

**VIETNAM NATIONAL UNIVERSITY – HOCHIMINH CITY
INTERNATIONAL UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING**



**MODERN MULTIDIMENSIONAL PID CONTROLLER:
STRUCTURE AND DESIGN METHODOLOGY
WITH APPLICATION TO A QUADROTOR**

**BY
PHAN TRUONG BUU**

**A THESIS SUBMITTED TO THE SCHOOL OF ELECTRICAL ENGINEERING IN
PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF SCHOOL OF ELECTRICAL ENGINEERING**

**HOCHIMINH CITY, VIETNAM
2016**

**MODERN MULTIDIMENSIONAL PID CONTROLLER:
STRUCTURE AND DESIGN METHODOLOGY
WITH APPLICATION TO A QUADROTOR**

BY
PHAN TRUONG BUU

APPROVED BY:

THESIS COMMITTEE

ACKNOWLEDGMENT

This thesis is a final part of my years studying in the school of Electrical Engineering at International University. I would like to thank Professor Huynh Huu Tue for his encouragement and support during my time doing this thesis project. The guidance in research methodology and theory I received from him have contributed greatly in this work.

I also want to thank my family for supporting me during this time, their support has enable me to achieve the goal of this thesis project.

TABLE OF CONTENTS

LIST OF SYMBOLS	iv
ACRONYMS	v
LIST OF TABLES.....	
LIST OF FIGURES.....	
ABSTRACT.....	
CHAPTER I: INTRODUCTION.....	1
1.1 Motivation.....	1
1.2 Objectives.....	2
CHAPTER II: DYNAMIC MODELLING OF QUADROTOR SYSTEM.....	4
2.1 System description and basic concepts.....	4
2.2 Quadrotor system dynamic modelling.....	7
2.2.1 Quadrotor dynamic model.....	8
2.2.2 Propeller and motor dynamic models.....	12
CHAPTER III: SYSTEM COMPONENTS AND THEIR PERFORMANCE	16
3.1 System's dynamic component.....	16
3.2 System parameters.....	18
3.3 Electronic components and measurement processing.....	19
3.3.1 Electronic components.....	19
3.3.2 Measurement processing.....	21
CHAPTER IV: CONTROLLER DESIGN.....	28
4.1 PID Controller.....	28

4.2	Cascade Control.....	30
4.3	Quadrotor control approach.....	30
4.3.1	Linearized model of quadrotor.....	30
4.3.2	Controllability and Observability.....	33
4.3.3	Controller Design.....	33
CHAPTER V: SIMULATION AND EXPERIMENTAL RESUTLS.....		38
5.1	Simulation results.....	38
5.2	Experimental results.....	43
CHAPTER VI: CONCLUSION AND RECOMMENDATION.....		48
APPENDIX A.....		50
APPENDIX B.....		52
APPENDIX C.....		54
C.1	Flight Controller Software.....	54
C.2	User Interface Guideline.....	55
REFERENCES.....		56

LIST OF SYMBOLS

s : $\sin()$

c : $\cos()$

φ : Roll angle

θ : Pitch angle

ψ : Yaw angle

m : Mass of quadrotor system

g : Acceleration due to gravitational force

C_x, C_y, C_z : Translational drag coefficients

A_x, A_y, A_z : Propeller disk area

I_{xx}, I_{yy}, I_{zz} : Inertia moment around x,y and z axis

F : Force

τ : Torque

k_f : Aerodynamic thrust coefficient

Oxyz: Global Coordinate frame

Ix'y'z': Body Coordinate frame

T_s : Sampling Time

ACRONYMS

PWM: Pulse Width Modulation

ESC: Electronic Speed Controller

IMU: Inertia Measurement Unit

PID: Proportional, Integral, Derivative

MIMO: Multiple Input- Multiple Output

SISO: Single Input- Single Output

SLAM: Simultaneously Localization and Mapping

LIST OF TABLES

Table 2.1 Quadrotor basic motion principle.....	6
Table 2.2 Motor and propeller data.....	12
Table 2.3 Cobra 2204/28 Motor Test Data.....	13
Table 3.1 Components and related data.....	17
Table 3.2 System's parameters.....	19
Table 3.3 Electronics parts.....	20
Table 5.1 PID Controller parameters.....	39

LIST OF FIGURES

Figure 1.1 Quadrotor from ETH Zurich Research Lab.....	2
Figure 2.1 Two configuration of the quadrotor.....	4
Figure 2.2 Two types of propeller.....	5
Figure 2.3 Coordinate system of the quadrotor.....	8
Figure 2.4 Thrust and angular speed fitting plot.....	14
Figure 2.5 Pulse Width Modulation for controlling brushless motor through ESC.....	14
Figure 3.1. The implemented quadrotor.....	17
Figure 3.2 Random pulse occurs during measurement at 21cm	22
Figure 3.3. Distance measurement after processing.....	23
Figure 3.4 Summary of optical flow.....	24
Figure 3.5 Illustration of calculating moved distance using optical flow.....	25
Figure 3.6 Raw measurement of flow and rate in X axis during rotation movement.....	26
Figure 3.7 Flow and Rate measurement being filtered during rotation movement.....	27
Figure 4.1 Block diagram of a process with a feedback controller.....	28
Figure 4.2 Cascade Control Structure.....	30
Figure 4.3 Altitude (z) control structures.....	35
Figure 4.4 Control Structure for rotation dynamics of quadrotors (system Q).....	36
Figure 4.5 Position x and y control architecture.....	37
Figure 5.1 Overall structure of Simulink files.....	38
Figure 5.2 Response of x position.....	40
Figure 5.3 Response of y position.....	40

Figure 5.4 Response of z position (altitude).....	41
Figure 5.5 Response of yaw angle (heading).....	41
Figure 5.6 Roll response of the quadrotor.....	42
Figure 5.7 Pitch response of the quadrotor.....	42
Figure 5.8 Altitude response between practical and simulation.....	44
Figure 5.9 Response of x position of implemented system.....	44
Figure 5.10 Response of y position of implemented system.....	45
Figure 5.11 Response of yaw angle of implemented system.....	45
Figure 5.12 Quadrotor vertical takeoff.....	46
Figure A.1 Approximate model for calculating moment of inertia of the quadrotor	50
Figure B.1 Three angles rotation roll, pitch and yaw.....	52
Figure B.2 Rotation in order z-x-y	53
Figure C.1 Flow chart of the quadrotor system.....	54
Figure C.2 User Interface in Matlab	55

ABSTRACT

Throughout many changes in technologies, the proportional-integral- derivative (PID) controller is still widely used as most common structure in practical control problems. The reason is that the PID controller has a simple structure, which can be easily implemented and tuned. However, this is only accurate for single input-single output (SISO) systems. In multiple input-multiple output system, the inputs and control variables are coupled, therefore, the controller parameters cannot be simply tuned as before. For controlling these kind of systems, the state PID controller with its pole placement method is introduced.

One of the common MIMO system nowadays is the quadrotor, which has 4 motors inputs with 12 control variables, this makes the system highly under-actuated and thus difficult to control. The objective of this thesis is to introduce the method of state PID in controlling this MIMO system through simulation, then the designed PID controller would be tentatively implemented in the real quadrotor system.

CHAPTER I

INTRODUCTION

1.1 Motivation

Flying vehicles have been an interesting topics for the field of control theory for many years. Due to the complexity presenting through the nonlinearity of the dynamics state equations of these systems, the task of designing an effective controller for them is challenging. In this thesis, the control theory will be applied for controlling one of such aerial vehicle: the quadrotor system.

The quadrotor is an aerial vehicle which contains four motors diagonally placed in one plane as actuators to control its movement. Its history dated back in 20th century in a race of inventing the man's first aerial vehicle. At that time, however, the design does not have enough supported technology and theory for controlling four motors to stabilize the system; therefore, the quadrotor soon fall back the race and the fixed-wing aircraft became the first aerial vehicle of mankind [1].

Decades later, the evolution of IC technology and the introduction of modern control theory made it available to control the quadrotor system. With the advantages of high maneuverability and flexibility compares to the fixed-wing aircraft design, the quadrotor has become popular recently with various application, ranging from photography, navigation to commercial flying vehicle.



Figure 1.1 Quadrotor from ETH Zurich Research Lab [2]

Although the control field has created many useful methods for controlling typical nonlinear systems, the PID controller and its variant are still the most widely used one in industry nowadays. In fact, many complex systems in industry are still controlled by PID method. Following the senior project, which applies the PID controller to the inverted pendulum system, the thesis goal is to design a PID controller for stabilize the quadrotor system in its hovering condition as a basis for further research topics later.

1.2 Objectives

Following the senior project, whose objectives is to regulate the inverted pendulum system by the multi-dimensional PID method, the thesis will apply the idea and methodology obtained from the senior project in the case of the quadrotor system, for both theoretical and experimental process.

The objectives of this thesis is presented as follows:

- Overview about the dynamic model of the quadrotor system.
- Design a PID controller structure for regulating the quadrotor attitudes.
- Using Simulink for support the simulation, designing and testing process.

- Use the obtained regulator for the real system and compare the responses of from simulation and practical implementation.
- Discuss the obtained result and provide several research directions related to the system.

CHAPTER II

DYNAMIC MODELLING OF QUADROTOR SYSTEM

This chapter will first present the description and basic concepts of quadrotor that will be used in this thesis project. After that, a mathematical model of the quadrotor system as well as related components will be discussed.

2.1 System description and basic concepts

A quadrotor is a type of aerial vehicle whose movements is determined by the forces and torques combination from four rotors. These four motors are typically located symmetrically through the center of the quadrotor on the XY plane. Based on the position of the inertial measurement unit (IMU) on the flight controller board or the way defining the body frame of the quadrotor on the coordinate system, the quadrotor will be determined as the “X configuration” of the “Plus configuration” as shown in figure 2.1.

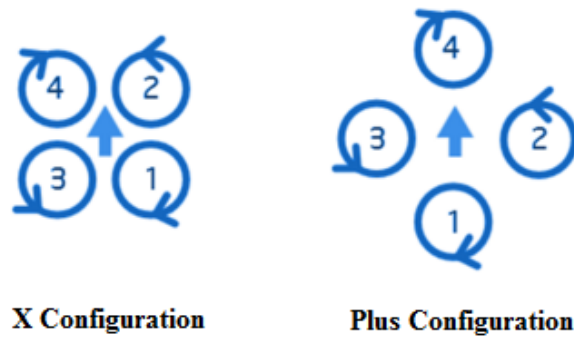


Figure 2.1 Two configuration of the quadrotor

(The arrow is the orientation of the IMU)

The rotating directions of four rotors is also shown in the above figure as clockwise and counter-clockwise. Each rotor direction of rotation must be opposite of the two neighbors in order

to maintain the balance of torques on the system. If such rotating direction is not followed, the quadrotor will rotate around its z-axis.

Attached to each motor is the propeller, which is in charge of generating the thrust and torque required for the quadrotor to move. While the torque responsible for the rotation around the vertical axis, thrust is a reaction force responsible for exerting force on the quadrotor. With four fixed-pitch propellers, the quadrotor has total four control inputs with six outputs and twelve control variables. Like the inverted pendulum system in the senior project, this system is under-actuated.



Figure 2.2 Two types of propeller

(Clockwise-type (right) and Counterclockwise-type (left) propeller)

Due to the opposition of rotating direction in the system, the propeller attached to each motor also have different configuration as clockwise and counter clockwise to maintain the upward thrust direction. The magnitude of the thrust and torque will be varied accordingly to the rotation speed of the motor which can be described by the corresponding aerodynamics function, which will be discussed in later section of this chapter.

The principle of motion depends on the motor rotation. In short, the basic flying principle is presented in Table 2.1.










QUADROTOR BASIC MOTION PRINCIPLE	
 Slow-Fast	
	
Rotate left through vertical axes	Rotate right through vertical axes
	
Move to right	Move to left
	
Move forward	Move backward
	
Descending	Ascending

Table 2.1 Quadrotor basic motion principle

2.2 Quadrotor system dynamic modelling

Although in the previous section, the concept of the quadrotor motion is simple, however, there are many factors such as noise, disturbance... in the system that cause the quadrotor to lose the stability during its flight. In order to maintain the stability of the quadrotor, it is important to have an accurate dynamic model which represent the physical factors that affects our system. Only after such model is obtained, the stability of the system can be achieved after being implemented a proper controller.

In this thesis, we consider the following assumptions:

- Quadrotor structure is a rigid body
- Hub forces (horizontal force on the blade)and moments are ignored
- Ground effect, which is the effect of external disturbance force occurred when the quadrotor is at low-level altitude (about size of its arm length).
- Propellers are rigid.
- Air friction will be neglected.

For the full modelling of quadrotor, [3] has introduced a full dynamics model of quadrotor, taking into account those assumptions above. In [3], these assumptions are shown to be acceptable under hovering condition with the listed motions in Table 2.1, except for the ground effect when the quadrotor is at low altitude.

2.2.1 Quadrotor dynamic model

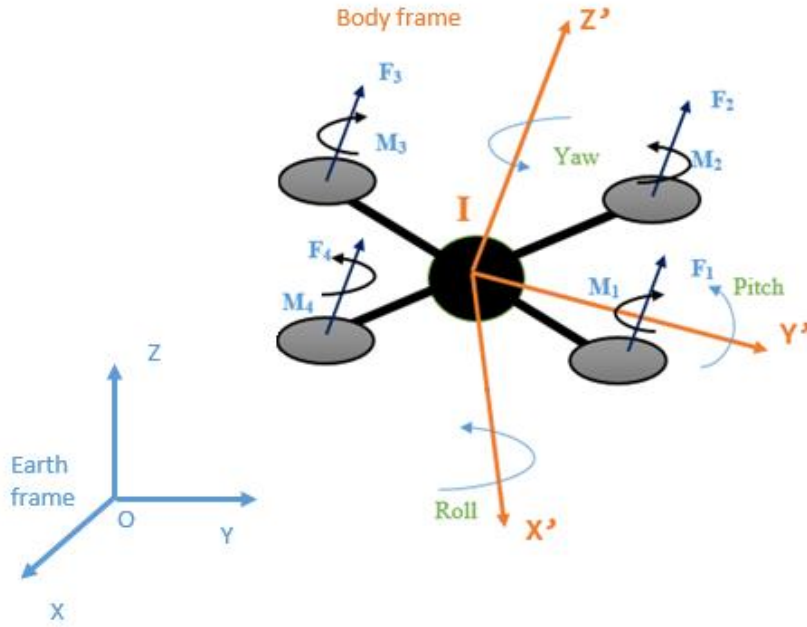


Figure 2.3 Coordinate system of the quadrotor

Figure 2.3 illustrates the coordinates involved in quadrotor dynamic model. The fixed-frame (ground frame) is $Oxyz$ and the body frame is $Ix'y'z'$. In the body frame, the forces and moments caused by rotors are also shown in the figure.

To transform the body frame to world frame, following the Appendix B, the following rotation matrix is used:

$$R_{zyx} = \begin{bmatrix} c\psi c\theta - s\phi s\psi s\theta & -c\phi s\psi & c\psi s\theta + c\theta s\phi s\psi \\ c\theta s\psi + c\psi s\phi s\theta & c\phi c\psi & s\psi s\theta - c\psi c\theta s\phi \\ -c\phi s\theta & s\phi & c\phi c\theta \end{bmatrix}$$

This matrix is used to describe the rotation motion of the quadrotor in the coordinate system.

The inputs of the system are the force and moment caused by the rotors. Consider the body frame, we have the following inputs:

$$U_1 = F_1 + F_2 + F_3 + F_4 \quad (\text{Total force})$$

$$U_2 = l(F_1 + F_2 - F_3 - F_4) \quad (\text{Rolling moment input})$$

$$U_3 = l(F_1 + F_4 - F_2 - F_3) \quad (\text{Pitching moment input})$$

$$U_4 = M_1 - M_2 + M_3 - M_4 \quad (\text{Yawing moment input})$$

Dynamical meaning, U_1 is the force which will push the quadrotor along the z axis; U_2 is the moment which will result the rotation around x axis of the quadrotor; U_3 is the moment result the rotation around y axis and U_4 is the moment in charge of rotation around z axis.

Using Newton's law, the quadrotor motion equation can be expressed as:

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = -mg \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + R_{zxy} U_1 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + D_{3 \times 1} + H_{3 \times 1}$$

Where $D_{3 \times 1} = \begin{bmatrix} -0.5C_x A_x \rho \dot{x} |\dot{x}| \\ -0.5C_y A_y \rho \dot{y} |\dot{y}| \\ -0.5C_z A_z \rho \dot{z} |\dot{z}| \end{bmatrix}$ is air drag while $H_{3 \times 1}$ is hub forces, which are neglected

in this thesis. This leads to the following equations:

$$m\ddot{x} = (c\psi s\theta + c\theta s\varphi s\psi)U_1$$

$$m\ddot{y} = (s\psi s\theta - c\psi c\theta s\varphi)U_1 \quad [2.1]$$

$$m\ddot{z} = (c\varphi c\theta)U_1 - mg$$

Which describes the motion of the quadrotor in the world frame based on its rotation angle and input thrust caused by four rotors. For the quadrotor system, it is also required to derive the rotation movement in the body frame.

For the body frame of the quadrotor, Euler's equation states that:

$$I\dot{\zeta} + \zeta \times (I\zeta) = \tau$$

Where $\zeta = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$ and $I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}$. Neglect moments that we discussed before, then:

$$\ddot{\phi} = \frac{I_{zz}-I_{yy}}{I_{xx}} \dot{\theta}\dot{\psi} + \frac{U_2}{I_{xx}}$$

$$\ddot{\theta} = \frac{I_{xx}-I_{zz}}{I_{yy}} \dot{\phi}\dot{\psi} + \frac{U_3}{I_{yy}} \quad [2.2]$$

$$\ddot{\psi} = \frac{I_{yy}-I_{xx}}{I_{zz}} \dot{\theta}\dot{\phi} + \frac{U_4}{I_{zz}}$$

These three equations describe the rotational dynamics of the quadrotor. Together with the equations of [2.1], these equations represent the relation between input and output of the system.

We want to bring the quadrotor to the steady state such that the position of the system is constant, which means:

$$x = \text{const}$$

$$y = \text{const}$$

$$z = \text{const}$$

$$\psi = \text{const}$$

And other state variables are 0.

From the dynamics model, the intuition control strategy can be formed. As the dynamics equation implies, translational movement as well as the position of the quadrotor depends on the rotation of the quadrotor to the world frame. This will give us an intuition of regulating the three angles (roll, pitch and yaw) of the quadrotor before controlling the position of the quadrotor.

Assumed that the three rotation above have been fully controlled, then for the stabilizing the z axis of the quadrotor, it is simply a second order linear differential equation. For the x and y axis, in the hovering condition, the roll and pitch rotation can be assumed small enough so that:

$$\varphi \approx s\varphi \quad ; \quad \theta \approx s\theta$$

For the U_1 input: $U_1 = mg$, this will reduced our x and y related equation as:

$$\ddot{x} = (\theta c\psi + \varphi s\psi)g$$

$$\ddot{y} = (\theta s\psi - \varphi c\psi)g$$

If we are able to find the PID feedback control law, for now denote as: $pid(x, x_{sp})$ and $pid(y, y_{sp})$ such that:

$$\ddot{x} = pid(x, x_{sp}), \ddot{y} = pid(y, y_{sp})$$

These laws will make $x \rightarrow x_{sp}$ and $y \rightarrow y_{sp}$, then we can obtain the required value of θ and φ by:

$$\theta = g \times (pid(x, x_{sp})c\psi + pid(y, y_{sp})s\psi)$$

$$\varphi = g \times (pid(x, x_{sp})s\psi - pid(y, y_{sp})c\psi)$$

2.2.2 Propeller and motor dynamic models

For each rotor, the vertical force F is related to the angular speed by the equation:

$$F_i = k_F \omega_i^2 \quad [2.3]$$

And the moment is defined by: $M_i = k_M \omega^2$

To be more exact, it is required to do the identification task for determining the transfer function of these relation. However, this thesis will neglect those effect and concentrate only on the steady state relations. Table 2.2 shows the data of the propeller and motor used in this project.



Items	Specification	
Brushless Motor: EMAX 1806- 2280KV		KV: 2280 Max Thrust: 460g Number of cell: 2-3S Propeller: 5" ~ 6" Length: 26.7mm Shaft: 2mm Diameter: 23mm Weight: 18g
Propeller: GEMFAN 5x3		Diameters: 5 inches Distance move in one revolution: 3 inches.

Table 2.2. Motor and propeller data

For having a simulation for the controller design, it is required for us to know the parameter of k_F and k_M . This requires measuring forces, angular speed of the motor as well as setting up tools for the measurement. However, there is available data of the motor Cobra 2204, $K_v = 2300$ with Gemfan 5x3 propeller at [4] which can be referred for our set up. Table 2.3 provides the data:

Data Collected at 11.1 volts with GemFan 5x3 Prop						
Throttle	Motor	Input	Prop	Thrust	Thrust	Efficiency
Setting	Amps	Watts	RPM	(Grams)	(Ounces)	Grams/W
10%	0.32	3.55	5,435	26.7	0.94	7.52
20%	0.79	8.74	8,409	66.1	2.33	7.57
30%	1.35	14.93	10,519	108.5	3.82	7.27
40%	1.93	21.46	12,178	142.1	5.01	6.62
50%	2.53	28.12	13,604	175.9	6.20	6.26
60%	3.12	34.65	14,821	206.7	7.28	5.96
70%	3.83	42.50	15,942	247.9	8.74	5.83
80%	4.42	49.04	16,908	274.3	9.67	5.59
90%	5.89	65.40	18,838	342.0	12.05	5.23
100%	8.00	88.80	21,190	431.7	15.21	4.86

Table 2.3 Cobra 2204/28 Motor Test Data

With data from the Table 2.3, we are able to determine the constant k_F in equation [2.3]:

$$k_F = 9.3928 \text{ (kg.m)} \text{ and } F = 9.3928 \times 10^{-9} \omega^2 \text{ [2.4]}$$

For the full motor modelling, it would require much more work to identify all parameters of the model. As I did not have enough time, I just consider the steady state given by [2.3].

Figure 2.4 shows the fitting plot of the function above.

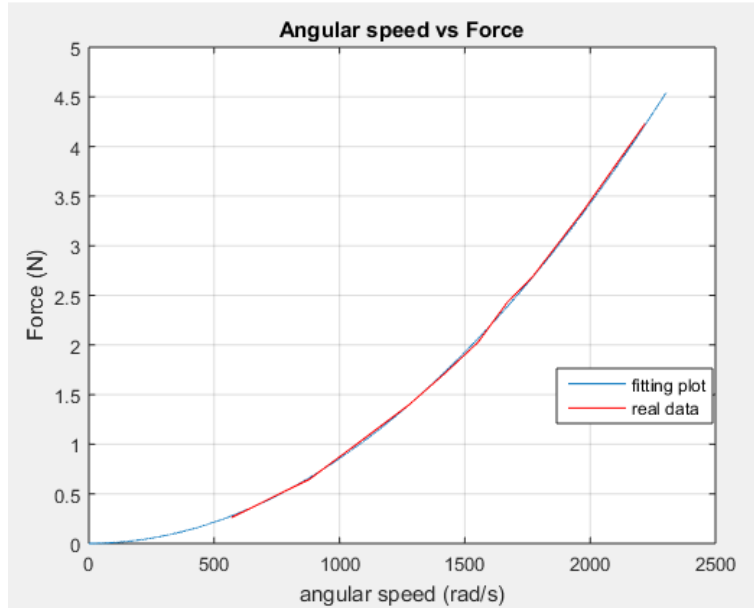


Figure 2.4. Thrust and angular speed fitting plot.

For controlling the speed of the motor, we will send the PWM signals to the ESC, which will generate corresponding three phase electric signal to the brushless motor. The pulse has period of 20ms with its width varies from 1ms to 2ms, which is shown in figure 2.5. Sending signal with 1ms width means that the motor will stop rotating.

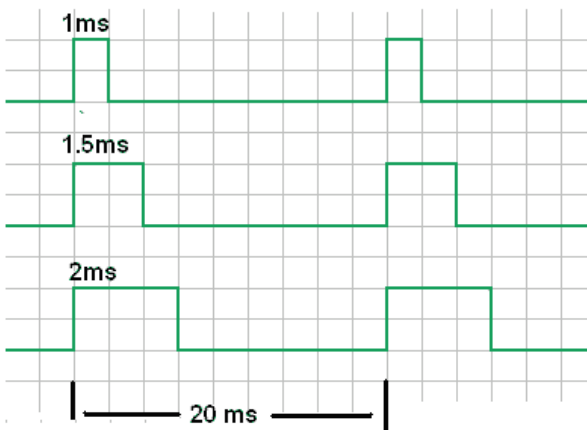


Figure 2.5. Pulse Width Modulation for controlling brushless motor through ESC.

It is also important to find the relationship between the throttle setting with the angular speed for controlling the motor, since this related to the pulse width required for controlling the brushless motor. In the table, the relation of throttle command with the pulse width is:

$$Width = 1000 + 10x \text{ (}\mu\text{s)} \text{ where } x\% \text{ is throttle command.}$$

We denotes s as sending signal with range from 0 to 1000 where:

$$s = Width - 1000$$

Using the Table 2.3, the relationship between s and ω is:

$$\omega = s \times 1.6417 + 536.4828 \text{ [2.4]}$$

Since the table do not show the measuring moments, we will control the yawing moment of the quadcopter based on PID tuning techniques. The equations [2.1] [2.2] [2.3] [2.4] are keys for building our simulation.

CHAPTER III

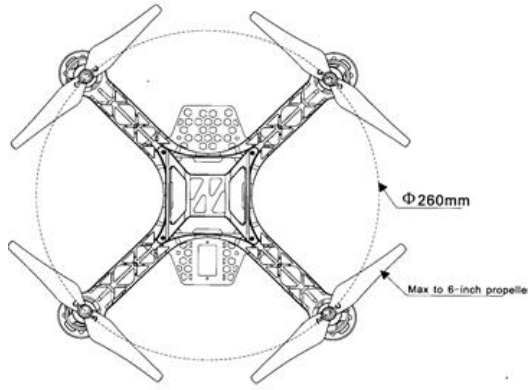
SYSTEM COMPONENTS AND THEIR PERFORMANCE

In order to design controller based on Simulink, it is essential to have parameters of the real system. This chapter will represent the principle of hardware design followed by the system's parameters.

3.1 System's dynamic components

Previously in chapter II, the details of the propeller and the rotor have been introduced in details. The process of choosing the motors and propeller is decided after the quadrotor's size and weight have been decided. Generally, for the quadrotor to be able to hovering as well as climbing, the maximum thrust produced by four rotors should be about twice the weight of the quadrotor. Since weight of the quadrotor matters, it is important to consider the mass of other components such as: motor, battery and electronic components.

With that concept, table 3.1 shows the data of chosen component and figure 3.1 shows the implemented quadrotor.

Components name	Figure	Parameters
Quadrotor frame FPV260		Arm length: 130mm Weight: 0.12kg


Battery NanoTech 3S-11.1V		Weight: 0.08kg
Motors and others electronic components	Figure of each electronic components will be shown in section 3.3	Weight: 0.25kg

Table 3.1 Components and related data

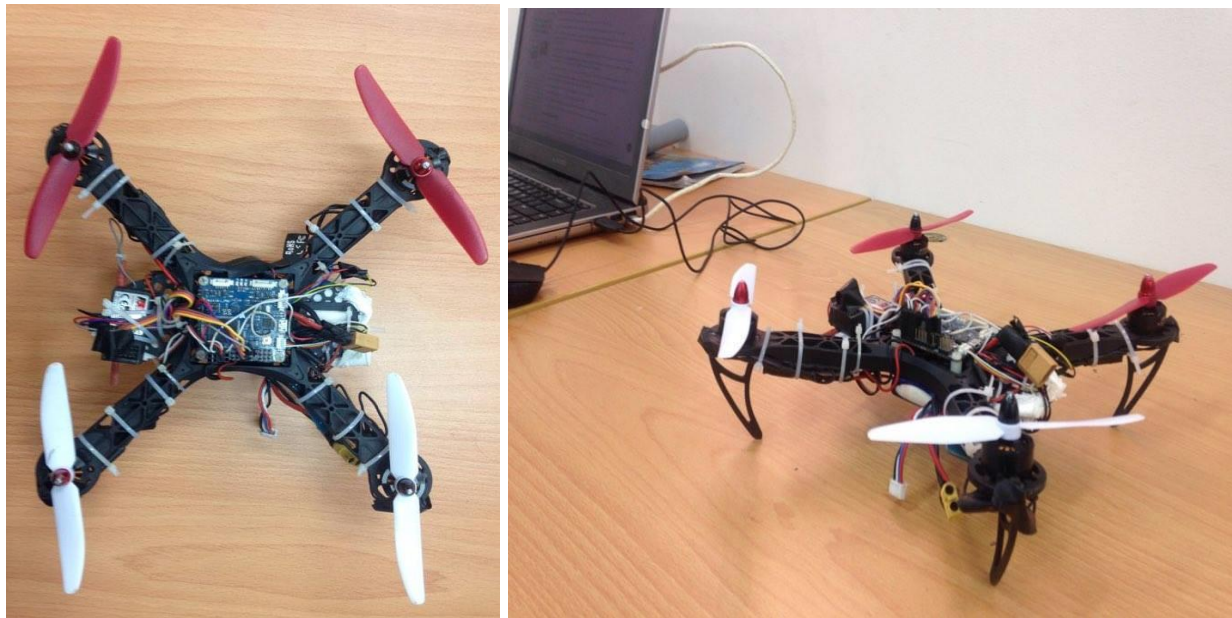


Figure 3.1 The implemented quadrotor

3.2 System parameters

With the concepts represented and data provided in previous section, it is required to know other parameters of the quadrotor, includes inertia matrix and effective length of the system. Since there is limited of mechanical measurement devices available, the inertia matrix is estimated based on rule of basics objects.

The effective length of the quadrotor can be calculated by:

$$l = \frac{\text{arm length}}{\sqrt{2}} = 9.19 \text{ cm} = 0.0919 \text{ m}$$

For the inertia matrix, based on the Appendix A, we can calculate the moments of inertia around each axis as:

$$I_{xx} = 0.00091 \text{ kgm}^2$$

$$I_{yy} = 0.00091 \text{ kgm}^2$$

$$I_{zz} = 0.0017 \text{ kgm}^2$$

$$\text{Or } I = \begin{bmatrix} 0.00091 & 0 & 0 \\ 0 & 0.00091 & 0 \\ 0 & 0 & 0.0017 \end{bmatrix}$$

Also, the mass of the quadrotor is equal to sum of each components, which is measured as:

$$m = 0.45 \text{ kg}$$

Table 3.2 summarizes the parameters of the design.


Parameters	Value
Mass (m)	0.45 kg
Moment of Inertia (x axis)	0.00091 kgm ²
Moment of Inertia (y axis)	0.00091 kgm ²
Moment of Inertia (z axis)	0.0017 kgm ²
Length	0.0919 m

Table 3.2 System's parameters

3.3 Electronic components and measurement processing

In order to control the quadrotor using feedback control, it is essential to be able to obtain the measurement of quadrotor states. This section will first introduce the microcontroller and electronic sensor used in the implemented system with their functionality. Then, the discrete section of the system such as sampling time, filtering will be focused.

3.1.1 Electronic components

Components	Characteristics
Microcontroller Atmega 2560 	<ul style="list-style-type: none"> -High Performance, low power Atmel AVR 8 bit MCU. -256KB Flash memory -USART, I2C and SPI communication channel. -Total 16 PWM channels, which is used for controlling the motor speed.

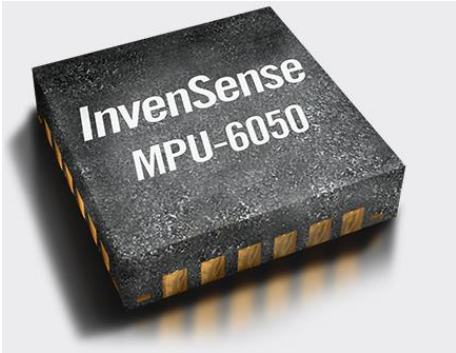


<p>IMU</p> <p>InvenSense MPU6050</p> 	<ul style="list-style-type: none"> -Integrated 6-axis Motion Tracking device which has 3 axis gyroscope and 3 axis accelerometer. -Communication channel: I2C -Support DMP for motion fusion on chip, reducing the workload of microcontroller with high accuracy. -Sampling Time: 5ms
<p>Ultrasonic Rangefinder SRF05</p> 	<ul style="list-style-type: none"> -Measure distance using ultrasonic wave reflection. -Max Range: 3m -Min Range: 3cm -Sampling Time: 50ms
<p>Optical Flow Sensor PX4flow</p> 	<ul style="list-style-type: none"> -Measure distance moved between 2 frames in pixels based on optical flow principle. -Works well in natural light. - 752×480 MT9V034 image sensor, L3GD20 3D Gyro - 16 mm M12 lens -Sampling Time: 5ms

Table 3.3 Electronics part

3.1.2 Measurement processing

a. Inertial Measurement Unit MPU6050

The MPU6050 sensor is in charge of measuring the rotation motion around 3 axis of the quadrotor, providing raw data of angular rate and accelerations in default. The most well-known issue of the gyro is the drift and that of the accelerometer is noise sensitivity. Often, in order to obtain the exact rotation motion, sensor fusion algorithm such as Kalman filter or Complementary filter is implemented on the microcontroller after reading raw data. However, the MPU6050 supports the built-in DMP, which enables the sensor to process data and output with 200Hz at the I2C channel.

b. Ultrasonic Rangefinder SRF05

In the quadrotor, the SRF05 sensor is responsible for measuring the distance between the quadrotor and ground based on ultrasonic wave principle.

The sensor only returns reliable value when facing the solid surface, this means that the sensor does not work well when facing soft surface.

The known issue of this sensor is an abnormal high amplitude pulse, which occurs randomly during measuring period. These pulses have its value randomly higher or lower than the expected reading of the sensor, shown in figure 3.2. This issue, although it does not occur frequently, still affects the performance of the quadrotor system and therefore, requires an algorithm to identify and eliminate them.

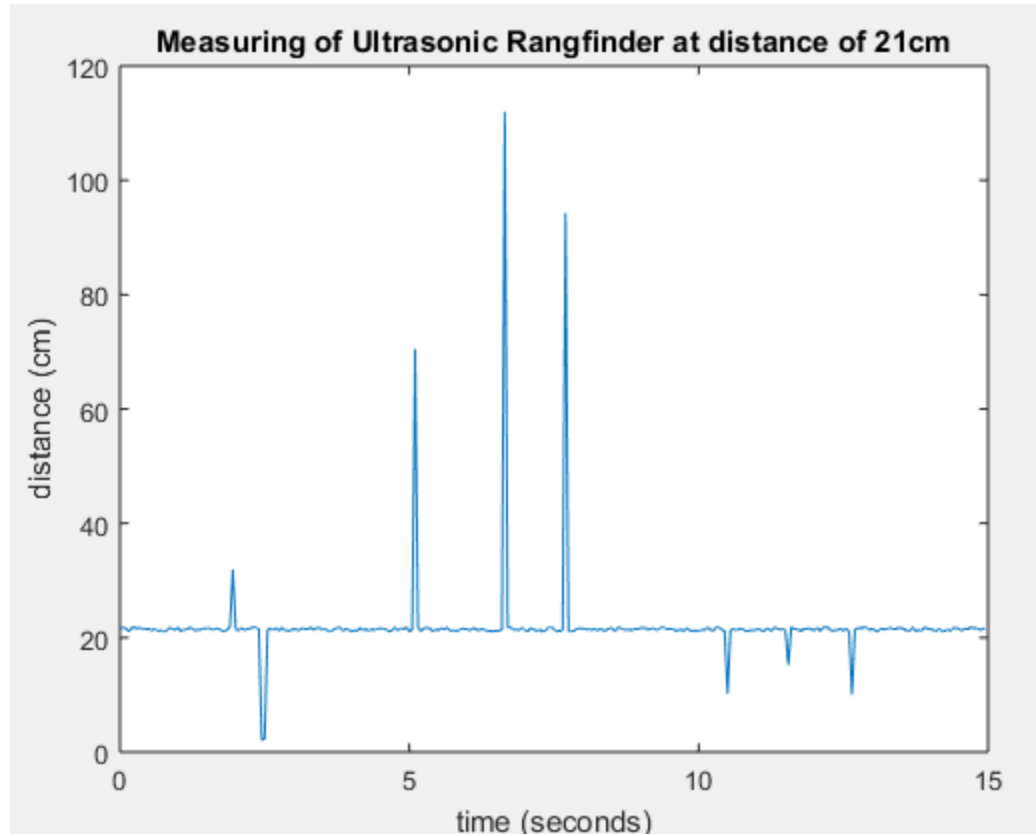


Figure 3.2 Random pulse occurs during measurement at 21cm

To overcome this issue, a peak detection algorithm is implemented in the software as follows:

Algorithm 1. Processing peak occurs in SRF05 reading

currentZ, previousZ

PeakProcessing:

If(abs(currentZ - previousZ) > 15):

currentZ = previousZ //ignore the mearsurement,

previousZ = currentZ

After applying the algorithm, the data is shown in figure 3.3. The reason why the 15cm is chosen as threshold is based on experimental climbing rate of the quadrotor when altitude (z axis) controller is implemented after. However, this algorithm is not perfect and therefore, some low amplitude pulses still remain after being processed.

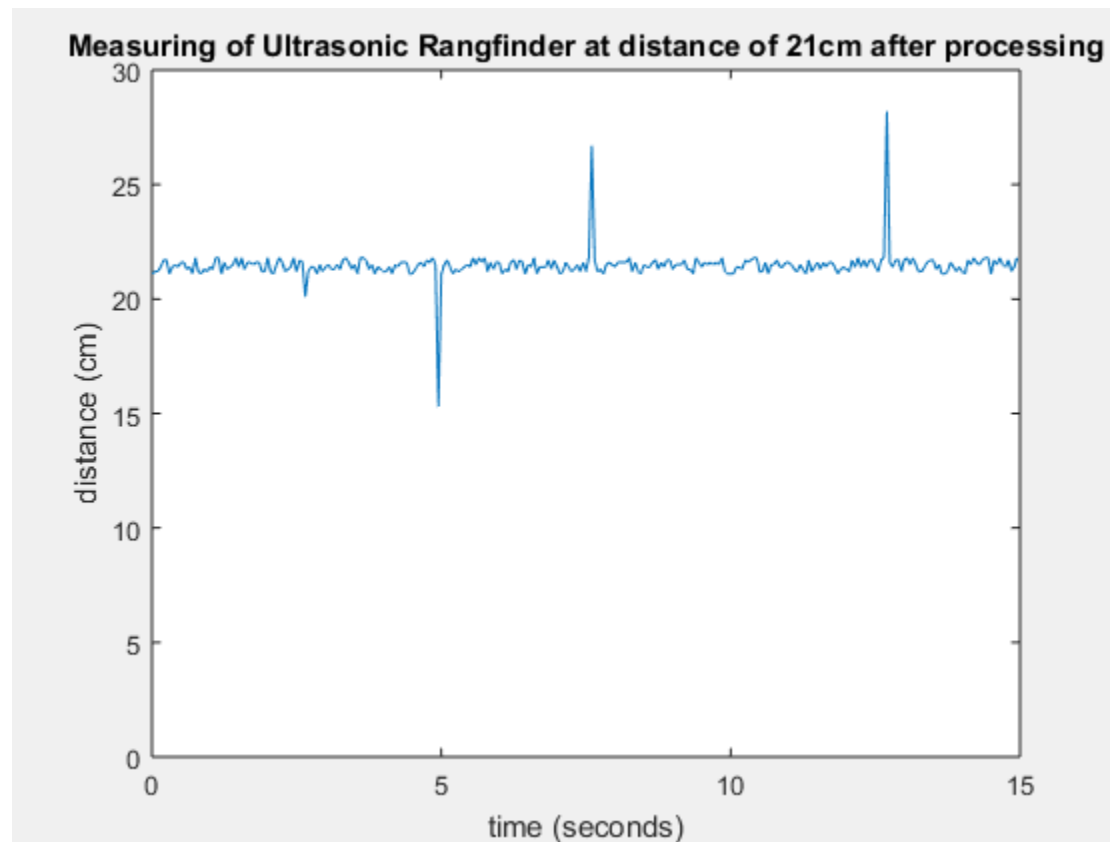


Figure 3.3. Distance measurement after processing

c. Optical Flow sensor PX4flow

There are many methods for measuring the XY movements of the quadrotor. In many drone research labs, the motion capture system is often employed due to its high accuracy. This kind of system use computer vision and estimation techniques to track the movement of the quadrotors. However, one significant drawback of this system is that the quadrotor can only operate inside the limited space where the system covers. In practice, to design an autonomous flying robot, the

camera with computer vision algorithm is often implemented in the system to locate and estimate the position of the quadrotor. In this thesis, one technique that involves the camera is optical flow sensor, which together with the ultrasonic rangefinder sensor, will estimate the position of the quadrotor system.

The optical flow is a motion of brightness patterns in the sequence of images. It measures how the image moved between consecutive frames in pixels with an assumption of image brightness constancy; which means that this principle will not work well under fluorescent light. The PX4flow has provided a built-in processing algorithm, which returns the flow movement of the camera.

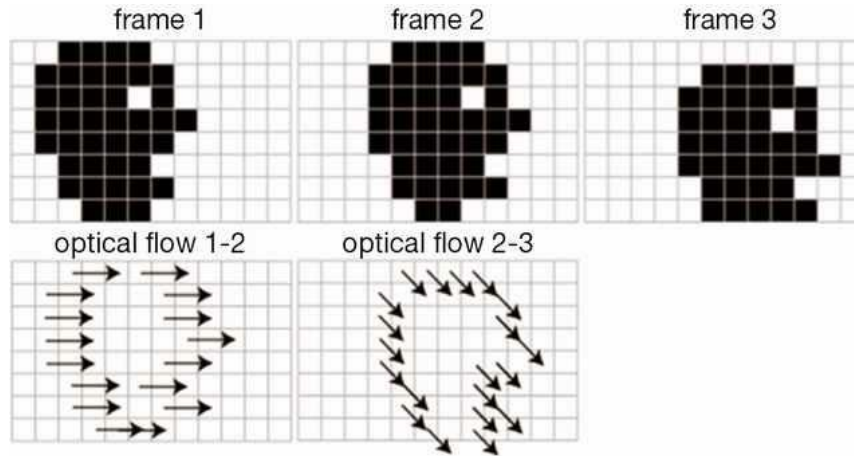


Figure 3.4 Summary of optical flow [5]

With the know field of view angle of the camera, the measurement in pixels can be converted into the corresponding measurements in radian. Then, for translational flow measurement γ with distance d to the surface, the moved distance Δ can be calculated as follows:

$$\Delta = 2 \times d \times \tan\left(\frac{\alpha}{2}\right) \approx \alpha \times d$$

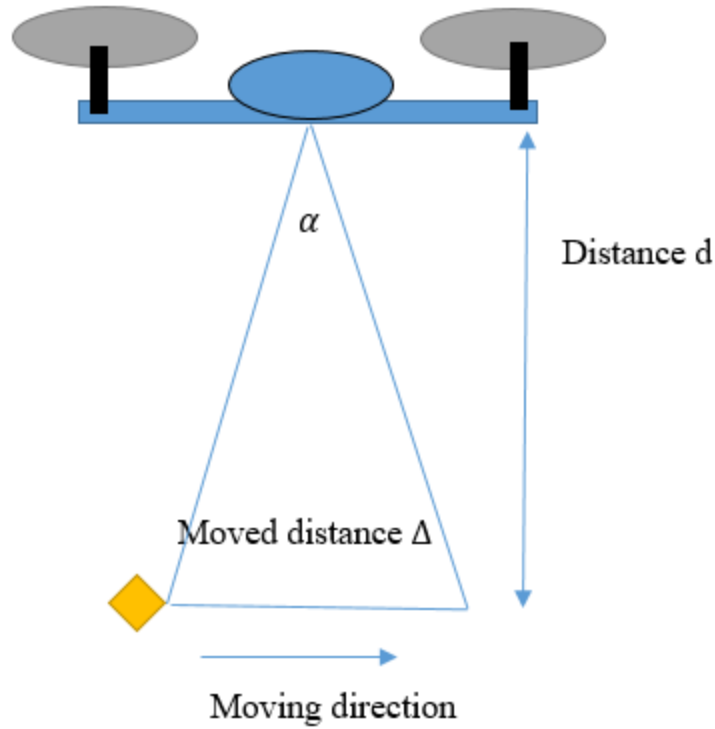


Figure 3.5 Illustration of calculating moved distance using optical flow

The sensor, however, will return the flow measurement of both translational and rotational movement; therefore, in practice, it is required for compensating the real flow movement with the rotation movement around x and y axis since these rotations also cause the flow measurement to change. Let β be the rotated angle, α be the flow measurement and γ be the real flow measurement due to translation, we have:

$$\gamma = \alpha - \beta$$

Figure 3.6 shows the value of flow and angle measurement during rotation around x axis. Ideally, the value of these two measurement should be equal. To obtain better result, a moving average filter is used, which is:

$$s[n] = 0.1 \times \sum_{i=0}^9 w[n-i]$$

Where $s[n]$ is the output and $w[n]$ is data measured. Figure 3.7 shows the plot of flow and angle during rotation movement with filter applied, which provides a better result than figure 3.6.

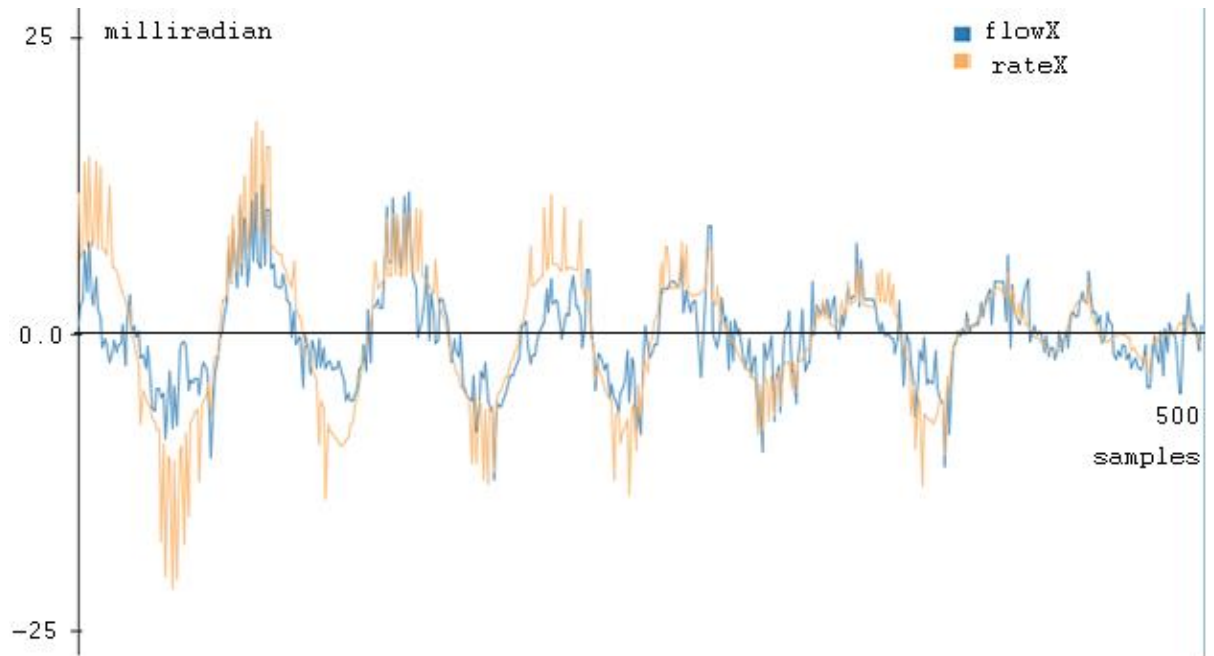


Figure 3.6 Raw measurement of flow and rate in X axis during rotation movement

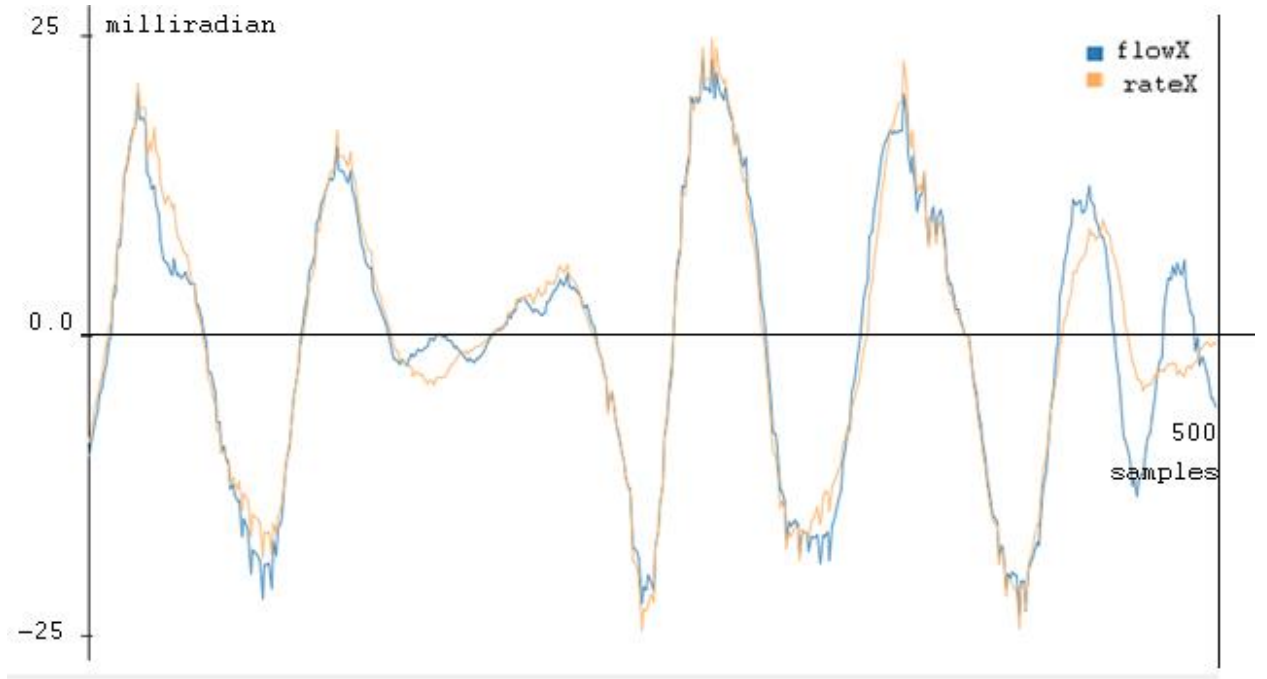


Figure 3.7 Flow and Rate measurement being filtered during rotation movement

After obtained the moved distance between 2 consecutives frames, the velocity v and distance D can be estimated as:

$$v = \frac{\Delta}{T_s}$$

$$D = \sum_{n=0}^{N=\frac{t}{T_s}} \Delta[n]$$

With T_s is a sampling time of the optical flow sensor and is 5ms in our system.

CHAPTER IV

CONTROLLER DESIGN

We are interested in implementing the PID control for the quadrotor system. This chapter prepare the general form of the PID controller and design method, which will be latter used for implement in the Simulink diagram.

4.1 PID Controller

The control systems are generally represented in figure 4.1. In this representation, the desired output is denoted by $r(t)$. The output of the plant is denoted by $y(t)$. The difference between the desired output $r(t)$ and the output of the plant $y(t)$ is the error, which is denoted by $e(t)$. The error signal $e(t)$ is the input of the controller, which base on that to generate the control signal $u(t)$ to the plant. In this representation, disturbance is denoted by $d(t)$.

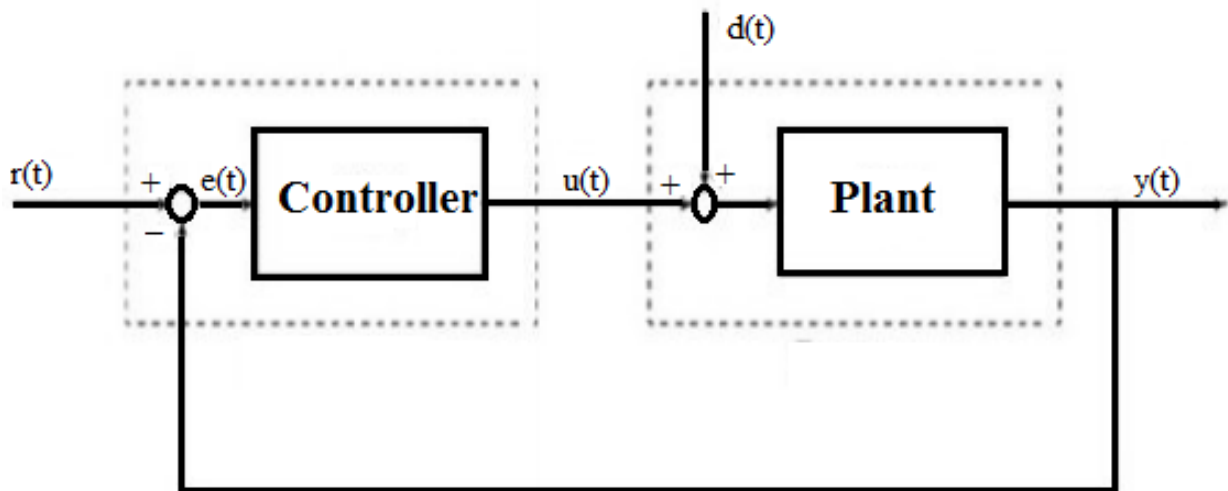


Figure 4.1 Block diagram of a process with a feedback controller

The PID controller is a feedback controller which consists of three control actions: proportional, integral and derivative and is expressed mathematically as:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} [6]$$

For simplicity in notation, let:

$$pid(w, w_{sp}) = u(t) \text{ and } e(t) = w(t) - w_{sp}$$

This notation will be used in latter design process. In terms of discrete control, [7] shows the form as:

$$u[n] = K_p e[n] + K_i T_s \sum_{i=0}^{n-1} e[i] + \frac{K_d (e[n] - e[n-1])}{T_s}$$

The discrete form will be used to control the quadrotor and will also be implemented in the simulation for consistency, therefore the sampling time is essential for be pre-determined before starting the design project.

The PID controller is a linear feedback controller; therefore it only works well with linear system. For the nonlinear system, the PID controller can only performs well under certain region around linearized point of the system.

In the senior project, the PID controller has been used to stabilize the inverted pendulum system. The inverted pendulum is an under-actuated system, having only one input with two outputs that exists cross coupling between the states. In order to control such system, it is required to have a PID controller for every state variable, then combine them into the input. In this thesis, that methodology will also be followed to regulate the quadrotor system.

4.2 Cascade Control

One of the common control technique is the Cascade Control. This technique allow designer to utilize the feedback control for disturbance rejection. Consider a system in figure 4.2 with two controller C_1 and C_2

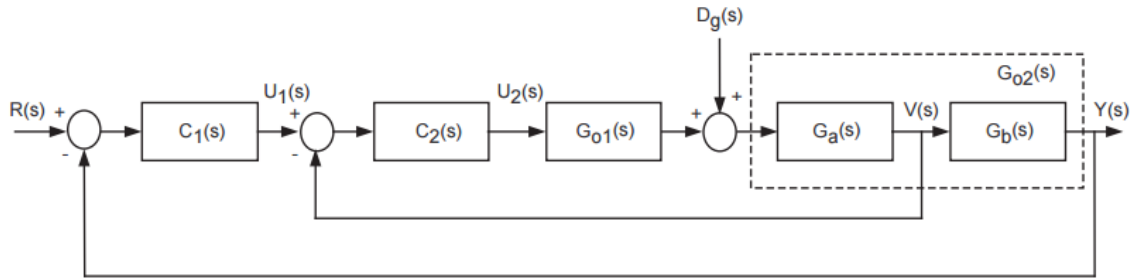


Figure 4.2 Cascade Control Structure.

In [8], it has shown that with the inner loop controller C_2 , the disturbance is attenuated when comparing with the control structure which there is only one control feedback loop.

Since the quadrotor system sustains many source of disturbance (wind, vibrations...), this control structures will enable us to regulate the system with higher robustness.

4.3 Quadrotor control approach

4.3.1 Linearized model of quadrotor

In this section, the dynamics of the quadrotor will be linearized in the hover condition. This linearized model will be used to prove that the system is controllable in hover condition and then be employed in the Simulink to design a regulator for the system.

In short, the dynamic of the quadrotor can be shortly represented as:

$$\dot{q} = f(q, u)$$

The steady state, in this representation is:

$$f(q_{ss}, u_{ss}) = 0$$

As shown in chapter 2, the quadrotor stable points is when the position and yaw angle is constant and roll, pitch angle is 0. Denote q_{ss}, u_{ss} as steady state and steady control action of the quadrotor, then we have the linearized model under hover condition as follow:

$$\dot{q} = A(q - q_{ss}) + B(u - u_{ss})$$

Where A and B are Jacobian matrices:

$$A = \frac{\partial f}{\partial x} \big|_{(x,u)=(x_{ss},u_{ss})}$$

$$B = \frac{\partial f}{\partial u} \big|_{(x,u)=(x_{ss},u_{ss})}$$

The value of u_{ss} after solved for at steady state condition is:

$$u_{ss} = \begin{bmatrix} mg \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Rewrite as shifted state vector $z = q - q_{ss}$ and shifted input $r = u - u_{ss}$ then:

$$\dot{z} = Az + Br$$

We then can obtained the value of matrices A and B as follows:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & g s \psi_{ss} & 0 & g c \psi_{ss} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -g c \psi_{ss} & 0 & g s \psi_{ss} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{I_{xx}} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix}$$

These two matrices represent the linearized dynamic equation of the quadrotor system around the hovering position $q_{ss} = [x_{ss} \ 0 \ y_{ss} \ 0 \ z_{ss} \ 0 \ 0 \ 0 \ 0 \ 0 \ \psi_{ss}]^T$. Before designing the controller for the system, we will first prove that the system is controllable.

4.3.2 Controllability and Observability

For the observability, in our implemented system, we are able to estimate each state of the quadrotor from our sensor, therefore it is guarantee that our system is observable, in other words, the output of the system is also its states. Mathematical speaking, the output of the system can be represented as $y = Cq$ where $C = I_{12 \times 12}$.

For the controllability, define:

$$CO = [B \ BA \ BA^2 \ \dots \ BA^{11}] \text{ as the controllability matrix of the system}$$

Then for the linearized system to be controllable, CO rank must be 12 which has been verified by checking with Matlab. Thus the quadrotor system is controllable under hovering condition.

4.3.3 Controller Design

As discussed in chapter 2 about the intuition about controlling the quadrotor system, this section will focus the mathematical view for designing the PID controller for the system. In fact, from the state space control view, the quadrotor in hovering condition can be controlled using the state feedback method, which in this case actually a combination of PD controller. To design such controller, one notable method is LQR method, which is a state feedback optimization and is as discussed in [9]. These linear control, although simple in structure, will not work when the system goes out of the linearized region. Other techniques includes nonlinear control, which is complex and required more insight of the model but will be able to work in the whole state space without any constraint [3].

One thing is not shown in the state equation is the dynamics of the rotors (actuators), which is also a part of the system. To integrate the model into the design, two approach can be made:

- Design a main controller without the dynamics of the actuators. After that, design a controller for the actuators such that the dynamics of the actuators tracks the output of the main controller. Finally, adjust the control parameters to obtain desired performance.
- Design one single controller for the combined actuators and the process.

Since doing the first method requires more insight about the dynamics of the rotors, which is unavailable in this thesis, the second one will be approached.

Rewrite the linearized model into differential equations form, we have:

$$\ddot{x} = (\theta c \psi_{ss} + \varphi s \psi_{ss})g$$

$$\ddot{y} = (\theta s \psi_{ss} - \varphi c \psi_{ss})g$$

$$\ddot{z} = \frac{U_1}{m} - g$$

$$\ddot{\varphi} = \frac{U_2}{I_{xx}}$$

$$\ddot{\theta} = \frac{U_3}{I_{yy}}$$

$$\ddot{\psi} = \frac{U_4}{I_{zz}}$$

For the z position, it is simply a second order linear differential system which can be controlled using the simple PID tuning methods. Furthermore, in the hovering condition, the input U_1 only effect z dynamics, therefore no coupling occurs in this situation. The control structure is presented in figure 4.3.

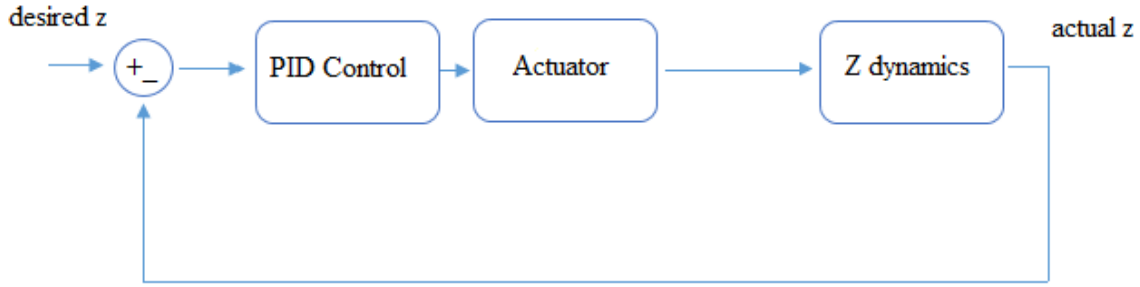


Figure 4.3 Altitude (z) control structures

Since \ddot{x} and \ddot{y} depends on θ, φ , it means that the translational motions in X and Y axis depend on U_2 and U_3 . Intuitively from chapter 2, the design process should start from designing a control law for x and y dynamics and then let θ and φ as:

$$\theta = g \times (pid(x, x_{sp})c\psi + pid(y, y_{sp})s\psi)$$

$$\varphi = g \times (pid(x, x_{sp})s\psi - pid(y, y_{sp})c\psi)$$

However, this arises an issues. The dynamics of θ and φ depends on U_2, U_3 which also need to be control; therefore, suppose control law $pid(\theta, \theta_{ss})$ and $pid(\varphi, \varphi_{ss})$ can be designed on U_2 and U_3 to stabilize θ and φ , the question is that whether the overall cascaded control that have been designed is stable or not. In short, the general form of this can be expressed as:

$$\dot{x} = f(x, \omega)$$

$$\dot{\omega} = s(\omega)$$

In [10], it has been shown that the cascade system will be stable if the inner system and outer system are both stable. Therefore, this methodology is acceptable when designing a position controller for the quadrotor system. In short, we can apply the design controller from the most inner loop to the outer loop, with adjustment during each design step to obtain an acceptable performance of the system. To keep the process on track, the design process in simulation and implemented system should be done in parallel in order to make sure that we do not miss the designing track.

With the stated method above, before controlling the position, it is required stabilize the rotation dynamics. This is done by using figure 4.4 cascade control structure and denote the overall system as Q:

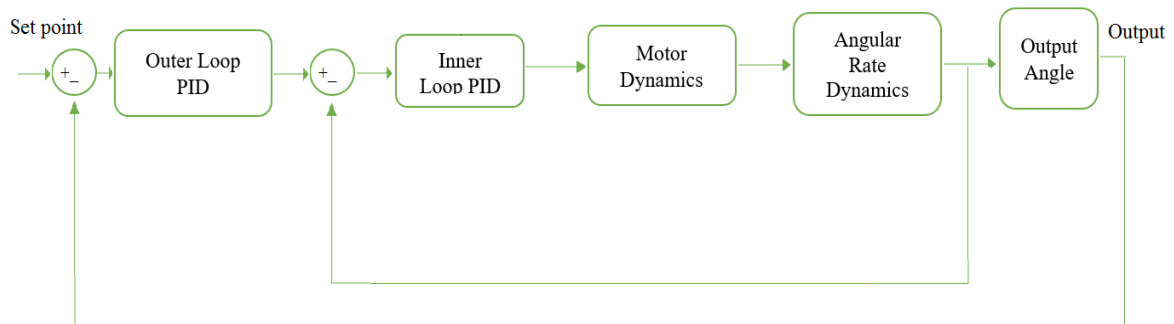


Figure 4.4 Control Structure for rotation dynamics of quadrotors (system Q)

Now, the step for designing the controller for the horizontal position can be done by structure in figure 4.5.

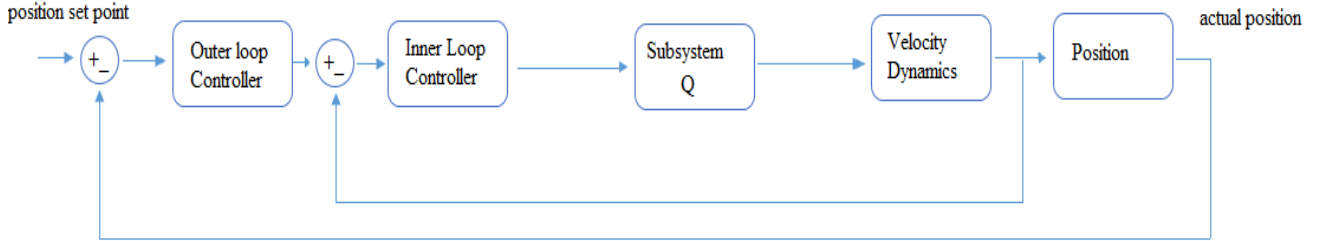


Figure 4.5 Position x and y control architecture

This concludes this chapter on how to stabilize the quadrotor system using PID control method. In summary of this chapter, the steps for design a PID control of the quadrotor is shown below:

- Design inner loop controllers for the roll, pitch and yaw movement
- Design outer loop controllers for the roll, pitch and yaw movement.
- For the altitude (z) control, design a single PID controller for stabilize the quadrotor to a certain altitude
- Design the inner loop controllers for the x and y translational movement
- Finally, design the outer loop controllers for the x and y translational movement.

In total, the number of PID controllers required for this quadrotor system is 11. For the z controller, we can separate the derivative action as the P gain for the altitude velocity (\dot{z}) control. So in other interpretation, 12 PID controllers are used in the system.

CHAPTER V

SIMULATION AND EXPERIMENTAL RESULTS

From the previous chapter, the methodology for controlling the system is already discussed. Following that methodology, this chapter will show the simulation result by using Simulink. After that, the controller will be implemented in the real system and the response between the two will be compared.

5.1 Simulation results

Based on results from chapter 3 and chapter 4, the Simulink model of the system is constructed. Since we are only interested in the hovering operating states of the quadrotor, the linearized equation of the quadrotor will be implemented in the Simulink model, which will help us reduce the model's complexity and still hold the important system's characteristic in the hovering operating points. The full diagram of the simulation file is attached to the CD with this thesis,

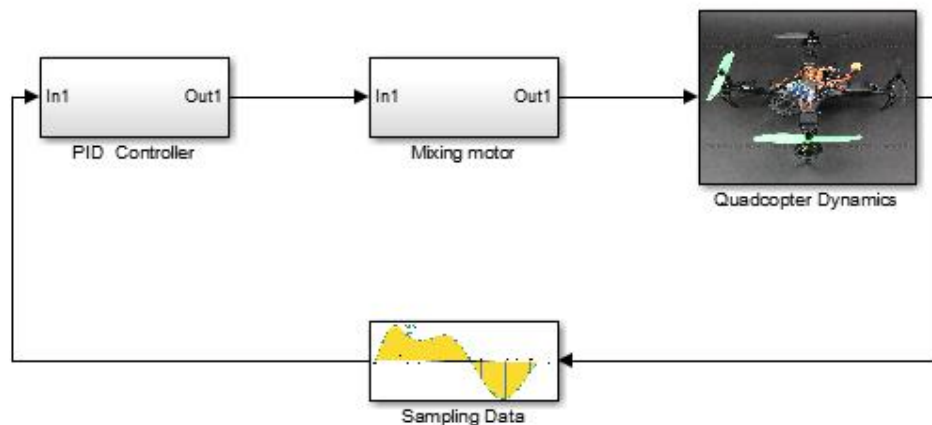


Figure 5.1 Overall structure of Simulink files

The PID controller in this simulation is implemented as discrete type controller. The tuning process for the PID parameter is done from the inner loop to the outer loop, which guarantees that the designed system will be stable.

The parameters of the PID controller is represented in Table 5.1

Controller Name	K_p	K_i	K_d
Roll and Pitch Inner Loop Controller	7.55	0.05	0.45
Roll and Pitch Outer Loop Controller	3.25	0.03	0.02
Yaw Inner Loop Controller	4.0	0.01	0.0
Yaw Outer Loop Controller	3.2	0.3	0.2
Altitude (z) Controller	2.5	2.0	1.2
Position Inner Loop Controller	8.40	0.02	0.02
Position Outer Loop Controller	0.9	0.001	0.001

Table 5.1 PID Controller parameters

Below is the plot of the quadrotor simulation movement with the following set points of its position:

$$x_{sp} = 0.2m$$

$$y_{sp} = 0.2m$$

$$z_{sp} = 0.45m = 45cm$$

$$\psi = 0 \text{ rad} = 0^\circ$$

For easier data interpretation, the plot of angle in the simulation is degree. The following plots have shown that our designed PID controller works well with accepted responses in the

simulation. Note that the plot of x and y is the same since in this thesis, we have assumed an absolute symmetry of the quadrotor in the system, which is obviously not perfect in the implemented system.

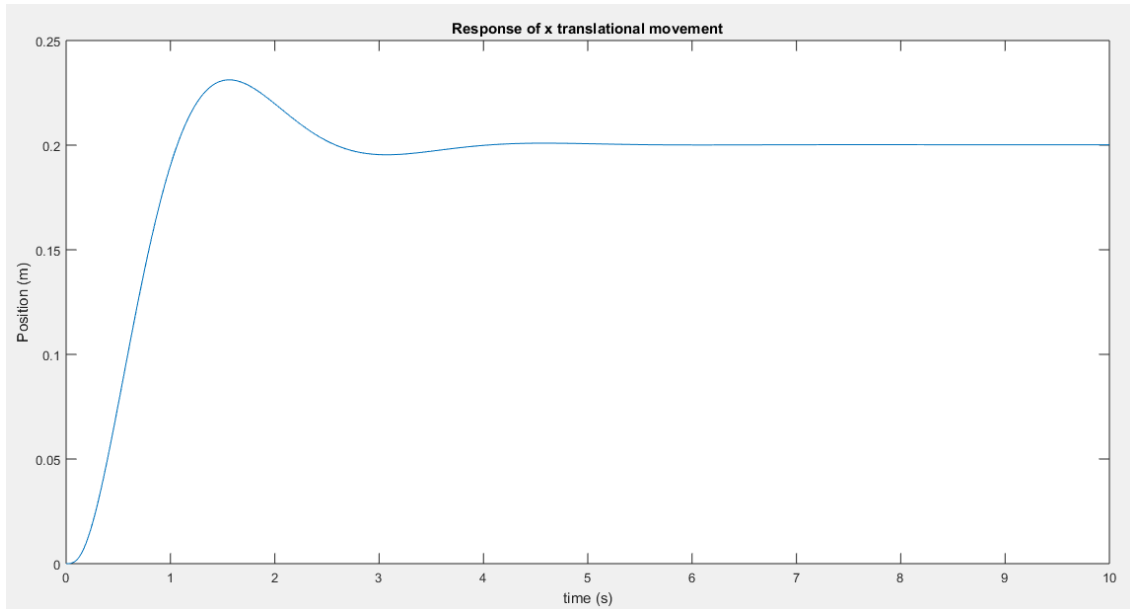


Figure 5.2 Response of x position

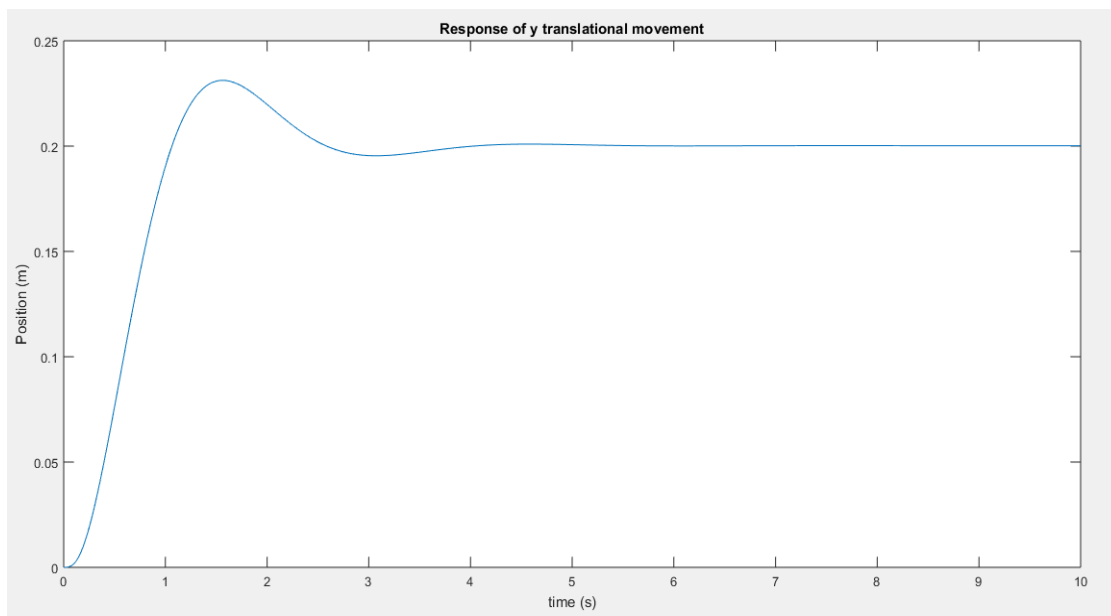


Figure 5.3 Response of y position

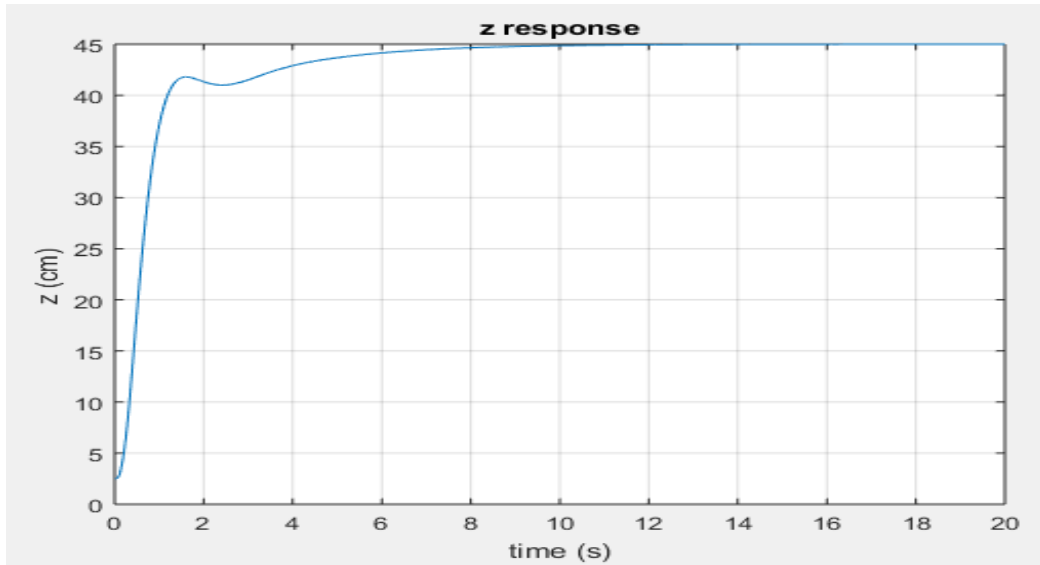


Figure 5.4 Response of z position (altitude)

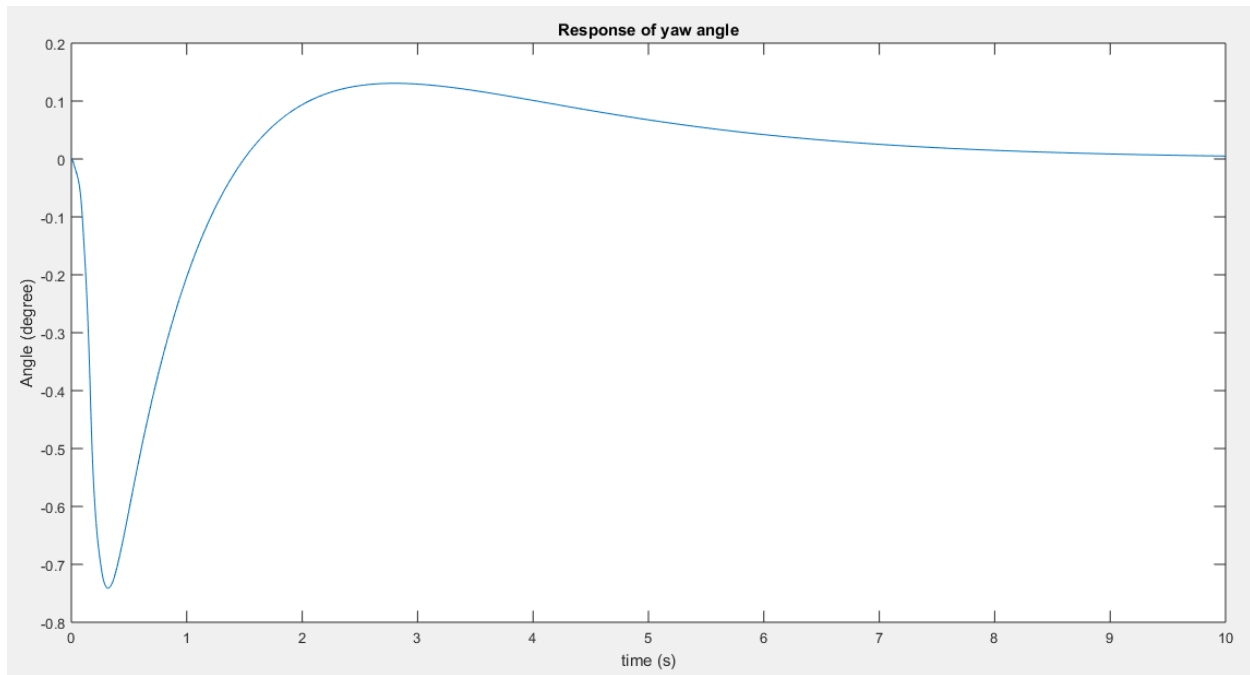


Figure 5.5 Response of yaw angle (heading)

To have more insights on how the quadrotor move to the desired position, the plot of the roll and pitch angle is shown in Figure 5.6 and 5.7

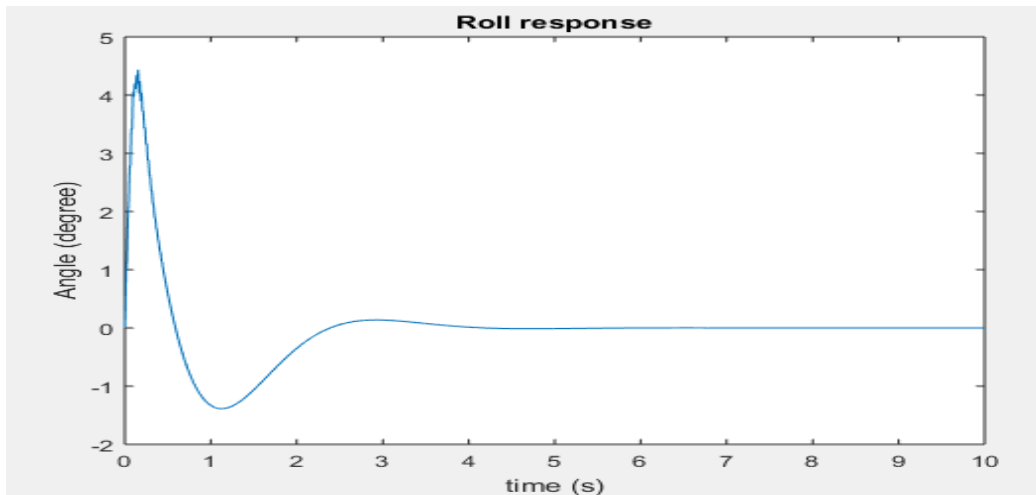


Figure 5.6 Roll response of the quadrotor

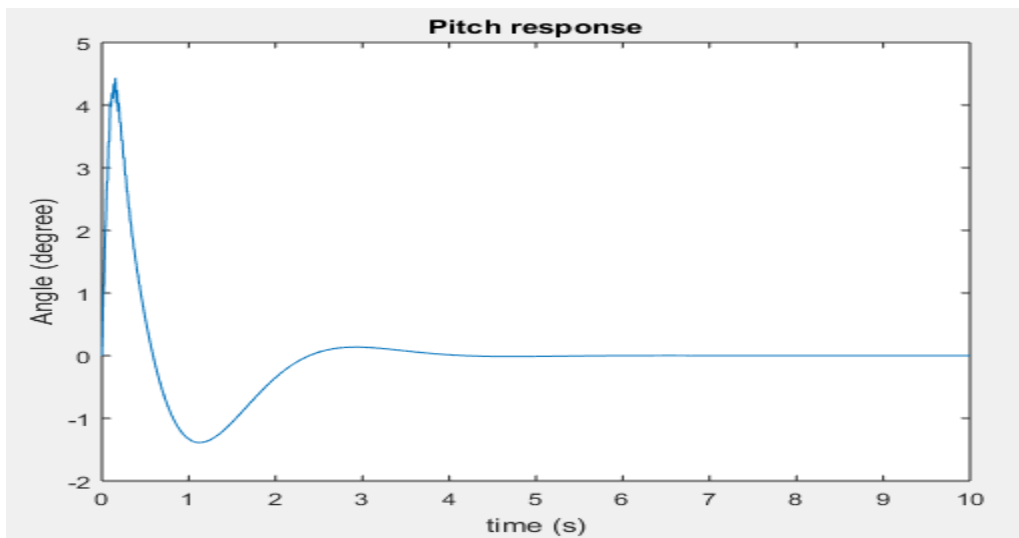


Figure 5.7 Pitch response of the quadrotor

5.2 Experimental results

In the experimental part, we have our quadrotor performs a vertical takeoff action, which means that the set point of the states are:

$$x = 0m$$

$$y = 0m$$

$$z = 0.45m = 45\text{ cm}$$

$$\psi = 0$$

The following figures shows the response from the implemented system. One important thing that needs to be mentioned is that due the requirement of sampling time, the Bluetooth communication can only sends one single type of data. This means that these plots are obtained from different experiments.

In the simulation, if we do vertical takeoff action, the responses of x and y will be zero (overlap with the set point in the simulation) since we do not include noise in the simulation. For the altitude, the response is easily to be observed.

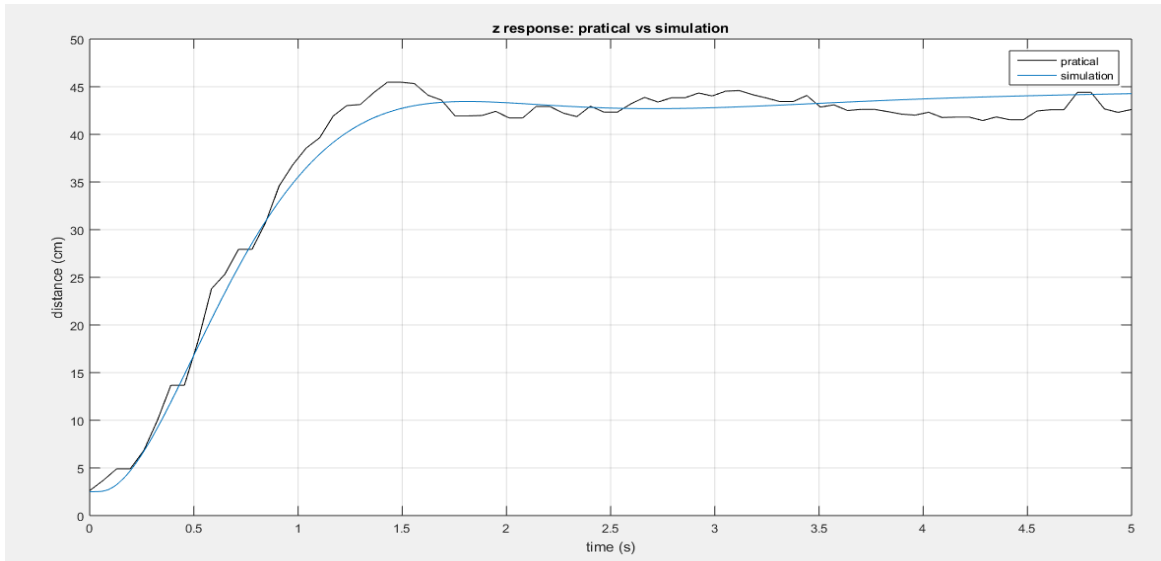


Figure 5.8 Altitude response between practical and simulation

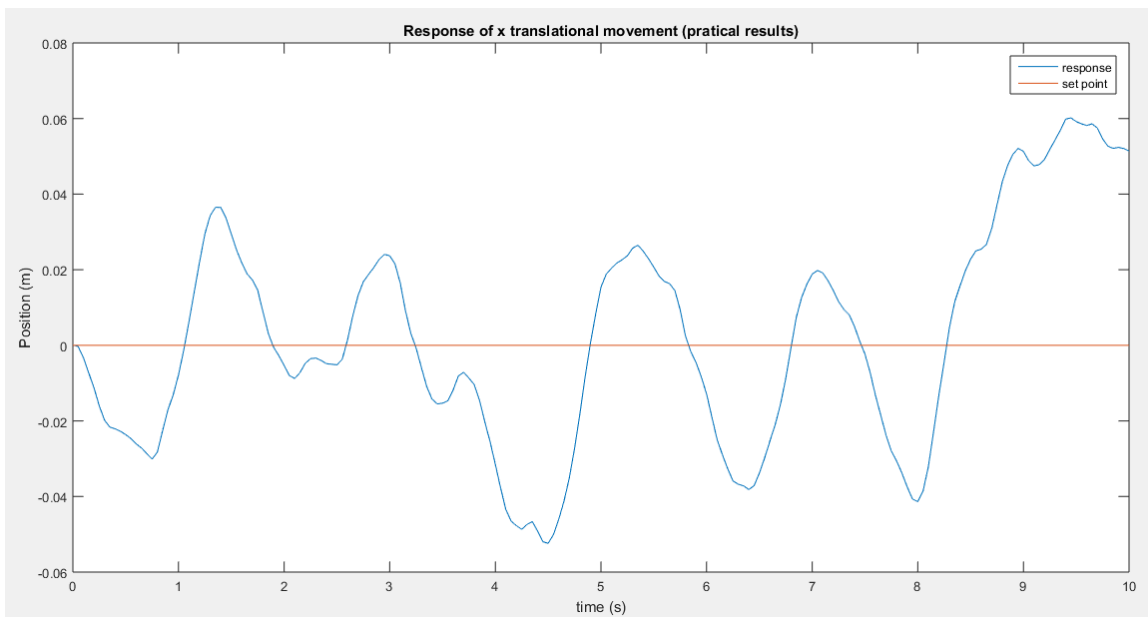


Figure 5.9 Response of x position of implemented system

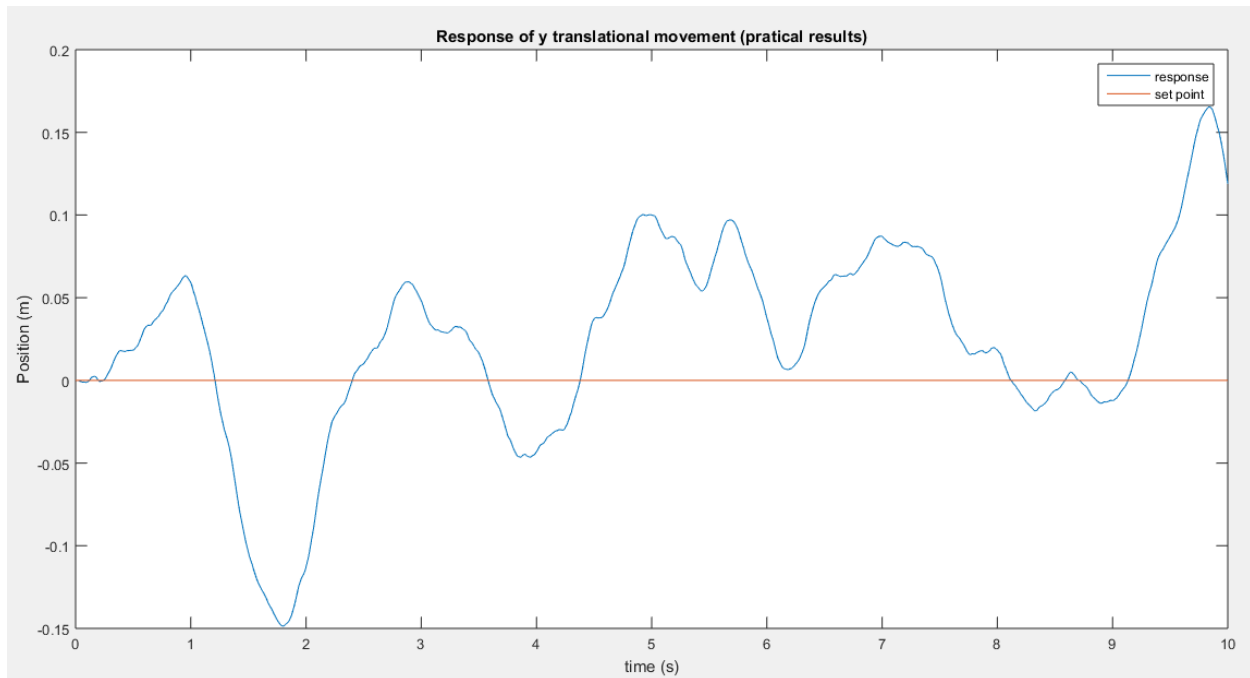


Figure 5.10 Response of y position of implemented system

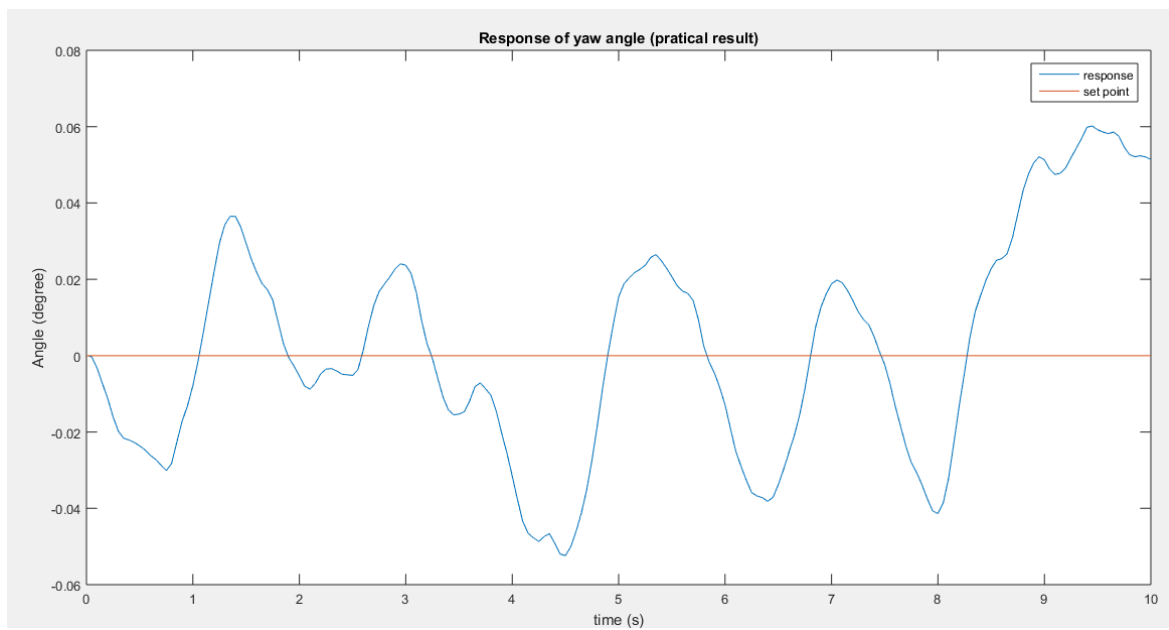


Figure 5.11 Response of yaw angle of implemented system

As can be seen from the graphs, the implemented system always try to bring the output close the set point of the system. The last figure shows the real quadrotor in action.



Figure 5.12 Quadrotor vertical takeoff

In terms of z control, the response is very close to the simulation result. For the yaw control, the angle stays close to 0 as it should be. However, it is noteworthy to notice a disturbance represented in the system when controlling the x and y positions. Visually, the quadrotor does not perfectly hovering at 1 place like other commercial product like the Phantom 4 Quadcopter from DJI. The reasons from this includes but not limit to:

- The implemented system is not perfectly symmetric
- Noise from vibration and wind
- The optical flow sensor is based on the intensity of the captured pictures. The shadow of the quadrotor during the flight can cause the sensor to be confuse.
- There is bias in center of mass in the quadrotor.

- The transfer function of the rotor is not included in the system, which can cause the designed PID controller not optimal in practice.
- Other aerodynamics effects that we ignore in chapter 2 can also contribute to the imperfect in response.

Once these factors are eliminated, performance should be better than the current states.

CHAPTER VI

CONCLUSION AND RECOMMENDATION

In this project, the method of controlling a quadrotor using a PID controller has been presented. The PID controller design procedure is based on simulation technique and it shows that the overall controlled system can stabilize itself. Going through all steps of the simulation and the design procedure helps us to get insight of “the control world”; the lesson learned is from this project is multifold: how to model and simulate complex systems and how to effectively use simulation in designing practical systems.

A quadrotor system, due to its MIMO structure, is in fact a complex system. Its state equations are unstable, which absolutely require a properly designed controller to make it stable. Though it is not shown in the thesis, during the process of experimenting the quadrotor flight, the implemented quadrotor has crashed many times; this might be due to some mistakes in implementing the controller as well as in building simulation files. With the method of controlling the quadrotor using PID controller, we are able to bring the quadrotor to its hovering conditions. The method of designing such controller is also discussed both intuitively and mathematically in this thesis.

This designed quadrotor is partially autonomous, which means that it does not require human actions to control like commercial quadrotors. Since time is limited, the interface to send command from human to the quadrotor is not fully developed; only interface is built for starting and ending the vertical takeoff action.

About the response of the quadrotor, as noted in chapter 5, one reason for the lagging response of the experimental quadrotor is due to the optical flow sensor. Although the flow method is simple, it is not really perfect. For a better position control without the motion capture system, feature recognition methods and SLAM are often involved [11].

For future work, it would be interesting to try to improve the performance of the experimental quadrotor. When acceptable response is achieved, there are multiple research paths involving the quadrotor can be done such as navigation, control theory, SLAM...Also one important thing to be recommended is the use of better performance communication links so that a full response can be recorded and sent without interrupting the sampling interval of the system.

APPENDIX A

MOMENTS OF INERTIA COMPUTATION OF QUADROTOR

Before computing the inertia of the quadrotor, we first assumed the shape of the quadcopter is a solid cylinder center with mass M , height H , radius R in the center and a point mass m_r represented a motor weight. Also, let L be the arm length.

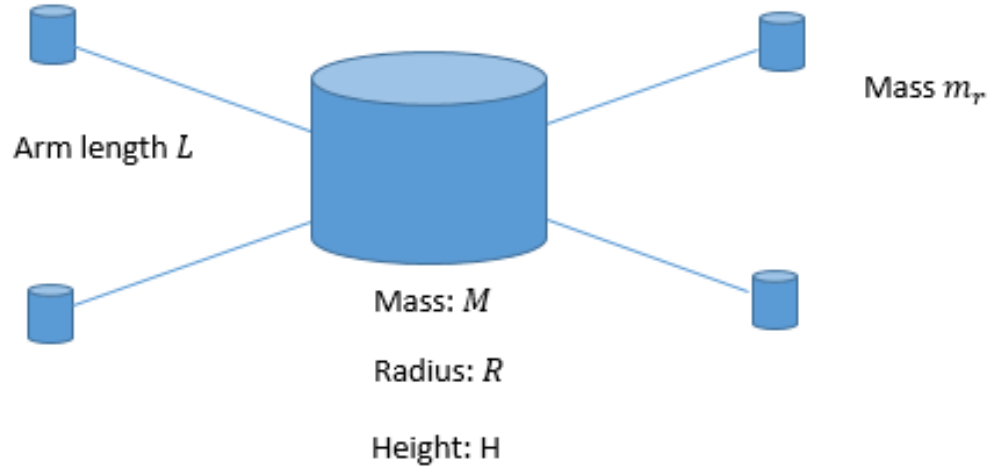


Figure A.1 Approximate model for calculating moment of inertia of the quadrotor

The inertia for the center hub can be calculated as:

$$I_{xx \text{ center}} = I_{yy \text{ center}} = \frac{1}{4}MR^2 + \frac{1}{3}MH^2$$

$$I_{zz \text{ center}} = \frac{1}{2}MR^2$$

The inertia for the point of mass around the center axis can be calculated as:

$$I = m_r L^2$$

Then, the quadrotor's moment of inertia can be calculated as:

$$I_{yy} = I_{xx} = I_{xx \text{ center}} + 2m_r L^2$$

$$I_{zz} = I_{zz \text{ center}} + 4m_r L^2$$

These equations provides us an approximate value for the system's moment of inertia.

APPENDIX B

EULER ANGLES AND THE ROTATION MATRICES

Euler angles are the angles which are used to describe the rotation motion of a rigid body. In fact, the Euler angles describes the rotation by the sequence of elemental rotation around axes. By this way, there are various ways to combine the three basics motion to describe the rotation of the rigid body. In this thesis, the rotation motions are defined by three angles around the z axis (ψ), x axis (φ) and y axis (θ) in the zxy order.

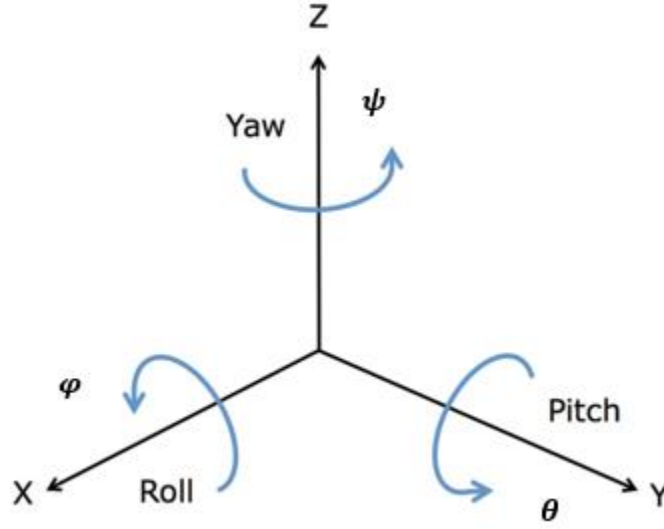


Figure B.1 Three angles rotation roll, pitch and yaw

For each rotation around a single axis, the corresponding rotation matrix is used for the transformation, which are:

- Around x axis: $R_x(\varphi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\varphi & s\varphi \\ 0 & -s\varphi & c\varphi \end{bmatrix}$

- Around y axis: $R_y(\theta) = \begin{bmatrix} c\theta & 0 & -s\theta \\ 0 & 1 & 0 \\ s\theta & 0 & c\theta \end{bmatrix}$

- Around z axis $R_z(\theta) = \begin{bmatrix} c\psi & s\psi & 0 \\ -s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Since we consider the rotation in the order zxy, then the transformation matrix results from those elemental rotation is:

$$R_{zxy} = R_z(\psi)R_x(\varphi)R_y(\theta) = \begin{bmatrix} c\psi c\theta - s\varphi s\psi s\theta & -c\varphi s\psi & c\psi s\theta + c\theta s\varphi s\psi \\ c\theta s\psi + c\psi s\varphi s\theta & c\varphi c\psi & s\psi s\theta - c\psi c\theta s\varphi \\ -c\varphi s\theta & s\varphi & c\varphi c\theta \end{bmatrix}$$

Figure B.2 illustrates the rotation above.

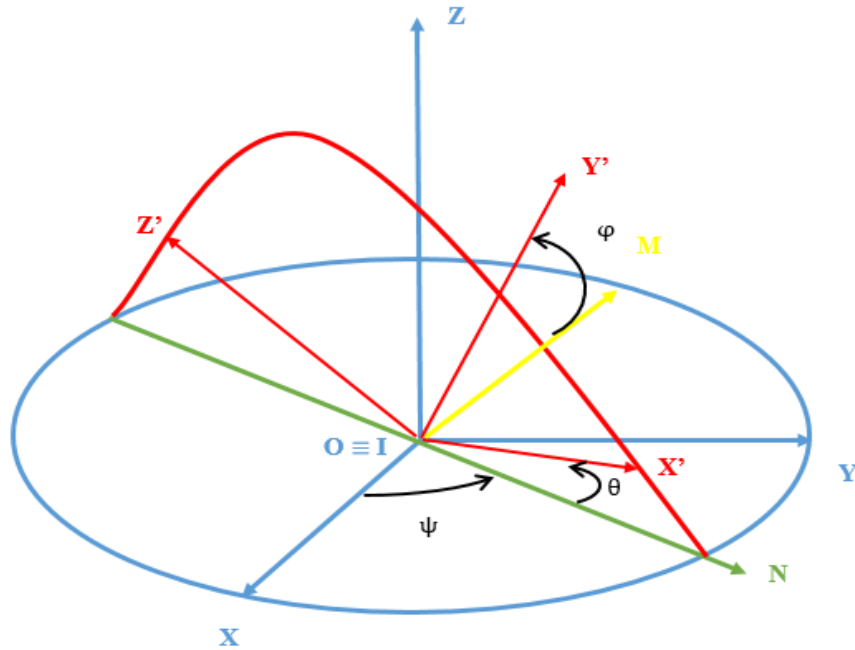


Figure B.2 Rotation in order z-x-y

APPENDIX C

PROJECT RELATED SOFTWARE

The related software for the implemented system are attached in the CD, contains:

- Flight Controller software written in C++ with Arduino IDE
- Matlab User Interface to control the quadrotor

C.1 Flight Controller Software:

Figure B.1 shows the flowchart of the flight controller software of the quadrotor.

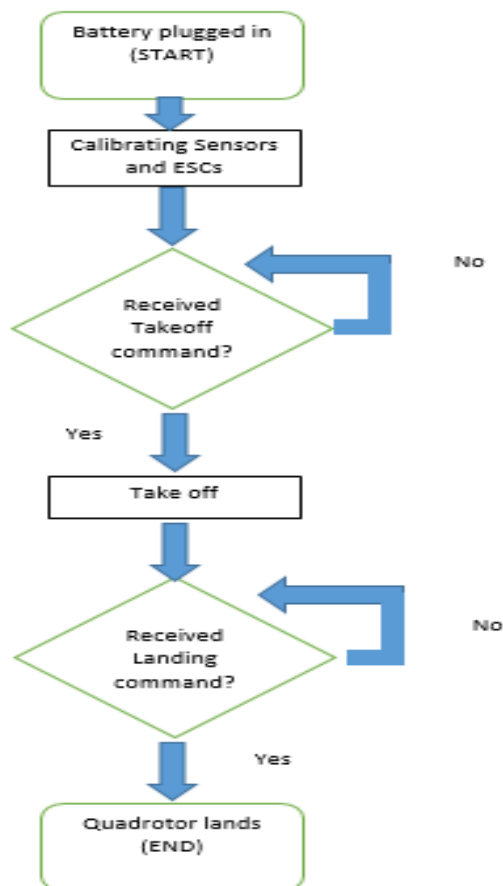


Figure C.1 Flow chart of the quadrotor system

C.2 User Interface Guideline:

In this guideline, the step of how to use the user interface to control the quadrotor is presented. Before using this file, it is required the user to enable the Bluetooth of their computer.

The steps is shown as follows:

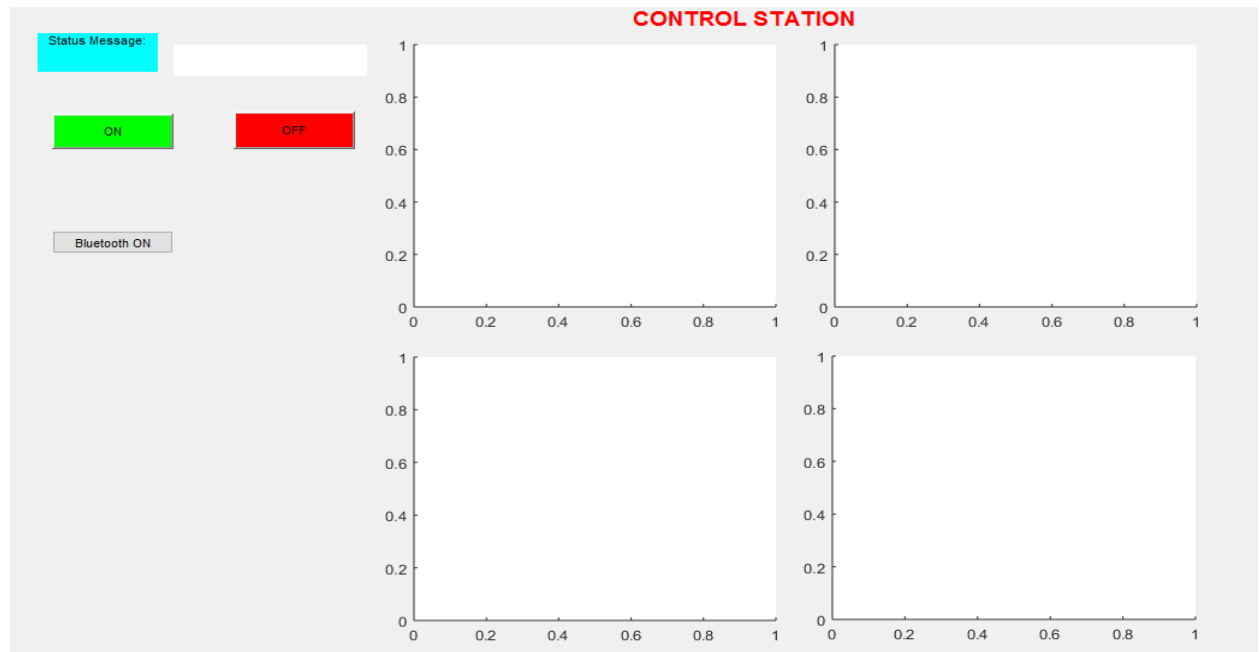


Figure C.2 User Interface in Matlab

- After the battery is plugged in, click on the button “Bluetooth ON” to establish a connection between the computer and the quadrotor.
- Wait until the message “Connection Ready” is shown on the text box
- Click on “ON” button for sending command to make the quadrotor take off
- Click on “OFF” button for sending command to make the quadrotor landing.

To enable the plotting features of this file, the instructions is shown in the Matlab code.

Note that for the system to be stable, only type of data can be plot at the time.

REFERENCES

- [1]"Quadcopter", Wikipedia, 2016. [Online]. Available:
<https://en.wikipedia.org/wiki/Quadcopter> [Accessed: 16- May- 2016].
- [2]"New algorithm makes quadcopters safer", Ethz.ch, 2016. [Online]. Available:
<https://www.ethz.ch/en/news-and-events/eth-news/news/2013/12/new-algorithm-makes-quadcopters-safer.html> [Accessed: 16- May- 2016].
- [3] S. Bouabdallah, "Design and control of quadrotors with application to autonomous flying", Ph.D, École polytechnique fédérale de Lausanne, 2007.
- [4] Cobra 2204/28 Motor Test Data with GemFan 5x3 Prop
<http://www.innov8tivedesigns.com/images/specs/CM-2204-28-GEM-5x3-Perf-3S.pdf>
- [5] C. Militello, L. Rundo and M. Gilardi, "Applications of imaging processing to MRgFUS treatment for fibroids: a review", Translational Cancer Research, vol. 3, no. 5, pp. 472-482, 2014.
- [6] K. Åström and T. Hägglund, *Advanced PID control*. Research Triangle Park, NC: ISA-The Instrumentation, Systems, and Automation Society, 2006.
- [7] S. Chander, P. Agarwal and I. Gupta, "Auto-tuned, discrete PID controller for DC-DC converter for fast transient response", India International Conference on Power Electronics 2010 (IICPE2010), 2011.
- [8] G. Goodwin, S. Graebe and M. Salgado, Control system design. Upper Saddle River, N.J.: Prentice Hall, 2001.

- [9] L. Argentim, W. Rezende, P. Santos and R. Aguiar, "PID, LQR and LQR-PID on a quadcopter platform", 2013 International Conference on Informatics, Electronics and Vision (ICIEV), 2013.
- [10] P. Seibert and R. Suarez, Global stabilization of nonlinear cascade systems, *Systems and Control Letters* 14, 347-352 (1990).
- [11] S. Grzonka, G. Grisetti and W. Burgard, "A Fully Autonomous Indoor Quadrotor", *IEEE Trans. Robot.*, vol. 28, no. 1, pp. 90-100, 2012.