat{W}^T f \tag{3.36}$$

$$u_1 = u_a + u_b \tag{3.37}$$

$$e = x - \hat{x} \tag{3.38}$$

$$y = Cx; \hat{y} = C\hat{x} \tag{3.39}$$

Chapter 4

Simulation

In this chapter two simulations were done using two different models of the vertical dynamic equation of the quadcopter.
In the first simulation, we designed our own baseline controller and the model used to run the simulation is the uncertain linear model derived in equation (3.1) and equation (3.2)
The second simulation is done using the quadcopter simulator provided by the FlyART laboratory of politecnico di Milano. In this second simulation, the baseline control input is already implemented inside the simulator.

4.1 Simulation using the uncertain linear model

From equation (3.13), it is evident that we want the baseline control input to control the system equations (4.1)

$$\dot{x} = A_r x + B_r u_b \tag{4.1}$$

Let's assume we want to stabilize the system in a given set point z_{ref} then we can define the error state as $e = z - z_{ref}$.
Now let augment the system equation (4.1) with the integral of the error as follows:

$$\dot{x} = A_r x + B_r u_b \tag{4.2}$$

$$y = z = C_r x, C_r = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{4.3}$$

$$\dot{r} = z - z_{ref} \tag{4.4}$$

In state space form we have

$$\begin{bmatrix} \dot{x} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} A_r & 0 \\ C_r & 0 \end{bmatrix} \begin{bmatrix} x \\ r \end{bmatrix} + \begin{bmatrix} B_r \\ 0 \end{bmatrix} u_b + \begin{bmatrix} 0 \\ 1 \end{bmatrix} z_{ref} \tag{4.5}$$

Let's define some matrice

$$A_1 = \begin{bmatrix} A_r & 0 \\ C_r & 0 \end{bmatrix}, B_1 = \begin{bmatrix} Br \\ 0 \end{bmatrix} \tag{4.6}$$

Since the system is controllable it is easy and fast for us to design the baseline control who can change the position of the eigenvalues of the systems.

$$u_b = -K_t \begin{bmatrix} x \\ r \end{bmatrix} \tag{4.7}$$

In our case, after changing several times the position of the desired eigenvalues of the matrix A_1, it came out that the following eigenvalues $Polref = -5; -10; -20$ was well stabilizing the system to the desired set point z_{ref}.

Then

$$K_t = \begin{bmatrix} 468.1 & 31.7 & 1510 \end{bmatrix} \tag{4.8}$$

<u>Simulation</u>

Figure (4.1) represents the modelisation of the uncertain mass of the system.

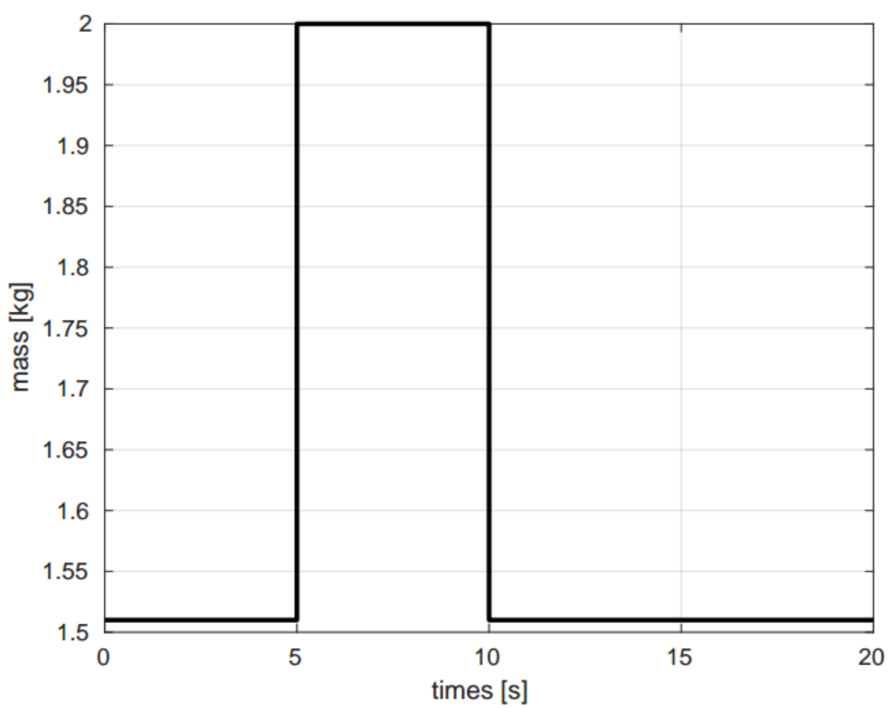


Figure 4.1: Temporal trend of the mass of the system

The set point is $z_{ref} = -1m$ altitude. We also considered that after 10 seconds a step disturbance of final value of 50Newton(Yes, 50 Newton are very big, but it is a good test to see how good the controller can reject disturbances.) will occur. In Figure 4.2 it is possible to see the implementation of the controlled system in MATLAB/SIMULINK.

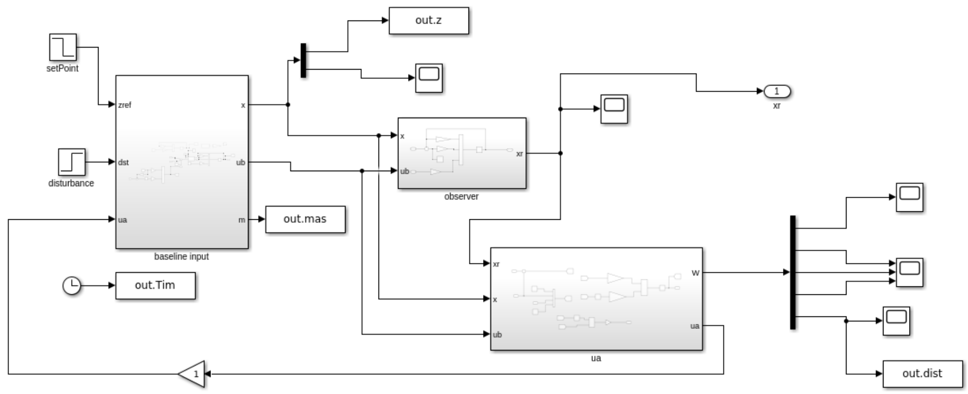


Figure 4.2: Uncertain model simulator

When we compare Figure 4.3 (where we do not consider the action of the adaptive control input) with Figure 4.4 (where we consider the adaptive control input), we can see how the adaptive control input reduces the effect of the uncertain mass and the step disturbance acting on the system. So in presence of adaptive control input the system is more robust.

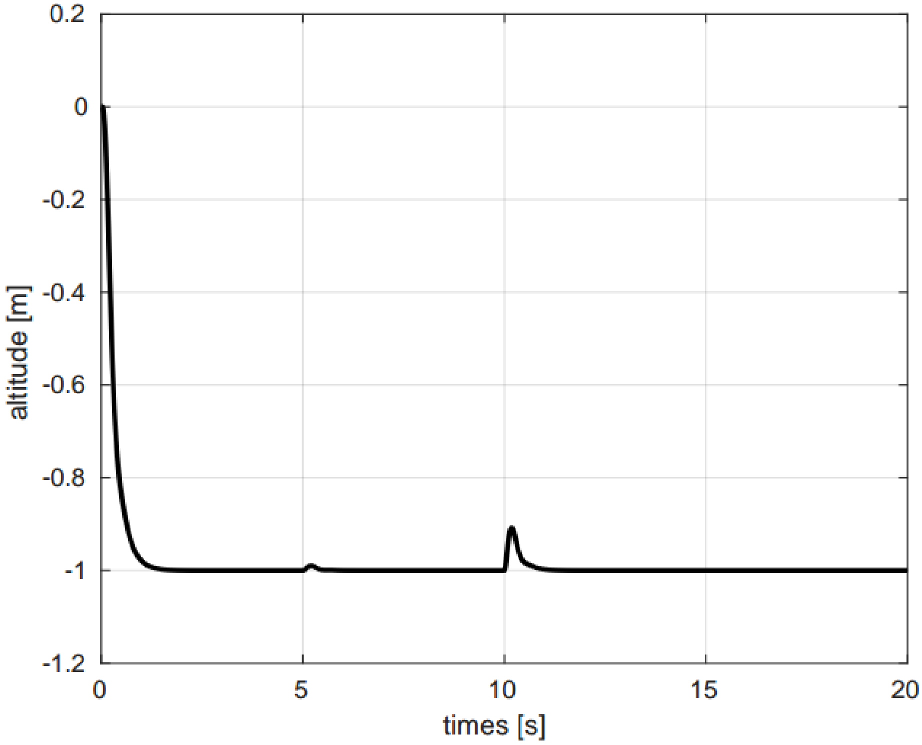


Figure 4.3: Altitude response without adaptive control part

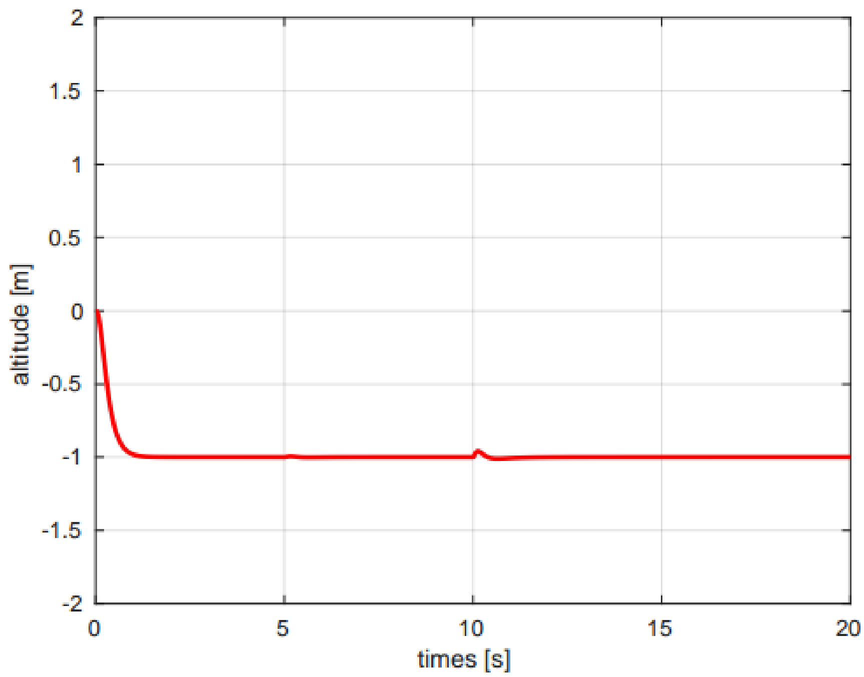


Figure 4.4: Altitude response with adaptive control part

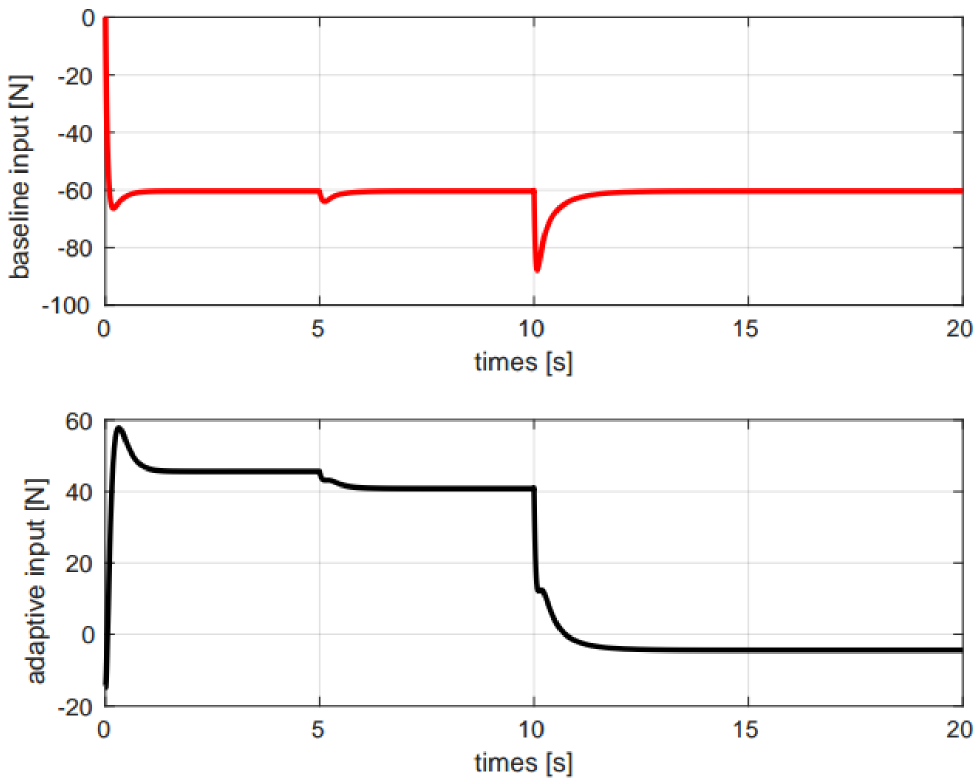


Figure 4.5: Baseline control input(Red line) and adaptive control input(Black line)

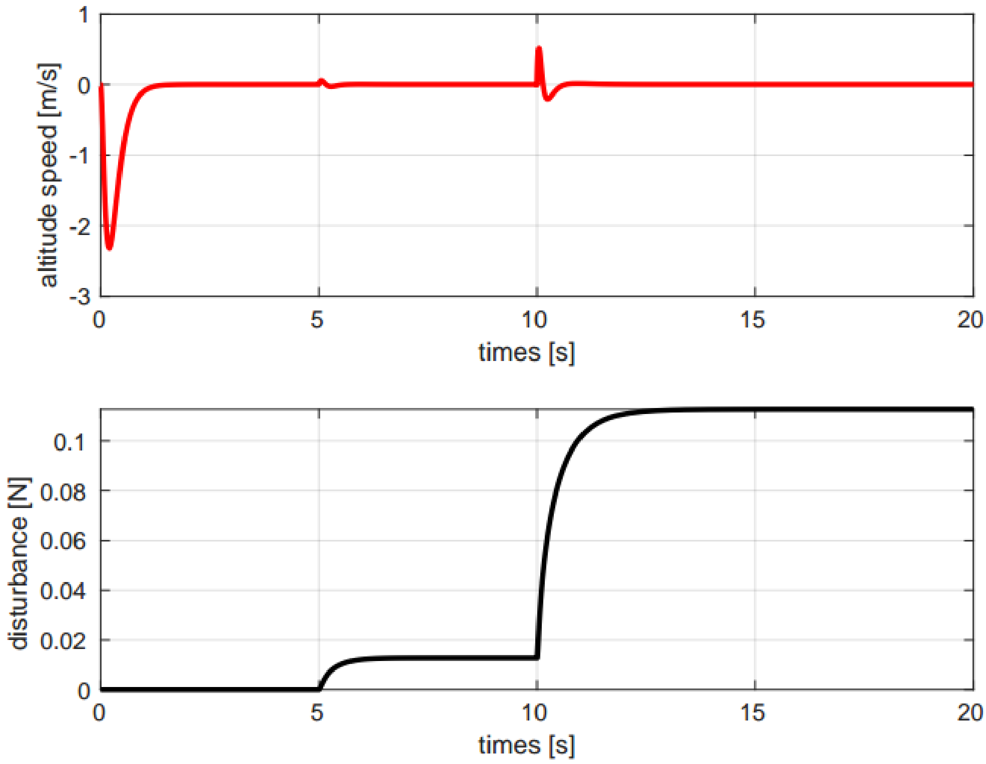


Figure 4.6: altitude speed (Red line) and disturbance estimation (Black line)

4.2 Simulator

The simulator (Figure 4.7), was designed and implemented by some student of politecnico di milano. The laboratory FlyART of politecnico di milano, recommends to all its students, to use that simulator before conducting real tests on the real quadcopter. The simulator is subdivided in three main blocks: the quadrotor block, the control laws block and finally the State filter block. As input, the simulator takes the set point and gives, in output, the state of the quadcopter.

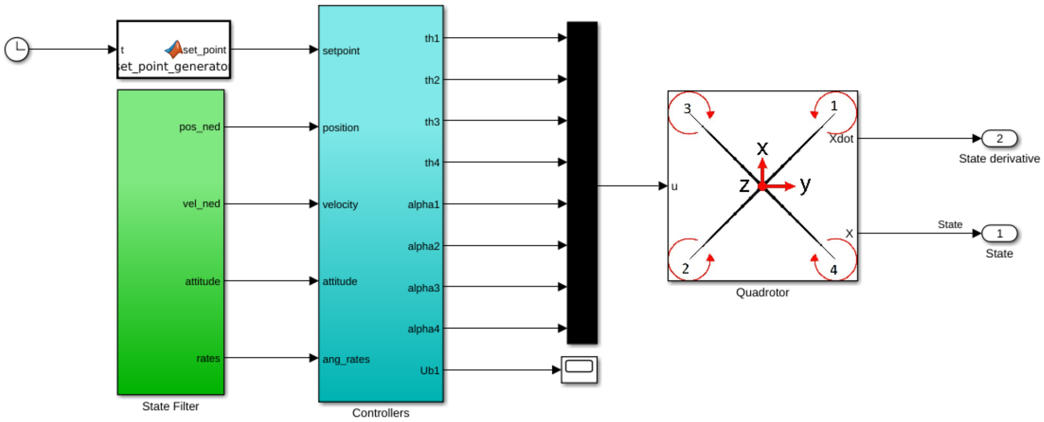


Figure 4.7: Simulator

After designing the control law (see Chapter 3) , I implemented it on MATLAB/Simulink and added it to the simulator inside the control laws block before running the simulation.

4.3 Altitude simulation

This part(Figure 4.8) presents the control algorithm written using MATLAB/Simulink. The block is also subdivided in three parts: the Luenberger observer, the update laws and finally the adaptive control input:

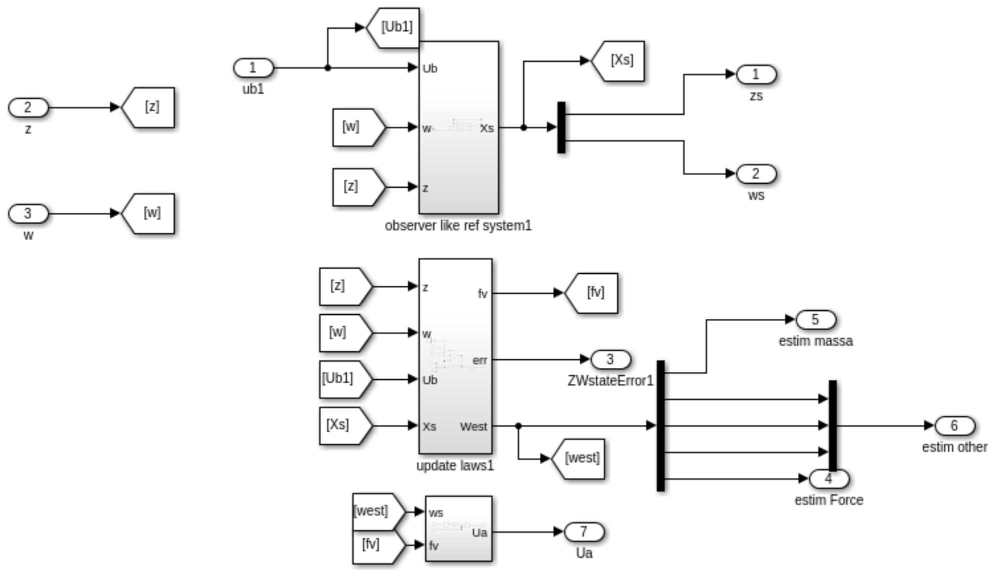


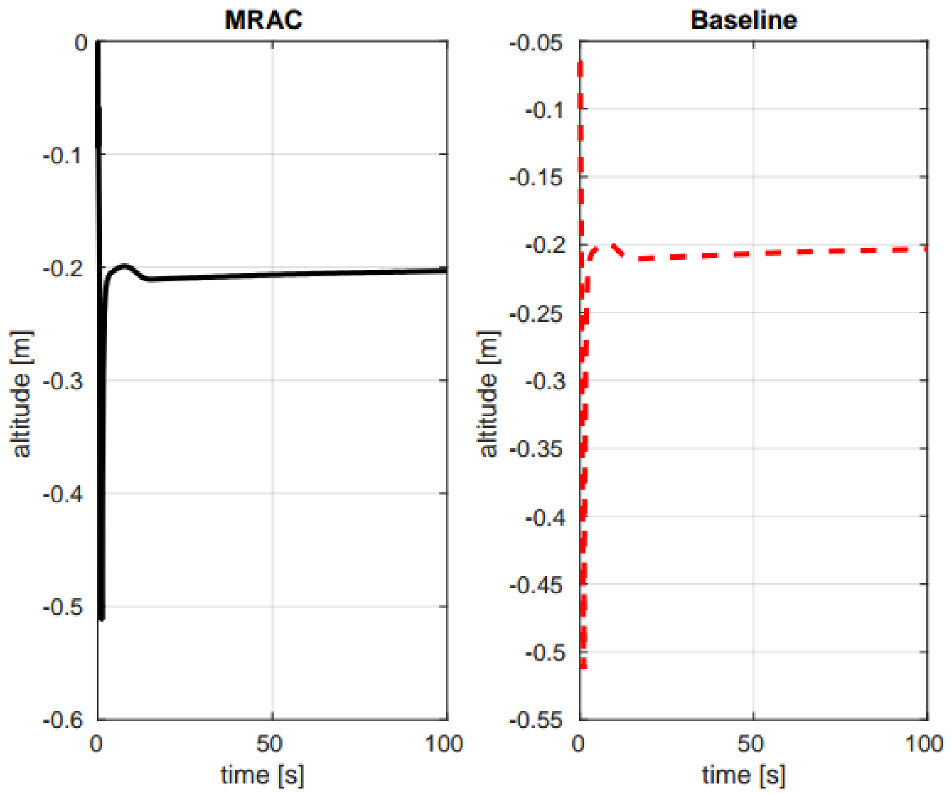
Figure 4.8: Control input block

z and w are respectivelly the altitude and altitude speed of the quadcopter.The output is the control adaptive input u_a.

4.4 Results

From Figure 4.9 we notice that the altitude response from the baseline input and the response from the total control input(adding the adaptive control input to the baseline) are quite the same after stabilization. The only difference occurs during the transient where we can clearly see that the MRAC-OBA need small time to deal with transient behaviour. The fact that applying the baseline control to the reference system and applying the proposed MRAC input to the uncertain system, both the response are the same after some time, allows us to confirm that the MRAC effectively forced the system to behave like the reference model as we wanted to achieve.

altitude:


MRAC
0
-0.1
-0.2
-0.3
-0.4
-0.5
-0.6
altitude [m]
0
50
100
time [s]
Baseline
-0.05
-0.1
-0.15
-0.2
-0.25
-0.3
-0.35
-0.4
-0.45
-0.5
-0.55
altitude [m]
0
50
100
time [s]

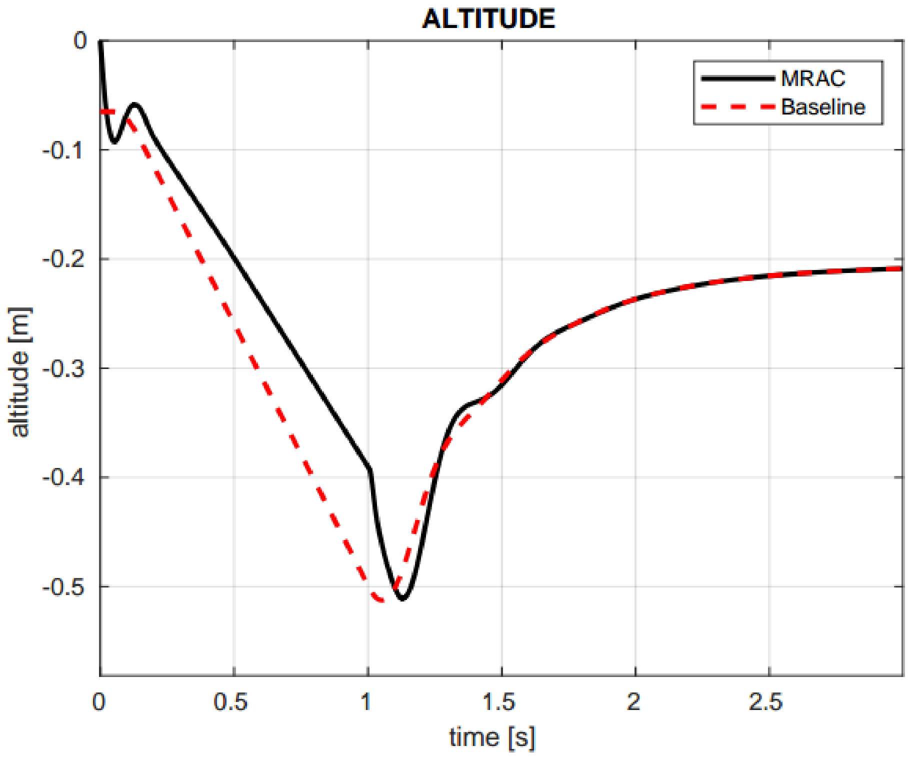
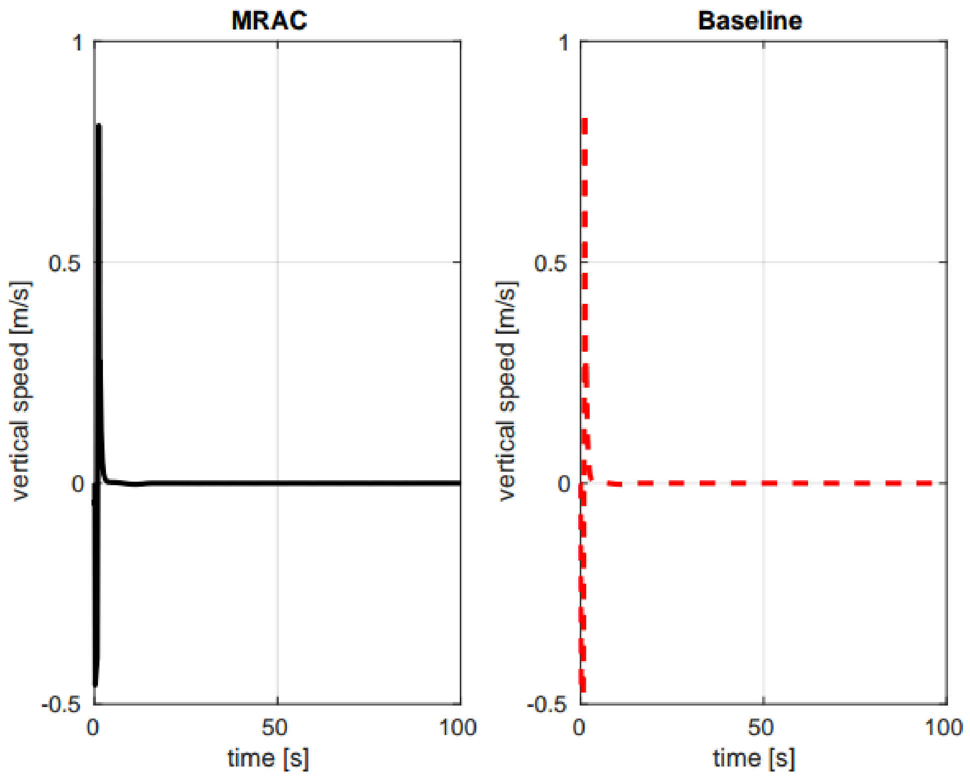


Figure 4.9: Altitude

altitude speed:



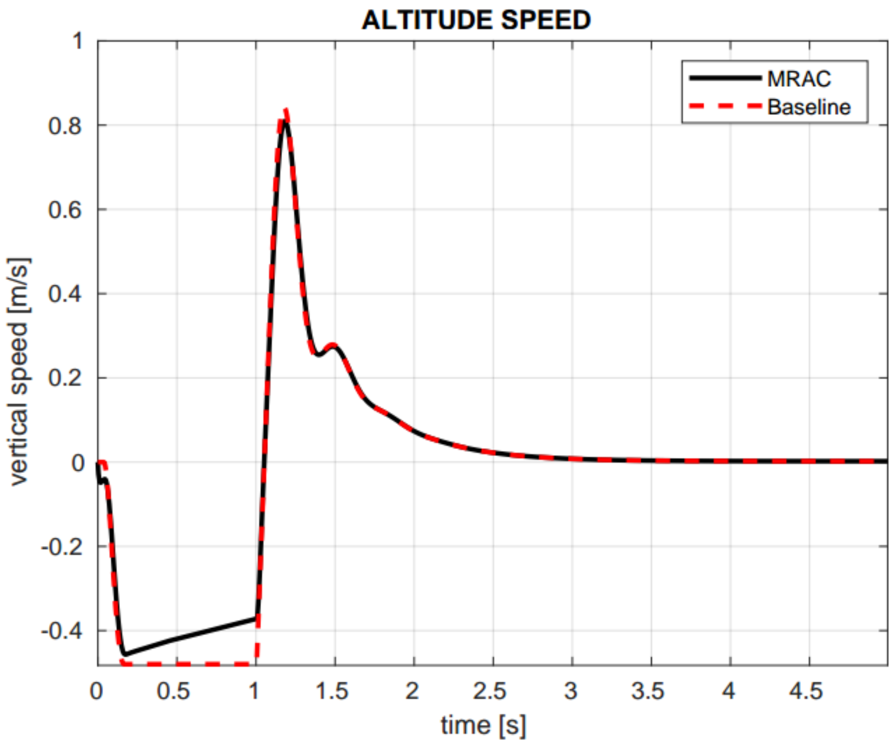


Figure 4.10: Altitude speed

error:

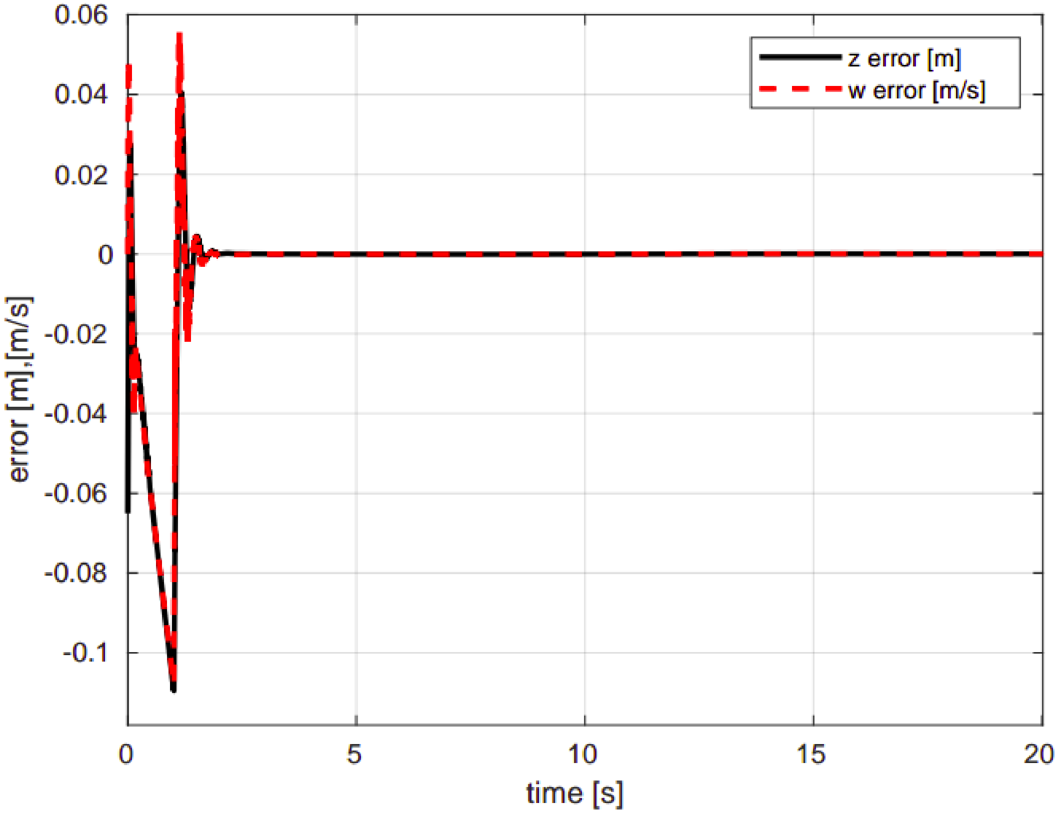


Figure 4.11: State estimation errors

disturbances estimated:

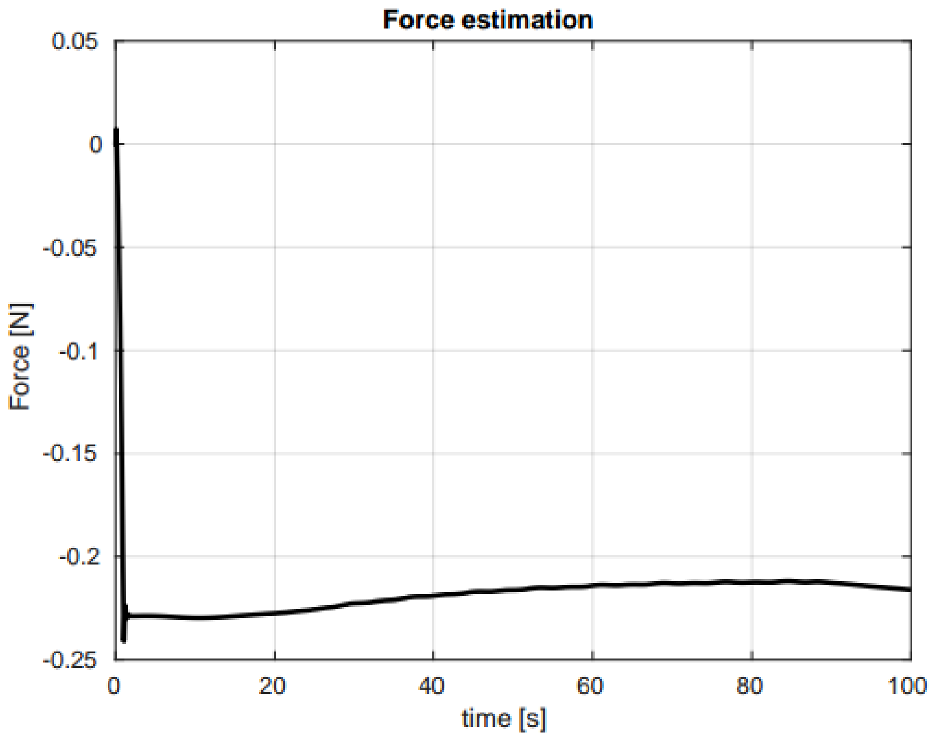


Figure 4.12: disturbance

Appendix A

First appendix

[1] Z. T. Dydek, A. M. Annaswamy, and E. Lavretsky, "Adaptive control of quadrotor uavs: A design trade study with flight evaluations," Control Systems Technology, IEEE Transactions on, vol. 21, no. 4, pp. 1400–1406, 2013.

[2]G. Bressan1,A. Russo1,D. Invernizzi1,M. Giurato1,S. Panza1,and M. Lovera1 "Adaptive augmentation of the attitude control system for a multirotor UAV".

[3] T. Yucelen and A. J. Calise, "Derivative-free model reference adaptive control," Journal of Guidance, Control, and Dynamics, vol. 34, no. 4, pp. 933–950, 2011.

[4] H. P. Whitaker, J. Yamron, and A. Kezer, Design of model-reference adaptive control systems for aircraft. Massachusetts Institute of Technology, Instrumentation Laboratory, 1958.

[5] B. Anderson, "Adaptive systems, lack of persistency of excitation and bursting phenomena," Automatica, vol. 21, no. 3, pp. 247–258, 1985.