People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University M'Hamed BOUGARA – Boumerdes

Department of Control and Power

Final Year Project Report Presented in Partial Fulfilment of
the Requirements for the Degree of

Master

In **CONTROL**
Option: **CONTROL**
Title:

Modeling and Control of Autonomous Quadrotor

Presented by:

- HAMDOUS Gaya
- BAHMED Kousseila

Supervisor
Dr. BOUSHAKI Razika

Academic Year: 2019/2020

Abstract

In this work, a detailed mathematical model for a quadrotor UAV is presented. The non-linear model has been derived using both Newton's and Euler's laws and then linearized around an equilibrium point. Three linear control techniques were developed to control the quadrotor's altitude, attitude, heading and position in space; a linear Proportional-Integral-Derivative, or PID, controller, a linear Proportional-Derivative, PD, Controller and a PD based Gain Scheduling Controller.

Ziegler-Nichols for tuning PID has been used to tune the PID and PD parameters and Genetic Algorithm has been used to tune the PD based Gain Scheduling controller. Simulation based experiments have been conducted using MATLAB/SIMULINK to evaluate and compare between the three developed approaches in term of dynamic performances, stability and disturbance effects.

Acknowledgements

We would like to express our deep gratitude to Dr. BOUSHAKI for his patient guidance, enthusiastic encouragement and useful critiques during this project work. We would also like to extend our thanks to everyone that helped us doing this humble work.

Contents

List of Figures

List of Tables

General Introduction

With the technological development, UAVs (abbreviation of Unmanned Aerial Vehicle) are becoming more and more needed and used in many fields as geography, topology, human assisting and so one. Hence, the need of more effective and efficient control methods.

An Unmanned Aerial Vehicle abbreviated UAV, also called drone is an aircraft that carries no pilot. Drone can be fully or partially autonomous but are more often controlled remotely by human. Several types of UAVs exist, among them Multirotor UAVs. A multirotor or multicopter is a rotorcraft with more than two lift-generating rotors. It is advanteged by its simple rotor mechanics required for flight control. Indeed, single-rotor and double-rotor aircrafts are more complex to control.

A quadrotor, or quadcopter[6] is composed of four rotors mounted at the ends of cross arm generally labeled from 1 through 4 as shown in Figure a. Motor 1 and 3 rotate in clockwise direction while motor 2 and 4 rotate in counterclockwise direction or vice versa. The center of mass attached to the quadrotor is also its origin and its center of the body-fixed frame B. Rotors 1, 2, 3 and 4 produce upward thrust F_1, F_2, F_3 and F_4, respectively. d is the length between the center of mass and center of rotor.

The quadrotor is a 6 degrees of freedom (DOF) object, meaning that 6 variables are required to express its position in space (x, y, z, ϕ, θ and ψ).

x, y and z represent the distance of the quadrotor's center of mass along the x, y and z axes respectively from an Earth fixed inertial frame while ϕ, θ and ψ are the three Euler angles representing the orientation of the quadrotor.

ϕ is called the roll angle which is the angle about the x-axis, θ is the pitch angle about the y-axis and ψ is the yaw angle about the z-axis.

The roll and pitch angles are said to be the attitude of the quadrotor, while the yaw angle is usually called the heading of the quadrotor. For the linear motion, the distance from the ground is referred to as the altitude and the x and y positions in space are often called the position of the quadrotor.

Quadrotor has many advantages over helicopter. Indeed, in the quadrotor, the rotor mechanics is simplified as it depends on four fixed pitch rotors unlike the variable pitch rotor in the helicopter. This leads to an easier manufacturing and maintenance. Moreover, the gyroscopic effects are reduced in quadrotors, due to the symmetry in the configuration, leading to much simpler control.

The presence of four propellers providing four thrust forces shifted with a fixed distance from the center of gravity instead of only one propeller centered in the middle as in the helicopter's structure leads to a more stable stationary hovering in quadrotors[10].

More advantages could be numerated as the vertical take-off and landing capabilities, better maneuverability and smaller size due to the absence of a tail[11], these capabilities make quadrotors useful in small area monitoring and buildings exploration[3].

On the other hand, the energy consumption is higher in quadrotors because of the presence of four propellers[3]. In addition to that, they have a large size and heavier than some of their counterparts again to the fact that there are four separate propellers.

The objective of this work is to derive a mathematical model for the quadrotor, utilize the model and existing estimation techniques to design and simulate a control system for the quadcopter. The reason behind choosing the quadrotor is, in addition to its advantages that will be addressed later, the research that is still facing challenges in the controlled; a quadrotor is a highly nonlinear, multivariable and underactuated system[7]. Underactuated systems are those having a smaller number of control inputs compared to the system's degrees of freedom (DOF). Indeed, a quadcopter or quadrotor is a six degrees of freedom object. However, it has only four actuators. This generates a real a very hard difficulty in controlling them due to the nonlinear coupling between the actuators and the degrees of freedom[10].

The most common fight control algorithms found in literature are linear fight controllers. These controllers can only perform when the quadrotor is around hover and under specific conditions. This makes them suffer from a huge performance degradation whenever the quadrotor leaves the nominal conditions required or in presence of disturbance[9].

Chapitre 1 Mathematical Modeling

In this chapter, a mathematical model for the quadcopter will be derived using physical lows governing the system. A mathematical model can be defined as a description of a system using mathematical concepts and language to facilitate proper explanation of a system or to study the effects of different components and to make predictions on patterns of behaviour[16].

1.1 Coordinate Frames

Before starting any analysis, the coordinate frames that will be used in order to study the kinematic of the system have to be first defined. Figure 1.1 shows the Earth reference frame G with N, E and D axis, and the body frame b with x, y and z axis.

The Earth frame is an inertial frame of reference that is fixed on a specific place at the ground level. The N-E-D axis notation is used to refer to North, East and Downward respectively.

The body frame is fixed at the center of mass of the quadcopter with x-axis pointing to the first propeller, the y-axis pointing to the second propeller and z-axis pointing to the ground.

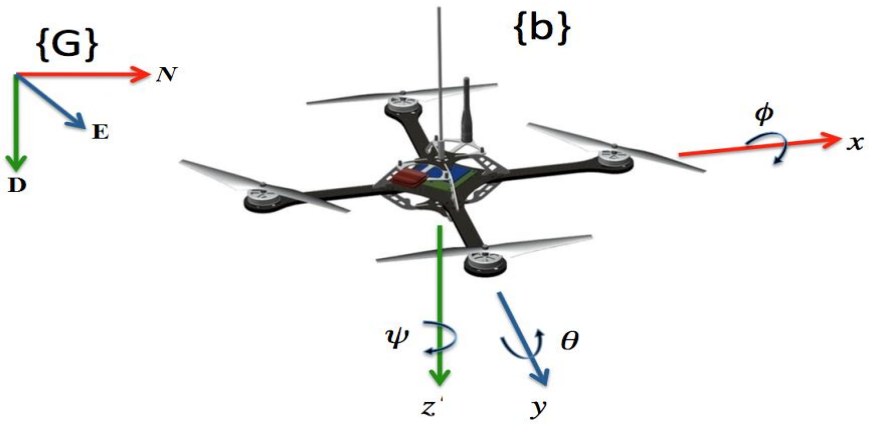

Figure 1.1: Coordinate frames and the Euler angles.

The distance between the Earth frame and the body frame is refered as the absolute position of the center of mass of the quadrotor $r = [x \quad y \quad z]^T$

The rotation R from the body frame to the inertial frame describes the orientation of the quadrotor. The orientation of the quadrotor is described using roll, pitch and yaw angles (ϕ, θ and ψ) representing rotations about the x, y and z-axes respectively.

Assuming the order of rotation to be roll (ϕ), pitch (θ) then yaw (ψ), the rotation matrix R, derived based on the sequence of principle rotations, is[18]:

$$R = \begin{bmatrix} co(\theta)co(\phi) & si(\psi)si(\theta)co(\phi) & co(\psi)si(\theta)co(\phi) + si(\phi)si(\psi) \\ co(\theta)si(\phi) & si(\psi)si(\theta)si(\phi) + co(\theta)co(\phi) & co(\psi)si(\theta)si(\phi) - si(\theta)co(\phi) \\ -si(\theta) & si(\psi)co(\theta) & co(\psi)co(\theta) \end{bmatrix} \quad (1.1)$$

Where co and si refer to $cosine$ and $sine$ respectively.

The rotation matrix R will be used in formulating the dynamics model of the quadrotor. Its significance is due to the fact that some states are measured in the body frame (e.g. the thrust forces produced by the propellers) while some others are measured in the inertial frame (e.g. quadrotor's position). Thus, to have a relation between both types of states, a transformation from one frame to the other is needed.

To relate the Euler rates $\dot{\eta} = [\dot{\phi} \quad \dot{\theta} \quad \dot{\psi}]^T$ that are measured in the inertial frame and angular body rates $\omega = [p \quad q \quad r]^T$, a transformation is needed as follows:

$$\omega = R_r \dot{\eta} \quad (1.2)$$

Where:

$$R_r = \begin{bmatrix} 1 & 0 & -sin(\theta) \\ 0 & cos(\phi) & sin(\phi)cos(\theta) \\ 0 & -sin(\phi) & cos(\phi)cos(\theta) \end{bmatrix} \quad (1.3)$$

Around the hover position, small angle assumption can be made where $cos(\theta) \approx 1$, $cos(\phi) \approx 1$, $sin(\theta) \approx 0$ and $sin(\phi) \approx 0$. Thus R_r can be simplified to the identity matrix I[14].

1.2 Dynamics of the Quadrotor

The quadrator's motion can be separated into two subsystems: a rotational subsystem (roll, pitch and yaw) and a translational subsystem (x, y and z positions). The rotational subsystem is fully actuated while the translational subsystem is underactuated[14].

1.2.1. Rotational Subsystem

The rotational equations of motion are derived in the body frame using the Newton-Euler method and have the following general formalism[17]:

$$J\dot{\omega} + \omega \times J\omega = M_B \tag{1.4}$$

Where:

J is the inertia matrix of the quadrotor

ω is the angular body rate

M_B is all the moments acting on the quadrotor in the body frame

1.2.2. Inertia Matrix J

The quadrotor's inertia matrix is a diagonal matrix, the off-diagonal elements, which are the product of inertia, are zero due to the symmetry of the quadrotor[8].

$$J = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{1.5}$$

Where I_{xx}, I_{yy} and I_{zz} are the area moments of inertia about the principle axes in the body frame.

1.2.3. Moments Acting on the Quadrotor M_B

Before defining the last term of equation (1.4), two physical effects have to be defined: the aerodynamic forces and moments produced by a rotor.

As an effect of rotation, the i^{th} rotor generates a force called the aerodynamic force F_i or the lift force (1.6) and a moment called the aerodynamic moment M_i (1.7)[1].

$$F_i = \frac{1}{2}\rho A C_T r^2 \Omega_i^2 \tag{1.6}$$

$$M_i = \frac{1}{2}\rho A C_D r^2 \Omega_i^2 \tag{1.7}$$

Where: ρ is the air density

A is the blade area

C_D, C_T are aerodynamic coefficients

r is the blade's radius

Ω_i is the angular velocity of the rotor i

It can be seen that the aerodynamic forces and moments depend clearly on the geometry of the propeller and the air density.

For the case of quadrotor, the maximum altitude is limited, thus the air density can be considered as constant. As a result, equations (1.6) and (1.7) can be simplified to the following[14]:

$$F_i = K_F \Omega_i^2 \tag{1.8}$$

$$M_i = K_M \Omega_i^2 \tag{1.9}$$

Where K_F and K_M are the aerodynamic force and moment constants respectively and Ω_i is the angular velocity of rotor i.

The moments M_B acting on the quadrotor can be studied by identifying the forces and moments generated by each propeller. Figure 1.2 shows the forces and moments acting on the quadrotor.

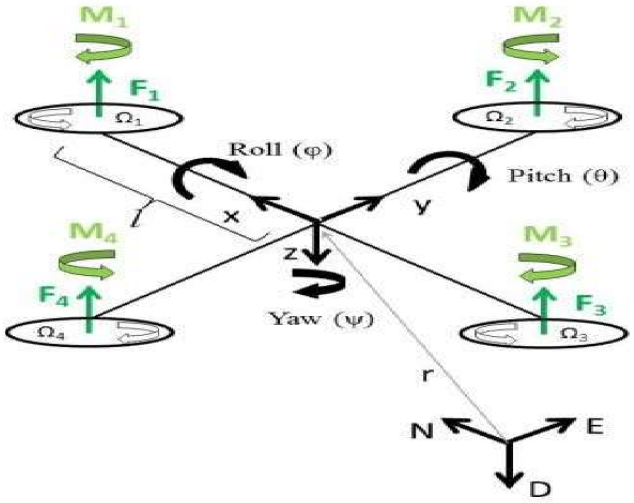

Figure 1.2: Forces and moments acting on the Quadrotor

Each rotor causes an upwards thrust force F_i and generates a moment M_i with an opposite direction with respect to the direction of rotation of the corresponding rotor i.

Starting with the moments about the body frame's x-axis. By using the right hand rule in association with the axes of the body frame, F_2 multiplied by the moment arm l generates a negative moment about the x-axis, while in the same manner, F_4 generates a positive moment. Hence, the total moment about the x-axis can be expressed as:

$$\begin{aligned} M_x &= -F_2 l + F_4 l = -lK_F\Omega_2^2 + lK_F\Omega_4^2 \\ &= lK_F(-\Omega_2^2 + \Omega_4^2) \end{aligned} \tag{1.10}$$

By the same manner, using the righ thand rule for the moments about the y-axis, the thrust of rotor 1 generates a positive moment, while the thrust of rotor 3 generates a negative moment about the y-axis. The total moment can be expressed as:

$$M_y = lK_F(\Omega_1^2 - \Omega_3^2) \tag{1.11}$$

Finally, about the body frame's z-axis, the thrust of the rotors does not cause any moment. However, moments caused by the rotors' rotation as per equation (1.7) do imply a moment on about the axis.

Using the right hand rule, the moments about the z-axis are expressed as:

$$\begin{aligned} M_z &= M_1 - M_2 + M_3 - M_4 \\ &= K_M\Omega_1^2 - K_M\Omega_2^2 + K_M\Omega_3^2 - K_M\Omega_4^2 \\ &= K_M(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \end{aligned} \tag{1.12}$$

The moment M_B is got by combining equations (1.10), (1.11)and (1.12) in vector form as follow:

$$M_B = \begin{bmatrix} lK_F(-\Omega_2^2 + \Omega_4^2) \\ lK_F(\Omega_1^2 - \Omega_3^2) \\ K_M(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \end{bmatrix} \tag{1.13}$$

Where l is the arm length or the distance between the axis of rotation of each rotor to the origin of the body reference frame which coincide with the center of mass of the quadrotor.

1.2.4. Translational Subsystem

The translational equations of motion are derived in the Earth referential frame using Newton's second law of motion and have the following formalism[14]:

$$m\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + RF_B \tag{1.14}$$

where:

R is the rotation matrix in equation (1.1)

$\ddot{r}$ is the time second derivative of the position vector r

F_B Nongravitational forces acting on the quadrotor in the body frame.

g is the gravitational acceleration

m is the mass of the quadrotor

1.2.5. Nongravitational Forces Acting on the Quadrotor

When the quadrotor is in a horizontal orientation (meaning that it is not rolling or pitching), the only nongravitational force acting on it is the thrust produced by the rotation of the propellers which is proportional to the square of the angular velocity of the propeller as per equation (1.6). Thus, the nongravitational forces F_B acting on the quadrotor, can be expressed as:

$$F_B = \begin{bmatrix} 0 \\ 0 \\ -K_F(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \end{bmatrix} \tag{1.15}$$

It can be remarked that the first two rows of the force vector are zeros as there is no forces in the x and y directions. The last row is simply an addition of the thrust forces produced by the four propellers. The negative sign is due to the fact that the thrust is upwards while the positive z-axis in the body framed is pointing downwards.

F_B is multiplied by the rotation matrix R to transform the thrust forces of the rotors from the body frame to the inertial frame, so that the equation can be applied in any orientation of the quadrotor.

1.3 Aerodynamic Effects

The aerodynamic effects acting on the quadrotor's body were excluded in the previous dynamics formulations. However, in order to have an accurate and realistic model to be used in simulations, aerodynamic effects should be included.

There are namely two types of aerodynamic effects, drag forces and drag moments[15].

1.3.1 Drag Forces

As the quadrotor body moves in air, a force resisting the motion acts on it due to the friction of the moving body. This force is proportional to the velocity of moving and can be expressed as:

$$F_A = K_T \dot{r} \tag{1.16}$$

Where:

K_T is a constant matrix called the aerodynamic translation coefficient matrix.

$\dot{r}$ is the time derivative of the position vector r.

Equation (1.15) should be rewritten as follow to indicate that there is an extra force acting on the quadrotor body.

$$m\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + RF_B - F_A \tag{1.17}$$

1.3.2 Drag Moments

As for the drag forces, due to the air friction, a drag moment M_A acts on the quadrotor body which can be expressed as[1]:

$$M_A = K_R \dot{\eta} \tag{1.18}$$

Where:

K_R is a constant matrix called the aerodynamic rotation coefficient matrix.

$\dot{\eta}$ is the Euler's rate.

Equation (1.4) should be rewritten as follow to indicate that there is an extra moment acting on the body.

$$J\dot{\omega} + \omega \times J\omega = M_B - M_A \tag{1.19}$$

1.4 State Space Representation

Formulating state space representation from the acquired mathematical equations will make the problem easier to handle.

1.4.1. State Vector X

The state vector of the quadrotor is defined to be:

$$X = [x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6 \quad x_7 \quad x_8 \quad x_9 \quad x_{10} \quad x_{11} \quad x_{12}]^T \tag{1.20}$$

This state vector is mapped to the degrees of freedom of the quadrotor in the following manner:

$$X = [\phi \quad \dot{\phi} \quad \theta \quad \dot{\theta} \quad \psi \quad \dot{\psi} \quad z \quad \dot{z} \quad x \quad \dot{x} \quad y \quad \dot{y}]^T \tag{1.21}$$

The state vector defines the position of the quadrotor in space and its angular and linear velocities.

1.4.2. Control Input Vector U

A control input vector consisting of four inputs U_1 through U_4 is defined to be:

$$U = [U_1 \quad U_2 \quad U_3 \quad U_4]^T \tag{1.22}$$

Where:

$$U_1 = K_F(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \quad (1.23a)$$
$$U_2 = K_F(-\Omega_2^2 + \Omega_4^2) \quad (1.23b)$$
$$U_3 = K_F(\Omega_1^2 - \Omega_3^2) \quad (1.23c)$$
$$U_4 = K_M(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \quad (1.23d)$$

U_1 is the resulting upwards force of the four rotors which is responsible for the altitude of the quadrotor and its rate of change (z,$\dot{z}$).

U_2 is the difference in thrust between the 2^{nd} and the 4^{th} rotor which is responsible for the roll rotation and its rate of change (ϕ, $\dot{\phi}$).

On the other hand, U_3 represents the difference in thrust between the 1^{st} and the 3^{rd} rotor, thus generating the pitch rotation and its rate of change (θ, $\dot{\theta}$).

Finally, U_4 is the difference in torque between the two clockwise turning rotors, and the two counterclockwise turning rotors generating the yaw rotation and ultimately its rate of change (ψ, $\dot{\psi}$).

The control vector U has been chosen so that the rotational system will be decoupled. Indeed, U_1 will generate the desired altitude of the quadrotor, U_2 will generate the desired roll angle, the desired pitch angle will be generated by U_3 whereas U_4 will generate the desired heading.

1.4.3. Rotational Subsystem

After substituting equations (1.23a) through (1.23d) in equation (1.13), the total moments acting on the quadrotor equation became:

$$M_B = [lU_2 \quad lU_3 \quad U_4]^T \quad (1.24)$$

Substituting equation (1.23d) into the rotational equation of motion (1.4) and expanding each term with their prior definition, the following relation can be derived:

$$\begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} lU_2 \\ lU_3 \\ U_4 \end{bmatrix} \quad (1.25)$$

Expanding that gives:

$$\begin{bmatrix} I_{xx}\ddot{\phi} \\ I_{yy}\ddot{\theta} \\ I_{zz}\ddot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\theta} I_{zz}\dot{\psi} - \dot{\psi} I_{yy}\dot{\theta} \\ \dot{\psi} I_{xx}\dot{\phi} - \dot{\phi} I_{zz}\dot{\psi} \\ \dot{\phi} I_{yy}\dot{\theta} - \dot{\theta} I_{xx}\dot{\phi} \end{bmatrix} = \begin{bmatrix} lU_2 \\ lU_3 \\ U_4 \end{bmatrix} \tag{1.26}$$

Rewriting the equations to have the angular accelerations in terms of the other variables and simplifying:

$$\begin{aligned} \ddot{\phi} &= b_1 U_2 + a_1 x_4 x_6 \\ \ddot{\theta} &= b_2 U_3 + a_2 x_2 x_6 \\ \ddot{\psi} &= b_3 U_4 + a_3 x_2 x_4 \end{aligned} \tag{1.27}$$

Where:

$$a_1 = \frac{I_{yy} - I_{zz}}{I_{xx}} \quad b_1 = \frac{l}{I_{xx}}$$
$$a_2 = \frac{I_{zz} - I_{xx}}{I_{yy}} \quad b_2 = \frac{l}{I_{yy}}$$
$$a_3 = \frac{I_{xx} - I_{yy}}{I_{zz}} \quad b_3 = \frac{l}{I_{zz}}$$

Clearly, the rotational subsystem is fully-actuated since it depends only on the rotational state variables x_1 to x_6 corresponding to ϕ, $\dot{\phi}$, θ, $\dot{\theta}$, ψ and $\dot{\psi}$ respectively.

1.4.4. Translational Subsystem

After substituting equations (1.23a) through (1.23d) in equation (1.15), the equation of the total forces acting on the quadrotor became:

$$F_B = [0 \quad 0 \quad -U_1]^T \tag{1.28}$$

After expanding and simplifying, the translational equation of motion (1.14) becomes then:

$$m\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + \begin{bmatrix} -(sin(\phi)sin(\psi) + cos(\phi)cos(\psi)sin(\theta))U_1 \\ (cos(\psi)sin(\phi) - cos(\phi)sin(\psi)sin(\theta))U_1 \\ -cos(\phi)cos(\theta)U_1 \end{bmatrix} \tag{1.29}$$

Rewriting equation (1.30) to have the accelerations in terms of the other variables.

$$\begin{aligned}\ddot{x} &= -\frac{U_1}{m}(sin(\phi)sin(\psi) + cos(\phi)cos(\psi)sin(\theta)) \\ \ddot{y} &= -\frac{U_1}{m}(cos(\phi)sin(\psi)sin(\theta) - cos(\psi)sin(\phi)) \\ \ddot{z} &= g - \frac{U_1}{m}cos(\phi)cos(\theta)\end{aligned} \tag{1.30}$$

In terms of the state vector X:

$$\begin{aligned}\ddot{x} &= -\frac{U_1}{m}(sin(x_1)sin(x_5) + cos(x_1)cos(x_5)sin(x_3)) \\ \ddot{y} &= -\frac{U_1}{m}(cos(x_1)sin(x_5)sin(x_3) - cos(x_5)sin(x_1)) \\ \ddot{z} &= g - \frac{U_1}{m}cos(x_1)cos(x_3)\end{aligned} \tag{1.31}$$

Clearly, the translational subsystem is underactuated since it dependent on both the translational and rotational state variables.

1.4.5. State Space Representation

Using equations (1.27) of rotational angular acceleration and equations (1.30) of translational acceleration, a complete mathematical model of the quadrotor can be written in state space form as follow:

$$\begin{aligned}\dot{x_1} &= \dot{\phi} = x_2 \\ \dot{x_2} &= \ddot{\phi} = a_1 x_4 x_6 + b_1 U_2 \\ \dot{x_3} &= \dot{\theta} = x_4 \\ \dot{x_4} &= \ddot{\theta} = a_2 x_2 x_6 + b_2 U_3 \\ \dot{x_5} &= \dot{\psi} = x_6 \\ \dot{x_6} &= \ddot{\psi} = a_3 x_2 x_4 + b_3 U_4 \\ \dot{x_7} &= \dot{z} = x_8 \\ \dot{x_8} &= \ddot{z} = g - \frac{U_1}{m}\cos(x_1)\cos(x_3) \\ \dot{x_9} &= \dot{x} = x_{10} \\ \dot{x_{10}} &= \ddot{x} = -\frac{U_1}{m}(\sin(x_1)\sin(x_5) + \cos(x_1)\cos(x_5)\sin(x_3)) \\ \dot{x_{11}} &= \dot{y} = x_{12} \\ \dot{x_{12}} &= \ddot{y} = -\frac{U_1}{m}(\cos(x_1)\sin(x_5)\sin(x_3) - \cos(x_5)\sin(x_1))\end{aligned} \tag{1.32}$$

$$f(X,U)=\begin{bmatrix} x_2 \\ a_1x_4x_6+b_1U_2 \\ x_4 \\ a_2x_2x_6+b_2U_3 \\ x_6 \\ a_3x_2x_4+b_3U_4 \\ x_8 \\ g-\frac{U_1}{m}\cos(x_1)\cos(x_3) \\ x_{10} \\ -\frac{U_1}{m}(\sin(x_1)\sin(x_5)+\cos(x_1)\cos(x_5)\sin(x_3)) \\ x_{12} \\ -\frac{U_1}{m}(\cos(x_1)\sin(x_5)\sin(x_3)-\cos(x_5)\sin(x_1)) \end{bmatrix} \tag{1.33}$$

Equation (1.33) can be more simplified and rewrote in control-affine form as follow:

$$\dot{X}=f(X)+\sum_{i=1}^{4} g_i(X)u_i \tag{1.34}$$

Where:

$$\begin{aligned} f(X) &= [x_2 \quad a_1x_4x_6 \quad x_4 \quad a_2x_2x_4 \quad x_6 \quad a_3x_2x_4 \quad x_8 \quad g \quad x_{10} \quad 0 \quad x_{12} \quad 0]^T \\ g_1 &= [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad G_1 \quad 0 \quad G_2 \quad 0 \quad G_3]^T \in \mathbb{R}^{12} \\ g_2 &= [0 \quad b_1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T \in \mathbb{R}^{12} \\ g_3 &= [0 \quad 0 \quad 0 \quad b_2 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T \in \mathbb{R}^{12} \\ g_4 &= [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad b_3 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T \in \mathbb{R}^{12} \end{aligned} \tag{1.35}$$

With:

$$\begin{aligned} G_1 &= -\frac{1}{m}(cos(x_1)cos(x_3)) \\ G_2 &= \frac{1}{m}(sin(x_1)sin(x_5)+cos(x_1)cos(x_5)sin(x_3)) \\ G_3 &= \frac{1}{m}(cos(x_1)sin(x_5)sin(x_3)-cos(x_5)sin(x_1)) \end{aligned} \tag{1.36}$$

1.5 Linearized Model

In order to linearize the model obtained in equation (1.33), an equilibrium point X_e for a fixed input U_e is needed. The linearized model will be valid only in the neighborhood of that equilibrium point (EP).

A point X_e for a fixed input U_e is said to be an equilibrium point if and only if:

$$f(X_e, U_e) = 0 \tag{1.37}$$

Because of the nonlinearity of the function f, it is difficult to find a solution for equation (1.38). For this system in hand particularly, the solution is difficult to find because of trigonometric functions that are related each other in no-elementary way.

To get ride of this difficulty, the linearization is performed on a simplified model that considers only small oscillations, meaning that the sine function is approximated by its argument and the cosine function by the unity. The approximation is valid only if the argument is small and the resulting system is as follow[4]:

$$\hat{f}(X,U) = \begin{bmatrix} x_2 \\ a_1 x_4 x_6 + b_1 U_2 \\ x_4 \\ a_2 x_2 x_6 + b_2 U_3 \\ x_6 \\ a_3 x_2 x_4 + b_3 U_4 \\ x_8 \\ g - \frac{U_1}{m} \\ x_{10} \\ -\frac{U_1}{m}(x_1 x_5 + x_3) \\ x_{12} \\ -\frac{U_1}{m}(x_5 x_3 - x_1) \end{bmatrix} \tag{1.38}$$

5.1.1. Linearization

The resulting equilibrium point from the equations above is:

$$X_e = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad z \quad 0 \quad x \quad 0 \quad y \quad 0]^T \in \mathbb{R}^{12} \tag{1.39}$$

From equation (1.40), the equilibrium point is obtained by the constant input vector:

$$U_e = [mg \quad 0 \quad 0 \quad 0]^T \in \mathbb{R}^4 \tag{1.40}$$

It can be noticed that this particular input vector U_e represents the force necessary to delete the quadrotor's weight and it consents its hovering.

After finding the equilibrium point X_e and the corresponding nominal vector U_e, the linear model can be found in the form:

$$\dot{X} = AX + BU \tag{1.41}$$

Such that:

$$A = \frac{\partial f(X,U)}{\partial X}\Big|_{\substack{X=X_e \\ U=U_e}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{1.42}$$

$$B = \frac{\partial f(X,U)}{\partial U}\Big|_{\substack{X=X_e \\ U=U_e}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & b_1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & b_2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & b_3 \\ 0 & 0 & 0 & 0 \\ \frac{-1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{1.43}$$

5.1.2. Controllability

One of major concepts of modern control system theory is controllability. Introduced by R. Kalman in 1960, it is defined as follow:

"The pair (A, B) is said to be controllable if for any initial state $X(0) = X_i \in \mathbb{R}^n$ and any final state $X_f \in \mathbb{R}^n$, there exists an input that transfers X_i to X_f in a finite time. Otherwise (A, B) is said to be uncontrollable[4]."

A necessary and sufficient condition for (A, B) to be controllable is[4]:

$$rank(w_c) = n \tag{1.44}$$

Where w_c is called Kalman's controllability matrix of size $n \times nm$ and it is this case it is equal to:

$$w_c = [B \quad AB \quad A^2B \quad A^3B \quad A^4B \quad A^5B \quad A^6B \quad A^7B \quad A^8B \quad A^9B \quad A^{10}B \quad A^{11}B] \in \mathbb{R}^{12\times 48}$$

MatLab has been used to check the rank of matrix w_c and it has been found to be of full rank. The system represented by equation (1.41) is controllable.

Chapitre 2 PID Based Controller

In this chapter, the formulated quadrotor model has been used in control design. The first controllers that has been developed is the Proportional-Integral-Derivative or PID controller. A Computer based simulation has been implemented on MATLAB/Simulink and has been used to assess the performance of the developed controller.

2.1. Overview

The proportional Integral derivative (PID) design are pointed out in many references that PID controllers can be used only for plants with relatively small-time delay for high performance devices like the quadrotor [5]. This controller takes many structures but the traditional one as in the following form:

$$u(t) = k_p \left[e(t) + \frac{1}{\tau_i} \int_0^t e(\tau) d\tau + \tau_d \frac{de(t)}{dt} \right] \quad (2.1)$$

Where $u(t)$ is the input signal, $e(t) = r(t) - y(t)$ is the error signal such that $r(t)$ is the reference signal and $y(t)$ is the output signal.

k_p is the proportional gain of the PID, τ_i and τ_d represent the integration time and the derivative time respectively.

2.2. PID Controller

The closed-loop configuration that has been used to design the PID controller is discussed in this section.

2.2.1. Altitude Controller

The altitude controller takes an error signal e as an input which is the difference between the desired altitude z_d and the actual altitude z and produces a control signal U_1. The block diagram of the altitude controller is shown in figure 2.1.

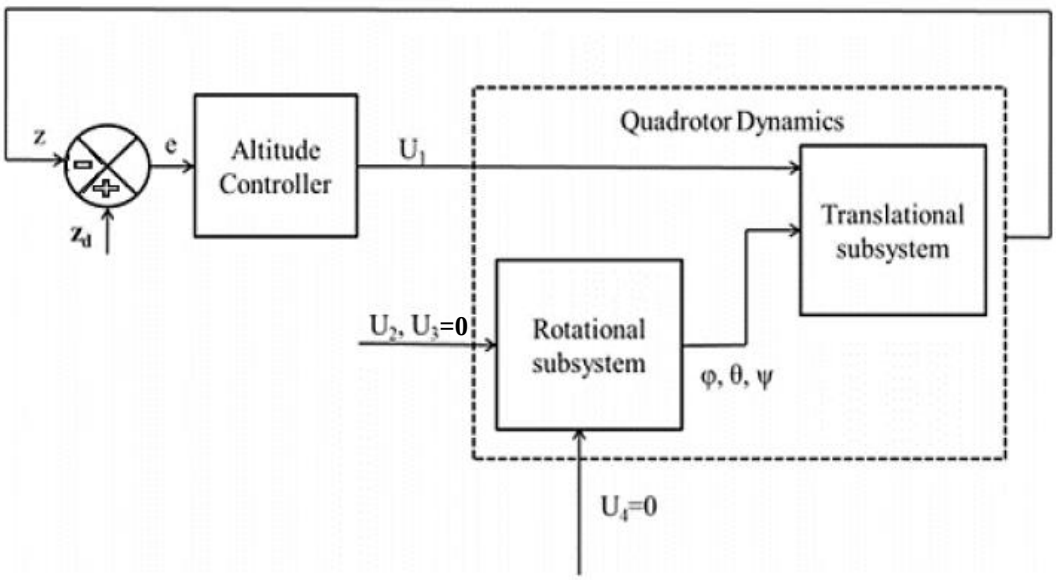

Figure 2.1: Block diagram of the altitude controller.

2.2.2. Attitude and Heading Controller

Similar to the altitude controller, the attitude and heading controller takes as an input an error signal e which is the difference between the desired pitch, roll and yaw angles (θ_d, ϕ_d and ψ_d respectively) and their respective actual values θ, ϕ and ψ. It generates the control signals U_2, U_3 and U_4 as represented in the block diagram shown in figure 2.2.

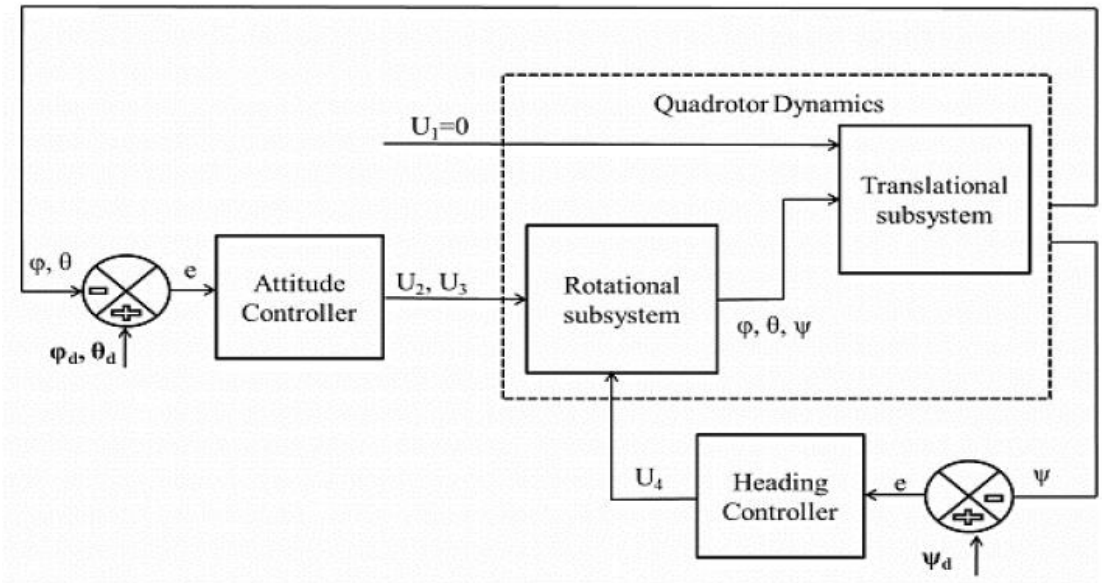

Figure 2.2: Block diagram of the attitude and heading controller.

2.2.3. Position Controller

Since the x and y position is not decoupled, it cannot be directly controlled using the four control signals U_1 through U_4. However, the x and y position can be controlled through the roll ϕ and pitch θ angles.

Using the translational equations of motion, equation (1.13) and (1.14), the desired roll ϕ_d and pitch θ_d angles can be calculated as follow:

$$\begin{aligned} \ddot{x} &= -\frac{U_1}{m}\big(sin(\phi)sin(\psi) + cos(\phi)cos(\psi)sin(\theta)\big) \\ \ddot{y} &= -\frac{U_1}{m}\big(cos(\phi)sin(\psi)sin(\theta) - cos(\psi)sin(\phi)\big) \end{aligned} \tag{2.2}$$

Since the quadrotor is operating around hover, meaning that small values can be assumed for the roll ϕ and pitch θ angles. Small angle assumption is used to simplify the above equations to:

$$\begin{aligned} \ddot{x} &= -\frac{U_1}{m}\big(\phi_d sin(\psi) + \theta_d cos(\psi)\big) \\ \ddot{y} &= -\frac{U_1}{m}\big(\theta_d sin(\psi) - \phi_d cos(\psi)\big) \end{aligned} \tag{2.3}$$

Which can be written in matrix from as:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = -\frac{U_1}{m}\begin{bmatrix} -\sin(\psi) & -\cos(\psi) \\ \cos(\psi) & -\sin(\psi) \end{bmatrix}\begin{bmatrix} \phi_d \\ \theta_d \end{bmatrix} \tag{2.4}$$

After some manipulations to get the desired roll ϕ and pitch θ in terms of the desired position x and y:

$$\begin{bmatrix} \phi_d \\ \theta_d \end{bmatrix} = \frac{m}{U_1}\begin{bmatrix} -\ddot{x_d}sin(\psi) + \ddot{y_d}cos(\psi) \\ -\ddot{x_d}cos(\psi) - \ddot{y_d}sin(\psi) \end{bmatrix} \tag{2.5}$$

To fulfill the small angle assumption, the roll ϕ and pitch θ angles have to be kept in the range between $-20°$ to $20°$[19].

Including the position controller in the previous block diagram, the overall block diagram of the whole system is shown in figure 2.3.

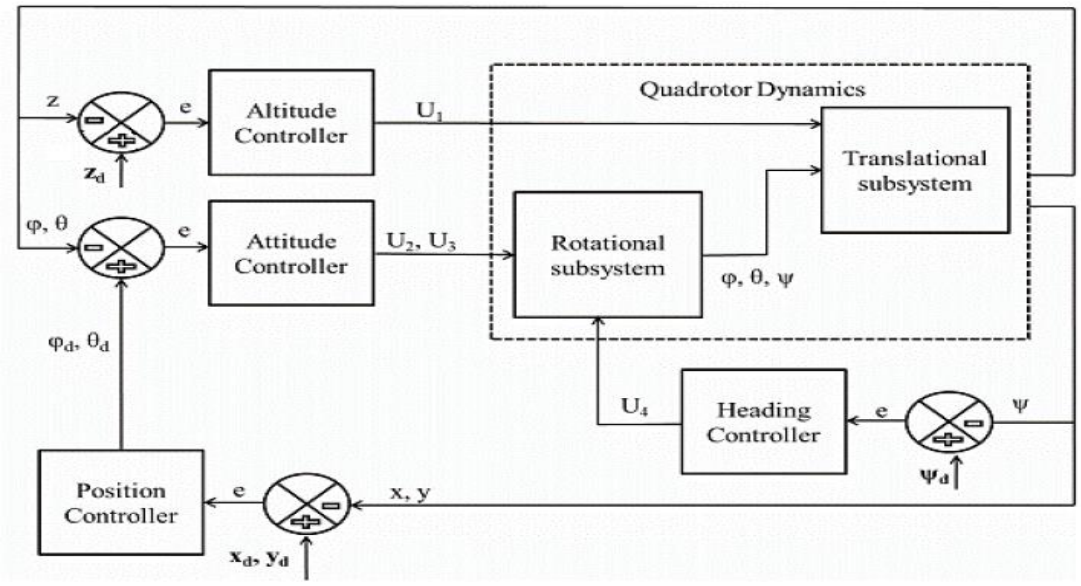

Figure 2.3: Block diagram of the position controller.

2.3. Control Laws

The developed PID controller generates the desired control inputs for the quadrotor. The block diagram for a PID controller is shown in figure 2.4.

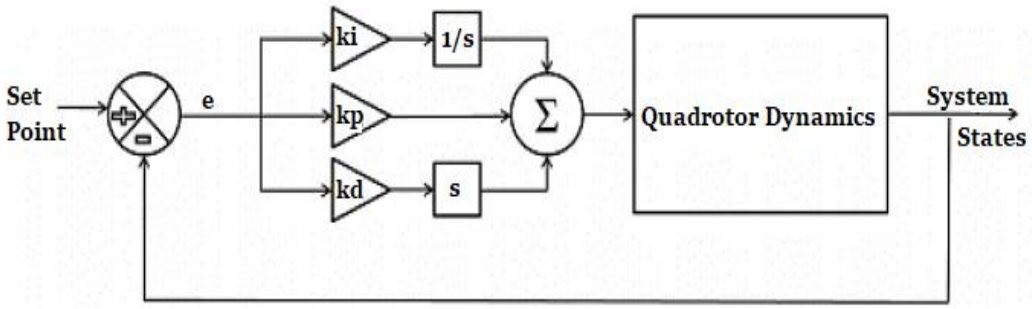

Figure 2.4: Block diagram of a PID controller.

k_p, k_d and k_i are the proportional, derivative and integral gains, respectively where:

$$k_d = k_p \tau_d \tag{2.6}$$

$$k_i = \frac{k_p}{\tau_i} \tag{2.7}$$

2.3.1 Altitude Controller

The altitude PID controller generates the control signal U_1 that is responsible of the altitude of the quadrotor as per equation (1.26a). The derived control law is as follow:

$$U_1 = k_p(z - z_d) + k_d(\dot{z} - \dot{z_d}) + k_i\int (z - z_d)dt \tag{2.8}$$

Where:

z_d is the desired altitude.

$\dot{z_d}$ is the desired altitude rate of change.

2.3.2 Attitude and Heading Controller

Roll Controller

The derived control law generates the input U_2 that controls the roll angle ϕ as follows:

$$U_2 = k_p(\phi - \phi_d) + k_d(\dot{\phi} - \dot{\phi_d}) + k_i\int (\phi - \phi_d)dt \tag{2.9}$$

Where:

ϕ_d is the desired roll angle.

$\dot{\phi_d}$ is the desired roll angle rate of change.

Pitch Controller

The derived control law generates the input U_3 that controls the pitch angle θ as follows:

$$U_3 = k_p(\theta - \theta_d) + k_d(\dot{\theta} - \dot{\theta_d}) + k_i\int (\theta - \theta_d)dt \tag{2.10}$$

Where:

θ_d is the desired pitch angle.

$\dot{\theta_d}$ is the desired pitch angle rate of change.

Yaw Controller

The derived control law generates the input U_4 that controls the yaw angle ψ as follows:

$$U_4 = k_p(\psi - \psi_d) + k_d(\dot{\psi} - \dot{\psi_d}) + k_i\int(\psi - \psi_d)dt \tag{2.11}$$

Where:

ψ_d is the desired yaw angle.

$\dot{\psi_d}$ is the desired yaw angle rate of change.

Position Controller

After acquiring stable controllers for the altitude and the attitude of the quadrotor, a complete position controller has been developed.

PID controllers are used to calculate the desired accelerations $\ddot{x}$ and $\ddot{y}$.

$$\begin{aligned} \ddot{x} &= k_p(x - x_d) + k_d(\dot{x} - \dot{x_d}) + k_i\int(x - x_d)dt \\ \ddot{y} &= k_p(y - y_d) + k_d(\dot{y} - \dot{y_d}) + k_i\int(y - y_d)dt \end{aligned} \tag{2.12}$$

Where:

x_d is the desired x position.

$\dot{x_d}$ is the desired x position rate of change.

y_d is the desired y position.

$\dot{y_d}$ is the desired y position rate of change.

Plugging the values of the desired accelerations $\ddot{x_d}$ and $\ddot{y_d}$ into equation (2.4), the desired roll angle ϕ_d and pitch angle θ_d can be calculated which are in turn fed into the attitude controller previously expressed in equations (2.6) and (2.7).

2.4. PID Controller Simulation

The Ziegler-Nichols method for tuning PID controllers was used in order to set the PID gains.

The Ziegler-Nichols method is a heuristic method for tuning a PID controller. It is an approach to problem solving that employs a practical method that is not guaranteed to be optimal, perfect, or rational, but is nevertheless sufficient for reaching an immediate, short-term goal. It is used to speed up the process of finding a satisfactory solution.

The system has been tested with and without disturbances (wind). The disturbance has been added in the form of additional forces and moments to the right side of the system's translational and rotational equations of motion (1.4) and (1.14).

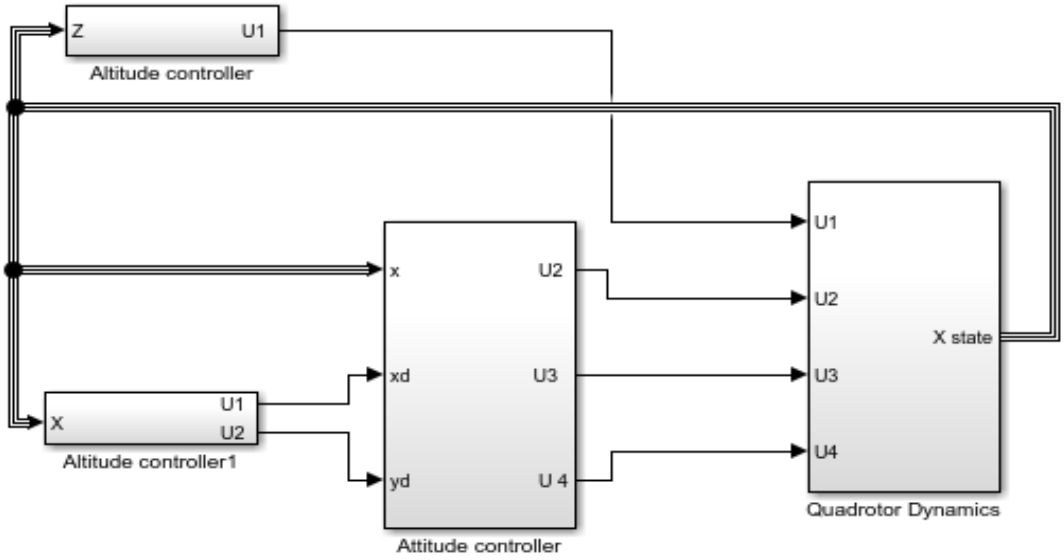

Figure 2.5: Block diagram of the Simulink model.

Table 2.1: Parameters of the PID Controller

	Desired value	k_p	k_d	k_i	Settling time (s)	Overshoot (%)
Altitude (z)	3.5	6.5	2.9	3.1	2.5	4.66
Position (x and y)	3.5	3.5	1.33	0.12	3	0
Attitude (θ and ϕ)	0	7.0	4.9	0	3.6	/
Heading (ψ)	5	5.1	1.12	0	0.96	/

2.4.1. PID Simulation Without Disturbance

Figures 2.6, 2.7 and 2.8 show the position responses, control inputs and the trajectory, respectively, in ideal conditions (without disturbances).

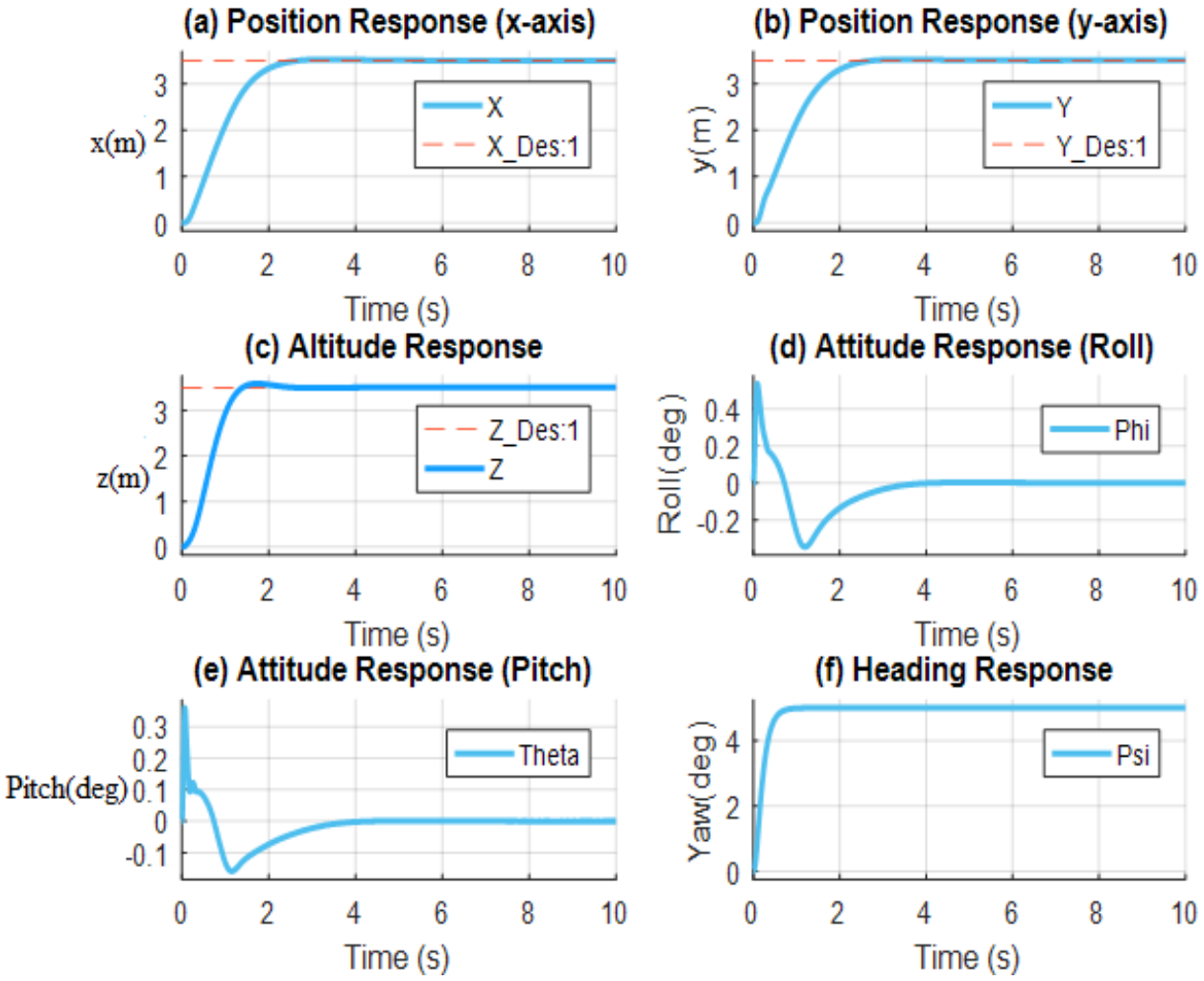

Figure 2.6: Position responses of the PID controller.

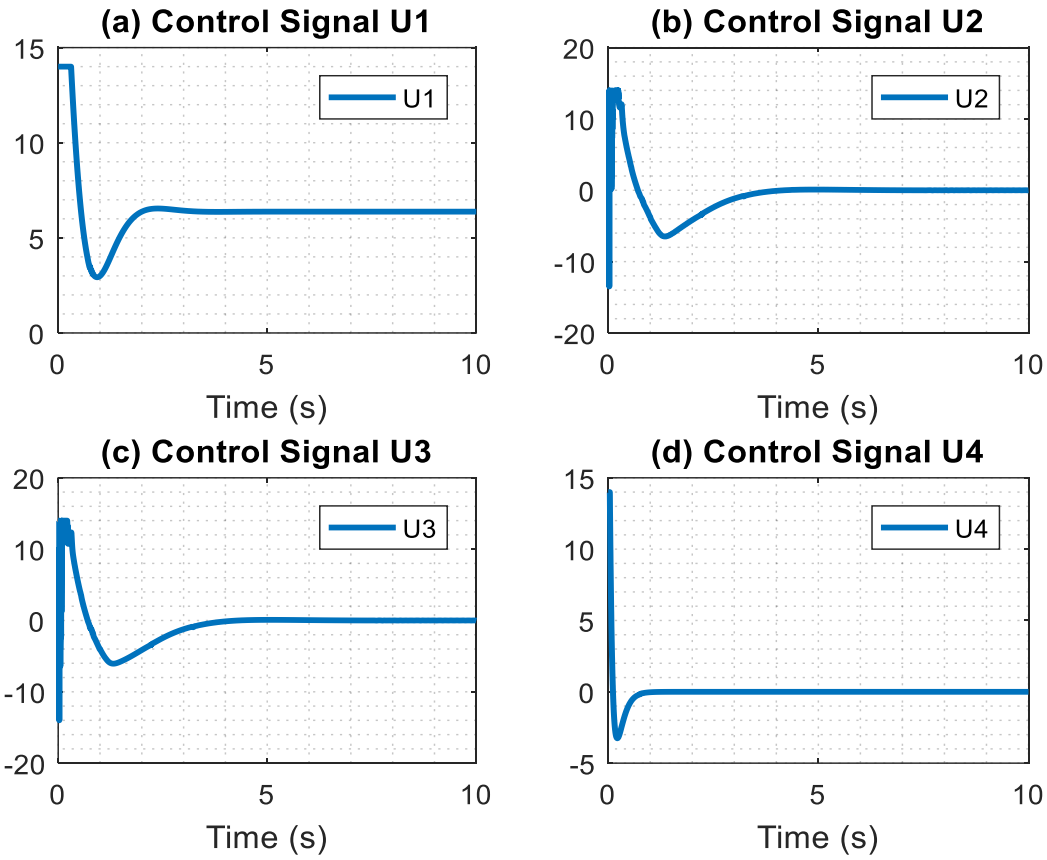

Figure 2.7: Control inputs responses.

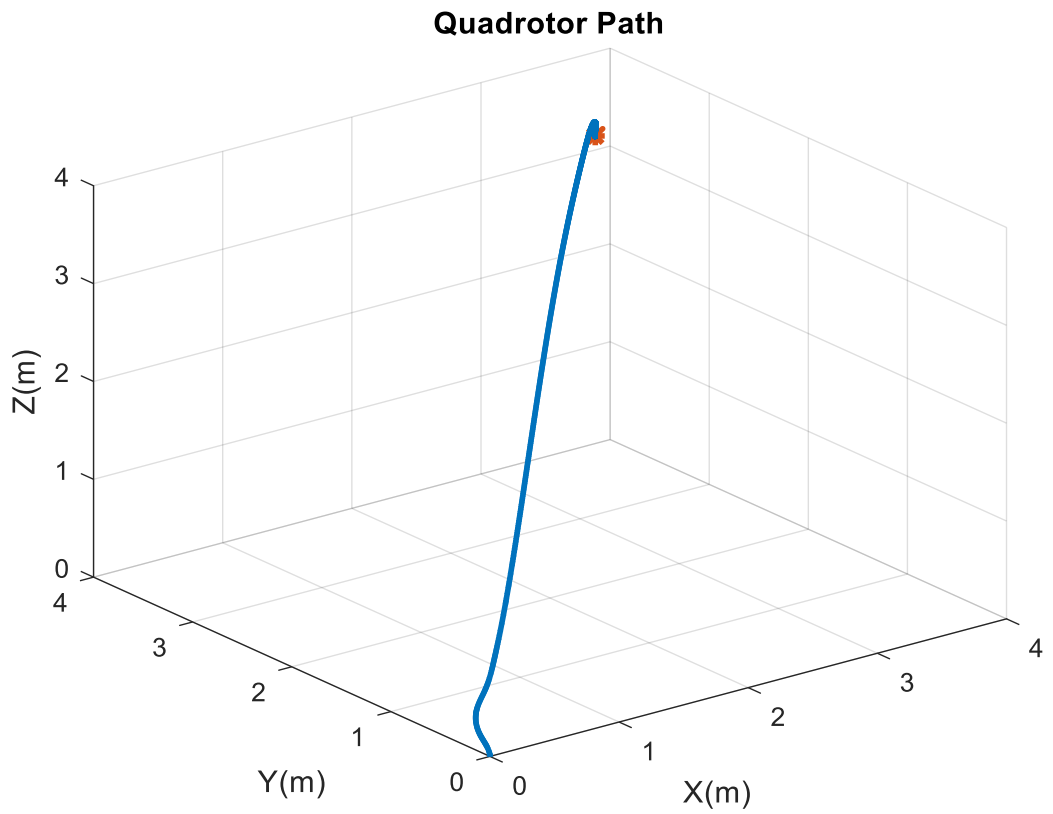

Figure 2.8: Quadrotor's trajectory.

2.4.2. PID Simulation with Disturbance

The disturbance has been added during simulation in the form of Gaussian noise[1]. Each controller has been fed with disturbance separately as shown in the block diagram of figure 2.9 below.

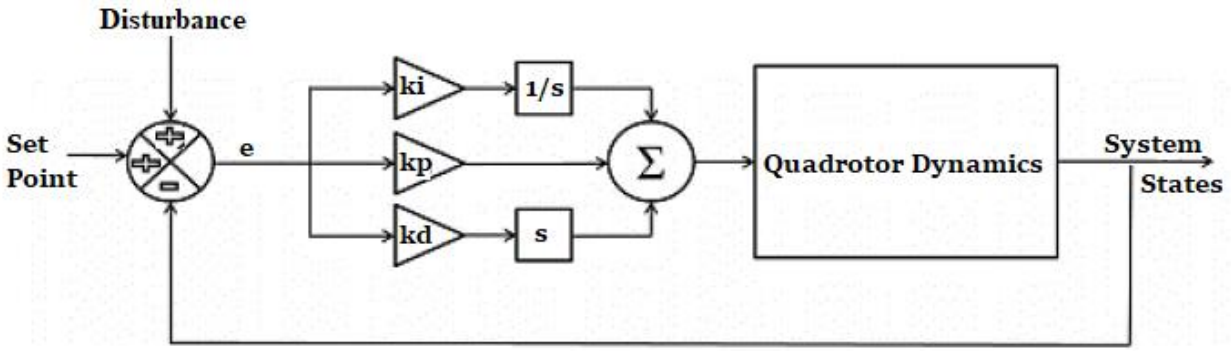

Figure 2.9: Block diagram of a PID Controller fed with disturbance.

[1] See Appendix

Figures 2.10 and 2.11 show the position responses and control inputs, respectively, in the presence of disturbance.

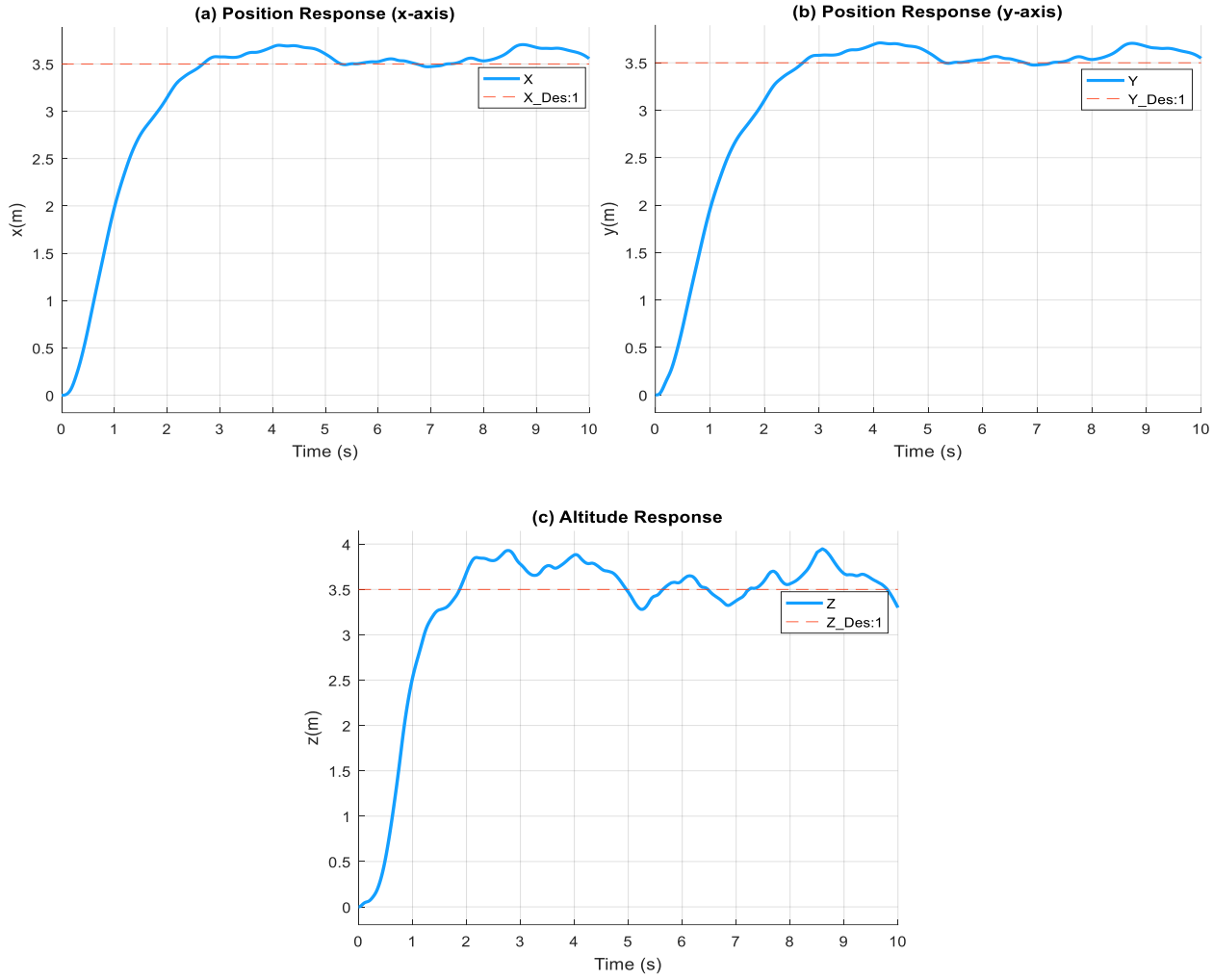

Figure 2.10: Position responses with the presence of disturbance.

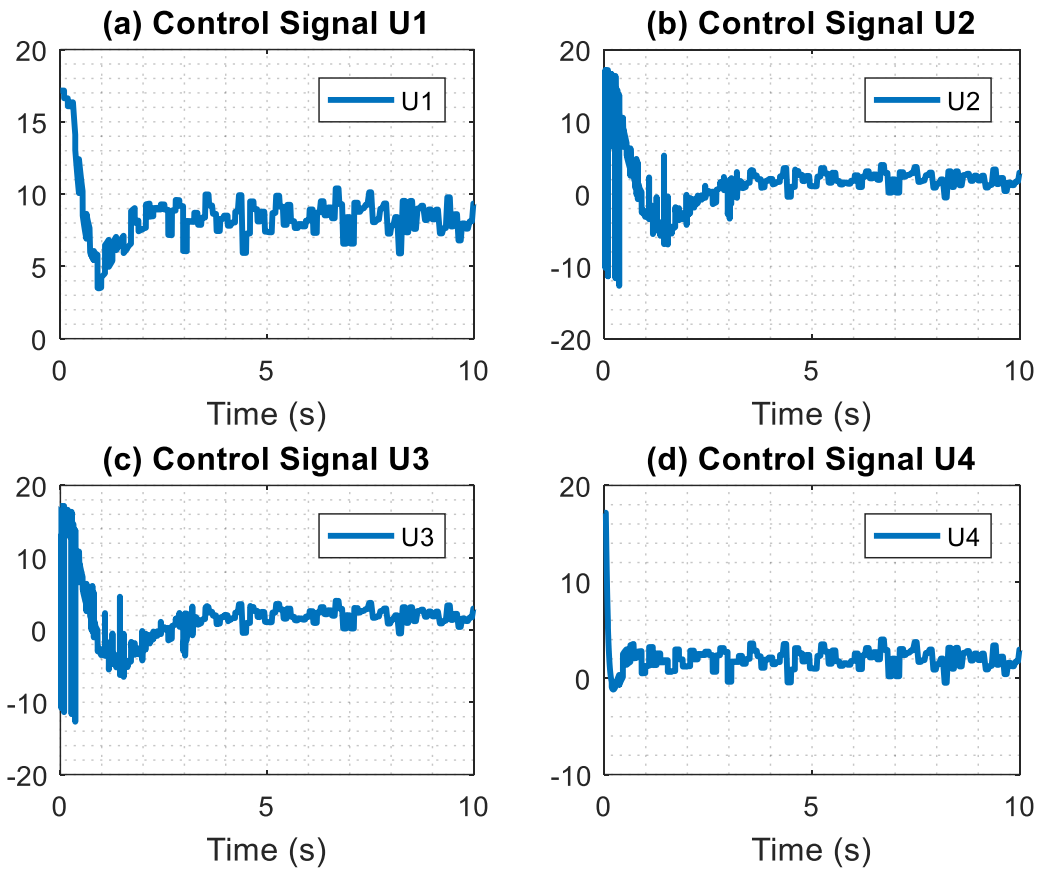

Figure 2.11: Control signals in the presence of disturbance.

Chapitre 3 PD Based Controller

After developing a PID controller to control the quadrotor, a PD based controller was developed. A PD controller is just a PID with $k_i = 0$. The block diagram for a PD controller is shown in Figure 3.1 below.

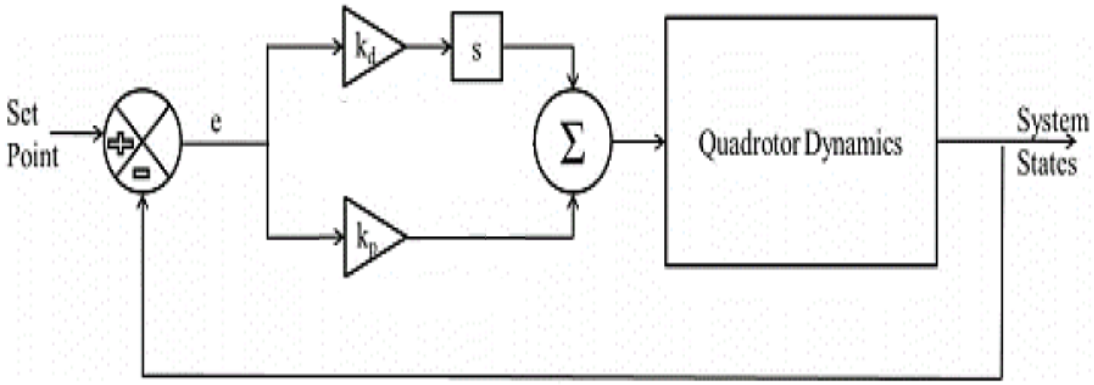

Figure 3.1: Block diagram of a PD controller.

3.1. Introduction to PD Control

The Proportional Derivate Control is one of the simplest linear control laws since it just needs a proportional and a derivative controller. It is very simple and computationally efficient and can be easily implemented using microcontrollers. The mathematical simplicity and ease of understanding makes it one of the most used control laws in Aerial Robotics.

The structure of a PD controller may be written in the following form:

$$u(t) = k_p e(t) + k_d \frac{de(t)}{dt} \tag{3.1}$$

Where $u(t)$ is the input signal, $e(t) = r(t) - y(t)$ is the error signal such that $r(t)$ is the reference signal and $y(t)$ is the output signal.

k_p and k_d represent the proportional and derivative gains of the PD controller, respectively.

3.2. Control laws

The control laws are derived are given as follow:

3.2.1. Altitude Control

The altitude PD controller generates the control signal U_1 that is responsible of the altitude of the quadrotor as per equation (1.26a). The derived control law is as follow:

$$U_1 = k_p(z - z_d) + k_d(\dot{z} - \dot{z_d}) \tag{3.2}$$

Where:

z_d is the desired altitude.

$\dot{z_d}$ is the desired altitude rate of change.

3.2.2. Attitude and Heading Control

Roll Controller

The derived control law generates the input U_2 that controls the roll angle ϕ as follows:

$$U_2 = k_p(\phi - \phi_d) + k_d(\dot{\phi} - \dot{\phi_d}) \tag{3.3}$$

Where:

ϕ_d is the desired roll angle.

$\dot{\phi_d}$ is the desired roll angle rate of change.

Pitch Controller

The derived control law generates the input U_3 that controls the pitch angle θ as follows:

$$U_3 = k_p(\theta - \theta_d) + k_d\left(\dot{\theta} - \dot{\theta_d}\right) \tag{3.4}$$

Where:

θ_d is the desired pitch angle.

$\dot{\theta_d}$ is the desired pitch angle rate of change.

Yaw Controller

The derived control law generates the input U_4 that controls the yaw angle ψ as follows:

$$U_4 = k_p(\psi - \psi_d) + k_d(\dot{\psi} - \dot{\psi_d}) \tag{3.5}$$

Where:

ψ_d is the desired yaw angle.

$\dot{\psi_d}$ is the desired yaw angle rate of change.

Position Control

PD controllers are used to calculate the desired accelerations $\ddot{x}$ and $\ddot{y}$.

$$\begin{aligned} \ddot{x} &= k_p(x - x_d) + k_d(\dot{x} - \dot{x_d}) \\ \ddot{y} &= k_p(y - y_d) + k_d(\dot{y} - \dot{y_d}) \end{aligned} \tag{3.6}$$

3.3. PD Controller Simulation

As for PID controller in Chapter 3, Ziegler-Nichols method for tuning PID controllers was used in order to set the PD gains.

The system has been tested with and without disturbances (wind). The disturbance has been added in the form of additional forces and moments to the right side of the system's translational and rotational equations of motion (1.4) and (1.15).

Table 3.1: Parameters of the PD Controller

	Desired value	k_p	k_d	Settling time (s)	Overshoot (%)
Altitude (z)	3.5	5.2	2.5	1.6	0
Position (x and y)	3.5	1.45	0.33	1.8	0
Attitude (θ and ϕ)	0	6.5	4.5	5	/
Heading (ψ)	5	4.5	0.82	0.49	/

3.3.1. PD Simulation Without Disturbance

Figures 3.2, 3.3 and 3.4 show the position responses, control inputs and the trajectory, respectively, in ideal conditions (without disturbances)

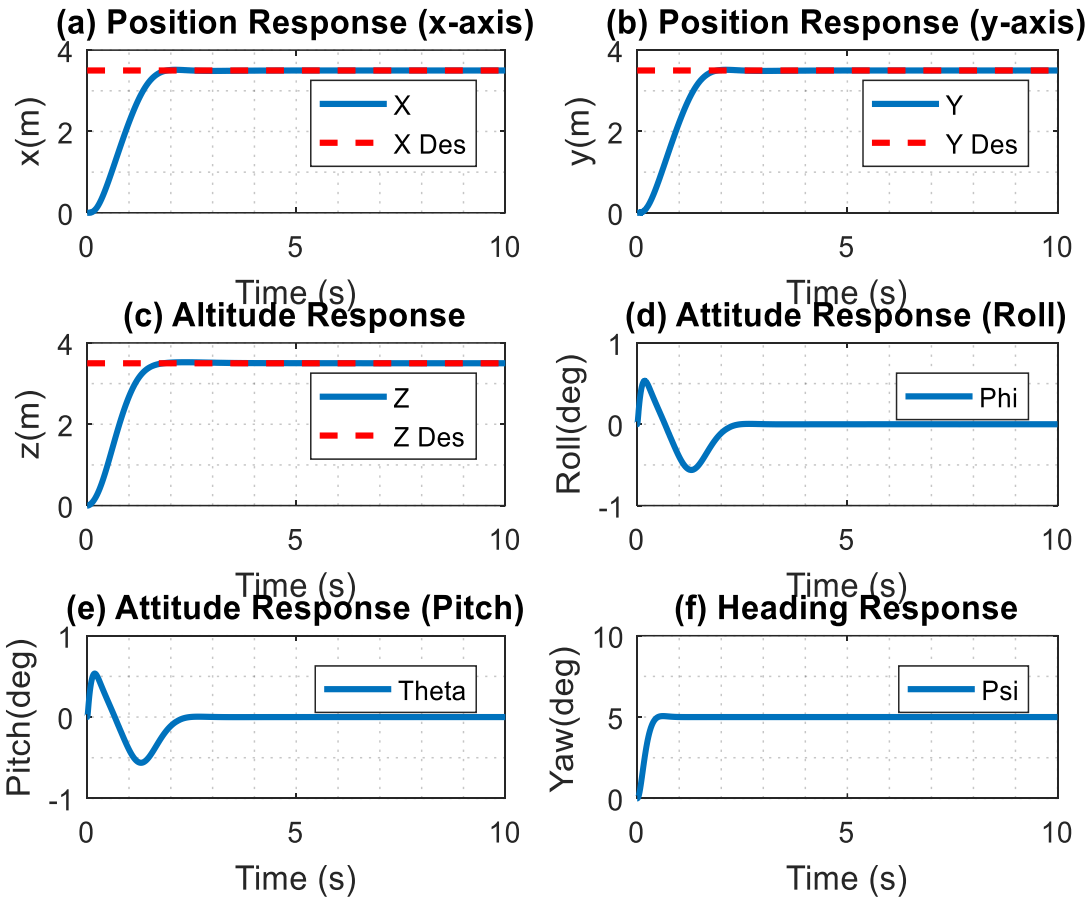

Figure 3.2: Position responses of the PD controller.

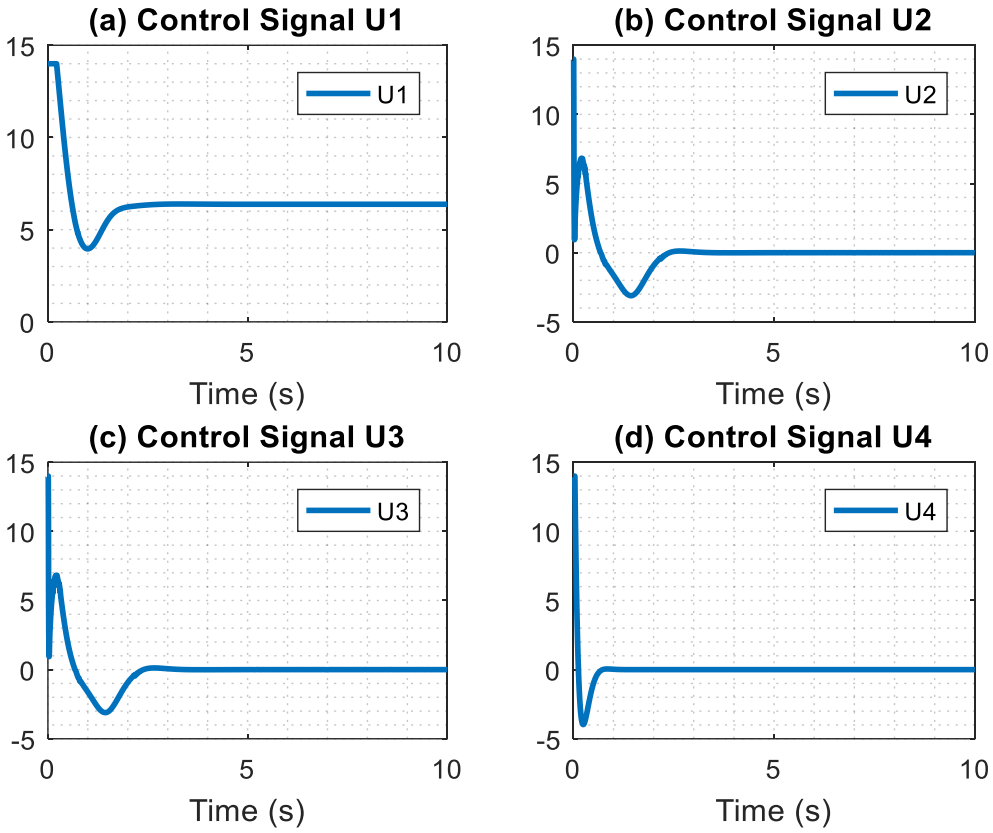

Figure 3.3: Control Signals of the PD controller.

3.3.2. PD Simulation with Disturbance

As done in Chapter 2 for PID Controller, the disturbance has been added during simulation in the form of Gaussian noise[2]. Figures 3.4 and 3.5 show the position responses and control inputs, respectively, in the presence of disturbance.

[2] See Appendix

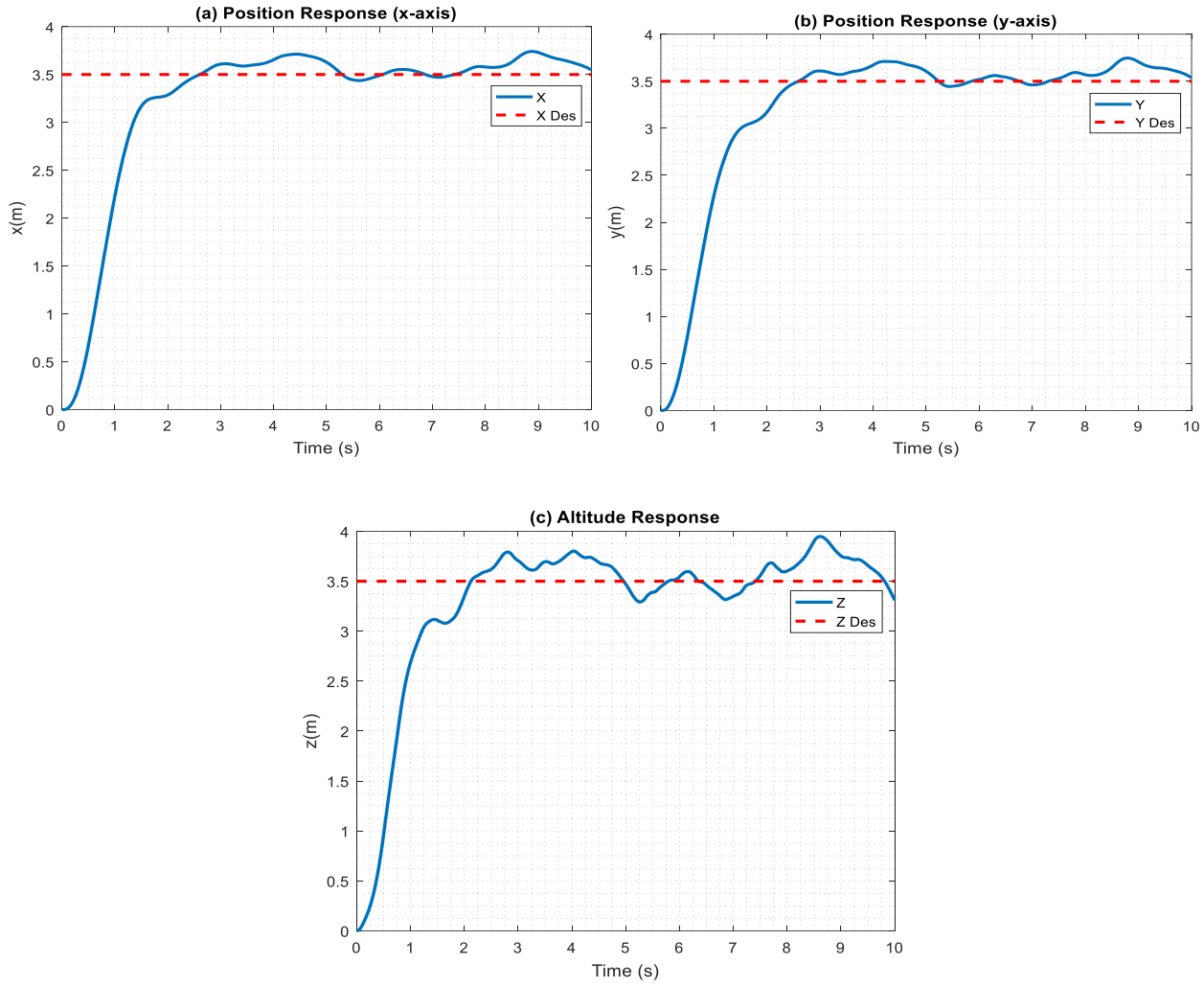

Figure 3.4: Position responses with the presence of disturbance.

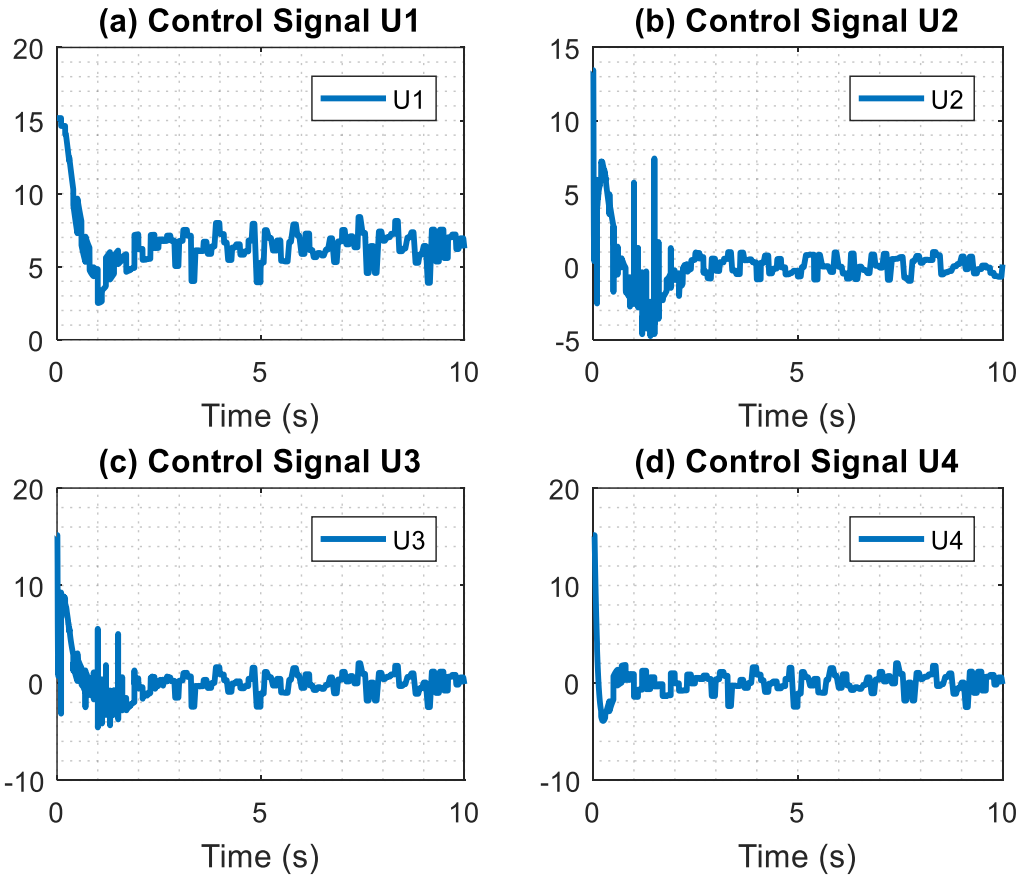

Figure 3.5: Control signals with the presence of disturbance.

Chapitre 4 PID Based Gain Scheduling Controller

In this chapter a Gain scheduling PID controller was developed to generate the desired control inputs for the quadrotor.

4.1. Introduction to Gain Scheduling

In control theory, gain scheduling is an approach to control non-linear systems. It uses a family of linear controllers, each of which provides satisfactory control for a different operating point of the system[12]. In this chapter, a family of PD controllers will be developed, the use of PD controllers came because of there simplicity to be developed. Each PD controller has different controller gains and will be able to stabilize the quadrotor system in certain range of operation. The acquired gains were used in a look up table, figure 4.1 shows the block diagram of a gain scheduling based PD controller.

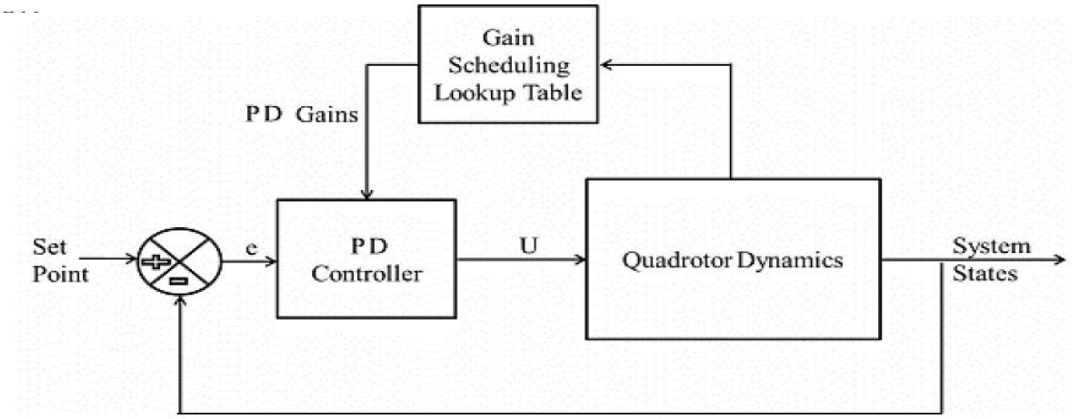

Figure 4.1: Block diagram of a PD based gain scheduling controller.

4.2. Gain Scheduling Controller Simulation

In order to tune the PD controller gains for a desired set of operating points, Genetic Algorithm[3] has been used.

[3] See Appendix

The quadrotor is commanded to follow a certain varying trajectory in order to show the PD based gain scheduling controller versus the classical PD controller.

4.2.1. Heading Controller

Table 4.1 shows the tuned parameters for the PD based Gain Scheduling controller for the different heading positions given in degrees. Figure 4.2 shows the response of the PD based Gain Scheduling controller and the classical PD controller in order to compare their performances.

Table 4.1: Attitude Tuned Parameters

Desired Heading (deg)	k_p	k_d	Settling time (s)
2	6.53	0.711	0.12
4	5.72	0.682	0.45
6	6.81	0.772	0.56
8	6.63	0.784	0.61
10	5.18	0.619	0.68
12	5.74	0.710	0.72
14	5.79	0.718	0.83
16	4.69	0.611	0.95
18	3.70	0.513	1.05
20	5.07	0.634	1.20

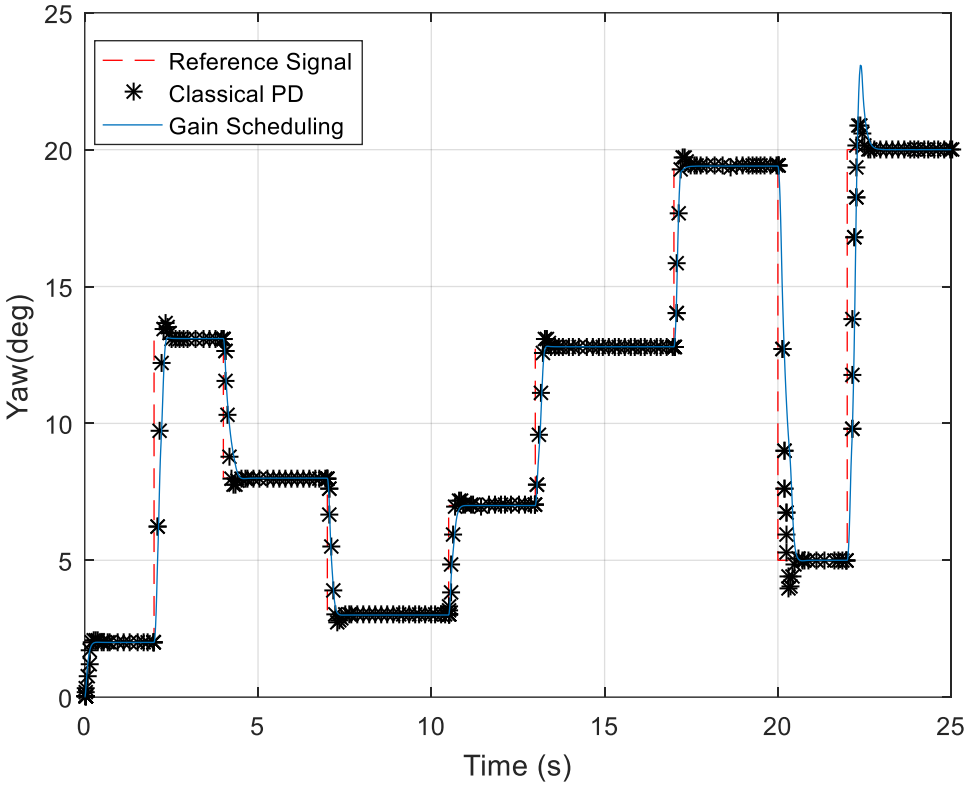

Figure 4.2: Heading response.

4.2.2. Altitude Controller

Table 4.2 shows the tuned parameters for the PD based Gain Scheduling controller for the different heading positions given in degrees. Figure 4.3 shows the response of the PD based Gain Scheduling controller and the classical PD controller in order to compare their performances.

Table 4.2: Altitude Tuned Parameters

Desired Altitude (m)	K_p	K_d	Settling time (s)
2	8.60	3.40	1.02
4	6.20	3.20	1.41
6	5.10	2.80	1.47
8	5.57	3.11	1.53
10	5.35	2.90	1.43
12	6.25	3.10	1.42
14	5.61	2.85	1.49
16	5.96	3.15	1.58
18	5.08	2.77	1.66
20	5.74	3.23	1.76

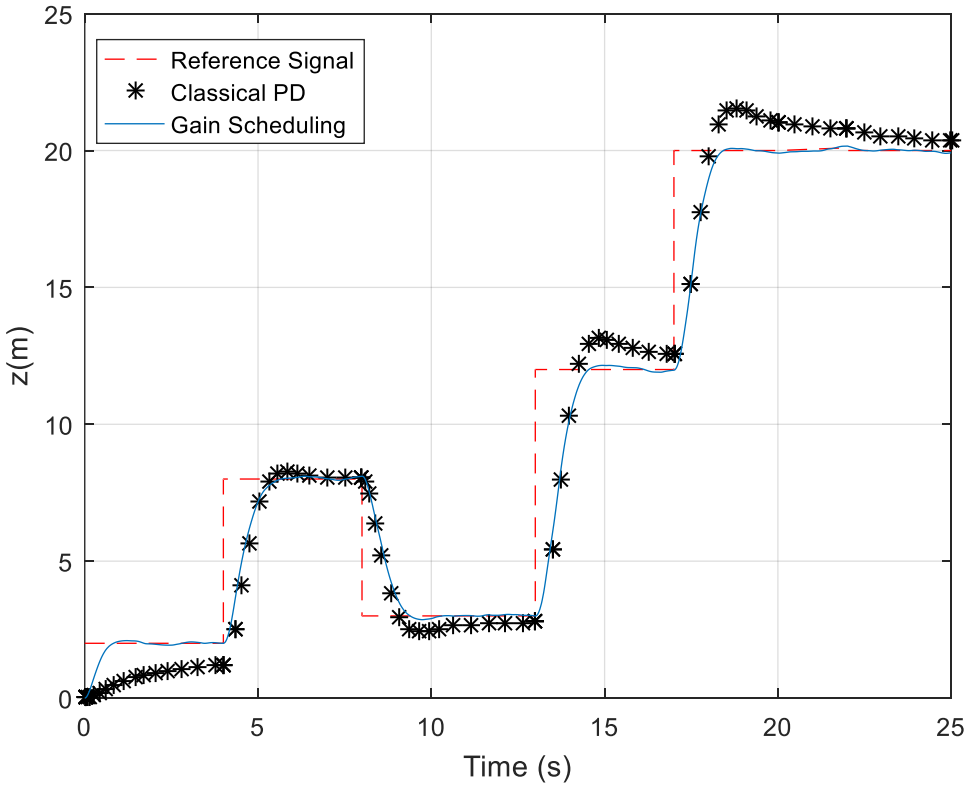

Figure 4.3: Altitude response.

4.2.3. Attitude Controller

Table 4.3 shows the tuned parameters for the PD based Gain Scheduling controller for the different heading positions given in degrees. Figure 4.4 shows the response of the PD based Gain Scheduling controller and the classical PD controller in order to compare their performances.

Table 4.3: Roll Tuned Parameters

Desired Roll (deg)	k_p	k_d	Settling time (s)
1	6.80	0.985	0.31
2	3.50	0.771	0.35
3	5.20	0.803	0.37
4	3.42	0.651	0.44
5	4.12	0.792	0.41
6	3.88	0.799	0.45
7	4.32	0.885	0.48
8	3.12	0.717	0.48
9	4.42	0.862	0.49
10	4.31	0.992	0.47

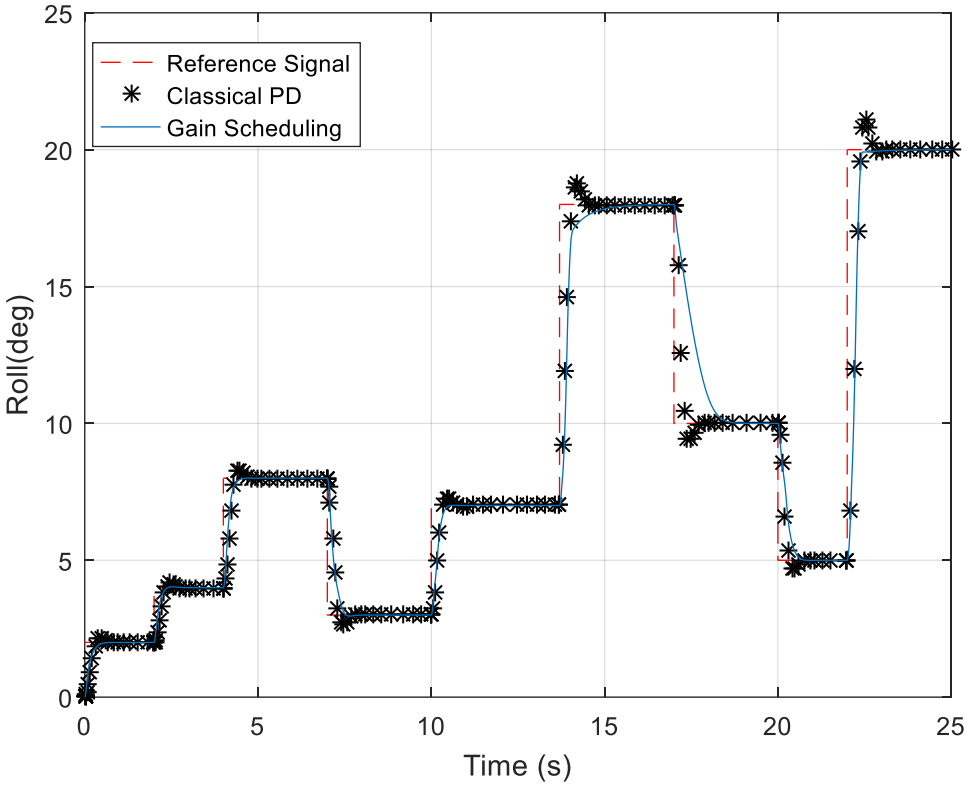

Figure 4.4: Attitude response.

4.2.4. Control Inputs

Figure 4.5 shows the control inputs generated by the PD based Gain Scheduling controller.

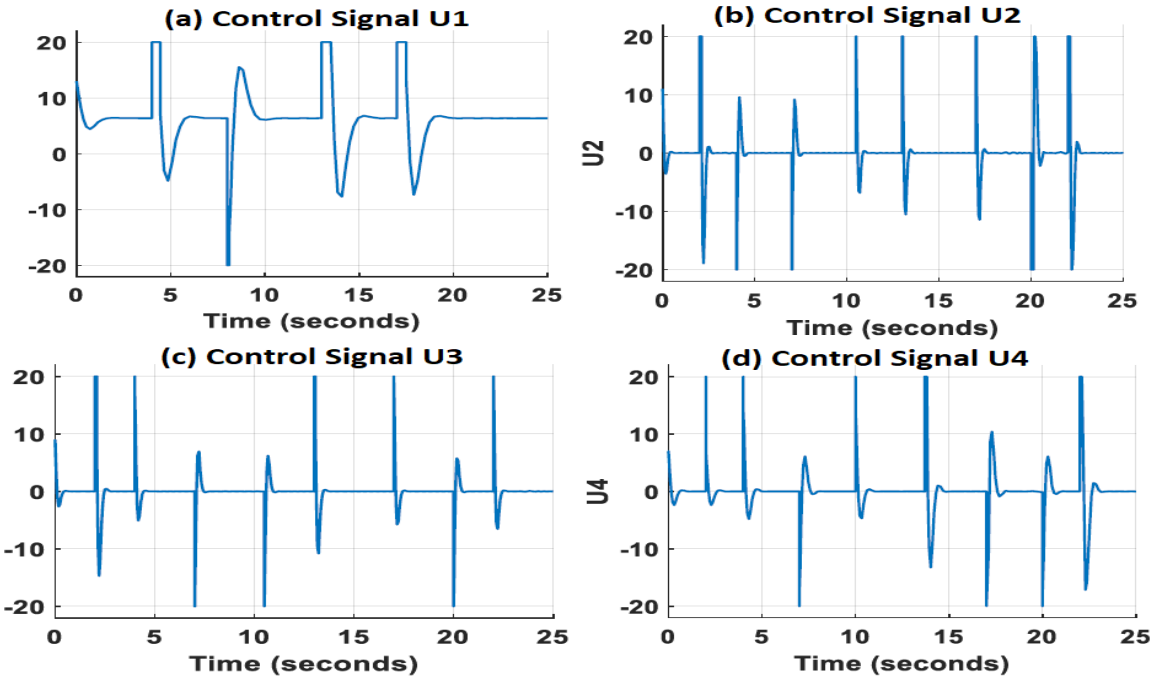

Figure 4.5: Control signals.

Chapitre 5 Results and Conclusion

5.1. Results Discussion

5.1.1. Proportional Integral Derivative (PID) Controller

The PID Control comes out with many advantages as:

-An easy tuning process.

-Few tuning parameters.

-The implementation process is easy and the mechanism is not hard to understand.

However, the control is not robust to noise, disturbances and parameters variation. It comes out also that the derivative (D) term amplifies the noise and it is not causal which makes it not exactly realizable.

5.1.2. Proportional Derivative (PD) Controller

The ease of use and the fewer tuning parameters of the PD Control makes it better than the PID Control since the Integral term is not used. However, the derivative term still makes a problem in implementation and noise amplification and the PD Control still not robust to disturbances and parameters variation.

5.1.3. PD Based Gain Scheduling Controller

The PD based Gain Scheduling Control is more adaptive to different operating conditions and gives better performances in practical applications of the quadrotor. However, its several tuning parameters and the switching time between a set of controller gains to another constitute its major drawbacks. Indeed, the switching time has to be done in infinitesimally small time to guarantee a good performance.

5.2. Comparison Between The Three Developed Controllers

In the linear region (near the hover), the three developed controllers give comparable dynamic performances. PID and PD controllers are practically same. Gain Scheduling controller is adaptive whereas PD and PID controllers are not.

Away from hover (outside the linear region), PD and PID controllers fail to stabilize the system due to their linear nature. Gain Scheduling controller faces the same problem in stabilizing the system since it is based on a PD controller which is linear.

In the presence of disturbance (windy conditions) PD and PID suffer from lack of robustness.

In terms of the generated control signals U_1 through U_4, PD comes out to be the most energy efficient followed by PID. PD based Gain Scheduling controller suffers from spikes in the control signals due to the sudden transition between one set of control gains to another.

5.3. Conclusion

The aim of this work was to derive a complete mathematical model for a quadrotor and develop three (3) linear control techniques: a linear Proportional-Integral-Derivative (PID) Controller, a linear Proportional-Derivative (PD) Controller and a PD based Gain Scheduling Controller to stabilize the states of the quadrotor.

A complete model was then implemented and simulated on MATLAB/Simulink. The tuning of the parameters of the three used controllers is done using Ziegler-Nichols Method.

The PD based Gain scheduling controller gave better performances than the classical PD and PID controllers when commanding the quadrotor to follow a varying trajectory, which is a more realistic application for a quadrotor UAV. The PD and PID controllers gave better performances in a windy environment and the three controllers performed comparably in near the hovering operation of the quadrotor.

5.4. Future Scops

As futur work, one main addition would be to make the control techniques more robust to disturbances and noise. In other words, the *"robustification"* of the control techniques againt wind as the simulation realized in *"windy"* conditions showed a huge degradation of the performances. Indeed, a more realistic application of the quadrotor UAV would be in the present of wind since this meteorological phenomenon is unpredictable but also ineluctable. Moreover, the designed quadrotor model parameters have been taken to be known accurately without any uncertainty which is not the case in reality. Developing adaptive control algorithm to count for uncertainties would strictly enhance the performances and the robustness of the quadrotor. Finally, because of the non-linear nature of the quadrotor, developing non-linear control techniques would be more efficient and effective in controlling it.

Appendix

Quadrotor Parameters

The quadrotor parameters used in simulation have been taken from Bouabdallah's thesis[2]:

Table 1: PID Parameters

Parameter	Description	Value	Unit
I_{xx}	MOI about body frame's x-axis	7.5×10^{-3}	$kg.m^2$
I_{yy}	MOI about body frame's y-axis	7.5×10^{-3}	$kg.m^2$
I_{zz}	MOI about body frame's z-axis	1.3×10^{-2}	$kg.m^2$
l	Moment arm	0.23	m
J_r	Rotor's inertia	6×10^{-5}	$kg.m^2$
m	Quadrotor's mass	0.650	kg
K_F	Aerodynamic force constant	3.13×10^{-5}	$N.m^2$
K_M	Aerodynamic moment constant	7.5×10^{-7}	$N.m.s^2$
R_{mot}	Motor circuit resistance	0.6	Ω
K_{mot}	Motor torque constant	5.2	$N.m.A^{-1}$
K_T	Aerodynamic translation coefficient	diag(0.1,0.1,0.15)	$N.s.m^{-1}$
K_R	Aerodynamic rotation coefficient	diag(0.1,0.1,0.15)	$N.m.s$

Genetic Algorithm

A genetic algorithm (GA) is a high-level procedure inspired by the process of natural selection that is commonly used to generate high-quality solutions to optimization and search problems by relying on biologically inspired operators such as mutation, crossover and selection. A sketch of a typical GA pseudocode is shown below:

Appendix

Algorithm: GA(n, χ, μ)

// **Initialize generation** 0:

$k := 0$;

$P_k :=$ a population of n randomly-generated individuals;

// **Evaluate** P_k:

$Computefitness(i)$ for each $i \in P_k$;

do

{ // **Create generation** $k + 1$:

// **1. Copy:**

Select $(1 - \chi) \times n$ members of P_k and insert into $P_k + 1$;

// **2. Crossover:**

Select $\chi \times n$ members of P_k; Pair them up; Produce offspring; Insert the offspring into $P_k + 1$;

// **3. Mutate:**

Select $\mu \times n$ members of $P_k + 1$; Invert a randomly-selected bit in each;

// **Evaluate** $P_k + 1$:

$Computefitness(i)$ for each $i \in P_k$;

// **Increment:**

$k := k + 1$;

}

while fitness of fittest individual in P_k is not high enough;

return the fittest individual from P_k;

Gaussian Noise

Gaussian noise is a statistical noise having a probability density function (PDF) equal to that of the normal distribution (Gaussian distribution) with $\mu = 0$ and $\sigma = 1$. In other words, the values that the noise can take on are Gaussian-distributed.

The PDF f of a Gaussian random variable x is given by:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Where:

e is the base of the natural logarithm.

μ is the mean of the distribution.

σ is the standard deviation.

Bibiography

[1] Azzam, A, and Xinhua Wang. 2010. “Quad Rotor Arial Robot Dynamic Modeling and Configuration Stabilization.” In *2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics (Car 2010)*, 1:438–44. IEEE.

[2] Bouabdallah, Samir. 2007. “Design and Control of Quadrotors with Application to Autonomous Flying.” Epfl.

[3] Bouabdallah, Samir, and Roland Siegwart. 2005. “Backstepping and Sliding-Mode Techniques Applied to an Indoor Micro Quadrotor.” In *Proceedings of the 2005 Ieee International Conference on Robotics and Automation*, 2247–52. IEEE.

[4] Chen, Chi-Tsong. 1998. *Linear System Theory and Design*. Oxford University Press, Inc.

[5] Gaeid, Khalaf Salloum, Haider AF Mohamed, Hew Wooi Ping, and Lokman H Hassan. (2009) n.d. “NNPID Controller for Induction Motors with Faults.” In *University of Malaya & University of Nottingham Malaysia Campus, The 2nd International Conference on Control, Instrumentation & Mechatronic (CIM2009).*

[6] Hoffmann, Gabe, Dev Gorur Rajnarayan, Steven L Waslander, David Dostal, Jung Soon Jang, and Claire J Tomlin. 2004. “The Stanford Testbed of Autonomous Rotorcraft for Multi Agent Control (Starmac).” In *The 23rd Digital Avionics Systems Conference (Ieee Cat. No. 04CH37576)*, 2:12–E. IEEE.

[7] Hou, Hongning, Jian Zhuang, Hu Xia, Guanwei Wang, and Dehong Yu. 2010. “A Simple Controller of Minisize Quad-Rotor Vehicle.” In *2010 Ieee International Conference on Mechatronics and Automation*, 1701–6. IEEE.

[8] J. Peraire, S. Widnall. 2008. “3D Rigid Body Dynamics: The Inertia Tensor: 16.07 Dynamics.” Massachusetts Institute of Technology: MIT OpenCourseWare. 2008. https://ocw.mit.edu.

[9] Kendoul, Farid. 2012. “Survey of Advances in Guidance, Navigation, and Control of Unmanned Rotorcraft Systems.” *Journal of Field Robotics* 29 (2): 315–78.

[10] Kim, Jinhyun, Min-Sung Kang, and Sangdeok Park. 2009. “Accurate Modeling and Robust Hovering Control for a Quad-Rotor Vtol Aircraft.” In *Selected Papers from the 2nd International Symposium on Uavs, Reno, Nevada, Usa June 8–10, 2009*, 9–26. Springer.

[11] Li, Jun, and Yuntang Li. 2011. “Dynamic Analysis and Pid Control for a Quadrotor.” In *2011 Ieee International Conference on Mechatronics and Automation*, 573–78. IEEE.

[12] McNichols, Kenneth H, and M Sami Fadali. 2003. “Selecting Operating Points for Discrete-Time Gain Scheduling.” *Computers & Electrical Engineering* 29 (2): 289–301.

[13] Mistler, V, A Benallegue, and NK M’sirdi. 2001. “Exact Linearization and Noninteracting Control of a 4 Rotors Helicopter via Dynamic Feedback.” In *Proceedings 10th Ieee International Workshop on Robot and Human Interactive Communication. ROMAN 2001 (Cat. No. 01TH8591)*, 586–93. IEEE.

[14] Nagaty, Amr, Sajad Saeedi, Carl Thibault, Mae Seto, and Howard Li. 2013. “Control and Navigation Framework for Quadrotor Helicopters.” *Journal of Intelligent & Robotic Systems* 70 (1-4): 1–12.

[15] Regula, Gergely. 2013. “Formation Control of Autonomous Aerial Vehicles.”

[16] Satyajit D. Sarker, Lutfun Nahar. 2018. *Computational Phytochemistry*. Elsevier.

[17] Schreier, Matthias. 2012. “Modeling and Adaptive Control of a Quadrotor.” In *2012 Ieee International Conference on Mechatronics and Automation*, 383–90. IEEE.

[18] DARGHAM, Redouane, SAYOUTI, Adil, et MEDROMI, Hicham. Euler and quaternion parameterization in VTOL UAV dynamics with test model efficiency. *coordinates*, 2015, vol. 2, no 3, p. 4.

[19] Tom Bennison, Edward Hall. 2017. “A-Level Mathematics Year One: A Comprehensive and Supportive Companion to the Unified Curriculum (Level Teaching Maths).”, Tarquin Eds.