Quadrotor control: modeling, nonlinear control design, and simulation

FRANCESCO SABATINO

KTH Electrical Engineering

Master's Degree Project
Stockholm, Sweden June 2015

XR-EE-RT 2015:XXX

Abstract

In this work, a mathematical model of a quadrotor's dynamics is derived, using Newton's and Euler's laws. A linearized version of the model is obtained, and therefore a linear controller, the Linear Quadratic Regulator, is derived. After that, two feedback linearization control schemes are designed. The first one is the dynamic inversion with zero dynamics stabilization, based on Static Feedback Linearization obtaining a partial linearization of the mathematical model. The second one is the exact linearization and non-interacting control via dynamic feedback, based on Dynamic Feedback Linearization obtaining a total linearization of the mathematical model. Moreover, these nonlinear control strategies are compared with the Linear Quadratic Regulator in terms of performances. Finally, the behavior of the quadrotor under the proposed control strategies is observed in virtual reality by using the Simulink 3D Animation toolbox.

Contents

Chapter 1

Introduction

Recent advances in sensor, in microcomputer technology, in control and in aerodynamics theory have made small Unmanned Aerial Vehicles (sUAV) a reality. The small size, low cost and maneuverability of these systems have made them potential solutions in a large class of applications. However, the small size of these vehicles poses significant challenges. The small sensors used on these systems are much noisier than their larger counterparts. The compact structure of these vehicles also makes them more vulnerable to environmental effects. In this work, control strategies for an sUAV platform are developed. Simulation studies and experimental results are provided.

The fundamental problem with the safe operation of vehicles with wingspan smaller than one meter is reliable stabilization, robustness to unpredictable changes in the environment, and resilience to noisy data from small sensor systems. Autonomous operation of aerial vehicles relies upon on-board stabilization and trajectory tracking capabilities, and significant effort has to be carried on to make sure that these systems are able to achieve stable flight. These problems are compounded at smaller scale, since as the vehicle is more susceptible to environmental effects (wind, temperature, etc.). Moreover, the small scale implies that lower quality and noisier compact Micro-Electro-Mechanical Systems (MEMS) sensor are used as primary sensor. The small scale also makes it harder for the MEMS sensors to be isolated from the vibrations that are common in these flight platforms.

Since the number and complexity of applications for such systems grows daily, the control techniques involved must also improve in order to provide better performance and increased versatility. Historically, simplistic linear control techniques were employed for computational ease and stable hover flight. However, with better modelling techniques and faster on board computational capabilities, comprehensive nonlinear techniques to be run on real-time have become an achievable goal. Nonlinear methodologies promise to rapidly increase the performances for these systems and make them more robust. This work presents several approaches to the automatic control of a quadrotor. Selected linear and nonlinear control methods are designed according to the system dynamics.

1.1 Literature review

1.1.1 Quadrotor history

Etienne Oehmichen was the first scientist who experimented with rotorcraft designs in 1920 [1]. Among the six designs he tried, his second multicopter had four rotors and eight propellers, all driven by a single engine. The Oehmichen used a steel-tube frame, with two-bladed rotors at the ends of the four arms. The angle of these blades could be varied by warping. Five of the propellers, spinning in the horizontal plane, stabilized the machine laterally. Another propeller was mounted at the nose for steering. The remaining pair of propellers was for forward propulsion. The aircraft exhibited a considerable degree of stability and controllability for its time, and made more than a thousand test flights during the middle 1920. By 1923 it was able to remain airborne for several minutes at a time, and on April 14, 1924 it established the first-ever Fédération Aéronautique Internationale (FAI) [2] distance record for helicopters of 360 m. Later, it completed the first 1 kilometer closed-circuit flight by a rotorcraft.

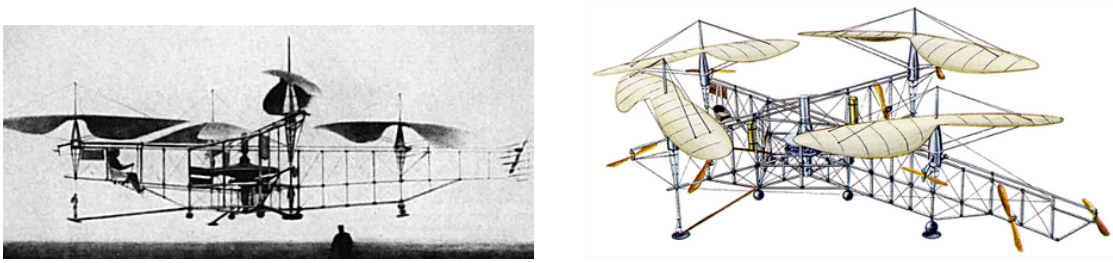


Figure 1.1: Oehmichen No.2 Quadrotor [1].

After Oehmichen, Dr. George de Bothezat and Ivan Jerome developed this aircraft [1], with six bladed rotors at the end of an X-shaped structure. Two small propellers with variable pitch were used for thrust and yaw control. The vehicle used collective pitch control. It made its first flight in October 1922. About 100 flights were made by the end of 1923. The highest it ever reached was about 5 m. Although demonstrating feasibility, it was underpowered, unresponsive, mechanically complex and susceptible to reliability problems. Pilot workload was too high during hover to attempt lateral motion.

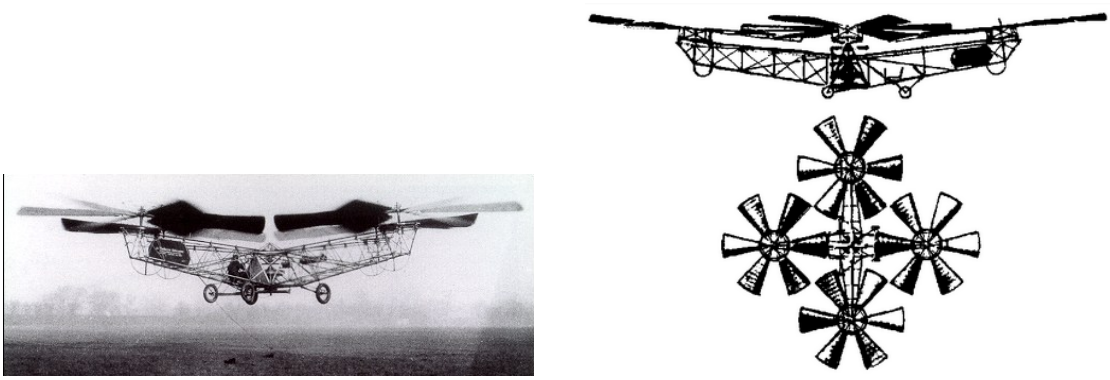


Figure 1.2: The de Bothezat Quadrotor [1].

Convertawings Model A Quadrotor (1956) was intended to be the prototype for a line of much larger civil and military quadrotor helicopters [1]. The design featured two engines driving four rotors with wings added for additional lift in forward flight. No tail rotor was needed and control was obtained by varying the thrust between rotors. Flown successfully many times in the mid-1950s, this helicopter proved the quadrotor design and it was also the first four-rotor helicopter to demonstrate successful forward flight. However, due to the lack of orders for commercial or military versions however, the project was terminated. Convertawings proposed a Model E that would have a maximum weight of 19,000 kg with a payload of 4,900 kg.

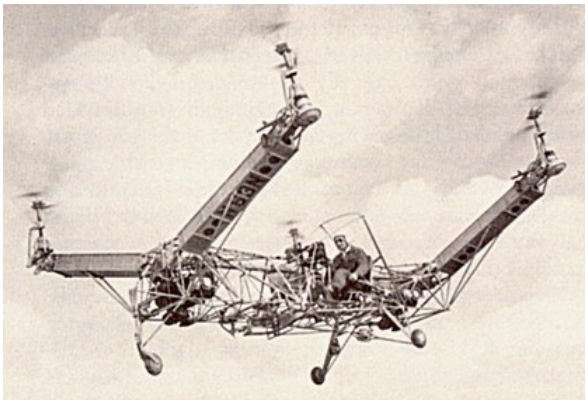


Figure 1.3: Model A quadrotor [1].

1.1.2 Control

Numerous control methods have been proposed for quadrotors, for both regulation and trajectory tracking. The goal is to find a control strategy that allows the states of a quadrotor to converge to an arbitrary set of time-varying reference states. Many previous works [3, 4, 5, 6] have demonstrated that it is possible to control the quadrotor using linear control techniques by linearizing the dynamics around an operating point, usually chosen to be the hover. However, a wider flight envelope and better performances can be achieved by using nonlinear control techniques that consider a more general form of the dynamics of the vehicle in all flight zones.
Within these nonlinear methods, backstepping [7, 8], sliding mode [9, 10] and feedback linearization [11] have been demonstrated to be effective for quadrotor control. Particularly, feedback linearization has shown significant promise for quadrotor vehicles. A recent work [11] suggests an feedback linearization structure that deconstructs the quadrotor dynamics into an inner loop containing the attitude and height of the vehicle and an outer loop containing the position. Another feedback linearization allows to get a linear model by Dynamic Feedback Linearization technique. In general, these control structures shows significant promise and is investigated in this work along with linear methods such as LQR.
All the control techniques suggested above require complete knowledge of the system model and model parameters, but errors in the identified values of the parameters can lead to significant deterioration of the controller performance. Furthermore, unmodeled variations in system parameters (such as mass or inertia) during flight can cause significant stabilization errors to occur. The need for an accurate nonlinear model of quadrotor dynamics can be overcome by using adaptive methods that can react to and correct errors in model parameter estimates, modify parameter estimates when they change and also adjust for external disturbances. Linear adaptive methods such as Model Reference Adaptive Control (MRAC) have been suggested [12]. However, as for most linear methods, the achievable trajectory of the quadrotor is restricted due to the assumption of linearization. The work of Huang et al. [13] suggests an adaptive backstepping method, and this approach was extended to include inertia parameters in the adaptation law by Zeng et al. [14]. Recent work in autonomous grasping and construction using quadrotors also use indirect adaptive methods, such as the least-squares method (for mass) proposed by Kumar et al. [15]. However, all indirect methods correct parameter errors based on the difference between the expected and actual plant outputs, but do not explicitly correct the model parameters (as done by direct adaptive methods). Direct adaptation methods were first suggested by Craig et al. [16] for mechanical manipulators.
In this work, I study two feedback linearization, *dynamic inversion with zero dynamics stabilization* and *exact linearization and non-interacting control via dynamic feedback*. They are compared with a linear control technique: *Linear Quadratic Regulator*.

1.2 Outline

The rest of this thesis is organized as follows.

Chapter 2. In this chapter a qualitative introduction on the principles of working of a quadrotor is discussed. Then the mathematical model is presented and a linearized version of the model is obtained.

Chapter 3. In this chapter a linear controller, the Linear Quadratic Regulator, is derived and two feedback linearization control schemes are designed. The first one is the dynamic inversion with zero dynamics stabilization, based on Static Feedback Linearization obtaining a partial linearization of the mathematical model. The second one is the exact linearization and non-interacting control via dynamic feedback, based on Dynamic Feedback Linearization obtaining a total linearization of the mathematical model.

Chapter 4. In this chapter results obtained with the different controllers are illustrated, and the differences are discussed. Finally, the behavior of the quadrotor under the proposed control strategies is observed in virtual reality by using the Simulink 3D Animation toolbox.

Chapter 5. Finally, in this chapter, conclusions and possible future developments of the work are presented.

Chapter 2

Mathematical model

2.1 Preliminar notions

The quadrotor, an aircraft made up of four engines, holds the electronic board in the middle and the engines at four extremities. Before describing the mathematical model of a quadrotor, it is necessary to introduce the reference coordinates in which we describe the structure and the position. For the quadrotor, it is possible to use two reference systems. The first is fixed and the second is mobile. The fixed coordinate system, called also inertial, is a system where the first Newton's law is considered valid. As fixed coordinate system, we use the O_{NED} systems, where NED is for *North-East-Down*. As we can observe from the following Figure 2.1, its vectors are directed to Nord, East and to the center of the Earth.

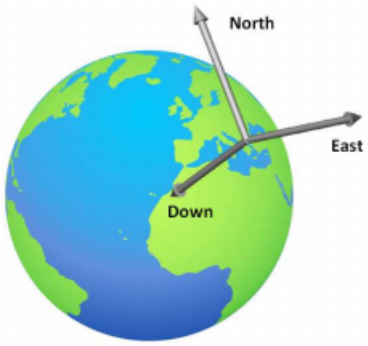


Figure 2.1: O_{NED} fixed reference system.

The mobile reference system that we have previously mentioned is united with the barycenter of the quadrotor. In the scientific literature it is called O_{ABC}

system, where ABC is for *Aircraft Body Center*. Figure 2.2 illustrates underlines the two coordinate systems.

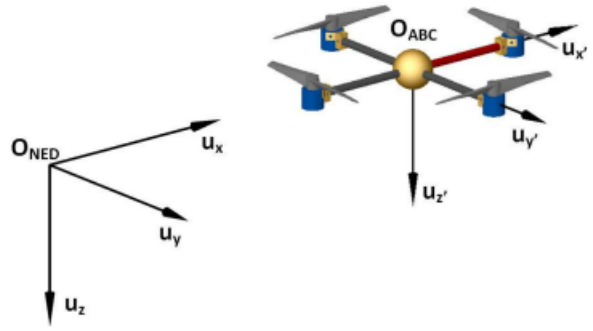


Figure 2.2: Mobile reference system and fixed reference system.

The attitude and position of the quadrotor can be controlled to desired values by changing the speeds of the four motors. The following forces and moments can be performed on the quadrotor: the thrust caused by rotors rotation, the pitching moment and rolling moment caused by the difference of four rotors thrust, the gravity, the gyroscopic effect, and the yawing moment. The gyroscopic effect only appears in the lightweight construction quadrotor. The yawing moment is caused by the unbalanced of the four rotors rotational speeds. The yawing moment can be cancelled out when two rotors rotate in the opposite direction. So, the propellers are divided in two groups. In each group there are two diametrically opposite motors that we can easily observe thanks to their direction of rotation. Namely, we distinguish:

- front and rear propellers (numbers 2 and 4 in Figure 2.3), rotating counterclockwise;
- right and left propellers (numbers 1 and 3 in Figure 2.3), rotating clockwise.

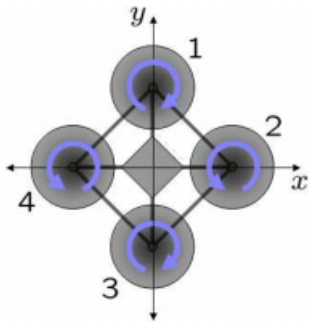


Figure 2.3: Direction of propeller's rotations.

The space motion of the rigid body aircraft can be divided into two parts: the barycenter movement and movement around the barycenter. Six degrees of freedom are required in describing any time space motion. They are three barycenter movements and three angular motions, namely, three translation and three rotation motions along three axes. The control for six degrees of freedom motions can be implemented by adjusting the rotational speeds of different motors. The motions include forward and backward movements, lateral movement, vertical motion, roll motion, and pitch and yaw motions. The yaw motion of the quadrotor can be realised by a reactive torque produced by the rotor. The size of the reactive torque is relative to the rotor speed. When the four rotor speeds are the same, the reactive torques will balance each other and quadrotor will not rotates, whereas if the four rotor speeds are not absolutely same, the reactive torques will not be balanced, and the quadrotor will start to rotate. When the four rotor speeds synchronously increase and decrease is also required in the vertical movement. Because of four inputs and six outputs in a quadrotor, such quadrotor is considered an underactuated nonlinear complex system. In order to control it, some assumptions are made in the process of quadrotor modeling: the quadrotor is a rigid body; the structure is symmetric; the ground effect is ignored.
Depending on the speed rotation of each propeller it is possible to identify the four basic movements of the quadrotor, which are showed in Figures 2.4 to 2.7.

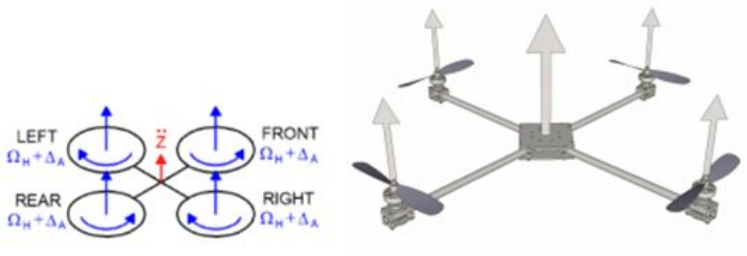


Figure 2.4: Thrust.

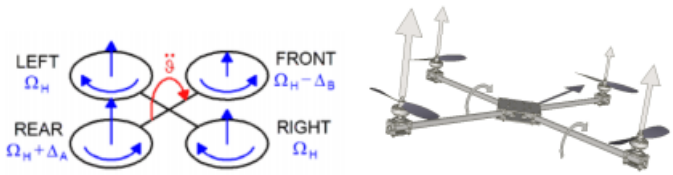


Figure 2.5: Pitch.

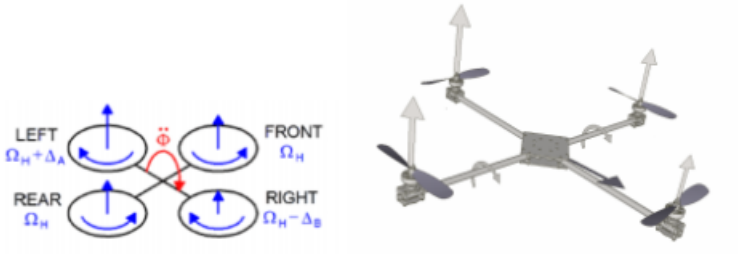


Figure 2.6: Roll.

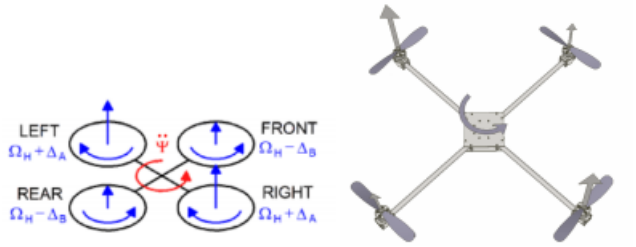


Figure 2.7: Yaw.

2.2 Euler angles

The Euler angles are three angles introduced by Leonhard Euler to describe the orientation of a rigid body. To describe such an orientation in the 3-dimensional Euclidean space, three parameters are required. They can be given in several ways; we will use ZYX Euler angles [26]. They are also used to describe the orientation of a frame of reference relative to another and they transform the coordinates of a point in a reference frame in the coordinates of the same point in another reference frame. The Euler angles are typically denoted as $\phi \in]-\pi, \pi]$, $\theta \in]\frac{\pi}{2}, \frac{\pi}{2}[$, $\psi \in]-\pi, \pi]$. Euler angles represent a sequence of three elemental rotations, i.e. rotations about the axes of a coordinate system, since any orientation can be achieved by composing three elemental rotations. These rotations start from a known standard orientation. This combination used is described by the following rotation matrices [27]:

$$\mathbf{R}_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\phi) & -s(\phi) \\ 0 & s(\phi) & c(\phi) \end{bmatrix}, \tag{2.1}$$

$$\mathbf{R}_y(\theta) = \begin{bmatrix} c(\theta) & 0 & s(\theta) \\ 0 & 1 & 0 \\ -s(\theta) & 0 & c(\theta) \end{bmatrix}, \tag{2.2}$$

$$\mathbf{R}_z(\psi) = \begin{bmatrix} c(\psi) & -s(\psi) & 0 \\ s(\psi) & c(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.3}$$

where $c(\phi) = \cos(\phi)$, $s(\phi) = \sin(\phi)$, $c(\theta) = \cos(\theta)$, $s(\theta) = \sin(\theta)$, $c(\psi) = \cos(\psi)$, $s(\psi) = \sin(\psi)$. So, the inertial position coordinates and the body reference coordinates are related by the rotation matrix $\mathbf{R}_{zyx}(\phi, \theta, \psi) \in SO(3)$:

$$\mathbf{R}_{zyx}(\phi, \theta, \psi) = \mathbf{R}_z(\psi) \cdot \mathbf{R}_y(\theta) \cdot \mathbf{R}_x(\phi)$$

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

This matrix describe the rotation from the body reference system to the inertial reference.

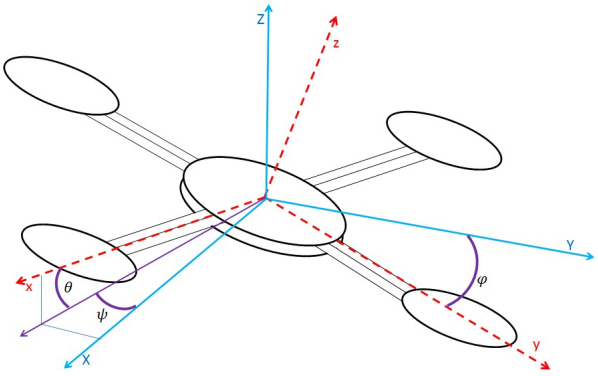


Figure 2.8: Euler Angles.

2.3 Quadrotor mathematical model

We provide here a mathematical model of the quadrotor, exploiting Newton and Euler equations for the 3D motion of a rigid body. The goal of this section is to obtain a deeper understanding of the dynamics of the quadrotor and to provide a model that is sufficiently reliable for simulating and controlling its behavior. Let us call $\begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^T$ the vector containing the linear and angular position of the quadrotor in the earth frame and $\begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^T$ the vector containing the linear and angular velocities in the body frame. From 3D body dynamics, it follows that the two reference frames are linked by the following relations:

$$\mathbf{v} = \mathbf{R} \cdot \mathbf{v_B}, \tag{2.5}$$

$$\boldsymbol{\omega} = \mathbf{T} \cdot \boldsymbol{\omega_B}, \tag{2.6}$$

where $\mathbf{v} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T \in \mathbb{R}^3$, $\boldsymbol{\omega} = \begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T \in \mathbb{R}^3$, $\mathbf{v_B} = \begin{bmatrix} u & v & w \end{bmatrix}^T \in \mathbb{R}^3$, $\boldsymbol{\omega_B} = \begin{bmatrix} p & q & r \end{bmatrix}^T \in \mathbb{R}^3$, and $\mathbf{T}$ is a matrix for angular transformations [27]

$$\mathbf{T} = \begin{bmatrix} 1 & s(\phi)t(\theta) & c(\phi)t(\theta) \\ 0 & c(\phi) & -s(\phi) \\ 0 & \frac{s(\phi)}{c(\theta)} & \frac{c(\phi)}{c(\theta)} \end{bmatrix}, \tag{2.7}$$

where $t(\theta) = \tan(\theta)$. So, the kinematic model of the quadrotor is:

$$\begin{cases} \dot{x} = w[s(\phi)s(\psi) + c(\phi)c(\psi)s(\theta)] - v[c(\phi)s(\psi) - c(\psi)s(\phi)s(\theta)] + u[c(\psi)c(\theta)] \\ \dot{y} = v[c(\phi)c(\psi) + s(\phi)s(\psi)s(\theta)] - w[c(\psi)s(\phi) - c(\phi)s(\psi)s(\theta)] + u[c(\theta)s(\psi)] \\ \dot{z} = w[c(\phi)c(\theta)] - u[s(\theta)] + v[c(\theta)s(\phi)] \\ \dot{\phi} = p + r[c(\phi)t(\theta)] + q[s(\phi)t(\theta)] \\ \dot{\theta} = q[c(\phi)] - r[s(\phi)] \\ \dot{\psi} = r\frac{c(\phi)}{c(\theta)} + q\frac{s(\phi)}{c(\theta)} \end{cases} \tag{2.8}$$

Newton's law states the following matrix relation for the total force acting on the quadrotor:

$$m(\boldsymbol{\omega_B} \wedge \mathbf{v_B} + \dot{\mathbf{v}}_\mathbf{B}) = \mathbf{f_B}, \tag{2.9}$$

where m is the mass of the quadrotor, $\wedge$ is the cross product and $\mathbf{f_B} = \begin{bmatrix} f_x & f_y & f_z \end{bmatrix}^T \in \mathbb{R}^3$ is the total force.
Euler's equation gives the total torque applied to the quadrotor:

$$\mathbf{I} \cdot \dot{\boldsymbol{\omega}}_\mathbf{B} + \boldsymbol{\omega_B} \wedge (\mathbf{I} \cdot \boldsymbol{\omega_B}) = \mathbf{m_B}, \tag{2.10}$$

where $\mathbf{m_B} = \begin{bmatrix} m_x & m_y & m_z \end{bmatrix}^T \in \mathbb{R}^3$ is the total torque and $\mathbf{I}$ is the diagonal inertia matrix:

$$\mathbf{I} = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \in \mathbb{R}^{3\times3}.$$

So, the dynamic model of the quadrotor in the body frame is:

$$\begin{cases} f_x = m(\dot{u} + qw - rv) \\ f_y = m(\dot{v} - pw + ru) \\ f_z = m(\dot{w} + pv - qu) \\ m_x = \dot{p}I_x - qrI_y + qrI_z \\ m_y = \dot{q}I_y + prI_x - prI_z \\ m_z = \dot{r}I_z - pqI_x + pqI_y \end{cases} \tag{2.11}$$

The equations stand as long as we assume that the origin and the axes of the body frame coincide with the barycenter of the quadrotor and the principal axes.

2.4 Forces and moments

The external forces in the body frame, $\mathbf{f_B}$ are given by:

$$\mathbf{f_B} = mg\mathbf{R}^T \cdot \hat{\mathbf{e}}_\mathbf{z} - f_t\hat{\mathbf{e}}_\mathbf{3} + \mathbf{f_w}, \tag{2.12}$$

where $\hat{\mathbf{e}}_\mathbf{z}$ is the unit vector in the inertial z axis, $\hat{\mathbf{e}}_\mathbf{3}$ is the unit vector in the body z axis, g is the gravitational acceleration, f_t is the total thrust generated by rotors and $\mathbf{f_w} = \begin{bmatrix} f_{wx} & f_{wy} & f_{wz} \end{bmatrix}^T \in \mathbb{R}^3$ are the forces produced by wind on the quadrotors. The external moments in the body frame, $\mathbf{m_B}$ are given by

$$\mathbf{m_B} = \boldsymbol{\tau_B} - \mathbf{g_a} + \boldsymbol{\tau_w}, \tag{2.13}$$

where $\mathbf{g_a}$ represents the gyroscopic moments caused by the combined rotation of the four rotors and the vehicle body, $\boldsymbol{\tau_B} = \begin{bmatrix} \tau_x & \tau_y & \tau_z \end{bmatrix}^T \in \mathbb{R}^3$ are the control torques generated by differences in the rotor speeds and $\boldsymbol{\tau_w} = \begin{bmatrix} \tau_{wx} & \tau_{wy} & \tau_{wz} \end{bmatrix}^T \in \mathbb{R}^3$ are the torques produced by wind on the quadrotors. $\mathbf{g_a}$ is given by

$$\mathbf{g_a} = \sum_{i=1}^{4} J_p(\boldsymbol{\omega_B} \wedge \hat{\mathbf{e}}_\mathbf{3})(-1)^{i+1}\Omega_i, \tag{2.14}$$

where J_p is the inertia of each rotor and Ω_i is the angular speed of the i^{th} rotor. According to [18], the J_p term is found to be small and, for this reason, the gyroscopic moments are removed in the controller formulation. In addition, there are numerous aerodynamic and aeroelastic phenomenon that affect the flight of the quadrotor, such as the ground effects: when flying close to the ground (or during the landing stage), the air flow generated by the propellers disturbs the dynamics of the quadrotors. So, the complete dynamic model of the quadrotor in the body frame is obteined substituting the force expression in (2.11):

$$\begin{cases} -mg[s(\theta)] + f_{wx} = m(\dot{u} + qw - rv) \\ mg[c(\theta)s(\phi)] + f_{wy} = m(\dot{v} - pw + ru) \\ mg[c(\theta)c(\phi)] + f_{wz} - f_t = m(\dot{w} + pv - qu) \\ \tau_x + \tau_{wx} = \dot{p}I_x - qrI_y + qrI_z \\ \tau_y + \tau_{wy} = \dot{q}I_y + prI_x - prI_z \\ \tau_z + \tau_{wz} = \dot{r}I_z - pqI_x + pqI_y \end{cases} \tag{2.15}$$

2.5 Actuator dynamics

Here we consider the inputs that can be applied to the system in order to control the behavior of the quadrotor. The rotors are four and the degrees of freedom we control are as many: commonly, the control inputs that are considered are one for the vertical thrust and one for each of the angular motions. Let us consider the values of the input forces and torques proportional to the squared speeds of the rotors [19]; their values are the following:

$$\begin{cases} f_t = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ \tau_x = bl(\Omega_3^2 - \Omega_1^2) \\ \tau_y = bl(\Omega_4^2 - \Omega_2^2) \\ \tau_z = d(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \end{cases} \tag{2.16}$$

where l is the distance between any rotor and the center of the drone, b is the thrust factor and d is the drag factor. Substituting (2.16) in (2.15), we have: So, the dynamic model of the quadrotor in the body frame is:

$$\begin{cases} -mg[s(\theta)] + f_{wx} = m(\dot{u} + qw - rv) \\ mg[c(\theta)s(\phi)] + f_{wy} = m(\dot{v} - pw + ru) \\ mg[c(\theta)c(\phi)] + f_{wz} - b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) = m(\dot{w} + pv - qu) \\ bl(\Omega_3^2 - \Omega_1^2) + \tau_{wx} = \dot{p}I_x - qrI_y + qrI_z \\ bl(\Omega_4^2 - \Omega_2^2) + \tau_{wy} = \dot{q}I_y + prI_x - prI_z \\ d(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 + \Omega_3^2) + \tau_{wz} = \dot{r}I_z - pqI_x + pqI_y \end{cases} \tag{2.17}$$

2.6 State-space model

Organizing the state's vector in the following way:

$$\mathbf{x} = \begin{bmatrix} \phi & \theta & \psi & p & q & r & u & v & w & x & y & z \end{bmatrix}^T \in \mathbb{R}^{12} \tag{2.18}$$

it is possible to rewrite the equations of the dynamics of the quadrotor in the state-space from (2.8) and (2.15):

$$\begin{cases} \dot{\phi} = p + r[c(\phi)t(\theta)] + q[s(\phi)t(\theta)] \\ \dot{\theta} = q[c(\phi)] - r[s(\phi)] \\ \dot{\psi} = r\frac{c(\phi)}{c(\theta)} + q\frac{s(\phi)}{c(\theta)} \\ \dot{p} = \frac{I_y - I_z}{I_x} rq + \frac{\tau_x + \tau_{wx}}{I_x} \\ \dot{q} = \frac{I_z - I_x}{I_y} pr + \frac{\tau_y + \tau_{wy}}{I_y} \\ \dot{r} = \frac{I_x - I_y}{I_z} pq + \frac{\tau_z + \tau_{wz}}{I_z} \\ \dot{u} = rv - qw - g[s(\theta)] + \frac{f_{wx}}{m} \\ \dot{v} = pw - ru + g[s(\phi)c(\theta)] + \frac{f_{wy}}{m} \\ \dot{w} = qu - pv + g[c(\theta)c(\phi)] + \frac{f_{wz} - f_t}{m} \\ \dot{x} = w[s(\phi)s(\psi) + c(\phi)c(\psi)s(\theta)] - v[c(\phi)s(\psi) - c(\psi)s(\phi)s(\theta)] + u[c(\psi)c(\theta)] \\ \dot{y} = v[c(\phi)c(\psi) + s(\phi)s(\psi)s(\theta)] - w[c(\psi)s(\phi) - c(\phi)s(\psi)s(\theta)] + u[c(\theta)s(\psi)] \\ \dot{z} = w[c(\phi)c(\theta)] - u[s(\theta)] + v[c(\theta)s(\phi)] \end{cases} \tag{2.19}$$

Below we obtain two alternative forms of the dynamical model useful for studying the control. From Newton's law we can write:

$$m\dot{\mathbf{v}} = \mathbf{R} \cdot \mathbf{f_B} = mg\hat{\mathbf{e}}_\mathbf{z} - f_t \mathbf{R} \cdot \hat{\mathbf{e}}_\mathbf{3}, \tag{2.20}$$

therefore:

$$\begin{cases} \ddot{x} = -\frac{f_t}{m}[s(\phi)s(\psi) + c(\phi)c(\psi)s(\theta)] \\ \ddot{y} = -\frac{f_t}{m}[c(\phi)s(\psi)s(\theta) - c(\psi)s(\phi)] \\ \ddot{z} = g - \frac{f_t}{m}[c(\phi)c(\theta)] \end{cases} \tag{2.21}$$

Now a simplification is made by setting $\begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T = \begin{bmatrix} p & q & r \end{bmatrix}^T$. This assumption holds true for small angles of movement [11]. So, the dynamic model of the quadrotor in the inertial frame is:

$$\begin{cases} \ddot{x} = -\frac{f_t}{m}[s(\phi)s(\psi) + c(\phi)c(\psi)s(\theta)] \\ \ddot{y} = -\frac{f_t}{m}[c(\phi)s(\psi)s(\theta) - c(\psi)s(\phi)] \\ \ddot{z} = g - \frac{f_t}{m}[c(\phi)c(\theta)] \\ \ddot{\phi} = \frac{I_y - I_z}{I_x}\dot{\theta}\dot{\psi} + \frac{\tau_x}{I_x} \\ \ddot{\theta} = \frac{I_z - I_x}{I_y}\dot{\phi}\dot{\psi} + \frac{\tau_y}{I_y} \\ \ddot{\psi} = \frac{I_x - I_y}{I_z}\dot{\phi}\dot{\theta} + \frac{\tau_z}{I_z} \end{cases} \tag{2.22}$$

Redefining the state's vector as:

$$\mathbf{x} = \begin{bmatrix} x & y & z & \psi & \theta & \phi & \dot{x} & \dot{y} & \dot{z} & p & q & r \end{bmatrix}^T \in \mathbb{R}^{12} \tag{2.23}$$

it is possible to rewrite the equations of the quadrotor in the spate-space:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \sum_{i=1}^{4} \mathbf{g_i}(\mathbf{x})u_i, \tag{2.24}$$

where

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ q\frac{s(\phi)}{c(\theta)} + r\frac{c(\phi)}{c(\theta)} \\ q[c(\phi)] - r[s(\phi)] \\ p + q[s(\phi)t(\theta)] + r[c(\phi)t(\theta)] \\ 0 \\ 0 \\ g \\ \frac{(I_y - I_z)}{I_x}qr \\ \frac{(I_z - I_x)}{I_y}pr \\ \frac{(I_x - I_y)}{I_z}pq \end{bmatrix} \tag{2.25}$$

and

$$\mathbf{g_1}(\mathbf{x}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & g_1^7 & g_1^8 & g_1^9 & 0 & 0 & 0 \end{bmatrix}^T \in \mathbb{R}^{12},$$

$$\mathbf{g_2}(\mathbf{x}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{I_x} & 0 & 0 \end{bmatrix}^T \in \mathbb{R}^{12},$$

$$\mathbf{g_3}(\mathbf{x}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{I_y} & 0 \end{bmatrix}^T \in \mathbb{R}^{12},$$

$$\mathbf{g_4}(\mathbf{x}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{I_z} \end{bmatrix}^T \in \mathbb{R}^{12},$$

with

$$g_1^7 = -\tfrac{1}{m}[s(\phi)s(\psi) + c(\phi)c(\psi)s(\theta)],$$
$$g_1^8 = -\tfrac{1}{m}[c(\psi)s(\phi) - c(\phi)s(\psi)s(\theta)],$$
$$g_1^9 = -\tfrac{1}{m}[c(\phi)c(\theta)].$$

2.7 Linear model

Set $\mathbf{u}$ the control vector: $\mathbf{u} = \begin{bmatrix} f_t & \tau_x & \tau_y & \tau_z \end{bmatrix}^T \in \mathbb{R}^4$. The linearization's procedure is developed around an equilibrium point $\bar{\mathbf{x}}$, which for fixed input $\bar{\mathbf{u}}$ is the solution of the algebric system: or rather that value of state's vector, which on fixed constant input is the solution of algebraic system:

$$\hat{\mathbf{f}}(\bar{\mathbf{x}}, \bar{\mathbf{u}}) = \mathbf{0}. \tag{2.26}$$

Since the function $\hat{\mathbf{f}}$ is nonlinear, problems related to the existence an uniqueness of the solution of system (2.26) arise. In particular, for the system in hand, the solution is difficult to find in closed form because of trigonometric functions related each other in no-elementary way. For this reason, the linearization is performed on a simplified model called *to small oscillations*. This simplification is made by approximating the sine function with its argument and the cosine function with unity. The approximation is valid if the argument is small. The resulting system is described by the following equations:

$$\begin{cases} \dot{\phi} \approx p + r\theta + q\phi\theta \\ \dot{\theta} \approx q - r\phi \\ \dot{\psi} \approx r + q\phi \\ \dot{p} \approx \frac{I_y - I_z}{I_x} rq + \frac{\tau_x + \tau_{wx}}{I_x} \\ \dot{q} \approx \frac{I_z - I_x}{I_y} pr + \frac{\tau_y + \tau_{wy}}{I_y} \\ \dot{r} \approx \frac{I_x - I_y}{I_z} pq + \frac{\tau_z + \tau_{wz}}{I_z} \\ \dot{u} \approx rv - qw - g\theta + \frac{f_{wx}}{m} \\ \dot{v} \approx pw - ru + g\phi + \frac{f_{wy}}{m} \\ \dot{w} \approx qu - pv + g + \frac{f_{wz} - f_t}{m} \\ \dot{x} \approx w(\phi\psi + \theta) - v(\psi - \phi\theta) + u \\ \dot{y} \approx v(1 + \phi\psi\theta) - w(\phi - \psi\theta) + u\psi \\ \dot{z} \approx w - u\theta + v\phi \end{cases} \tag{2.27}$$

which can be written in the compact form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}). \tag{2.28}$$

2.7.1 Linearization

As said above, in order to perform the linearization, an equilibrium point is needed. Such an equilibrium point can be:

$$\bar{\mathbf{x}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{x} & \bar{y} & \bar{z} \end{bmatrix}^T \in \mathbb{R}^{12}. \tag{2.29}$$

From the equations, we can find that the equilibrium point (2.29) is obtained by the constant input value:

$$\bar{\mathbf{u}} = \begin{bmatrix} mg & 0 & 0 & 0 \end{bmatrix}^T \in \mathbb{R}^4. \tag{2.30}$$

Note that this particular value represents the force necessary to delete the quadrotor's weight and it consents its hovering. After determined the equilibrium point $\bar{\mathbf{x}}$ and the corresponding nominal input $\bar{\mathbf{u}}$, we have that the matrices associated to the linear system are given by relations:

$$\mathbf{A} = \left.\frac{\partial \mathbf{f}(\mathbf{x},\mathbf{u})}{\partial \mathbf{x}}\right|_{\substack{\mathbf{x}=\bar{\mathbf{x}}\\ \mathbf{u}=\bar{\mathbf{u}}}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{2.31}$$

$$\mathbf{B} = \left.\frac{\partial \mathbf{f}(\mathbf{x},\mathbf{u})}{\partial \mathbf{u}}\right|_{\substack{\mathbf{x}=\bar{\mathbf{x}}\\ \mathbf{u}=\bar{\mathbf{u}}}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{I_x} & 0 & 0 \\ 0 & 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & 0 & \frac{1}{I_z} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.32}$$

If we consider the disturbance by wind, set:

$$\mathbf{d} = \begin{bmatrix} f_{wx} & f_{wy} & f_{wz} & \tau_{wx} & \tau_{wy} & \tau_{wz} \end{bmatrix}^T \in \mathbb{R}^6 \tag{2.33}$$

$$\mathbf{D} = \left.\frac{\partial \mathbf{f}(\mathbf{x},\mathbf{u},\mathbf{d})}{\partial \mathbf{d}}\right|_{\substack{\mathbf{x}=\bar{\mathbf{x}}\\ \mathbf{u}=\bar{\mathbf{u}}}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{I_x} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{I_z} \\ \frac{1}{m} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{m} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.34}$$

the linear model is:

$$\dot{\mathbf{x}} = \mathbf{A}\cdot\mathbf{x} + \mathbf{B}\cdot\mathbf{u} + \mathbf{D}\cdot\mathbf{d} \tag{2.35}$$

$$\begin{cases} \dot{\phi} = p \\ \dot{\theta} = q \\ \dot{\psi} = r \\ \dot{p} = \frac{\tau_x + \tau_{wx}}{I_x} \\ \dot{q} = \frac{\tau_y + \tau_{wy}}{I_y} \\ \dot{r} = \frac{\tau_z + \tau_{wz}}{I_z} \\ \dot{u} = -g\theta + \frac{f_{wx}}{m} \\ \dot{v} = g\phi + \frac{f_{wy}}{m} \\ \dot{w} = \frac{f_{wz} - f_t}{m} \\ \dot{x} = u \\ \dot{y} = v \\ \dot{z} = w \end{cases} \tag{2.36}$$

2.7.2 Controllability and observability of the linear system

Controllability and observability represent two major concepts of modern control system theory [20]. These concepts were introduced by R. Kalman in 1960. They can be roughly defined as follows.

- Controllability: In order to be able to do whatever we want with the given dynamic system under control input, the system must be controllable.
- Observability: In order to see what is going on inside the system, the system must be observable.

The concepts of controllability and observability for a linear time-invariant dynamical system can be related to suitable linear systems of algebraic equations. It is well known that a solvable system of linear algebraic equations has a solution if and only if the rank of the system matrix is full. Observability and controllability tests will be connected to the rank tests of certain matrices: the controllability and observability matrices. For the purpose of studying its observability, we consider the linear system:

$$\dot{\mathbf{x}} = \mathbf{A} \cdot \mathbf{x} \qquad \mathbf{x}(t_0) = \mathbf{x_0} \tag{2.37}$$

where $\mathbf{A}$ is given in (2.31) and with the corresponding measurements:

$$\boldsymbol{y} = \mathbf{C} \cdot \mathbf{x} \tag{2.38}$$

of dimensions $\mathbf{x} \in \mathbb{R}^{12}, \mathbf{y} \in \mathbb{R}^{12}, \mathbf{A} \in \mathbb{R}^{12x12}, \mathbf{C} \in \mathbb{R}^{12\times12}$. The observability matrix is given by:

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{C} \cdot \mathbf{A} \\ \mathbf{C} \cdot \mathbf{A}^2 \\ \vdots \\ \mathbf{C} \cdot \mathbf{A}^{11} \end{bmatrix} \in \mathbb{R}^{144\times12} \tag{2.39}$$

The linear continuous-time system (2.37) with measurements (2.38) is observable if and only if the observability matrix has full rank.
The controllability matrix is given by:

$$\mathcal{C} = \begin{bmatrix} \mathbf{B} & \mathbf{A} \cdot \mathbf{B} & \mathbf{A}^2 \cdot \mathbf{B} & \cdots & \mathbf{A}^{11} \cdot \mathbf{B} \end{bmatrix} \in \mathbb{R}^{12\times48} \tag{2.40}$$

where $\mathbf{B}$ is given in (2.32). The linear continuous-time system (2.37) with measurements (2.38) is controllable if and only if the controllability matrix has full rank. To check its observability and controllability, we used MATLAB. The linear system results to be controllable and observable.

Chapter 3

Control strategies

In this section we discuss three control strategies. One control strategy is linear (Linear Quadratic Regulator), and the other two control strategies are nonlinear (exact linearization and non-interacting control via dynamic feedback and dynamic inversion with zero-dynamics stabilization). Some comparisons about these control strategies are done.

3.1 Linear Quadratic Regulator control

The objective of the optimal control [30] is to determine control signal so that the system to be controlled can meet physical constraints and minimize/maximize a cost/performance function. Namely, the solution of an optimization problem is supposed to bring the system's state $\mathbf{x}(t)$ to the desired trajectory $\mathbf{x_d}$ minimizing some cost. Furthermore, it minimize the use of the control inputs, thus reducing the use of actuators.

The underling needs of optimization control are:

- A model able to best describe the behavior of the dynamic system target of control;
- A cost index J, taking into account specifications and need of the designer;
- Possible boundary conditions and physical constraints limiting the system.

Let us consider a dynamic system and set $\mathbf{x}$ as state and set $\mathbf{u}$ as input:

$$\dot{\mathbf{x}}(t) = f[\mathbf{x}(t), \mathbf{u}(t), t]. \tag{3.1}$$

Being $\mathbf{x}$ and $\mathbf{u}$ vectors of $n-1$ and $r-1$ length, it is possible to define J as:

$$J = e[\mathbf{x}(t_f)] + \int_{t_0}^{t_f} w[\mathbf{x}(t), \mathbf{u}(t), t]dt, \tag{3.2}$$

where the *weight* function w and the terminal cost e are non-negative function such as $w(\mathbf{0}, \mathbf{0}, t) = 0$ and $e(\mathbf{0}) = 0$. The boundary conditions are:

- $\mathbf{x}(t_0) = \mathbf{x_0}$;
- $\mathbf{x}(t_f)$ and t_f unconstrained.

It is then possible to define the solution of an optimization problem:

$$\mathbf{u}(t), \forall t \in [t_0, t_f]. \tag{3.3}$$

Such a function aims at minimizing J. Considering the limit of the time interval as approaching infinitive, highlighting the system and the cost index:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A} \cdot \mathbf{x} + \mathbf{B} \cdot \mathbf{u} \\ \mathbf{y} = \mathbf{C} \cdot \mathbf{x} \end{cases} \tag{3.4}$$

$$J = \int_{t_0}^{\infty} \{\mathbf{u}(t)^T \cdot \mathbf{R} \cdot \mathbf{u}(t) + [\mathbf{x}(t) - \mathbf{x_d}(t)]^T \cdot \mathbf{Q} \cdot [\mathbf{x}(t) - \mathbf{x_d}(t)]\}dt, \tag{3.5}$$

being $\mathbf{R}$ and $\mathbf{Q}$ such matrixes that:

- $\mathbf{R}$ is the cost of actuators ($\mathbf{R} = \mathbf{R}^T$ positive definite matrix; $\mathbf{R} \in \mathbb{R}^{m \times m}$);
- $\mathbf{Q}$ is the cost of the state ($\mathbf{Q} = \mathbf{Q}^T$ positive semi-definite matrix; $\mathbf{Q} \in \mathbb{R}^{r \times r}$).

It is possible to demonstrate [30] that the control's input $\mathbf{u}(\cdot)$ which minimizes the functional is a state linear feedback as:

$$\mathbf{u}(t) = -\mathbf{K} \cdot [\mathbf{x}(t) - \mathbf{x_d}(t)], \tag{3.6}$$

where:

$$\mathbf{K} = \mathbf{R}^{-1} \cdot \mathbf{B}^T \cdot \mathbf{S}. \tag{3.7}$$

The $\mathbf{S}$ matrix is a solution of the Riccati's algebraic equation:

$$\mathbf{S} \cdot \mathbf{A} + \mathbf{A}^T \cdot \mathbf{S} - \mathbf{S} \cdot \mathbf{B}\mathbf{R}^{-1} \cdot \mathbf{B}^T \cdot \mathbf{S} + \mathbf{C}^T \cdot \mathbf{Q} \cdot \mathbf{C} = \mathbf{0}, \tag{3.8}$$

where $\mathbf{S}$ is a positive definite matrix. The Figure 3.1 shows a scheme of the implemented system:

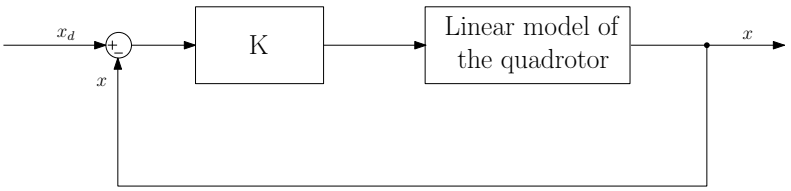


Figure 3.1: LQR control.

The algebraic equation can be solved through Riccati's method, performed by Matlab through LQR function:

$$\mathbf{K} = LQR(\mathbf{A}, \mathbf{B}, \mathbf{Q}, \mathbf{R}).$$

Given a LQR problem [30] over an infinity horizon and factorized the matrix $\mathbf{Q} = \mathbf{E}^T \cdot \mathbf{E}$, if the ($\mathbf{A}$,$\mathbf{B}$) couple is controllable and the ($\mathbf{A}$,$\mathbf{E}$) couple is observable, we will get:

- there exists one $\mathbf{S}$ solution positive definite of the Riccati's algebraic equation;
- the closed loop's system $\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{B} \cdot \mathbf{K}) \cdot \mathbf{x}$ is asymptotically stable, with $\mathbf{K} = \mathbf{R}^{-1} \cdot \mathbf{B}^T \cdot \mathbf{S}$.

We consider the linear system (2.35) without disturbance. We choose the matrices $\mathbf{Q}$ and $\mathbf{R}$ taking in account $\mathbf{A}$ (2.31) and $\mathbf{B}$ (2.32). We apply the LQR control using the LQR function from MATLAB/SIMULINK.

3.2 Feedback linearization control

Feedback linearization is an approach to nonlinear control design [28] that has attracted several researches in the last years. The central idea is to algebraically transform nonlinear systems dynamics into (fully or partly) linear ones, so that linear control techniques can be applied. Two nonlinear control design techniques are discussed here in detail.
In this chapter, we will focus on continuous-time, state-space models of the form

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x}) \cdot \mathbf{u} \\ \mathbf{y} = \mathbf{h}(\mathbf{x}) \end{cases} \tag{3.9}$$

where: $\mathbf{x} \in \mathbb{R}^n$ is the vector of state variables, $\mathbf{u} \in \mathbb{R}^m$ is the vector of control input variables, $\mathbf{y} \in \mathbb{R}^m$ is the vector of output variables, $\mathbf{f}(\mathbf{x})$ is an n-dimensional vector of nonlinear functions, $\mathbf{G}(\mathbf{x})$ is an $(n \times m)$-dimensional matrix of nonlinear functions and $\mathbf{h}(\mathbf{x})$ is an m-dimensional vector of nonlinear functions. The single-input, single-output (SISO) case where $m = 1$ will be emphasized to explain the basic concepts.
Consider the Jacobian linearization 2.7.1 of the nonlinear model (3.9) around an equilibrium point $(\mathbf{u_0}, \mathbf{x_0}, \mathbf{y_0})$. In this way the model can be written as a linearized state-space system,

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A} \cdot \mathbf{x} + \mathbf{B} \cdot \mathbf{u} \\ \mathbf{y} = \mathbf{C} \cdot \mathbf{x} \end{cases} \tag{3.10}$$

with obvious definition for the matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$. It is important to note that (3.2) is an exact representation of nonlinear model only at the point $(\mathbf{x_0}, \mathbf{u_0})$. As a result, a control strategy based on a linearized model may yield unsatisfactoy performance and robustness at other operating points.
In this section we show that this kind of nonlinear control techniques can produce a linear model that is an *exact* representation of the original nonlinear model over a large set of operating conditions. The feedback linearization is based on two operations:

- nonlinear change of coordinates;
- nonlinear state feedback.

After the feedback linearization, the input-output model is linear in the new set of coordinates. Specifically, we have:

$$\begin{cases} \dot{\boldsymbol{\xi}} = \mathbf{A} \cdot \boldsymbol{\xi} + \mathbf{B} \cdot \mathbf{v} \\ \mathbf{w} = \mathbf{C} \cdot \boldsymbol{\xi} \end{cases} \tag{3.11}$$

where: $\boldsymbol{\xi} \in \mathbb{R}^r$ is a vector of transformed state variables, $\mathbf{v} \in \mathbb{R}^m$ is a trasformed input variables, $\mathbf{w} \in \mathbb{R}^m$ is a vector of transformed output variables and the matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ have a very simple canonical structure. If $r < n$, an additional $n-r$ state variables must be introduced to complete the coordinate transformation. The integer r is called the *relative degree* and is a fundamental characteristic of a nonlinear system.
Most feedback linearization approaches are based on *input-output linearization* or *state-space linearization*. In the input-output linearization approach, the objective is to linearize the map beetween the trasformed input $\mathbf{v}$ and the actual output $\mathbf{y}$. A controller is then designed for the linearized input-output model. In the state-space linearization approach, the goal is to linearize the map between the transformed inputs and the entire vector of transformed state variables. A linear controller is then synthesized for the linear input-state model. However, this approach may lead to a complex controller design task because the map between the transformed inputs and the original outputs $\mathbf{y}$ is generally nonlinear. Feedback linearization produces a linear model by the use of nonlinear coordinate transformations and nonlinear state feedback. In some applications, the control objectives can be achieved with a nonlinear *static* feedback control law of the form,

$$\mathbf{u} = \boldsymbol{\alpha}(\mathbf{x}) + \boldsymbol{\beta}(\mathbf{x}) \cdot \mathbf{v}, \tag{3.12}$$

where $\boldsymbol{\alpha}$ is an m-dimensional vector of nonlinear functions and $\boldsymbol{\beta}$ is an $m \times m$ matrix of nonlinear functions. For some processes, it is not possible to satisfy the control objective with a static control and a *dynamic* state feedback control law must be employed,

$$\begin{cases} \dot{\boldsymbol{\zeta}} = \boldsymbol{\gamma}(\mathbf{x}, \boldsymbol{\zeta}) + \boldsymbol{\Delta}(\mathbf{x}, \boldsymbol{\zeta}) \cdot \mathbf{v} \\ \mathbf{u} = \boldsymbol{\alpha}(\mathbf{x}, \boldsymbol{\zeta}) + \boldsymbol{\beta}(\mathbf{x}, \boldsymbol{\zeta}) \cdot \mathbf{v} \end{cases} \tag{3.13}$$

where $\boldsymbol{\zeta}$ is an q-dimensional vector of controller state variables; $\boldsymbol{\gamma}$ is an q-dimensional vector of nonlinear functions; and $\boldsymbol{\Delta}$ is a $q \times m$ matrix of nonlinear

functions.
The quadrotor has six outputs $\mathbf{y} = \begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}$ and the vehicle has four inputs. There are two degree of freedom that are left uncontrollable. A solution to this problem [11] is to decompose it into two distinct control loops (Dynamic inversion with zero-dynamics stabilization). Another solution provides the use of *dynamic* feedback control (Exact linearization and non-interacting control via dynamic feedback). Such control structures are based on the input-output linearization described in appendix A.

3.2.1 Exact linearization and non-interacting control via dynamic feedback

This section deals with the design of a feedback control law (and a change of coordinates in the state-space) to the purpose of transforming the nonlinear system (2.24) into a linear and controllable one: this problem is known in the literature as the exact linearization problem [23, 24, 25]. Moreover, from an input-output point of view, we would like to reduce the system , to an aggregate of independent single-input single-output channels: this is the non-interacting control problem [23] or input-output decoupling problem [24]. It will be shown that none of these two problems is solvable for the nonlinear system (2.24) by means of a static state feedback control law but by means of a dynamic feedback control law. First, it is necessary to define the control objective by choosing an output function for the system (2.24). To avoid unnecessary complications, we set the number of input channels equal to the number of output channels. We would like to control the absolute position of the quadrotor $\begin{bmatrix} x & y & z \end{bmatrix}^T$ and the yaw angle ψ. Therefore, the output function is chosen as:

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) = \begin{bmatrix} x & y & z & \psi \end{bmatrix}^T. \tag{3.14}$$

We assume the state $\mathbf{x}$ of the system being fully available for measurements and we seek a static state feedback control law of the form:

$$\mathbf{u} = \boldsymbol{\alpha}(\mathbf{x}) + \boldsymbol{\beta}(\mathbf{x}) \cdot \mathbf{v}, \tag{3.15}$$

where $\mathbf{v}$ is an external reference input to be defined later,
$\boldsymbol{\alpha}(\mathbf{x}) = \begin{bmatrix} \alpha_1(\mathbf{x}) & \alpha_2(\mathbf{x}) & \alpha_3(\mathbf{x}) & \alpha_4(\mathbf{x}) \end{bmatrix}^T$ and $\boldsymbol{\beta}(\mathbf{x}) \in \mathbb{R}^{4\times4}$.
Let $\begin{bmatrix} r_1 & r_2 & r_3 & r_4 \end{bmatrix}^T$ be the relative degree vector of the system (2.24). We have

$$\begin{bmatrix} y_1^{(r_1)} & y_2^{(r_2)} & y_3^{(r_3)} & y_4^{(r_4)} \end{bmatrix}^T = \mathbf{b}(\mathbf{x}) + \boldsymbol{\Delta}(\mathbf{x}) \cdot \mathbf{u}, \tag{3.16}$$

where

$$\boldsymbol{\Delta}(\mathbf{x}) = \begin{bmatrix} L_{g1}L_f^{r_1-1}h_1(\mathbf{x}) & L_{g2}L_f^{r_1-1}h_1(\mathbf{x}) & L_{g3}L_f^{r_1-1}h_1(\mathbf{x}) & L_{g4}L_f^{r_1-1}h_1(\mathbf{x}) \\ L_{g1}L_f^{r_2-1}h_2(\mathbf{x}) & L_{g2}L_f^{r_2-1}h_2(\mathbf{x}) & L_{g3}L_f^{r_2-1}h_2(\mathbf{x}) & L_{g4}L_f^{r_2-1}h_2(\mathbf{x}) \\ L_{g1}L_f^{r_3-1}h_3(\mathbf{x}) & L_{g2}L_f^{r_3-1}h_3(\mathbf{x}) & L_{g3}L_f^{r_3-1}h_3(\mathbf{x}) & L_{g4}L_f^{r_3-1}h_3(\mathbf{x}) \\ L_{g1}L_f^{r_4-1}h_4(\mathbf{x}) & L_{g2}L_f^{r_4-1}h_4(\mathbf{x}) & L_{g3}L_f^{r_4-1}h_4(\mathbf{x}) & L_{g4}L_f^{r_4-1}h_4(\mathbf{x}) \end{bmatrix}, \tag{3.17}$$

$$\mathbf{b}(\mathbf{x}) = \begin{bmatrix} L_f^{r_1}h_1(\mathbf{x}) \\ L_f^{r_2}h_2(\mathbf{x}) \\ L_f^{r_3}h_4(\mathbf{x}) \\ L_f^{r_4}h_4(\mathbf{x}) \end{bmatrix}. \tag{3.18}$$

The input-output decoupling problem is solvable if and only if the matrix $\boldsymbol{\Delta}(\mathbf{x})$ is nonsingular. In this case, the static state feedback (3.15) with:

$$\begin{cases} \boldsymbol{\alpha}(\mathbf{x}) = -\boldsymbol{\Delta}^{-1}(\mathbf{x}) \cdot \mathbf{b}(\mathbf{x}) \\ \boldsymbol{\beta}(\mathbf{x}) = \boldsymbol{\Delta}^{-1}(\mathbf{x}) \end{cases} \tag{3.19}$$

renders the closed loop system linear and decoupled from an input-output point of view. More precisely, we have

$$y_i^{(r_i)} = v_i \text{ for all } i,\ 1 \leq i \leq 4.$$

However, for the nonlinear system (2.24), we have

$$r_1 = r_2 = r_3 = r_4 = 2$$

and

$$\mathbf{\Delta}(\mathbf{x}) = \begin{bmatrix} \delta_{1,1} & 0 & 0 & 0 \\ \delta_{2,1} & 0 & 0 & 0 \\ \delta_{3,1} & 0 & 0 & 0 \\ 0 & 0 & \delta_{4,3} & \delta_{4,4} \end{bmatrix},$$

with:

$$\delta_{1,1} = g_1^7;$$

$$\delta_{2,1} = g_1^8;$$

$$\delta_{3,1} = g_1^9;$$

$$\delta_{4,3} = \frac{s(\phi)}{I_y c(\theta)};$$

$$\delta_{4,4} = \frac{c(\phi)}{I_z c(\theta)}.$$

Obviously $\mathbf{\Delta}(\mathbf{x})$ is singular for all $\mathbf{x}$ and therefore the input-output decoupling problem is not solvable for the system (2.24) by means of a static state feedback control law. The reason why the matrix $\mathbf{\Delta}(\mathbf{x})$ is singular is that the derivatives $y_1^{(2)}$, $y_2^{(2)}$ and $y_3^{(2)}$ are allaffected by the input u_1 and not by u_2, u_3, u_4. Thus, in order to get $\mathbf{\Delta}(\mathbf{x})$ nonsingular, we could try to make $y_1^{(2)}$, $y_2^{(2)}$ and $y_3^{(2)}$ indipendent of u_1, that is to delay the appearance of u_1 to higher order derivatives of y_1, y_2 and y_3 and hope that the others inputs show up [23]. In order to achieve this result, we set u_1 equal to the output of a double integrator driven by $\bar{u_1}$, i.e.

$$\begin{cases} u_1 = \zeta \\ \dot{\zeta} = \xi \\ \dot{\xi} = \bar{u}_1 \end{cases} \tag{3.20}$$

For consistency of notation we also set, for the other input channels which have been left unchanged, the following

$$\begin{cases} u_2 = \bar{u_2} \\ u_3 = \bar{u_3} \\ u_4 = \bar{u_4} \end{cases} \tag{3.21}$$

Note that u_1 is not anymore an input for the system (2.24) but becomes the internal state ξ for the new dynamical system (3.20). The extended system obtained is described by equations of the form

$$\dot{\bar{\mathbf{x}}} = \bar{\mathbf{f}}(\bar{\mathbf{x}}) + \sum_{i=1}^{4} \bar{\mathbf{g}}_\mathbf{i}(\bar{\mathbf{x}})\bar{\mathbf{u}}_i, \tag{3.22}$$

in which

$$\bar{\mathbf{x}} = \begin{bmatrix} x & y & z & \psi & \theta & \phi & \dot{x} & \dot{y} & \dot{z} & \zeta & \xi & p & q & r \end{bmatrix}^T \in \mathbb{R}^{14}, \tag{3.23}$$

$$\bar{\mathbf{f}}(\bar{\mathbf{x}}) = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ q\frac{s(\phi)}{c(\theta)} + r\frac{c(\phi)}{c(\theta)} \\ q[c(\phi)] - r[s(\phi)] \\ p + q[s(\phi)t(\theta)] + r[c(\phi)t(\theta)] \\ g_1^7(\psi,\theta,\phi)\zeta \\ g_1^8(\psi,\theta,\phi)\zeta \\ g_1^9(\psi,\theta,\phi)\zeta \\ \xi \\ 0 \\ \frac{(I_y - I_z)}{I_x}qr \\ \frac{(I_z - I_x)}{I_y}pr \\ \frac{(I_x - I_y)}{I_z}pq \end{bmatrix}, \tag{3.24}$$

and

$$\bar{\mathbf{g}}_{\mathbf{1}}(\bar{\mathbf{x}}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}^T \in \mathbb{R}^{14},$$

$$\bar{\mathbf{g}}_{\mathbf{2}}(\bar{\mathbf{x}}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{I_x} & 0 & 0 \end{bmatrix}^T \in \mathbb{R}^{14},$$

$$\bar{\mathbf{g}}_{\mathbf{3}}(\bar{\mathbf{x}}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{I_y} & 0 \end{bmatrix}^T \in \mathbb{R}^{14}$$

$$\bar{\mathbf{g}}_{\mathbf{4}}(\bar{\mathbf{x}}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{I_z} \end{bmatrix}^T \in \mathbb{R}^{14}$$

Now, the input-output decoupling problem is solvable for the nonlinear system (2.24) by means of a dynamic feedback control law if it is solvable via a static feedback for the extended system (3.22). For the nonlinear system , the relative degree vector $\{r_1,r_2,r_3,r_4\}$ is given by

$$r_1 = r_2 = r_3 = 4, \qquad r_4 = 2,$$

and we have

$$\begin{bmatrix} y_1^{(r_1)} & y_2^{(r_2)} & y_3^{(r_3)} & y_4^{(r_4)} \end{bmatrix}^T = \mathbf{b}(\bar{\mathbf{x}}) + \mathbf{\Delta}(\bar{\mathbf{x}})\mathbf{u}, \tag{3.25}$$

where $\mathbf{\Delta}(\bar{\mathbf{x}})$ and $\mathbf{b}(\bar{\mathbf{x}})$ are computed using equations (3.17) and (3.18). The matrix $\mathbf{\Delta}(\bar{\mathbf{x}})$ is nonsingular at any point characterized by $\zeta \neq 0, -\frac{\pi}{2} < \phi < \frac{\pi}{2}, -\frac{\pi}{2} < \theta < \frac{\pi}{2}$.
Therefore, the input-output decoupling problem is solvable for the system (2.24) by means of a dynamic feedback control law of the form:

$$\bar{\mathbf{u}} = \boldsymbol{\alpha}(\bar{\mathbf{x}}) + \boldsymbol{\beta}(\bar{\mathbf{x}}) \cdot \mathbf{v}; \tag{3.26}$$

where $\boldsymbol{\alpha}(\bar{\mathbf{x}})$ and $\boldsymbol{\beta}(\bar{\mathbf{x}})$ are computed using 3.19. Recall the relation between $\mathbf{u}$ and $\bar{\mathbf{u}}$ (3.20 and 3.21), we get the structure in Figure 3.2 for the control law of the original system (2.24).

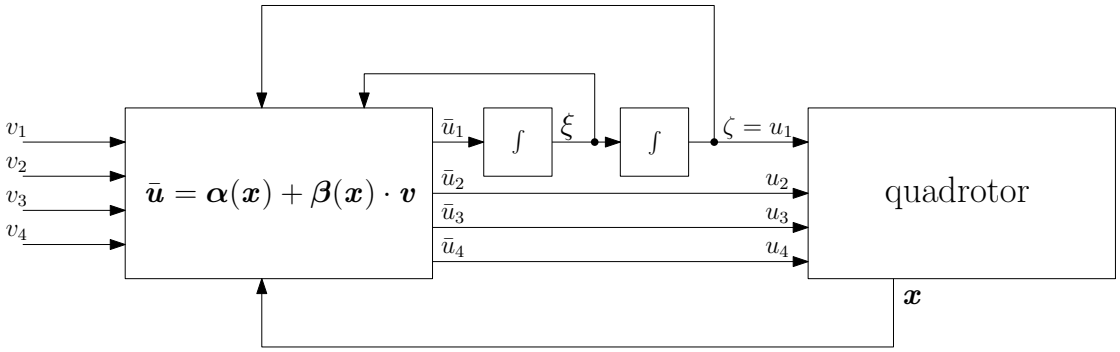


Figure 3.2: Block diagram of the control law.

Moreover, since the extended system (3.22) has dimension n = 14, the condition

$$r_1 + r_2 + r_3 + r_4 = n,$$

is fulfilled and, therefore, the system can be transformed via a dynamic feedback into a system which, in suitable coordinates, is fully linear and controllable. The change of coordinates $\mathbf{z} = \mathbf{\Phi}(\bar{\mathbf{x}})$ is given by

$$\begin{cases} z_1 = h_1(\mathbf{x}) = x \\ z_2 = L_f h_1(\mathbf{x}) = \dot{x} \\ z_3 = L_f^2 h_1(\mathbf{x}) = \ddot{x} \\ z_4 = L_f^3 h_1(\mathbf{x}) = x^{(3)} \\ z_5 = h_2(\mathbf{x}) = y \\ z_6 = L_f h_2(\mathbf{x}) = \dot{y} \\ z_7 = L_f^2 h_2(\mathbf{x}) = \ddot{y} \\ z_8 = L_f^3 h_2(\mathbf{x}) = y^{(3)} \\ z_9 = h_3(\mathbf{x}) = z \\ z_{10} = L_f h_3(\mathbf{x}) = \dot{z} \\ z_{11} = L_f^2 h_3(\mathbf{x}) = \ddot{z} \\ z_{12} = L_f^3 h_3(\mathbf{x}) = z^{(3)} \\ z_{13} = h_4(\mathbf{x}) = \psi \\ z_{14} = L_f h_4(\mathbf{x}) = \dot{\psi} \end{cases} \tag{3.27}$$

In the new coordinates, the system appears as

$$\begin{cases} \dot{\mathbf{z}} = \mathbf{A} \cdot \mathbf{z} + \mathbf{B} \cdot \mathbf{v} \\ \mathbf{y} = \mathbf{C} \cdot \mathbf{z} \end{cases} \tag{3.28}$$

where

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

$$\mathbf{v} = \begin{bmatrix} v_1 & v_2 & v_3 & v_4 \end{bmatrix}^T \in \mathbb{R}^4,$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{A_1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A_1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A_2} \end{bmatrix} \in \mathbb{R}^{14x14} \qquad \mathbf{B} = \begin{bmatrix} \mathbf{B_1} \\ \mathbf{B_2} \\ \mathbf{B_3} \\ \mathbf{B_4} \end{bmatrix} \in \mathbb{R}^{14x4}$$

$$\mathbf{C} = \begin{bmatrix} \mathbf{c_1}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{c_1}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{c_1}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{c_2}^T \end{bmatrix} \in \mathbb{R}^{4x14}$$

where

$$\mathbf{A}_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{4x4}, \qquad \mathbf{A}_2 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{2x2},$$

$$\mathbf{B}_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{4x4}, \qquad \mathbf{B}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{4x4},$$

$$\mathbf{B}_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \in \mathbb{R}^{4x4}, \qquad \mathbf{B}_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{2x4},$$

$$\mathbf{c}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T \in \mathbb{R}^4, \qquad \mathbf{c}_2 = \begin{bmatrix} 1 & 0 \end{bmatrix}^T \in \mathbb{R}^2,$$

In Figure 3.3 the scheme of the linear system is shown.

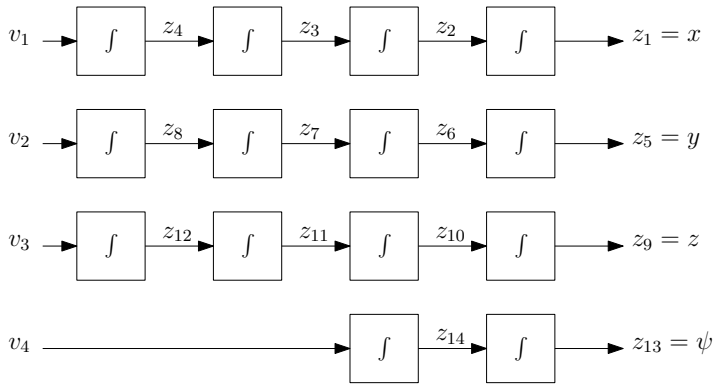


Figure 3.3: Block diagram of the closed loop system.

On the linear system (3.28) it is possible to impose a further feedback control, using for example the Linear Quadratic Regulator control.

3.2.2 Dynamic inversion with zero-dynamics stabilization

In this control there are two control loops, namely an inner loop that controls the height and attitude of the system, and an outer loop that controls the position. This solution was first suggested in [11]. Consider the inner loop of the quadrotor system as the height and attitude:

$$\mathbf{x_q} = \begin{bmatrix} z & \phi & \theta & \psi \end{bmatrix}^T. \tag{3.29}$$

For this system, the traditional feedback linearization methodology can be applied here with the output equation $\mathbf{y} = \mathbf{x_q}$. Picking this inner loop as the output equation, the first derivative would lead to

$$\dot{\mathbf{x}}_\mathbf{q} = \begin{bmatrix} \dot{z} & \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T. \tag{3.30}$$

Note here that the control input dependent term $L_g h(\mathbf{x})$ is zero. Consider the subsystem of the (2.22), it appear as,

$$\ddot{\mathbf{x}}_\mathbf{q} = \begin{bmatrix} \ddot{z} \\ \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \mathbf{b}(\mathbf{x}) + \mathbf{\Delta}(\mathbf{x}) \cdot \mathbf{u}, \tag{3.31}$$

where

$$\mathbf{b}(\mathbf{x}) = \begin{bmatrix} g \\ \dot{\theta}\dot{\psi}\frac{I_y - I_x}{I_x} \\ \dot{\phi}\dot{\psi}\frac{I_z - I_x}{I_y} \\ \dot{\phi}\dot{\theta}\frac{I_x - I_y}{I_z} \end{bmatrix}, \tag{3.32}$$

$$\mathbf{\Delta}(\mathbf{x}) = \begin{bmatrix} -\frac{1}{m}c_\theta c_\phi & 0 & 0 & 0 \\ 0 & \frac{1}{I_x} & 0 & 0 \\ 0 & 0 & \frac{1}{I_y} & \\ 0 & 0 & 0 & \frac{1}{I_y} \end{bmatrix}, \tag{3.33}$$

$$\mathbf{u} = \begin{bmatrix} f_t \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}, \tag{3.34}$$

A simplification is made by setting $\begin{bmatrix} \phi & \theta & \psi \end{bmatrix} = \begin{bmatrix} p & q & r \end{bmatrix}$. The assumption holds for smaller angles of movement. Based on the dynamics described in (3.31), the control input can be selected as (A.8) to be

$$\mathbf{u} = \boldsymbol{\alpha}(\mathbf{x}) + \boldsymbol{\beta}(\mathbf{x}) \cdot \mathbf{v}. \tag{3.35}$$

with

$$\begin{cases} \boldsymbol{\alpha}(\mathbf{x}) = -\boldsymbol{\Delta}^{-1}(\mathbf{x}) \cdot \mathbf{b}(\mathbf{x}) \\ \boldsymbol{\beta}(\mathbf{x}) = \boldsymbol{\Delta}^{-1}(\mathbf{x}) \end{cases} \tag{3.36}$$

The remaining linear dynamics after feedback linearization are an integrator chain, $\ddot{\mathbf{x}}_\mathbf{q} = \mathbf{v}$, which can be controlled by the linear controller:

$$\mathbf{v} = \ddot{\mathbf{x}}_\mathbf{q}^\mathbf{d} - \mathbf{K}_\mathbf{v} \cdot (\dot{\mathbf{x}}_\mathbf{q} - \dot{\mathbf{x}}_\mathbf{q}^\mathbf{d}) - \mathbf{K}_\mathbf{p} \cdot (\mathbf{x}_\mathbf{q} - \mathbf{x}_\mathbf{q}^\mathbf{d}), \tag{3.37}$$

where, $\ddot{\mathbf{x}}_\mathbf{q}^\mathbf{d}$ is the desired acceleration of the inner loop, $\mathbf{x}_\mathbf{q}^\mathbf{d}$ and $\dot{\mathbf{x}}_\mathbf{q}^\mathbf{d}$ are the desired trajectories for the position and their velocities. Finally, $\mathbf{K}_\mathbf{v}$ and $\mathbf{K}_\mathbf{p}$, positive definite matrix, are tunable gains that can be used to place the poles of the subsequent feedback linearized dynamics on the left hand side plane. Using the controller described in (3.35) and (3.37) the attitude of the controller can be stabilized. However, the zero dynamics (internal states x, y), namely states that are not observable from the output of the previous subsystem still remain uncontrolled. These dynamics according to (2.22) are:

$$\begin{cases} \ddot{x} = -\frac{u}{m}[s(\phi)s(\psi) + c(\phi)c(\psi)s(\theta)] \\ \ddot{y} = \frac{u}{m}[c(\psi)s(\phi) - c(\phi)s(\psi)s(\theta)] \end{cases} \tag{3.38}$$

Das et al. [11] suggests a method to control these dynamics by controlling the desired roll θ_d and pitch angle ϕ_d shown in (3.37) as part of $\mathbf{x}_\mathbf{q}^\mathbf{d}$. It consist about two simplifying assumptions: small oscillations and to impose $\psi = 0$. Consequently we obtain the following simplified system:

$$\begin{cases} \ddot{x} = -\frac{u}{m}\theta \\ \ddot{y} = \frac{u}{m}\phi \end{cases} \tag{3.39}$$

As before, there is a chain of two integrators to get to the desired position variables (x, y). Therefore, the same linear controller used in the inner loop control is applied to stabilize the outer loop dynamics,

$$\begin{cases} \theta_d = -\frac{m}{u}[\ddot{x}_d + k_{11}(\dot{x}_d - \dot{x}) + k_{12}(x_d - x)] \\ \phi_d = \frac{m}{u}[\ddot{y}_d + k_{11}(\dot{y}_d - \dot{y}) + k_{12}(y_d - y)] \end{cases} \tag{3.40}$$

Figure 3.4 shows the control scheme:

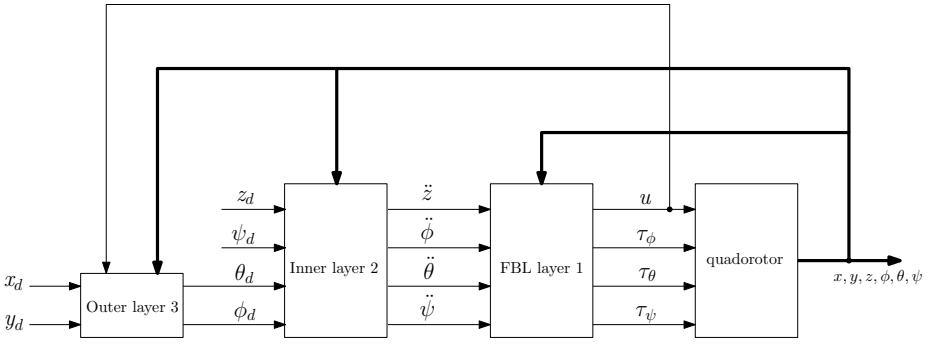


Figure 3.4: Dynamic inversion with zero-dynamics stabilization.

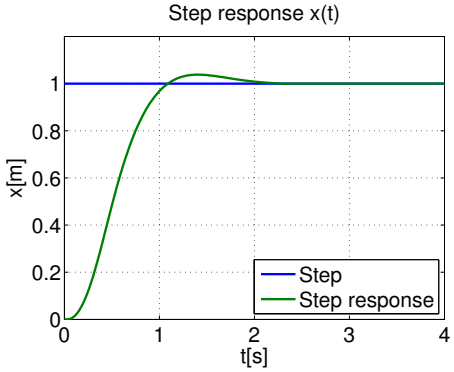
Chapter 4

Simulation results

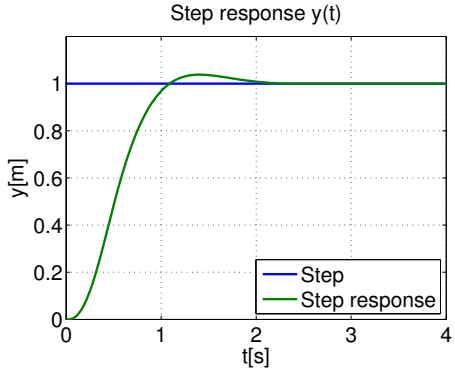
In this chapter we show the results obtained with MATLAB/SIMULINK and we analyze the differences between the several controllers illustrated above. For each control, we show the step response of the output variables x, y, z, ψ, then we show the double circle shape trajectory with the simulated results.

4.1 Linear Quadratic Regulator results

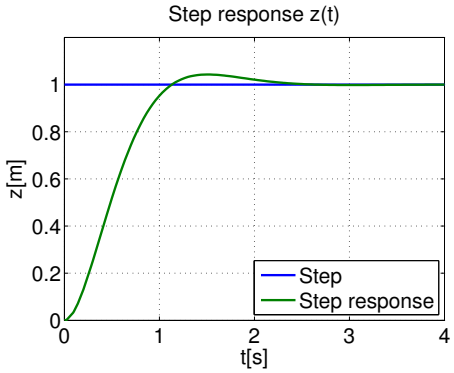
Figure 4.1 shows the step response of the output variables x, y, z, ψ when the Linear Quadratic Regulator control is used.



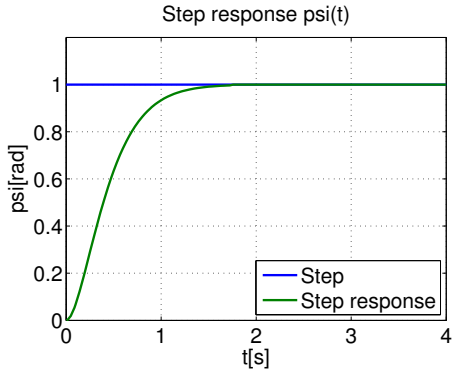
(a) $x(t)$ response to a step input.



(b) $y(t)$ response to a step input.



(c) $z(t)$ response to a step input.



(d) $\psi(t)$ response to a step input.

Figure 4.1: Position and yaw response of the Linear Quadratic Regulator to a step input.

The Table 4.1 shows some characteristic parameters of the step response when the linear quadrotor regolator control is used.

Table 4.1: Characteristic parameters to a step input.

	$\mathbf{x(t)}$	$\mathbf{y(t)}$	$\mathbf{z(t)}$	$\boldsymbol{\psi(t)}$
Rise time [s]	0.75	0.75	0.72	0.8
Overshoot [m]	4%	4%	4.3%	0%
Settling time [s]	2.3	2.3	2.6	1.75

Figure 4.2 shows the ideal and simulated trajectory for the infinite shape.

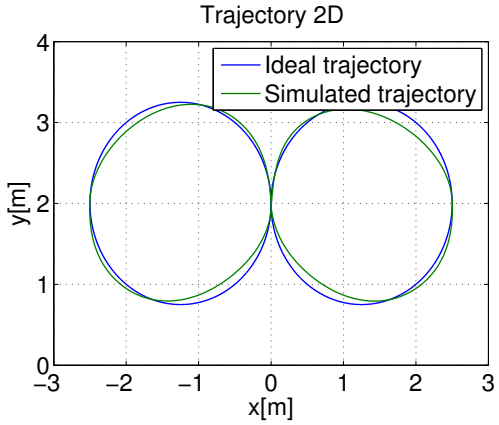


Figure 4.2: Comparing simulated and ideal trajectory with Linear Quadratic Regulator.

In the next figures are represented the time laws for the x, y variables.

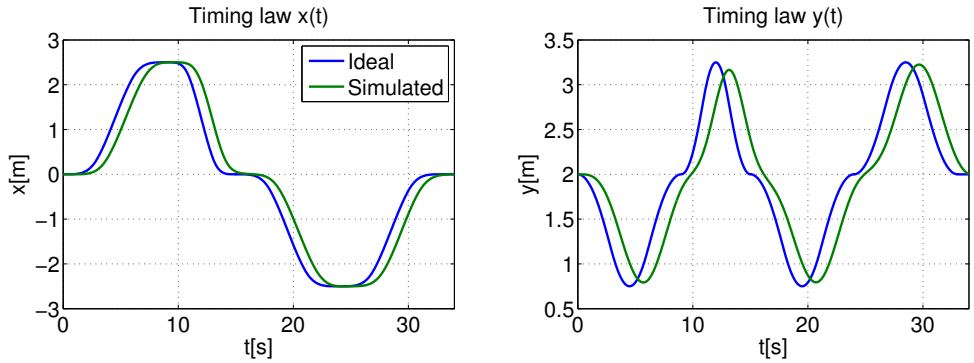
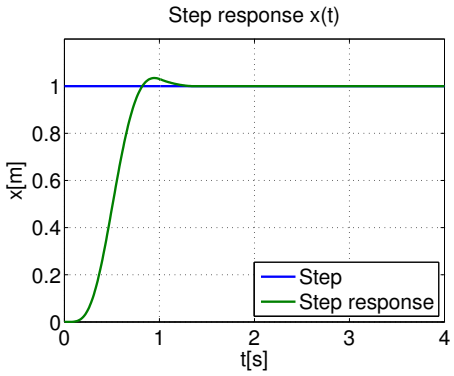


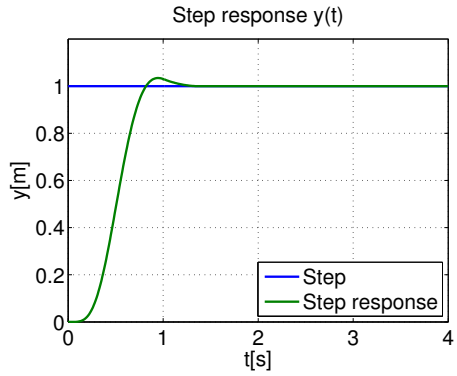
Figure 4.3: Ideal and simulated timing law with Linear Quadratic Regulator.

4.2 Exact linearization and non-interacting control via dynamic feedback results

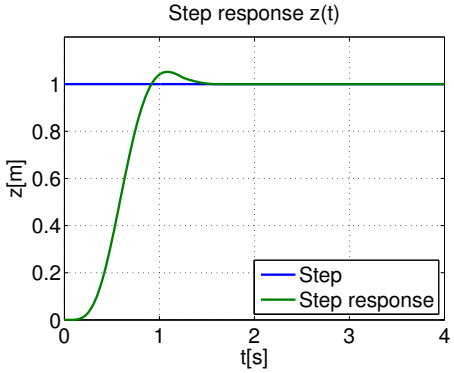
Figure 4.4 shows the step response of the output variables x, y, z, ψ when the exact linearization and non-interacting control via dynamic feedback is used.



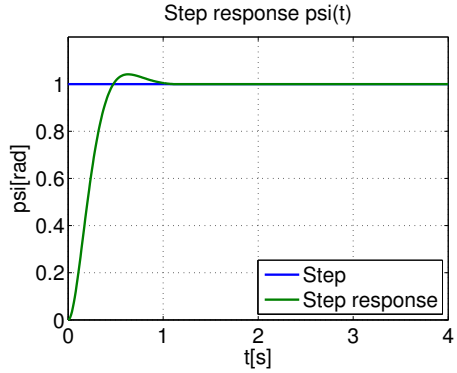
(a) $x(t)$ response to a step input.



(b) $y(t)$ response to a step input.



(c) $z(t)$ response to a step input.



(d) $\psi(t)$ response to a step input.

Figure 4.4: Position and yaw response of the exact linearization and non-interacting control via dynamic feedback to a step input.

Table 4.2 shows some characteristic parameters of the step response when the exact linearization and non-interacting control via dynamic feedback is used.

Table 4.2: Characteristic parameters to a step input.

	$\mathbf{x(t)}$	$\mathbf{y(t)}$	$\mathbf{z(t)}$	$\boldsymbol{\psi(t)}$
Rise time [s]	0.4	0.4	0.47	0.31
Overshoot [m]	4%	4%	5.2%	4.2%
Settling time [s]	1.3	1.3	1.55	1.7

Figure 4.5 shows the ideal and simulated trajectory for the double circle shape.

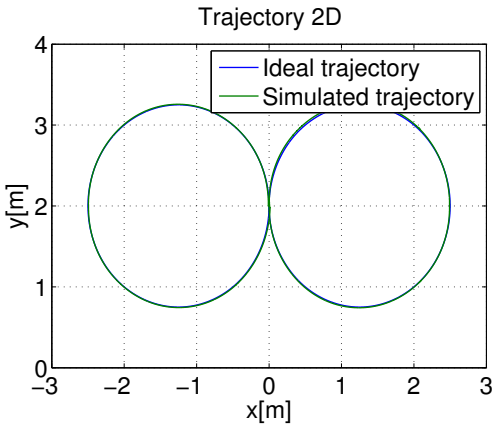


Figure 4.5: Comparing simulated and ideal trajectory with exact linearization and non-interacting control via dynamic feedback.

In the next figures are represented the time laws for the x, y variables.

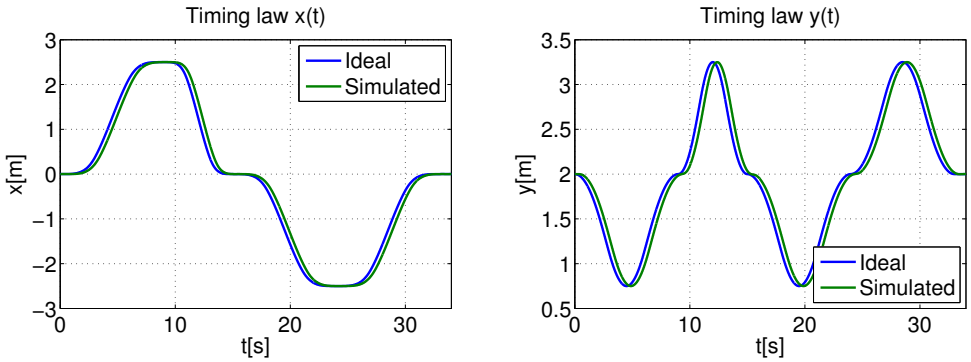
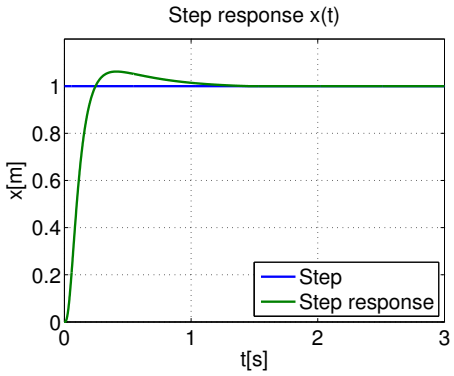


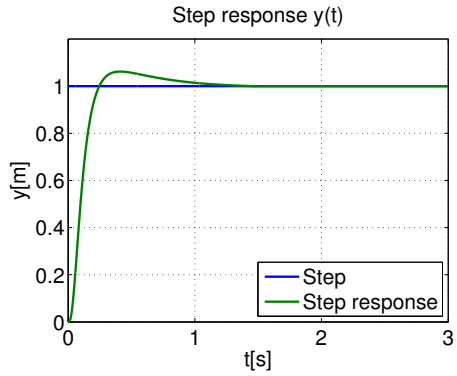
Figure 4.6: Ideal and simulated timing law with exact linearization and non-interacting control via dynamic feedback.

4.3 Dynamic inversion with zero-dynamics stabilization results

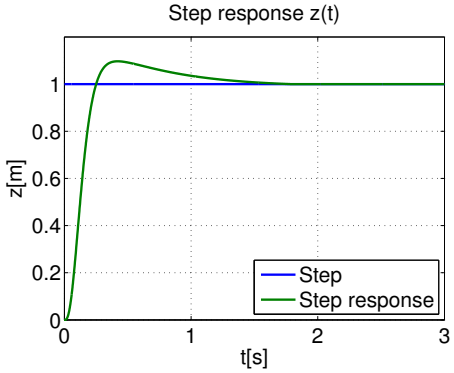
Figure 4.7 shows the step response of the output variables x, y, z, ψ when the dynamic inversion with zero-dynamics stabilization control is used.



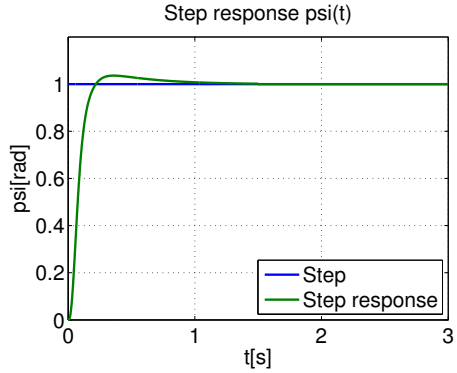
(a) $x(t)$ response to a step input.



(b) $y(t)$ response to a step input.



(c) $z(t)$ response to a step input.



(d) $\psi(t)$ response to a step input.

Figure 4.7: Position and yaw response of dynamic inversion with zero-dynamics stabilization to a step input.

Table 4.3 shows some characteristic parameters of the step response when the dynamic inversion with zero-dynamics stabilization control is used.

Table 4.3: Characteristic parameters to a step input.

	x(t)	**y(t)**	**z(t)**	**$\psi(t)$**
Rise time [s]	0.2	0.2	0.2	0.15
Overshoot [m]	7%	7%	9%	4%
Settling time [s]	1.4	1.4	1.7	1.5

Figure 4.8 shows the ideal and simulated trajectory for the double circle shape.

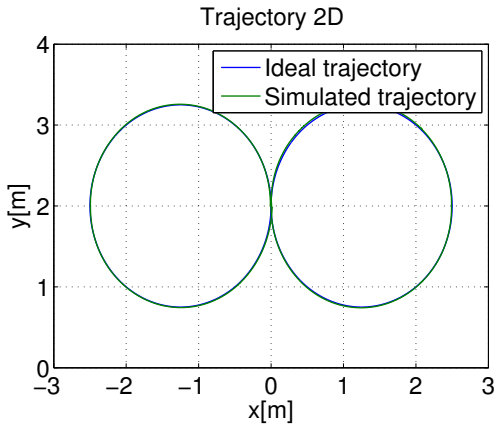


Figure 4.8: Comparing simulated and ideal trajectory with dynamic inversion with zero-dynamics stabilization.

In the next figures are represented the time laws for the x, y variables.

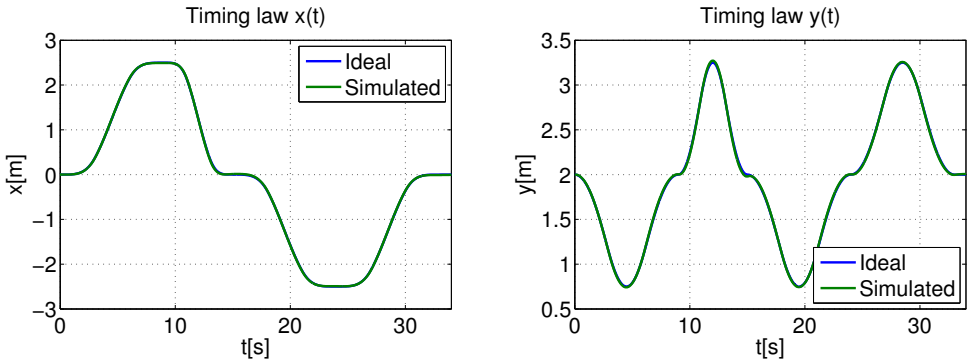
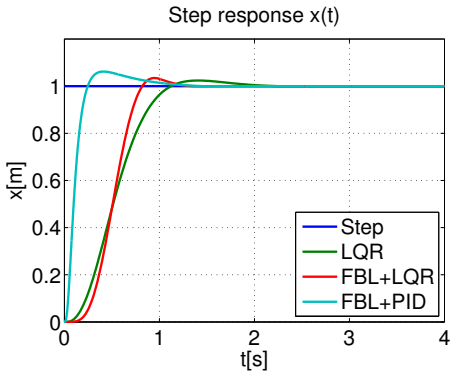


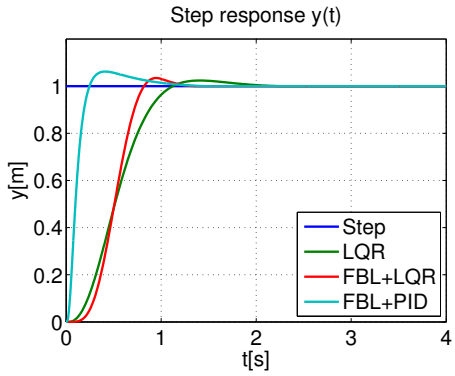
Figure 4.9: Ideal and simulated timing law with dynamic inversion with zero-dynamics stabilization.

4.4 Comparison

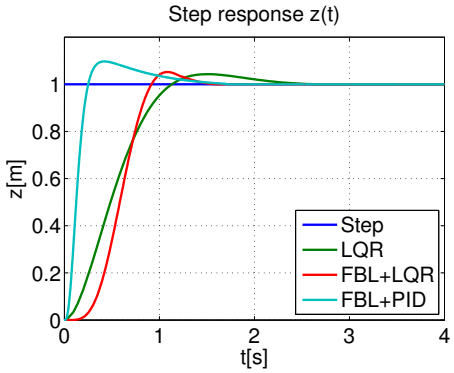
Figure 4.10 shows the step response of the output variables x, y, z, ψ when different controllers are used.



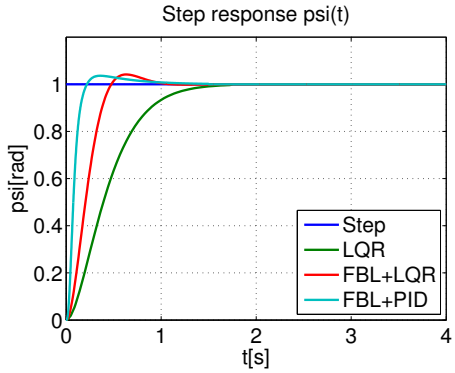
(a) $x(t)$ response to a step input with the three controllers.



(b) $y(t)$ response to a step input with the three controllers.



(c) $z(t)$ response to a step input with the three controllers.



(d) $\psi(t)$ response to a step input with the three controllers.

Figure 4.10: Position and yaw response of the three controls to a step input.

Tables 4.4-4.7 show some characteristic parameters of these step response where D-FBL stand for Dynamic Feedback Linearization and S-FBL stand for Static Feedback Linearization.

Table 4.4: Characteristic parameters to a step input for the $x(t)$ variable using the control strategies studied above.

x(t)	**LQR**	**D-FBL**	**S-FBL**
Rise time [s]	0.75	0.4	0.2
Overshoot [m]	4%	4%	7%
Settling time [s]	2.3	1.3	1.4

Table 4.5: Characteristic parameters to a step input for the $y(t)$ variable using the control strategies studied above.

y(t)	**LQR**	**D-FBL**	**S-FBL**
Rise time [s]	0.75	0.4	0.2
Overshoot [m]	4%	4%	7%
Settling time [s]	2.3	1.3	1.4

Table 4.6: Characteristic parameters to a step input for the $z(t)$ variable using the control strategies studied above.

z(t)	**LQR**	**D-FBL**	**S-FBL**
Rise time [s]	0.72	0.47	0.2
Overshoot [m]	4.3%	5.2%	9%
Settling time [s]	2.6	1.55	1.4

Table 4.7: Characteristic parameters to a step input for the $\psi(t)$ variable using the control strategies studied above.

$\boldsymbol{\psi(t)}$	**LQR**	**D-FBL**	**S-FBL**
Rise time [s]	0.8	0.31	0.15
Overshoot [m]	0%	4.2%	4%
Settling time [s]	1.75	1.7	1.7

From the data reported in these tables, we can make the following observation.

- The Linear Quadratic Regulator control is slower and it presents a low value of overshoot.
- The dynamic inversion with zero-dynamics stabilization control is faster but it presents a higher value of overshoot.
- The exact linearization and non-interacting control via dynamic feedback is slower than dynamic inversion with zero-dynamics stabilization control because, after the feedback linearization, the associated linear system presents the fourfold integrators.

4.5 3D Animation

The Simulink 3D Animation package provides apps for linking Simulink models and Matlab algorithms to 3D graphics objects. This package can be used to visualize and verify dynamic system behavior in a virtual reality environment. Objects are represented in the Virtual Reality Modeling Language

(VRML), a standard 3D modeling language. You can animate a 3D world by changing position, rotation, scale, and other object properties during desktop or real-time simulation [29]. In the following link, https://youtu.be/EXZJWUDSRus, the result of the simulation in SIMULINK 3D ANIMATION is shown, when the dynamic inversion with zero-dynamics stabilization control is used and the desired trajectory is the double circle shape. To realize the desired trajectory has been used a particular MATLAB script. Such MATLAB script realizes the desired trajectory using a trapezoidal velocity profile [31].

Chapter 5

Conclusions and future developments

In this work, firstly, a mathematical model of a quadrotor dynamics is derived, using Newton's and Euler's laws. Then a linearized version of the model is obtained, and therefore a linear controller, the Linear Quadratic Regulator, is derived. After that, two feedback linearization control schemes are designed. The first one is the dynamic inversion with zero dynamics stabilization, based on static feedback linearization obtaining a partial linearization of the mathematical model. The second one is the exact linearization and non-interacting control via dynamic feedback, based on dynamic feedback linearization obtaining a total linearization of the mathematical model. Moreover, these nonlinear control strategies are compared with the Linear Quadratic Regulator in terms of performances. Finally, the behavior of the quadrotor under the proposed control strategies is observed in virtual reality by using the Simulink 3D Animation toolbox. A future project could be the applying nonlinear control technique, like feedback linearization, adaptive feedback linearization, sliding mode control, for the real quadrotor to obtain best performance.

Appendix A

Input-output Feedback Linearization

Consider a system with state $\mathbf{x} \in \mathbb{R}^n$, input $u \in \mathbb{R}$ and output $y \in \mathbb{R}$ whose dynamics are given by

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})u \\ y = h(\mathbf{x}) \end{cases} \tag{A.1}$$

where $\mathbf{f}$, $\mathbf{g}$ and h are sufficiently smooth. We focus now on a single-input, single-output system, i.e., $u, y \in \mathbb{R}$. The derivative of the output y can be expressed as [17]

$$\dot{y} = \frac{\partial h}{\partial \mathbf{x}}[\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})u]. \tag{A.2}$$

The derivative of h along the trajectory of the state $\mathbf{x}$ is known as the *Lie Derivate* and is denoted as [17]

$$\dot{y} = \frac{\partial h}{\partial \mathbf{x}}[\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})u] = L_f h(\mathbf{x}) + L_g h(\mathbf{x})u. \tag{A.3}$$

If on the first derivative $L_g h(\mathbf{x}) = 0$, we have

$$\dot{y} = y^{(1)} = L_f h(\mathbf{x}). \tag{A.4}$$

Note that, in this case, the output y remains independent of input u. However, further higher order derivatives can be considered, specifically,

$$y^{(2)} = L_f^2 h(\mathbf{x}) + L_g L_f h(\mathbf{x}) u, \tag{A.5}$$

$$y^{(3)} = L_f^3 h(\mathbf{x}) + L_g L_f^2 h(\mathbf{x}) u, \tag{A.6}$$

$$y^{(i)} = L_f^i h(\mathbf{x}) + L_g L_f^{i-1} h(\mathbf{x}) u, \tag{A.7}$$

and if for a certain $i, L_g L_f^{i-1} h(\mathbf{x}) u \neq 0$, then the equation (A.7) can be linearized with full state feedback by,

$$u = \frac{1}{L_g L_f^{i-1} h(\mathbf{x})} (-L_f^i h(\mathbf{x}) + v), \tag{A.8}$$

in the state region where the inverse $\frac{1}{L_g L_f^{i-1} h(\mathbf{x})}$ exists. In this region the feedback linearized model becomes,

$$y^{(i)} = v. \tag{A.9}$$

The value i is defined to be the relative degree of the system. The resulting linear dynamic system defined in (A.9) can be stabilized by standarùd linear control techniques and consists of a set of $i-1$ integrators up to the required output y. Moreover, with this linearization a linear controller can be designed such that the overall system can be proven to be exponentially stable [21].
The concepts used for SISO systems can be also extended to MIMO systems [22]. In the MIMO case, we consider square systems (that is systems with the same number of inputs and outputs) of the form,

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x}) \cdot \mathbf{u} \\ \mathbf{y} = \mathbf{h}(\mathbf{x}) \end{cases}, \tag{A.10}$$

where $\mathbf{x} \in \mathbb{R}^n$ is a state vector, $\mathbf{u} \in \mathbb{R}^m$ is a control input vector (of components u_i), $\mathbf{y} \in \mathbb{R}^m$ is a vector of system output (of components y_i), $\mathbf{f}$, $\mathbf{h}$ are smooth vector fields, and $\mathbf{G}$ in an $n \times m$ matrix whose columns are smooth vector fields $\mathbf{g_i}$. Input-output linearization of MIMO systems is obtained similarly to the SISO case, by differentiating the outputs y_i until the inputs appear. Assume that r_i is the smallest integer such that at least one of the inputs appears in $y_i^{(r_i)}$ then,

$$y_i^{(r_i)} = L_f^{r_i} h_i(\mathbf{x}) + \sum_{j=1}^{m} L_{g_j} L_f^{r_i - 1} h_i(\mathbf{x}) u_j, \tag{A.11}$$

with $L_{g_j} L_f^{r_i - 1} h_i(\mathbf{x}) \neq 0$ for some $\mathbf{x}$. Performing the above procedure for each output y_i yields

$$\begin{bmatrix} y_1^{(r_1)} \\ \vdots \\ y_m^{(r_m)} \end{bmatrix} = \begin{bmatrix} L_f^{r_1} h_1(\mathbf{x}) \\ \vdots \\ L_f^{r_m} h_m(\mathbf{x}) \end{bmatrix} + \mathbf{E}(\mathbf{x})\mathbf{u}, \tag{A.12}$$

where the mxm matrix $\mathbf{E}(\mathbf{x})$ is defined obviously. If $\mathbf{E}(\mathbf{x})$ is invertible for all $\mathbf{x}$, then, similarly to the SISO case, the input transformation

$$\mathbf{u} = -\mathbf{E}^{-1} \cdot \begin{bmatrix} L_f^{r_1} h_1(\mathbf{x}) \\ \vdots \\ L_f^{r_m} h_m(\mathbf{x}) \end{bmatrix}, \tag{A.13}$$

yields m equations of the simple form

$$y_i^{(r_i)} = v_i. \tag{A.14}$$

Since the input v_i only affects the output y_i, (A.13) is called a *decoupling control law*, and the invertible matrix $\mathbf{E}(\mathbf{x})$ is called *decoupling matrix* of the system. The system (A.10), is then said to have *relative degree* $(r_1, ..., r_m)$, and the scalar

$r = r_1 + ... + r_m$ is called the *total relative degree* of the system.
An interesting case corresponds to the total relative degree being n. In this case, there are no internal dynamics. With the control law in the form of (A.13), we thus obtain an input-state linearization of the original nonlinear system. With the equivalent inputs v_i designed as in the SISO case, both stabilization and tracking can then be achieved for the system without any worry about the stability of the internal dynamics.

List of Figures

Bibliography

[1] http://krossblade.com/history-of-quadcopters-and-multirotors/

[2] http://www.fai.org/

[3] R.V. Jategaonkar. Flight vehicle system identification: a time domain methodology. American Institute of Aeronautics and Astronautics, 2006.

[4] I.D. Cowling, O.A. Yakimenko, J.F. Whidborne, and A.K. Cooke. A prototype of an autonomous controller for a quadrotor uav. In European Control Conference, pages 1–8, 2007.

[5] S. Bouabdallah, A. Noth, and R. Siegwart. PID vs LQ control techniques applied to an indoor micro quadrotor. In International Conference on Intelligent Robots and Systems 2004, volume 3, pages 2451 – 2456 vol.3, 2004.

[6] P. Pounds, R. Mahony, and P. Corke. Modelling and control of a quadrotor robot. In Australasian conference on robotics and automation 2006, Auckland, NZ, 2006.

[7] T. Madani and A. Benallegue. Backstepping Control for a Quadrotor Helicopter. In Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, pages 3255 –3260, 2006.

[8] S. Bouabdallah and R. Siegwart. Backstepping and sliding-mode techniques applied to an indoor micro quadrotor. In International Conference on Robotics and Automation 2005, pages 2247 – 2252, april 2005.

[9] R. Xu and U. Ozguner. Sliding mode control of a quadrotor helicopter. In Decision and Control, 2006 45th IEEE Conference on, pages 4957 –4962, dec. 2006.

[10] D. Lee, H. Jin Kim, and S. Sastry. Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter. International Journal of Control, Automation and Systems, 7(3):419–428, 2009.

[11] A. Das, K. Subbarao, and F. Lewis. Dynamic inversion with zero-dynamics stabilisation for quadrotor control. Control Theory & Applications, IET, 3(3):303–314, 2009.

[12] B. Whitehead and S. Bieniawski. Model Reference Adaptive Control of a Quadrotor UAV. In Guidance Navigation and Control Conference 2010, Toronto, Ontario, Canada, 2010. AIAA.

[13] M.Huang, B.Xian, C.Diao, K.Yang, and Y.Feng. Adaptive tracking control of underactuated quadrotor unmanned aerial vehicles via backstepping. In American Control Conference (ACC), 2010, pages 2076 –2081, 30 2010-july 2 2010.

[14] W. Zeng, B. Xian, C. Diao, Q. Yin, H. Li, and Y. Yang. Nonlinear adaptive regulation control of a quadrotor unmanned aerial vehicle. In Control Applications (CCA), 2011 IEEE International Conference on, pages 133 –138, 2011.

[15] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar. Design, modeling, estimation and control for aerial grasping and manipulation. In Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, pages 2668 –2673, sept. 2011.

[16] J.J. Craig, P. Hsu, and S.S. Sastry. Adaptive control of mechanical manipulators. The International Journal of Robotics Research, 6(2):16, 1987.

[17] Military Specification. Flying Qualities of Piloted Airplanes. Technical Report U.S. Military Specification MIL-F-8785C.

[18] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar. Design, modeling, estimation and control for aerial grasping and manipulation. In Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, pages 2668 –2673, sept. 2011.

[19] T. Bresciani, "Modelling, Identification and Control of a Quadrotor Helicopter", Master's thesis, Lund University, Sweden, 2008.

[20] Fondamenti di controlli automatici, P. Bolzern, R. Scattolini, N. Schiavoni

[21] H.K. Khalil and JW Grizzle. Nonlinear systems, volume 3. Prentice Hall New Jersey, 2002.

[22] Applied Nonlinear Control, Li, Slotine

[23] A. Isidori. Nonlinear Control Systems. Springer- Verlag, 1989

[24] H. Nijmeijer and A. van der Schaft. Nonlinear Dynamical Control Systems. Springer-Verlag, 1990.

[25] A.J. Fossard and D. Normand-Cyrot. Systmes on linaires Tome 3 Commande. Masson.

[26] B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo. Robotics. McGraw-Hill.

[27] D. Lee, T. Burg, D. Dawson, D. Shu, B. Xian, and E. Tatlicioglu, "Robust tracking control of an underactuated quadrotor aerial-robot based on a parametric uncertain model", in Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on, 2009, pp. 3187–3192.

[28] Michael A. Henson, Dale E. Seborg. Feedback linearinz control.

[29] Simulink 3D Animation: User's Guide, http://cn.mathworks.com/help/pdf_doc/sl3d/sl3d.pdf.

[30] M.Murray. Optimization-Based Control. California Institute of Technology

[31] S.Scotto di Clemente. Quadrotor control; implementation cooperation, and human-vehicle interaction.