People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

University M'Hamed BOUGARA – Boumerdes

Institute of Electrical and Electronic Engineering

Department of Power and Control

Final Year Project Report Presented in Partial Fulfilment of

the Requirements for the Degree of

Master

In: Control Engineering

Option: Control

Title:

Control Algorithms for Autonomous Quadrotors

Presented by:

- **Zerroug Khalil**
- **Larit Mohammed Islem**

Supervisor:

- **Pr.Boushaki**

Registration number ……/2021

Abstract:

In this thesis, a detailed mathematical model for a Vertical Take-off and Landing (VTOL) type Unmanned Aerial Vehicle (UAV) known as the quadrotor is presented. The nonlinear dynamic model has been derived using Newton's and Euler's laws. Three control approaches were developed to control the altitude, attitude, heading and position of the quadrotor in space. The first approach is based on a linear Proportional-Integral-Derivative (PID) controller. The second developed controller is Backstepping while the third one is a Gain Scheduling control.

The Genetic Algorithm technique has been used to get an optimal tuning for the fore mentioned controllers (gains and parameters) and, hence, improving the dynamic response. Simulation based experiments were conducted using MATLAB to evaluate and compare between the three developed control techniques in terms of dynamic performance, stability and possible disturbances effect.

Acknowledgment

First of all, we thank Allah almighty for showing us the right path and guiding us through all the difficulties we faced in our lives, allowing us to reach this honored place.

We would like to thank Pr. Boushaki for all her guidance, encouragement and useful critiques during the project work, and the unique opportunity to work under her supervision.

We are sincerely obliged to many worthy persons, without whom we could have never had the opportunity of learning greatly with interest.

Dedication:

I dedicate this modest work to my beloved family for their love and support through the years and to my dear friends and colleagues.

Zerroug. K

I humbly dedicate this piece of work to my loving parents for their endless guidance and support, to my friends for their inspiring pieces of advices.

Larit M.I

Table of content:

List of figures:

List of tables:

Acronyms:

UAV: Unmanned Aerial Vehicle.

VTOL: Vertical Take-Off and Landing.

LQ: Linear Quadratic.

LQR: Linear Quadratic Regulator.

LQG: Linear Quadratic Gaussian.

PID: Proportional Integral Derivative.

MPC: Model Predictive Control.

DC: Direct Current.

KVL: Kirchhoff's Voltage Law.

BLDC: Brush-Less Direct Current.

GA: Genetic Algorithm.

General introduction:

An unmanned aerial vehicle (UAV) commonly known as a drone, is an aircraft without any human pilot. The flight of UAVs may operate under remote control by a human operator, as remotely-piloted aircraft (RPA), or with various degrees of autonomy, such as autopilot assistance, up to fully autonomous aircraft that have no provision for human intervention [1] UAVs were originally developed through the twentieth century for military missions too "dull, dirty or dangerous" for humans [2]. As control technologies improved and costs fall, their use in the twenty-first century is rapidly finding many more applications [3]through the years the interest of the research community in the UAV is growing tremendously and that can be linked to its specific capabilities such as surveillance, search and rescue and military missions [4], one widely used UAV is the quadrotor, the quadrotor is an UAV with four rotors which enable flight in a similar way to that of a helicopter. Movement is attained by varying the speeds of each rotor creating different thrust forces.

A quadrotor is difficult to be stabilized by human control. Hence, a he UAV must autonomously follow predefined paths in 2D or 3D space. 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. Firstly kinematics is framed from a different perspective. This is followed by an effective use of Newton-Euler's method to derive equations with respect to the body frame of reference and inertial frame of reference. A mathematical model is derived, coordinate systems are defined and explained. By using those coordinate systems, relations between parameters defined in the earth coordinate system and in the body coordinate system are defined. Solutions have been obtained for important parameters such as linear acceleration, angular acceleration and torque. The developed mathematical framework is implemented in MATLAB.

Chapter I: Generalities on quadrotor

1.1 The Quadrotor Concept:

A quadrotor has four motors mounted at the ends of cross arm which are labelled as 1 through 4, in order to avoid a yaw movement (the device turning on itself), it is necessary for two motors to rotate in one direction and that the other two rotate in the opposite direction, the motor 1 & 3 rotate in counter clockwise whereas motor 2 & 4 rotates in clockwise direction or vice versa, The center of the body-fixed frame B is the center of mass and the origin, which is attached to the quadrotor. Rotor 1, 2, 3 and 4 produces upward thrust F1, F2, F3 and F4 respectively, and d is the length between the center of mass and center of rotor as can be seen in Figure 1.1 depicted below:

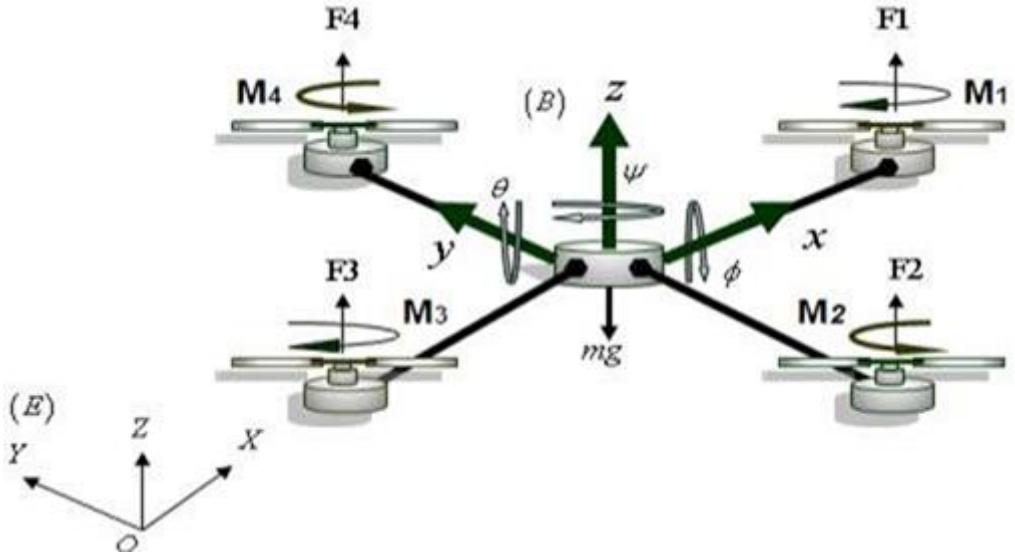

Figure 1.1: Quadrotor notation for four motors.

To operate a quad copter it is enough to vary the power of the engines so it gets up or down, so it tilts from left to right (roll) or moves from backward to forward (pitch) or rotates on itself (yaw), the quad copter has six degrees of freedom, three movements of translation and three rotation movements, those six degrees of freedom shall be controlled by means of only four triggers, so it is an under-actuated system.

1.2 Possible movements of a quadrotor:

As defined in the previous part, we recall that the quadrotor is a machine flywheel with four rotors placed at the ends of a cross. These four rotors provide the three possible movements for a quadrotor: the yaw, pitch and roll.

1.2.1 Rolling motion:

In aeronautics, the rotation around the x-axis is called roll. Applying a thrust difference between the rotor two and the rotor four (generates a torque around the x-axis) varies the roll angle. This movement is coupled with a translation movement along the y-axis.

1.2.2 Pitching motion:

The rotation around the y-axis is called pitching in aeronautics. A variation of the pitch angle is achieved through a difference of thrust between the rotor one and three (generates a torque around the y-axis). This movement is coupled with a translation motion along x-axis.

1.2.3 Lace motion (Yaw):

The rotation around the z-axis is called the aeronautical yaw. When the engines rotate at equal speed anti-rotation torque is zero and the quadrotor does not rotate. To modify the angle of yaw, it is necessary to apply a difference of speed between the couple rotors (one, three) and (two, four). This movement is not a direct result of the thrust but by the reactive couples produced by the rotation of the rotors. The direction of the pushing force does not shift during the motion but the increase in the lift force in a pair of rotors implies the decrease by equal amount of the other pair to ensure that all pushing force stays the same.

1.3 Advantages and Drawbacks of Quadrotor:

The quadrotor can be used in several applications such as policing and the surveillance, the data collection, the aerial photography, the agriculture and the drone racing, they can be used in the commercial, the scientific and the recreational.

An advantage of the quadrotor over helicopter is the presence of four propellers providing four thrust forces shifted a fixed distance from the centre of gravity instead of only one propeller centred in the middle as in the helicopters structure leads to a more stable stationary hovering in quadrotors [1].

More advantages are the vertical take-off and landing capabilities, better maneuver-ability and smaller size due to the absence of a tail [5], these capabilities make quadrotors useful in small area monitoring and buildings exploration [6].

Moreover, quadrotors have higher payload capacities due to the presence of four motors thus providing higher thrust also they are easily affected by the weather.

On the other hand, quadrotors consume a lot of energy due to the presence of four separate propellers [6]Also, they have a large size and heavier than some of their counterparts again to the fact that there are four separate propellers [7]

1.4 Control

Since the quadrotor has six degrees of freedom, three movements of translation and three rotation movements, those six degrees of freedom shall be controlled by means of only four triggers, so it is an underactuated system, those systems are very difficult to control due to the nonlinear decoupling between the actuator and the degree of freedom there are variety of controller that can be used for this purpose, those controllers can be divided into three main categories [8]

1) Linear flight controllers: are the most common flight control algorithm, these controllers can only perform when the quadrotor is flying around the hover and under specific conditions ,PID, Linear Quadratic Regulator (LQR), Linear Quadratic Gaussian (LQG), gain scheduling fall under linear control category [9].

2) Nonlinear controllers: this category can be subdivided into:

- ❒ linearized model using various techniques to linearize the system including feedback linearization, adaptive control, model predictive control (MPC) [10]
- ❒ Fully nonlinear model: in case of backestepping.

3) Model-free: this method is learning based technique including neural network, fuzzy logic and human based learning techniques.

2 Chapter II: System modelling and control

The kinematics and dynamics models of a quadrotor to be developed in this chapter using Newton-Euler formalism and some basic physical laws, before the derivation of the mathematical model the following assumptions should be taken into consideration:

- ❒ The structure of the quadrotor is rigid and symmetrical.
- ❒ The centre of mass of the quadrotor coincides with the body fixed frame origin.
- ❒ The propellers are rigid.
- ❒ The trust and drag are proportional to the square of propeller's speed.

2.1 Kinematic Model:

In order to study kinematic model the coordinate frames should be defined first, as depicted in figure 1.1 as it shows the Inertial frame E,N and D labelled for east, north and downward respectively and the Body frame x, y and z , the inertial reference frame also known as earth frame fixed on a specific point at ground level, while body frame is centred at the centre of mass of the quadrotor where the x-axis pointing to the first propeller, y-axis to the second and z-axis pointing downward (to the ground) .

The position of the quadrotor is described with: $r = [x\ y\ z]^T$ which is the position vector of the centre of gravity of the quadrotor relative to the Earth reference frame.

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

The three single rotations are described separately by:

- ❒ Rotation around xx-axis.

$$R\,(x, \phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & -sin\phi \\ 0 & sin\phi & cos\phi \end{bmatrix} \quad (1.1)$$

- ❒ Rotation around yy-axis.

$$R\,(y, \theta) = \begin{bmatrix} cos\theta & 0 & sin\theta \\ 0 & 1 & 0 \\ -sin\theta & 0 & cos\theta \end{bmatrix} \quad (1.2)$$

- ❒ Rotation around z-axis.

$$R\,(z,\psi)=\begin{bmatrix} cos\psi & -sin\psi & 0 \\ sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1.3}$$

Assuming the order of rotation to be roll, pitch then yaw, the rotation matrix R matrix -or transformation from the Body frame to the Earth frame is the product of the previous three successive rotations:

$$R(\phi,\theta,\psi) = R(x,\phi)R(y,\theta)R(z,\psi)$$

The complete rotation:

$$R=\begin{bmatrix} \cos\theta\, cos\psi & \cos\psi\sin\theta\sin\phi-\sin\psi\cos\phi & cos\psi\sin\theta\cos\phi+\sin\psi\sin\phi \\ \sin\psi\, cos\,\theta & \sin\psi\sin\theta\sin\phi+\cos\psi\, cos\phi & \sin\psi\sin\theta\cos\phi-\sin\phi\sin\psi \\ -sin\theta & \cos\theta\sin\phi & \cos\theta\, cos\,\phi \end{bmatrix} \tag{1.4}$$

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 as follow:

$$\omega = R_r\dot{\eta} \tag{1.5}$$

Where: $\dot{\eta}$ represent the Euler rates that are measured in the inertial frame: $\dot{\eta} = [\dot{\phi}\ \dot{\theta}\ \dot{\psi}]$

ω represent angular body rates: $\omega = [p\ q\ r]$

$$R_r=\begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & cos\phi\cos\theta \end{bmatrix} \tag{1.6}$$

Around the hover position, small angle assumption is made where $\cos\phi \approx 1$, $cos\theta \approx 1$, and $\sin\phi \approx 0$, $\sin\theta \approx 0$. Then, one can write $[\phi\ \theta\ \psi] \approx [p\ q\ r]$. Thus R_r can be simplified to an identity matrix I.

2.2 Dynamic Model:

The motion of the quadrotor can be divided into two subsystems; rotational subsystem (roll ϕ, pitch θ and yaw ψ angles) and translational subsystem (altitude z, and x and y position), the rotational subsystem is fully actuated while the translational subsystem is under actuated.

2.2.1 Rotational motion:

The rotational equations of motion are derived in the body frame using the Newton-Euler method with the following general formalism:

$$J\dot{\omega} + \omega \times J\omega = M_B \tag{1.7}$$

Where:

J : Quadrotor's diagonal inertia Matrix.

ω: Angular body rate.

M_B: Moments acting on the quadrotor in the body frame.

Inertia matrix:

The inertia matrix for the quadrotor is a diagonal matrix, the off-diagonal elements, which are the product of inertia, are zero due to the symmetry of the quadrotor:

$$J = \begin{pmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{pmatrix} \tag{1.8}$$

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

Moments acting on the quadrotor:

For the last term of equation (1.3), there is a need to define two physical effects which are the aerodynamic forces and moments produced by a rotor. As an effect of rotation, there is a generated force called the aerodynamic force or the lift force and there is a generated moment called the aerodynamic moment. Equations (1.5) and (1.6) show the aerodynamic force F_i and moment M_i produced by the i^{th} rotor [11].

$$F_i = \frac{1}{2}\rho A C_T r^2 \Omega_i^2 \tag{1.9}$$

$$M_i = \frac{1}{2}\rho A C_D r^2 \Omega_i^2 \tag{1.10}$$

Where:

ρ: Air density.

A: Blade area.

C_T, C_D:Aerodynemic coefficient.

r : Radius of blade.

Ω_i Angular velocity of rotor i.

Obviously, the areodynamic effect depends on the geometry of the propeller and the air density, since for the case of quadrotor, the maximum altitude is usually limited, thus the air density can be considered as a constant, equations (1.5) and (1.6) can be simplified to [12]

$$F_i = K_f \Omega_i^2 \tag{1.11}$$

$$M_i = K_M \Omega_i^2 \tag{1.12}$$

Where K_f and K_M are the aerodynamic force and moment constants respectively and Ωi is the angular velocity of rotor i. The aerodynamic force and moment constants can be determined experimentally for each propeller type.

By identifying the forces and moments generated by the propellers, we can study the moments M_B acting on the quadrotor. Figure 1.1 shows the forces and moments acting on the quadrotor. Each rotor causes an upwards thrust force F_i and generates a moment Mi with direction opposite 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, F4 generates a positive moment.

Thus the total moment about the x-axis can be expressed as:

$$\begin{aligned} M_x &= -F_2 l + F_4 l \\ &= -\left(K_f \Omega_2^2\right)! + \left(K_f \Omega_4^2\right) l \\ &= l K_f (-\Omega_2^2 + \Omega_4^2) \end{aligned} \tag{1.13}$$

For the moments about the body frame's y-axis, also using the right-hand-rule, the thrust of rotor 1generates a positive moment, while the thrust of rotor 3 generates a negative moment about the y-axis. The total moment can be expressed as:

$$\begin{aligned} M_x &= F_1 l - F_3 l \\ &= \left(K_f \Omega_1^2\right) l - \left(K_f \Omega_3^2\right) l \\ &= l K_f (\Omega_1^2 - \Omega_3^2) \end{aligned} \tag{1.14}$$

For the moments about the body frame's z-axis, the thrust of the rotors does not cause a moment. On the other hand, moment caused by the rotors' rotation as per Equation (1.6). By using the right-hand-rule, the moment about the body frame's z-axis can be expressed as:

$$\begin{aligned} M_z &= M_1 - M_2 + M_3 - M_4 \\ &= (K_M\Omega_1^2) - (K_M\Omega_2^2) + (K_M\Omega_3^2) - (K_M\Omega_4^2) \\ &= K_M(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \end{aligned} \tag{1.15}$$

Combining equations (1.9), (.10) and (1.11) in vector form, we get:

$$M_B = \begin{pmatrix} lK_f(-\Omega_2^2 + \Omega_4^2) \\ lK_f(\Omega_1^2 - \Omega_2^2) \\ K_M(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \end{pmatrix} \tag{1.16}$$

Where l is the arm length, which is the distance between the axis of rotation of each rotor to the origin of the body reference frame which should coincide with the centre of the quadrotor [12].

2.2.2 Translational motion:

The translation equations of motion for the quadrotor are based on Newton's second law and they are derived in the Earth inertial frame:

$$m\ddot{r} = P + R\,(\phi, \theta, \psi)\,F_B \tag{1.17}$$

Where:

- $\boldsymbol{r}$: Quadrotor distance from inertial frame: r =$[x \;\; y \;\; z]^T$.
- **P**: The gravitational force acting on the body frame: P= [0 0 -mg] ^{T}.

Where g is the gravitational acceleration: g=9.81 m/s$^{2.}$

- R: Rotational equation of motion: Equation (1.4)
- F_B: Non-gravitational force acting on the body frame.

When the quadrotor is in a horizontal orientation (i.e. it is not rolling or pitching), the only non-gravitational forces acting on it is the thrust F_i produced by the rotation of the propellers which is proportional to the square of the angular velocity Ω_i^2 of the propeller as $F_i = b\Omega_i^2$. Thus, the non-gravitational forces acting on the quadrotor, F_B, can be expressed as:

$$F_B = \begin{bmatrix} 0 \\ 0 \\ \sum_{i=0}^{4} F_i \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ b\,(\,\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2\,) \end{bmatrix} \tag{1.18}$$

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, F_B is multiplied by the rotation matrix $R(\phi, \theta, \psi)$ 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.

2.2.3 General Dynamics:

2.2.3.1 Aerodynamic effects:

In order to have an accurate and realistic model, the aerodynamic effects must not be neglected and should be included. There are namely two types of aerodynamic effects, drag forces and drag moments [13].

2.2.3.2 Drag forces:

Due to the friction of the moving quadrotor body with air, a force acts on the body of the quadrotor resisting the motion. As the velocity of travel of the quadrotor increases, the drag forces in turn increase. The drag forces F_a can be approximated by:

$$F_a = K_t \dot{r} \tag{1.19}$$

Where: K_t is a constant matrix called the aerodynamic translation coefficient matrix and $\dot{r}$ is the time derivative of the position vector r. This indicates that there is an extra force acting on the quadrotor body, the translational equation of motion Equation (1.13) should be rewritten to be:

$$m\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + RF_B - F_a \tag{1.20}$$

2.2.3.3 Drag moments:

The same as the drag force, due to the air friction, there is a drag moment M_a acting on the quadrotor body which can be approximated by [11]:

$$M_a = K_r \dot{\eta} \tag{1.21}$$

Where: K_r is a constant matrix called the aerodynamic rotation coefficient matrix and is the Euler rates accordingly, as 5 the rotational equation of motion expressed by equation 1.3 can be rewritten to:

$$J\dot{\omega} + \omega \times J\,\omega = M_B - M_a \tag{1.22}$$

2.2.4 Rotor Dynamics:

Brushless DC motors are the mostly used in quadrotor providing high torque and less friction. In this derivation the motors are assumed to be non-geared with rigid mechanical coupling between the motors and propellers. [7]. In steady state, the dynamics of a brushless DC motor is the same as the conventional DC motor and it is shown in Figure 2.1:

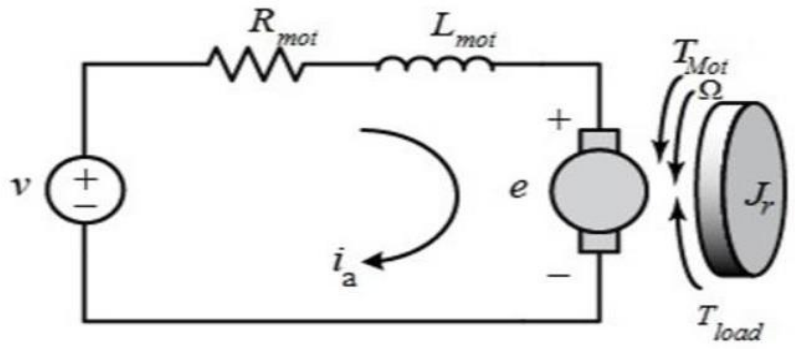

Figure 2.1: DC Motor schematic diagram.

Using Kirchhoff's Voltage Law (KVL):

$$v=R_{mot}\, i_a + L_{mot}\, \frac{di_a}{dt} + K_{mot}\, \Omega_i \tag{1.23}$$

R_{mot} And L_{mot} are the i^{th} motor's resistance and inductance respectively. i_a Is the armature current and v is the input voltage while$K_{mot}\Omega$ is the generated emf e, with K_{mot} the motor torque constant.

Since the quadrotor relies on small motors, their inductance is very small and can be neglected leading to:

$$v=R_{mot}\, i_a + K_{mot}\, \Omega_i \tag{1.24}$$

And

$$i_a = \frac{v-K_{mot}\Omega_i}{R_{mot}} \tag{1.25}$$

The mechanical derivation is then:

$$J_r\dot{\Omega}_l = T_{\text{mot}} - T_{\text{load}} \tag{1.26}$$

$T_{\text{mot}} = K_e i_a$ Is the torque produced by the motor where K_e is the motor's electric constant and for small motors it is approximated to $K_{\text{mot}} \cdot T_{\text{load}} = K_M\Omega^2$ is the load torque generated by from the propeller system as per equation (1.10).

Substituting T_{mot} and T_{load} with equation (2.25) yields to:

$$J_r\dot{\Omega}_l = K_{\text{mot}}\frac{v-K_{mot}\Omega_i}{R_{mot}} - K_M\Omega_l^2 \tag{1.27}$$

Which can be simplified to:

$$v = \frac{R_{mot}}{K_{mot}} J_r \dot{\Omega}_l + K_{mot}\Omega_i + K_M R_{mot}\Omega_i^2 \quad (1.28)$$

OS4 is equipped with four fixed-pitch rotors (no swash plate), each one includes a Brushless Direct Current (BLDC) motor, a one-stage gear box and a propeller.

The entire rotor dynamics was identified and validated using the Matlab Identification Toolbox. A first-order transfer function (2.29) is sufficient to reproduce the dynamics between the propeller's speed set-point and its true speed.

$$G(s) = \frac{0.936}{0.178s+1} \quad (1.29)$$

It is worthwhile to note the non-unity gain in (2.29), this is visible in Figure 2.2, which superimposes the model output (red) and the sensor data (blue) to a step in put (green). In fact, sensor-less Brush Less DC motors require a minimum speed to run thus, the set-point does not start from zero. The motor used does not incorporate hall effect sensors; the identification was carried out using a reflective encoder placed under the propeller gear.

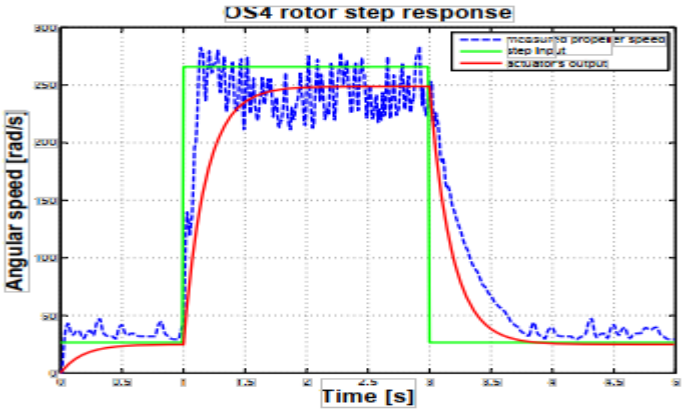

Figure 2.2: Rotor and model step response, measured at propeller's shaft. [7]

2.2.5 State space model:

2.2.5.1 State vector:

The state space vector define the position of the quadrotor in space and its linear and angular velocity:

The state space vector is defined as:

$$X = [x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6 \quad x_7 \quad x_8 \quad x_9 \quad x_{10} \quad x_{11} \quad x_{12}]^T \quad (1.30)$$

Which mapped to the degree of freedom of the quadrotor as:

$$X = [\varphi \quad \dot{\varphi} \quad \theta \quad \dot{\theta} \quad \psi \quad \dot{\psi} \quad z \quad \dot{z} \quad x \quad \dot{x} \quad y \quad \dot{y}]^\tau \quad (1.31)$$

2.2.5.2 Control input vector:

Vector U is defined as the control Input vector which consists of four control equations used to keep the quadrotor in the reference values despite of the external disturbances, the signal U_1 is to guarantee that the altitude z follows its corresponding reference value, while U_2, U_3 and U_4 are used to control the roll, pitch and yaw angles of the system respectively [14].

$$U = [U_1\ U_2\ U_3\ U_4]^T \tag{1.32}$$

Where:

$$\begin{cases} U_1 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ U_2 = b(-\Omega_1^2 + \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \\ U_3 = b(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \\ U_4 = d(\Omega_1^2 + \Omega_2^2 - \Omega_3^2 - \Omega_4^2) \end{cases} \tag{1.33}$$

Noting that omegas are the rotor's i velocity.

Equation 2.31 can be rearranged in a matrix form:

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ -b & b & b & -b \\ b & -b & b & -b \\ d & d & -d & -d \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \tag{1.34}$$

2.2.5.3 Rotational equation of motion:

After the derivation of equations in 2.31 and by substituting them into equation 2.16 we get the equation of the total moments acting on the quadrotor as follow:

$$M_B = l \begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} \tag{1.35}$$

Substitute of the later equation into the rotational equation of motion 2.7 and by expanding we get:

$$\begin{bmatrix} I_{xx}\ddot{\phi} \\ I_{yy}\ddot{\theta} \\ I_{zz}\ddot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\theta} I_{zz}\dot{\psi} - \dot{\psi} I_{zz}\dot{\theta} \\ \dot{\psi} I_{xx}\dot{\phi} - \dot{\phi} I_{zz}\dot{\psi} \\ \dot{\phi} I_{yy}\dot{\theta} - \dot{\theta} I_{zz}\dot{\phi} \end{bmatrix} + \begin{bmatrix} \dot{\theta} J_r \Omega_r \\ -\dot{\phi} J_r \Omega_r \\ 0 \end{bmatrix} = l \begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} \tag{1.36}$$

Rewriting this equation we derive the angular acceleration as:

$$\begin{aligned}\ddot{\phi} &= \frac{l}{I_{xx}}U_2 - \frac{J_r}{I_{xx}}\dot{\theta}\Omega_r + \frac{I_{yy}}{I_{xx}}\dot{\psi}\dot{\theta} - \frac{I_{zz}}{I_{xx}}\dot{\theta}\dot{\psi} \\ \ddot{\theta} &= \frac{l}{I_{yy}}U_3 - \frac{J_r}{I_{yy}}\dot{\phi}\Omega_r + \frac{I_{zz}}{I_{yy}}\dot{\phi}\dot{\psi} - \frac{I_{xx}}{I_{yy}}\dot{\psi}\dot{\phi} \\ \ddot{\psi} &= \frac{l}{I_{zz}}U_4 + \frac{I_{xx}}{I_{zz}}\dot{\theta}\dot{\phi} - \frac{I_{yy}}{I_{zz}}\dot{\phi}\dot{\theta}\end{aligned} \tag{1.37}$$

In sake of simplicity we notate the following:

$b_1 = \frac{l}{I_{xx}}$, $b_2 = \frac{l}{I_{yy}}$, $b_3 = \frac{l}{I_{zz}}$

$$\begin{aligned}a_1 &= \frac{I_{yy} - I_{zz}}{I_{xx}} \\ a_2 &= \frac{J_r}{I_{xx}} \\ a_3 &= \frac{I_{zz} - I_{xx}}{I_{yy}} \\ a_4 &= \frac{J_r}{I_{yy}} \\ a_5 &= \frac{I_{xx} - I_{yy}}{I_{zz}}\end{aligned}$$

Substituting the above notation into equation 2.35 we get as a final result:

$$\begin{aligned}\ddot{\phi} &= b_1U_2 - a_2x_4\Omega_r + a_1x_4x_6 \\ \ddot{\theta} &= b_2U_3 + a_4x_2\Omega_r + a_3x_2x_6 \\ \ddot{\psi} &= b_3U_4 + a_5x_2x_4\end{aligned} \tag{1.38}$$

2.2.5.4 Translational equation of motion:

After the derivation of equations in 2.31 and by substituting them into equation 2.18 we get the equation of the total moments acting on the quadrotor as follow:

$$F_B = [0\ 0\ U_1]^T \tag{1.39}$$

Substitute of the later equation into the Translational equation of motion 2.17 and by expanding we get:

$$m\begin{bmatrix}\ddot{x} \\ \ddot{y} \\ \ddot{z}\end{bmatrix} = \begin{bmatrix}0 \\ 0 \\ -mg\end{bmatrix} + \begin{bmatrix}(\sin(\phi)\sin(\psi) + \cos(\phi)\cos(\psi)\sin(\theta))U_1 \\ (\cos(\phi)\sin(\psi)\sin(\theta) - \cos(\psi)\sin(\phi))U_1 \\ \cos(\phi)\cos(\theta)U_1\end{bmatrix} \tag{1.40}$$

Rewriting in term of acceleration we get as a final result:

$$\ddot{x} = \frac{U_1}{m}\left(\sin(x_1)\sin(x_5) + \cos(x_1)\cos(x_5)\sin(x_3)\right)$$
$$\ddot{y} = \frac{U_1}{m}\left(\cos(x_1)\sin(x_5)\sin(x_3) - \cos(x_5)\sin(x_1)\right) \quad (1.41)$$
$$\ddot{z} = -g + \frac{U_1}{m}\left(\cos(x_1)\cos(x_3)\right)$$

It is worthwhile to note in the latter system that the angles and their time derivatives do not depend on translation components. On the other hand, the translations depend on the angles. One can ideally imagine the overall system described by (1.38) and (1.41) as constituted of two subsystems, the angular rotations and the linear translations [7]

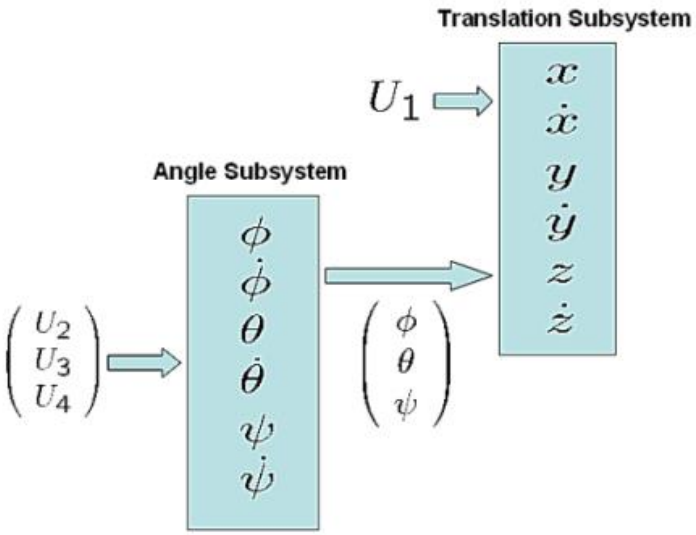

Figure 2.3: relation between rotational and translational subsystem.

2.2.6 State space representation:

Based on the equations of the rotational angular acceleration. Equations (1.38), and the equations of translational (1.41), the complete mathematical model of the quadrotor can be written in a state space representation as follows:

$$
\begin{aligned}
\dot{x_1} &= \dot{\phi} = x_2 \\
\dot{x_2} &= \ddot{\varphi} = x_4 x_6 a_1 - x_4 \Omega_r a_2 + b_1 U_2 \\
\dot{x_3} &= \dot{\theta} = x_4 \\
\dot{x_4} &= \ddot{\theta} = x_2 x_6 a_3 + x_2 \Omega_r a_4 + b_2 U_3 \\
\dot{x_5} &= \dot{\psi} = x_6 \\
\dot{x_6} &= \ddot{\psi} = x_2 x_4 a_5 + b_3 U_4 \\
\dot{x_7} &= \dot{z} = x_8 \\
\dot{x_8} &= \ddot{z} = -g + \frac{U_1}{m}(\cos x_1 \cos x_2) \\
\dot{x_9} &= \dot{x} = x_{10} \\
\dot{x_{10}} &= \ddot{x} = \frac{U_1}{m}(\sin x_1 \sin x_5 + \cos x_1 \sin x_3 \cos x_5) \\
\dot{x_{11}} &= \dot{y} = x_{12} \\
\dot{x_{12}} &= \ddot{y} = \frac{U_1}{m}(-\sin x_1 \cos x_5 + \cos x_1 \sin x_3 \sin x_5)
\end{aligned}
\tag{1.42}
$$

2.3 Control strategy of Quadrotors:

2.3.1 Open loop simulation:

The open loop simulation is used to formulate the quadrotor model to verify the mathematical model, the quadrotor's parameter were taken from [7] thesis, the open loop simulation was carried out using Matlab\Simulink as follow:

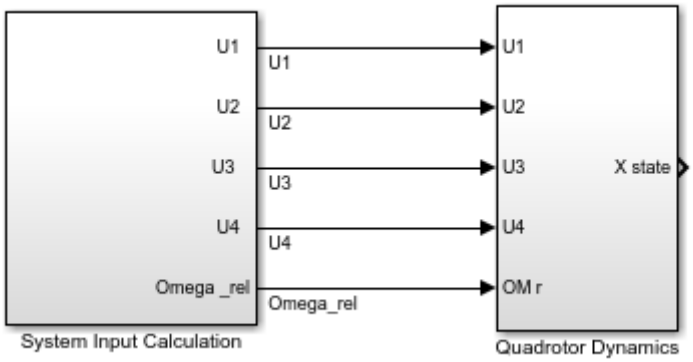

Figure 2.4: open loop simulation block diagram.

The hovering point of the quadrotor was calculated from this equation.

$mg = 4Fi$

$mg = 4(K_f \, \Omega_{it}^2)$

It is assumed that the only force acting on the quadrotor are Ωi and the gravitational acceleration. When Ωi have the same values all the state variables (x, y, ϕ, θ, ψ) are zero the only change was in the altitude z. It was observed there cases when Ωi less then the

hovering angular velocity the altitude go downwards, at the hovering point $z = 0$ by increasing Ωi over the hovering angular velocity the altitude go upwards. Varying the angular velocity of the 4 motors produced the roll, pith or yaw motion.

- Roll: when Ω1 and Ω3 have the same values but Ω2 and Ω4 are different from each other.
- Pitch: when Ω2 and Ω4 have the same values but Ω1 and Ω3 are different from each other.
- Yaw: when Ω1 and Ω3 have the same values but Ω2 and Ω4 have another same values. Rotor dynamics are included in the 'control input calculation' block as the first order lag transfer function shown in equation (2.29).

2.3.2 Closed loop (Control) simulation:

In the field of quadrotors control variety of control strategies were considered, in the work of [7] the PID, LQ, Lyapunov, backstepping and integral backstepping were discussed with a proper simulations.

The design process of the quadrotor is developed using Euler's angles for modelling and control discussed earlier in this chapter and it consisted of the altitude, attitude, heading and position controllers:

2.3.2.1 Altitude Controller:

The altitude control takes an error signal e as an input which is the difference between the desired altitude Z_d and the altitude Z and produced a control signal U_1, as shown in the block diagram in Figure 2.5 below.

2.3.2.2 Attitude & heading Controller:

The attitude and heading controller take as an input an error signal e which is the difference between the desired roll, pitch and yaw and their actual values, the attitude and heading controller produces the output signals U_2, U_3 and U_4, as shown in Figure2.5.

2.3.2.3 Position Controller:

The x and y position can be controlled through the roll and pitch angles, because the system is not decoupled and cannot be directly controlled using one of the four control laws U_1 through U_4.

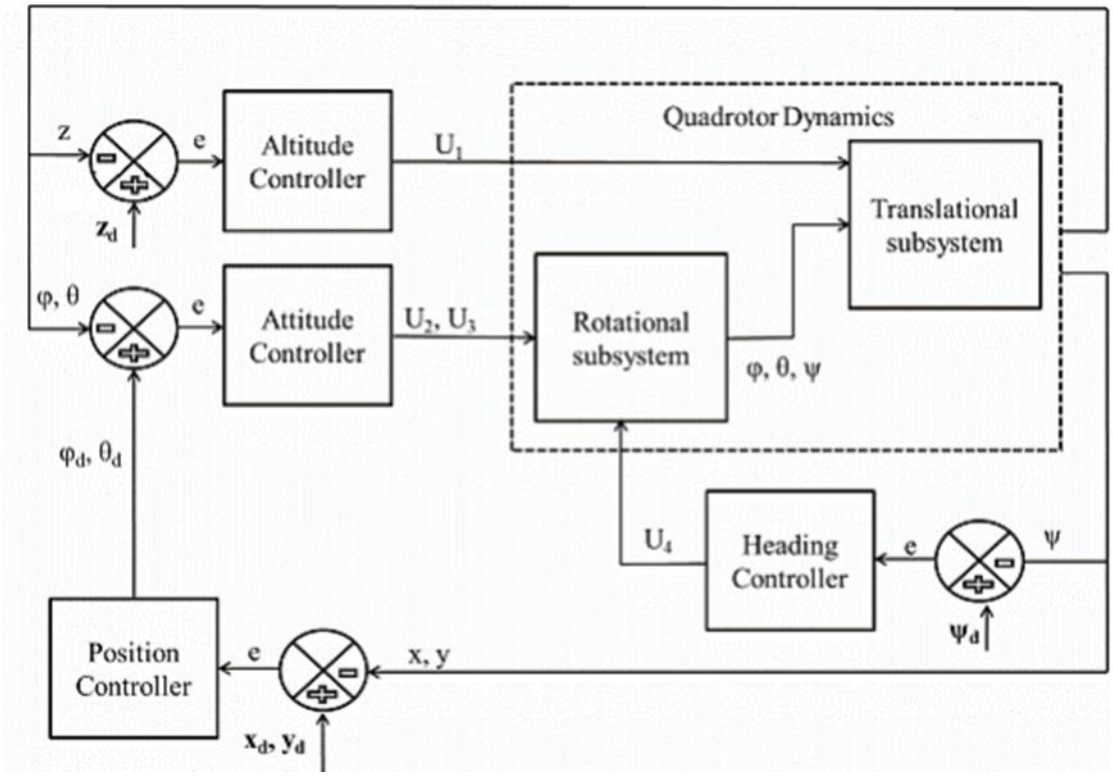

Figure 2.5: Process control bock diagram.

Control of a quadrotor as shown in Figure 2.5, one can see in the figure that the inner loop is an attitude control loop while the outer loop is the position control loop. It is logical to infer that the dynamics of the inner loop must be faster than the dynamics of the outer loop. In hover configurations the dynamics of attitude do not matter much in general, but in cases where the robot has to make manoeuvres, it is imperative to have a faster attitude control loop.

2.3.2.4 3D Trajectory generator:

The main idea in trajectory generation is to use calculus of variation and the Euler Lagrange equations to find smooth trajectory with the provided boundary conditions since the quadrotor is a dynamic entity with mass and moment of inertia, not using waypoints and higher polynomial trajectory may results discontinuities in the velocity and infinite acceleration which the actuator cannot handle, so the idea is to design a trajectory that minimize the integral of the square of the snap (2^{nd} derivative of acceleration $\ddot{a}$) which allow us to satisfy different constraints on the states and the input. [15]

We shall not discuss more about the problem of trajectory generation because a helical trajectory is used in this case and a circular trajectory is infinitely differentiable. So, the

desired acceleration (a), jerk ($\dot{a}$) and snap ($\ddot{a}$) (and other higher derivatives) are all finite and bounded.

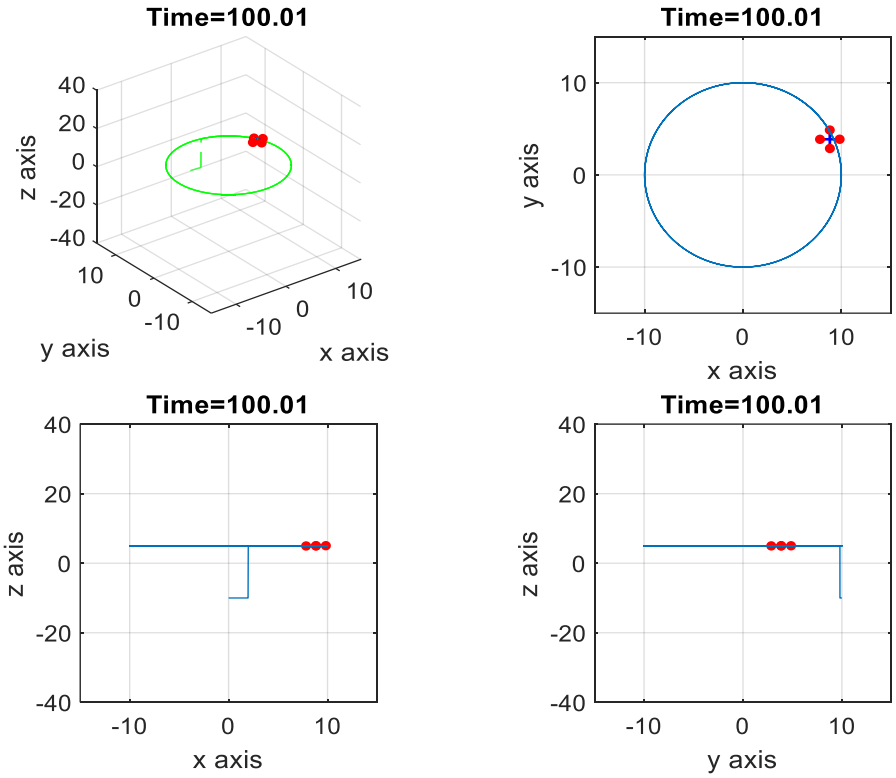

Figure 2.6: 3D, x, y and z Projection of the circular trajectory.

3 Chapter III: PID Controller:

3.1 Introduction to PID:

A proportional–integral–derivative controller (PID controller) is a control loop mechanism employing feedback that is widely used in industrial control systems and a variety of other applications requiring continuously modulated control. A PID controller continuously calculates an error value as the difference between a desired setpoint and a measured process variable and applies a correction based on proportional, integral, and derivative terms (denoted P, I, and D respectively), hence the name. The Proportional Integral Derivate Control is one of the simplest linear control laws. It is very simple and computationally efficient. The distinguishing feature of the PID controller is the ability to use the three control terms of proportional, integral and derivative influence on the controller output to apply accurate and optimal control. Each of these three parameters has a different impact on the controller output:

- The proportional component depends only on the computed error term between the System's output and the setpoint. In other words، "P" depends on the present error.
- The integral component sums the error term over time. The result is that even a small error term will cause the integral component to increase slowly. The integral response will continually increase over time unless the error is zero. In other words, "I" depends on the accumulation of past errors.
- The derivative component causes the controller output to decrease if the system's output is increasing rapidly. The derivative response is proportional to the rate of change of the process variable.

The following figure shows the principles of how these terms are generated and applied:

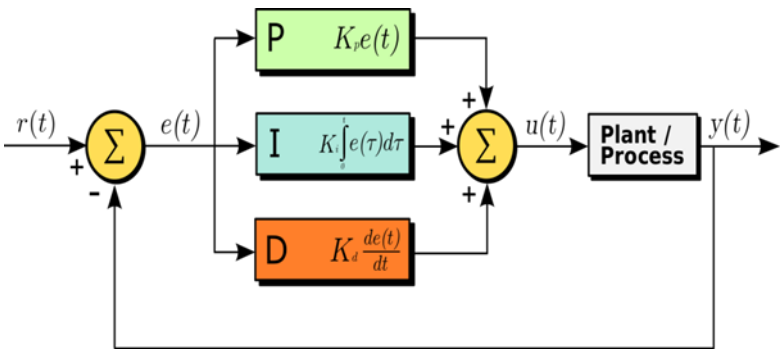

Figure 3.1: PID Controller Block Diagram

3.2 Mathematics of PD Control of Quadrotors:

The dynamic equations of the Quadrotor are given in Equations (2.1) and (2.2). The Equation (2.1) concerns the dynamics of linear equation and the Equation (2.2) concerns the dynamics of angular motion.

- $$m\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + {}^{W}R_{B} \begin{bmatrix} 0 \\ 0 \\ u_1 \end{bmatrix} \quad (2.1)$$

- $$\mathbf{I} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = u_2 - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \mathbf{I} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.2)$$

Where:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} c\theta & 0 & -c\phi s\theta \\ 0 & 1 & s\phi \\ s\theta & 0 & c\phi c\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \rightarrow \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} c\theta & 0 & s\theta \\ s\theta t\phi & 1 & -c\theta t\phi \\ -\frac{s\theta}{c\phi} & 0 & \frac{c\theta}{c\phi} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.3)$$

$[p \quad q \quad r]^{\mathbf{T}}$**:** body angular accelerations measured by the gyroscope.

$[\phi \quad \theta \quad \psi]^{\mathbf{T}}$**:** Roll, Pitch and Yaw angles.

m: system mass.

I: system moment of inertia.

u_1: The thrust input.

u_2: The moment input (3 × 1 vector) and

$$^{W}R_{B} = \begin{bmatrix} c\psi c\theta - s\phi s\psi s\theta & -c\phi s\psi & c\psi s\theta + c\theta s\phi s\psi \\ c\theta s\psi + s\phi c\psi s\theta & c\phi c\psi & s\psi s\theta - c\theta s\phi c\psi \\ -c\phi s\theta & s\phi & c\phi c\theta \end{bmatrix} \quad (2.4)$$

The desired trajectory is:

$$\mathbf{r}_T = \begin{bmatrix} x_{\text{des}} \\ y_{\text{des}} \\ z_{\text{des}} \\ \psi_{\text{des}} \end{bmatrix} \quad (2.5)$$

Let us define:

$$e_p = r_T - r$$

$$e_v = r_T - r$$

And we want:

$$\ddot{r}_T - \ddot{r}_c + k_{d,r}.e_v + k_{p,r}.e_p = 0$$

Here: $\ddot{\boldsymbol{r}}_{\boldsymbol{c}}$ is the commanded acceleration, calculated by the controller. We design the control for hovering and linearize the dynamics at the hover configuration, where we have**:**

$u_1 \approx mg$, $\theta \approx 0$, $\phi \approx 0$, $\psi \approx \psi_0$, $u_2 \approx 0$, $p \approx 0$, $q \approx 0$ and $r \approx 0$.

Using these approximations and all the previous data, one can deduce the following equations:

$$\begin{aligned}
\phi_c &= \frac{1}{g}\left(\ddot{r}_{1,c}\sin(\psi_{\text{des}}) - \ddot{r}_{2,c}\cos(\psi_{\text{des}})\right) \\
\theta_c &= \frac{1}{g}\left(\ddot{r}_{1,c}\cos(\psi_{\text{des}}) + \ddot{r}_{2,c}\sin(\psi_{\text{des}})\right) \\
\psi_c &= \psi_{\text{des}} \\
\ddot{r}_{3,c} &= \ddot{r}_{3,\text{ des}} + k_{d,3}\left(\dot{r}_{3,\text{ des}} - \dot{r}_3\right) + k_{p,3}\left(r_{3,\text{ des}} - r_3\right)
\end{aligned}$$

The control laws can be written as:

$$u_1 = m\left(g + \ddot{r}_{3,c}\right)$$

$$u_2 = I\begin{bmatrix} k_{p,\phi}(\phi_c - \phi) + k_{d,\phi}(p_c - p) \\ k_{p,\theta}(\theta_c - \theta) + k_{d,\theta}(q_c - q) \\ k_{p,\psi}(\psi_c - \psi) + k_{d,\theta}(q_c - q) \end{bmatrix}$$

Using Equation (2.3) and (2.4) one can get:

$[p_c, q_c, r_c]^T$.

Also note that:

$$[r_1, r_2, r_3]^T = [x, y, z]^T$$

3.3 PID Controller Simulation:

The objective of the control is to track a Circular Trajectory. Figure 3.2 shows the SIMULINK model of the quadrotor with the position and attitude control blocks in loop. The system has been tested with and without disturbances. The disturbance is wind with velocity vector: $Vw = 4\hat{}i + 4\hat{}j + 4\hat{}k$ m.s^{-1}.

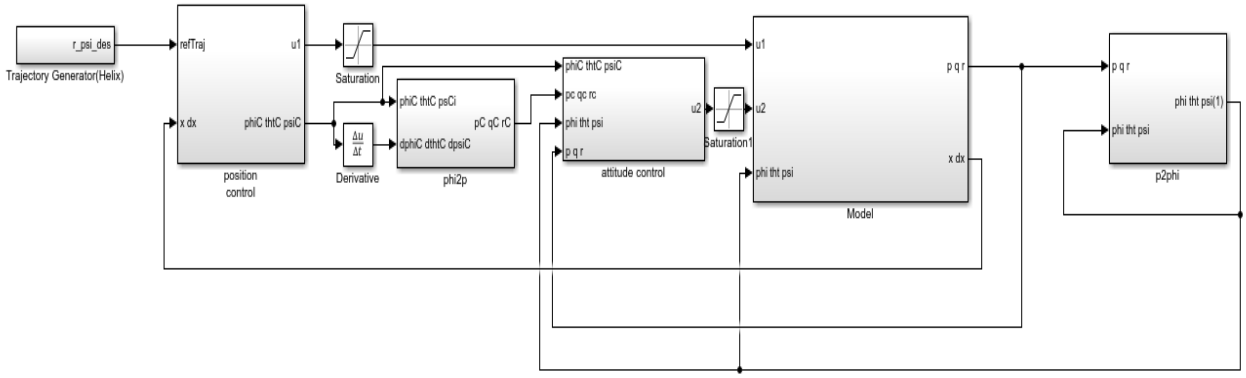

Figure 3.2: Global view of the Simulink model of system with PD control

3.3.1 Results without Disturbances:

Without any disturbance, the tracking by PD control is excellent and the tracking error is negligible. The Figures 10, 11, 12 show the positions, orientations, control inputs and the trajectory error in the absence of disturbances.

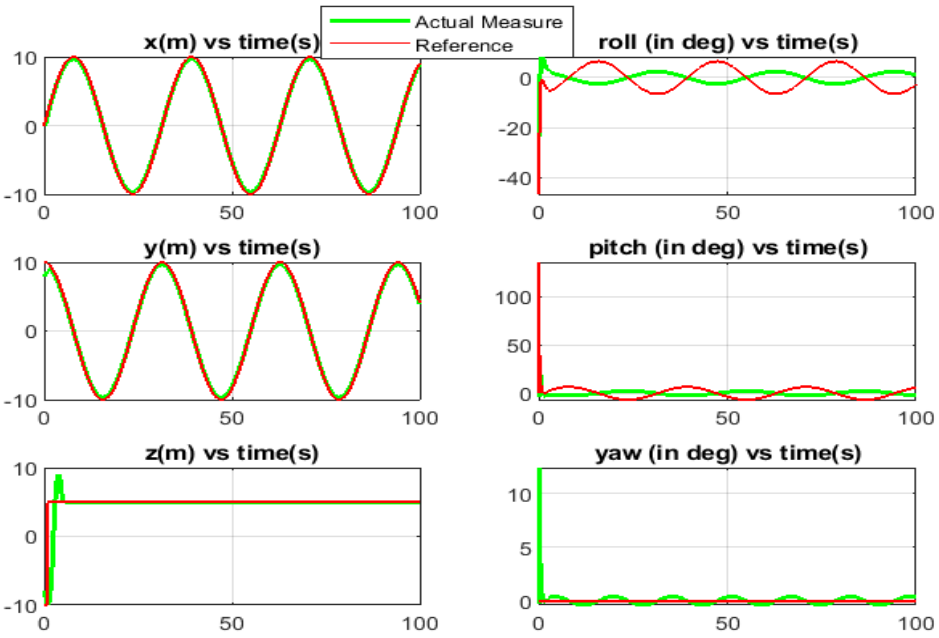

Figure 3.3: Position and Orientation vs Time

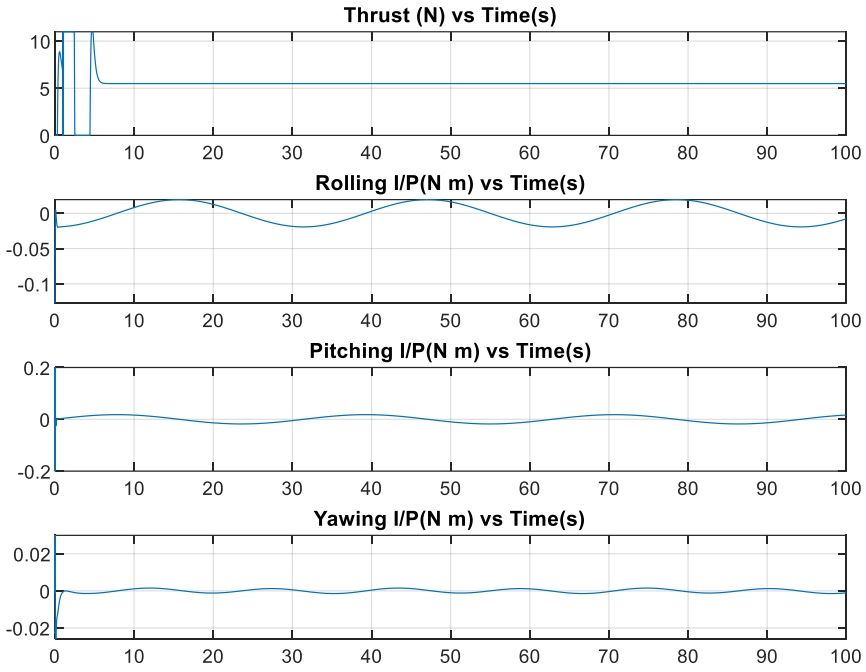

Figure 3.4: Thrust, Rolling, Pitching and Yawing Inputs vs Time

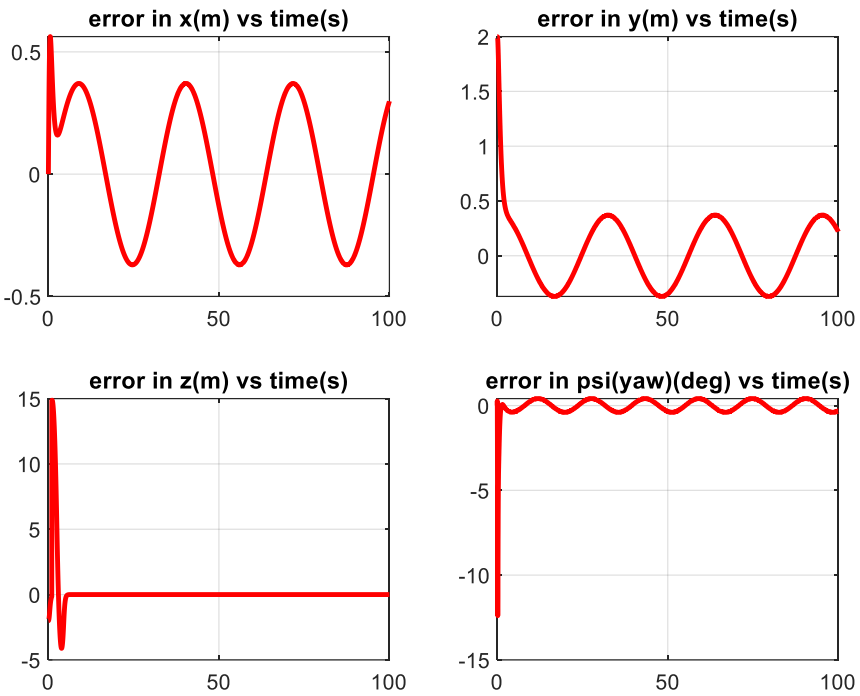

Figure 3.5: Errors in x, y, z and yaw

3.3.2 Results with Disturbances:

The disturbance is wind of velocity:

$$V_w = 4\hat{\imath} + 4\hat{\jmath} + 4\hat{k}\ \text{m}.s^{-1}$$

It is applied as a step input at time t = 25 s. The Figures 13, 14, 15 show the positions, orientations, control inputs and the trajectory error in the presence of disturbances. It can be concluded from the plots that the control is not robust enough to tolerate the effect of winds. The system becomes completely unstable with the addition of wind and all the coordinates except z diverge to ∞.

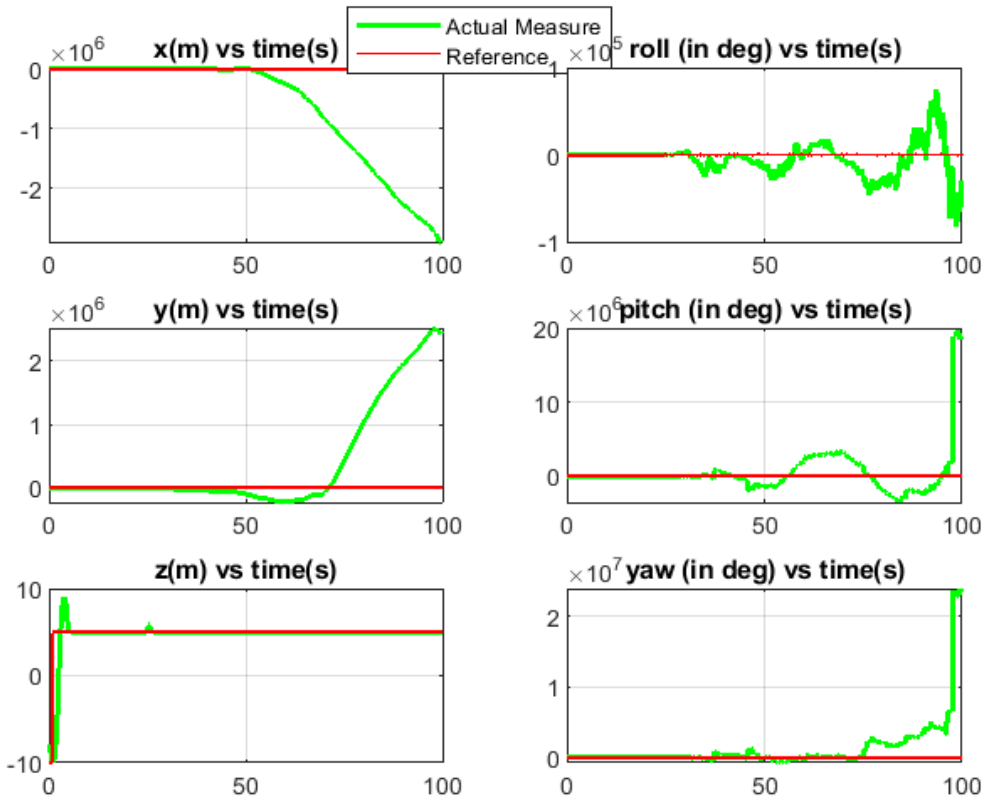

Figure 3.6: Position and Orientation vs Time [With Disturbance at 25s]

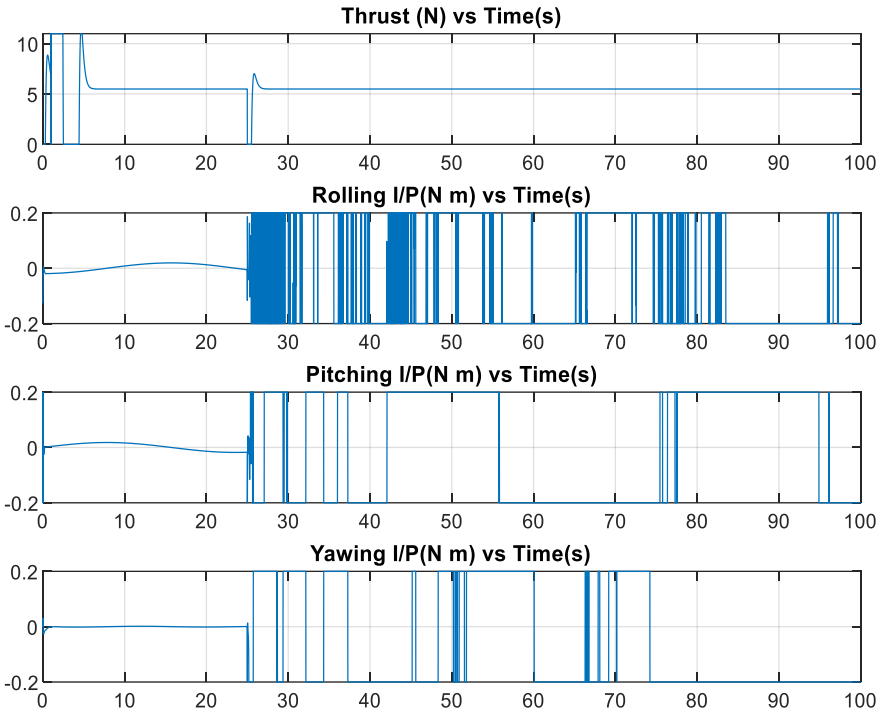

Figure 3.7: Thrust, Rolling, Pitching and Yawing Inputs vs Time [With Disturbance at 25s]

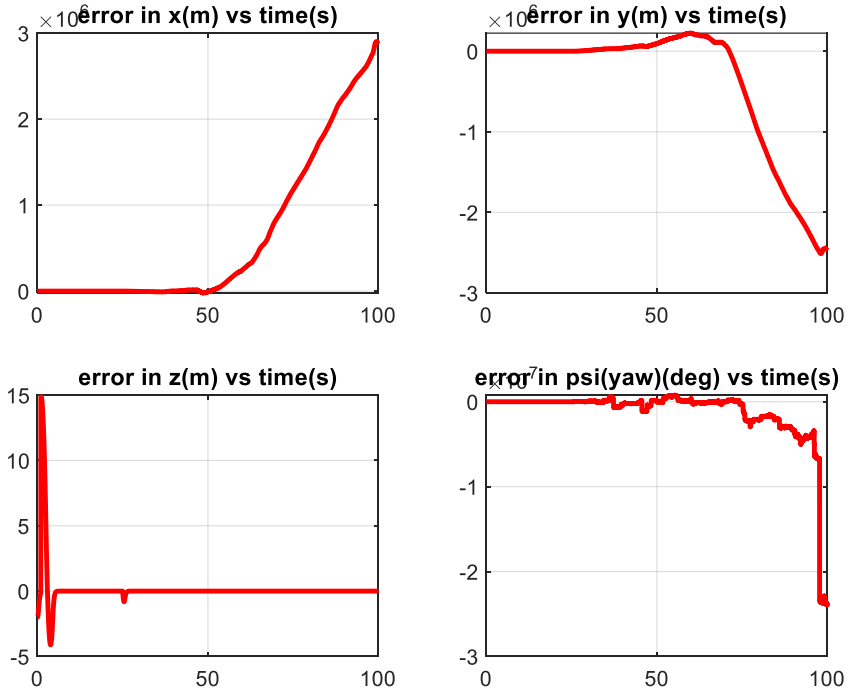

Figure 3.8: Errors in x, y, z and yaw [With Disturbance at 25s]

The main advantages of the PD control are:

- The tuning process is done by changing each PD coefficients of Roll then Pitch then Yaw and see their effect until the system stabilizes, hence it is not complicated.
- Few tuning parameters.
- Easy to understand and implement.

However, this controller has some important drawbacks:

- The control is not robust to noise and disturbances. The derivative term amplifies the noise, hence it makes problem in implementation.
- Cannot deal with multiple inputs and outputs at the same time.

4 Chapter IV: Backstepping Control

4.1 Introduction to backstepping

In control theory, backstepping is a technique developed in 1990 by Petar V. Kokotovic and others [16] for designing stabilizing controls for a special class of nonlinear dynamical systems. The basic control principal of backstepping is that: firstly decompose the complex nonlinear system into several subsystems which are no more than the order of nonlinear system. And then design the lyapunov function and virtual function for each subsystem. Because of this recursive structure, the designer can start the design process at the known-stable system and "back out" new controllers that progressively stabilize each outer subsystem. The process terminates when the final external control is reached. Hence, this process is known as backstepping.

4.2 Mathematics of Backstepping Control

The dynamic model presented in Equation 1.42 is rewritten for the sake of convenience. The state variables are rewritten as follows:

$$
\begin{aligned}
\dot{x_1} &= \dot{\phi} = x_2 \\
\dot{x_2} &= \ddot{\varphi} = x_4 x_6 a_1 - x_4 \Omega_r a_2 + b_1 U_2 \\
\dot{x_3} &= \dot{\theta} = x_4 \\
\dot{x_4} &= \ddot{\theta} = x_2 x_6 a_3 + x_2 \Omega_r a_4 + b_2 U_3 \\
\dot{x_5} &= \dot{\psi} = x_6 \\
\dot{x_6} &= \ddot{\psi} = x_2 x_4 a_5 + b_3 U_4 \\
\dot{x_7} &= \dot{z} = x_8 \\
\dot{x_8} &= \ddot{z} = -g + \frac{U_1}{m}(\cos x_1 \cos x_2) \\
\dot{x_9} &= \dot{x} = x_{10} \\
\dot{x_{10}} &= \ddot{x} = \frac{U_1}{m}(\sin x_1 \sin x_5 + \cos x_1 \sin x_3 \cos x_5) \\
\dot{x_{11}} &= \dot{y} = x_{12} \\
\dot{x_{12}} &= \ddot{y} = \frac{U_1}{m}(-\sin x_1 \cos x_5 + \cos x_1 \sin x_3 \sin x_5)
\end{aligned}
$$

4.2.1 Roll Controller:

The states x_1 and x_2 are the roll and its rate of change. Considering the first two state variables only:

$$\dot{x_1} = x_2 \tag{3.1}$$

$$\dot{x}_2 = x_4 x_6 a_1 - x_4 \Omega_r a_2 + b_1 U_2 \tag{3.2}$$

The roll angle subsystem is in the strict feedback form (only the last state is a function of the control input U_2) which makes it easy to pick a positive definite Lyapunov function for it.

$$V_1 = \frac{1}{2} {z_1}^2 \tag{3.3}$$

Where z_1 is the error between the desired and actual roll angle defined as follows.

$$z_1 = x_{1d} - x_1 \tag{3.4}$$

The time derivative of the Lyapunov function defined in Equation (3.3) is derived to be.

$$\dot{V}_1 = z_1 \dot{z_1} \tag{3.5}$$

$$= z_1 \, (\dot{x_{1d}} - \dot{x_1}) \tag{3.6}$$

And from Equation (3.1) this can be written as:

$$\dot{V}_1 = z_1 (\dot{x_{1d}} - x_2) \tag{3.7}$$

According to Krasovskii-LaSalle principle, the system is guaranteed to be stable system if the time derivative of a positive definite Lyapunov function is negative semi definite [16].To achieve that, we choose a positive definite bounding function:

$$W_1(z) = \; c_1 {z_1}^2$$

To bound $\dot{V}_1$ as in Equation (3.8).This choice of W_1(z) is also a common choice for a bounding function for strict feedback systems [16].

$$\dot{V}_1 = z_1 (\dot{x}_{1d} - x_2) \; \leq \; -c_1 {z_1}^2 \tag{3.8}$$

Where c_1 is a positive constant, to satisfy this inequality the virtual control input can be chosen to be:

$$(x_2)_{desired} \; = \dot{x}_{1d} + c_1 z_1 \tag{3.9}$$

Defining a new error variable z_2 to be the deviation of the state x_2 from its desired value:

$$z_2 = x_2 - \dot{x}_{1d} - c_1 z_1 \tag{3.10}$$

Substituting by the value of x_2 from the last equation in equation (3.7) we get:

$$\begin{aligned}\dot{V}_1 &= z_1 \dot{z_1} \\ &= z_1(\dot{x}_{1d} - x_2) \\ &= z_1(\dot{x}_{1d} - (z_2 + \dot{x}_{1d} + c_1 z_1)) \\ &= -z_1 z_2 - c_1 {z_1}^2 \end{aligned} \tag{3.11}$$

The presence of the term $z_1 z_2$ in $\dot{V}_1$ may not lead to a negative semi-definite time derivative but this will be taken care of in the next iteration of the backstepping algorithm, the next step is to augment the first Lyapunov function V_1 with a quadratic term in the second error variable z_2 to get a positive definite:V_2,

$$V_2 = V_1 + \frac{1}{2} {z_2}^2 \tag{3.12}$$

With time derivative,

$$\dot{V}_2 = \dot{V}_1 + z_2 \dot{z_2}$$

$$\dot{V}_2 = -z_1 z_2 - c_1 {z_1}^2 + z_2(\dot{x_2} - \ddot{x_{1d}} - c_1 \dot{z_1}) \tag{3.13}$$

Choosing $W_2(\mathrm{z}) = -c_1 {z_1}^2 - c_2 {z_2}^2$to be the positive definite bounding function, with c_2 is positive constant and by replacing the value of $\dot{x_2}$ from equation (3.2) leads to the following simplified inequality:

$$\begin{aligned}\dot{V}_2 &= -z_1 z_2 - c_1 {z_1}^2 + z_2(x_4 x_6 a_1 - x_4 \Omega_r a_2 + b_1 U_2 - \ddot{x_{1d}} - c_1 \dot{z} \\ &\leq -c_1 {z_1}^2 - c_2 {z_2}^2 \end{aligned} \tag{3.14}$$

The control input U_2 can be written as:

$$U_2 = \frac{1}{b_1}(-c_2 z_2 + z_1 - x_4 x_6 a_1 + x_4 \Omega_r a_2 + \ddot{x_{1d}} + c_1 \dot{x_{1d}} - c_1 x_2) \tag{3.15}$$

4.2.2 Pitch controller:

To derive the pitch controller we follow the same steps as we did before for the Roll controller, extracting the third and the fourth states of the state space model in Equation (1.42) which are the pitch angle and its rate of change, we get:

$$\dot{x_3} = x_4 \tag{3.16}$$

$$\dot{x_4} = x_2 x_6 a_3 + x_2 \Omega_r a_4 + b_2 U_3 \tag{3.17}$$

And the error in pitch is defined as $z_3 = x_{3d} - x_3$leading to a positive definite Lyapunov function:

$$V_3 = \frac{1}{2} {z_3}^2 \tag{3.18}$$

With time derivative:

$$\dot{V_3} = z_3 \dot{z_3}$$

$$= z_3(\dot{x_{3d}} - x_4) \tag{3.19}$$

Choosing $W_3(z) = -c_3 {z_3}^2$ to be the bounding function where c_3 a positive constant, the desired x_4 state is:

$$(x_4)_{desired} = \dot{x_{3d}} + c_3 z_3 \tag{3.20}$$

And the error in state x_4 is:

$$z_4 = x_4 - \dot{x_{3d}} - c_3 z_3 \tag{3.21}$$

Substituting by the value of x_4 from the last equation in equation (3.19) we get:

$$\dot{V_3} = z_3 \dot{z_3}$$

$$= z_3\left(\dot{x_{3d}} - (z_4 + \dot{x_{3d}} + c_3 z_3)\right)$$

$$= -z_3 z_4 - c_3 {z_3}^2 \tag{3.22}$$

Augmenting the previous Lyapunov function with a quadratic term in the error variable z_4:

$$V_4 = V_3 + \frac{1}{2}{Z_4}^2 \tag{3.23}$$

Defining a new bounding function to be: $W_4(z) = -c_3{z_3}^2 - c_4{z_4}^2$ with c_4 a positive constant, the following inequality can be reached:

$$\dot{V_4} = -z_3 z_4 - c_3{z_3}^2 + z_4(\dot{x_4} - \ddot{x_{3d}} - c_3 z_3) \leq -c_3{z_3}^2 - c_4{z_4}^2 \tag{3.24}$$

Replacing x_4 with its definition from equation (3.17) and solving for U_3.The roll angle control input is found to be:

$$U_3 = \frac{1}{b_2}(-c_4 z_4 + z_3 - x_2 x_6 a_3 - x_2 \Omega_r a_4 + \ddot{x_{3d}} + c_3 \dot{x_{3d}} - c_3 x_4) \tag{3.25}$$

4.2.3 Yaw Controller:

Following exactly the same steps as the roll and pitch controllers, the control input for the yaw angle is derived to be:

$$U_4 = \frac{1}{b_3}(-c_6 z_6 + z_5 - x_2 x_4 a_5 + \ddot{x}_{5d} + c_5 \dot{x}_{5d} - c_5 x_6) \tag{3.26}$$

With

$$z_5 = x_{5d} - x_5 \tag{3.27}$$

$$z_6 = x_6 - \dot{x}_{5d} - c_5 z_5 \tag{3.28}$$

And c_5 and c_6 are positive constants.

4.2.4 Altitude Controller:

For the altitude controller, the control input U_1 is derived in the same manner as $U_2, U_3, and\ U_4$ to be:

$$U_1 = \frac{m}{\cos x_1 \cos x_3}(z_7 + g - \ddot{x}_{7d} - c_7 \dot{x}_{7d} + c_7 x_8 - c_8 z_8) \tag{3.29}$$

With

$$z_7 = x_{7d} - x_7 \tag{3.30}$$

$$z_8 = x_8 - \dot{x}_{7d} - c_7 z_7 \tag{3.31}$$

And c_7 and c_8 are positive constants

4.3 Backstepping controller simulation

The objective of the control is to track a trajectory. Figure 4.1 shows the SIMULINK model of the quadrotor with the position and attitude control blocks in loop. The system has been tested with and without disturbances. The disturbance is wind with velocity vector

$V_w = 5\hat{\imath} + 5\hat{\jmath} + 5\hat{k}\ m.s^{-1}$

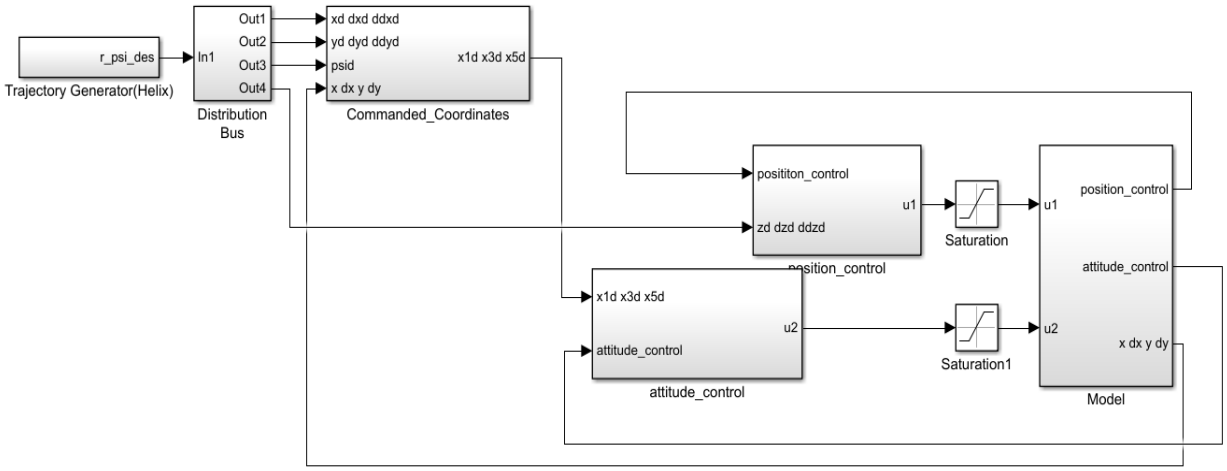

Figure 4.1: Global view of the Simulink model of the system with the Backstepping control

4.3.1 Results without Disturbances:

Without any disturbance, Backstepping control gives comparable results with proportional derivative. The Figures 4.2, 4.3 and 4.4 show the positions, orientations, control inputs and the trajectory error in the absence of disturbances.

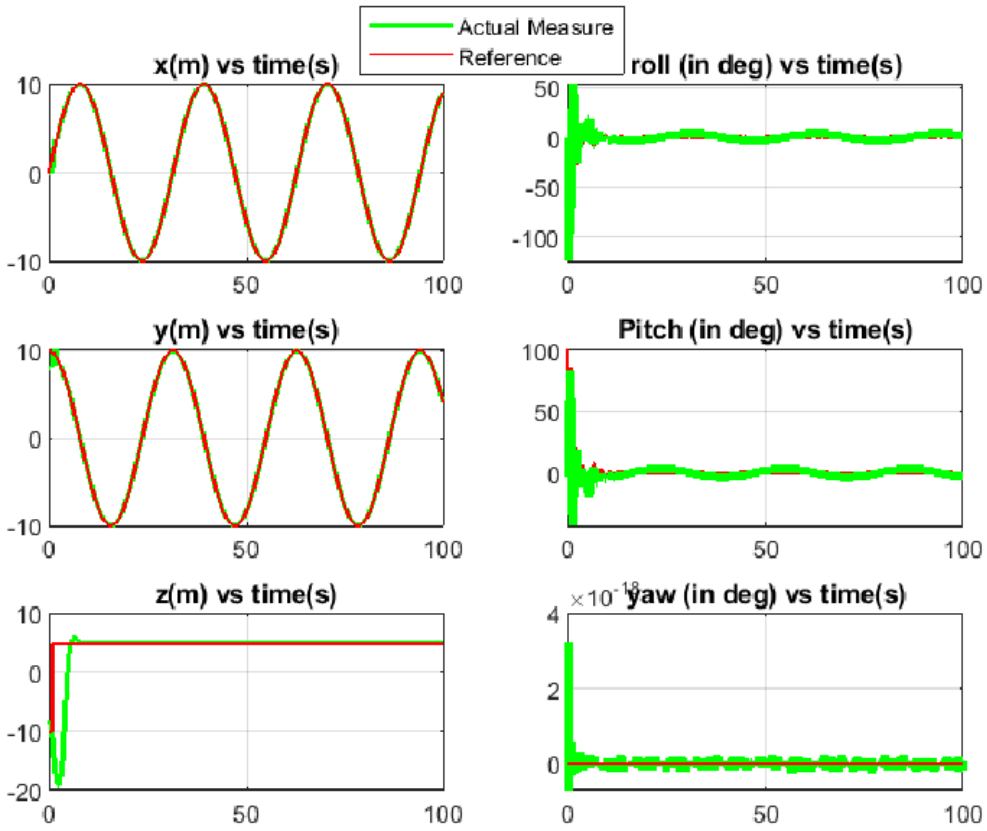

Figure 4.2: Position and Orientation vs Time [Without Disturbance]

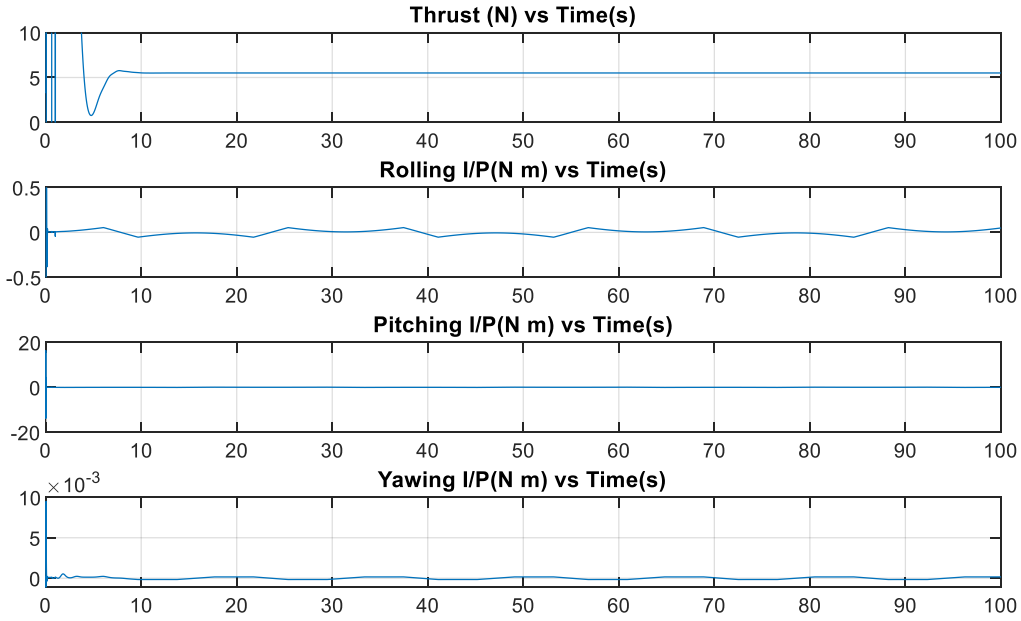

Figure 4.3: Thrust, Rolling, Pitching and Yawing Inputs vs Time [Without Disturbance]

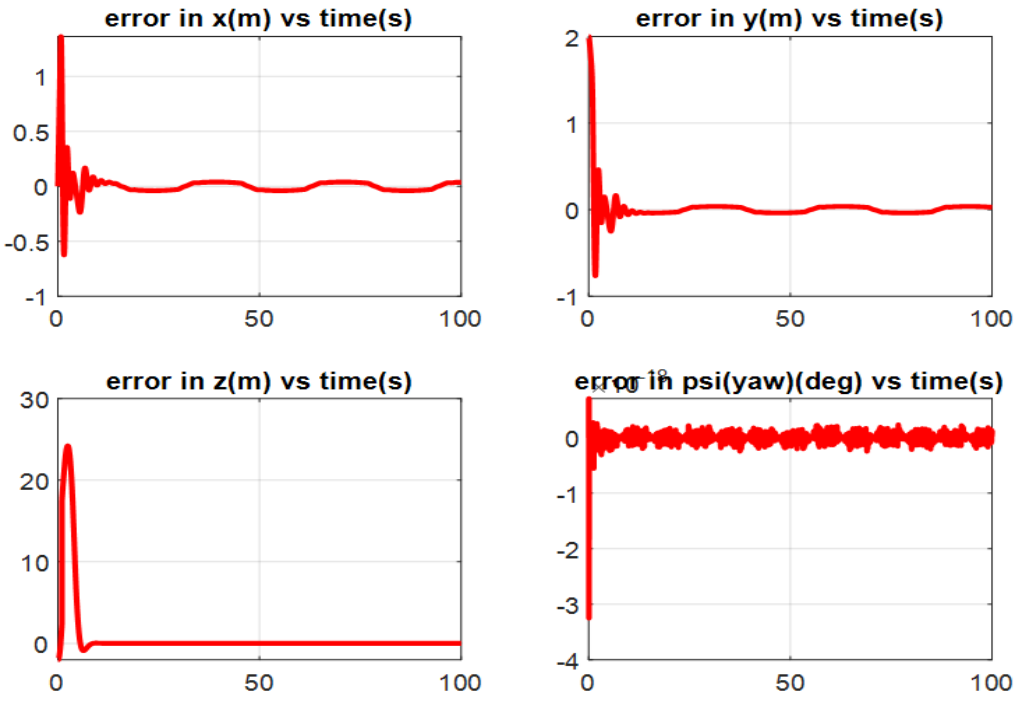

Figure 4.4: Errors in x, y, z and yaw

4.3.2 Results with disturbances:

The disturbance is wind $V_w = 5\hat{\imath} + 5\hat{\jmath} + 5\hat{k}\ m.s^{-1}$ It is applied as a step input at time t=25 s. The figures 4.5, 4.6, 4.7 show the position, orientations, control inputs and the trajectory error in the presence of disturbances.

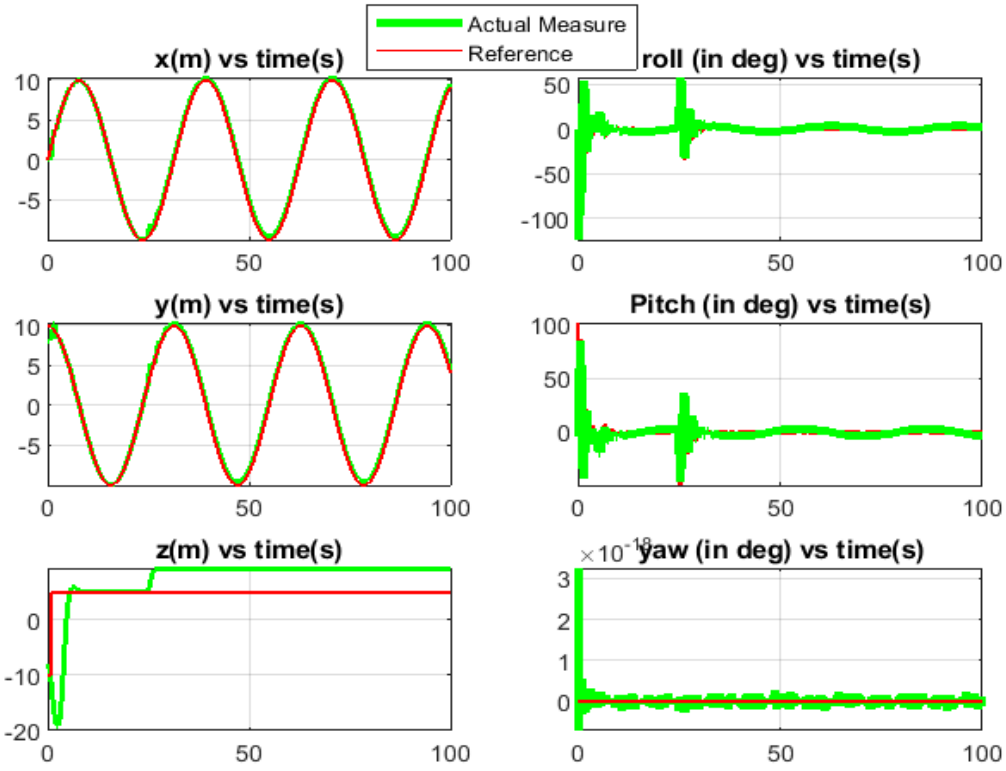

Figure 4.5: Position and Orientation vs Time [With Disturbance at 25s]

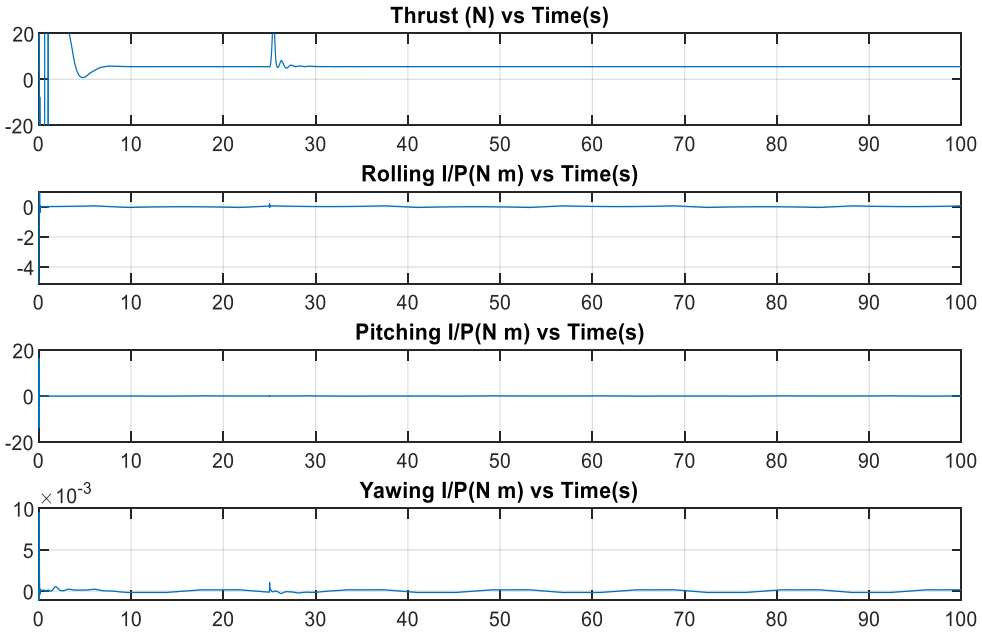

Figure 4.6: Thrust, Rolling, Pitching and Yawing Inputs vs Time [With Disturbance at 25s]

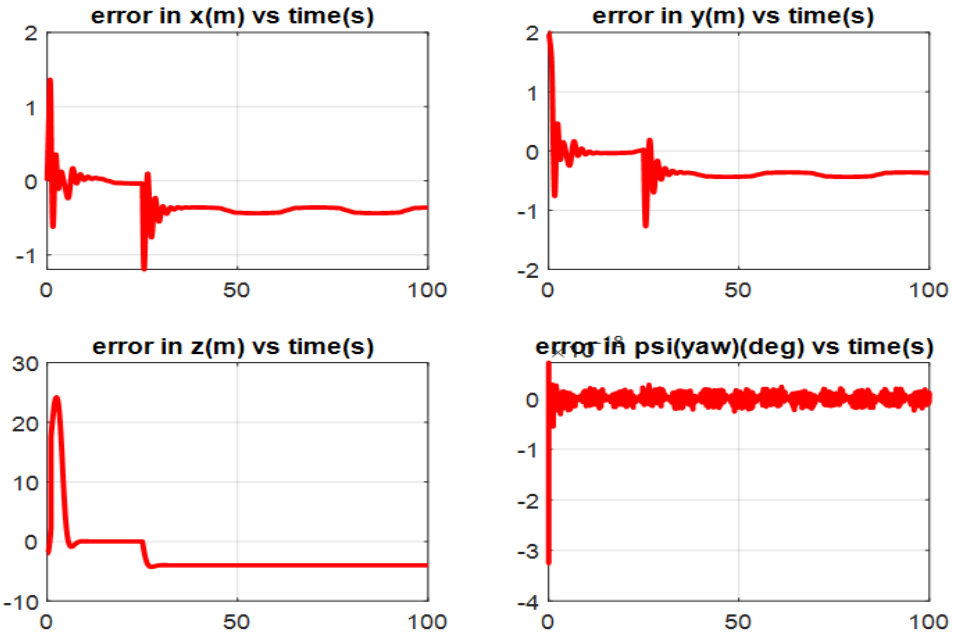

Figure 4.7: Errors in x, y, z and yaw [With Disturbance at 25s]

The Backstepping control comes out with various advantages:

- ❒ The backstepping control design is derived based on the Lyapunov function so that the stability of the system can be guaranteed.
- ❒ Unlike the feedback linearization method, which has issues like the precise model requirement and the cancellation of valuable nonlinear components, the backstepping approach provides a variety of design tools for nonlinearity accommodation and can avoid undesirable cancellations.

Its disadvantages are:

- ❒ Over-parameterization.
- ❒ Difficult to choose proper parameters.
- ❒ The theory is mathematically exacting

5 Chapter V Gain scheduling based PD Controller:

5.1 Introduction to Gain Scheduling :

A gain scheduling based PD controller is proposed to overcome the linear PD controller's limitations of only being able to work in the linear near hover region. Gain scheduling is based on the development of a set of controllers for various operating points and switching between these controllers depending on the system's operating point [45]. In this work, a family of PD controllers will be developed, each PD controller having different controller gains and will be able to stabilize the quadrotor system in a certain range of operation. Gain scheduling will then be used to choose an appropriate controller from the family of developed PD controllers. This strategy turns the traditional PD controller into an adaptive controller since the controller's parameters are adapting to different operating conditions. The acquired gains were used in a look up table fashion in the developed MATLAB/Simulink model.

The Block Diagram for the developed Gain Scheduling based PD controller is shown in Figure 5.1.

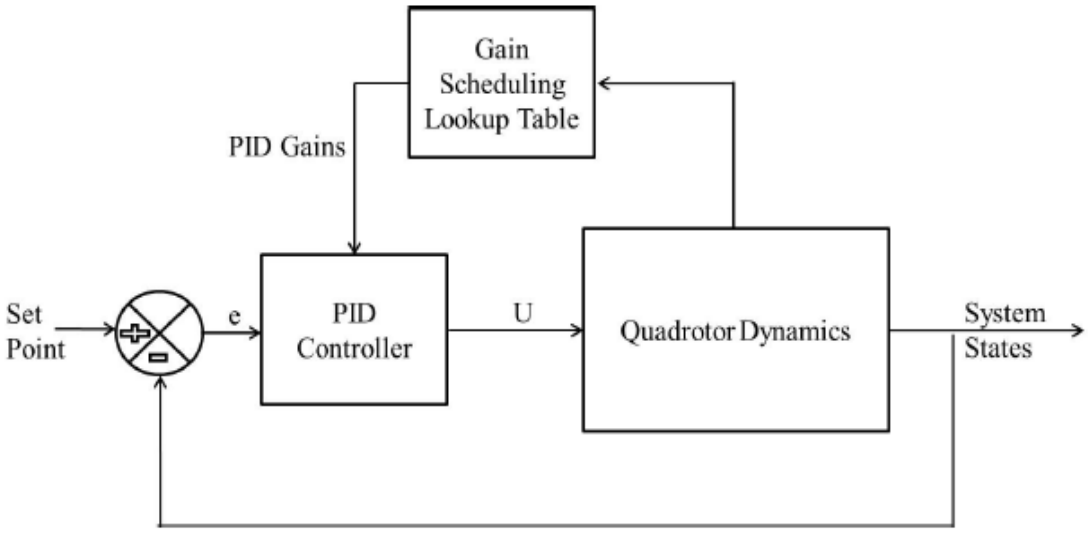

Figure 5.1: Gain Scheduling Block Diagram

5.2 Gain Scheduling Based PD Controller Simulation Results:

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

5.2.1 Attitude controller:

In this chapter we command the quadrotor to follow a varying trajectory to show the performance of the Gain Scheduling PD against the classical PD controller.

For the Roll and Pitch control, GA was also used to find the controller gains for a set of operating points.

Table 1.1 shows the desired roll and pitch angles with their control gains and performance. Figure 5.2 shows the comparison between the response of the Gain Scheduling PD and the classical PD in following a varying trajectory.

Table 1.1: Attitude Gain Scheduling Based PD controller Gains

Desired Attitude (deg)	K_P	K_D	Settling Time (S)
2	6.23	0.682	0.9
4	5.93	0.680	1.2
6	7.06	0.737	1.4
8	7.02	0.742	1.54
10	4.18	0.563	1.61
12	5.74	0.622	1.69
14	5.85	0.678	1.76
16	5.30	0.633	1.82
18	3.05	0.486	1.9
20	5.07	0.656	2.1

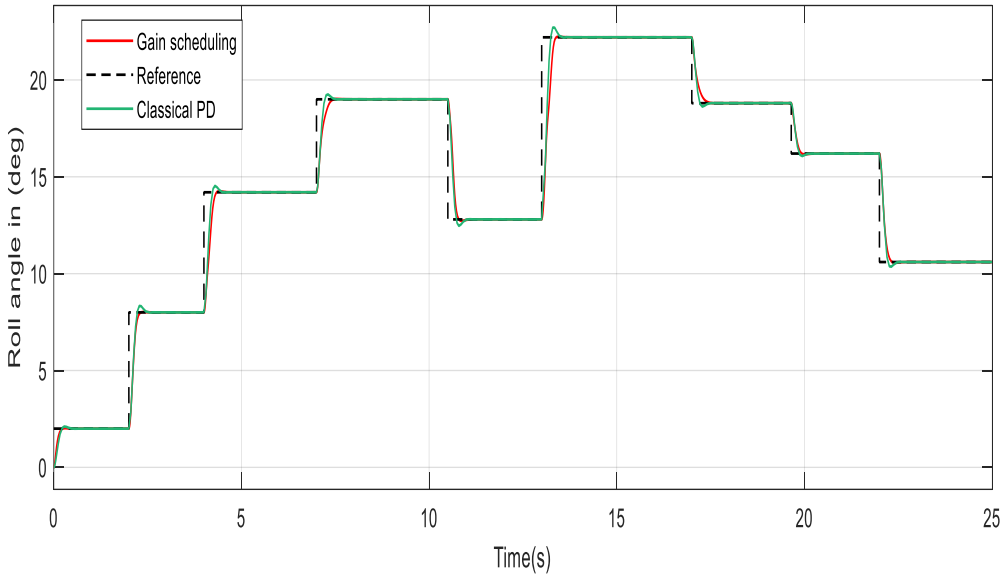

Figure 5.2: Attitude Response

5.2.2 Heading Controller:

Similar to the attitude controller, a look up table was synthesized for the heading controller. The controller gains and their respective performances at multiple desired points are shown in table 5.2.and the response is shown in figure 5.3.

Table 5.2: Heading Gain Scheduling Based PD controller Gains

Desired heading (deg)	K_P	K_D	Settling Time (S)
2	6.62	0.995	1.4
4	3.65	0.671	1.5
6	4.90	0.924	1.5
8	4.00	0.633	1.75
10	4.21	0.784	2.0
12	3.45	0.800	2.2
14	5.32	0.844	2.2
16	3.12	0.765	2.5

18	3.42	0.861	2.71
20	5.31	0.992	2.8

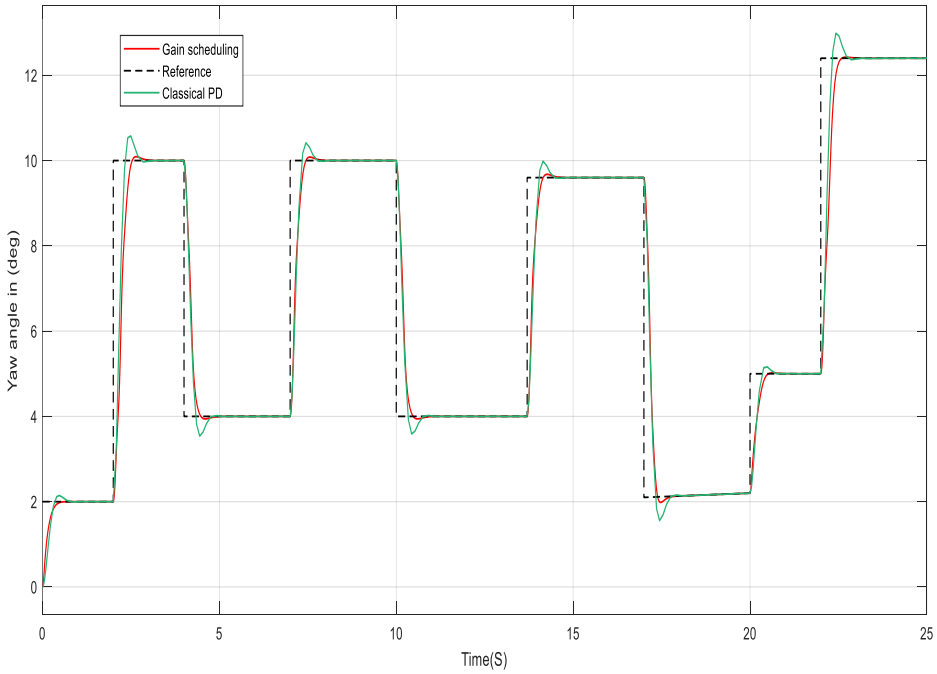

Figure 5.3: Heading Response

5.2.3 Altitude Control:

Table 5.3 shows the tuned parameters of the PD controller for different desired altitudes.

Figure 5.4 shows the comparison between the response of the Gain Scheduling PD and the classical PD.

Table 5.3: Altitude Gain Scheduling Based PD controller Gains

Desired Altitude (m)	K_P	K_D	Settling Time (S)
1	8.60	3.60	1.02
2	5.61	2.95	1.41
3	5.01	2.88	1.47
4	9.22	3.44	1.53
5	5.61	3.00	1.83
6	6.22	2.89	2.12

7	5.18	2.88	2.49
8	6.36	3.44	2.78
9	5.00	2.67	3.36
10	5.31	3.90	3.76

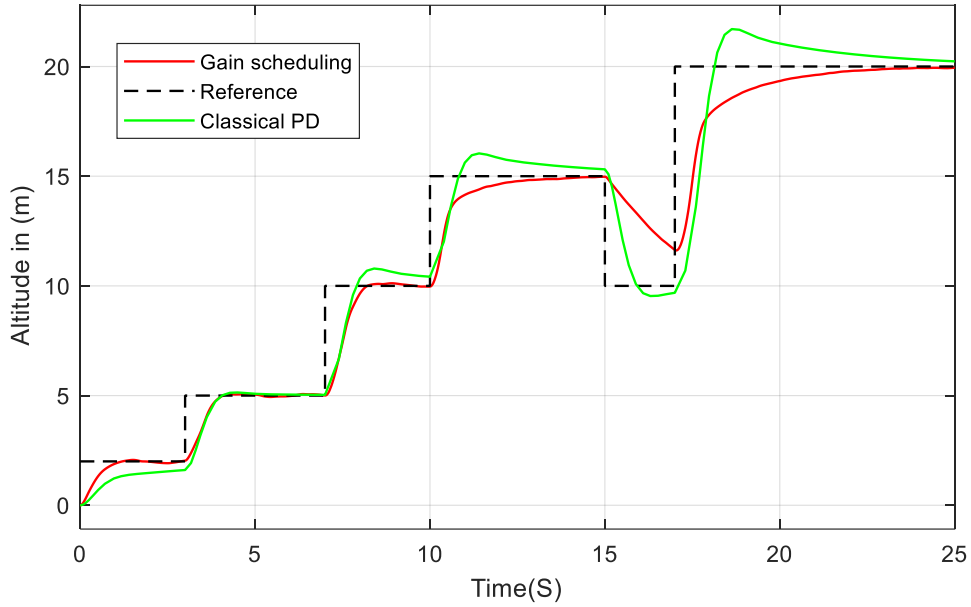

Figure 5.4: Altitude Response

5.3 Control inputs:

Figure 5.5 shows the control inputs generated by the gain scheduling based PD controller

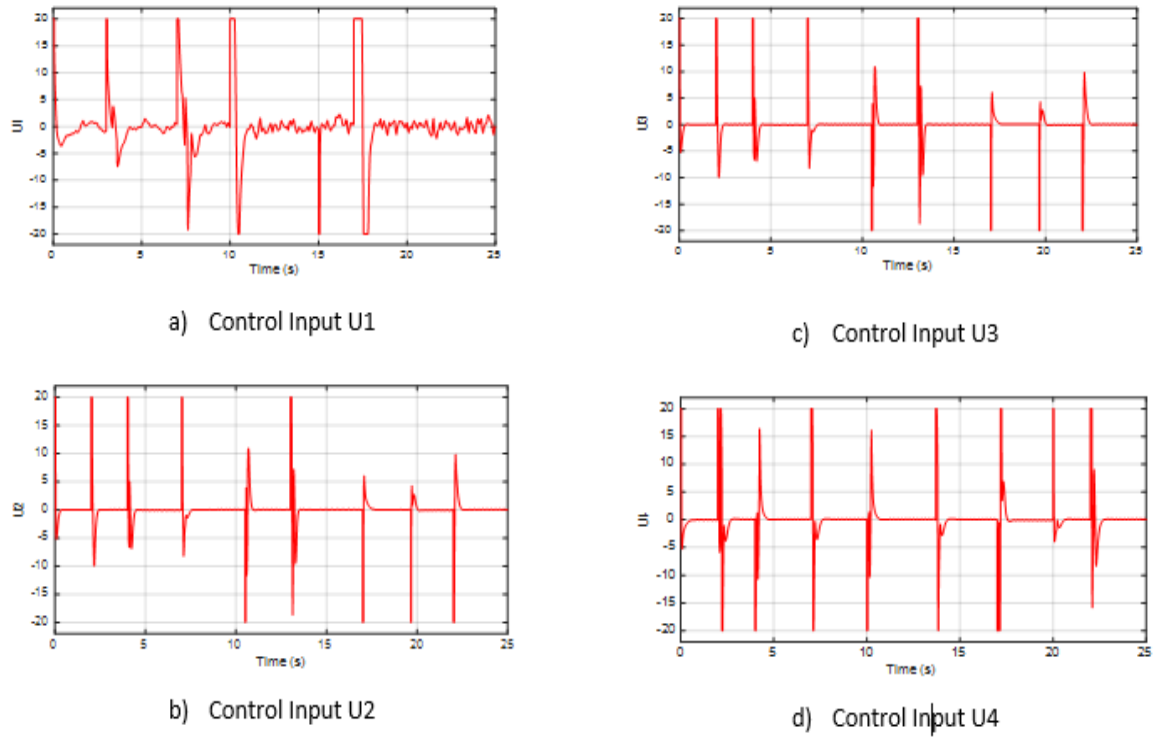

Figure 5.5: Control Inputs

The main advantages of the Gain Scheduling PD controller are:

- ❒ Gain scheduling based PD Control increases the performance of the quadrotor in tracking the desired trajectory, which gives better performance in practical applications.
- ❒ Easy design and implementation.

However, an important drawback of this controller is the criticality of the switching time, moving from one set of controller gains to another must be done in a small amount of time to ensure good performance. This is a serious issue in some quadrotor applications that rely heavily on Gain Scheduling, such as load drop applications, where the quadrotor may overshoot and become unstable if the switching is not completed after the load is dropped

Comparison between Developed Controllers:

The Backstepping controller was applied in order to operate the system outside its linear region, the controller is able to stabilize the system with a good dynamic performance. On the other hand, away from hover, the PD and the gain scheduled PD controllers failed to stabilize the system due to the fact that they represent linear controllers. The three developed controllers have equivalent dynamic performances in the linear region (near hover).

From the resulted control signals, U_1 through U_4, we conclude that the PD controller is the most energy efficient figure 3.4 followed by the Backstepping figure 4.3. In Gain Scheduling based PD, the abrupt change from one set of control gains to the other causes spikes in the control signals as shown in figure 5.5.

If the controller is used on a real system, these abrupt changes will probably cause system instability. To avoid this effect, a smoothing filter must be applied before the signals are sent to the actuators, resulting in a smooth transition.

In the presence of disturbance (windy conditions), the PD and Backstepping controllers performed similarly in terms of controlling the quadrotor's altitude, Backstepping did suffer a little performance degradation in stabilizing the quadrotor’s attitude.

Conclusion

The goal of this work was to derive a complete mathematical model of the quadrotor Unmanned Aerial Vehicle (UAV) and develop linear and nonlinear control algorithms to stabilize the quadrotor's states, such as altitude, attitude, heading, and position in space, as well as to test the performance of these controllers using comparisons via computer simulation. Three control approaches were developed; a linear Proportional-Derivative (PD) controller, a Gain Scheduling based PD controller and a nonlinear Backstepping controller. A complete simulation was implemented on MATLAB/SIMULINK. The simulation environment was used to evaluate and compare the dynamic performances of the described controllers under various input situations. The tuning of the parameters of the three used controllers is done using genetic algorithm GA. Near hovering operation the three controllers performed similarly, The Gain Scheduling based PD controller gave a better performance than the traditional PD controller when the quadrotor was commanded to follow a varying trajectory. Backstepping controller performed better outside of the linear hovering zone because of its nonlinear character, moreover, in the presence of disturbance PD and Backstepping controllers performed better.

Future work

As future work, an important addition would be to make the control strategies more robust to disturbances and noises, since our simulation results showed huge degradation of the performance of the controllers when wind disturbance was added. Furthermore, in our work, all model parameters are assumed to be known precisely without any uncertainties, which is not the case in reality; thus, developing adaptive control algorithms to count for system uncertainties would improve the quadrotor's performance when operating in a real environment. Moreover, applying an integral backstepping controller is suggested to stabilize quadrotor attitude. This controller could ensure the promising performance of all the states of the system considering the external disturbances to the system.

References

[1] **Cary, Leslie and Coyne, James. 2012.** ICAO Unmanned Aircraft Systems (UAS), Circular 328.

[2] **Tice, Brian P.** "Unmanned aerial vehicles: The force multiplier of the 1990s." Airpower Journal 5, no. 1 (1991): 41-55.

[3] **Alvarado, Ed (3 May 2021.** *237 Ways Drone Applications Revolutionize Business.* 2021. https://droneii.com/product/drone-application-report-2021.

[4] **Gillula, J.H., Hoffmann, G.M., Huang, H., Vitus, M.P. and Tomlin, C.J., 2011**. Applications of hybrid reachability analysis to robotic aerial vehicles. The International Journal of Robotics Research, 30(3), pp.335-354.

[5] **Li, Jun, and Yuntang Li.** "Dynamic analysis and PID control for a quadrotor." In 2011 IEEE International Conference on Mechatronics and Automation, pp. 573-578. IEEE, 2011.

[6] **Bouabdallah, S., Siegwart, R.:** Backstepping and sliding-mode techniques applied to an indoor micro quadrotor. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Robotics and Automation 2005. ICRA 2005, pp. 2247–2252 (2005).

[7] **Bouabdallah, S.:** Design and control of quadrotors with application to autonomous flying. Ph.D. dissertation, Lausanne (2007).

[8] **Pounds, P., 2007**. Design, construction and control of a large quadrotor micro air vehicle. The Australian National University (Doctoral dissertation, Tesis doctoral).

[9] **J. Yang, Z. Cai, Q. Lin and Y. Wang,** "Self-tuning PID control design for quadrotor UAV based on adaptive pole placement control," 2013 Chinese Automation Congress, 2013, pp. 233-237, doi: 10.1109/CAC.2013.6775734.

[10] **Kendoul, F. (2012),** Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems. J. Field Robotics, 29: 315-378. https://doi.org/10.1002/rob.20414.

[11] **A. Azzam and Xinhua Wang,** "Quad rotor arial robot dynamic modeling and configuration stabilization," 2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR 2010), 2010, pp. 438-444, doi: 10.1109/CAR.2010.5456804.

[12] **Nagaty, A., Saeedi, S., Thibault, C. et al.** Control and Navigation Framework for Quadrotor Helicopters. J Intell Robot Syst 70, 1–12 (2013).

[13] **Derafa, Laloui, Tarek Madani, and Abdelaziz Benallegue**. "Dynamic modelling and experimental identification of four rotors helicopter parameters." In 2006 IEEE international conference on industrial technology, pp. 1834-1839. IEEE, 2006.

[14] **Kugelberg, Ingrid.** "Black-box modeling and attitude control of a quadcopter." (2016).

[15] **Mellinger, Daniel.** Trajectory generation and control for quadrotors. University of Pennsylvania, 2012.

[16] **Krstic, Miroslav, Petar V. Kokotovic, and Ioannis Kanellakopoulos.** Nonlinear and adaptive control design. John Wiley & Sons, Inc., 1995.

[17] **nežka Chovancová, Tomáš Fico, Ľuboš Chovanec, Peter Hubinsk,**Mathematical Modelling and Parameter Identification of Quadrotor (a survey),Procedia Engineering,Volume 96,2014,Pages 172-181,ISSN 1877-7058.

Appendix:

Dynamic model parameter specification:

In system modelling and control the listed below parameter were used to develop the dynamic model where its values that have been used in the simulation are taken from Bouabdallah thesis [7]

Table A.1: Dynamic model parameters and its values

Parameter	Description	value
I_{xx}	quadrotor moment of inertia in x_b	7.5×10^{-3} Kg.m^2
I_{yy}	quadrotor moment of inertia in y_b	7.5×10^{-3} Kg.m^2
I_{zz}	quadrotor moment of inertia in z_b	1.3×10^{-3} Kg.m^2
m	mass of the stack-up frame and electronics	0.56 Kg
l	Length of arm	0.23 m
Jr	Rotor's Inertia	6×10^{-5} Kg.m^2
R_{mot}	Motor circuit resistance	0.6 Ω
K_{mot}	Motor torque constant	5.2 N.m.A^{-1}
K_F	Aerodynamic force constant	3.13×10^{-5} $N.m^2$
K_M	Aerodynamic moment constant	7.5×10^{-7} $N.m.s^2$
g	Gravitational force	9.8 N.m.s^2

Genetic Algorithm:

Genetic-algorithm-PID-controller-tuner is an example of using a genetic algorithm to tune a PID controller. A PID controller is any controller that uses a combination of integral, proportional, and derivative parameters (I, P, D) to follow some target value. We integrate, derive, and multiply the distance from our target by certain constants: K_P, K_I, K_D and these parameters are optimized using GA to get a better performance.

Its working principle is to generate a population of points at each iteration. The best point in the population approaches an optimal solution and selects the next population by computation which uses random number generators.

The pseudocode below explain the steps of GA

Algorithm: GA(n, χ, μ)

// Initialize generation 0:

: = 0;

: = a population of n randomly-generated individuals;

// Evaluate Pk:

$Computefitn(i)$ for each $i \in Pk$;

do

{ // Create generation k + 1:

// 1. Copy:

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

// 2. Crossover:

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

// 3. Mutate:

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

// Evaluate Pk + 1:

$Computefitn(i)$ for each $i \in Pk$;

// Increment:

: = k + 1;

}

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

return the fittest individual from Pk;