Université Claude Bernard Lyon 1

Department of Mechanics

Automated Systems Engineering

Master Internship Report

Modeling And Control Of Drone Flight Subject To Turbulence

Author:

Riyadh Abbes Belbecir

Supervisors:

Madiha Nadri
Laurent Bako

Internship Laboratory :

LAGEPP

September 29, 2021

Contents

Chapter 1

Introduction

Unmanned aerial vehicle (UAV) nowadays are one of robots that witness a huge advancement in both military and civil domains since it presents different properties such as vertical flying, autonomous motions, remote control and many others that is hard to find in other robots together. Those properties could take place thanks to the recent advances in many scientific disciplines such as control and aerodynamics theory, computers and sensors technologies.

However, UAV's are still facing a lot of technical challenges due to environments with unpredictable changes, sensors noisy data, power supplying, mathematical nonlinear behaviour, so it makes UAV's a fertile field for researchers to develop much reliable technologies to overcome those challenges and benefit from the advantages presented by those systems.

1.1 Context

The present work is part of an ongoing research project named "Delicio" (2019 - 2023), based on collaboration between 4 Laboratories : LAGEPP, LIRIS, CITI and ONERA. It proposes fundamental researches in the areas of Machine Learning/IA and process control with applications to drone (UAV) fleet control.

The general aim of the project is to address learning stable strategies which permit drones to achieve high-level missions (navigation, visual search) while keeping stable flight trajectories. This lead to combine two different fields: the first one is control theory which is used to design a basic controller based on a model obtained by physical laws that contains uncertainties, this controller can be augmented by the machine learning field techniques with a performance-enhancing component which may be based on the online estimation of the initially unmodelled dynamics and of external signals. The work-packages are shown in Figure 1.1 and Figure 1.2:

Present Work Objective

In this report an UAV type quadcopter is studied. It has been chosen for the previous reasons and also for their simple mechanical structure, adaptability to different missions and tasks.

Our approach aims to present a mathematical model of a quadcopter based on physical laws which is considered

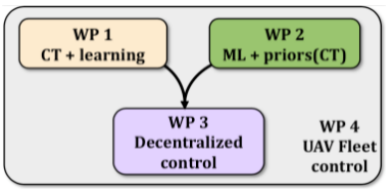


Figure 1.1: Delicio Project Work Packages

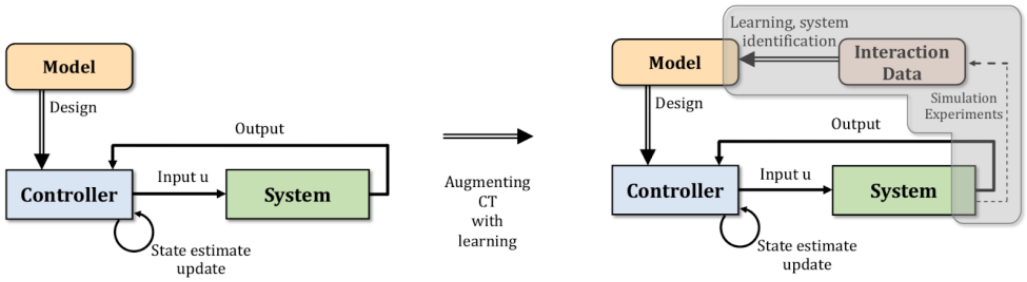


Figure 1.2: First Work Package

ideal in the first place and choose a set of controllers that exist in the literature to ensure a desired performance. The second part is to introduce an unmodelled disturbance type ground effect in order to verify how far those controllers can be robust and do we need to estimate and measure those disturbances to improve those controllers. The results obtained will contribute in developing the first package in "Delicio" Project (Figure 1.2).

1.2 Literature Review

In the past twenty years, the technological development specially inertial sensors (MEMS - Micro Electronic Mechanical System) allowed different laboratories to build different quadcopters prototypes.

The control of quadcopters becomes quickly a popular research subject because of its low cost and simple structure with the maintain of theoretical and practical challenges that manifest in the complex non linear dynamics, under-actuated and coupled nature. This raising popularity makes the literature wide and diverse. The first objective in studying these vehicles was the design of controllers able to stabilize their attitude in fast and robust ways. Today challenges are trajectory control, fault tolerance, path planning and obstacles avoidance ([1],[2],[3]).

By looking at the literature, control techniques applied on quadcopters can be categorised into four classes:

- The classical linear control techniques such as PID ([4],[5],[6]), LQR ([7],[8]),H_∞ ([9]) applied by linearizing the system around an operating point, generally chosen to be the hover.
- The non linear techniques that provide larger domain of attraction and better performances. We can cite Backstepping ([10],[11]), Sliding Mode ([12]), Feedback Linearization ([13],[14]), and model predictive control ([15],[16]).
- The learning-Based techniques, which differ from the two previous controls by the fact that they don't need

an accurate model of the system, but a several experiments took place and flight data are utilised to train the system. From those techniques, we can mention Fuzzy Logic and Neural Networks ([17],[18]).

- Recently, Hybrid approaches are being used in order to guarantee better performances, so a combination of different previous techniques are used to control different states of the system ([19],[20]).

For this work, we investigate two approaches: (*i*) The Feedback Linearization because it provides access to all linear techniques applicable in certain region of the state space which are easier to analyse and to implement compared with other non linear techniques. Furthermore, this technique has a simple control structure and the facility of formal proofs of error convergence when appropriate conditions are met [1].

(*ii*) The Backstepping technique considered as a modern non linear control design method. Known for good performances for tracking problems and well adapted to the under-actuated nature of quadcopters. It provides a larger region of attraction than other controllers with good robustness against disturbances.

1.3 Outline

The rest of this report is organized as follows:

- **Chapter 2.** The primary mathematical tools needed to describe the motion of quadcopters are presented in this chapter such as Euler's angles and Euler-Newton formalism. Then a mathematical model is extracted.
- **Chapter 3.** This chapter contains two main sections that discuss the control design approaches in the absence of disturbances. The first one is the Feedback Linearization. The second is the Backstepping. Both sections are organized in the same way. So they explain the theoretical aspect of the technique then an application of that technique in the case of quadcopters and finally, a simulations obtained by Matlab/Simulink will show the behaviour of quadcopter under each technique.
- **Chapter 4.** Here, the ground effect and gyroscopic effect disturbances are modeled and added to the quadcopter model in order to verify the robustness of each controller. Then, the simulations and results are compared and discussed.
- **Chapter 5.** The final chapter present a conclusion and the possible future works linked to this one.

Chapter 2

The Quadcopter Model

Quadcopters are flying robots easy to understand its general structure and how it works in an intuitive way, since it needs only an X shape frame structure that contains four motors each one in a corner. Those motors hold propellers with specific aerodynamic properties that makes it able to generates the thrust force. And controlled by an electronic system placed generally in the frame's center that contains control processing unit, different sensors and power unit.

However, the quadcopters are a complex systems with regard to the theoretical aspect and dynamic since it has an under-actuated nature (control inputs less than the degree of freedom) and many physical phenomena that affect the dynamic. This complexity makes getting an accurate mathematical model able to predict in the best way possible the real dynamics a challenging task.

Indeed, a compromise must be established between the complexity of the model and the desired performances by adopting simplifying assumptions in order to overcome practical constraints.

The model presented in this work is derived using Newton-Euler Formalism. Under the following classical assumptions:

- The body is rigid and symmetrical.
- The quadcopter's center of gravity matches the body frame origin.
- The propellers are rigid that the blades flapping effect is neglected.
- The quadcopter operates near the hover so a small angles variations are accepted.

In next sections, we will introduce the model by establishing its kinematics and dynamics.

2.1 Kinematics

2.1.1 Frames

As any 3D motion study, references need to be defined. In the case of quadcopters, at least two reference systems are required [21]. The first one is the fixed frame named O_{NED}, linked to the earth's surface considered inertial and taken with NED convention, which means that its axes X_n Y_n Z_n are oriented to true (geographic) North East Down respectively as shown in Figure 2.1 and the origin O_n is determined using latitude and longitude.

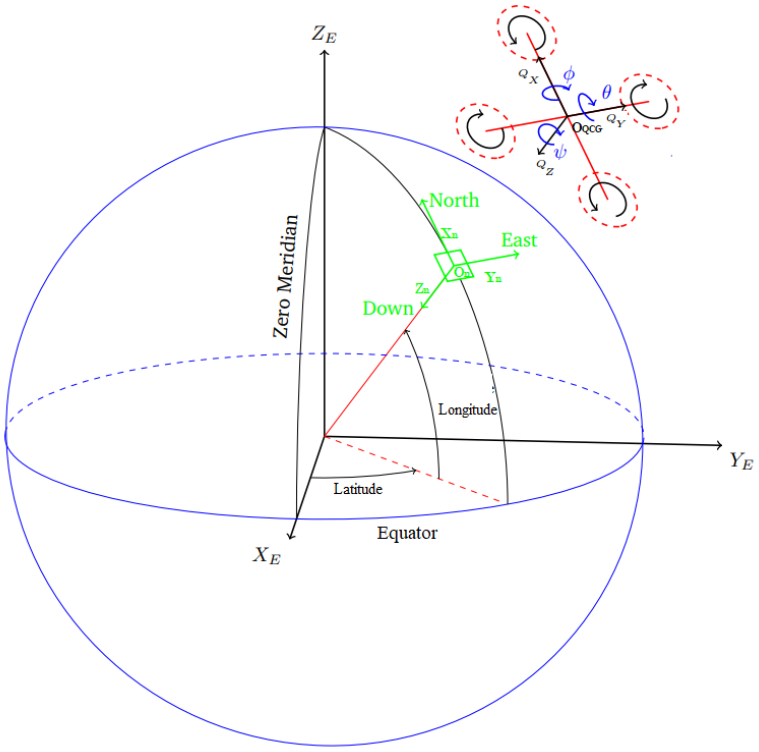


Figure 2.1: O_{NED} fixed frame shown in Earth-centered Earth-fixed (ECEF) frame [22]

The second frame is the mobile one linked to the body of quadcopter. Its origin O_{QCG}[1] is at the center of gravity of the vehicle and the three axes are oriented as shown in Figure 2.2.

Now, with the frames defined, we can express the translation motion using distance between the fixed frame origin O_n and the quadcopter's frame origin O_{QCG}. By projecting this latter on the axis X_n Y_n Z_n, we get the absolute QCG position with respect to NED frame $\eta = [x \quad y \quad z]^T$.

It remains to express the rotation of quadcopter frame in the fixed frame. In geometry, there exist different formalisms to describe it in three dimensions such as quaternions, Rodrigues formula ([23],[24])...etc. In this work the orientation is described using Euler's angles known as roll (ϕ), pitch(θ) and yaw(ψ) angles representing rotations about Q_x, Q_y and Q_z respectively.

[1] Quadcopter's center of gravity

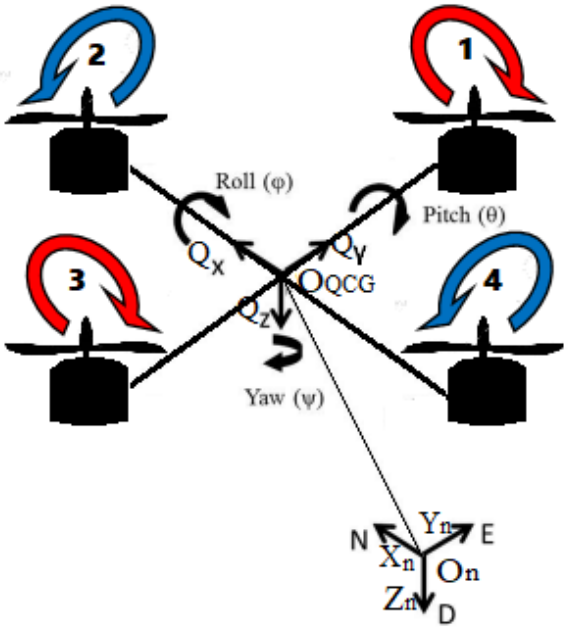


Figure 2.2: The quadcopter frame and the fixed frame

2.1.2 Euler's Angles

We can define Euler's angles as three angular parameters that specify the orientation of a body with respect to reference axes. They are the minimal representation possible to describe orientation with three independent parameters. Since the special orthogonal group also known as rotation group "SO(m)" requires m(m-1)/2 parameters, we need 3 angles minimum to parameterize SO(3) [25].

Notation. *The aeronautical convention ZYX angles known as Roll-Pitch-Yaw angles is used.*

The rotation resulting from Roll–Pitch–Yaw angles can be obtained as follows:

- Rotate the NED frame by the angle ϕ about axis X_n (yaw); this rotation is described by the matrix $R_{Xn}(\phi)$ which is formally defined in (2.1).
- Rotate the NED frame by the angle θ about axis Y_n (pitch); this rotation is described by the matrix $R_{Yn}(\theta)$ which is formally defined in (2.2).
- Rotate the NED frame by the angle ψ about axis Z_n (roll); this rotation is described by the matrix $R_{Zn}(\psi)$ which is formally defined in (2.3).

$$R_{X_n}(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\phi) & -s(\phi) \\ 0 & s(\phi) & c(\phi) \end{bmatrix} \tag{2.1}$$

$$R_{Y_n}(\theta) = \begin{bmatrix} c(\theta) & 0 & s(\theta) \\ 0 & 1 & 0 \\ -s(\theta) & 0 & c(\theta) \end{bmatrix} \tag{2.2}$$

$$R_{Z_n}(\psi) = \begin{bmatrix} c(\psi) & -s(\psi) & 0 \\ s(\psi) & c(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.3}$$

Notation. *$c(\phi) = cos(\phi)$, $s(\phi) = sin(\phi)$, $c(\theta) = cos(\theta)$, $s(\theta) = sin(\theta)$, $c(\psi) = cos(\psi)$, $s(\psi) = sin(\psi)$. This notation is kept throughout all this work.*

Finally and by multiplying the three elementary rotation matrices, we get in (2.4) the quadcopter orientation composed of rotations $\begin{pmatrix} \phi & \theta & \psi \end{pmatrix}$ with respect to the NED fixed frame:

$$R = R_{Z_n Y_n X_n}\begin{pmatrix} \phi & \theta & \psi \end{pmatrix} = R_{z_n}(\psi) R_{Y_n}(\theta) R_{X_n}(\phi)$$

$$R = \begin{bmatrix} c(\theta)c(\psi) & s(\phi)s(\theta)c(\psi) - c(\phi)s(\psi) & c(\phi)s(\theta)c(\psi) + s(\phi)s(\psi) \\ c(\theta)s(\psi) & s(\phi)s(\theta)s(\psi) + c(\phi)c(\psi) & c(\phi)s(\theta)s(\psi) - s(\phi)c(\psi) \\ -s(\theta) & s(\phi)c(\theta) & c(\phi)c(\theta) \end{bmatrix}. \tag{2.4}$$

In order to make the transformations between the mobile frame and the fixed frame linear velocities, we use the matrix R. But also we need to define a second transformation matrix for the angular velocities named T. Let us call $\begin{bmatrix} \eta & \Psi \end{bmatrix}^T = \begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^T$ the vector containing the linear and angular position of the quadcopter in the fixed frame and $\begin{bmatrix} V & \Omega \end{bmatrix}^T = \begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^T$ the vector containing the linear and angular velocities in the mobile frame, those vectors are linked by the following relation [26]:

$$\begin{bmatrix} \dot{\eta} \\ \dot{\Psi} \end{bmatrix} = \begin{bmatrix} R & 0_{3\times 3} \\ 0_{3\times 3} & T \end{bmatrix} \begin{bmatrix} V \\ \Omega \end{bmatrix}, \tag{2.5}$$

where T is the angular velocity transformation matrix given by:

$$T = \begin{bmatrix} 1 & s(\phi)t(\theta) & c(\theta)t(\theta) \\ 0 & c(\phi) & -s(\phi) \\ 0 & \frac{s(\phi)}{c(\theta)} & \frac{c(\phi)}{c(\theta)} \end{bmatrix}. \tag{2.6}$$

2.2 Dynamic Model

2.2.1 Euler-Newton Formalism

We utilize Newton's laws in the fixed frame (considered inertial) to get the translation motion formula generated by the total forces exerting on quadcopter.

From newton's law we have:

$$m\ddot{\eta} = R\sum F_{ext_{QCG}} + \sum F_{ext_{NED}} = -f_t R Q_z + mgZ_n + f_w, \tag{2.7}$$

where m is the mass of the quadcopter. $\ddot{\eta}$ is the acceleration of quadcopter in the fixed frame. g is the gravitational acceleration. $\sum F_{ext_{QCG}}$ is the sum of external forces applied on quadcopter in the mobile frame. $\sum F_{ext_{NED}}$ is the sum of external forces applied on quadcopter in the fixed frame. f_t is the total thrust generated by rotors. $Z_n = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^T \in \mathbb{R}^3$ is the unit vector of the fixed frame Z axis. $Q_z = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^T \in \mathbb{R}^3$ is the unit vector of the mobile frame Z axis. $f_w = \begin{bmatrix}f_{wx} & f_{wy} & f_{wz}\end{bmatrix}^T \in \mathbb{R}^3$ are the disturbances effects expressed in the inertial frame.

By developing (2.7), we get:

$$\begin{cases} m\ddot{x} = -(c(\phi)s(\theta)c(\psi) + s(\phi)s(\psi))f_t + f_{wx} \\ m\ddot{y} = -(c(\phi)s(\theta)s(\psi) - s(\phi)c(\psi))f_t + f_{wy} \\ m\ddot{z} = mg - (c(\phi)c(\theta))f_t + f_{wz}. \end{cases} \tag{2.8}$$

Now for the torques, we use Euler's equation in the mobile frame given by :

$$I\dot{\Omega} + \Omega \times (I\Omega) = M, \tag{2.9}$$

where $M = \begin{bmatrix}m_x & m_y & m_z\end{bmatrix}^T \in \mathbb{R}^3$ is the total torque applied on the quadcopter in the mobile frame. $I \in \mathbb{R}^{3\times 3}$ is the the diagonal inertia matrix (it is so because the assumption of perfectly symmetrical structure) defined as:

$$I = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix}, \tag{2.10}$$

where I_x, I_y , I_z are the inertia on mobile axis Q_x, Q_y and Q_z respectively.

The total torque is given by:

$$M = \tau - \tau_g + \tau_{w_Q}, \tag{2.11}$$

where τ_g is the gyroscopic moment caused by the rotation of motors and the mass which is considered as unmodeled disturbance so we neglect it in the first model. $\tau = \begin{bmatrix}\tau_x & \tau_y & \tau_z\end{bmatrix}^T \in \mathbb{R}^3$ are the control torques generated by rotors. $\tau_{w_Q} = \begin{bmatrix}\tau_{wx} & \tau_{wy} & \tau_{wz}\end{bmatrix}^T \in \mathbb{R}^3$ are the disturbances torques expressed in the body frame.

From (2.9), we get:

$$\begin{cases} m_x = \tau_x + \tau_{wx} = I_x\dot{p} + qr(I_z - I_y) \\ m_y = \tau_y + \tau_{wy} = I_y\dot{q} + pr(I_x - I_z) \\ m_z = \tau_z + \tau_{wz} = I_z\dot{r} + pq(I_y - I_x). \end{cases} \tag{2.12}$$

2.2.2 Motors Dynamics

The motion of quadcopters is based on the change in rotation speed of propellers that generates the force and torques. The matrix that links the force and torques with propellers velocities is given by [27]:

$$\begin{bmatrix} f_t \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} +b & +b & +b & +b \\ -bl & 0 & +bl & 0 \\ 0 & -bl & 0 & +bl \\ -d & +d & -d & +d \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix}, \tag{2.13}$$

where $\begin{bmatrix} \omega_1 & \omega_2 & \omega_3 & \omega_4 \end{bmatrix}^T \in \mathbb{R}^4$ are the velocities of each propeller (Figure 2.2). b is the thrust factor of rotor. d is the drag factor of rotor. l is the distance between the rotor center and the quadcopter center.

We note that the thrust generated by each rotor is given by:

$$f_{R_i} = b.\omega_i^2 \qquad\qquad i = \{1, 2, 3, 4\}. \tag{2.14}$$

2.3 State Space Model

Using equations (2.8) and (2.12), we obtain the linear accelerations in the fixed frame and the rotational accelerations in the mobile frame:

$$\begin{cases} \ddot{x} = -\frac{1}{m}(c(\phi)s(\theta)c(\psi) + s(\phi)s(\psi))f_t + \frac{1}{m}f_{wx} \\ \ddot{y} = -\frac{1}{m}(c(\phi)s(\theta)s(\psi) - s(\phi)c(\psi))f_t + \frac{1}{m}f_{wy} \\ \ddot{z} = g - \frac{1}{m}(c(\phi)c(\theta))f_t + \frac{1}{m}f_{wz} \\ \dot{p} = \frac{(I_y - I_z)}{I_x}qr + \frac{\tau_x + \tau_{wx}}{I_x} \\ \dot{q} = \frac{(I_z - I_x)}{I_y}pr + \frac{\tau_y + \tau_{wy}}{I_y} \\ \dot{r} = \frac{(I_x - I_y)}{I_z}pq + \frac{\tau_z + \tau_{wz}}{I_z}. \end{cases} \tag{2.15}$$

In the hypothesis, we assumed that the quadcopter operates near the hover. We can use the small angle approximation that states: for a small angle displacement expressed in radians, its sine and tangent are approximately equal to the angle and its cosine is approximately equal to 1 can be applied on our model.

This simplification is set by considering $sin(\phi) \approx 0$, $sin(\theta) \approx 0$, $cos(\phi) \approx 1$ and $cos(\theta) \approx 1$. The matrix T in (2.6) becomes the identity matrix which leads to the equality between the angular velocities in the mobile frame and in the fixed frame $\begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T = \begin{bmatrix} p & q & r \end{bmatrix}^T$.

The dynamic model of the quadcopter in the inertial frame becomes:

$$\begin{cases} \ddot{x} = -(s(\phi)s(\psi) + c(\phi)c(\psi)s(\theta))\frac{f_t}{m} + \frac{f_{wx}}{m} \\ \ddot{y} = -(c(\phi)s(\psi)s(\theta) - c(\psi)s(\phi))\frac{f_t}{m} + \frac{f_{wy}}{m} \\ \ddot{z} = g - (c(\phi)c(\theta))\frac{f_t}{m} + \frac{f_{wz}}{m} \\ \ddot{\phi} = \frac{I_y - I_z}{I_x}\dot{\theta}\dot{\psi} + \frac{\tau_x + \tau_{wx}}{I_x} \\ \ddot{\theta} = \frac{I_z - I_x}{I_y}\dot{\phi}\dot{\psi} + \frac{\tau_y + \tau_{wy}}{I_y} \\ \ddot{\psi} = \frac{I_x - I_y}{I_z}\dot{\phi}\dot{\theta} + \frac{\tau_z + \tau_{wz}}{I_z}. \end{cases} \tag{2.16}$$

By taking the states vector $x \in \mathbb{R}^{12}$ such that:

$$\begin{aligned} \mathbf{x} &= [\ x \quad y \quad z \quad \psi \quad \theta \quad \phi \quad \dot{x} \quad \dot{y} \quad \dot{z} \quad \dot{\phi} \quad \dot{\theta} \quad \dot{\psi}\ ]^T \\ &= [\ x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6 \quad x_7 \quad x_8 \quad x_9 \quad x_{10} \quad x_{11} \quad x_{12}\ ]^T \end{aligned} \tag{2.17}$$

and

$$\begin{bmatrix} f_t \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \qquad \begin{bmatrix} f_{wx} \\ f_{wy} \\ f_{wz} \\ \tau_{wx} \\ \tau_{wy} \\ \tau_{wz} \end{bmatrix} = \begin{bmatrix} d_x \\ d_y \\ d_z \\ d_\phi \\ d_\theta \\ d_\psi \end{bmatrix} \tag{2.18}$$

The state space representation becomes:

$$\begin{cases} \dot{x}_1 &= x_7 \\ \dot{x}_2 &= x_8 \\ \dot{x}_3 &= x_9 \\ \dot{x}_4 &= x_{12} \\ \dot{x}_5 &= x_{11} \\ \dot{x}_6 &= x_{10} \\ \dot{x}_7 &= -\frac{1}{m}(c(x_6)c(x_4)s(x_5) + s(x_6)s(x_4))u_1 + \frac{1}{m}d_x \\ \dot{x}_8 &= -\frac{1}{m}(c(x_6)s(x_4)s(x_5) - c(x_4)s(x_6))u_1 + \frac{1}{m}d_y \\ \dot{x}_9 &= g - \frac{1}{m}(c(x_6)c(x_5))u_1 + \frac{1}{m}d_z \\ \dot{x}_{10} &= \frac{I_y - I_z}{I_x}x_{11}x_{12} + \frac{1}{I_x}u_2 + \frac{1}{I_x}d_\phi \\ \dot{x}_{11} &= \frac{I_z - I_x}{I_y}x_{10}x_{12} + \frac{1}{I_y}u_3 + \frac{1}{I_y}d_\theta \\ \dot{x}_{12} &= \frac{I_x - I_y}{I_z}x_{10}x_{11} + \frac{1}{I_z}u_4 + \frac{1}{I_z}d_\psi. \end{cases} \tag{2.19}$$

2.3.1 Open Loop Simulation

In order to verify the behaviour of the model (2.19), an open loop simulation is made by MATLAB/Simulink codes. By giving forces and torques as inputs and cancelling disturbances, we can verify if an expected behaviour is followed.

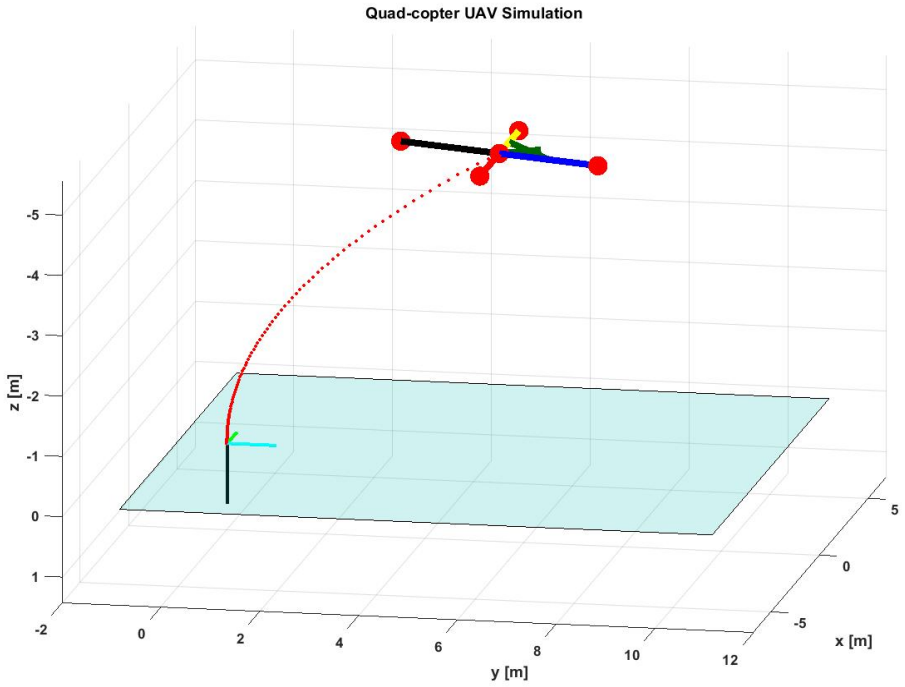
The parameters of the quadcopter used for this simulation and all the next ones are taken from [28] that belong to the OS4 UAV and they are listed in Table 2.1.

Parameter	Notation	Value	Unit
gravitational acceleration	g	9.81	$m.s^{-2}$
mass	m	7.5e-3	$kg.m^2$
inertia on x axis	I_x	7.5e-3	$kg.m^2$
inertia on y axis	I_y	7.5e-3	$kg.m^2$
inertia on z axis	I_z	1.3e-2	$kg.m^2$
rotor inertia	J_p	6e-5	$kg.m^2$
thrust coefficient	b	3.13e-5	$N.s^2$
drag coefficient	d	7.5e-7	$N.m.s^2$
rotor radius	R_{ad}	0.15	m
arm length	l	0.23	m

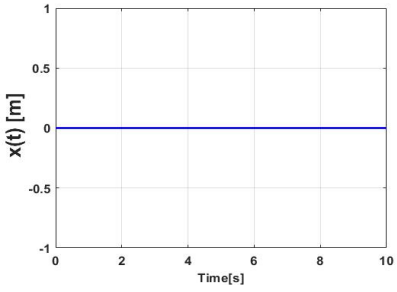
Table 2.1: Simulations Parameters based on OS4

In absence of disturbances and if the initial state is equal to zero, the magnitude of u_1 must be equal to $(\frac{m}{c(x_6)c(x_5)}(g+\delta u_1))$ with $\delta u_1 = 0$ (δu_1 is a term that control the altitude freely when all other terms are cancelled) so the quadcopter can hover. We want the quadcopter to go up slowly with a slight rotation on x-axis (rotation on x-axis generates translation on y-axis). So we give $\delta u_1 = 0.1N$ and the torque u_2 =1e-5 $N.m$. The rest of states are kept null.

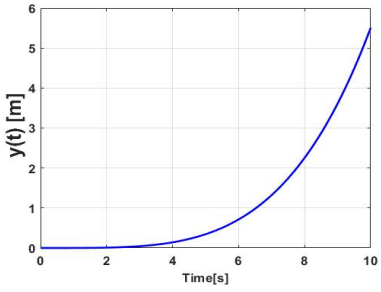
The results are shown in Figure 2.3 and they reflect the correct expectations from our model.



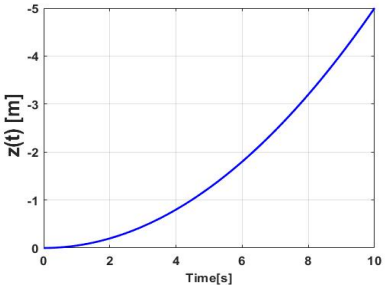
(a) The trajectory in 3D



(b) x position [m]



(c) y position [m]



(d) z position [m]

Figure 2.3: Open loop simulation results

Chapter 3

Control Techniques

This section will discuss the design of controllers using the Feedback Linearization technique and the Backstepping technique. For both strategies, a general theoretical presentation is introduced then the application of those theories in the case of quadcopter model is developed. At the end, simulations using Matlab/Simulink will take place to illustrate and validate the results in the absence of disturbances.

3.1 Feedback Linearization Technique

The Feedback Linearization (FBL) is one of the most used techniques in control theory thanks to its simple structure. This technique aims to cancel the non linearities of models partially in a certain region of the state space or globally if it is possible by a nonlinear inversion so the linear techniques (PID, LQR, H_∞, ...) can be applied easily.

Contrary to tangent linearization technique where a local approximation of the dynamics is considered without feedback, the FBL is obtained by exact state transformation and feedback [29].

3.1.1 Feedback Linearization Concept

Before we start the concept development, some basic definitions in the FBL literature are needed.

We consider a class of affine nonlinear systems of the form:

$$\begin{cases} \dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u} \\ \mathbf{y} = h(\mathbf{x}), \end{cases} \tag{3.1}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state space vector. $\mathbf{u} \in \mathbb{R}^m$ is the control input vector. $\mathbf{y} \in \mathbb{R}^m$ is the output vector. $f(\mathbf{x}) \in \mathbb{R}^n$ is a nonlinear functions vector. $g(\mathbf{x}) \in \mathbb{R}^{n \times m}$ is a nonlinear functions matrix. $h(\mathbf{x}) \in \mathbb{R}^m$ is a nonlinear functions vector.

Definition 3.1.1 (Lie derivative [30]). *Let $f : \mathbb{R}^n \to \mathbb{R}^n$ be a vector field and $h : \mathbb{R}^n \to \mathbb{R}$ another vector field, the Lie derivative is the scalar function, denoted $L_f h$, giving the derivative of $h(\mathbf{x})$ in the direction $f(\mathbf{x})$, as follows:*

$$L_f h(\mathbf{x}) = \nabla h \times f = \langle \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}}, f(\mathbf{x}) \rangle = \begin{bmatrix} \frac{\partial h(\mathbf{x})}{\partial x_1} & \frac{\partial h(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial h(\mathbf{x})}{\partial x_n} \end{bmatrix} \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_n(\mathbf{x}) \end{bmatrix}. \tag{3.2}$$

the second order Lie derivative is given by:

$$L_f^2 h(\mathbf{x}) = \langle \frac{\partial L_f h(\mathbf{x})}{\partial \mathbf{x}}, f(\mathbf{x}) \rangle. \tag{3.3}$$

a generalisation for any order can be made, for $i \in \mathbb{N}^*$:

$$L_f^i h(\mathbf{x}) = \langle \frac{\partial L_f^{i-1} h(\mathbf{x})}{\partial \mathbf{x}}, f(\mathbf{x}) \rangle, \tag{3.4}$$

where $L_f^0 h(\mathbf{x}) = h(\mathbf{x})$.

Lie derivative can also be defined for multiple vector fields in the same way:

$$L_g L_f h(\mathbf{x}) = \langle \frac{\partial L_f h(\mathbf{x})}{\partial (\mathbf{x})}, g(\mathbf{x}) \rangle. \tag{3.5}$$

Definition 3.1.2 (Lie Bracket). *Let f and g two vector fields in $\mathbb{R}^n$. The Lie bracket of f and g is defined as:*

$$[f,g](\mathbf{x}) = \frac{\partial g}{\partial \mathbf{x}} f(\mathbf{x}) - \frac{\partial f}{\partial \mathbf{x}} g(\mathbf{x}) = ad_f g(\mathbf{x}), \tag{3.6}$$

where $\frac{\partial g}{\partial \mathbf{x}}$ and $\frac{\partial f}{\partial \mathbf{x}}$ are the Jacobian matrices of g and f respectively [30].

Notation. *$ad_f^0 g = g$, $ad_f^1 g = [f,g]$, $ad_f^2 g = [f,[f,g]]$, $ad_f^{i+1} g = [f,ad_f^i g]$ for $i \in \mathbb{N}$.*

Definition 3.1.3 (Diffeomorphism). *A function $\Phi : U \to V$ between two open subsets of $\mathbb{R}^n$ is called a **diffeomorphism** of U onto V if: (i) Φ is a bijection, (ii) Φ is differentiable, and (iii) Φ^{-1} is differentiable[31].*

Remark. *The diffeomorphism:*

$$z = \Phi(x) = \begin{bmatrix} \Phi_1(x) \\ \Phi_2(x) \\ \vdots \\ \Phi_n(x) \end{bmatrix} \tag{3.7}$$

is also known as a change of coordinates

Definition 3.1.4 (Relative Degree [32]). *The system (3.1) is said to have a relative degree r at a point $\mathbf{x}_0$ if:*

1. *$L_g L_f^k h(\mathbf{x}) = 0$ for all $\mathbf{x}$ in a neighbourhood of $\mathbf{x}_0$ for $k = 0,1,...,(r-2)$*
2. *$L_g L_f^{r-1} h(\mathbf{x}_0) \neq 0$*

Remark. 1. *The relative degree can be interpreted as how many times we need to derivate the output until the input appears explicitly in that derivative of output.*

2. *There may be points where a relative degree cannot be defined [32].*

In the case of MIMO systems with m outputs the relative degree becomes a vector $\mathbf{r} = \begin{bmatrix} r_1 & \cdots & r_m \end{bmatrix}^T$ instead of a scalar [31].

Lemma ([31]). *Suppose the system (3.1) has the relative degree vector* $\mathbf{r} = \begin{bmatrix} r_1 & \cdots & r_m \end{bmatrix}^T$. *Then:*

$$\sum_{i=1}^{m} r_i \leq n$$

The primary definitions having been set, the Feedback Linearization technique can be categorised into different problems according to the relative degree of the system and the number of inputs and outputs (MIMO Systems).Next, we present a shortcut list of interesting cases for this work:

- If $\mathbf{r} = n$ at all points $\mathbf{x}$, then the system is said to be fully feedback linearizable or exactly linearizable via Feedback.
- If $\mathbf{r} < n$, at all points $\mathbf{x}$ then the system is said to be partially state feedback linearizable [33].

Both cases require a static feedback control law (its formula will be presented next), it is called so because it is a memoryless function of $\mathbf{x}$. So at time t the feedback depends only on the values of the state $\mathbf{x}$ and of the external reference input at the same instant of time [32][30].

Another categorization can be made: either it is an input-state linearization which aims to linearize the map between the transformed inputs and the entire vector of transformed state variables, or input-output linearization that linearize the map between the transformed inputs and actual outputs [34], this process will result in a decoupled systems.

In some cases, when the exact linearization is not possible, one solution to this problem is by adding a dynamic compensator (or dynamic feedback). In this case, the control transformation is no longer static, the control is itself the output of an appropriate dynamical system having its own internal state driven by $\mathbf{x}$ and by the external reference input [32].

Remark. *By adding the dynamic compensator, the relative degree will be equal to the state space dimension "*$\mathbf{r} = n$*", so applying a static exact feedback linearization control will be possible.*

For the quadcopter model control an input output feedback linearization is developed with both dynamic compensator and a static feedback control as will be shown next.

3.1.1.1 Static Feedback Linearization

Since the linearization by a static feedback (S-FBL) is the basic technique, we need to present it first.

We begin with formulating the problem of exact feedback linearization when $(r = n)$ at the neighborhood of the origin for a SISO system. An extension for the MIMO systems will be presented later.

Remark. *The statements presented next are given without proofs. but references are mentioned.*

Consider a system described by:

$$\begin{cases} \dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u} \\ y = h(\mathbf{x}), \end{cases} \tag{3.8}$$

where f and g are smooth vector fields on some open subset $\mathbf{X} \subseteq \mathbb{R}^n$ containing 0, and $f(0) = 0$. $\mathbf{S}(\mathbf{X})$ is the set of all smooth real-valued functions mapping $\mathbf{X}$ into $\mathbb{R}$.

We are seeking to know when there exist smooth functions d_0, $d \in \mathbf{S}(\mathbf{x})$ with $d(\mathbf{x}) \neq 0$ for all $\mathbf{x}$ in some neighborhood of the origin, and a local diffeomorphism Φ on $\mathbb{R}^n$ with $\Phi(0) = 0$, such that if we define:

$$\mathbf{v} = d_0(\mathbf{x}) + d(\mathbf{x})\mathbf{u} \tag{3.9}$$

$$\mathbf{z} = \Phi(\mathbf{x}), \tag{3.10}$$

then the resulting variables $\mathbf{z}$ and $\mathbf{v}$ satisfy a linear differential equation of the form:

$$\dot{\mathbf{z}} = A\mathbf{z} + B\mathbf{v}. \tag{3.11}$$

where $\mathbf{v}$ is an external input. $\mathbf{z}$ is the new state variables in the new linear coordinates system and the pair (A, B) is controllable. If those conditions are met then the system (3.8) is said to be feedback linearizable.

From (3.9) and since $d(\mathbf{x}) \neq 0$ in some neighborhood of 0, we get:

$$\mathbf{u} = -\frac{d_0(\mathbf{x})}{d(\mathbf{x})} + \frac{1}{d(\mathbf{x})}\mathbf{v} = \alpha(\mathbf{x}) + \beta(\mathbf{x})\mathbf{v}, \tag{3.12}$$

where $\alpha(\mathbf{x})$ and $\beta(\mathbf{x})$ are also smooth functions.

The equation (3.12) is a well-known structure as regular static state feedback control law [35].

Definition 3.1.5. *Let f_1,f_2, $\cdots$, f_k be vector fields on $\mathcal{D} \subset \mathbb{R}^n$. At any fixed point $\mathbf{x} \in \mathcal{D}$, $f_1(\mathbf{x})$,$f_2(\mathbf{x})$,$\cdots$,$f_k(\mathbf{x})$ are vectors in R^n and $\Delta(\mathbf{x}) = span\{f_1(\mathbf{x}),f_2(\mathbf{x}),\cdots,f_k(\mathbf{x})\}$ is a subspace of R^n which we call a distribution.*

Two vector fields $g_1(\mathbf{x})$ and $g_2(\mathbf{x})$ are said to be involutive if $g_1(\mathbf{x}) \in \Delta(\mathbf{x})$ and $g_2(\mathbf{x}) \in \Delta(\mathbf{x}) \Rightarrow [g_1(\mathbf{x}),g_2(\mathbf{x})] \in \Delta(\mathbf{x})$ [30].

Theorem 3.1.1 ([31])**.** *The feedback linearization problem for the single input case has a solution if and only if the following two conditions are satisfied in some neighborhood of the origin:*

1. *The set of vector fields $\{ad_f^i g, 0 \leq i \leq n-1\}$ is linearly independent.*
2. *The set of vector fields $\{ad_f^i g, 0 \leq i \leq n-2\}$ is involutive.*

In the case where the previous two conditions are verified, the change of coordinates that construct the linear system is given by:

$$\mathbf{z} = \Phi(\mathbf{x}) = \begin{bmatrix} \Phi_1(\mathbf{x}) \\ \Phi_2(\mathbf{x}) \\ \vdots \\ \Phi_n(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} h(\mathbf{x}) \\ L_f h(\mathbf{x}) \\ \vdots \\ L_f^{n-1} h(\mathbf{x}) \end{bmatrix}. \tag{3.13}$$

And the resultant linear system is given by the Brunovsky's canonical form:

$$\begin{cases} \dot{z}_1 & = z_2 \\ \dot{z}_2 & = z_3 \\ & \vdots \\ \dot{z}_{n-1} & = z_n \\ \dot{z}_n & = d_0(\Phi^{-1}(\mathbf{z})) + d(\Phi^{-1}(\mathbf{z}))\mathbf{u}. \end{cases} \tag{3.14}$$

Finally, the control law expressed in terms of functions $f(\mathbf{x})$, $g(\mathbf{x})$, $h(\mathbf{x})$ that characterize the original non linear system is given by:

$$u = \frac{1}{L_g L_f^{n-1} h(\mathbf{x})}(-L_f^n h(\mathbf{x}) + v). \tag{3.15}$$

3.1.1.2 MIMO Systems Extension

We consider the system:

$$\begin{cases} \dot{\mathbf{x}} = f(\mathbf{x}) + \sum_{j=1}^{m} u_j g_j(\mathbf{x}) & u \in \mathbb{R}^m \\ y_i = h_i(\mathbf{x}) \qquad i = 1, \cdots, m & y \in \mathbb{R}^m \end{cases} \tag{3.16}$$

where $f, g_1, \cdots, g_m$ are vector fields on some open subset $\mathbf{X} \subseteq \mathbb{R}^n$ and $h_i \in \mathbf{S}(\mathbf{X})$ for all i are the outputs functions.

The system (3.16) is said to have the relative degree $\mathbf{r} = \begin{bmatrix} r_1 & r_2 & \cdots & r_m \end{bmatrix}^T$ if: (i) there exists a neighborhod $\mathbf{U}$ of 0 such that $(L_{g_j} L_f^k h_i)(\mathbf{x}) \equiv 0$, for $k = 0, \cdots, r_i - 2$, and (ii) the $m \times m$ matrix $\mathbf{D}$ defined by $\mathbf{D}_{ij} = L_{g_j} L_f^{r_i - 1} h_i$, for $i = 1, \cdots, m$, is nonsingular at $\mathbf{x} = 0$.

If this is the case, then the static state feedback control law becomes:

$$\mathbf{v} = \mathbf{D_0}(\mathbf{x}) + \mathbf{D}(\mathbf{x})\mathbf{u} \tag{3.17}$$

$$\mathbf{u} = \mathbf{D}(\mathbf{x})^{-1}(-\mathbf{D_0}(\mathbf{x}) + \mathbf{v}) \tag{3.18}$$

where the non-multiplicative vector $\mathbf{D_0}(\mathbf{x}) \in \mathbb{R}^m$ is given by [31]:

$$\mathbf{D_0}(\mathbf{x}) = \begin{bmatrix} L_f^{r_1} h_1(\mathbf{x}) & \dots & L_f^{r_m} h_m(\mathbf{x}) \end{bmatrix}^T \tag{3.19}$$

and the multiplicative matrix $\mathbf{D}(\mathbf{x}) \in \mathbb{R}^{n \times n}$ known as the decoupling matrix is given by:

$$\mathbf{D}(\mathbf{x}) = \begin{bmatrix} L_{g_1} L_f^{r_1 - 1} h_1(\mathbf{x}) & \cdots & L_{g_m} L_f^{r_1 - 1} h_1(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ L_{g_1} L_f^{r_m - 1} h_m(\mathbf{x}) & \cdots & L_{g_m} L_f^{r_m - 1} h_m(\mathbf{x}) \end{bmatrix} \tag{3.20}$$

The extension of (3.12) is made by the identification with (3.18):

$$\begin{cases} \alpha(\mathbf{x}) = -\mathbf{D}(\mathbf{x})^{-\mathbf{1}}\mathbf{D_0}(\mathbf{x}) \\ \beta(\mathbf{x}) = \mathbf{D}(\mathbf{x})^{-\mathbf{1}}. \end{cases} \tag{3.21}$$

And the resulting input-output relationship of the new linear system is described by [35]:

$$\frac{d^{r_i} y_i}{dt^i} = v_i \qquad\qquad i = 1, \cdots, m. \tag{3.22}$$

where the decoupled linear time-invariant system have the Brunovsky's canonical form and it is obtained by the same diffeomorphism (3.13).

We conclude that the feedback law (3.17) is expected to achieve two things: first, the input-output relationship is decoupled (y_i depends only on v_i). Second, the relationship between the i-th output and the i-th input is an integrator of order r_i [31].

3.1.1.3 Dynamic Feedback Linearization

As mentioned earlier, in case ($\mathbf{r} < n$) a state feedback is not possible by a static state feedback transformation because the relative degree is invariant under this type of feedback and the singularity of decoupling matrix cannot be removed [32]. So we design a dynamic state feedback that add new set of state variables and augment the main system.

The dynamic state feedback control law takes the form :

$$\begin{cases} \dot{\zeta} = \gamma(\zeta,\mathbf{x}) + \delta(\zeta,\mathbf{x})\mathbf{v} & \zeta \in \mathbb{R}^q \\ \mathbf{u} = \alpha(\zeta,\mathbf{x}) + \beta(\zeta,\mathbf{x})\mathbf{v}, & \mathbf{v} \in \mathbb{R}^m \end{cases} \tag{3.23}$$

where ζ is a vector containing the internal states, driven by the "inputs" $\mathbf{v}$ and $\mathbf{x}$.

This has the effect of augmenting the state space to:

$$\begin{cases} \begin{cases} \dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})(\alpha(\mathbf{x},\zeta) + \beta(\mathbf{x},\zeta)\mathbf{v}) \\ \dot{\zeta} = \gamma(\mathbf{x},\zeta) + \delta(\mathbf{x},\zeta)\mathbf{v} \end{cases} \\ \mathbf{y} = \mathbf{h}(\mathbf{x}). \end{cases} \tag{3.24}$$

Remark. *The diffeomorphism is also extended by $\bar{n} = n + q$, where $\bar{n}$ is the order of the new system :*

$$\mathbf{z} = \Phi(\mathbf{x}, \zeta). \tag{3.25}$$

We can note that the process of augmenting the state as described above doesn't affect the number of inputs and outputs. Thus, $\mathbf{r}$ and n will tend to approache the condition ($\mathbf{r} = n$) and we are able to manipulate the decoupling matrix so that it is no longer singular. The most known example of compensator construction is done by adding integrators which delay the effect of some input channels with respect to the others. By this addition, higher-order Lie derivatives can be reached and thereby increase the possibility of getting the decoupling matrix nonsingular. The application of the dynamic feedback will be shown on the quadcopter example.

3.1.2 The Quadcopter Feedback Linearization

As we mentioned before, the static approach is only applicable when the condition ($\mathbf{r} = n$) is met. In the quadcopter model (2.19) we can show that this condition is not fulfilled. We present the relationship between the inputs and the outputs $\mathbf{y} = \begin{bmatrix} x & y & z & \psi \end{bmatrix}^T = \begin{bmatrix} y_1 & y_2 & y_3 & y_4 \end{bmatrix}^T$. From (2.19):

$$\begin{cases} \ddot{y}_1 = -\frac{1}{m}(c(x_6)c(x_4)s(x_5) + s(x_6)s(x_4))u_1 \\ \ddot{y}_2 = -\frac{1}{m}(c(x_6)s(x_4)s(x_5) - c(x_4)s(x_6))u_1 \\ \ddot{y}_3 = g - \frac{1}{m}(c(x_6)c(x_5))u_1 \\ \ddot{y}_4 = \frac{I_x - I_y}{I_z}x_{10}x_{11} + \frac{1}{I_z}u_4. \end{cases} \tag{3.26}$$

we have $r_1 = r_2 = r_3 = r_4 = 2$ the relative degrees with respect to y_1, y_2, y_3 and y_4 respectively.

The outputs depend only on u_1 and u_4, and we need to differentiate them twice to obtain the inputs explicitly, the relative degree is $\mathbf{r} = 8$. On the other hand we have $n = 12$, which makes the decoupling matrix singular (not invertible) and the static feedback not possible. Then, a dynamic compensator will be used.

The idea that can be applied on the quadcopter model (2.19) is to separate u_1 from $\ddot{y}_1$, $\ddot{y}_2$ and $\ddot{y}_3$ by delaying the appearance of u_1 to higher order derivatives of outputs so the relative degree will increase.

This operation results also an augmented state space model (n increases). Therefore $\mathbf{r}$ and n will tend to approach the condition sought in previous section ($\mathbf{r} = n$) and the static feedback linearization will be applicable on the augmented system.

First, let's put the model (2.19) in the form (3.1). We rearrange the state vector:

$$\begin{aligned} \mathbf{x} &= \begin{bmatrix} x & y & z & \psi & \theta & \phi & \dot{x} & \dot{y} & \dot{z} & \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T \\ &= \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} & x_{11} & x_{12} \end{bmatrix}^T \end{aligned} \quad \in \mathbb{R}^{12} \tag{3.27}$$

The outputs vector is:

$$\mathbf{y} = \begin{bmatrix} x & y & z & \psi \end{bmatrix}^T \quad \in \mathbb{R}^4 \tag{3.28}$$

$$\mathbf{y} = h(\mathbf{x}) = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad \in \mathbb{R}^4 \tag{3.29}$$

The functions f and g that describe the system are:

$$f(\mathbf{x}) = \begin{bmatrix} x_7 \\ x_8 \\ x_9 \\ x_{12} \\ x_{11} \\ x_{10} \\ 0 \\ 0 \\ g \\ \frac{(I_y - I_z)}{I_x} x_{11} x_{12} \\ \frac{(I_z - I_x)}{I_y} x_{10} x_{12} \\ \frac{(I_x - I_y)}{I_z} x_{10} x_{11} \end{bmatrix} \in \mathbb{R}^{12} \qquad g(\mathbf{x}) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{m}[c(x_6)s(x_4) + c(x_6)c(x_4)s(x_5)] & 0 & 0 & 0 \\ -\frac{1}{m}[c(x_4)s(x_5) - c(x_6)s(x_4)s(x_5)] & 0 & 0 & 0 \\ -\frac{1}{m}[c(x_6)c(x_5)] & 0 & 0 & 0 \\ 0 & \frac{1}{I_x} & 0 & 0 \\ 0 & 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & 0 & \frac{1}{I_z} \end{bmatrix} \in \mathbb{R}^{12 \times 4} \tag{3.30}$$

Now, we delay the input u_1 by double integrator as discussed earlier in section (3.1.1.3). The system's augmentation using integrators is made by:

$$\begin{cases} u_1 &= x_{a1} \\ \dot{x}_{a1} &= x_{a2} \\ \dot{x}_{a2} &= \bar{u}_1 \\ u_2 &= \bar{u}_2 \\ u_3 &= \bar{u}_3 \\ u_4 &= \bar{u}_4 \end{cases} \tag{3.31}$$

Notation. *The augmented system terms are expressed by a bar above them* " $\bar{}$ "

The augmented state space vector $\bar{\mathbf{x}} \in \mathbb{R}^{14}$ becomes:

$$\begin{aligned} \bar{\mathbf{x}} &= [\ x \quad y \quad z \quad \psi \quad \theta \quad \phi \quad \dot{x} \quad \dot{y} \quad \dot{z} \quad x_{a_1} \quad x_{a_2} \quad \phi \quad \theta \quad \psi\ ]^T \\ &= [\ x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6 \quad x_7 \quad x_8 \quad x_9 \quad x_{a_1} \quad x_{a_2} \quad x_{10} \quad x_{11} \quad x_{12}\ ]^T \end{aligned} \tag{3.32}$$

The outputs remain the same:

$$\bar{\mathbf{y}} = \begin{bmatrix} x & y & z & \psi \end{bmatrix}^T \qquad \in R^4 \tag{3.33}$$

$$\bar{\mathbf{y}} = \bar{h}(\bar{\mathbf{x}}) = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \in \mathbb{R}^{4 \times 1} \tag{3.34}$$

and the non linear augmented model:

$$\dot{\bar{\mathbf{x}}} = f(\bar{\mathbf{x}}) + g(\bar{\mathbf{x}})\bar{\mathbf{u}} \tag{3.35}$$

is given by:

$$\bar{f}(\bar{\mathbf{x}}) = \begin{bmatrix} x_7 \\ x_8 \\ x_9 \\ x_{12} \\ x_{11} \\ x_{10} \\ -\frac{1}{m}(c(x_6)s(x_4) + c(x_6)c(x_4)s(x_5))x_{a1} \\ -\frac{1}{m}(c(x_4)s(x_5) - c(x_6)s(x_4)s(x_5))x_{a1} \\ g - \frac{1}{m}(c(x_6)c(x_5))x_{a1} \\ x_{a2} \\ 0 \\ \frac{(I_y - I_z)}{I_x}x_{11}x_{12} \\ \frac{(I_z - I_x)}{I_y}x_{10}x_{12} \\ \frac{(I_x - I_y)}{I_z}x_{10}x_{11} \end{bmatrix} \qquad \bar{g}(\bar{\mathbf{x}}) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & \frac{1}{I_x} & 0 & 0 \\ 0 & 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & 0 & \frac{1}{I_z} \end{bmatrix} \tag{3.36}$$

From (3.36) and (3.34), we can see that to make the inputs appear explicitly in the outputs formula, we need to differentiate each of the outputs y_1, y_2 and y_3 four times each one and y_4 two times so the new relative degree becomes ($\bar{r} = 14$) and ($\bar{n} = 14$). The system (3.36)-(3.34) is now decouplable and linearizable by a S-FBL. All we have to do is to apply the procedure described in section (3.1.1.1).

The decoupling and linearizing control $\bar{\mathbf{u}}$ is given by [31]:

$$\bar{\mathbf{u}} = D(\bar{\mathbf{x}})^{-1}(-D_0(\bar{\mathbf{x}}) + \mathbf{v}) \tag{3.37}$$

with

$$D = \begin{bmatrix} L_{\bar{g}_1}L_{\bar{f}}^{r_1-1}\bar{h}_1(\bar{\mathbf{x}}) & L_{\bar{g}_2}L_{\bar{f}}^{r_1-1}\bar{h}_1(\bar{\mathbf{x}}) & L_{\bar{g}_3}L_{\bar{f}}^{r_1-1}\bar{h}_1(\bar{\mathbf{x}}) & L_{\bar{g}_4}L_{\bar{f}}^{r_1-1}\bar{h}_1(\bar{\mathbf{x}}) \\ L_{\bar{g}_1}L_{\bar{f}}^{r_2-1}\bar{h}_2(\bar{\mathbf{x}}) & L_{\bar{g}_2}L_{\bar{f}}^{r_2-1}\bar{h}_2(\bar{\mathbf{x}}) & L_{\bar{g}_3}L_{\bar{f}}^{r_2-1}\bar{h}_2(\bar{\mathbf{x}}) & L_{\bar{g}_4}L_{\bar{f}}^{r_2-1}\bar{h}_2(\bar{\mathbf{x}}) \\ L_{\bar{g}_1}L_{\bar{f}}^{r_3-1}\bar{h}_3(\bar{\mathbf{x}}) & L_{\bar{g}_2}L_{\bar{f}}^{r_3-1}\bar{h}_3(\bar{\mathbf{x}}) & L_{\bar{g}_3}L_{\bar{f}}^{r_3-1}\bar{h}_3(\bar{\mathbf{x}}) & L_{\bar{g}_4}L_{\bar{f}}^{r_3-1}\bar{h}_3(\bar{\mathbf{x}}) \\ L_{\bar{g}_1}L_{\bar{f}}^{r_4-1}\bar{h}_4(\bar{\mathbf{x}}) & L_{\bar{g}_2}L_{\bar{f}}^{r_4-1}\bar{h}_4(\bar{\mathbf{x}}) & L_{\bar{g}_3}L_{\bar{f}}^{r_4-1}\bar{h}_4(\bar{\mathbf{x}}) & L_{\bar{g}_4}L_{\bar{f}}^{r_4-1}\bar{h}_4(\bar{\mathbf{x}}) \end{bmatrix} \tag{3.38}$$

$$D_0 = \begin{bmatrix} L_{\bar{f}}^{r_1}\bar{h}_1(\bar{\mathbf{x}}) \\ L_{\bar{f}}^{r_2}\bar{h}_2(\bar{\mathbf{x}}) \\ L_{\bar{f}}^{r_3}\bar{h}_3(\bar{\mathbf{x}}) \\ L_{\bar{f}}^{r_4}\bar{h}_4(\bar{\mathbf{x}}) \end{bmatrix}. \tag{3.39}$$

Since their are many terms and the Lie derivative calculation can be easily tricky and source of errors, D_0, D and D^{-1} are obtained using symbolic MATLAB Code.

Remark. *1. To obtain* $\mathbf{u}$*, we have to integrate twice* $\bar{u}_1$ *so :* $\mathbf{u} = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 \end{bmatrix}^T = \begin{bmatrix} \int\int \bar{u}_1 & \bar{u}_2 & \bar{u}_3 & \bar{u}_4 \end{bmatrix}^T$*, This operation will allow us to use the original system instead of the augmented system for simulation.*

2. The matrix D *is nonsingular at any point characterized by* $x_{a1} \neq 0$, $-\frac{\pi}{2} < \phi < +\frac{\pi}{2}$ *and* $-\frac{\pi}{2} < \theta < +\frac{\pi}{2}$ *so the Input-Output exact feedback linearization problem is solvable.*

By replacing (3.37) in (3.35), we have:

$$\dot{\bar{\mathbf{x}}} = f(\bar{\mathbf{x}}) + g(\bar{\mathbf{x}})(\mathbf{D}(\bar{\mathbf{x}})^{-\mathbf{1}}(-\mathbf{D_0}(\bar{\mathbf{x}}) + \mathbf{v})) \tag{3.40}$$

At this stage, the relationship between the inputs and the outputs is linear and expressed as follow:

$$\begin{bmatrix} y_1^{(4)} \\ y_2^{(4)} \\ y_3^{(4)} \\ y_4^{(2)} \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} \tag{3.41}$$

The change of coordinates is given by:

$$\mathbf{z} = \Phi(\bar{\mathbf{x}}) = \begin{cases} z_1 = \bar{h}_1(\bar{\mathbf{x}}) = x \\ z_2 = L_{\bar{f}}\bar{h}_1(\bar{\mathbf{x}}) = \dot{x} \\ z_3 = L^2_{\bar{f}}\bar{h}_1(\bar{\mathbf{x}}) = \ddot{x} \\ z_4 = L^3_{\bar{f}}\bar{h}_1(\bar{\mathbf{x}}) = \dddot{x} \\ z_5 = h_2(\bar{\mathbf{x}}) = y \\ z_6 = L_{\bar{f}}\bar{h}_2(\bar{\mathbf{x}}) = \dot{y} \\ z_7 = L^2_{\bar{f}}\bar{h}_2(\bar{\mathbf{x}}) = \ddot{y} \\ z_8 = L^3_{\bar{f}}\bar{h}_2(\bar{\mathbf{x}}) = \dddot{y} \\ z_9 = h_3(\bar{\mathbf{x}}) = z \\ z_{10} = L_{\bar{f}}\bar{h}_3(\bar{\mathbf{x}}) = \dot{z} \\ z_{11} = L^2_{\bar{f}}\bar{h}_3(\bar{\mathbf{x}}) = \ddot{z} \\ z_{12} = L^3_{\bar{f}}\bar{h}_3(\bar{\mathbf{x}}) = \dddot{z} \\ z_{13} = h_4(\bar{\mathbf{x}}) = \psi \\ z_{14} = L_{\bar{f}}\bar{h}_4(\bar{\mathbf{x}}) = \dot{\psi} \end{cases} \tag{3.42}$$

And the new system in which the linear techniques can be applied appears in Brunovsky's canonical form as:

$$\begin{cases} \dot{\mathbf{z}} = A\mathbf{z} + B\mathbf{v} \\ \mathbf{y} = C\mathbf{z} + D\mathbf{v} \end{cases} \tag{3.43}$$

where

$$\mathbf{z} = \begin{bmatrix} z_1 & z_2 & z_3 & z_4 & z_5 & z_6 & z_7 & z_8 & z_9 & z_{10} & z_{11} & z_{12} & z_{13} & z_{14} \end{bmatrix}^T \tag{3.44}$$

$$\mathbf{v} = \begin{bmatrix} v_1 & v_2 & v_3 & v_4 \end{bmatrix}^T \tag{3.45}$$

$$A = \begin{bmatrix} A_x & 0 & 0 & 0 \\ 0 & A_y & 0 & 0 \\ 0 & 0 & A_z & 0 \\ 0 & 0 & 0 & A_\psi \end{bmatrix} \quad B = \begin{bmatrix} B_x \\ B_y \\ B_z \\ B_\psi \end{bmatrix} \quad C = \begin{bmatrix} C_x & 0 & 0 & 0 \\ 0 & C_y & 0 & 0 \\ 0 & 0 & C_z & 0 \\ 0 & 0 & 0 & C_\psi \end{bmatrix} \quad D = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.46}$$

$$A \in \mathbb{R}^{14\times 4} \qquad B \in \mathbb{R}^{14\times 4} \qquad C \in \mathbb{R}^{4\times 14} \qquad D \in \mathbb{R}^{4\times 4} \tag{3.47}$$

where

$$A_x = A_y = A_z = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad A_\psi = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \tag{3.48}$$

$$B_x = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad B_y = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad B_z = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad B_\psi = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.49}$$

$$C_x = C_y = C_z = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \qquad C_\psi = \begin{bmatrix} 1 & 0 \end{bmatrix}. \tag{3.50}$$

Remark. *Using Kalman criteria, it is easy to check that the pairs (C, A) and (A, B) are observable and controllable respectively. So a full state feedback control and stabilisation by pole placement are possible.*

3.1.2.1 Pole Placement

In this section, we will design the linear control $\mathbf{v}$ by a pole placement since the necessary and sufficient condition of full controllability is verified for the system (3.43):

$$\begin{cases} M_c = \begin{bmatrix} B & AB & A^2B & \cdots & A^{13}B \end{bmatrix} \\ rank(M_c) = 14 \end{cases} \tag{3.51}$$

The full state feedback control law for tracking trajectory problem while the controllability condition is verified is given by:

$$\mathbf{v} = -\mathbf{K}.\mathbf{z} + \mathbf{H}.\mathbf{r}_{ef} \tag{3.52}$$

where $\mathbf{r}_{ef} \in R^4$ is the desired trajectory (reference signals). $K \in R^{4\times 14}$ is the feedback gain matrix. $H \in R^{4\times 4}$ is an extra gain matrix used to scale the closed loop transfer function.

The system (3.43) becomes:

$$\begin{cases} \dot{\mathbf{z}} = (A - BK)\mathbf{z} + BH\mathbf{r}_{ef} \\ \mathbf{y} = (C - DK)\mathbf{z} + DH\mathbf{r}_{ef} \end{cases} \tag{3.53}$$

In order to design a tracking reference control, if we consider $\mathbf{r}_{ef}$ as constant and if $(A - BK)$ is Hurwitz, the following relation is true and applicable in our situation:

$$\lim_{t\to+\infty} \mathbf{y} = [\mathbf{D} - (\mathbf{C} - \mathbf{DK})(\mathbf{A} - \mathbf{BK})^{-1}\mathbf{B}]\mathbf{H}\mathbf{r}_{ef} \tag{3.54}$$

This control problem is called the eigenvalue assignment problem or pole placement problem. Note that the **H** does not affect the stability of the system which is determined by the eigenvalues of (A - BK), but does affect the steady state solution. Hence **H** should be chosen such that $y = r_{ef}$ (the desired output value).

Remark. *Note that this development assumed that* $\mathbf{r}_{ef}$ *was constant, but it could also be used if* $\mathbf{r}_{ef}$ *is a slowly time-varying reference.*

The eigenvalues vector P of $A - BK$ is chosen in a way to ensure desired performances (stability, response time, overshot ...) for the closed loop with at least ensuring the loop stability by imposing all poles in Real Half Plane. This operation can be easily done by MATLAB command: $K = place(A,B,P)$.

Remark. *A possible way to choose the closed loop poles for a desired performances is to place them not too far from the open loop poles so the control signal still realisable and reduce the overshoots amplitude, but the response time becomes slower.*

Now, we still have to determine **H** such that $y(t) \to r_{ef} \quad$ when $\quad t \to +\infty$. From Eq.(3.54) we can take:

$$[D - (C - DK)(A - BK)^{-1}B]H = I, \tag{3.55}$$

where $I \in \mathbb{R}^{4\times4}$ is the identity matrix (we must have the same number of inputs and controlled outputs).

3.1.2.2 Luenberger Observer design

Since the state of the linearized system (3.43) is not fully available, the implementation of Luenberger observer will be sufficient to estimate the state based on the outputs of the nonlinear system (3.29) and its inputs **v**. The observer's structure is given by:

$$\begin{cases} \dot{\hat{\mathbf{z}}} = A\hat{\mathbf{z}} + B\mathbf{v} + \mathbf{L}(C\hat{\mathbf{z}} - \mathbf{y}) \\ \hat{\mathbf{y}} = C\hat{\mathbf{z}}, \end{cases} \tag{3.56}$$

where $\hat{\mathbf{z}} \in R^{14}$ is the observed states and the outputs vector of Luenberger Observer. $\mathbf{y} \in \mathbb{R}^{\mathbf{4}}$ and $\mathbf{v} \in \mathbb{R}^{\mathbf{4}}$ are the inputs of the observer. $\mathbf{L} \in \mathbb{R}^{\mathbf{14\times4}}$ is the observer gain matrix that adjusts the state due to error $(\mathbf{C}.\hat{\mathbf{z}} - \mathbf{y})$.

Theorem 3.1.2. *If the pair (C, A) is observable , then we can place the eigenvalues of A+LC in the negative real half plane to ensure stability through appropriate choice of L [36].*

Remark.

1. *The system (3.56) is used to calculate the linear control* **v** *which will be used in (3.40).*
2. **L** *is also chosen by pole placement. We prefer to assign its poles more far in the negative half plane than the controller poles, so the observer estimation error converges to 0 faster.*

Now, all the elements for the simulation are available. The final structure of our quadcopter controlled by a FBL is shown in Figure 3.1.

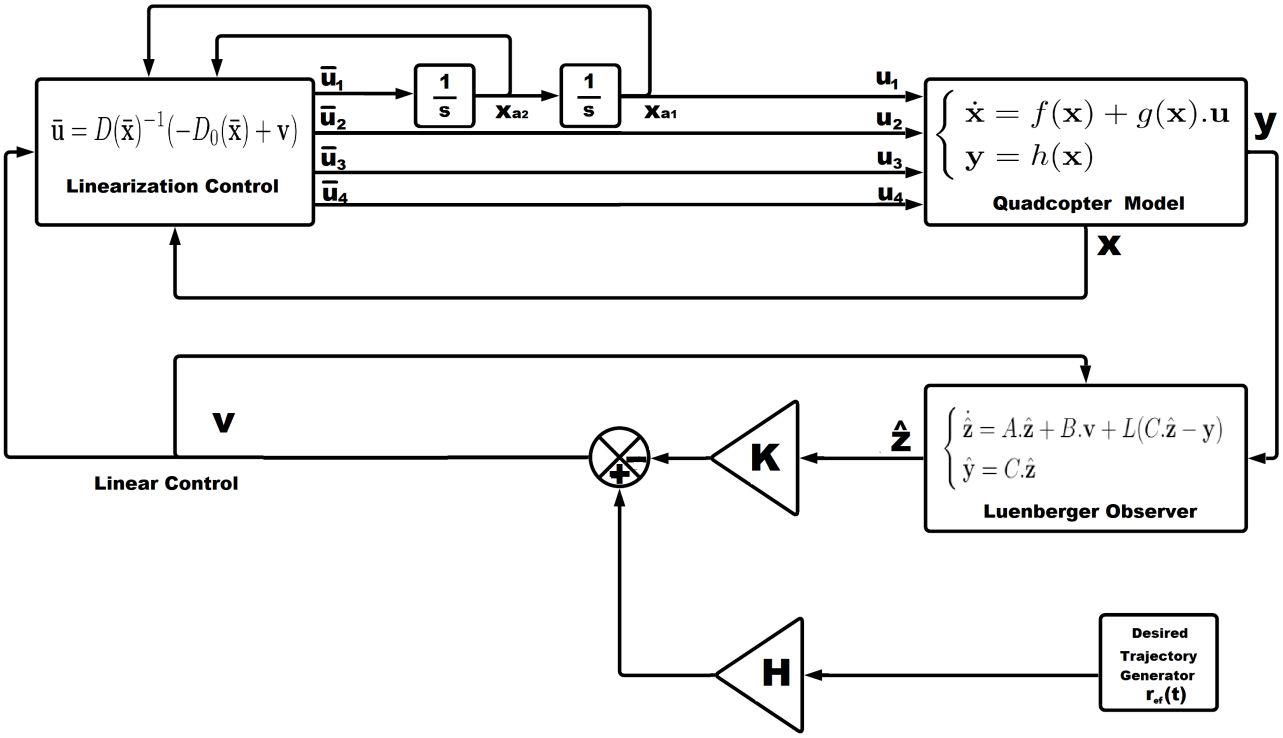


Figure 3.1: The complete structure of the quadcopter controlled by a FBL technique

3.1.3 Simulation and Results

In this section a simulation carried out with Matlab/Simulink to show the behavior of FBL developed earlier for a helical trajectory of OS4, the desired outputs are shown in Table 3.1:

Output	Desired Value	Initialisation
$x_d(t)$	$5sin((2\pi)0.1t)$	0 $[m]$
$y_d(t)$	$5(1 - cos((2\pi)0.1t))$	0 $[m]$
$z_d(t)$	$-0.2t$	0 $[m]$
$\psi_d(t)$	0	0 $[rad]$

Simulation Time	100 $[s]$

Table 3.1: Desired values for the helical trajectory

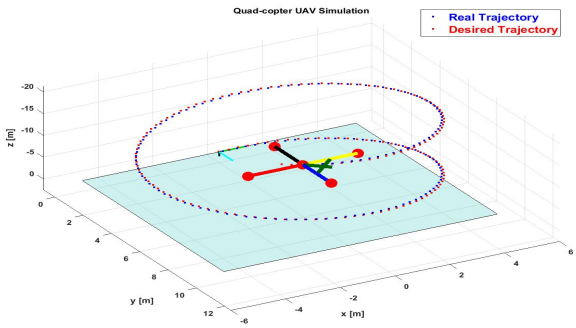
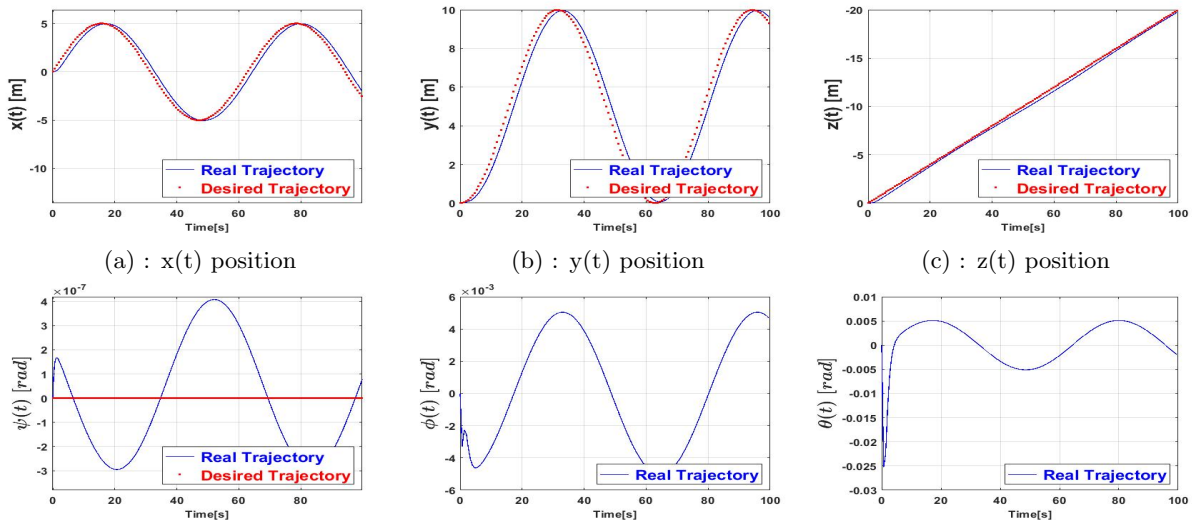


Figure 3.2: Real and desired trajectory with FBL control

From Figure 3.3, we can see that the controller managed to lead the system to follow the desired outputs



(a) : x(t) position (b) : y(t) position (c) : z(t) position

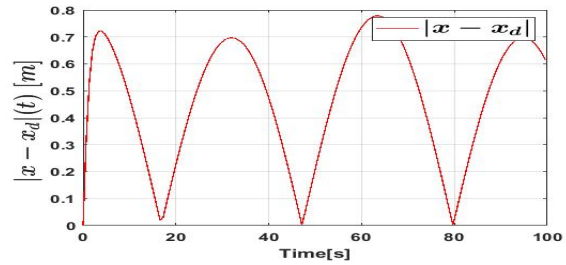
(d) : ψ(t) angle (e) : ϕ(t) angle (f) : θ(t) angle

Figure 3.3: Position and attitude throw time for FBL

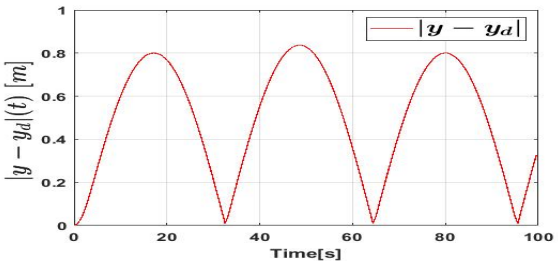
$\begin{bmatrix} x & y & z & \psi \end{bmatrix}^T$ and kept the angles $\begin{bmatrix} \phi & \theta \end{bmatrix}^T$ within a stable zone that respect the small oscillations assumption. But a remarkable static error can be observed on the four desired states. In Figure 3.4, we can remark the order of static error for positions is greater than 50 cm and has sinusoidal shape with peaks at each maximal value of the tangent of desired trajectory that can be interpreted by the sensitivity of the controller to the change of desired outputs. Also the sinusoidal desired outputs x and y affected the rest of states as z and ψ.

Also, by looking at the control behaviour in Figure 3.5, we can see that it is realisable physically, since it respects the nature of thrust f_t that should be always positive and the small values of torques since we assumed small angles of oscillations.

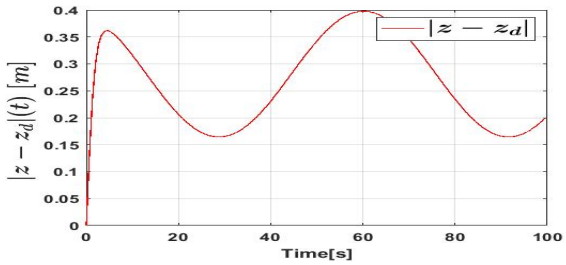
We can conclude for the FBL controller in the absence of disturbances and with the respect of initial conditions, that it is able to achieve the desired references with realisable control and insurance of stability but with a significant static error in the case those references are dynamic. In order to increase the controler performances a deeper work on the pole placement and the references is needed.



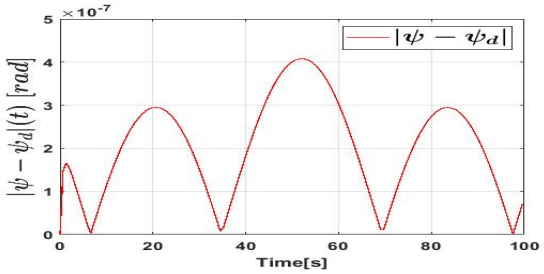
(a) : x error



(b) : y error

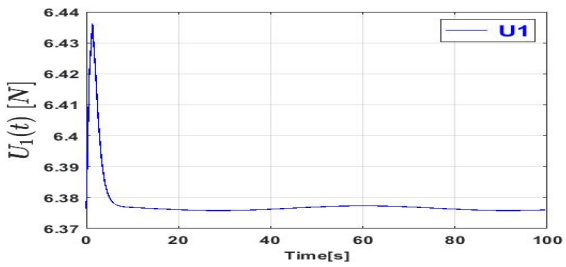


(c) : z error

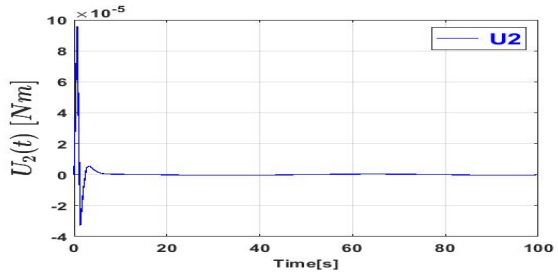


(d) : ψ error

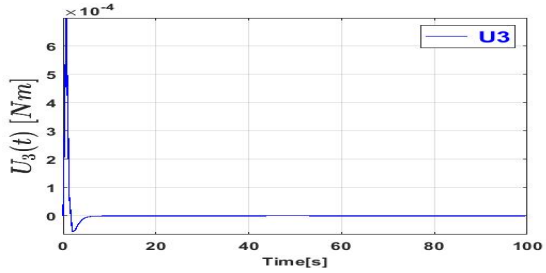
Figure 3.4: Error between real and desired outputs



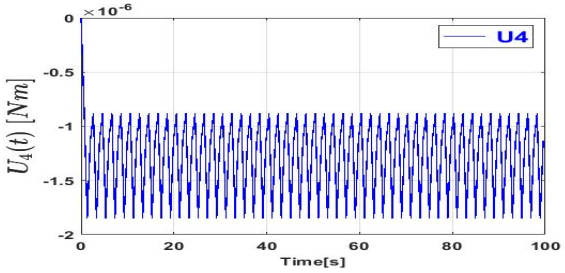
(a) : thrust f_t



(b) : moment τ_x



(c) : moment τ_y



(d) : moment τ_z

Figure 3.5: The outputs of FBL controller

3.2 Backstepping technique

Backstepping (BS) is one of recursive non linear control techniques based on Lyapounov theory to guarantee global asymptotic stability that has been used widely on quadcopters. This approach is known for its robustness against wind disturbance, convergence to the desired path and controlling under-actuated systems with large regions of attraction [1].

The origin of BS is hard to track since it appeared implicitly in different publications before 1995 as mentioned in [37] but in general, it is attributed to Petar V. Kokotovic [38]. The idea behind this technique is to divide a full system control problem into subsystems that have a recursive structure. Each subsystem will be controlled virtually by the next one until we arrive to the final subsystem that contain the real control which will be extracted at this point.

One class of systems on which the BS algorithm is applicable is so-called "Strict-feedback" form (3.57). It is named so because every interconnection in the system is a feedback connection from the states located farther from the input [39]. We will show later that the model of the quadcopter can be put in this form.

$$\begin{cases} \dot{x}_1 & = f_1(x_1) + g_1(x_1)x_2 \\ \dot{x}_2 & = f_2(x_1,x_2) + g_2(x_1,x_2)x_3 \\ & \vdots \\ \dot{x}_{n-1} & = f_{n-1}(x_1,x_2,...,x_{n-1}) + g_{n-1}(x_1,x_2,...,x_{n-1})x_n \\ \dot{x}_n & = f_n(x_1,x_2,...,x_n) + g_n(x_1,x_2,...,x_n)u. \end{cases} \tag{3.57}$$

3.2.1 The Backstepping Concept

Since there are different forms of BS, in this section we'll describe the BS approach for a strict feedback form for explanation purpose.

We are looking to design a feedback controller u that stabilizes the system around zero. Let's consider the following strict-feedback system:

$$\begin{cases} \dot{\mathbf{z}} = f(\mathbf{z}) + g(\mathbf{z})\zeta \\ \dot{\zeta} = u, \end{cases} \tag{3.58}$$

where $\begin{bmatrix}\mathbf{z}^T & \zeta\end{bmatrix}^T \in \mathbb{R}^{k+1}$ is the state space vector. $u \in \mathbb{R}$ is the control input. $f : D \to \mathbb{R}^k$ and $g : D \to \mathbb{R}^k$ are smooth[3] nonlinear functions vectors in a domain $D \subset \mathbb{R}^n$ that contains $\mathbf{z} = 0$ and $f(0) = 0$.

Definition 3.2.1. *A function V(x) satisfying:*

$$V(x) \to \infty \text{ as } \|x\| \to \infty \tag{3.59}$$

is said to be radially unbounded.

Definition 3.2.2 (Positive Definite function)**.** *A continuously differentiable function $V : \mathbb{R}^n \to \mathbb{R}$ is positive definite in a region $D \subset \mathbb{R}^n$ containing the origin if $V(0) = 0$ and $V(x) > 0$, $x \in D$ and $x \neq 0$*

if $V(x) \geq 0, z \in D$ and $z \neq 0$ then it is said to be positive semi-definite.

Theorem 3.2.1. *Let $\mathbf{z} = 0$ be an equilibrium point for (3.58) and $\mathcal{D} \subset \mathbb{R}^n$ be a domain containing $\mathbf{z} = 0$. Let $V : \mathcal{D} \to \mathbb{R}$ be a function assumed to be positive definite and radially unbounded that its derivative verify :*

$$\dot{V}(\mathbf{z}) = \left(\frac{dV(\mathbf{z})}{d\mathbf{z}}\right)^T \dot{\mathbf{z}} = \left(\frac{dV(\mathbf{z})}{d\mathbf{z}}\right)^T (f(\mathbf{z}) + g(\mathbf{z})v) \leq -W(\mathbf{z}), \tag{3.60}$$

where $W(\mathbf{z})$ is any positive semi-definite function, then the solution $\mathbf{z}(t) = 0$ of the system $\dot{\mathbf{z}} = f(\mathbf{z}) + g(\mathbf{z})v$ is stable. If $W(\mathbf{z})$ is any positive definite function, then the solution $\mathbf{z}(t) = 0$ of the same system is also asymptotically stable.[33]

Definition 3.2.3. *A continuously differentiable function $V(\mathbf{z})$ satisfying the theorem 3.2.1 is called a Lyapunov function.*

Now that the basic definitions are set, let's introduce an explanatory case. For this purpose consider the system defined by:

$$\begin{cases} \dot{x}_1 = f(x_1) + g(x_1)x_2 \\ \dot{x}_2 = u \end{cases} \qquad f(0) = 0 \text{ and } g(0) \neq 0 \tag{3.61}$$

We separate the system (3.61) into two subsystems Σ_1 and Σ_2 respectively:

$$\begin{cases} \Sigma_1 : \quad \dot{x}_1 = f(x_1) + g(x_1)v \\ \\ \Sigma_2 : \quad \dot{x}_2 = u \end{cases} \tag{3.62}$$

The idea is to consider x_2 as a virtual control v for the first subsystem, once we obtain its control formula and since we can't directly impose it because x_2 is a state so we design a second control that forces the second subsystem to follow the desired first virtual control throw u.

Suppose the first subsystem can be stabilized by a smooth state feedback control:

$$v = \phi(x_1) \qquad \text{with} \quad \phi(0) = 0 \tag{3.63}$$

We write the equations of the original system as:

$$\begin{cases} \dot{x}_1 = f(x_1) + \phi(x_1)g(x_1) + g(x_1)(x_2 - \phi(x_1)) \\ \dot{x}_2 = u \end{cases} \tag{3.64}$$

Suppose further that we know a radially unbounded positive definite Lyapunov function $V = V(x_1)$ that satisfies (3.60).

Now we introduce a new state variable that is the error between the real state x_2 and the virtual control (here, we begin controlling the second subsystem Σ_2):

$$\bar{x}_2 = x_2 - \phi(x_1) \tag{3.65}$$

We differentiate (3.65), we get new control $\bar{u}$ such that:

$$\begin{cases} \dot{\bar{x}}_2 = \bar{u} \\ \bar{u} = u - \dot{\phi} = u - \frac{d\phi(x_1)}{dx_1}(f(x_1) + g(x_1)\phi(x_1)) \end{cases} \tag{3.66}$$

The equation (3.66) reduces the system to the cascade connection. The real dynamics are now given by:

$$\begin{cases} \dot{x}_1 = f(x_1) + \phi(x_1)g(x_1) + g(x_1)\bar{x}_2 \\ \dot{\bar{x}}_2 = \bar{u}. \end{cases} \tag{3.67}$$

The system (3.67) is similar to the system we started from, except that the first component has a stable origin when the input is zero. The complete system will be stabilized by designing $\bar{u}$ using this feature.

Consider the Lyapounov candidate function:

$$V_2 = V(x_1) + \frac{1}{2}(\bar{x}_2)^2, \tag{3.68}$$

Hence:

$$\dot{V}_2 = \dot{V}(x_1) + \bar{x}_2\dot{\bar{x}}_2 = \dot{V}(x_1) + \bar{x}_2\bar{u}. \tag{3.69}$$

Now, it's up to choose the controller $\bar{u}$ such $\dot{V}_2$ is negative semi-definite to guarantee the stability. Then, we extract the original control u.

A convenient choice of $\bar{u}$ is $-A\bar{x}_2$ with $A > 0$ a scalar to be tuned. Substituting in (3.66) we get:

$$\bar{u} = -A\bar{x}_2 = u - \frac{d\phi(x_1)}{dx_1}(f(x_1) + g(x_1)x_2) = -A(x_2 - \phi(x_1)). \tag{3.70}$$

And the augmented Lyapunov function V_2 becomes:

$$\dot{V}_2 = \dot{V}(x_1) - A_2(\bar{x}_2)^2 \leq 0, \tag{3.71}$$

this shows that the origin ($x_1 = 0,\ \ \bar{x}_2 = 0$) is stable. Since $\phi(0) = 0$, we conclude that the origin ($x_1 = 0, x_2 = 0$) is also stable.

Finally, the control law of the original system is:

$$u = \frac{d\phi(x_1)}{dx_1}(f(x_1) + g(x_1)x_2) - A(x_2 - \phi(x_1)). \tag{3.72}$$

Thus, the origin(x_1,x_2) = (0,0) is globally stable with the aforementioned control law. This is the principle behind backstepping. We can formally state this as lemma that could be applied recursively to higher-order systems [33].

Lemma 3.2.1. *Consider the system (3.58). Let $v(\mathbf{z})$ be a stabilizing state feedback control for the subsystem $\dot{\mathbf{z}}$ in (3.58) with $v(0) = 0$ and $V(\mathbf{z})$ be a Lyapunov function that satisfies (3.60) with some positive definite function $W(\mathbf{z})$. Then the state feedback control:*

$$u = \frac{dv}{d\mathbf{z}}[f(\mathbf{z}) + g(\mathbf{z})\zeta] - \frac{dV}{d\mathbf{z}}g(\mathbf{z}) - A[\zeta - v(\mathbf{z})] \qquad A > 0 \tag{3.73}$$

stabilizes the origin of (3.58), with

$$V_a(\mathbf{z}, \zeta) = V(\mathbf{z}) + \frac{1}{2}A[\zeta - v(\mathbf{z})]^2, \tag{3.74}$$

as V_a the augmented Lyapunov function. Moreover, if all the assumptions hold globally and $V(\mathbf{z})$ is radially unbounded, the origin will be globally stable. An extension to the asymptotic stability can be made [30].

Example: Consider the following system:

$$\begin{cases} \dot{x}_1 = x_1^2 + [cos(x_1) + 2]x_2 \\ \dot{x}_2 = u. \end{cases} \tag{3.75}$$

We desire to design a BS for this plant that will stabilize it around zero (regulation problem). Since, it is already in strict feedback form, we will proceed as follows:

Step 1: Consider the first sub-system:

$$\dot{x}_1 = (x_1)^2 + (cos(x_1) + 2)v. \tag{3.76}$$

The virtual control law $v = \frac{-(x_1)^2}{cos(x_1)+2} + \frac{-A_1 x_1}{cos(x_1)+2} \leq 0$ can stabilize the system for $(A_1 > 0)$ and $(cos(x_1) + 2 \neq 0)$

We choose Lyapounov candidate function:

$$\begin{cases} V_1 = \frac{1}{2}(x_1)^2 \\ \dot{V}_1 = x_1\dot{x}_1 = x_1((x_1)^2 + (cos(x_1) + 2)v). \end{cases} \tag{3.77}$$

Replacing v by its expression gives:

$$\dot{V}_1 = -A_1(x_1)^2. \tag{3.78}$$

Step 2 : Now let's control the second subsystem:

$$\dot{x}_2 = u. \tag{3.79}$$

We want x_2 to match v, we calculate the error $x_2 - v$, from the expression of $\dot{x}_1$ in (3.75) we get the real state x_2:

$$x_2 = \frac{\dot{x}_1 - (x_1)^2}{cos(x_1) + 2}. \tag{3.80}$$

Now we have:

$$x_2 - v = \frac{\dot{x}_1 + A_1 x_1}{cos(x_1) + 2}. \tag{3.81}$$

We know that $(cos(x_1) + 2 > 0 \quad \forall \ x_1 \in \mathbb{R})$ and for simplicity reasons we define a new variable $z = \dot{x}_1 + A_1 x_1$

The augmented Lyapunov candidate function becomes:

$$V_2 = \frac{x_1^2}{2} + \frac{z^2}{2} = \frac{(x_1)^2}{2} + \frac{((cos(x_1) + 2)x_2 + (x_1)^2 + A_1 x_1)^2}{2}. \tag{3.82}$$

It is clear to notice that V_2 is definite positive. All we need is to get $\dot{V}_2$ semi-definite negative so the stability is ensured.

We differentiate and rearrange V_2, we get:

$$\dot{V}_2 = ((c(x_1)+2)x_2 + x_1^2 + A_1x_1)(c(x_1)+2)(u + \frac{x_1 + (2x_1 - s(x_1) + A_1)(x_1^2 + (c(x_1)+2)x_2)}{(c(x_1)+2)}) - A_1x_1^2. \quad (3.83)$$

The control law is chosen as:

$$u + \frac{x_1 + (2x_1 - s(x_1) + A_1)(x_1^2 + (c(x_1)+2)x_2)}{c(x_1)+2} = -A_2(c(x_1)+2)z, \quad (3.84)$$

where $A_2 > 0$.

By replacing u in $\dot{V}_2$ we get :

$$\dot{V}_2 = -A_2[(cos(x_1)+2)x_2 + x_1^2 + A_1x_1]^2[cos(x_1)+2]^2 - A_1x_1^2 \leq 0. \quad (3.85)$$

Finally, we get $V_1 > 0$ and $V_2 > 0$ also $\dot{V}_1 \leq 0$ and $\dot{V}_2 \leq 0$. The real control u is extracted:

$$u = -\frac{x_1 + (2x_1 - s(x_1) + A_1)(x_1^2 + (c(x_1)+2)x_2)}{c(x_1)+2} - A_2((c(x_1)+2)x_2 + x_1^2 + A_1x_1)(c(x_1)+2), \quad (3.86)$$

where A_1 and A_2 are a positive real gains that need to be tuned.

Remark. *The regulation problem can be transformed into tracking trajectory problem by making the variable change $e_1 = x_1 - x_{1_d}$ where e_1 is the error between the real state x_1 and the desired state x_{1_d} into Lyapunov candidate function, an illustration is made in section 3.2.2.*

3.2.2 Backstepping: Application to Our Model

In order to apply BS on the quadcopter, let's rearrange the model (2.19) to make the design easier. The new state space vector $\mathbf{x} \in \mathbb{R}^{12}$ is given by:

$$\begin{aligned} \mathbf{x} &= [\ \phi\ \ \dot{\phi}\ \ \theta\ \ \dot{\theta}\ \ \psi\ \ \dot{\psi}\ \ z\ \ \dot{z}\ \ x\ \ \dot{x}\ \ y\ \ \dot{y}\ ]^T \\ &= [\ x_1\ \ x_2\ \ x_3\ \ x_4\ \ x_5\ \ x_6\ \ x_7\ \ x_8\ \ x_9\ \ x_{10}\ \ x_{11}\ \ x_{12}\ ]^T. \end{aligned} \quad (3.87)$$

The rearranged system becomes:

$$\begin{cases} \Sigma_1 : \begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{I_y - I_z}{I_x} x_4 x_6 + \frac{1}{I_x} u_2 + d_\phi \end{cases} \\ \Sigma_2 : \begin{cases} \dot{x}_3 &= x_4 \\ \dot{x}_4 &= \frac{I_z - I_x}{I_y} x_2 x_6 + \frac{1}{I_y} u_3 + d_\theta \end{cases} \\ \Sigma_3 : \begin{cases} \dot{x}_5 &= x_6 \\ \dot{x}_6 &= \frac{I_x - I_y}{I_z} x_4 x_2 + \frac{1}{I_z} u_4 + d_\psi \end{cases} \\ \Sigma_4 : \begin{cases} \dot{x}_7 &= x_8 \\ \dot{x}_8 &= g - \frac{1}{m}(c(x_1)c(x_3))u_1 + d_z \end{cases} \\ \Sigma_5 : \begin{cases} \dot{x}_9 &= x_{10} \\ \dot{x}_{10} &= -\frac{1}{m}(c(x_1)c(x_5)s(x_3) + s(x_1)s(x_5))u_1 + d_x \end{cases} \\ \Sigma_6 : \begin{cases} \dot{x}_{11} &= x_{12} \\ \dot{x}_{12} &= -\frac{1}{m}(c(x_1)s(x_3)s(x_5) - c(x_5)s(x_1))u_1 + d_y. \end{cases} \end{cases} \tag{3.88}$$

We consider the disturbances to be null ($d_x = d_y = d_z = d_\phi = d_\theta = d_\psi = 0$)

The control is designed in two stages. The first one is the inner loop control for the attitude and the altitude $\begin{bmatrix} \phi & \theta & \psi & z \end{bmatrix}^T$ by Backstepping technique. The second is the outer loop control for the horizontal position $\begin{bmatrix} x & y \end{bmatrix}^T$ by Proportional Derivative technique (PD).

This structure is chosen because the strict feedback form of subsystems in (3.88) allows to apply BS only on the attitude and one position from $\begin{bmatrix} x & y & z \end{bmatrix}^T$ since those three states are coupled with u_1 (under-actuated nature of the system).

But what is more interesting in autonomous applications is the translation movement rather then the rotational one, even this latter is the major responsible for quadcopters motions and stability with the altitude (Thrust and Torques). For this reason an outer loop control is necessary and will provide the desired positions $\begin{bmatrix} x & y \end{bmatrix}^T$. So the controller PD brings the angles $\begin{bmatrix} \phi & \theta \end{bmatrix}^T$ to values that ensure as minimum criteria the stability and an acceptable trajectory tracking.

3.2.2.1 Altitude and Attitude Control

Our outputs vector is $\begin{bmatrix} \phi & \theta & \psi & z \end{bmatrix}^T$. The procedure followed to design the inner controllers is similar to the one described in section 3.2.1.

We start by designing the control for ϕ angle, governed by subsystem Σ_1. We choose $u_2 = I_x(-\frac{I_y - I_z}{I_x} x_4 x_6 + \bar{u}_2)$

and get:

$$\Sigma_1 : \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \bar{u}_2. \end{cases} \tag{3.89}$$

Remark. *Such variable change will be useful to normalize the subsystems in order to avoid doing the same calculations 4 times for the next controls. we choose the same change of variable for the other three inputs:*

$$\begin{cases} u_1 = \frac{m}{c(x_1)x(x_3)}(+g - \bar{u}_1) \\ u_3 = I_y(-\frac{I_z - I_x}{I_y} x_2 x_6 + \bar{u}_3) \\ u_4 = I_z(-\frac{I_x - I_y}{I_z} x_2 x_4 + \bar{u}_4). \end{cases} \tag{3.90}$$

The system (3.89) is divided again into two subsystems, the first one is controlled virtually by v_2:

$$\dot{x}_1 = v_2. \tag{3.91}$$

The second one is controlled by the real control u_2 in order to force the real state x_2 to match v_2:

$$\dot{x}_2 = \bar{u}_2. \tag{3.92}$$

We consider the tracking error:

$$z_1 = \phi_d - \phi = x_{1_d} - x_1. \tag{3.93}$$

A Lyapunov candidate function is chosen:

$$V(z_1) = \frac{1}{2} z_1^2 \tag{3.94}$$

$$\dot{V}(z_1) = z_1 \dot{z}_1 = z_1(\dot{x}_{1_d} - \dot{x}_1) = z_1(\dot{x}_{1_d} - v_2). \tag{3.95}$$

The stabilization of z_1 is obtained by the virtual control v_2:

$$v_2 = \dot{x}_{1_d} + \alpha_1 z_1 \qquad with \quad \alpha_1 > 0 \text{ is a positive gain.} \tag{3.96}$$

The equation (3.95) becomes:

$$\dot{V}(z_1) = -\alpha_1 z_1^2. \tag{3.97}$$

Now, we want to force x_2 to follow v_2, we use variable change:

$$z_2 = x_2 - v_2 = x_2 - \dot{x}_{1_d} - \alpha_1 z_1. \tag{3.98}$$

The augmented Lyapunov candidate function is becomes:

$$V(z_1,z_2) = \frac{1}{2}(z_1^2 + z_2^2). \tag{3.99}$$

Its time derivative is:

$$\begin{aligned} \dot{V}(z_1,z_2) &= z_1\dot{z}_1 + z_2\dot{z}_2 = -\alpha_1(z_1)^2 + z_2\dot{z}_2 = -\alpha_1(z_1)^2 + z_2(\dot{x}_2 - \dot{v}_2) \\ &= -\alpha_1(z_1)^2 + z_2(\bar{u}_2 - \ddot{x}_{1_d} - \alpha_1(\dot{x}_{1_d} - \dot{x}_1)). \end{aligned} \tag{3.100}$$

$\bar{u}_2$ is chosen in way to make $\dot{V}(z_1,z_2) \leq 0$:

$$\begin{aligned} [\bar{u}_2 - \ddot{x}_{1_d} - \alpha_1(\dot{x}_{1_d} - \dot{x}_1)] &= -\alpha_2 z_2 \\ \bar{u}_2 &= +\ddot{x}_{1_d} + \alpha_1(\dot{x}_{1_d} - \dot{x}_1) - \alpha_2 z_2, \end{aligned} \tag{3.101}$$

where $\alpha_2 > 0$ is a positive gain.

We substitute in (3.100):

$$\dot{V}(z_1,z_2) = -\alpha_1(z_1)^2 - \alpha_2(z_2)^2 \quad \leq 0. \tag{3.102}$$

Finally, the real control can be extracted by replacing (3.101) in u_2:

$$u_2 = I_x(-\frac{I_y - I_z}{I_x} x_4 x_6 + \ddot{x}_{1_d} + \alpha_1(\dot{x}_{1_d} - \dot{x}_1) - \alpha_2(x_2 - \dot{x}_{1_d} - \alpha_1(x_{1_d} - x_1))). \tag{3.103}$$

The same steps are followed to obtain u_3 , u_4 and u_1, we can avoid the repetition as mentioned earlier by setting:

$$\begin{cases} \bar{u}_1 = +\ddot{x}_{7_d} + \alpha_7(\dot{x}_{7_d} - \dot{x}_7) - \alpha_8 z_8 \\ \bar{u}_3 = +\ddot{x}_{3_d} + \alpha_3(\dot{x}_{3_d} - \dot{x}_3) - \alpha_4 z_4 \\ \bar{u}_4 = +\ddot{x}_{5_d} + \alpha_5(\dot{x}_{5_d} - \dot{x}_5) - \alpha_6 z_6. \end{cases} \tag{3.104}$$

We replace (3.104) in (3.90):

$$u_1 = \frac{m}{cos(x_1)cos(x_3)}(g - \ddot{x}_{7_d} + \alpha_7(\dot{x}_{7_d} - \dot{x}_7) - \alpha_8(x_8 - \dot{x}_{7_d} - \alpha_7(x_{7_d} - x_7))) \tag{3.105}$$

$$u_3 = I_y(-\frac{I_z - I_x}{I_y} x_2 x_6 + \ddot{x}_{3_d} + \alpha_3(\dot{x}_{3_d} - \dot{x}_3) - \alpha_4(x_4 - \dot{x}_{3_d} - \alpha_3(x_{3_d} - x_3))) \tag{3.106}$$

$$u_4 = I_z(-\frac{I_x - I_y}{I_z} x_2 x_4 + \ddot{x}_{5_d} + \alpha_5(\dot{x}_{5_d} - \dot{x}_5) - \alpha_6(x_6 - \dot{x}_{5_d} - \alpha_5(x_{5_d} - x_5))). \tag{3.107}$$

We can define:

$$\begin{cases} z_3 = x_{3_d} - x_3 \\ z_4 = x_4 - \dot{x}_{3_d} - \alpha_3 z_3 \\ z_5 = x_{5_d} - x_5 \\ z_6 = x_6 - \dot{x}_{5_d} - \alpha_5 z_5 \\ z_7 = x_{7_d} - x_7 \\ z_8 = x_8 - \dot{x}_{7_d} - \alpha_7 z_7, \end{cases} \tag{3.108}$$

where $\begin{bmatrix} \alpha_3 & \alpha_4 & \alpha_5 & \alpha_6 & \alpha_7 & \alpha_8 \end{bmatrix} > 0$ are a positive gains.

The control vector of inner loop becomes:

$$U = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} -\frac{m}{cos(x_1)cos(x_3)}(-g + \ddot{x}_{7_d} + \alpha_7(\dot{x}_{7_d} - x_8) - \alpha_8(x_8 - \dot{x}_{7_d} - \alpha_7(x_{7_d} - x_7))) \\ I_x(-\frac{I_y - I_z}{I_x} x_4 x_6 + \ddot{x}_{1_d} + \alpha_1(\dot{x}_{1_d} - x_2) - \alpha_2(x_2 - \dot{x}_{1_d} - \alpha_1(x_{1_d} - x_1))) \\ I_y(-\frac{I_z - I_x}{I_y} x_2 x_6 + \ddot{x}_{3_d} + \alpha_3(\dot{x}_{3_d} - x_4) - \alpha_4(x_4 - \dot{x}_{3_d} - \alpha_3(x_{3_d} - x_3))) \\ I_z(-\frac{I_x - I_y}{I_z} x_2 x_4 + \ddot{x}_{5_d} + \alpha_5(\dot{x}_{5_d} - x_6) - \alpha_6(x_6 - \dot{x}_{5_d} - \alpha_5(x_{5_d} - x_5))) \end{bmatrix}. \tag{3.109}$$

Remark. *The conditions* $-\frac{\pi}{2} < \phi < +\frac{\pi}{2}$ *and* $-\frac{\pi}{2} < \theta < +\frac{\pi}{2}$ *are necessary in order to avoid the singularity in the input* u_1*. This singularity comes from the model itself that uses Euler's angles and it is known as "Gimbal Lock".*

3.2.2.2 Position Control

In this section, we are going to force roll and pitch angles to provide the desired x and y, because those latter aren't directly controlled by the vector U (under-actuated system), The desired roll and pitch angles formulas (ϕ_d, θ_d) are obtained from $(\dot{x}_8, \dot{x}_{10}, \dot{x}_{12})$ in (3.88):

$$\begin{cases} \ddot{z} = g - \frac{1}{m}(c(\phi_d)c(\theta_d))u_1 \\ \ddot{x} = -\frac{1}{m}(c(\phi_d)c(\psi)s(\theta_d) + s(\phi_d)s(\psi))u_1 \\ \ddot{y} = -\frac{1}{m}(c(\phi_d)s(\psi)s(\theta_d) - s(\phi_d)c(\psi))u_1. \end{cases} \tag{3.110}$$

Since our model is obtained for small oscillations and the desired altitude acceleration is chosen small enough to be neglected, the following assumptions are made to simplify the equations in (3.110):

$$\begin{cases} \ddot{z}_d & \approx & 0 \\ sin(\theta_d) & \approx & \theta_d \\ sin(\phi_d) & \approx & \phi_d \\ cos(\theta_d) & \approx & 1 \\ cos(\phi_d) & \approx & 1. \end{cases} \tag{3.111}$$

The equation (3.110) becomes:

$$\begin{cases} u_{1_d} = mg \\ \ddot{x} = -\frac{1}{m}(c(\psi)\theta_d + s(\psi)\phi_d)u_{1_d} \\ \ddot{y} = -\frac{1}{m}(s(\psi)\theta_d - c(\psi)\phi_d)u_{1_d}. \end{cases} \tag{3.112}$$

From this last equation, we get:

$$\begin{cases} \phi_d = -\frac{1}{g}(s(\psi)\ddot{x} - c(\psi)\ddot{y}) \\ \theta_d = -\frac{1}{g}(c(\psi)\ddot{x} + s(\psi)\ddot{y}). \end{cases} \tag{3.113}$$

Final step is to control position accelerations $(\ddot{x}, \ddot{y})$, so (x, y) and its derivatives match (x_d, y_d) and its desired derivatives. For that we design a PD controller as follows.

We denote $\ddot{x}_c$ and $\ddot{y}_c$ the controlled accelerations of $\ddot{x}_d$ and $\ddot{y}_d$ respectively. The proportional derivative controller signals can be written as:

$$\begin{cases} \ddot{x}_c = K_{p_x}(x_d - x) + K_{d_x}(\dot{x}_d - \dot{x}) + \ddot{x}_d \\ \ddot{y}_c = K_{p_y}(y_d - y) + K_{d_y}(\dot{y}_d - \dot{y}) + \ddot{y}_d, \end{cases} \tag{3.114}$$

where K_{p_x}, K_{p_y} are proportional gains for $\ddot{x}_d$ and $\ddot{y}_d$ respectively. K_{d_x},K_{d_y} are derivative gains for $\ddot{x}_d$ and $\ddot{y}_d$ respectively.

Let's denote the errors between the desired values and the real ones of x and y by e_x and e_y respectively. From (3.114) we get:

$$\begin{cases} \ddot{e}_x + K_{d_x}\dot{e}_x + K_{p_x}e_x = 0 \\ \ddot{e}_y + K_{d_y}\dot{e}_y + K_{p_y}e_y = 0. \end{cases} \tag{3.115}$$

We can write the characteristic polynomials of error equations in Laplace domain:

$$\begin{cases} P_x(p) = p^2 + K_{d_x}p + K_{p_x} \\ P_y(p) = p^2 + K_{d_y}p + K_{p_y}. \end{cases} \tag{3.116}$$

Now, the gains are chosen to ensure stability. For that we put the poles in the real negative half plane $p_1 = -4$ and $p_2 = -1$ we get $K_{d_x} = K_{d_y} = 5$ and $K_{p_x} = K_{p_y} = 4$.

Final thing is to replace (3.114) in (3.113):

$$\begin{cases} \phi_d &= -\frac{1}{g}(s(\psi)\ddot{x}_c - c(\psi)\ddot{y}_c) \\ \theta_d &= -\frac{1}{g}(c(\psi)\ddot{x}_c + s(\psi)\ddot{y}_c). \end{cases} \tag{3.117}$$

Here, all elements are calculated, we can show the block diagram of the Backstepping control complete structure for the quadcopter in Figure 3.6

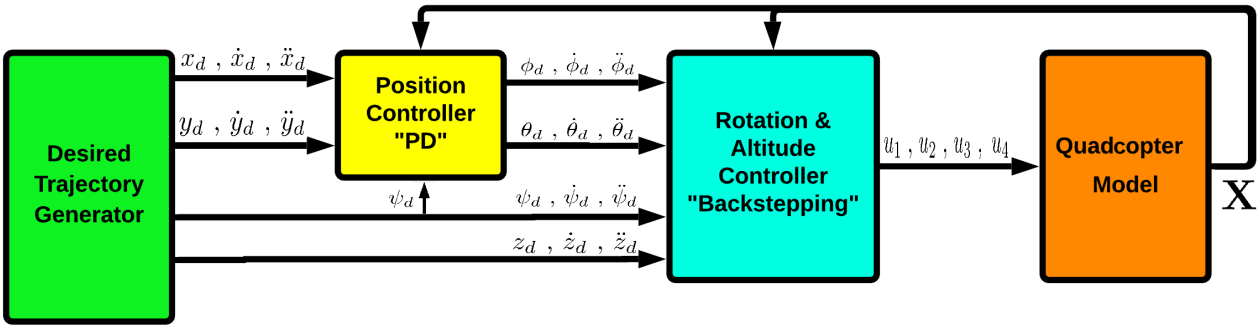


Figure 3.6: The complete structure of the quadcopter controlled by a BS technique

3.2.3 Simulation and Results

The simulation for the Backstepping controller had been made under the same conditions as the FBL, so a comparison can take place. The desired trajectory is also the helical one and the parameters belong to the OS4 quadcopter. The controllers parameters are : $\{\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \alpha_5 = \alpha_6 = 100\}$, $\{\alpha_7 = \alpha_8 = 1\}$, $\{K_{p_x} = K_{p_y} = 4\}$ and $\{K_{d_x} = K_{d_y} = 5\}$. The desired outputs are shown in Table 3.2:

Output	Desired Value	Initialisation
$x_d(t)$	$5sin((2\pi)0.1t)$	0 $[m]$
$y_d(t)$	$5(1 - cos(2\pi)0.1t))$	0 $[m]$
$z_d(t)$	-0.2t	0 $[m]$
$\psi_d(t)$	0	0 $[rad]$

Simulation Time	100 $[s]$

Table 3.2: Desired values for the helical trajectory

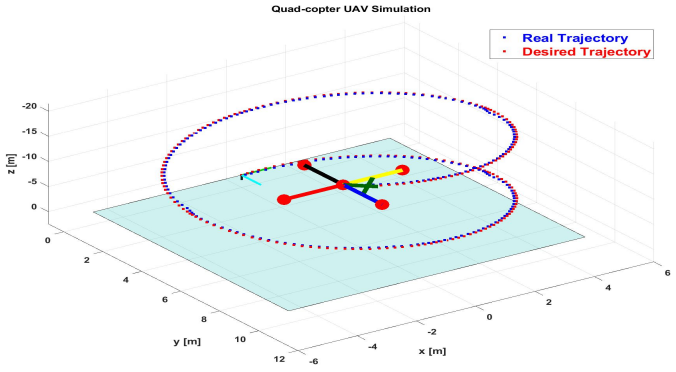
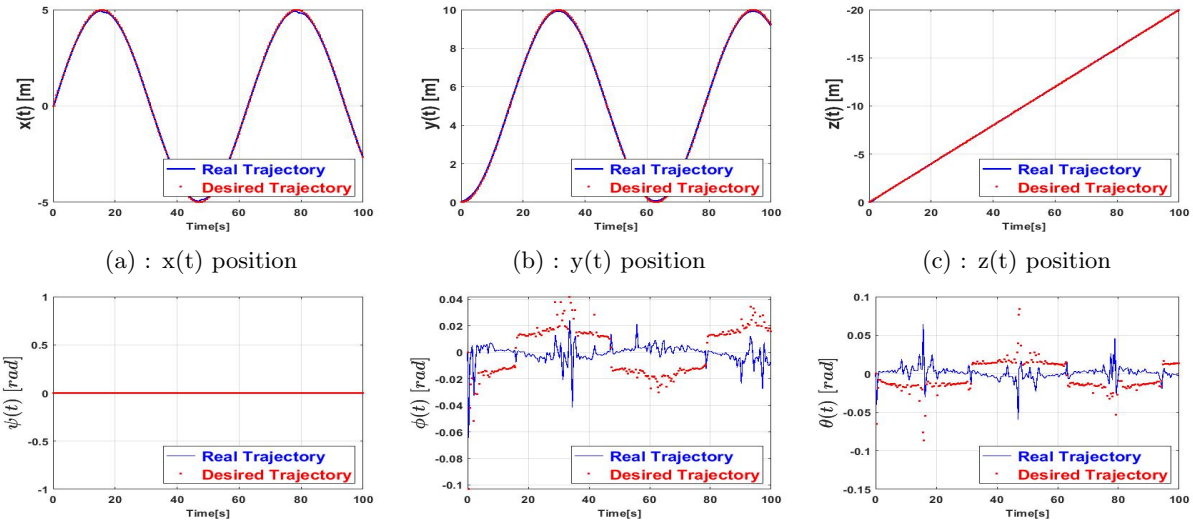


Figure 3.7: Real and desired trajectory with BS control



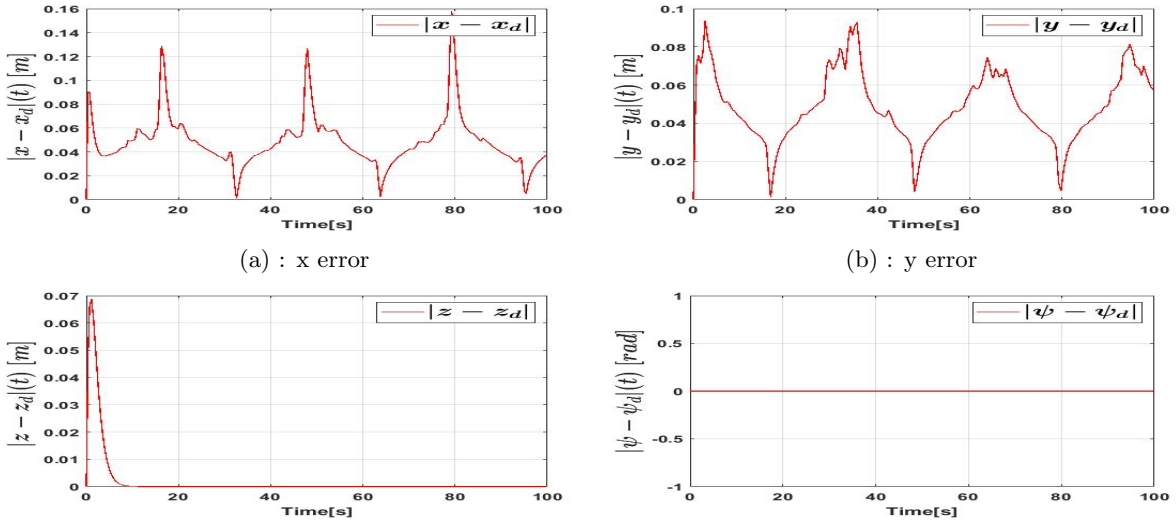
(a) : x(t) position (b) : y(t) position (c) : z(t) position

(d) : ψ(t) angle (e) : ϕ(t) angle (f) : θ(t) angle

Figure 3.8: Position and attitude throw time for BS

The Figure 3.8 shows the behaviour of the six degrees of freedom toward the desired values. We can see that the error for the states $\begin{bmatrix} x & y & z \end{bmatrix}^T$ is barely noticeable, and completely null for ψ. Also we remark that the angles ψ and θ controlled by the PD respect the boundaries of small oscillations. For more precise analysis we take a look into the error Figure 3.9. The order of error is less than 15 cm and null for the angle ψ, but unlike the FBL, the peaks are know located in the minimum values of the tangent of desired x and y due to the PD controller since at each peak of the desired x and y it has to make a change in the angles ϕ and θ which will create an error that will

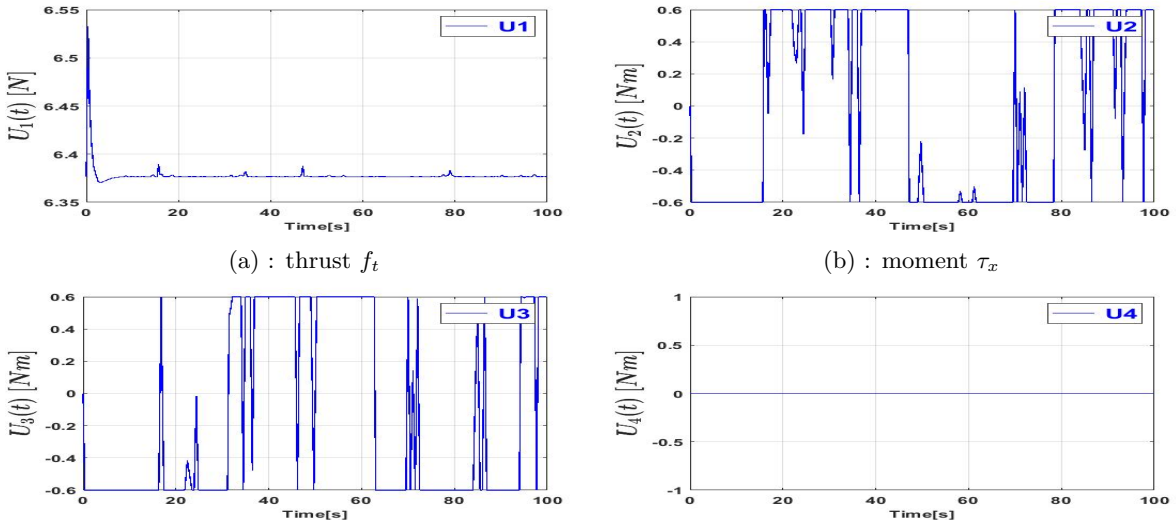
be compensated by the whole controllers.



(a) : x error (b) : y error

(c) : z error (d) : ψ error

Figure 3.9: Error between real and desired outputs



(a) : thrust f_t (b) : moment τ_x

(c) : moment τ_y (d) : moment τ_z

Figure 3.10: The outputs of BS controller

Finally, for the BS controller outputs (thrust and torques), we see that they are also realisable, but τ_x and τ_y reached the saturation multiple times. For better performances, more optimized choice of the controller parameters can lead to better results.

Chapter 4

Ground Effect Disturbance

The controllers designed earlier need to be tested under disturbances in order to verify robustness. For that, we choose to introduce the ground effect phenomena that appears when flying near the ground ($\approx$ at half rotor diameter[28]) and add more thrust caused by rotors airflow. The model is described in [40] based on a commonly used model described in [41] with slight adaptations to quadcopters.

4.1 Ground Effect

The simplified ground effect model is given by:

$$\frac{f_t}{f_t + f_{ge}} = 1 - \rho\left(\frac{R_{ad}}{4z}\right)^2, \tag{4.1}$$

where z is the altitude (we suppose the rotors have the same altitude as the CoG). R_{ad} is the radius of rotor. f_t is the thrust generated by the rotors. $(f_t + f_{ge})$ is the total thrust containing both the rotors force and the ground effect force noted f_{ge}. ρ is a correction coefficient to be determined.

The ground effect force formula is extracted as follows:

$$f_{ge} = f_t\left(\frac{1}{1 - \rho(\frac{R_{ad}}{4z})^2} - 1\right) \tag{4.2}$$

In order to avoid an infinite thrust, the ground effect is assumed to be null for a all values of z under a certain cut-off altitude which can be chosen as the landing gears height.

Before we add the ground effect to the quadcopter model, we will also give the gyroscopic effect formula that results from the unbalance of propellers rotation speed coupled with the body rotation:

$$\tau_g = \begin{bmatrix}\tau_{xg}\\ \tau_{yg}\\ \tau_{zg}\end{bmatrix} = \sum_{i=1}^{4} J_p(\begin{bmatrix}p\\ q\\ r\end{bmatrix} \times Q_z)(-1)^{i+1}\omega_i, \tag{4.3}$$

where J_p is the inertia of each rotor. ω_i is the angular speed of the i^{th} rotor (we can obtain each term by inverting (2.13)).

Considering the small oscillations assumption, the equation (4.3) becomes:

$$\tau_g = \begin{bmatrix} \tau_{xg} \\ \tau_{yg} \\ \tau_{zg} \end{bmatrix} == \sum_{i=1}^{4} J_p(\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times Q_z)(-1)^{i+1}\omega_i. \tag{4.4}$$

Now, we integrate the ground effect into our quadcopter model. Since it is a relative force generated in the mobile frame, Newton's equation (2.7) in fixed frame becomes:

$$\begin{cases} m\dot{\eta} = mgZ_n - f_t RQ_z - f_{ge} RQ_z \\ \\ m\dot{\eta} = mgZ_n - (f_t + f_{ge})RQ_z. \end{cases} \tag{4.5}$$

From (4.1), we extract the new total thrust, so it can be replaced in (4.5):

$$f_t + f_{ge} = \frac{f_t}{1 - \rho\left(\frac{R_{ad}}{4z}\right)^2}. \tag{4.6}$$

And for the gyroscopic effect, Euler's equation (2.12) becomes:

$$\begin{cases} I_x\ddot{\phi} + \dot{\theta}\dot{\psi}(I_z - I_y) & = \tau_x + J_p\dot{\theta}(\omega_1 - \omega_2 + \omega_3 - \omega_4) \\ I_y\ddot{\theta} + \dot{\phi}\dot{\psi}(I_x - I_z) & = \tau_y - J_p\dot{\phi}(\omega_1 - \omega_2 + \omega_3 - \omega_4) \\ I_z\ddot{\psi} + \dot{\phi}\dot{\theta}(I_y - I_x) & = \tau_z. \end{cases} \tag{4.7}$$

Notation. $\omega = (\omega_1 - \omega_2 + \omega_3 - \omega_4)$

The ground effect and gyroscopic effect are now implemented in our model:

$$\begin{cases} \dot{x}_1 & = x_7 \\ \dot{x}_2 & = x_8 \\ \dot{x}_3 & = x_9 \\ \dot{x}_4 & = x_{12} \\ \dot{x}_5 & = x_{11} \\ \dot{x}_6 & = x_{10} \\ \dot{x}_7 & = -\frac{1}{m}(c(x_6)c(x_4)s(x_5) + s(x_6)s(x_4))(1 - \rho(R_{ad}/(4z))^2)^{-1}u_1 \\ \dot{x}_8 & = -\frac{1}{m}(c(x_6)s(x_4)s(x_5) - c(x_4)s(x_6))(1 - \rho(R_{ad}/(4z))^2)^{-1}u_1 \\ \dot{x}_9 & = g - \frac{1}{m}(c(x_6)c(x_5))(1 - \rho(R_{ad}/(4z))^2)^{-1}u_1 \\ \dot{x}_{10} & = \frac{I_y - I_z}{I_x}x_{11}x_{12} + \frac{J_p}{I_x}x_4\omega + \frac{1}{I_x}u_2 \\ \dot{x}_{11} & = \frac{I_z - I_x}{I_y}x_{10}x_{12} - \frac{J_p}{I_y}x_2\omega + \frac{1}{I_y}u_3 \\ \dot{x}_{12} & = \frac{I_x - I_y}{I_z}x_{10}x_{11} + \frac{1}{I_z}u_4 \end{cases} \tag{4.8}$$

The controllers presented earlier are not designed to overcome the ground effect disturbances, so we can observe how far they can be robust against it.

The ground effect is significant when the altitude is too low. The next section gives a trajectory planning for taking off and landing so the ground effect can be observed.

4.2 Trajectory Planning

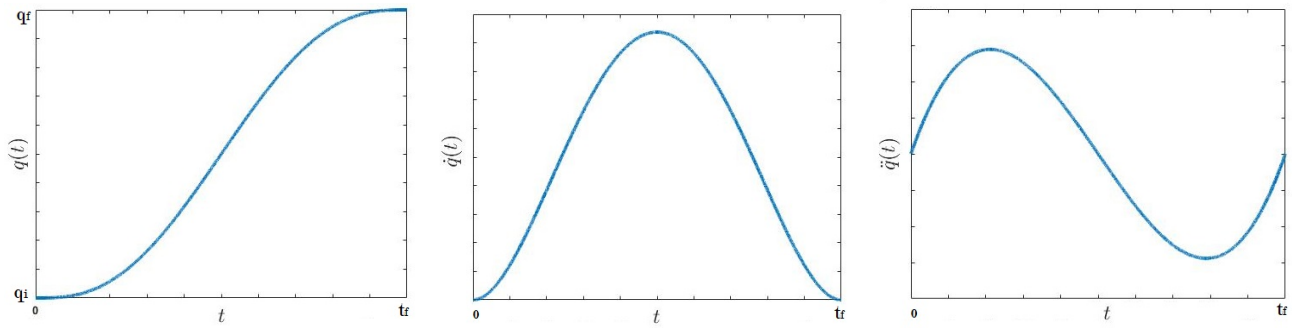
In order to have a smooth trajectory tracking and prevent the controller to produce high and aggressive signals, a trajectory planning is needed. This later will result the mathematical expressions of the desired outputs $\begin{bmatrix} x & y & z & \psi \end{bmatrix}^T$.

The trajectory is planned to go from a point to another (ex: taking-off and landing) by specifying the departure and arrival points and the duration between the two. We choose a 5 order polynomial formula:

$$\begin{cases} q(t) = q_i + \left(\frac{10\Delta q}{t_f^3}\right).t^3 - \left(\frac{15\Delta q}{t_f^4}\right).t^4 + \left(\frac{6\Delta q}{t_f^5}\right).t^5 \\ \dot{q}(t) = + \left(\frac{30\Delta q}{t_f^3}\right).t^2 - \left(\frac{60\Delta q}{t_f^4}\right).t^3 + \left(\frac{30\Delta q}{t_f^5}\right).t^4 \\ \ddot{q}(t) = + \left(\frac{60\Delta q}{t_f^3}\right).t - \left(\frac{180\Delta q}{t_f^4}\right).t^2 + \left(\frac{120\Delta q}{t_f^5}\right).t^3 \end{cases} \tag{4.9}$$

where $q(t)$ is the desired trajectory (position). $\dot{q}(t)$ is the desired trajectory first derivative (velocity). $\ddot{q}(t)$ is the desired trajectory second derivative (acceleration). q_i is the initial position. Δq is the difference between initial position q_i and final position q_f ($\Delta q = q_f - q_i$). t_f is the desired duration between q_i and q_f.

The shape of the trajectory is shown in Figure 4.1:



(a) desired position (b) desired velocity (c) desired acceleration

Figure 4.1: Desired trajectory shape

Remark. *The equations (4.9) tend to infinity when t tends to infinity. In order to stay on q_f after t_f, a conditions must be added to the desired trajectory. (ex : if $(t > t_f)$ then $(q(t) = q_f$ and $\dot{q}(t) = 0$ and $\ddot{q}(t) = 0))$.*

4.3 Simulation and Results

In this section, we will show the results of simulations obtained by Matlab/Simulink for both controllers BS and FBL under the unmodeled disturbances: ground effect and gyroscopic effect. The trajectory is described in Table 4.1.The ground effect parameter is $\rho = 1$. The cut-off altitude for the ground effect is $Z_{cutoff} = -0.05$ so when $z > -0.05$ the ground effect is null. The saturations on the controller outputs are $0 < u_1 < 2mg$ and $|\{u_2, u_3, u_4\}| < 0.6$.

Output	q_i	q_f	Δt	Initialisation
$x_d(t)$	0	5	$0 < t < 25$	0 $[m]$
	5	5	$25 \leq t \leq 120$	
$y_d(t)$	0	5	$0 < t < 25$	0 $[m]$
	5	5	$25 \leq t \leq 120$	
$z_d(t)$	-0.1	-3	$0 < t < 50$	0 $[m]$
	-3	-0.1	$50 \leq t < 100$	
	-0.1	-0.1	$100 \leq t \leq 120$	
$\psi_d(t)$	0	15	$0 < t < 100$	0 $[rad]$
	15	15	$100 \leq t \leq 120$	

Simulation Time	120 $[s]$

Table 4.1: Desired values for the helical trajectory

Backstepping simulation

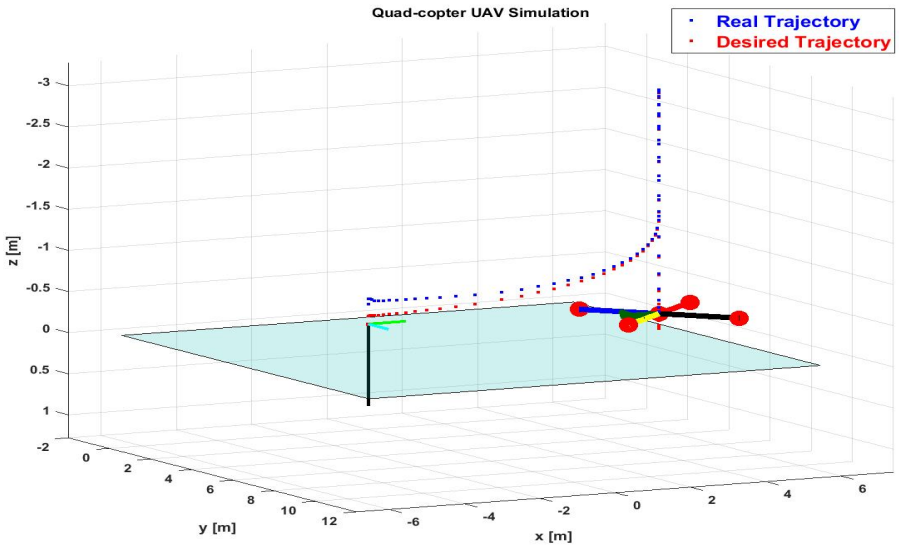


Figure 4.2: Real and desired trajectory with BS control (Ground Effect + Gyroscopic Effect)

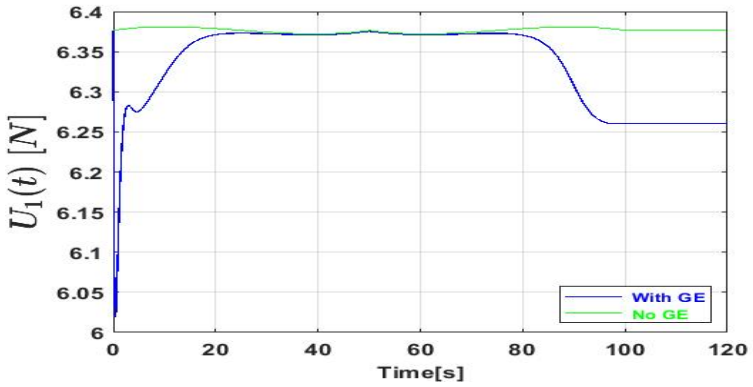
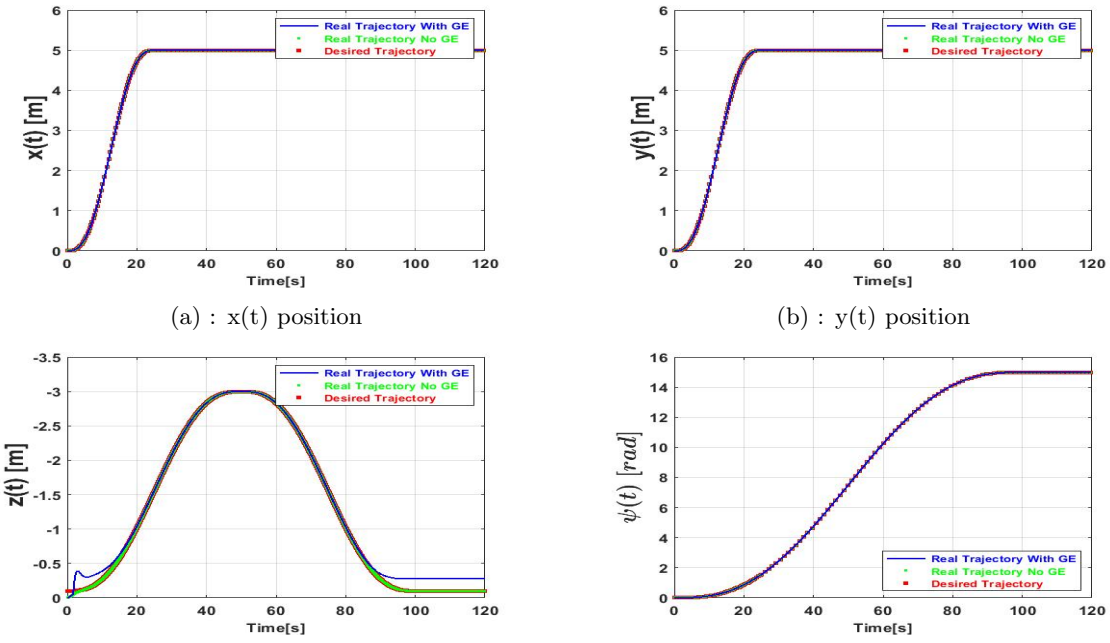


Figure 4.3: Control u_1

From Figure 4.4 we see that the BS controller managed to follow the desired trajectory and keep a stable performance even when the ground effect and gyroscopic effect aren't modeled in its formula. A static error is noticeable when taking off and landing due to ground effect that can be overcome by giving bigger α_1 and α_2 since the ones chosen for this simulation are small. With regard the gyroscopic disturbance, its effect is barely noticeable since J_p is too small (It is more significant in the small quadcopters realisations) so we can conclude that it can be neglected.

For the control in Figure 4.3, it remains realisable and the effect of ground on the controller is clear when the



(a) : x(t) position

(b) : y(t) position

(c) : z(t) position

(d) : $\psi(t)$ angle

Figure 4.4: Position and heading outputs for BS controller

quadcopter is close to land.

Overall, the Backstepping showed robustness against two unmodeled disturbances. Some other simulations took the existence of noise on different parts (1% on actuators, 1% on model, 0.1% on sensors). The BS controller showed also robustness against them.

Feedback Linearization simulation

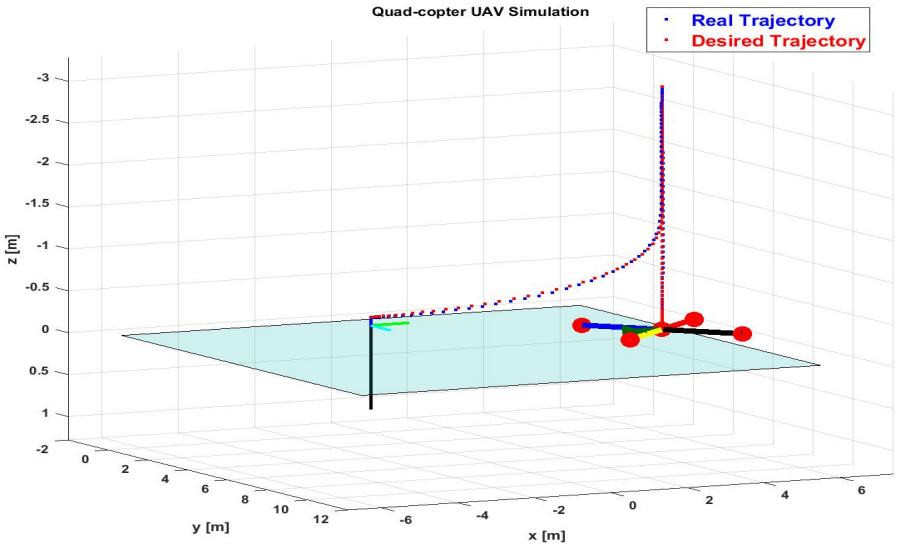


Figure 4.5: Real and desired trajectory with FBL control (Ground Effect + Gyroscopic Effect)

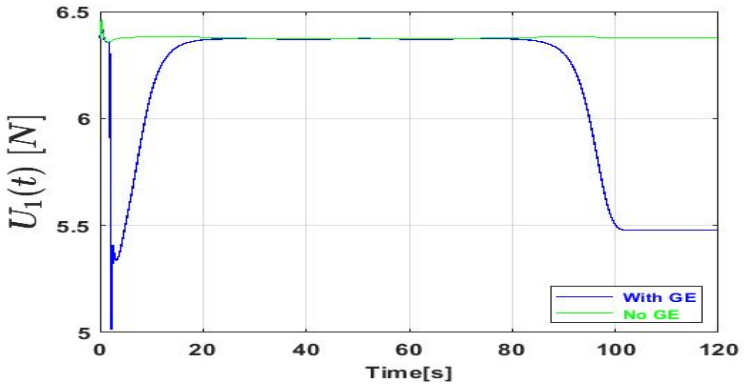
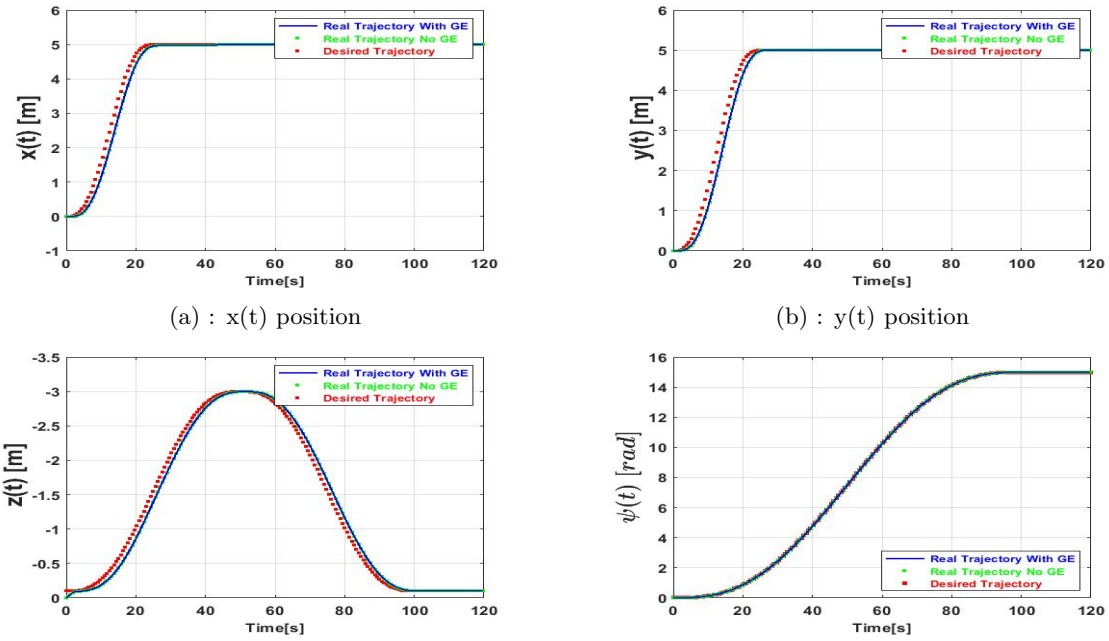


Figure 4.6: Control u_1

The FBL controller also could follow the trajectory and showed robustness against both gyro effect (which is negligible as we saw previously) and the ground effect. In landing phase were all states are constant we also notice that the controller could even cancel the ground effect.

But when a white noise had been added into the simulation, an instability appeared even for smaller values compared to those applied on BS controller. This is because the FBL can linearize the system only in neighborhood of the operation point, so the presence of uncertainties will makes the nonlinearities appear again and the performances



(a) : x(t) position

(b) : y(t) position

(c) : z(t) position

(d) : $\psi(t)$ angle

Figure 4.7: Position and heading outputs for FBL controller

degrade remarkably. We may therefore conclude that the FBL controller, with the above specified parameters, is less robust to noise than the BS one.

Comparison

Now, we have seen the behaviour of quadcopter with both controllers BS and FBL for different trajectories and disturbances. We can summarize the results as follows:

- In the absence of disturbances both controllers managed to insure a stable behaviour and tracking the desired trajectories. The FBL showed a larger margin of error specially when the rate of change in the desired outputs is big (FBL is sensitive to trajectory variations). On the other hand the BS had much less error and also gave very good yaw control.
- In the presence of disturbances (mainly the ground effect) and absence of noise, both controllers showed robustness against them. During the landing phase, the FBL achieved a very low error.
- But when a white noise was added (on actuators, sensors and model), only the BS maintained the robustness, stability and performances while the FBL showed a significant degradation in stability and performances for much smaller values of noise compared to BS.
- Both controllers have the disadvantage of the parameters' number needed to be tuned and also the requirement of many states information.

We conclude that the BS technique showed much more advantages for controlling a quadcopter compared to the FBL technique. It showed better stability, performances and specially better robustness toward unmodeled uncertainties. However, it still possible to expect better performances in case a feedforward control that estimates those disturbances is added.

Chapter 5

Conclusion

This work presented a controller design for an UAV type quadcopter. A mathematical model was developed and tested in open loop simulation, then two control techniques were designed: the Backstepping and the Feedback Linearization to ensure desired performances and robustness. So theoretical and practical aspects were presented and finally simulations took place to verify the controllers and their robustness against the ground effect and noise. We concluded that the Backstepping gave promising results specially the robustness against unmodeled disturbances that will nominate it to be used for future quadcopters works.

Those future works will focus principally on two topics: the first one is the design of a feedforward control that takes corrective actions to compensate for the disturbances by estimating it and adding sensors to measure it. This has the interest in case the disturbances are considerably strong. The second topic is to develop the controller so it can handle more aggressive flights since the one designed in the present work is valid only for small oscillations motions.

The results will contribute in the first work package of "Delicio" project (Figure 1.2) as mentioned earlier and also in some applications on real quadcopters and other robotic projects.

Bibliography

[1] Bartomeu Rubí, Ramon Pérez, and Bernardo Morcego. A survey of path following control strategies for uavs focused on quadrotors. *Journal of Intelligent & Robotic Systems*, 98(2):241–265, May 2020.

[2] Guillaume Charland-Arcand. Contrôle non linéaire par backstepping d'un hélicoptère de type quadrotor pour des applications autonomes. *Mémoire de maîtrise électronique, Montréal, École de technologie supérieure*, 2014.

[3] K.P. Valavanis. *Advances in Unmanned Aerial Vehicles: State of the Art and the Road to Autonomy*. Intelligent Systems, Control and Automation: Science and Engineering. Springer Netherlands, 2008.

[4] H. Kim and D. Shim. A flight control system for aerial robots: algorithms and experiments. 2003.

[5] S. Bouabdallah, A. Noth, and R. Siegwart. Pid vs lq control techniques applied to an indoor micro quadrotor. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 3, pages 2451–2456 vol.3, 2004.

[6] Gabriel Hoffmann, Haomiao Huang, Steven Waslander, and Claire Tomlin. *Quadrotor Helicopter Flight Dynamics and Control: Theory and Experiment*. Proc. of the AIAA Guidance, Navigation, and control Conf. and Exhibit, 2007.

[7] Matko Orsag, Marina Poropat, and Stjepan Bogdan. Hybrid fly-by-wire quadrotor controller. In *2010 IEEE International Symposium on Industrial Electronics*, pages 202–207, 2010.

[8] Jonathan P. How, Brett Behihke, Adrian Frank, Daniel Dale, and John Vian. Real-time indoor autonomous vehicle test environment. *IEEE Control Systems Magazine*, 28(2):51–64, 2008.

[9] Guilherme V. Raffo, Manuel G. Ortega, and Francisco R. Rubio. An integral predictive/nonlinear h_∞ control structure for a quadrotor helicopter. *Automatica*, 46(1):29–39, 2010.

[10] Ashfaq Ahmad Mian and Wang Daobo. Modeling and backstepping-based nonlinear control strategy for a 6 dof quadrotor helicopter. *Chinese Journal of Aeronautics*, 21(3):261–268, 2008.

[11] Tarek Madani and Abdelaziz Benallegue. Backstepping control for a quadrotor helicopter. pages 3255 – 3260, 11 2006.

[12] A. Benallegue, A. Mokhtari, and L. Fridman. High-order sliding-mode observer for a quadrotor uav. *International Journal of Robust and Nonlinear Control*, 18(4-5):427–440, 2008.

[13] Abhijit Das, K. Subbarao, and F. Lewis. Dynamic inversion with zero-dynamics stabilisation for quadrotor control. *Iet Control Theory and Applications*, 3:303–314, 2009.

[14] A. Mokhtari, A. Benallegue, and B. Daachi. Robust feedback linearization and GH_∞ controller for a quadrotor unmanned aerial vehicle. *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1198–1203, 2005.

[15] Kostas Alexis, George Nikolakopoulos, and Anthony Tzes. Switching model predictive attitude control for a quadrotor helicopter subject to atmospheric disturbances. *Control Engineering Practice*, 19(10):1195–1207, 2011.

[16] Iman Sadeghzadeh, Mahyar Abdolhosseini, and Youmin M. Zhang. Payload drop application of unmanned quadrotor helicopter using gain-scheduled pid and model predictive control techniques. In Chun-Yi Su, Subhash Rakheja, and Honghai Liu, editors, *Intelligent Robotics and Applications*, pages 386–395, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[17] M Önder Efe. Neural network assisted computationally simple $\mathrm{pi}^{\lambda}\mathrm{d}^{\mu}$ control of a quadrotor uav. *IEEE Transactions on Industrial Informatics*, 7(2):354–361, 2011.

[18] Pedro Ponce, Arturo Molina, Israel Cayetano, Jose Gallardo, Hugo Salcedo, and Jose Rodriguez. Experimental fuzzy logic controller type 2 for a quadrotor optimized by anfis. *IFAC-PapersOnLine*, 48:2435–2441, 12 2015.

[19] A. Azzam and Xinhua Wang. Quad rotor arial robot dynamic modeling and configuration stabilization. *2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR 2010)*, 1:438–444, 2010.

[20] Amr Nagaty, G. SajadSaeedi, Carl Thibault, M. Seto, and Howard Li. Control and navigation framework for quadrotor helicopters. *Journal of Intelligent & Robotic Systems*, 70:1–12, 2013.

[21] Ali Kissai and Milton Smith. Electronic navigation system based on the use of alternate coordinate system and polar stereographic projection for uavs operating in polar regions. *Int J Aeronaut Aerosp Eng*, 1(2):46–53, 2019.

[22] Hector Garcia de Marina. *Distributed formation control for autonomous robots*. PhD thesis, university of groningen, 06 2016.

[23] De Xin Xu, Yan Hui Wei, and Kun Peng He. Trajectory tracking control of a quad-rotor uav. In *Mechanics and Mechatronics*, volume 419 of *Applied Mechanics and Materials*, pages 718–724. Trans Tech Publications Ltd, 12 2013.

[24] Jian S. Dai. Euler–rodrigues formula variations, quaternion conjugation and intrinsic connections. *Mechanism and Machine Theory*, 92:144–152, 2015.

[25] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: Modelling, Planning and Control*. Springer Publishing Company, Incorporated, 2nd edition, 2009.

[26] Tarek Madani and Abdelaziz Benallegue. Backstepping control for a quadrotor helicopter. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3255–3260, 2006.

[27] Tommaso Bresciani. *Modelling, Identification and Control of a Quadrotor Helicopter*. Department of Automatic Control, Lund University, 2008.

[28] Bouabdallah Samir. *Design And Control Of Quadrotors With Application To Autonomous Flying*. PhD thesis, Ecole polytechnique fédérale de Lausanne, 2007.

[29] Bekhiti Belkacem, Dahimene Abdelhakim, and Harriche Kamel. *Intelligent Guidance and Control of Quadrotor Unmanned Aerial Vehicl*. LAMBERT Academic Publishing, December 2017.

[30] H.K. Khalil. *Nonlinear Systems*. Pearson Education. Prentice Hall, 2002.

[31] M. Vidyasagar. *Nonlinear Systems Analysis: Second Edition*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 2002.

[32] A. Isidori. *Nonlinear Control Systems*. Communications and Control Engineering. Springer London, 1995.

[33] Ranjan Vepa. *Nonlinear Control of Robots and Unmanned Aerial Vehicles: An Integrated Approach*. CRC Press, 08 2016.

[34] Michael A. Henson and Dale E. Seborg. *Feedback Linearizing Control*, page 149–231. Prentice-Hall, Inc., USA, 1997.

[35] Henk Nijmeijer and Arjan van der Schaft. *Nonlinear dynamical control systems*. Springer, 1990.

[36] Luenberger David G. *Introduction to Dynamic Systems: Theory, Models, and Applications*. Wiley, 1979.

[37] Guillaume Charland-Arcand. *Contrôle non linéaire par backstepping d'un hélicoptère de type quadrotor pour des applications autonomes*. Montréal, École de technologie supérieure, 2014.

[38] Miroslav Krstic, Kanellakopoulos Ioannis, and Petar V. Kokotović. *Nonlinear and Adaptive Control Design*. A Wiley-Interscience publication. Wiley, 1995.

[39] R. Sepulchre, M. Janković, and P.V. Kokotović. *Constructive Nonlinear Control*. Communications and Control Engineering. Springer London, 1997.

[40] Li Danjun, Zhou Yan, Shi Zongying, and Lu Geng. Autonomous landing of quadrotor based on ground effect modelling. In *2015 34th Chinese Control Conference (CCC)*, pages 5647–5652, July 2015.

[41] I. C. Cheeseman, Ph. D, and W. E. Bennett. The effect of ground on a helicopter rotor in forward flight. Technical report, Aeronautical Research Council Reports And Memoranda, 1957.