

© 2016 by Mingyo Seo. All rights reserved.

APPLICATION OF A MODEL-BASED NONLINEAR ATTITUDE CONTROL FOR
QUADTOROR UAVS

BY

MINGYO SEO

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Mechanical Science and Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2016

Urbana, Illinois

Master's Committee:

Abstract

Acknowledgments

Table of Contents

List of Tables	vi
List of Figures	vii
List of Symbols	ix
Chapter 1 Introduction	1
1.1 Background	1
1.2 Related Works	1
1.3 Motivation	2
1.4 Objective	3
1.5 Outline	3
Chapter 2 System and Hardware Description	5
2.1 Hardware Requirements	5
2.2 Hardware Specification	6
2.2.1 Frame	6
2.2.2 Actuators and Electronic Speed Controls	6
2.2.3 Propeller	6
2.2.4 Autopilot	8
2.2.5 Companion Computer	9
2.2.6 Camera	9
2.2.7 Ground Control Station	9
2.3 System Overview	10
2.4 Physical Properties	11
Chapter 3 Control System	16
3.1 Control System Overview	16
3.2 Coordinate System	16
3.3 Outer-loop Controller	19
3.3.1 Dynamics Description	19
3.3.2 Computation of Desired Velocity	21
3.3.3 Computation of Desired Thrust and Attitude	21
3.4 Inner-loop Controller	22
3.4.1 Dynamics Description	22
3.4.2 Computation of Reference Angular Velocity	24
3.4.3 Computation of Desired Torque	26
3.5 Motor Control	29
3.5.1 Mapping Attitude and Thrust Control into Motor Speeds	29
3.5.2 Approximation of the Aerodynamic Coefficient Matrix	31
3.6 Control of Motors	32

Chapter 4 Evaluation of the Control System	37
4.1 Linear Attitude Control for Comparison	37
4.2 Simulation Study for Inner-loop Controller	38
4.2.1 Simulation Setup	38
4.2.2 Results	39
4.2.3 Discussion	39
4.3 Flight Experiment	42
4.3.1 Motion Capture Arena	42
4.3.2 Motion Capture System	44
4.3.3 Experiment Setup	44
4.3.4 Experiment Results	46
4.3.5 Discussion	46
Chapter 5 Vision-based Position Estimator	55
5.1 Camera Calibration	55
5.2 Marker Detection	57
5.3 Position Estimation	57
5.4 Evaluation Experiment	59
5.4.1 Experimental Setup	59
5.4.2 Results	60
5.4.3 Discussion	60
Chapter 6 Conclusion	63
References	64

List of Tables

2.1 Table caption text	8
----------------------------------	---

List of Figures

1.1 A Crazyflie Nano Quadrotor with a Downward Camera	3
2.1 Illustration of the Quadrotor Frame and the Rotors	6
2.2 Brushless DC Motor	7
2.3 Electric Speed Control (ESC)	7
2.4 11.1V 2700mAh LiPo Battery	7
2.5 Pixhawk Autopilot	8
2.6 Odroid XU4 Single-board computer	9
2.7 USB Camera	10
2.8 Quadrotor Testbed	11
2.9 Description of the Quadrotor Assemble	12
2.10 System Architecture for Controlling the Quadrotor	13
2.11 CAD model of the Quadrotor (upper side)	15
2.12 CAD model of the Quadrotor (under side)	15
3.1 The Illustration of the Control System.	17
3.2 The Illustration of the inertial and body frames.	18
3.3 The Thrust and Torque Generated by Each Rotor	29
3.4 Comparison of experimental and approximated thrust coefficient	33
3.5 Comparison of experimental and approximated power coefficient	33
3.6 Change of RPM output corresponding to PWM command	35
4.1 Simulation Result (Case 1)	40
4.2 Simulation Result (Case 2)	40
4.3 Simulation Result (Case 3)	40
4.4 Simulation Result (Case 4)	41
4.5 Simulation Result (Case 5)	41
4.6 Simulation Result (Case 6)	41
4.7 Simulation Result (Case 7)	43
4.8 Motion Capture Arena of the Intelligent Robotics Laboratory	43
4.9 Vicon Motion Capture Cameras	44
4.10 Vicon Motion Capture Interface	45
4.11 Experiment Result (Case 1)	47
4.12 Experiment Result (Case 2)	48
4.13 Experiment Result (Case 3)	49
4.14 Experiment Result (Case 4)	50
4.15 Experiment Result (Case 5)	51
4.16 Experiment Result (Case 6)	52
4.17 Experiment Result (Case 7)	53
5.1 Camera Calibration (Left: Raw Image, Right: Restored Image)	56
5.2 Gaussian-filtered and Thresholded Image	58

5.3	Restoration of Borders	58
5.4	Final Result of Marker Detection	58
5.5	Image Markers for the Evaluation	60
5.6	Experiment Result of the Position Estimator (x-axis Position)	61
5.7	Experiment Result of the Position Estimator (y-axis Position)	61
5.8	Experiment Result of the Position Estimator (z-axis Position)	61

List of Symbols

Chapter 1

Introduction

1.1 Background

An unmanned aerial vehicle(UAV) is defined as an aircraft that is maneuvered without a human pilot onboard.[1] A quadrotor is an UAV that consists of two pairs of counter-rotating rotors with fixed-pitch blades.[2] With independent output of each motor, the quadrotor has full control of its orientation(roll, pitch, and yaw) and thrust. The quadrotor's attitude and thrust are coupled with the translation motion of the quadrotor; the direction of the translation motion is determined by the orientation and the magnitude of the translation motion is determined by the thrust since thrust occurs in only the perpendicular direction of the quadrotor's orientation. Therefore, the quadrotor controls its translation by altering its attitude and has a capacity of both rotation and translation maneuver. Due to the advantage, quadrotors have been used for various purposes and performed tasks in environments that are nearly impossible to reach as human. As an example, Michael et al. developed an autonomous quadrotor to explore an earthquake-damaged building with multiple floors collaborating with a ground robot.[3] Chambers et al. developed a quadrotor system that maps rivers with a self-supervised river classification system.[4]

A control system for quadrotors is usually consists of position control(outer-loop) and attitude control(inner-loop). Position control maps input of translation motion into desired orientation and thrust. Attitude control computes desired torque corresponding to the desired and current attitude. However, because of factors like the effect of rotation on translation motion, and aerodynamic effects, the dynamic model of the quadrotor is highly non-linear. Therefore, there is a demand of advanced control system for the quadrotor's performance.

1.2 Related Works

For the improvement of quadrotors' performance, research on advanced control system for a quadrotor has been conducted actively. Raffo et al. applied a nonlinear \mathcal{H}_∞ control to attitude control and designed a backstepping control for path tracking of a quadrotor. The control system was evaluated by simulations.[5]

Morgan et al. designed and implemented a dynamic-model-based nonlinear attitude control in the swarm assignment and trajectory optimization.[6] In the case of the application of adaptive control, Mallikarjunan et al. demonstrated an application of \mathcal{L}_1 adaptive control for quadrotors' attitude control by experiments.[7] Also, Diao et al. suggested a Lyapunov based nonlinear adaptive control for a quadrotor under uncertainties of mass, inertia, and aerodynamic damping coefficients, and evaluated the control by simulations.[8]

On the other side, trajectories of a quadrotor is also studied by many researchers to improve its performance. Ritz et al. designed a method to compute a quadrotor's time-optimal trajectory between two given states, and evaluated the method by simulations.[9] Mellinger et al. developed an algorithm of trajectory generator for a quadrotor's aggressive maneuvers, and validated the performance by experiments.[10] In addition, Dierksand and Jagannathan suggested a control scheme that uses control inputs computed by neural networks.[10]

1.3 Motivation

Due to the high nonlinearities of a quadrotor's dynamics, the application of advanced controls is expected to improve the performance of the quadrotor, compared with ordinary control systems, such as PID and Linear-Quadratic Regulator. The application of studies of a quadrotor's dynamics is expected to increase the robustness of its system.

The Aerospace Robotics and Control Laboratory at the University of Illinois had developed quadrotor platforms based on Crazyflie 1.0 to validate the performance of nonlinear controllers.[6] Crazyflie is a open-source nano quadrotor system developed by Bitcraze.[13] A Crazyflie quadrotor is about 19g weight and each leg is about 6cm length. However, due to the quadrotor's physical constraints, such as its load limit and precision of the sensors, the application of the platforms is limited. Especially, in order to develop a vision-based position control system and other high-level functions for the quadrotor, 8g of load is allowed for sensors, equipments, and mount for them. Therefore, the choice of sensors and algorithm is limited.

In order to solve such problems, a micro quadrotor can be used alternatively. With greater load capacity, a micro quadrotor can equip with various light-weighted devices onboard, such as a wide-angle camera and a companion computer. In addition, more various autopilot systems are available for a micro quadrotor than for a nano quadrotor. For these advantages, a micro quadrotor is expected to be a better research platform for various purposes, and therefore, this research aims for developing a new micro quadrotor system.



Figure 1.1: A Crazyflie Nano Quadrotor with a Downward Camera

1.4 Objective

In this research, we aim to evaluate the application of a nonlinear attitude control, suggested by Bandyopadhyay and Chung, that was originally designed for a spacecraft.[12] In order to implement the control system, we develop a new micro quadrotor system. The quadrotor's dynamics is modeled and characterized based on experimental data. The performance of the quadrotor with the nonlinear attitude control is evaluated by both flight experiments and simulations and compared with ordinary PID attitude control. In addition, we suggest a light-computing algorithm to estimate the position of the quadrotor by capturing image data of visual markers, and the algorithm is evaluated by experiments, independently from the quadrotor's flight.

1.5 Outline

The thesis is organized as follows. Chapter 2 describes the overview of the quadrotor system. This chapter also states the hardware specification of each component, the physical properties of the quadrotor, and the coordinate system. Chapter 3 focuses on the dynamic model and the control system. To be more specific, Section 3.2 defines the coordinate systems used for the quadrotor system. Section 3.3 describes the dynamic model of the quadrotor's translation motion and its position control. Section 3.4 presents the dynamics of the quadrotor's rotation and the nonlinear attitude control. The proof of the stability of the nonlinear control is also stated in this chapter. In chapter 3.5, a model of DC motors are introduced and a open-loop control of the motors is described. Chapter 4 presents the evaluation of the control system stated in chapter 3. In particular, section 3.1 describes an PID attitude control that is compared with the customized control system. Section 3.2 discuss numerical simulation results of the nonlinear and PID attitude control of the

quadrotor. In section 3.3, the results of flight experiments is presented with various inputs. Chapter 5 introduces a light-computing method to estimate the quadrotor’s position with given image data of visual markers, and discuss the performance of the algorithm. Finally, chapter 6 summarizes the contents of this thesis and suggests future works.

Chapter 2

System and Hardware Description

This research aims for the development and validation of a dynamic model based nonlinear attitude control for a quadrotor. In this chapter, the hardware requirements, specifications, and physical properties of the quadrotor system will be explored.

2.1 Hardware Requirements

An ordinary quadrotor system is equipped with a micro controller that provides stability and control per inputs from a human user using a radio control (RC) transmitter. In order to achieve autonomous operation, the quadrotor must meet the following requirements:

- The quadrotor must use wireless communication methods to access sensor values and set points
- The quadrotor must have sufficient computing power to process visual data
- The quadrotor must be able to execute an algorithm for autonomous operation

An autopilot's microcontroller does not have sufficient power to process visual data and execute the autonomous operation algorithm by itself. Therefore, an external processor is necessary. The processor can be either onboard the quadrotor or offboard. If the processor is onboard the quadrotor, then the quadrotor is now a more complex system, with higher power requirements and additional weight from the processor. The processing power would also be restricted due to size, weight, and power limitations. If the processor is offboard in a ground control station, then the restrictions of size, weight, and power would be lifted. However, this method increases latency. Since quadrotors are agile systems, low latency computation is preferred. Therefore, the system consists of an autopilot micro controller, a companion processor, actuators, and a backup RC controller with human pilot.

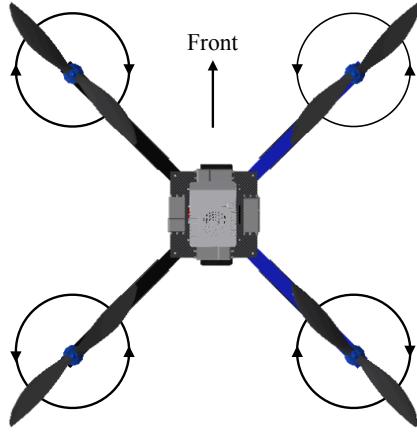


Figure 2.1: Illustration of the Quadrotor Frame and the Rotors

2.2 Hardware Specification

2.2.1 Frame

In the quadrotor system, a glass fiber X-shaped frame is used, as shown at the figure 2.1. The length of the quadrotor's arm is 0.225m and the weight is 270g. At the end of an arm, a brushless DC motor is installed. To reduce the moment of inertia, an autopilot, a battery and a power-distribution board are located at the center of the frame. To simplify the dynamics model and decrease the error between sensor values and the quadrotor's rotational motion, the motors and the autopilot is installed on the same plane of the quadrotor. Electronic Speed Controls (ESCs) are installed on each leg symmetrically.

2.2.2 Actuators and Electronic Speed Controls

A quadrotor controls its attitude and thrust by independent inputs of its four motors. Since we use brushless DC motors, we require ESCs to run the motors. Both the motors and ESCs are produced by 3D Robotics. Power is provided to each motor by an 11.1V 2700mAh 3-cell lithium polymer battery. A motor rotates at a frequency corresponding to its voltage, and the average voltage of a motor can be controlled by pulse width modulation (PWM). Details of motor control are stated in section 5.3. Each motor spins a APC 10 × 4.7 propeller, providing thrust and moment to each motor.

2.2.3 Propeller

In the quadrotor, 2 pairs of propellers, APC 10 × 4.7 Slow Flyer(SF) and Slow Flyer Pusher(SFP) propellers are used. Both SF and SFP propellers have same geometries as shown at the table 2.1 but are mirror

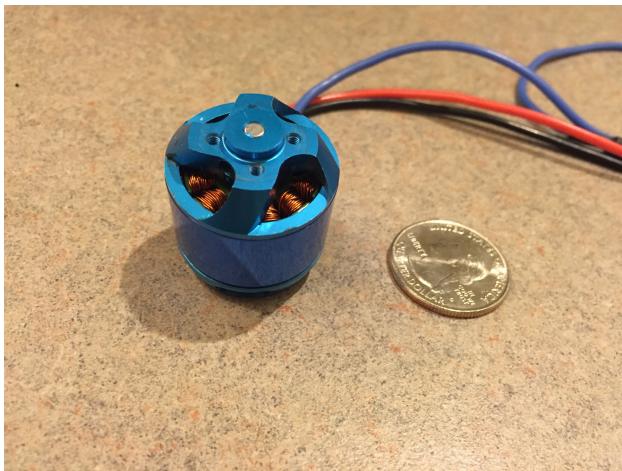


Figure 2.2: Brushless DC Motor



Figure 2.3: Electric Speed Control (ESC)



Figure 2.4: 11.1V 2700mAh LiPo Battery

	Entire Diameter	Hub Diameter	Hub Thickness	Shaft Diameter
Length [10^{-2} m]	25.4	1.27	0.74	0.64

Table 2.1: Geometry of APC 10×4.7 Slow Flyer



Figure 2.5: Pixhawk Autopilot

symmetric to each other. SF propellers are designed to rotate clockwise and SFP propellers are designed to rotate counter-clockwise. As shown in the figure 2.1, the front-right and rear-left rotors rotate counter-clockwise with SF propellers, and the front-left and rear-right rotors rotate clockwise with SFP propellers. The weight of each propeller is measured as 0.012kg.

2.2.4 Autopilot

A quadrotor is typically equipped with a microcontroller autopilot that is controlled by high level commands, such as position or attitude setpoints from a RC transmitter. Based on the current state and setpoint commands, the controller computes desired actuations and commands the actuators to respond accordingly. In order to detect the current state, the autopilot requires an accelerometer, gyroscope, and some form of actuator interface in order to function. Since most autopilot systems use a PID controller to control attitude, it is necessary to rewrite the control algorithm. In order to facilitate this, we choose to use the open source PX4 autopilot system. We used a 3D Robotics Pixhawk microprocessor to run our PX4 code. Pixhawk contains all the necessary sensors, computation power, and input/output interfaces. In addition to an accelerometer and gyroscope, the pixhawk has a magnetometer and barometer to better sense for altitude and attitude changes.

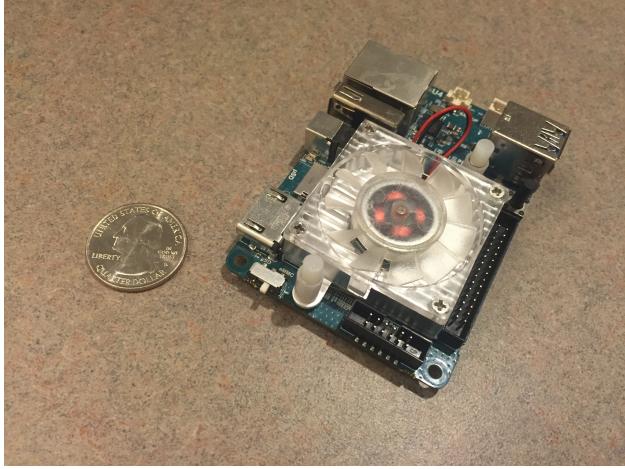


Figure 2.6: Odroid XU4 Single-board computer

2.2.5 Companion Computer

The companion computer is necessary for estimating the quadrotor position through computer vision. The computer receives image data from a ground facing camera and, using the quadrotor state from the autopilot, computes the position estimation of the quadrotor and sends the estimation data to the autopilot. The companion computer can also receive commands from a ground control station via SSH. The quadrotor uses a single Hardkernel Odroid XU4. It was a quadcore 1.3 GHz CPU and 2 GB of RAM. The computer runs on a 5 volt 4 amp power supply, which is provided by a voltage converter from the main battery.

2.2.6 Camera

The autopilot does not have precise position measurement, so it is necessary to estimate the quadrotor's position precisely using a computer vision system. Four circular markers on the floor are tracked by the quadrotor's camera to provide estimates as to its relative position with respect to the markers. A ground facing camera is used to capture the images of the markers. The USB webcam we use has a focal length of 2.1mm and has about 120° view angle. The camera sends images of 640×480 pixels at a rate of 30 frames per second.

2.2.7 Ground Control Station

The ground control station allows us to send commands and receive data remotely. Since the quadrotor system uses Wi-Fi network to communicate, any computer can be used as a ground control system so long as it is on the same Wi-Fi network. The companion computer can be accessed using any wireless interface, such as SSH. Using this interface, the quadrotor's operation can be started.

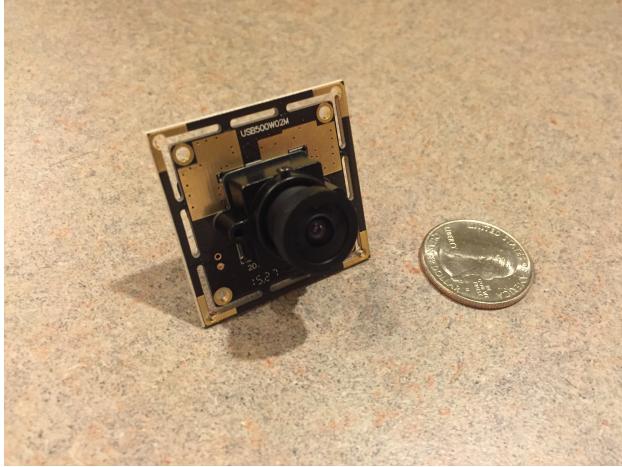


Figure 2.7: USB Camera

2.3 System Overview

The complete assembly of the quadrotor is shown in figure 2.7 and 2.8. An overview of the autonomous systems of the quadrotor is shown in figure 2.9.

In order to operate the quadrotor, the onboard companion computer must be connected to the ground control station through a Wi-Fi network. Now that the companion computer's console is available from the ground controls station. Through the GNU interface, the user can start and stop the autonomous operations of the quadrotor.

When the autonomous operation starts, the companion computer arms the quadrotor and increases the altitude of the quadrotor until the markers are in view. As soon as the ground facing camera captures the marker, the companion computer starts to estimate the position of the quadrotor. In order to estimate the position, the orientation of the camera must be known. Since the camera is fixed, the quadrotor's orientation is same as the orientation of the camera and its information can be measured by the magnetometer of the autopilot. Micro Air Vehicle Link (MAVLink) messages are used via uorb serial communication to communicate between the companion computer and the autopilot.

There are multiple modules necessary for quadrotor flight in the autopilot: Inner-loop controller (attitude control), outer-loop controller (position control), position/altitude estimator, attitude control, mixer. These modules are managed by a flight management system. The quadrotor's orientation, angular, rate, and acceleration are measured by the built in magnetometer, gyroscope, and accelerometer respectively. The quadrotor's position is estimated through the double integration of the acceleration and filters. The estimated altitude is corrected by the built in barometer. However, position in the other two dimensions are not corrected. As the autopilot receives MAVLink messages from the companion computer, the flight

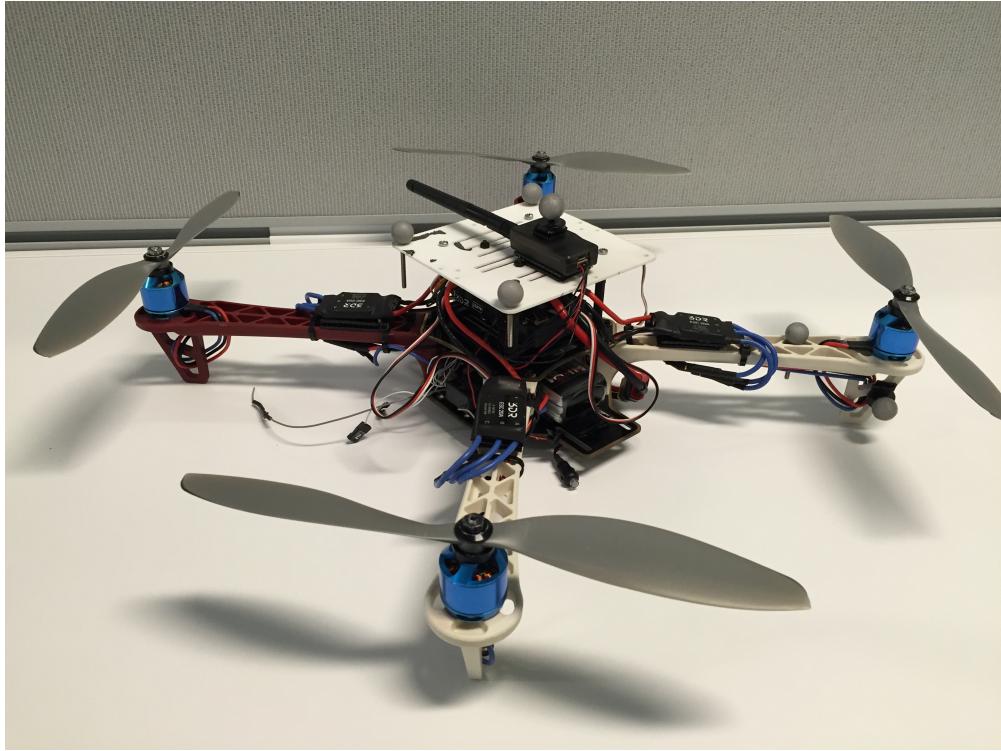


Figure 2.8: Quadrotor Testbed

management application arms the quadrotor and starts to send setpoint data to the outer-loop controller module. Then, the outer-loop controller computes desired attitude and sends the data to the inner-loop controller module. Continuously, the inner-loop controller module computer desired torques. The mixer module subscribes the desired torques, computes desired motor speeds, and sends signal to the ESCs to control the motors.

Open loop control is used for motor speed since the DC motors do not have encoders to measure motor speed. Instead, PWM signals are computed according to desired motor speeds to the module of the autopilot. Given PWM signals from the autopilot, each ESC controls the average voltage of its motor, which in turn controls the frequency at which the motor spins.

2.4 Physical Properties

In order to use the dynamic model of the quadrotor, we need to know the total mass and the moment of inertia. The total mass of the assembled quadrotor is 1.404kg. The moment of inertia is difficult to measure directly. There are two methods to estimate the moment of inertia: The first is to estimate the moment of inertia from the period of a physical pendulum. The other is to approximate the moment of inertia using a

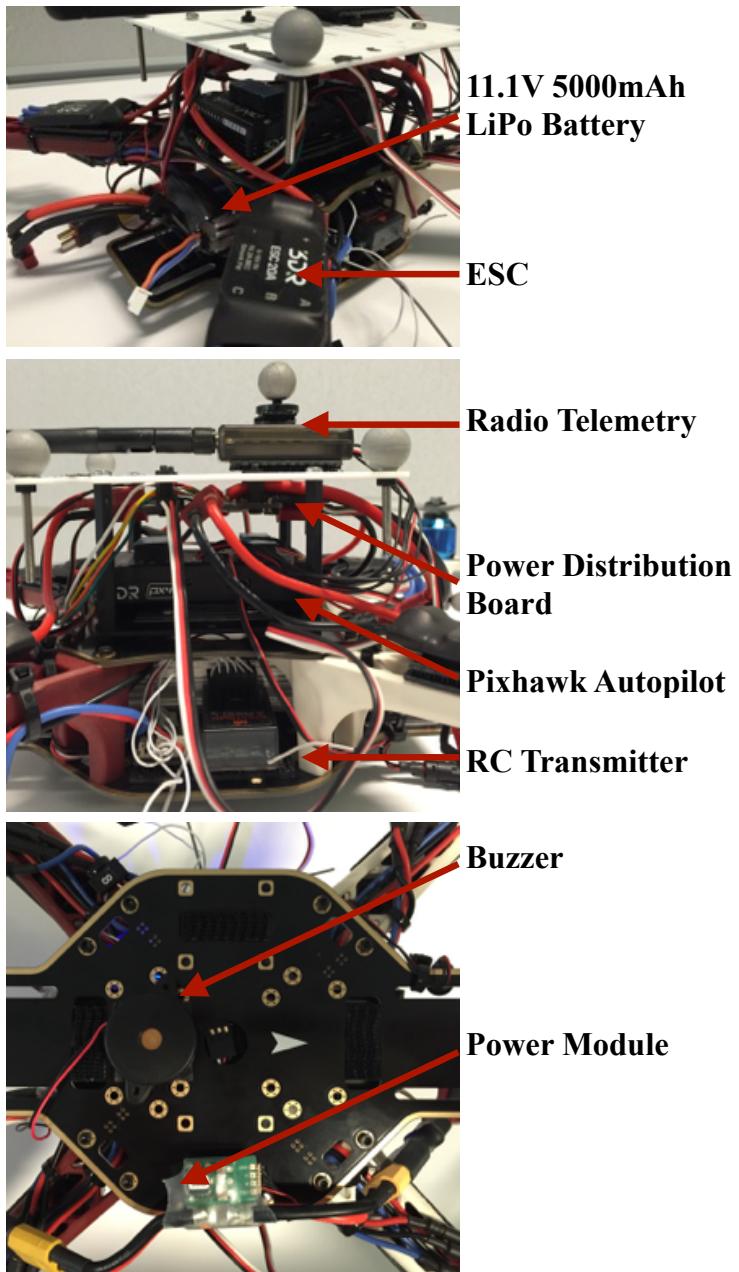


Figure 2.9: Description of the Quadrotor Assemble

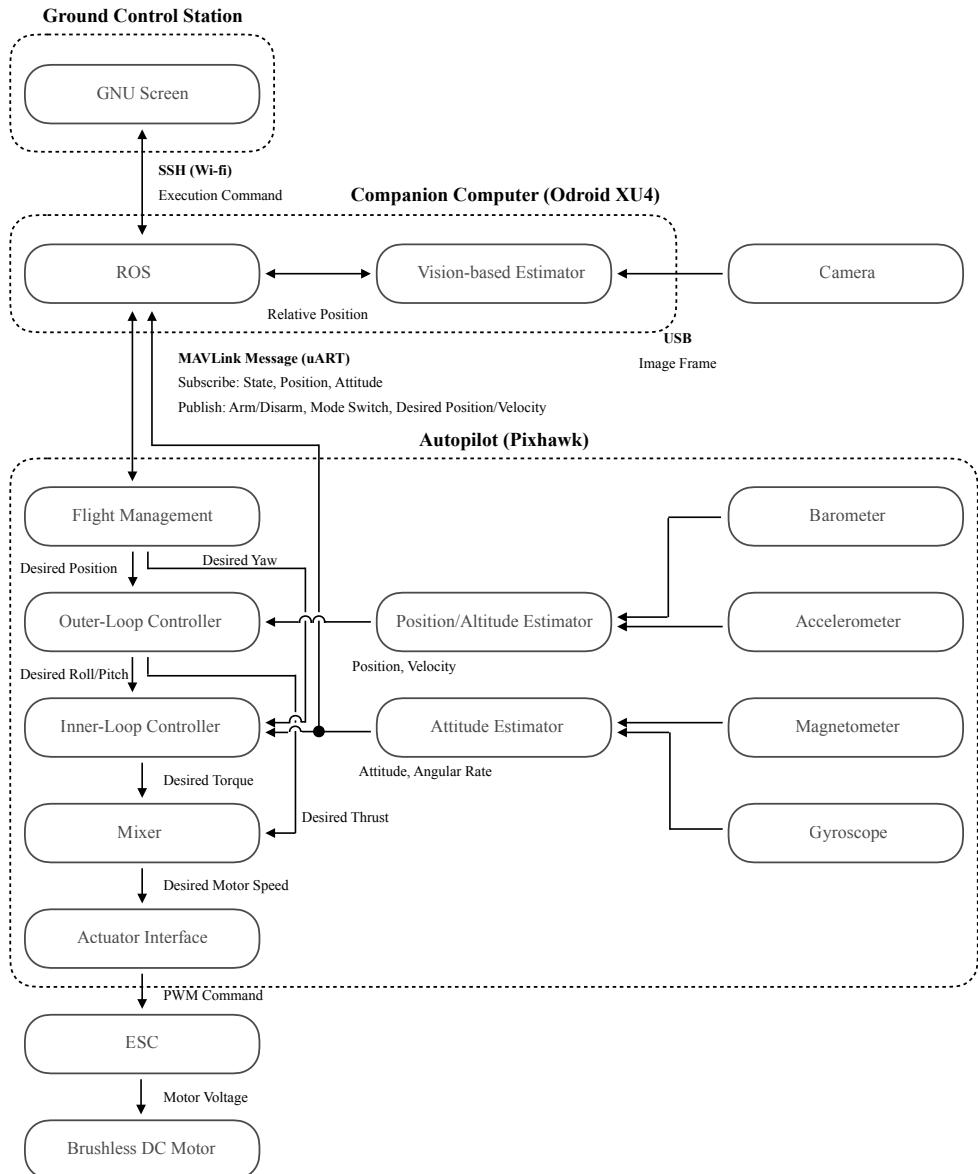


Figure 2.10: System Architecture for Controlling the Quadrotor

CAD model. Due to the convenience of CAD and the difficulties of installing a physical pendulum, we used a CAD model to estimate the moment of inertia. The origin point to compute the moment of inertia is set to be the center of the autopilot controller. Since the quadrotor has a symmetric structure, the moment of inertia J is approximated as a positive diagonal matrix.

$$\begin{aligned} J &= \begin{bmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{xy} & J_{yy} & J_{yz} \\ J_{xz} & J_{yz} & J_{zz} \end{bmatrix} \\ &\approx \begin{bmatrix} 1.64604 & 0 & 0 \\ 0 & 1.602407 & 0 \\ 0 & 0 & 2.940648 \end{bmatrix} \times 10^{-2} \end{aligned} \quad (2.1)$$



Figure 2.11: CAD model of the Quadrotor (upper side)



Figure 2.12: CAD model of the Quadrotor (under side)

Chapter 3

Control System

The control system is one of the most important parts in the quadrotor's operation. In this research, we implemented a customized control system based on the dynamic model of the quadrotor. In this chapter, coordinate systems for the quadrotor's control are defined, the dynamic model of the quadrotor is introduced, and the control system is described.

3.1 Control System Overview

The overview of the control system is illustrated at figure 3.1. First, the outer-loop controller subscribe desired position and desired attitude. Then, the outer-loop controller receives current position data, and computes desired attitude and thrust corresponding to the position error of the quadrotor. Current position is estimated by synchronization of the internal accelerometer and barometer sensors, and other external sensors, such as a camera and a motion capture system. The inner-loop controller receives the attitude and angular velocity of the quadrotor from the onboard gyroscope and magnetometer, and computes desired torque to control the attitude. The quadrotor's current attitude is measured by the internal magnetometer and the angular velocity is measured by the internal gyroscope. Finally, the mixer, an open-loop motor controller, subscribes the desired torque from the outer-loop controller and the desired torque from the inner-loop controller. Then, the mixer maps the desired thrust and attitude into desired motor speeds, and estimate desired voltage of each motor.

3.2 Coordinate System

In order to control the motion and state of the quadrotor, coordinate systems must be defined for the representation of the state and motion.[14] First, a reference coordinate system must be defined to describe the orientation and position of the quadrotor. The reference coordinate system is called an "inertial coordinate system". Commonly, a inertial coordinate system is defined to be a Cartesian coordinate system that has its x, y-plane fixed on the ground. For convenience of control, the z-axis of the quadrotor is defined to face

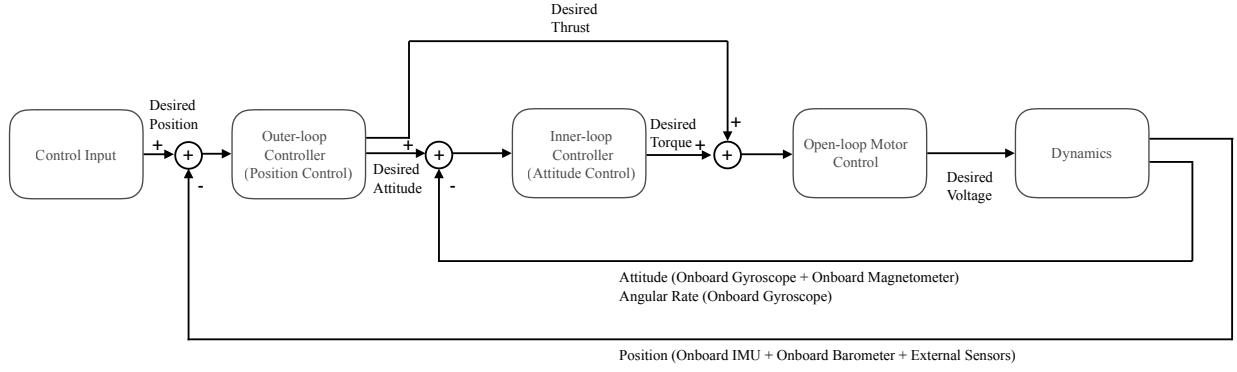


Figure 3.1: The Illustration of the Control System.

the ground. The origin can be arbitrary set, but it is defined to be the initial position of the quadrotor in this research.

In addition, it helps description of the quadrotor's motion to define a body-attached coordinate system since the onboard sensors of the quadrotor is fixed on the quadrotor's frame. In the quadrotor system, the body coordinate system follows the convention; the x, y-axes are defined to be parallel to the quadrotor's frame and the z-axis is defined to be downward. The origin is set to be at the center of mass of the quadrotor.

Using the above coordinate systems, the motion of the quadrotor is represented completely. The mapping of the origin in the body frame into the inertial frame represents the quadrotor's position. The orientation of the body frame can be represented by multiple methods, such as the X-Y-Z fixed angles convention, and the Z-Y-X Euler angles convention. In this research, the Z-Y-X Euler angles convention, also known as the Tait-Bryan angles convention, is used. In the Z-Y-X Euler angle convention, the mapping into the body frame starts with rotating about the z-axis by a yaw angle ψ . Rotation of the y-axis by a pitch angle θ , and the z-axis by a roll angle ϕ is followed in order. The transformation matrix of an orientation from the body frame into the inertial frame is given as,

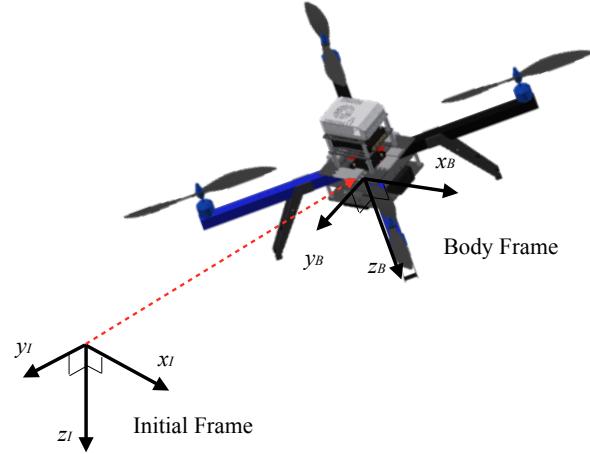


Figure 3.2: The Illustration of the inertial and body frames.

$$\begin{aligned}
 {}^I_B R &= R_\psi R_\theta R_\phi \\
 &= \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \\
 &= \begin{bmatrix} \cos \theta \cos \psi & \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ \cos \theta \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix} \tag{3.1}
 \end{aligned}$$

and the inverse transformation is given as,

$$\begin{aligned}
{}^B_I R &= {}^B_I R^{-1} \\
&= R_\phi^{-1} R_\theta^{-1} R_\psi^{-1} \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \sin \phi \cos \theta \\ \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi & \cos \phi \cos \theta \end{bmatrix}
\end{aligned} \tag{3.2}$$

3.3 Outer-loop Controller

The outer-loop controller calculates desired attitude of the quadrotor is computed based on a given desired position and the quadrotor's current position. As a method of controlling the position of the quadrotor, PID control is applied. In this section, the outer-loop controller will be introduced. First, dynamics with respect to the quadrotor's position is described. Then, the position control of the quadrotor is stated.

3.3.1 Dynamics Description

A quadrotor is accelerated by thrust generated by its four rotors. Let $\mathbf{F} = (\bar{X}, \bar{Y}, \bar{Z})$ to be the quadrotor's thrust on inertial frame. Then, the equation of motion is given as,

$$m\ddot{\mathbf{r}}_I = \mathbf{F} + m\mathbf{g} \tag{3.3}$$

where m is the mass of the quadrotor, $\mathbf{r}_I = (x_I, y_I, z_I)$ is the position of the quadrotor in the inertial frame, and $\mathbf{g} = (0, 0, -g)$ is the gravitational acceleration.

Since the rotation axis of each rotor is fixed perpendicularly to the body, thrust can be generated only in the perpendicular direction of the body frame. Therefore, thrust can be computed by a rotation transformation from body frame into inertial frame. From equation (3.1), thrust vector in the inertial frame

is given as,

$$\begin{aligned}
\mathbf{F} &= {}^I_B R \begin{bmatrix} 0 \\ 0 \\ F \end{bmatrix} \\
&= \begin{bmatrix} \cos \theta \cos \psi & \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ \cos \theta \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ F \end{bmatrix} \\
&= F \begin{bmatrix} \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \\ \cos \phi \cos \theta \end{bmatrix}
\end{aligned} \tag{3.4}$$

where $F = \sqrt{\bar{X}^2 + \bar{Y}^2 + \bar{Z}^2}$ is the magnitude of the thrust and $\boldsymbol{\eta} = (\phi, \theta, \psi)$ is the attitude (roll, pitch, and yaw, respectively) of the quadrotor. Then, the thrust along each axis of the inertial frame is given as,

$$\begin{aligned}
\bar{X} &= F(\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \\
\bar{Y} &= F(\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) \\
\bar{Z} &= F \cos \phi \cos \theta
\end{aligned} \tag{3.5}$$

Therefore, from equations (3.3) and (3.5), the acceleration of the quadrotor $(\ddot{x}_I, \ddot{y}_I, \ddot{z}_I)$ is computed as,

$$\begin{aligned}
\ddot{x}_I &= \frac{F}{m}(\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \\
\ddot{y}_I &= \frac{F}{m}(\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) \\
\ddot{z}_I &= \frac{F}{m} \cos \phi \cos \theta - g
\end{aligned} \tag{3.6}$$

The sensors are fixed on the quadrotor, it is necessary to formulate the relation of the quadrotor's motion in the body frame and the inertial frame. Let $\mathbf{v}_B = (u, v, w)$ be the quadrotor's velocity in the body frame and $\boldsymbol{\omega} = (p, q, r)$ be the quadrotor's angular velocity. From the Coriolis effect, the equation of motion is given with respect to \mathbf{v}_B as,

$$m\ddot{\mathbf{r}}_I = m(\dot{\mathbf{v}}_B + \boldsymbol{\omega} \times \mathbf{v}_B) \tag{3.7}$$

Hence, from equation (3.3) the equation of the quadrotor's thrust \mathbf{F} is written as,

$$\mathbf{F} = m(\dot{\mathbf{v}}_B + \boldsymbol{\omega} \times \mathbf{v}_B - \mathbf{g}) \quad (3.8)$$

3.3.2 Computation of Desired Velocity

In order to control the quadrotor's position, the position controller sets desired velocity $\mathbf{v}_d = (v_{xd}, v_{yd}, v_{zd})$ with respect to the error of desired position $\mathbf{r}_d = (x_{dI}, y_{dI}, z_{dI})$ and observed position $\mathbf{r}_I = (x_I, y_I, z_I)$. The desired velocity is generated by P control. Desired control along each axis of inertial frame is given as,

$$\begin{aligned} v_{dx} &= k_{pos_x}(x_{dI} - x_I) \\ v_{dy} &= k_{pos_y}(y_{dI} - y_I) \\ v_{dz} &= k_{pos_z}(z_{dI} - z_I) \end{aligned} \quad (3.9)$$

where k_{pos_x} , k_{pos_y} , k_{pos_z} are gains of the controller. In order to prevent abnormal performance of the quadrotor, we set upper limits of desired velocity.

3.3.3 Computation of Desired Thrust and Attitude

In order to control the quadrotor's velocity, PID control can be applied to the motion of each axis in the inertial frame. Therefore, desired thrust of PID control is given as,

$$\begin{aligned} \dot{x}_I &= \frac{1}{m} \left(k_{Px}(v_{xd} - \dot{x}_I) + k_{Dx}(\dot{v}_{xd} - \ddot{x}_I) + k_{Ix} \int_{t_0}^t (v_{xd} - \dot{x}_I) dt \right) \\ \ddot{y}_I &= \frac{1}{m} \left(k_{Py}(v_{yd} - \dot{y}_I) + k_{Dy}(\dot{v}_{yd} - \ddot{y}_I) + k_{Iy} \int_{t_0}^t (v_{yd} - \dot{y}_I) dt \right) \\ \ddot{z}_I &= \frac{1}{m} \left(k_{Pz}(v_{zd} - \dot{z}_I) + k_{Dz}(\dot{v}_{zd} - \ddot{z}_I) + k_{Iz} \int_{t_0}^t (v_{zd} - \dot{z}_I) dt \right) \end{aligned} \quad (3.10)$$

where k_{Px} , k_{Py} , k_{Pz} are gains of the proportional terms, k_{Dx} , k_{Dy} , k_{Dz} are gains of the differential terms, and k_{Ix} , k_{Iy} , k_{Iz} are gains of the integral terms. Then, from equation (3.3), the desired thrust \overline{X}_d , \overline{Y}_d , \overline{Z}_d can be computed as,

$$\begin{aligned} \overline{X}_d &= k_{Px}(v_{xd} - \dot{x}_I) + k_{Dx}(\dot{v}_{xd} - \ddot{x}_I) + k_{Ix} \int_{t_0}^t (v_{xd} - \dot{x}_I) dt \\ \overline{Y}_d &= k_{Py}(v_{yd} - \dot{y}_I) + k_{Dy}(\dot{v}_{yd} - \ddot{y}_I) + k_{Iy} \int_{t_0}^t (v_{yd} - \dot{y}_I) dt \\ \overline{Z}_d &= mg + k_{Pz}(v_{zd} - \dot{z}_I) + k_{Dz}(\dot{v}_{zd} - \ddot{z}_I) + k_{Iz} \int_{t_0}^t (v_{zd} - \dot{z}_I) dt \end{aligned} \quad (3.11)$$

In this research, the gains of differential terms are set small so that the effect of the differential terms is comparatively ignorable. Since the desired velocity is set proportionally with the error of the quadrotor's position, the control law becomes similar to PD control in terms of the quadrotor's position.

As described in the above paragraph, the direction of the thrust is defined by the orientation of the quadrotor. From equations (3.5) and (3.11), with given yaw angle ψ , desired roll ϕ_d and pitch θ_d are computed as,

$$\begin{aligned}\phi_d &= \arcsin \frac{\bar{X}_d \sin \psi - \bar{Y}_d \cos \psi}{F_d} \\ \theta_d &= \arctan \frac{\bar{X}_d \cos \psi + \bar{Y}_d \sin \psi}{\bar{Z}_d}\end{aligned}\quad (3.12)$$

Also, the magnitude of desired thrust F_d is given as,

$$F_d = \sqrt{\bar{X}_d^2 + \bar{Y}_d^2 + \bar{Z}_d^2} \quad (3.13)$$

This implies that the position of the quadrotor is controllable with any yaw angle. Therefore, yaw angle is also controllable while holding the quadrotor's position.

3.4 Inner-loop Controller

The outer-loop controller computes desired attitude of the quadrotor based on position control. The inner-loop controller computes the torque required to control the quadrotor's attitude. PID control is often used as a method of attitude control. However, in this research, a more advanced control is applied for the inner-loop. In this section, an advanced inner-loop controller will be introduced. Firstly, dynamics with regard to the quadrotor's attitude is described. Then, a process of computing desired angular rate and torque of the quadrotor is explained. Finally, a proof of the controller's stability is stated.

3.4.1 Dynamics Description

Let $\mathbf{u} = (u_p, u_q, u_r)$ the torque in the body frame. Then, from the effect of rotation, the equation of moment is given with respect with the angular velocity of the quadrotor $\boldsymbol{\omega} = (p, q, r)$ as,

$$\mathbf{u} = \frac{d}{dt}(J\boldsymbol{\omega}) = J\dot{\boldsymbol{\omega}} - \text{Skew}(J\boldsymbol{\omega})\boldsymbol{\omega} \quad (3.14)$$

where J is the inertia matrix of the quadrotor and $\text{Skew}(\mathbf{a})$ is a skew-symmetric matrix,

$$\text{Skew}(\mathbf{a}) = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \quad (3.15)$$

By considering the symmetry of the quadrotor along each axis of inertial frame, the diagonal elements J_{xx} , J_{yy} , J_{zz} are much greater than the other elements and the non-diagonal elements are negligible. Therefore, the inertia matrix J can be approximated as a positive diagonal matrix.

$$J = \begin{bmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{xy} & J_{yy} & J_{yz} \\ J_{xz} & J_{yz} & J_{zz} \end{bmatrix} \approx \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} \quad (3.16)$$

Since the attitude $\boldsymbol{\eta} = (\phi, \theta, \psi)$ of the quadrotor is on the body frame, there is a relation between the angular velocity $\boldsymbol{\omega} = (p, q, r)$ and the derivative $\dot{\boldsymbol{\eta}} = (\dot{\phi}, \dot{\theta}, \dot{\psi})$ of the attitude, given by,

$$\begin{aligned} \begin{bmatrix} p \\ q \\ r \end{bmatrix} &= \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + R_\phi^{-1} \left(\begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R_\theta^{-1} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \right) \\ &= \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \left(\begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \right) \\ &= \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \end{aligned} \quad (3.17)$$

Define a transform matrix Z from the body frame into the world frame with respect to the attitude of the quadrotor's attitude $\boldsymbol{\eta}$ as,

$$Z = Z(\boldsymbol{\eta}) = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix}^{-1} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \quad (3.18)$$

Then, from the transformation of coordinate systems, the derivative of the attitude $\dot{\boldsymbol{\eta}}$ can be written as,

$$\dot{\boldsymbol{\eta}} = Z\boldsymbol{\omega} \quad (3.19)$$

Since the transformation matrix Z always has an inverse matrix regardless of the quadrotor's attitude $\boldsymbol{\eta}$, the inverse form of the above equation is also satisfied. Therefore, the angular velocity is represented with the derivative of the attitude $\dot{\boldsymbol{\eta}}$ and the transform matrix Z as the below equation.

$$\boldsymbol{\omega} = Z^{-1}\dot{\boldsymbol{\eta}} \quad (3.20)$$

3.4.2 Computation of Reference Angular Velocity

The inner-loop controller subscribes desired attitude $\boldsymbol{\eta}_d = (\phi_d, \theta_d, \psi_d)$ from the outer-loop controller, and desired angular velocity $\boldsymbol{\omega}_d$ is computed from it. Desired angular velocity is given from the controller as,

$$\boldsymbol{\omega}_d = Z^{-1}\dot{\boldsymbol{\eta}}_d \quad (3.21)$$

With the value of the desired attitude $\boldsymbol{\eta}_d$, there are multiple methods to compute the derivative of the desired attitude $\dot{\boldsymbol{\eta}}_d$.

First, $\dot{\boldsymbol{\eta}}_d = (\dot{\phi}_d, \dot{\theta}_d, \dot{\psi}_d)$ is computed by direct differentiation of the desired attitude $\dot{\boldsymbol{\eta}}$ defined at equation (3.12). From equation (3.12), $\dot{\boldsymbol{\eta}}_d$ is computed as,

$$\begin{aligned} \dot{\phi}_d &= \frac{1}{\sqrt{1 - c_1^2}} \left(c_2 \dot{\psi} + \frac{\sin \psi}{F_d} \dot{\bar{X}}_d - \frac{\cos \psi}{F_d} \dot{\bar{X}}_d - \frac{\dot{F}_d c_1}{F_d^2} \right) \\ \dot{\theta}_d &= \frac{1}{1 + c_2^2} \left(-c_1 \dot{\psi} + \frac{\cos \psi}{\bar{Z}_d} \dot{\bar{X}}_d + \frac{\sin \psi}{\bar{Z}_d} \dot{\bar{X}}_d + -\frac{\dot{\bar{Z}}_d c_2}{\bar{Z}_d^2} \right) \end{aligned} \quad (3.22)$$

where c_1 and c_2 are defines as,

$$\begin{aligned} c_1 &= \frac{\bar{X}_d \sin \psi - \bar{Y}_d \cos \psi}{F_d} \\ c_2 &= \frac{\bar{X}_d \cos \psi + \bar{Y}_d \sin \psi}{\bar{Z}_d} \end{aligned} \quad (3.23)$$

From geometric relation, F_d is also written as,

$$F_d = \bar{Z}_d \sec \theta \sec \psi \quad (3.24)$$

Therefore, the roll rate $\dot{\phi}_d$ is decoupled with F_d , and represented with respect only with the attitude $\boldsymbol{\eta}$ and the desired thrust $(\bar{X}_d, \bar{Y}_d, \bar{Z}_d)$ as below equation.

$$\dot{\phi}_d = \frac{1}{\sqrt{1 - c_1^2}} \left(c_2 \dot{\psi} + \frac{\sin \psi}{F_d} \dot{\bar{X}}_d - \frac{\cos \psi}{F_d} \dot{\bar{Z}}_d + \frac{c_1}{F_d^2} \left(\bar{Z}_d (\dot{\phi} \sec \phi \tan \phi + \dot{\theta} \sec \theta \tan \theta) - \dot{\bar{Z}}_d \right) \right) \quad (3.25)$$

Usually, yaw is controlled as constant, or yaw rate $\dot{\psi}_d$ is given.

However, in the case that the quadrotor does not have precise measurement of its position or the attitude of the quadrotor is controlled manually, the above formulation may not be available. Therefore, an alternative computation method of desired angular velocity using a backward differentiation approximation can be applied. Numerically differentiated values usually have noise and therefore, a low-pass filter is used to reduce the noise. In discrete-time, let t_k be a discrete time and $\Delta t_k = t_k - t_{k-1}$ be a time step. Then, a first-ordered low-pass filter, also known as exponentially weighted moving average filter is given as the below equation,

$$y(t_k) = (1 - a_k)u(t_k) - a_k y(t_{k-1}) \quad (3.26)$$

where the filter parameter a is defined with respect to the filter constant T_f as,

$$a_k = \frac{\Delta t_k}{T_f + \Delta t_k} \quad (3.27)$$

In the above equation, the contribution of new input $u(t_k)$ decreases and become less noisy as the filter constant T_f become greater. From the first-ordered differentiation approximation for the time derivative, raw derivative of the desired attitude is computed as the below equation.

$$\dot{\boldsymbol{\eta}}_{d_{raw}}(t_k) = \frac{\boldsymbol{\eta}_d(t_k) - \boldsymbol{\eta}_d(t_{k-1})}{\Delta t_k} \quad (3.28)$$

From equations (3.26) and (3.28), the filtered derivative of the desired attitude is computed as the below equation.

$$\dot{\boldsymbol{\eta}}_d(t_k) = (1 - a_k)\dot{\boldsymbol{\eta}}_{d_{raw}}(t_k) - a_k \dot{\boldsymbol{\eta}}_d(t_{k-1}) \quad (3.29)$$

3.4.3 Computation of Desired Torque

In this research, the nonlinear attitude control developed by Bandyopadhyay and Chung is used.[12] In the control law, previous to the computation of desired torque, it is necessary to define reference angular velocity. Let $\boldsymbol{\eta} = (\phi, \theta, \psi)$ to be actual attitude of the quadrotor. Define reference attitude $\dot{\boldsymbol{\eta}}_r$ as,

$$\dot{\boldsymbol{\eta}}_r = \dot{\boldsymbol{\eta}}_d + \Lambda(\boldsymbol{\eta}_d - \boldsymbol{\eta}) \quad (3.30)$$

where Λ is a positive gain matrix as,

$$\Lambda = \begin{bmatrix} \lambda_\phi & 0 & 0 \\ 0 & \lambda_\theta & 0 \\ 0 & 0 & \lambda_\psi \end{bmatrix} > 0 \quad (3.31)$$

Reference angular velocity $\boldsymbol{\omega}_r$ and its derivative $\dot{\boldsymbol{\omega}}_r$ are given as,

$$\begin{aligned} \boldsymbol{\omega}_r &= Z^{-1}\dot{\boldsymbol{\eta}}_d + Z^{-1}\Lambda(\boldsymbol{\eta}_d - \boldsymbol{\eta}) \\ &= \boldsymbol{\omega}_d + Z^{-1}\Lambda(\boldsymbol{\eta}_d - \boldsymbol{\eta}) \end{aligned} \quad (3.32)$$

$$\begin{aligned} \dot{\boldsymbol{\omega}}_r &= \dot{\boldsymbol{\omega}}_d + Z^{-1}\Lambda(\dot{\boldsymbol{\eta}}_d - \dot{\boldsymbol{\eta}}) - Z^{-1}\dot{Z}Z^{-1}\Lambda(\boldsymbol{\eta}_d - \boldsymbol{\eta}) \\ &= \dot{\boldsymbol{\omega}}_d + \Lambda((\boldsymbol{\omega}_d - \boldsymbol{\omega}) - Z^{-1}\dot{Z}Z^{-1}(\boldsymbol{\eta}_d - \boldsymbol{\eta})) \end{aligned} \quad (3.33)$$

Since the matrix Z is an unitary matrix, $\dot{\boldsymbol{\omega}}_r$ can be simplified as,

$$\dot{\boldsymbol{\omega}}_r = \dot{\boldsymbol{\omega}}_d + \Lambda(\boldsymbol{\omega}_d - \boldsymbol{\omega}) - Z^{-1}\dot{Z}Z^{-1}\Lambda(\boldsymbol{\eta}_d - \boldsymbol{\eta}) \quad (3.34)$$

For the implementation of the attitude control, we assume that $\ddot{\boldsymbol{\omega}}_d = 0$, and this is equivalent to $\dot{\boldsymbol{\omega}}_d = 0$. The current attitude of the quadrotor $\boldsymbol{\eta}$ is measured by the internal magnetometer, and its derivative $\dot{\boldsymbol{\eta}}$ is measured by the internal gyroscope.

According to the nonlinear attitude controller developed by Bandyopadhyay and Chung, the control law is given by the below equation.[12]

$$\mathbf{u} = J\dot{\boldsymbol{\omega}}_r - \text{Skew}(J\boldsymbol{\omega})\boldsymbol{\omega}_r + K_\omega(\boldsymbol{\omega}_r - \boldsymbol{\omega}) \quad (3.35)$$

where is K_ω a positive gain matrix,

$$K_\omega = \begin{bmatrix} k_{\omega_p} & 0 & 0 \\ 0 & k_{\omega_q} & 0 \\ 0 & 0 & k_{\omega_r} \end{bmatrix} > 0 \quad (3.36)$$

In order to prove the stability of the nonlinear attitude controller, the Lyapunov method is used. From equation (3.19), (3.35), the below equations are obtained.

$$\boldsymbol{\omega}_r - \boldsymbol{\omega} = \boldsymbol{\omega}_d - \boldsymbol{\omega} + Z^{-1}\Lambda(\boldsymbol{\eta}_d - \boldsymbol{\eta}) \quad (3.37)$$

$$(\dot{\boldsymbol{\eta}}_d - \dot{\boldsymbol{\eta}}) - Z(\boldsymbol{\omega}_r - \boldsymbol{\omega}) + \Lambda(\boldsymbol{\eta}_d - \boldsymbol{\eta}) = 0 \quad (3.38)$$

From the equation (3.14) and (3.35), the below equations can be obtained.

$$J\dot{\boldsymbol{\omega}} - \text{Skew}(J\boldsymbol{\omega})\boldsymbol{\omega} = J\dot{\boldsymbol{\omega}}_r - \text{Skew}(J\boldsymbol{\omega})\boldsymbol{\omega}_r + K_\omega(\boldsymbol{\omega}_r - \boldsymbol{\omega}) \quad (3.39)$$

$$J(\dot{\boldsymbol{\omega}}_r - \dot{\boldsymbol{\omega}}) + (K_\omega - \text{Skew}(J\boldsymbol{\omega}))(\boldsymbol{\omega}_r - \boldsymbol{\omega}) = 0 \quad (3.40)$$

Let $\boldsymbol{s}_1 = \boldsymbol{\omega}_r - \boldsymbol{\omega}$ and $\boldsymbol{s}_2 = Z^{-1}(\boldsymbol{\eta}_d - \boldsymbol{\eta})$. Then, the above equations can be written as,

$$\begin{bmatrix} J & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{s}}_1 \\ \dot{\boldsymbol{s}}_2 \end{bmatrix} + \begin{bmatrix} \text{Skew}(J\boldsymbol{\omega}) + K_\omega & 0 \\ -Z & \Lambda \end{bmatrix} \begin{bmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (3.41)$$

where I is the 3×3 identity matrix. Define a Lyapunov function V as below,

$$V(\boldsymbol{s}_1, \boldsymbol{s}_2) = \begin{bmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \end{bmatrix}^T \begin{bmatrix} J & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \end{bmatrix} \quad (3.42)$$

Since the inertia matrix J , and the identity matrix I is positive definite, the Lyapunov function $V(\boldsymbol{s}_1, \boldsymbol{s}_2)$ is positive, if $\boldsymbol{s}_1, \boldsymbol{s}_2$ are not zeros.

$$V(\boldsymbol{s}_1, \boldsymbol{s}_2) = \boldsymbol{s}_1^T J \boldsymbol{s}_1 + \boldsymbol{s}_2^T I \boldsymbol{s}_2 > 0 \quad (3.43)$$

Derivative $\dot{V}(\boldsymbol{s}_1, \boldsymbol{s}_2)$ of the Lyapunov function is given below, and from the definition of K_ω and Λ , the derivative of the Lypunov function is negative definite, if $\boldsymbol{s}_1, \boldsymbol{s}_2$ are not zeros.

$$\begin{aligned}
\dot{V}(\mathbf{s}_1, \mathbf{s}_2) &= \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix}^T \begin{bmatrix} \dot{\mathbf{J}} & 0 \\ 0 & \dot{\mathbf{I}} \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix} + 2 \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix}^T \begin{bmatrix} \mathbf{J} & 0 \\ 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{s}}_1 \\ \dot{\mathbf{s}}_2 \end{bmatrix} \\
&= 2 \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix}^T \begin{bmatrix} \mathbf{J} & 0 \\ 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix}^T \begin{bmatrix} 2\text{Skew}(\mathbf{J}\boldsymbol{\omega}) - 2K_\omega & 0 \\ 2Z & -2\Lambda \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix} \\
&= 2\mathbf{s}_1^T \text{Skew}(\mathbf{J}\boldsymbol{\omega})\mathbf{s}_1 - 2\mathbf{s}_1^T K_\omega \mathbf{s}_1 - 2\mathbf{s}_2^T \Lambda \mathbf{s}_2 \\
&= -2\mathbf{s}_1^T K_\omega \mathbf{s}_1 - 2\mathbf{s}_2^T \Lambda \mathbf{s}_2 \\
&> 0
\end{aligned} \tag{3.44}$$

From the definition of K_ω and Λ , $\dot{V}(\mathbf{s}_1, \mathbf{s}_2)$ is negative, if $\mathbf{s}_1, \mathbf{s}_2$ are not zeros.

$$\begin{aligned}
\dot{V}(\mathbf{s}_1, \mathbf{s}_2) &= \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix}^T \begin{bmatrix} 2\text{Skew}(IZ^{-1}\dot{\boldsymbol{\eta}}) - 2K_\omega & 0 \\ 2I & -2\Lambda \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix} \\
&= 2\mathbf{s}_1^T \text{Skew}(IZ^{-1}\dot{\boldsymbol{\eta}})\mathbf{s}_1 - 2\mathbf{s}_1^T K_\omega \mathbf{s}_1 - 2\mathbf{s}_2^T \Lambda \mathbf{s}_2 \\
&= -2\mathbf{s}_1^T K_\omega \mathbf{s}_1 - 2\mathbf{s}_2^T \Lambda \mathbf{s}_2 \\
&< 0
\end{aligned} \tag{3.45}$$

Hence, by the Lyapunov 2nd method, the controller is globally asymptotic stable with an equilibrium point where $\mathbf{s}_1 = 0$ $\mathbf{s}_2 = 0$. Therefore, the quadrotor's attitude $\boldsymbol{\eta}$ and angular velocity $\boldsymbol{\omega}$ converge to $\boldsymbol{\eta}_d$ and $\boldsymbol{\omega}_d$, respectively.

Further, define $c_1 = \min(J_{xx}, J_{yy}, J_{zz}, 1)$ and $c_2 = \max(J_{xx}, J_{yy}, J_{zz}, 1)$, and then, the below condition of the Lyapunov function $V(\mathbf{s}_1, \mathbf{s}_2)$ is satisfied.

$$\forall \mathbf{s}_1, \mathbf{s}_2 \neq \mathbf{0}, \quad c_2(\|\mathbf{s}_1\|^2 + \|\mathbf{s}_2\|^2) > V(\mathbf{s}_1, \mathbf{s}_2) \approx \mathbf{s}_1^T \mathbf{J} \mathbf{s}_1 + \mathbf{s}_2^T \mathbf{I} \mathbf{s}_2 > c_1(\|\mathbf{s}_1\|^2 + \|\mathbf{s}_2\|^2) > 0 \tag{3.46}$$

Also, define $c_3 = 2 \min(k_{\omega_p}, k_{\omega_q}, k_{\omega_r}, \lambda_\phi, \lambda_\theta, \lambda_\psi)$ and then, the below condition of $\dot{V}(\mathbf{s}_1, \mathbf{s}_2)$ is satisfied.

$$\forall \mathbf{s}_1, \mathbf{s}_2 \neq \mathbf{0}, \quad \dot{V}(\mathbf{s}_1, \mathbf{s}_2) \leq -c_3(\|\mathbf{s}_1\|^2 + \|\mathbf{s}_2\|^2) < 0 \tag{3.47}$$

From Lyapunov's 1st method, the controller is globally exponentially stable as well.

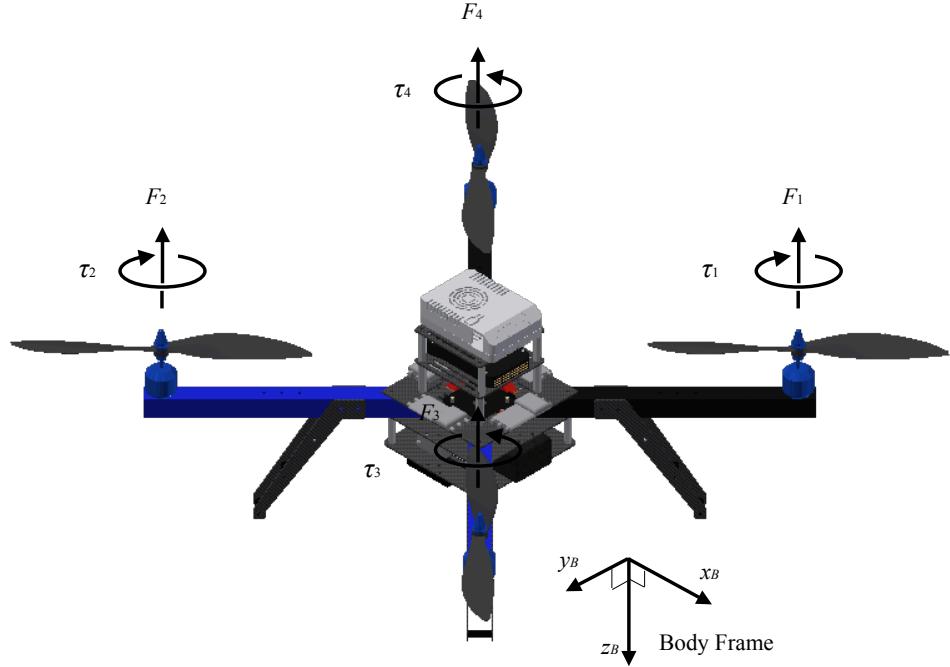


Figure 3.3: The Thrust and Torque Generated by Each Rotor

3.5 Motor Control

As described in the section 3, the desired torque of the quadrotor is computed by the inner-loop controller. The next step is to compute the desired motor speed corresponding to the torque, and to control the motors. In the quadrotor system, the part that performs this step is called the "mixer". In this section, an advanced mixer for the quadrotor will be introduced. First, an aerodynamic model with respect to rotors is described, and a mapping method from attitude and thrust control into motor speed is explained. Then, dynamic model of a brushless DC motor for the quadrotor and its control is stated.

3.5.1 Mapping Attitude and Thrust Control into Motor Speeds

In the mixer, the magnitude of desired thrust F_d from the outer-loop and desired attitude control $\tau = (\tau_\phi, \tau_\theta, \tau_\psi)$ from the inner-loop is subscribed. In order to map attitude control and desired thrust into motor speed $\mathbf{n} = (n_1, n_2, n_3, n_4)$, aerodynamics model of the quadrotor can be applied.

It is known that the property of propeller is given with non-dimensional thrust coefficient C_T and power coefficient C_P as,

$$C_T(n_i) = \frac{T}{\rho n_i^2 D^4} \quad (3.48)$$

$$C_P(n_i) = \frac{P}{\rho n_i^3 D^5} \quad (3.49)$$

where ρ is the density of air, D is the propeller's diameter, and T, P are thrust and torque generated by a propeller, respectively. Here, C_T and C_P change according to rotation speed, and their curves are varied across different geometry of propellers and other conditions. Let F_i be the thrust force of each motor ($i = 1, 2, 3, 4$) corresponding to motor speed n_i . With the experimental data of C_T and C_P , thrust F_i and torque τ_i generated by a motor according to its motor speed n_i are given as below,

$$F_i = C_T(n_i) \rho n_i^2 D^4 \quad (3.50)$$

$$\begin{aligned} \tau_i &= \frac{P}{2\pi n_i} \\ &= \frac{C_P(n_i) \rho n_i^2 D^5}{2\pi} \end{aligned} \quad (3.51)$$

Since the quadrotor uses a X-shape frame in which the formation of the motors is a square, there is a linear relation as,

$$\begin{aligned} F &= F_1 + F_2 + F_3 + F_4 \\ \tau_\phi &= \frac{l}{\sqrt{2}}(-F_1 + F_2 + F_3 - F_4) \\ \tau_\theta &= \frac{l}{\sqrt{2}}(F_1 - F_2 + F_3 - F_4) \\ \tau_\psi &= \tau_1 + \tau_2 - \tau_3 - \tau_4 \end{aligned} \quad (3.52)$$

where l to the length of the frame. We define an aerodynamics coefficient matrix $B(\mathbf{n})$ as,

$$B(\mathbf{n}) = \begin{bmatrix} C_T(n_1) & C_T(n_2) & C_T(n_3) & C_T(n_4) \\ -\frac{l}{\sqrt{2}}C_T(n_1) & \frac{l}{\sqrt{2}}C_T(n_2) & \frac{l}{\sqrt{2}}C_T(n_3) & -\frac{l}{\sqrt{2}}C_T(n_4) \\ \frac{l}{\sqrt{2}}C_T(n_1) & -\frac{l}{\sqrt{2}}C_T(n_2) & \frac{l}{\sqrt{2}}C_T(n_3) & -\frac{l}{\sqrt{2}}C_T(n_4) \\ \frac{D}{2\pi}C_P(n_1) & \frac{D}{2\pi}C_P(n_2) & -\frac{D}{2\pi}C_P(n_3) & -\frac{D}{2\pi}C_P(n_4) \end{bmatrix} \quad (3.53)$$

Then, the above equation can be written as,

$$\begin{bmatrix} F_d \\ u_p \\ u_q \\ u_r \end{bmatrix} = \rho D^4 B(\mathbf{n}) \begin{bmatrix} n_1^2 \\ n_2^2 \\ n_3^2 \\ n_4^2 \end{bmatrix} \quad (3.54)$$

With the above equation, motor speed \mathbf{n} corresponding to the magnitude of the desired thrust F_d and the attitude control τ can be computed as,

$$\begin{bmatrix} n_1^2 \\ n_2^2 \\ n_3^2 \\ n_4^2 \end{bmatrix} = \frac{1}{\rho D^4} B^{-1}(\mathbf{n}) \begin{bmatrix} F_d \\ u_p \\ u_q \\ u_r \end{bmatrix} \quad (3.55)$$

3.5.2 Approximation of the Aerodynamic Coefficient Matrix

Since the coefficient matrix $B(\mathbf{n})$ is not independent from motor speed n_i , it is difficult to compute a solution of equation (3.55). To solve the problem, the linear approximation in terms of $\frac{1}{n_i^2}$ as,

$$\begin{aligned} C_T(n_i) &= C_{T0} + C_{T1} \frac{1}{n_i^2} + O\left(\frac{1}{n_i^4}\right) \\ &\approx C_{T0} + C_{T1} \frac{1}{n_i^2} \end{aligned} \quad (3.56)$$

$$\begin{aligned} C_P(n_i) &= C_{P0} + C_{P1} \frac{1}{n_i^2} + O\left(\frac{1}{n_i^4}\right) \\ &\approx C_{P0} + C_{P1} \frac{1}{n_i^2} \end{aligned} \quad (3.57)$$

can be applied. Then, equation (3.54) can be written as,

$$\begin{bmatrix} F_d \\ u_p \\ u_q \\ u_r \end{bmatrix} \approx \rho D^4 \left(B_0 \begin{bmatrix} n_1^2 \\ n_2^2 \\ n_3^2 \\ n_4^2 \end{bmatrix} + B_1 \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \right) \quad (3.58)$$

where the approximation terms of B_0 , B_1 of the matrix $B(\mathbf{n})$ are defined as,

$$B_0 = \begin{bmatrix} C_{T0} & C_{T0} & C_{T0} & C_{T0} \\ -\frac{l}{\sqrt{2}}C_{T0} & \frac{l}{\sqrt{2}}C_{T0} & \frac{l}{\sqrt{2}}C_{T0} & -\frac{l}{\sqrt{2}}C_{T0} \\ \frac{l}{\sqrt{2}}C_{T0} & -\frac{l}{\sqrt{2}}C_{T0} & \frac{l}{\sqrt{2}}C_{T0} & -\frac{l}{\sqrt{2}}C_{T0} \\ \frac{D}{2\pi}C_{P0} & \frac{D}{2\pi}C_{P0} & -\frac{D}{2\pi}C_{P0} & -\frac{D}{2\pi}C_{P0} \end{bmatrix} \quad (3.59)$$

$$B_1 = \begin{bmatrix} C_{T1} & C_{T1} & C_{T1} & C_{T1} \\ -\frac{l}{\sqrt{2}}C_{T1} & \frac{l}{\sqrt{2}}C_{T1} & \frac{l}{\sqrt{2}}C_{T1} & -\frac{l}{\sqrt{2}}C_{T1} \\ \frac{l}{\sqrt{2}}C_{T1} & -\frac{l}{\sqrt{2}}C_{T1} & \frac{l}{\sqrt{2}}C_{T1} & -\frac{l}{\sqrt{2}}C_{T1} \\ \frac{D}{2\pi}C_{P1} & \frac{D}{2\pi}C_{P1} & -\frac{D}{2\pi}C_{P1} & -\frac{D}{2\pi}C_{P1} \end{bmatrix} \quad (3.60)$$

Then, desired motor speeds \mathbf{n} are easily computed from the below equation,

$$\begin{bmatrix} n_1^2 \\ n_2^2 \\ n_3^2 \\ n_4^2 \end{bmatrix} \approx B_0^{-1} \left(\frac{1}{\rho D^4} \begin{bmatrix} F_d \\ u_p \\ u_q \\ u_r \end{bmatrix} - B_1 \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \right) \quad (3.61)$$

To compute the model of thrust coefficient (3.56) and power coefficient (3.57), the experimental data of the propeller APC 10x3.7 in the UIUC Propeller Database Vol.1 are used, and the method of least mean square error (LMSE) linear regression was applied in terms of $\frac{1}{n_i^2}$.[17] The approximated models of $C_T(n_i)$ and $C_P(n_i)$ are given as the equations(3.62) and (3.63), and the experimental data and the approximated models are compared in the Figures 3.4, 3.5.

$$C_T(n_i) = 0.1627 - 73047 \times \frac{1}{n_i^2} \quad (3.62)$$

$$C_P(n_i) = 0.0791 - 49112 \times \frac{1}{n_i^2} \quad (3.63)$$

3.6 Control of Motors

In the quadrotor system, propellers are rotated by four brushless DC motors. In order to control DC motors, the method of pulse width modulation (PWM) is often used. In the PWM method, averages of constant-voltage pulsing signals actuates a motor. Therefore, motor voltage is controllable by changing a pulse signal's width. Let d_{pwm} to be duty cycle as,

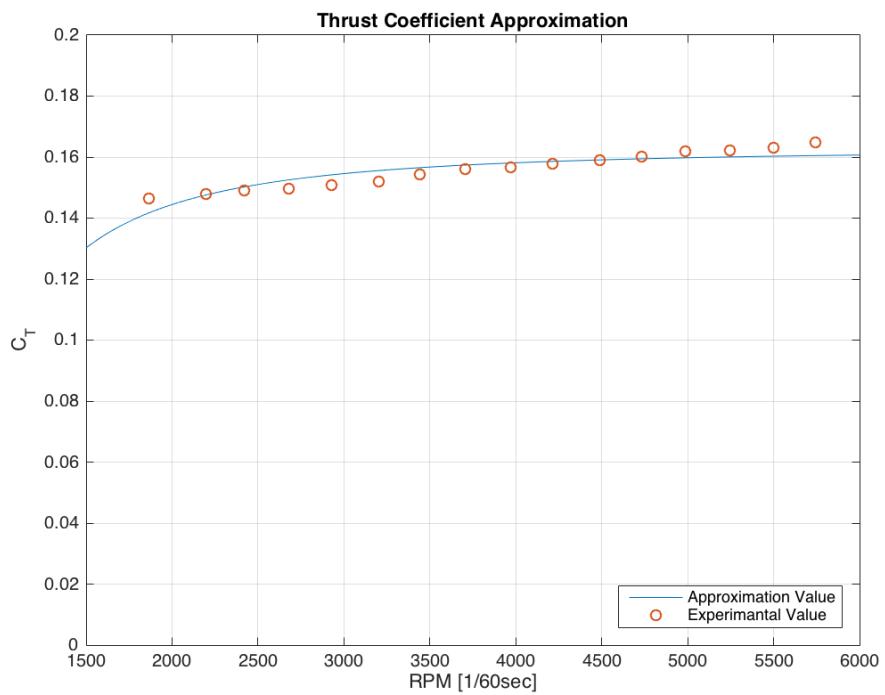


Figure 3.4: Comparison of experimental and approximated thrust coefficient

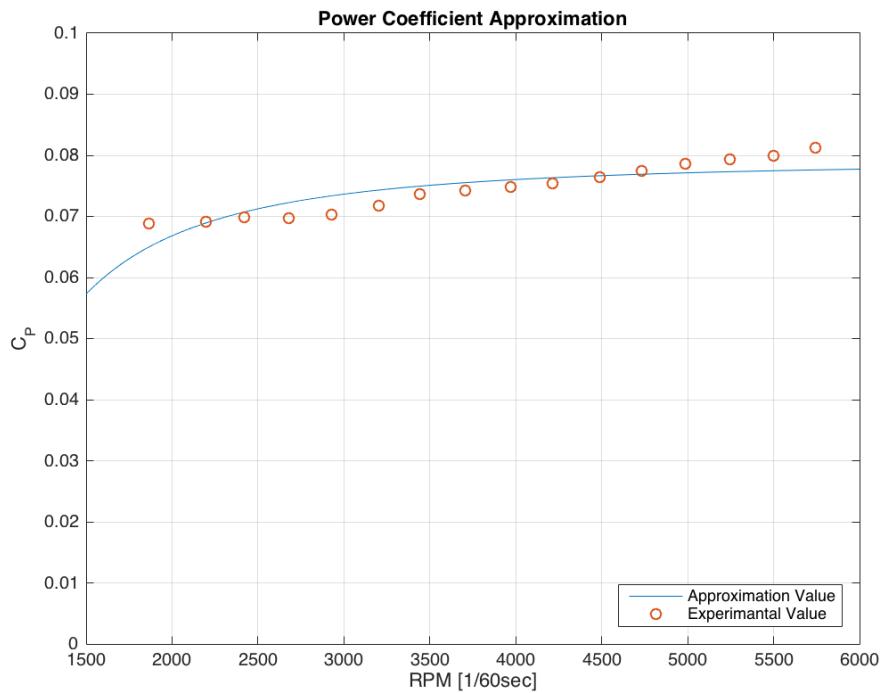


Figure 3.5: Comparison of experimental and approximated power coefficient

$$d_{pwm} = H f_{pwm} : \quad 0 \leq d_{pwm} \leq 1 \quad (3.64)$$

where H is signal width and f_{pwm} is the fixed frequency of signals. Then, the relation between motor voltage V_{mot} and duty cycle pwm is formalized as,

$$V_{mot} = d_{pwm} V_0 \quad (3.65)$$

where V_0 is the constant voltage of signals.

The electric and dynamic models of DC models are given in terms of motor speed n and V_{mot} as,

$$V_{mot} = 2\pi k_b n + Ri + L \frac{di}{dt} \quad (3.66)$$

$$k_m i = 2\pi k_f n + 2\pi I_{mot} \frac{dn}{dt} + M_R \text{sgn}(n) \quad (3.67)$$

where k_b , k_f , k_m are the motor's torque constant, back electromotive force constant, and viscous damping coefficient, respectively. M_R is frictional torque, and R is the motor's internal resistor and L is the motor's internal inductance. I_{mot} is the total inertia of the motor and the propeller. Then, the above equation is written as the following second-order system with respect to motor speed n .

$$V_{mot} = 2\pi \left((k_b + \frac{Rk_f}{k_m})n + \frac{1}{k_m} (RI_{mot} + Lk_f) \frac{dn}{dt} + \frac{LI_{mot}}{k_m} \frac{d^2n}{dt^2} \right) + RM_R \text{sgn}(n) : \quad n \neq 0 \quad (3.68)$$

Also, from equation (3.70), there is a voltage threshold due to frictional torque so that a motor doesn't spin below the threshold.

Assuming motor speed changes smoothly so that the effect of internal resistor and inductance is small enough, equation (3.68) can be simplified as,

$$V_{mot} \approx 2\pi(k_b + \frac{Rk_f}{k_m})n + RM_R \text{sgn}(n) \quad (3.69)$$

Define composite coefficients of the tangent α , and the intercept β as,

$$\alpha = \frac{k_m V_0}{2\pi(k_b k_m + R k_f)} \quad (3.70)$$

$$\beta = \frac{k_m R M_R}{2\pi(k_b k_m + R k_f)} \quad (3.71)$$

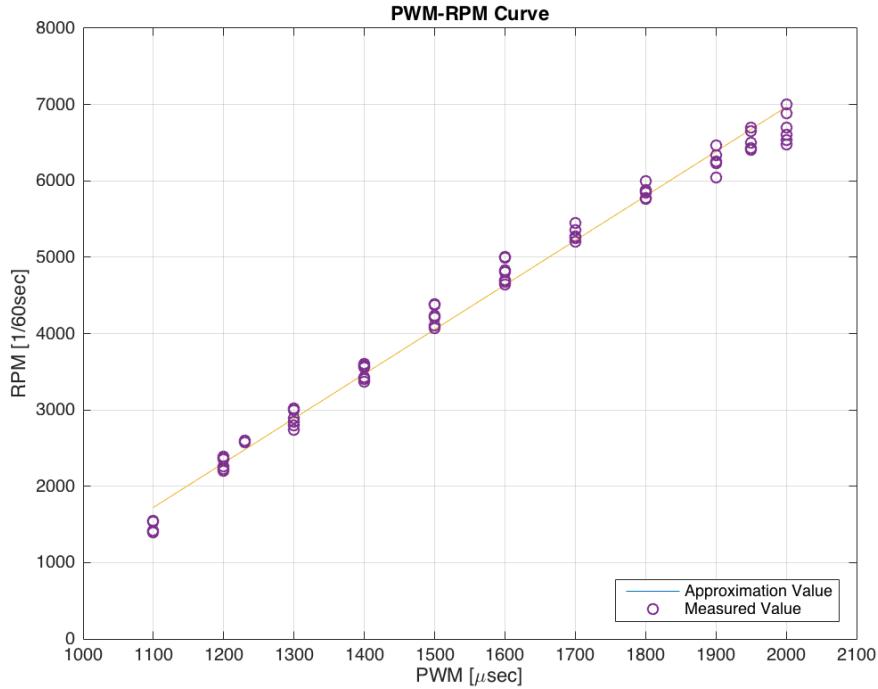


Figure 3.6: Change of RPM output corresponding to PWM command

Then, from equations (3.65) and (3.69), the relation between motor speed n and duty cycle d_{pwm} is written as the below equation.

$$n \approx \begin{cases} \alpha d_{pwm} - \beta & d_{threshold} \leq d_{pwm} \leq 1 \\ 0 & d_{pwm} \leq d_{threshold} \end{cases} \quad (3.72)$$

If the characteristic information of the motors is available, the parameter α , β of the model of equation (3.72) can be computed. However, there is no available information about the motor constants, it is necessary to calibrate the relation between PWM input and RPM output of a motor with a propeller. Therefore, the composite parameters of the gradient α and the intercept β are calibrated by experiments. In the calibration, a reflective marker sticker is put on each propeller, and a tachometer measures rotation speed of the markers. PWM signal of voltage is controlled by an ESC, and the range of PWM command is between 1000 μ sec and 2000 μ sec. The result of the experiment and the approximated model by the RMSE linear regression method are shown at the Figure 3.6 and equation (3.73).

$$n = \begin{cases} 0.1716 \times d_{pwm} - 803.4 & 1100 \leq d_{pwm} \leq 2000 \\ 0 & \text{otherwise} \end{cases} \quad (3.73)$$

Since a brushless DC motor used in the quadrotor system does not have an encoder to measure its motor speed, open-loop control based on equation (3.73) is applied to control motor speed. Also, PWM command limit is set between 1230 μ sec and 1950 μ sec to prevent inappropriate performance.

Chapter 4

Evaluation of the Control System

In this chapter, in order to validate the advantage of the control system stated in chapter 3, the nonlinear attitude control and an ordinary PID attitude control are compared with each other, by simulation and flight experiments. First, the PID attitude control that is used for comparison and its linear mapping model are described. Then, the simulation setup is explained and the simulation results of the two control systems are stated. Finally, the experimental setup is described and the trajectory results of the two control systems are analyzed and discussed.

4.1 Linear Attitude Control for Comparison

To compare with the customized control system of chapters 3, an ordinary inner-loop controller is also tested for comparison. At the linear controller, desired angular rate $\boldsymbol{\omega}_d$ is also computed as described in chapter 3.3. Similarly, the reference angular velocity $\boldsymbol{\omega}_r = (p_r, q_r, r_r)$ is given as the below equation.

$$\begin{aligned}\boldsymbol{\omega}_r &= Z^{-1} \dot{\boldsymbol{\eta}}_r \\ &= Z^{-1} \dot{\boldsymbol{\eta}}_d + Z^{-1} \Lambda(\boldsymbol{\eta}_d - \boldsymbol{\eta})\end{aligned}\tag{4.1}$$

However, instead of equation (3.35) of the customized control system, PID control with respect to the quadrotor's angular velocity is applied to the linear controller. For convenience, we assumed that the derivative of the reference angular velocity is zero. The control law is given as,

$$\begin{aligned}u_p &= k_{P\phi}(p_r - p) + k_{D\phi}\dot{p} + k_{I\phi} \int_{t_0}^t (p_r - p) dt \\ u_q &= k_{P\theta}(q_r - q) + k_{D\theta}\dot{q} + k_{I\theta} \int_{t_0}^t (q_r - q) dt \\ u_r &= k_{P\psi}(r_r - r) + k_{D\psi}\dot{r} + k_{I\psi} \int_{t_0}^t (r_r - r) dt\end{aligned}\tag{4.2}$$

Also, the mixer uses more simplified linearized model as below, instead of the equation (3.64). The matrix

for mapping the desired torques into desired motor speeds are constant as the below equations.

$$\begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \end{bmatrix} = M^{-1} \begin{bmatrix} k_f F_d \\ u_p \\ u_q \\ u_r \end{bmatrix} \quad (4.3)$$

where k_f is a gain for the scale of thrust and M is defined as,

$$M = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 1 & 1 & -1 & -1 \end{bmatrix} \quad (4.4)$$

The matrix M represents only the proportional effect of each motor on torques and thrusts due to the geometry of the frame. Other physical factors are considered by tuning the gains of the attitude PID control.

4.2 Simulation Study for Inner-loop Controller

It is difficult to evaluate of the performance of the control systems in a limited area, since changing and holding the quadrotor's attitude makes the quadrotor to move out of our physical test area. Therefore, it is more suitable to use simulation rather than real flight for evaluation of the quadrotor's performance.

PID inner-loop controller and a linear mapping model from torque into motor speeds are often used for attitude control of a quadrotor. Let it be called "linear attitude controller". In this chapter, to validate the advantage of the control system stated in chapter 3 and 4, simulation of these two systems is stated, and comparison of the results is discussed to validate the advantage of the customized attitude control of section 3.4. First, the attitude controller based on PID inner-loop and a linear mapping model is described, and setup for simulation is explained. Then, simulation results of the two control systems are discussed.

4.2.1 Simulation Setup

In simulation, the dynamic models of the equations (3.7) and (3.14) are applied. The physical quantities of the quadrotor are the same as stated in the chapter (1) in each case. The inner-loop controller was simulated with the following cases;

$$1. \ \boldsymbol{\eta}_d = (\frac{\pi}{6}, 0, 0)$$

$$2. \ \boldsymbol{\eta}_d = (\frac{\pi}{3}, 0, 0)$$

$$3. \ \boldsymbol{\eta}_d = (0, \frac{\pi}{6}, 0)$$

$$4. \ \boldsymbol{\eta}_d = (0, \frac{\pi}{3}, 0)$$

$$5. \ \boldsymbol{\eta}_d = (0, 0, \frac{\pi}{4})$$

$$6. \ \boldsymbol{\eta}_d = (0, 0, \frac{\pi}{2})$$

$$7. \ \boldsymbol{\eta}_d = \begin{cases} (\frac{\pi}{6}, \frac{\pi}{6}, 0) & 1 \leq t \leq 2 \\ (-\frac{\pi}{6}, -\frac{\pi}{6}, 0) & 3 \leq t \leq 4 \\ (0, 0, 0) & \text{otherwise} \end{cases}$$

In every case, the magnitude of thrust control F_d is constant to be mg , and each controller is well-tuned.

4.2.2 Results

The results of simulation is shown below. Figure (4.1) - (4.7) shows the result of each case at section 4.2.3.

4.2.3 Discussion

In most cases, the attitude of the quadrotor converges to desired attitude faster with the customized control system than the ordinary linear control system. With roll and pitch changes, the attitude becomes close enough to desired attitude within 0.3sec at the customized attitude control, and 0.8sec at the linear attitude control. In the cases of yaw, the rate of convergence is approximately same at both control systems; the attitude converges to desired attitude within and 1.5sec. From these, it can be concluded that the customized attitude controller has the advantage of fast conversion. The improvement of the attitude control is expected from the global exponential stability of the customized attitude controller. In figure 4.7, response time of the customized attitude controller is fast enough so that the attitude of the quadrotor stabilize quickly, even with rapid change of desired attitude. Therefore, the customized attitude controller is more suitable for high-agility performance. The deference of response time of the customized attitude control with respect to roll, pitch, and yaw is thought to be caused by the difference of the inertia values along axises, I_{xx} , I_{yy} and I_{zz} .

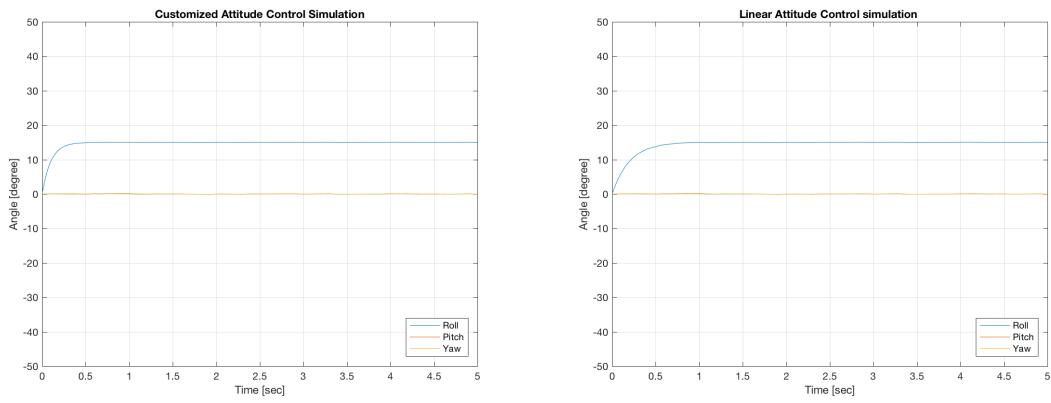


Figure 4.1: Simulation Result (Case 1)

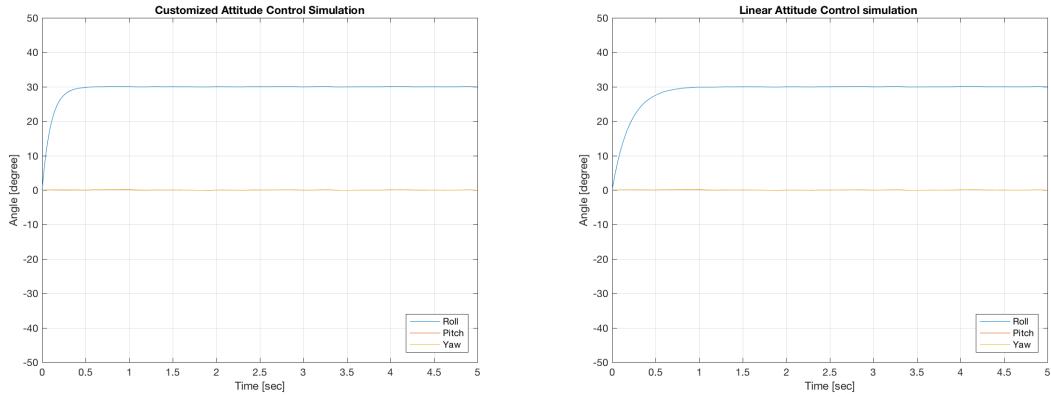


Figure 4.2: Simulation Result (Case 2)

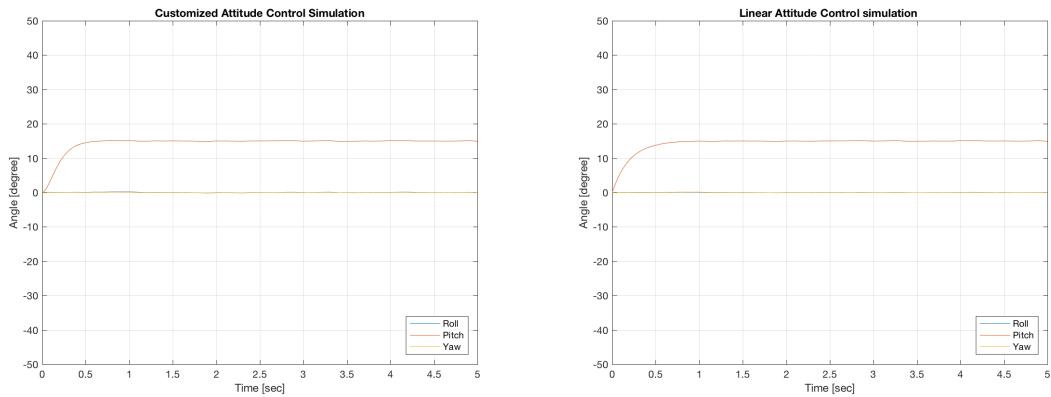


Figure 4.3: Simulation Result (Case 3)

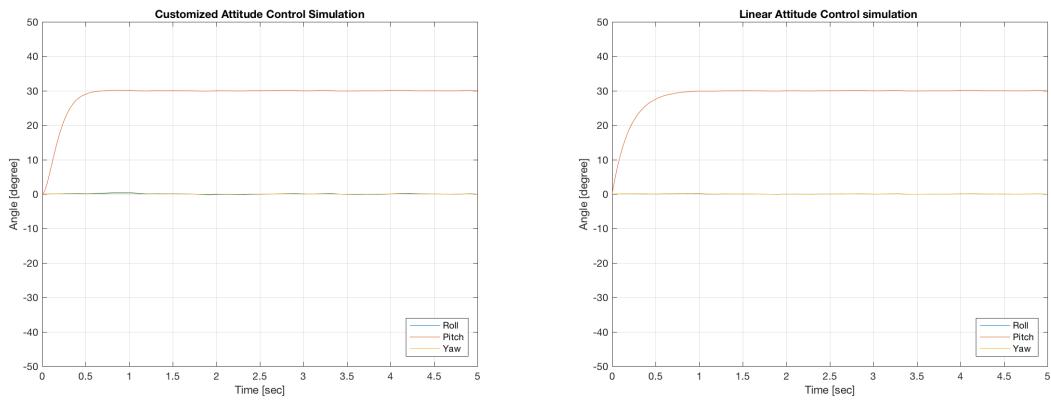


Figure 4.4: Simulation Result (Case 4)

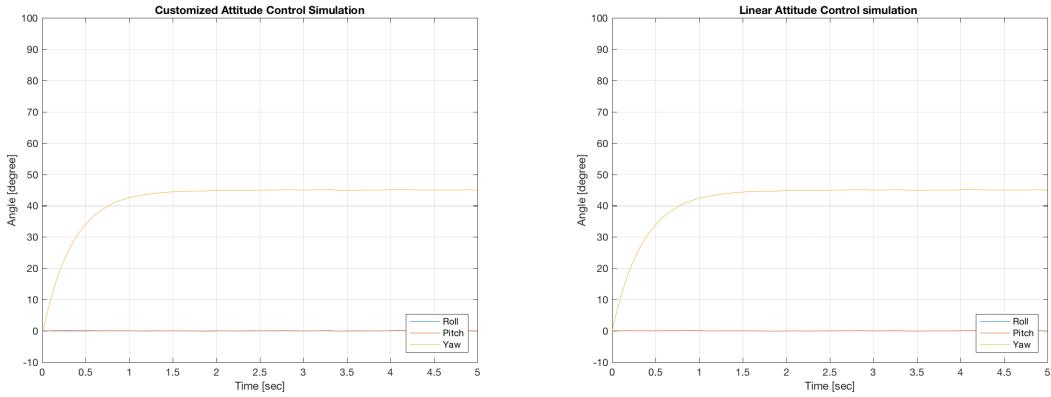


Figure 4.5: Simulation Result (Case 5)

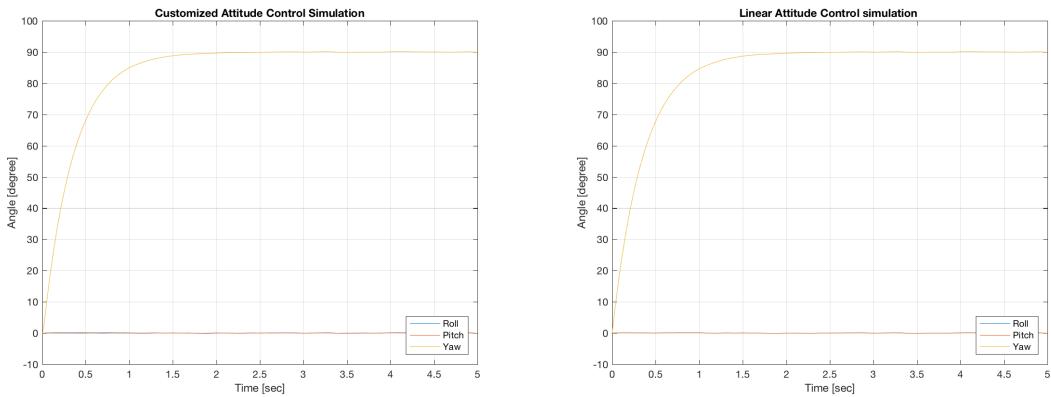


Figure 4.6: Simulation Result (Case 6)

However, yaw errors of the customized attitude control are bigger when desired attitude changes in the case 5, than the one of the linear attitude control. This is thought to be caused by dependency among each angular velocity parameter p , q , r at the customized attitude controller. As shown at equation (3.36), each parameter cannot be decoupled, while the PID control deals each angular velocity independently. Also, the performance of roll and pitch is not similar at the customized attitude controller, while the ones of the linear attitude controller do. The difference of the inertia among each axis of the body frame is thought to be caused the asymmetry of the performance.

4.3 Flight Experiment

In addition to the simulation study at section 4.2, the quadrotor system was tested by experiments to evaluate the controller. The flight experiments were executed at the motion capture arena of the Intelligent Robotics Laboratory(IRL) at the University of Illinois at Urbana-Champaign. The quadrotor is controlled with the outer-loop and inner-loop control stated at chapter 3. The linear attitude control stated at section 4.1 are also tested with the outer-loop control of section 3.3 to compare with the performance of the customized attitude control. The motion of the quadrotor is recorded by a motion capture system. In this chapter, the experimental setup for the evaluation of the quadrotor is stated, and then, the experiment results and discussion about them are followed.

4.3.1 Motion Capture Arena

The quadrotor experiments were done at IRL's motion capture arena for human safety. The arena is equipped with safety facilities for UAVs' flight experiments. The arena is isolated so that the quadrotor does not damage anything out of the arena. In order to limit the range of quadrotors' maneuver and protect them from potential damage, the arena are surrounded by nets. In addition, the floor of the arena is covered with shock absorbing foams to protect the quadrotor from damaging itself. With the setup, potential accidents from the quadrotor's malfunction can be prevented.

Mounted on the ceiling of the four corners of the room, there are installed two motion capture camera; in total eight cameras consists of the motion capture system. The whole arena area is covered by the motion capture range, and the motion capture system measure motion of the object that has motion capture markers on it.

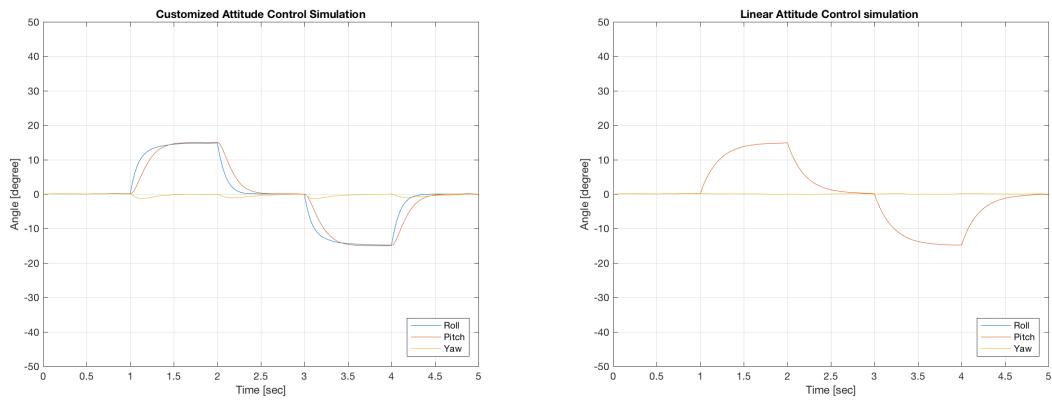


Figure 4.7: Simulation Result (Case 7)



Figure 4.8: Motion Capture Arena of the Intelligent Robotics Laboratory



Figure 4.9: Vicon Motion Capture Cameras

4.3.2 Motion Capture System

IRL's motion capture system consists of 8 Vicon T-series Cameras mounted around the arena. Motion capture cameras of Vicon radiate infrared rays and capture infrared rays reflected from the markers. By synchronizing the pixel coordinate of the reflective markers at multiple motion capture cameras, the motion capture system obtains the position information of the markers.

Prior to the experiment, it is necessary to register an object that is a rigid body and has multiple markers on it. Then, the motion capture system saves the geometric information of the reflective markers in the database and recognizes the object. In order to measure both the position and attitude of the quadrotor, at least 3 reflector markers must be located on the quadrotor since a plane consist of at least three points. In this experiment, 7 markers are fixed on the quadrotor to guarantee the precision of the motion information; three are located on the quadrotor's legs and the other markers are stick on the top of the quadrotor. To avoid incorrect measurement of the quadrotor's attitude, the markers are installed asymmetrically.

The motion capture system of IRL works with Tracker, a graphic user interface software for Vicon, and Vicon Bridge, a ROS package developed by the autonomous systems laboratory at ETH Zurich are used for streaming the data of the quadrotor's position and attitude.

4.3.3 Experiment Setup

Prior to experiment, we tuned the control gains of the customized and linear inner-loop controllers, based on the simulated control gains at section 4.2 and adjusted the gains by flight tests.

In the experiments, the quadrotor was controlled by the PID outer-loop controller described at section 3.3 and either the customized attitude control of chapter 3 or the linear attitude control of section 4.1.

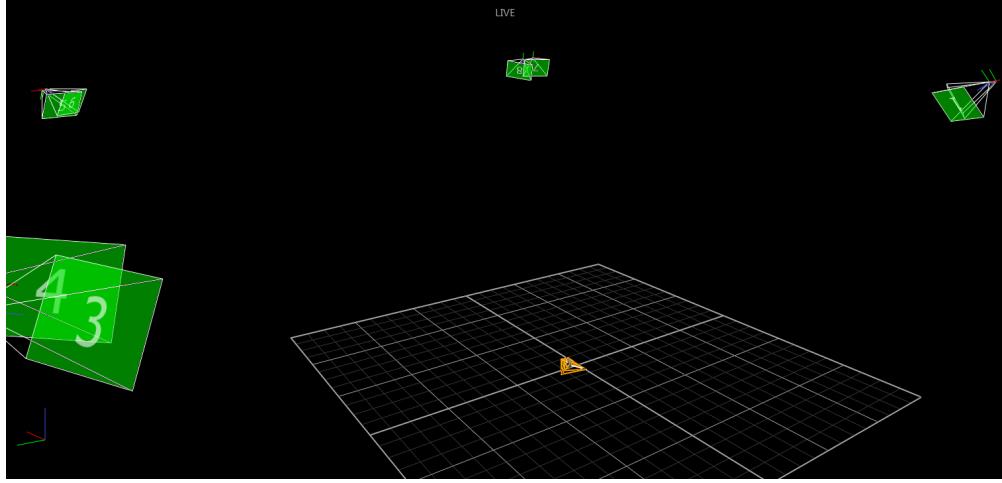


Figure 4.10: Vicon Motion Capture Interface

Since it is difficult to set the same desired state of the quadrotor for its position, the position and attitude trajectories of the quadrotor to reach desired positions are used to evaluate the performance of each inner-loop controller. The x, y position control at outer-loop control is expected to drift since the quadrotor does not include internal position measurement for x, y-axis direction. Therefore, the desired position of each flight experiment is set to be as far as possible in the motion capture arena from the initial position.

The quadrotor's experiments were conducted, focused on control system's stability and agility. First, in order to evaluate the stability of the quadrotor, we changed the altitude of the waypoint from 1.8 m to 3 m. The quadrotor first would hover and would stabilize its position at the first waypoint and moves to the next waypoint and land. To evaluate the roll agility of the quadrotor, we changed the waypoint to either left side, right side from the first waypoint, and then control the quadrotor to land. The waypoint was changed to either forward or backward to validate the pitch agility. Finally, the performance of yaw was also tested by setting desired yaw to be $\pm \frac{\pi}{2}$ while the quadrotor is hovering on the waypoint. Similarly, as yaw converges to the desired yaw, the quadrotor landed. To summarize, the following cases were tested.

1. Take off, change altitude from 1.8m to 3.0m, and land.
2. Take off, move right side, and land.
3. Take off, move left side, and land.
4. Take off, move forward, and land.
5. Take off, move backward, and land.
6. Take off, turn $\pm \frac{\pi}{2}$ to the right, and land.

7. Take off, turn $\pm \frac{\pi}{2}$ to the left, and land.

4.3.4 Experiment Results

All the experimental data are filtered by a low-pass filter and transformed into the inertial frame that is defined at chapter 2. Also the origin point of each case was defined to be the initial position of the quadrotor. The trajectories and attitude changes over time are plotted at figures 4.11 - 4.17. Full lines represents the quadrotor's actual trajectories and dashed lines represents the desired position or waypoints.

4.3.5 Discussion

As discussed above, the quadrotor does not have internal position measurement of the x,y-axis directions, there are drifts among x,y-axes direction in every case. The position estimator uses integration of internal accelerometer, the error is accumulated by iteration and position drift is not avoidable. The errors caused by drift are usually less sever with the customized control system introduced at chapter 3. The drift occurs especially when the quadrotor takes off and lands. The potential reason of the drift at taking off and landing are by dynamic factors, such as corrosion between the quadrotor's leg and the ground. Also, the aerodynamic model may be different near the ground and, therefore, may cause drifts as well.

While the errors of x,y-axis position at the customized controller are generally less than half of the one at the linear controller. In case 1, y-axis drift is less than 1m at the customized control system while it is about up to 3m at the linear control system, and x-axis drift is similar with both controllers. In case 2 and 3, x,y-axis position error at the linear attitude control is bounded up to 1m, but the error of the linear grows up to 3m. The tendency of x,y-axis error is similar in case 4 and 5. However, compared with case 2 and 3, the error at the linear control system does not exceed 2m. In case 6 and 7, the rate of x,y-axis error at the customized control system is half of the one at the linear control system. From these, we can find that the customized control system has an advantage of stability compared with the linear control system.

However, the customized control system does not seem to have an advantage with respect to z-axis variance over an ordinary linear control system. There are increasing or decreasing of altitude while the quadrotor changes its attitude, but the error of z-axis position is less than 0.5m at both controllers in most cases. As mentioned in chapter 2, the Pixhawk autopilot is equipped with an internal barometer to measure altitude and frequently corrects position estimation of double integration of acceleration. Therefore, with more precise altitude estimation, the quadrotor stabilizes its altitude better than its x,y-axis position.

In addition, at both controllers, time for the quadrotor to reach desired position is approximately same, since there is a upper limit of desired velocity. However, as shown at the graphs of x and y position, the

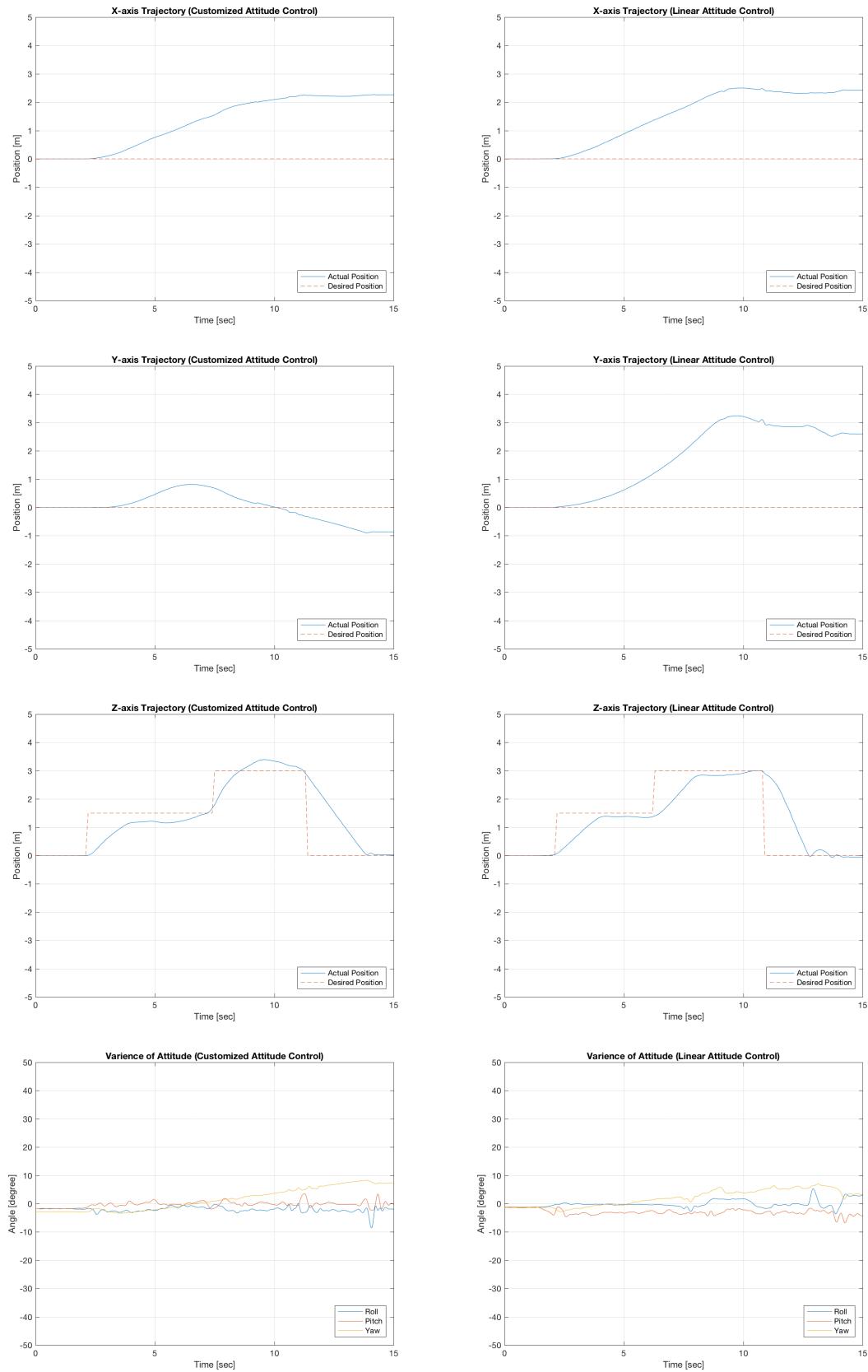


Figure 4.11: Experiment Result (Case 1)

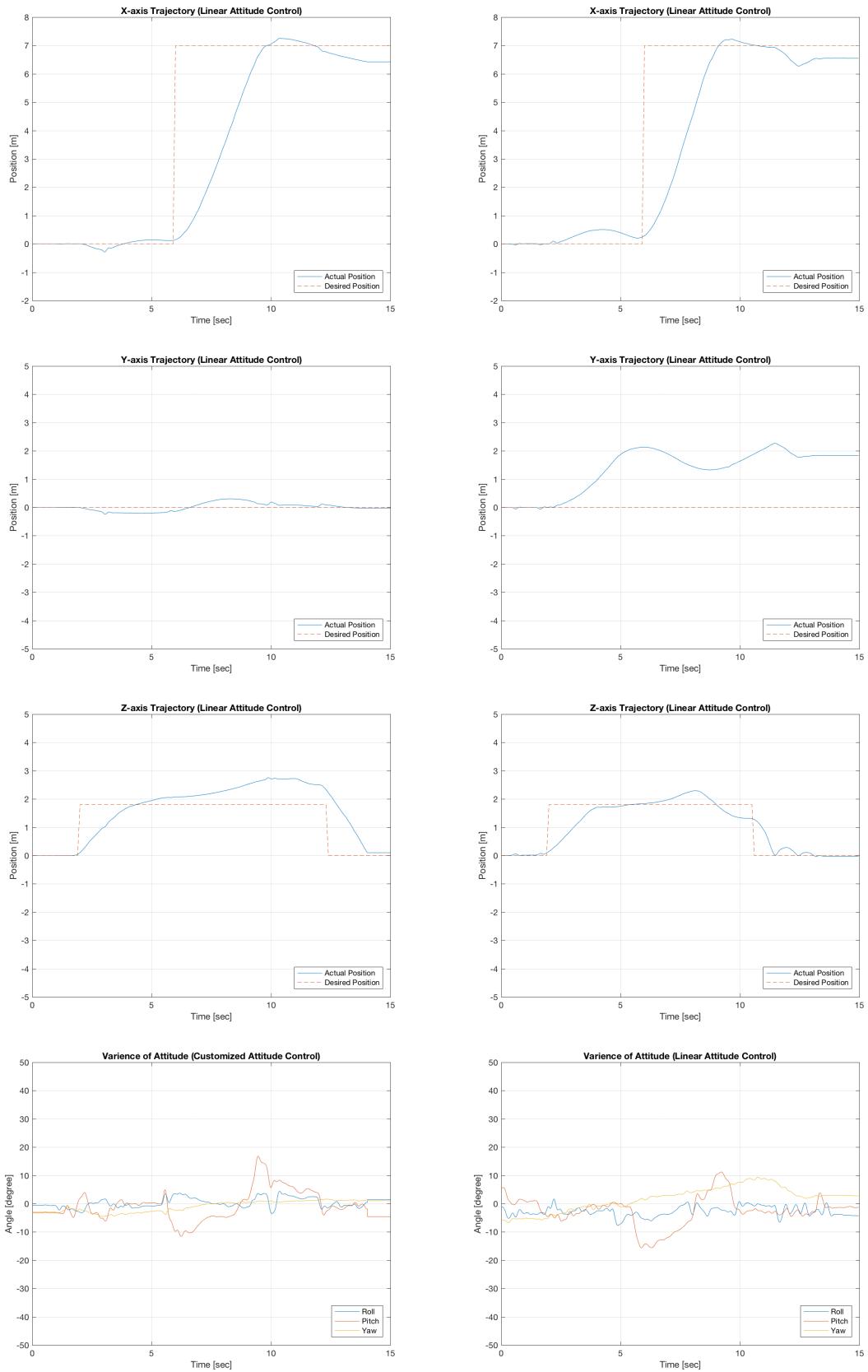


Figure 4.12: Experiment Result (Case 2)

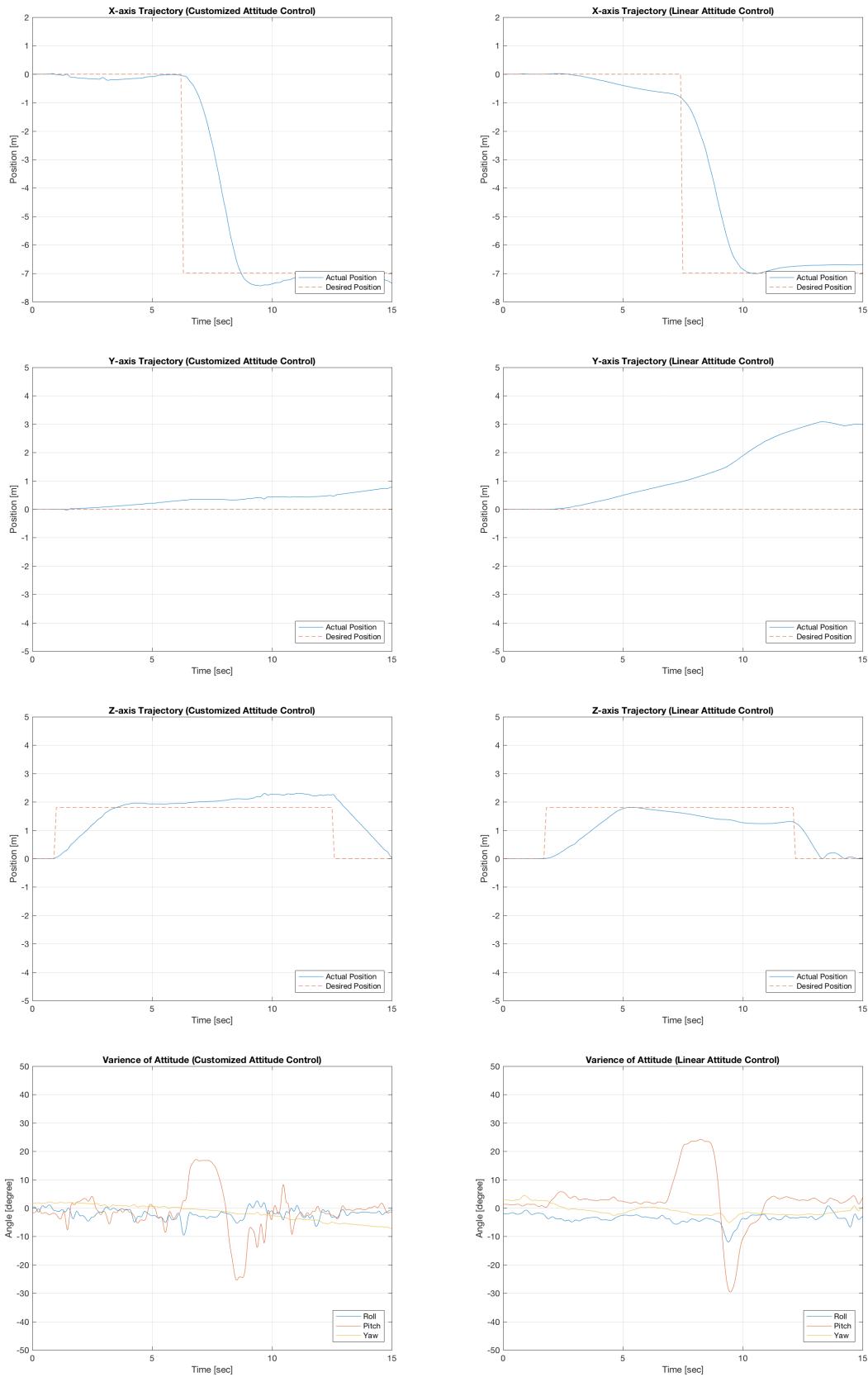


Figure 4.13: Experiment Result (Case 3)

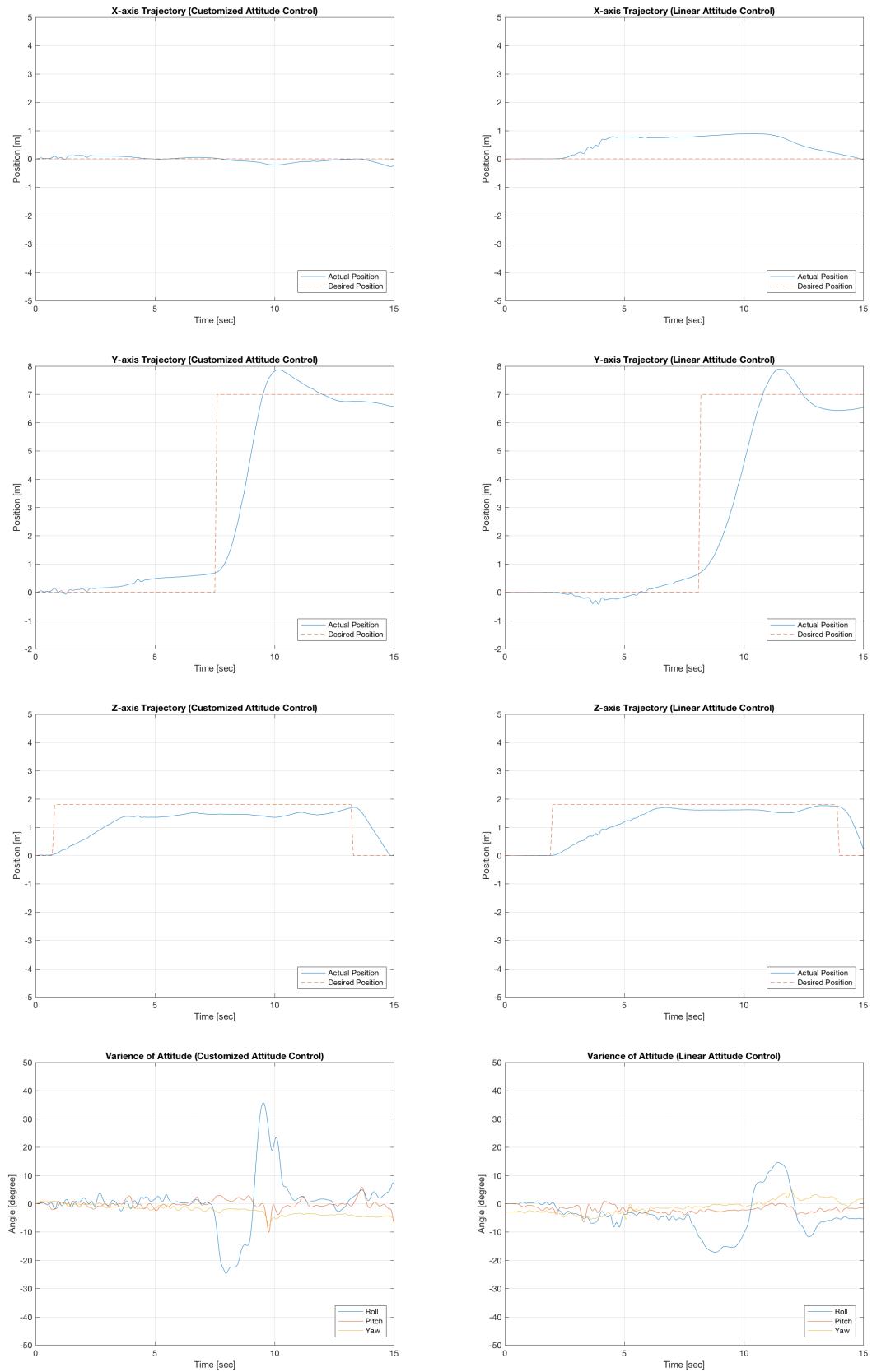


Figure 4.14: Experiment Result (Case 4)

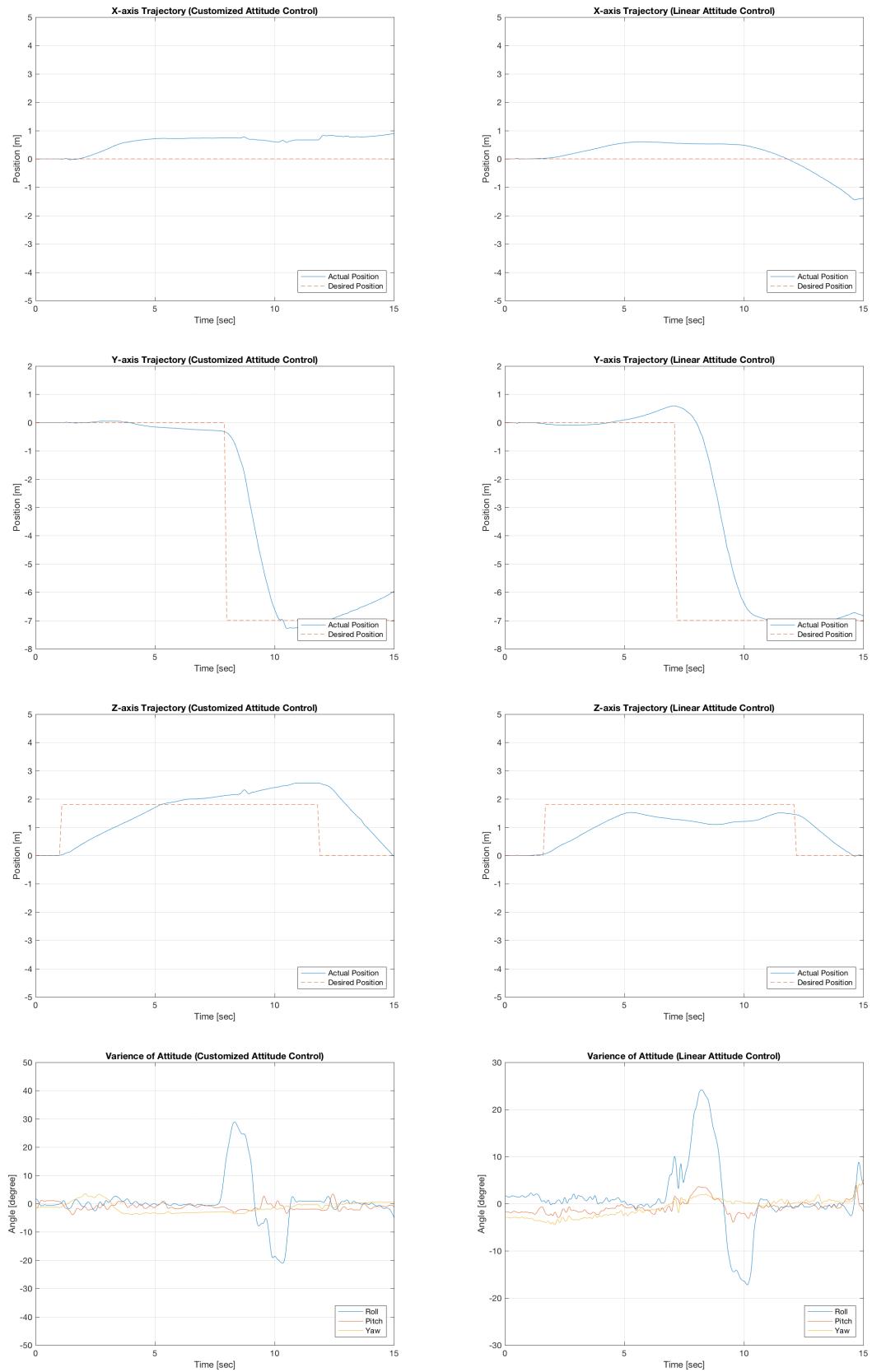


Figure 4.15: Experiment Result (Case 5)

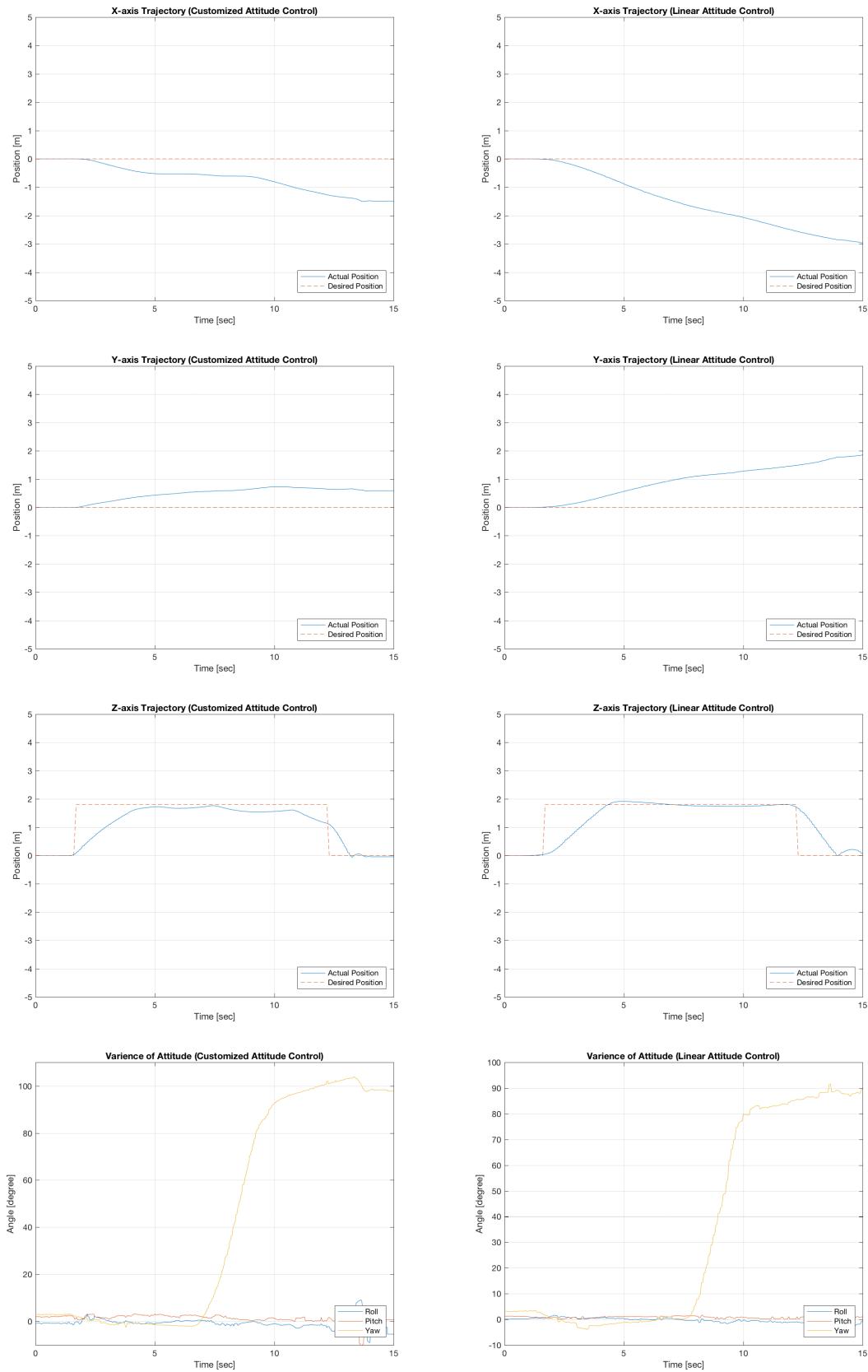


Figure 4.16: Experiment Result (Case 6)

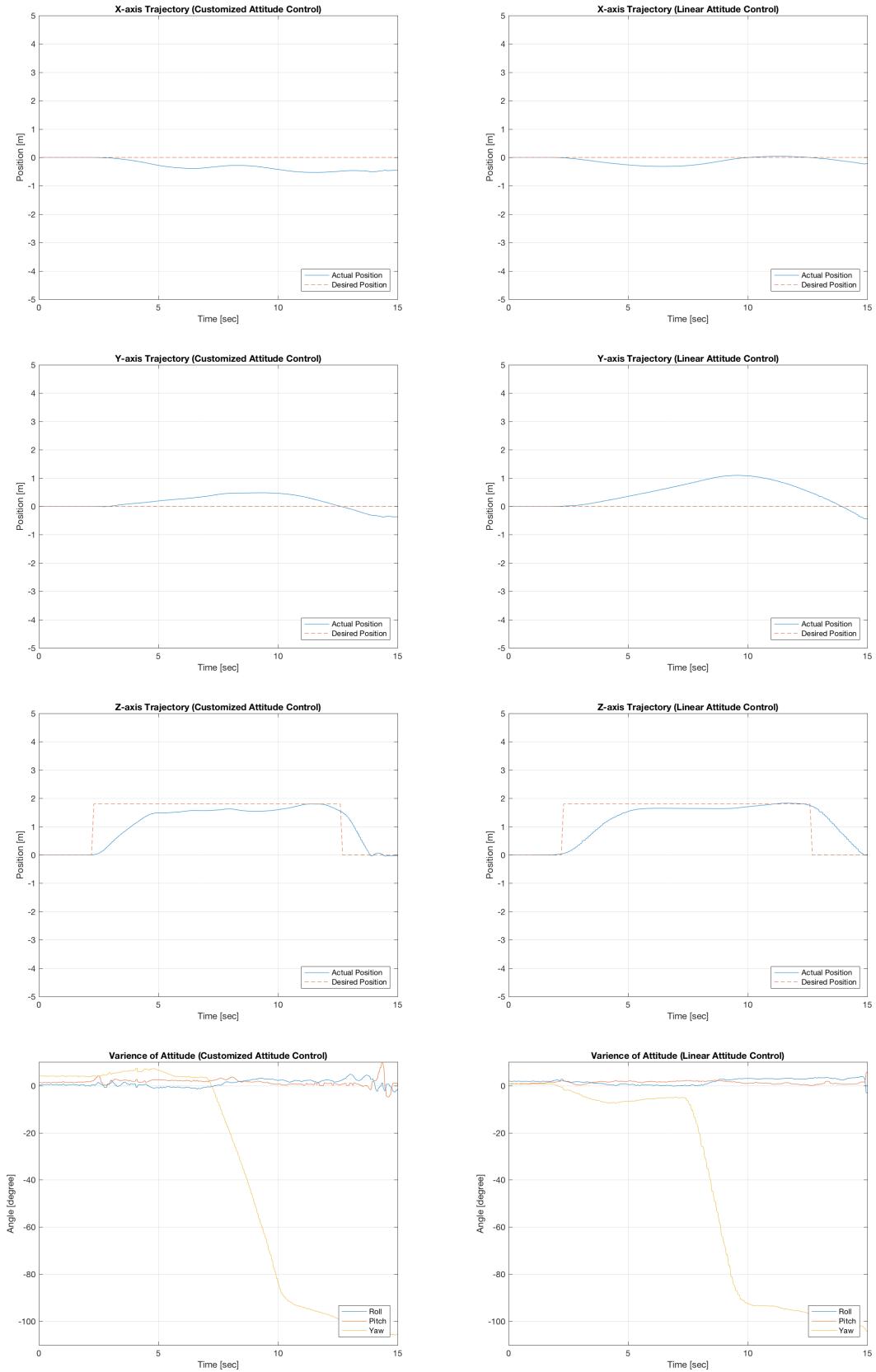


Figure 4.17: Experiment Result (Case 7)

quadrotor with the customized control system responses fast and reaches to the maximum velocity, while it moves more smoothly with the linear control system. Also, as shown at figure 4.16 and 4.17, yaw control is more stabilized with the customized control system. These support the globally exponentially stability of the nonlinear attitude control at chapter 3.4 and the results of simulation at section 4.2 that the customized control converges to desired attitude faster than PID attitude control does since the quadrotor's attitude affects the direction of acceleration. Therefore, we can conclude that the customized control system also has an advantage of its stability.

There is constant yaw drift at both control systems in case 2 and 3, Yaw angle increases when the quadrotor moves to right, while the yaw angle decreases when the quadrotor moves to the left. We can presume that the misalignment of the internal gyroscope or the different output curves of the DC motors caused the drift.

Chapter 5

Vision-based Position Estimator

As discussed at chapter 7, the absence of precise position measurement causes the quadrotor's drift. Therefore, the quadrotor must be equipped with proper position measurement, such as a GPS sensor and an image sensor, for its stable autonomous operation. Therefore, a simple and light position estimator based on image processing is introduced in this chapter. We assumed that the quadrotor hovers on the floor with four markers.

This chapter consists of following contents. First, the camera distortion model used in the system is stated. Then, the image processes for detecting markers are described. Finally, a geometric relation between the pixel coordinates of the markers and the quadrotor's position is described, and a method to estimate the quadrotor's position is introduced.

5.1 Camera Calibration

Image from a camera is often distorted by mechanical factors; radial distortion caused by the shape of the lens, and tangential distortion caused by the angle and the distance between the image sensor and the lens. The camera used in the quadrotor system has 120° view angle, and due to its wide angle, radial distortion is especially severe. In order to collect precise image data from the camera, calibration is required.

In this research, camera calibration is processed by GML C++ Camera Calibration Toolbox, developed by the Graphics and Media Laboratory at Moscow State University.[15, 16] The calibration toolbox is based on the polynomial camera model. Let $x_d = (x_d, y_d)$ be the normalized point coordinate with distortion, and $x = (x, y)$ be the normalized point coordinate without distortion. Then, the model is given as the below equation.

$$x_d = (1 + k_{c1}r^2 + k_{c2}r^4 + k_{c5}r^6)x + \begin{bmatrix} 2k_{c3}xy + k_{c4}(r^2 + 2x^2) \\ k_{c3}(r^2 + 2y^2 + 2k_{c4}xy) \end{bmatrix} \quad (5.1)$$



Figure 5.1: Camera Calibration (Left: Raw Image, Right: Restored Image)

where $\mathbf{k}_c = (k_{c1}, k_{c2}, k_{c3}, k_{c4}, k_{c5})$ is the distortion coefficients of the camera and

$$r^2 = x^2 + y^2 \quad (5.2)$$

In equation 6.1, the first term of the right side represents radial distortion and the second term of the right side represents tangential distortion. In addition to distortion, the pixel coordinates of the projection p is given as,

$$p = KK\mathbf{x}_d \quad (5.3)$$

where KK is the camera matrix, defined as,

$$KK = \begin{bmatrix} f_{c1} & \alpha_c f_{c1} & c_{c1} \\ 0 & f_{c2} & c_{c2} \\ 0 & 0 & 1 \end{bmatrix} \quad (5.4)$$

f_{c1}, f_{c2} are the focal distances and α_c is the angle between the x and y sensor axes. Therefore, by measuring the distortion coefficients \mathbf{k}_c and the camera matrix KK , the undistorted coordinate \mathbf{x} can be restored from the raw pixel coordinate p . Figure 6.1 shows the results of the marker image restored from the raw image by camera calibration.

5.2 Marker Detection

Marker detection on the image can be achieved by several image processes. First of all, monochrome image is used by the image process. In order to decrease noise and find clear borders, the image is filtered by the Gaussian filter and thresholded. Then, the morphology closing transformation is processed to restore holes on the borders. Once the image have clean borders, the markers are detected with its contours. To be specific, "findContours" function of OpenCV 2.4 was used for the process of finding contours.

From the above image process, markers on the image are detectable. However, due to environmental factors, such as lightness and the floor state, empty coordinates may be detected as a marker. Therefore, the follow strategy is applied in the system. Consider the case that more than 4 coordinates p_i are detected as markers. Since the reference makers are fixed and clustered, and the number of wrongly-detected coordinates are few, usually zero or at most, two, markers are located close to the center of the set of candidates. By using this method, wrong candidates are excluded easily.

The results of each step is shown at figure 6.2 - 6.4. The white circle at figure 6.4 points the center of the image markers.

5.3 Position Estimation

Let \vec{h} be the normal vector of the camera's orientation. Then, it can be represented with respect to the quadrotor's attitude $\boldsymbol{\eta}$ as,

$$\vec{h}(\boldsymbol{\eta}) = Z^{-1} \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} = \begin{bmatrix} \sin \theta \\ -\sin \phi \cos \theta \\ -\cos \phi \cos \theta \end{bmatrix} \quad (5.5)$$

Let \mathbf{A} to be the area of the polygon in the inertial frame, and \vec{l}_i to be the relative position of a marker according to the position of the camera. Then, the projection of \mathbf{A} is represented as below,

$$\begin{aligned} \vec{h} \cdot \mathbf{A} &= \frac{1}{2}(\vec{l}_2 - \vec{l}_1) \times (\vec{l}_4 - \vec{l}_1) + \frac{1}{2}(\vec{l}_4 - \vec{l}_3) \times (\vec{l}_2 - \vec{l}_3) \\ &= \frac{1}{2}(\vec{l}_1 \times \vec{l}_2 + \vec{l}_2 \times \vec{l}_3 + \vec{l}_3 \times \vec{l}_4 + \vec{l}_4 \times \vec{l}_1) \end{aligned} \quad (5.6)$$

From perspective projection of the camera, the relative position \vec{l}_i is represented with respect to the normalized pixel coordinate \mathbf{p}_i ,

$$\vec{l}_i = \frac{d}{\vec{h} \cdot (R_\psi^{-1} \mathbf{p}_i)} R_\psi^{-1} \mathbf{p}_i \quad (5.7)$$

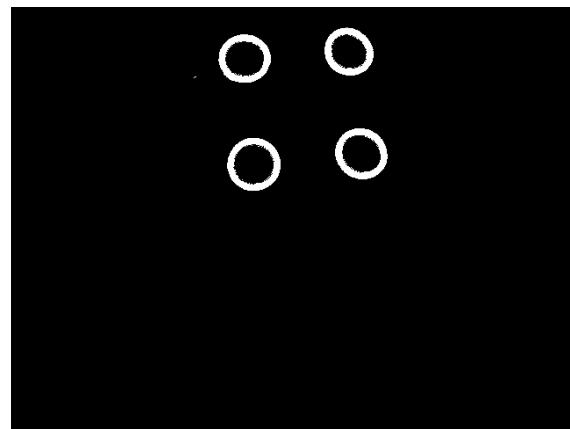


Figure 5.2: Gaussian-filtered and Thresholded Image

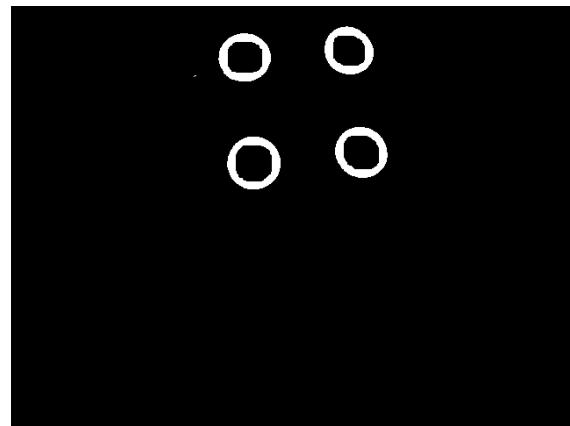


Figure 5.3: Restoration of Borders

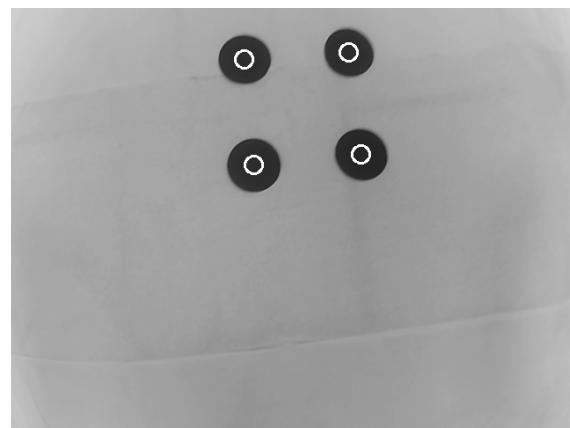


Figure 5.4: Final Result of Marker Detection

where d is the perpendicular distance from the camera to the plane of the markers and it is unknown. For convenience of computation, define the composite coordinate,

$$\mathbf{p}_{\psi_i} = R_{\psi}^{-1} \mathbf{p}_i \quad (5.8)$$

Then, equation (6.6) can be written as,

$$\vec{h} \cdot \mathbf{A} = \frac{\Gamma d}{2} \quad (5.9)$$

where Γ is composite value defined as,

$$\Gamma = \left(\frac{\mathbf{p}_{\psi_1} \times \mathbf{p}_{\psi_2}}{(\vec{h} \cdot \mathbf{p}_{\psi_1})(\vec{h} \cdot \mathbf{p}_{\psi_2})} + \frac{\mathbf{p}_{\psi_2} \times \mathbf{p}_{\psi_3}}{(\vec{h} \cdot \mathbf{p}_{\psi_2})(\vec{h} \cdot \mathbf{p}_{\psi_3})} + \frac{\mathbf{p}_{\psi_3} \times \mathbf{p}_{\psi_4}}{(\vec{h} \cdot \mathbf{p}_{\psi_3})(\vec{h} \cdot \mathbf{p}_{\psi_4})} + \frac{\mathbf{p}_{\psi_4} \times \mathbf{p}_{\psi_1}}{(\vec{h} \cdot \mathbf{p}_{\psi_4})(\vec{h} \cdot \mathbf{p}_{\psi_1})} \right) \quad (5.10)$$

From the above equation, d is computed as,

$$d = \sqrt{\frac{2\vec{h} \cdot \mathbf{A}}{\Gamma}} \quad (5.11)$$

and, therefore, the relative position of the camera according to the markers' center is given as,

$$\begin{aligned} \tilde{\mathbf{r}} &= -\frac{1}{4}(\vec{l}_1 + \vec{l}_2 + \vec{l}_3 + \vec{l}_4) \\ &= -\frac{d}{4} \left(\frac{\mathbf{p}_{\psi_1}}{\vec{h} \cdot \mathbf{p}_{\psi_1}} + \frac{\mathbf{p}_{\psi_2}}{\vec{h} \cdot \mathbf{p}_{\psi_2}} + \frac{\mathbf{p}_{\psi_3}}{\vec{h} \cdot \mathbf{p}_{\psi_3}} + \frac{\mathbf{p}_{\psi_4}}{\vec{h} \cdot \mathbf{p}_{\psi_4}} \right) \end{aligned} \quad (5.12)$$

Since the related position calculated from the markers is noisy, the Kalman filter is applied to estimate the relative position. Values computed by equation 6.12 pass through the Kalman filter, and the filtered values are used as relative position to the quadrotor, corresponding to the markers.

5.4 Evaluation Experiment

5.4.1 Experimental Setup

The vision-based position estimator was tested by experiments. The experiments were also executed at the IRL motion capture arena. The quadrotor is not armed and the autopilot only publish the sensor values to the onboard companion computer. The algorithm was run by the companion computer real-time. To evaluate the results, the Vicon motion capture system of the arena was used, and data from the position estimator and the motion capture system are compared. As described at chapter 5, the Vicon system detects

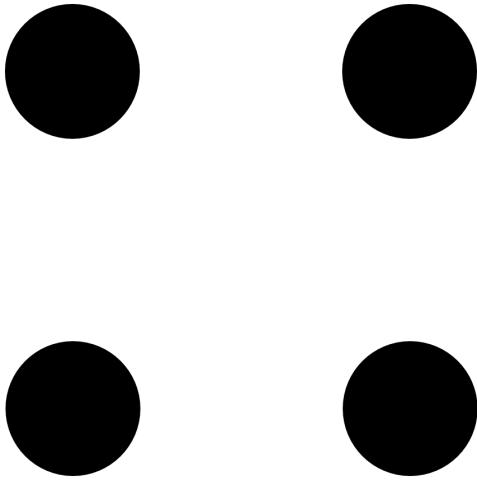


Figure 5.5: Image Markers for the Evaluation

the reflective markers on the quadrotor and measure the position of the quadrotor. The image markers used in the experiment for position estimation are shown at figure 6.5. Each image marker is a 15cm-diameter-circle and the center is located away from the ones of neighbors. The placement of the image markers forms a square. The origin of the inertial coordinate system is set on the center of the image markers.

5.4.2 Results

All the position data from the motion capture system are filtered by a low-pass filter and transformed into the inertial frame defined at chapter 2. The position of data from the position estimator are processed by a real-time Kalman filter. The results of the experiment is shown at figure 6.6 - 6.9. Each figures shows both the position measurements of the above vision-based position estimator and the motion capture system.

5.4.3 Discussion

As shown at figure 6.6 - 6.9, the error of the position estimator is about 0.3m when the position of camera moves slowly. Therefore it can be concluded that the position estimator roughly correct. However, the position estimator was not able to reflect small change of the position among z-axis. Therefore, the position estimator is more suitable at slow flight, rather than agile performance. In addition, the camera may not capture the image markers if the camera is too close from the ground. The absence of the markers on the image frame can be critical reason for the failure of position estimation. The problem expected to happen when the quadrotor has high roll or pitch angle. Therefore, it require to develop a system that deals with

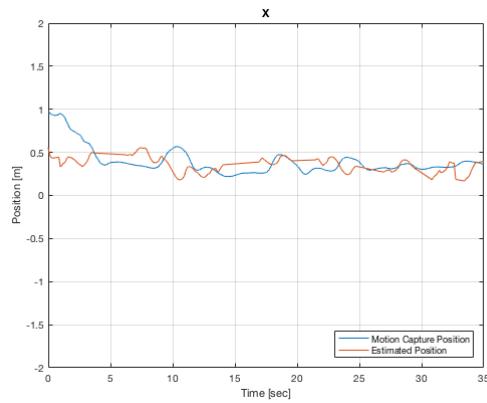


Figure 5.6: Experiment Result of the Position Estimator (x-axis Position)

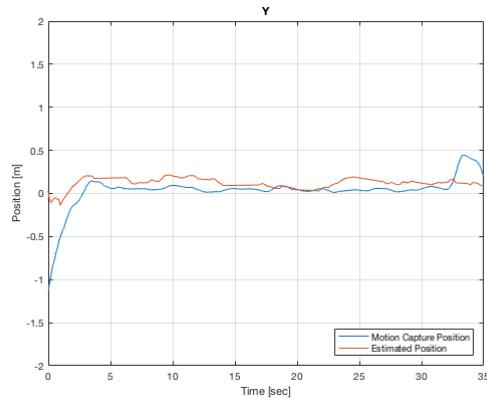


Figure 5.7: Experiment Result of the Position Estimator (y-axis Position)

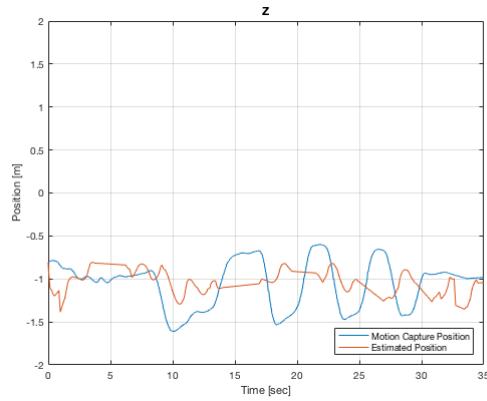


Figure 5.8: Experiment Result of the Position Estimator (z-axis Position)

temporary failure of marker detection.

Chapter 6

Conclusion

In this research, a dynamic model of the quadrotor was studied and, based on the model, a nonlinear attitude control was implemented in the quadrotor system. The inner-loop control has a advantage of globally exponentially stability. Therefore, if a proper dynamic model for the quadrotor can be defined, the control law is expected to allow the better performance of the quadrotor. Also, the advantage of the nonlinear controller, compared with ordinary controllers, was validated by simulation studies. Further, the nonlinear attitude control was implemented in the quadrotor system, and the performance with the nonlinear control was compared with the ordinary attitude control and evaluated by experiments.

However, in order to develop fully-autonomous system for the quadrotor, it is necessary to install position measurement on the quadrotor. With accelerometer, a position measurement make the motion of the quadrotor fully observable. Therefore, at the last part of the thesis, a light-computation vision-based position estimator was designed. Since the position estimator is executable real-time, it is expected to be used in the quadrotor system so that the quadrotor collects precise position data.

As future works, synchronized with the nonlinear attitude control, the application of an advanced outer-loop control is expected to further improve the quadrotor's performance. Also, in this research, the dynamics of motors are approximated by experiments, but precise modeling of a motor would increase the robustness of the inner-loop control as well.

References

- [1] *ICAO's circular 328 AN/190 : Unmanned Aircraft Systems* [Retrieved 15 April 2016]
- [2] Hoffmann, Gabriel M., Haomiao Huang, Steven L. Waslander, and Claire J. Tomlin. *Quadrotor helicopter flight dynamics and control: Theory and experiment?* In Proc. of the AIAA Guidance, Navigation, and Control Conference, vol. 2. 2007.
- [3] Michael, Nathan, Shaojie Shen, Kartik Mohta, Yash Mulgaonkar, Vijay Kumar, Keiji Nagatani, Yoshito Okada et al. *Collaborative mapping of an earthquake-damaged building via ground and aerial robots.* Journal of Field Robotics 29, no. 5 (2012): 832-841.
- [4] Chambers, Andrew, Supreeth Achar, Stephen Nuske, Jrn Rehder, Bernd Kitt, Lyle Chamberlain, Justin Haines, Sebastian Scherer, and Sanjiv Singh. *Perception for a river mapping robot.* In Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, pp. 227-234. IEEE, 2011.
- [5] Raffo, Guilherme V., Manuel G. Ortega, and Francisco R. Rubio. *Backstepping/nonlinear H₂ control for path tracking of a quadrotor unmanned aerial vehicle.* In American Control Conference, 2008, pp. 3356-3361. IEEE, 2008.
- [6] Morgan, Daniel, Giri P. Subramanian, Soon-Jo Chung, and Fred Y. Hadaegh. *Swarm Assignment and Trajectory Optimization Using Variable-Swarm, Distributed Auction Assignment and Sequential Convex Programming.* (2016).
- [7] Mallikarjunan, Srinath, Bill Nesbitt, Evgeny Kharisov, Enric Xargay, Naira Hovakimyan, and Chengyu Cao. *L1 adaptive controller for attitude control of multirotors.* In AIAA Guidance, Navigation and Control Conference, Minneapolis, AIAA-2012-4831. 2012.
- [8] Diao, Chen, Bin Xian, Qiang Yin, Wei Zeng, Haotao Li, and Yungao Yang. *A nonlinear adaptive control approach for quadrotor UAVs.* In Control Conference (ASCC), 2011 8th Asian, pp. 223-228. IEEE, 2011.
- [9] Ritz, Robin, Markus Hehn, Sergei Lupashin, and Raffaello D'Andrea. *Quadrocopter performance benchmarking using optimal control.* In Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, pp. 5179-5186. IEEE, 2011.
- [10] Mellinger, Daniel, Nathan Michael, and Vijay Kumar. *Trajectory generation and control for precise aggressive maneuvers with quadrotors.* The International Journal of Robotics Research (2012): 0278364911434236.
- [11] Dierks, Travis, and Sarangapani Jagannathan. *Output feedback control of a quadrotor UAV using neural networks.* Neural Networks, IEEE Transactions on 21, no. 1 (2010): 50-66.
- [12] Bandyopadhyay, Saptarshi, Soon-Jo Chung, and Fred Y. Hadaegh. *Nonlinear Attitude Control of Spacecraft with a Large Captured Object.* Journal of Guidance, Control, and Dynamics (2016): 1-16.
- [13] *Crazyflie* <http://www.bitcraze.io> [Retrieved 15 April 2016]
- [14] Craig, John J. *Introduction to robotics: mechanics and control.* Vol. 3. Upper Saddle River: Pearson Prentice Hall, 2005.

- [15] *GML C++ Camera Calibration Toolbox* <http://graphics.cs.msu.ru/en/node/909> [Retrieved 13 Jan 2016]
- [16] *Camera Calibration Toolbox for Matlab* http://www.vision.caltech.edu/bouguetj/calib_doc [Retrieved 15 April 2016]
- [17] *UIUC Airfoil Coordinates Database* [Retrieved 13 Jan 2016] http://m-selig.ae.illinois.edu/ads/coord_database.html [Retrieved 24 Aug 201]