

© 2016 by Mingyo Seo. All rights reserved.

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

BY

MINGYO SEO

THESIS

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

Urbana, Illinois

Advisor:

Professor Seth Hutchinson  
Associate Professor Soon-Jo Chung

# Abstract

A quadrotor, a UAV equipped with four rotors, has an advantage of its maneuver, and has been used for various purposes. For this advantage, many researchers have studied a quadrotor's operation. An attitude control system for a quadrotor is one of the most important parts in order to improve the quadrotor's performance.

This thesis developed a quadrotor testbed and applied a model-based nonlinear attitude control, originally designed for a space craft, to the quadrotor. In order to implement a nonlinear attitude control system, the dynamic model of the quadrotor is studied and the quadrotor's physical properties are characterized based on the model. Then, stability and agility of the nonlinear attitude control are validated by both simulations and experiments and its performance is compared with one of an conventional PID attitude control.

Finally, this thesis proposes a computationally-efficient position estimator for the quadrotor's operation. The position estimator detects visual markers of an image from the camera, and computes the quadrotor's position. The position estimator was also evaluated by experiments.

# Acknowledgments

The author wishes to express sincere gratitude to Prof. Soon-Jo Chung and Prof. Seth Hutchinson, for their advice and guidance throughout this work. The author gratefully acknowledges the contributions of Elias Waddington to the hardware setup and the experiments reported in Chapters 2 and 4. Chapter 5 was conducted as joint work with Lingyu Ma, and the author would like to express thanks to him for his insightful discussions. Finally, special thanks go to his family, Nari Kim, Taito Nabeshima, Wookyum Kim, Daniel Cho, Shenyu Liu, JCT members, and CEA members for their support during the process of earning his Master's degree.

# Table of Contents

<b>List of Tables . . . . .</b>	<b>vi</b>
<b>List of Figures . . . . .</b>	<b>vii</b>
<b>Chapter 1 Introduction . . . . .</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Related Works . . . . .	1
1.3 Motivation . . . . .	2
1.4 Objective . . . . .	3
1.5 Outline . . . . .	3
<b>Chapter 2 System and Hardware Description . . . . .</b>	<b>5</b>
2.1 Hardware Requirements . . . . .	5
2.2 Hardware Specification . . . . .	6
2.2.1 Frame . . . . .	6
2.2.2 Actuator Motors . . . . .	7
2.2.3 Propeller . . . . .	9
2.2.4 Autopilot . . . . .	9
2.2.5 Companion Single-Board Computer . . . . .	9
2.2.6 Camera . . . . .	10
2.2.7 Wireless Communication Interface . . . . .	11
2.2.8 Ground Control Station . . . . .	11
2.3 System Overview . . . . .	13
2.4 Physical Properties . . . . .	18
<b>Chapter 3 Control System . . . . .</b>	<b>21</b>
3.1 Control System Overview . . . . .	21
3.2 Coordinate System . . . . .	21
3.3 Outer-Loop Controller . . . . .	24
3.3.1 Dynamics Description . . . . .	24
3.3.2 Computation of Desired Velocity . . . . .	26
3.3.3 Computation of Desired Thrust and Attitude . . . . .	26
3.4 Inner-Loop Controller . . . . .	28
3.4.1 Dynamics Description . . . . .	28
3.4.2 Computation of Derivative of the Desired Attitude . . . . .	29
3.4.3 Computation of Desired Torque . . . . .	31
3.5 Motor Control . . . . .	34
3.5.1 Mapping Attitude and Thrust Control into Motor Speeds . . . . .	34
3.5.2 Approximation of the Aerodynamic Coefficient Matrix . . . . .	37
3.5.3 Motor Control . . . . .	38

<b>Chapter 4 Evaluation of the Control System</b>	<b>42</b>
4.1 Linear Attitude Control for Comparison	42
4.2 Simulation Study for Inner-loop Controller	43
4.2.1 Simulation Setup	43
4.2.2 Results	44
4.2.3 Discussion	44
4.3 Flight Experiment	47
4.3.1 Motion Capture Arena	47
4.3.2 Motion Capture System	48
4.3.3 Experiment Setup	50
4.3.4 Experiment Results	50
4.3.5 Discussion	51
<b>Chapter 5 Vision-based Position Estimator</b>	<b>60</b>
5.1 Camera Calibration	60
5.2 Marker Detection	62
5.3 Position Estimation	64
5.4 Evaluation Experiment	65
5.4.1 Experimental Setup	65
5.4.2 Results	66
5.4.3 Discussion	66
<b>Chapter 6 Conclusion</b>	<b>69</b>
<b>References</b>	<b>71</b>

# List of Tables

2.1 Dimension and Mass of Each Component . . . . .	6
2.2 Geometry of APC 10 × 4.7 Slow Flyer [16] . . . . .	9

# List of Figures

1.1 A Crazyflie Nano Quadrotor with a Downward Camera . . . . .	3
2.1 Illustration of the Quadrotor Frame and the Rotors . . . . .	7
2.2 Brushless DC Motor . . . . .	7
2.3 Electric Speed Control (ESC) . . . . .	8
2.4 11.1V 2700mAh LiPo Battery . . . . .	8
2.5 APC 10 × 4.7 Slow Flyer [16] . . . . .	9
2.6 Pixhawk Autopilot . . . . .	10
2.7 Odroid XU4 Single-board Computer . . . . .	10
2.8 USB Camera . . . . .	11
2.9 3DR radio V2 Telemetry . . . . .	12
2.10 AR610 6-channel DSMX Aircraft Receiver . . . . .	12
2.11 Quadrotor Testbed . . . . .	13
2.12 Description of the Quadrotor Assemble . . . . .	14
2.13 System Architecture of Offboard Control . . . . .	15
2.14 System Architecture of Onboard Control . . . . .	16
2.15 CAD model of the Quadrotor (upper side) . . . . .	20
2.16 CAD model of the Quadrotor (under side) . . . . .	20
3.1 Illustration of the Control System. . . . .	22
3.2 Illustration of the Inertial and Body Frames. . . . .	23
3.3 Thrust and Torque Generated by Each Rotor . . . . .	35
3.4 Comparison of experimental and approximated thrust coefficient . . . . .	39
3.5 Comparison of experimental and approximated power coefficient . . . . .	39
3.6 Change of RPM output corresponding to PWM command . . . . .	41
4.1 Simulation Result (Case 1) . . . . .	45
4.2 Simulation Result (Case 2) . . . . .	45
4.3 Simulation Result (Case 3) . . . . .	45
4.4 Simulation Result (Case 4) . . . . .	46
4.5 Simulation Result (Case 5) . . . . .	46
4.6 Simulation Result (Case 6) . . . . .	46
4.7 Simulation Result (Case 7) . . . . .	47
4.8 Motion Capture Arena of the Intelligent Robotics Laboratory . . . . .	48
4.9 Vicon Motion Capture Cameras . . . . .	49
4.10 Vicon Motion Capture Interface . . . . .	49
4.11 Experiment Result (Case 1) . . . . .	52
4.12 Experiment Result (Case 2) . . . . .	53
4.13 Experiment Result (Case 3) . . . . .	54
4.14 Experiment Result (Case 4) . . . . .	55
4.15 Experiment Result (Case 5) . . . . .	56
4.16 Experiment Result (Case 6) . . . . .	57

4.17 Experiment Result (Case 7) . . . . .	58
5.1 Camera Calibration (Left: Raw Image, Right: Restored Image) . . . . .	61
5.2 Gaussian-filtered and Thresholded Image . . . . .	62
5.3 Restoration of Borders . . . . .	63
5.4 Final Result of Marker Detection . . . . .	63
5.5 Schematic Diagram of the Position Estimation . . . . .	64
5.6 Image Markers for the Evaluation . . . . .	66
5.7 Experiment Result of the Position Estimator (x-axis Position) . . . . .	67
5.8 Experiment Result of the Position Estimator (y-axis Position) . . . . .	67
5.9 Experiment Result of the Position Estimator (z-axis Position) . . . . .	67

# Chapter 1

## Introduction

### 1.1 Background

An unmanned aerial vehicle (UAV) is defined as an autonomous aircraft that can fly without a human pilot on board [1]. A quadrotor is a 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 translation motion is coupled with the quadrotor attitude and thrust since the thrust occurs in only the perpendicular direction of the quadrotor orientation; the direction of the translation is determined by the orientation, and the magnitude of the translation is determined by the thrust. Therefore, the quadrotor controls its translation by altering its attitude and has a capacity of both rotation and translation maneuvers. Due to the advantage, quadrotors have been used for various purposes and performed tasks in environments that are nearly impossible to reach as a 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 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 rotational effects 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 improvement of quadrotors' performance, various research on advanced control systems for a quadrotor have been conducted actively. Raffo et al. applied a nonlinear  $\mathcal{H}_\infty$  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 a trajectory generator for a quadrotor's aggressive maneuvers, and validated the quadrotor's performance by experiments [10]. In addition, Dierksand and Jagannathan suggested a control scheme that uses control inputs computed by neural networks [11].

Here, we develop a model based attitude control system focused on addressing the nonlinearity of quadrotor dynamics in order to improve agility and stability.

### 1.3 Motivation

Due to high nonlinearities of quadrotor dynamics, application of advanced controllers is expected to improve performance of the quadrotor, compared with conventional controllers, such as PID and Linear-Quadratic Regulator. Studies of quadrotor dynamics are expected to increase the robustness of the control systems.

The Aerospace Robotics and Control Laboratory at the University of Illinois had developed quadrotor platforms based on Crazyflie 1.0 to validate performance of nonlinear controllers [6]. Crazyflie is a open-source nano quadrotor system developed by Bitcraze [12]. A Crazyflie quadrotor is about 19 g weight and each leg is about 6 cm length. However, due to the quadrotor's physical constraints, such as its load limit and sensor errors, its application is limited. Especially, in order to develop a vision-based position control system and other high-level functions for the quadrotor, only 8 g of load is allowed for sensors, equipments, and mount for them, and therefore, choice of sensors and algorithms is limited. A Crazyflie quadcopter with a camera and mount is shown in Figure 1.1.

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



Figure 1.1: A Crazyflie Nano Quadrotor with a Downward Camera

for various purposes, and therefore, this research aims for developing a new micro quadrotor system.

## 1.4 Objective

In this research, we aim for the application and evaluation of the nonlinear attitude control, suggested by Bandyopadhyay and Chung, that was originally designed for a spacecraft [13]. Also, we validate the proposed approach of inner-loop nonlinear attitude control and linear outer-loop position control with feedforward cancellation used in [6]. In order to implement the control system, we developed a new micro quadrotor system. The quadrotor dynamics is modeled and characterized based on experimental data. Performance of the quadrotor with the nonlinear attitude control is evaluated by both flight experiments and simulations and compared with conventional PID attitude control. In addition, we suggest a computationally-efficient 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. Chapter 2 also describes the hardware specification of each component and the physical properties of the quadrotor. Chapter 3 focuses on the dynamic model and the control system. To be more specific, Section 3.1 presents the control system overview. 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 rotation and the nonlinear attitude control. The proof

of the stability of the nonlinear control is also stated in this section. In Section 3.5, a model of DC motors is introduced and an open-loop control of the motors is described. Chapter 4 presents the evaluation of the control system that is stated in Chapter 3. In particular, Section 4.1 describes an PID attitude control that is compared with the nonlinear control system. Section 4.2 discusses numerical simulation results of the nonlinear and PID attitude control of the quadrotor. In Section 4.3, the results of flight experiments are presented with various inputs. Chapter 5 introduces a computationally-efficient method to estimate the quadrotor position with given image data of visual markers, and discusses 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 or a preprogrammed autonomous operation. In order to achieve implementation of a dynamic model based nonlinear attitude controller, the quadrotor must meet the following requirements:

- All the physical parameters used in the nonlinear control system must be characterized.
- The quadrotor must provide full access of its control and measurement, and they must be repro- grammable by a user.
- The quadrotor must use wireless communication methods to access sensor values and setpoints.
- The quadrotor must be able to execute an algorithm for autonomous operation either offboard or onboard.

To satisfy these requirements, an offboard control system is used for the validation of the nonlinear attitude control and the evaluation of its performance. The system consists of an autopilot controller, actuators, and wireless communication interface including radio telemetry and a backup RC controller with human pilot. PX4, an open-source autopilot system, is installed in the autopilot, and it runs with a nonlinear controller. Also, all the physical parameters used in the control system are measured and characterized. They are described in Section 2.2, 2.3 and 3.5.

In addition to the development of the control system for the quadrotor, in order to implement a vision-based position estimator, the quadrotor must have sufficient computing power to process visual data. The

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 there would no longer be restrictions on the size, weight, and power. However, this method increases latency. Since quadrotors are agile systems, low latency computation is preferred, and therefore, an onboard companion computer is added to the quadrotor system in order to implement the vision-based position estimator.

## 2.2 Hardware Specification

The dimension and mass of each component are shown at Table 2.1.

Components	Dimension [ $10^{-3}\text{m}$ ]	Mass [g]
Brushless DC motor	$\varnothing 28 \times 30$	62
Electric Speed Control	$48 \times 25 \times 5$	21
LiPo Battery	$129 \times 39 \times 20$	216
Autopilot Controller	$82 \times 50 \times 16$	38
Onboard Computer	$82 \times 58 \times 22$	60
USB Camera	$38 \times 38 \times 24$	15
RC Receiver	$32 \times 22 \times 12$	11
Radio Telemetry	$160 \times 24 \times 12$	24

Table 2.1: Dimension and Mass of Each Component

### 2.2.1 Frame

In the quadrotor system, a glass fiber X-shaped frame is used, as shown in Figure 2.1. The length of the quadrotor's arm is 0.225 m and the weight is 0.270 kg. 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 dynamic model and decrease sensor errors of the quadrotor's rotational motion, the motors and the autopilot are installed on the same plane of the quadrotor. Electronic Speed Controls (ESCs) are installed on each leg symmetrically.

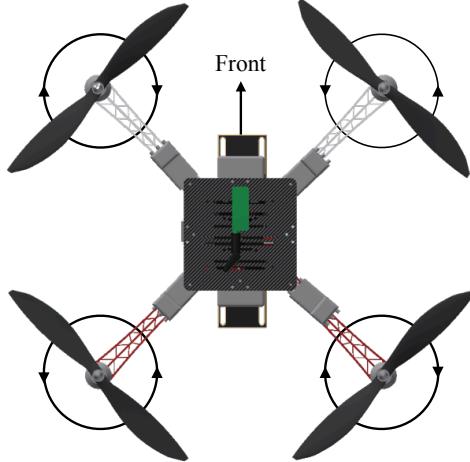


Figure 2.1: Illustration of the Quadrotor Frame and the Rotors

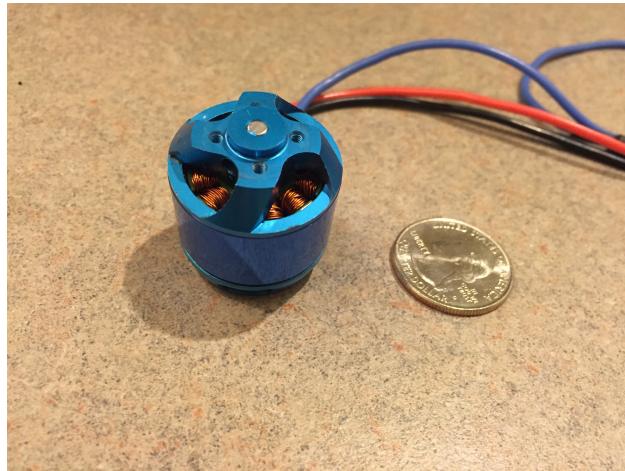


Figure 2.2: Brushless DC Motor

### 2.2.2 Actuator Motors

A quadrotor controls its attitude and thrust by independent inputs of its four motors. AC2830-358 850kV Brushless DC motors developed by 3D Robotics are used in this research [14]. The maximum power of the motors is 187 W. Since we use brushless DC motors, the quadrotor requires ESCs to run the motors. ESCs are also developed by 3D Robotics [15]. 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 3.5. Each motor spins a propeller, providing thrust and moment to each motor.



Figure 2.3: Electric Speed Control (ESC)



Figure 2.4: 11.1V 2700mAh LiPo Battery



Figure 2.5: APC  $10 \times 4.7$  Slow Flyer [16]

### 2.2.3 Propeller

In the quadrotor, two pairs of propellers, APC  $10 \times 4.7$  Slow Flyer (SF) and Slow Flyer Pusher (SFP) propellers, developed by APC Propellers Company, are used [17]. Both SF and SFP propellers have same geometries as shown at table 2.2 but are mirror symmetric to each other. SF propellers are designed to rotate clockwise and SFP propellers are designed to rotate counter-clockwise. As shown in 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 0.012 kg.

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

Table 2.2: Geometry of APC  $10 \times 4.7$  Slow Flyer [16]

### 2.2.4 Autopilot

A quadrotor is typically equipped with a microcontroller autopilot that is controlled by high level commands, such position or attitude setpoints from an RC transmitter. Based on the current state and setpoint commands, the controller computes desired actuation 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 [18]. We used a 3D Robotics Pixhawk microprocessor to run our PX4 code [19]. Pixhawk contains all the necessary sensors, computation power, and input/output interfaces. In addition to an accelerometer and a gyroscope, the Pixhawk has a magnetometer and barometer to measure altitude and attitude changes [20].

### 2.2.5 Companion Single-Board 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,



Figure 2.6: Pixhawk Autopilot

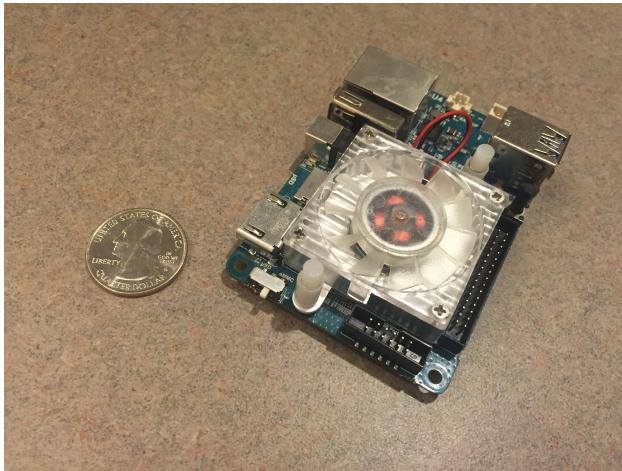


Figure 2.7: Odroid XU4 Single-board Computer

computes the position estimation of the quadrotor and sends the estimation data to the autopilot. The quadrotor uses a single Hardkernel Odroid XU4 [21]. It is 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

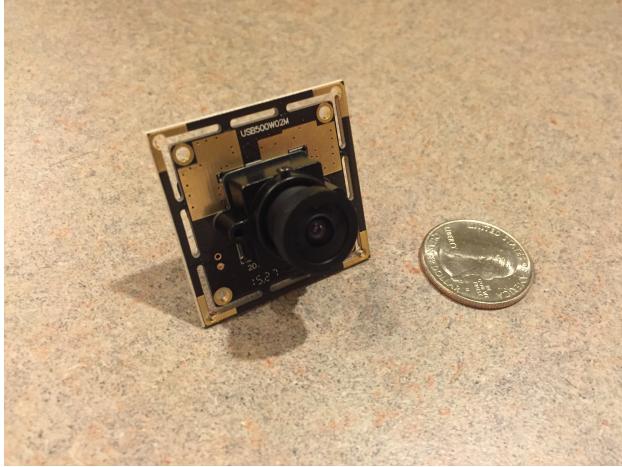


Figure 2.8: USB Camera

2.1mm and has about  $120^\circ$  view angle. The camera sends images of  $640 \times 480$  pixels at a rate of 30 frames per second. This sensor is used for vision-based estimation presented in Chapter 5.

### 2.2.7 Wireless Communication Interface

In the quadrotor system, two different communication methods are used. First, in the case of offboard control, bidirectional radio communication is used to exchange messages between a ground control station and the quadrotor, and unidirectional radio communication is used to send inputs from a human user using an RC transmitter. A pair of 3DR Radio V2 kits, developed by 3D Robotics, was used for the bidirectional radio communication, and an AR610 6-channel DSMX Aircraft Receiver, developed by Spektrum, is used for unidirectional communication [22][23]. The communication for the ground control station uses 915 MHz frequency and the one for the RC transmitter uses 2.4 GHz frequency.

Secondly, in the case of onboard control, messages between the onboard computer and the autopilot are exchanged directly via wired serial communication, and a remote communication interface between the onboard computer and the ground control station is necessary instead. Therefore, a Wi-Fi network is used for communication between the onboard computer and the ground control station. The same communication between the quadrotor and the RC transmitter can be also used as backup.

### 2.2.8 Ground Control Station

The ground control station allows us to send commands and receive data remotely. For offboard control, the ground control station must be connected with a radio telemetry with a proper baud rate to receive messages. The telemetry that is connected to the ground station computer must be paired with the one on



Figure 2.9: 3DR radio V2 Telemetry

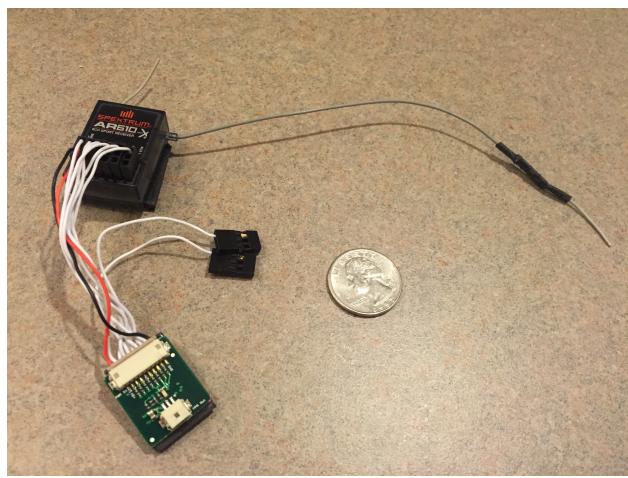


Figure 2.10: AR610 6-channel DSMX Aircraft Receiver

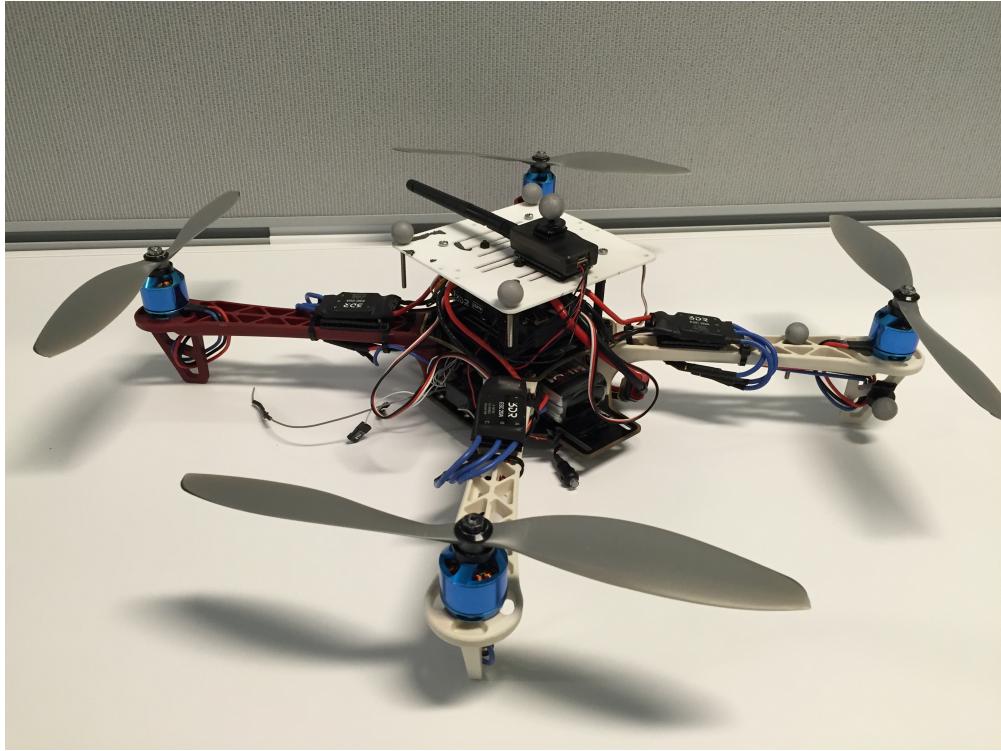


Figure 2.11: Quadrotor Testbed

the quadrotor.

In the case of onboard control, any computer can be used as a ground control system so long as it is on the same Wi-Fi network since the quadrotor system uses the Wi-Fi network to communicate. The companion computer can be accessed using any wireless interface, such as SSH. Through this interface, a user can use the command console on the onboard companion computer remotely, and the quadrotor's operation can be started.

### 2.3 System Overview

The complete assembly of the quadrotor is shown in Figure 2.11 and 2.12. An overview of the offboard systems of the quadrotor is shown in Figure 2.13 and one of the onboard system for vision-based estimation is shown in Figure 2.14.

In order to operate the quadrotor offboard, the quadrotor must be connected to the ground control station directly through the radio telemetries. A preprogrammed script at the ground control station sends Micro Air Vehicle Link (MAVLink) messages with control commands via radio communication between the radio telemetries stated at Subsection 2.2.8. MAVLink is a protocol that is usually used to communicate UAVs

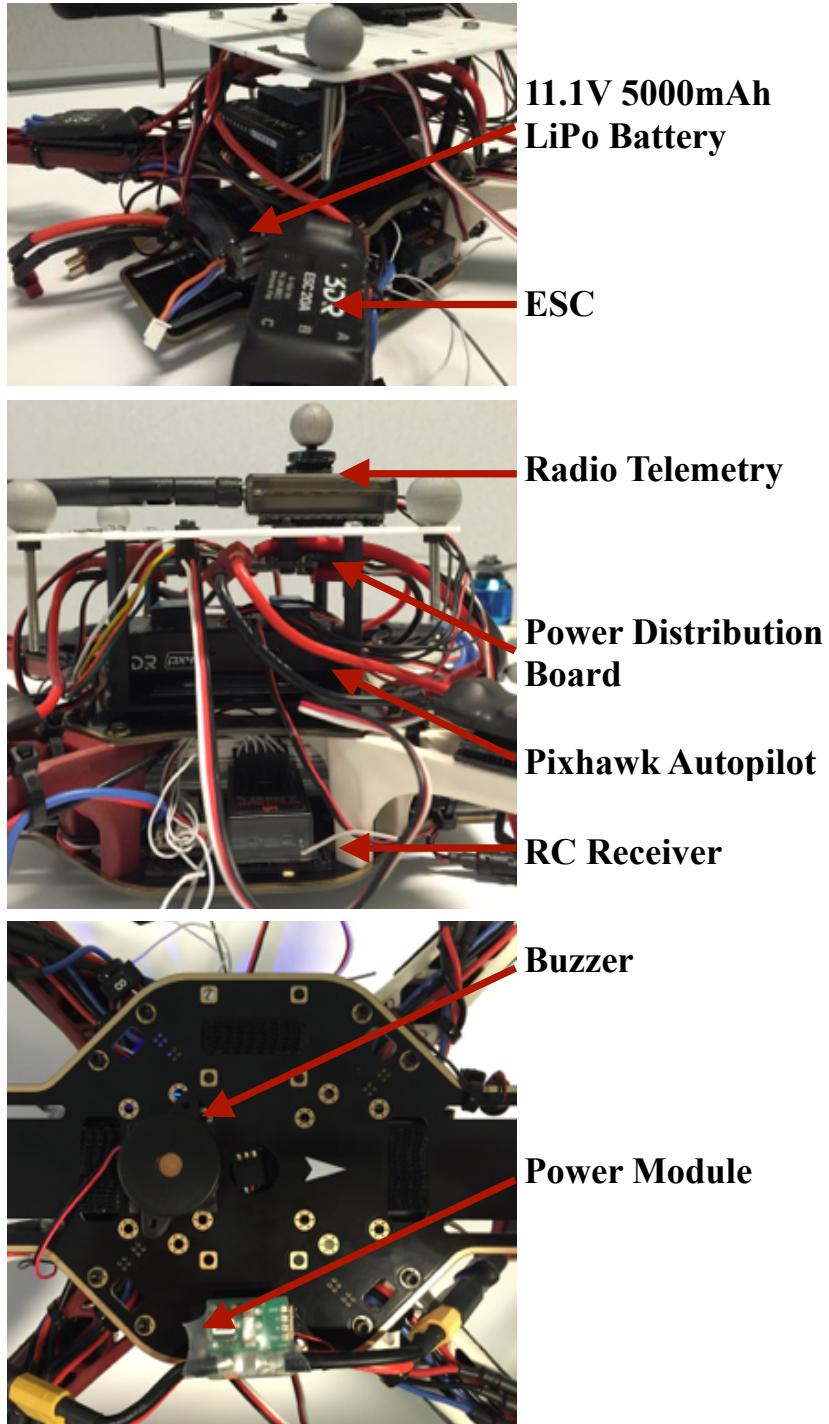


Figure 2.12: Description of the Quadrotor Assemble

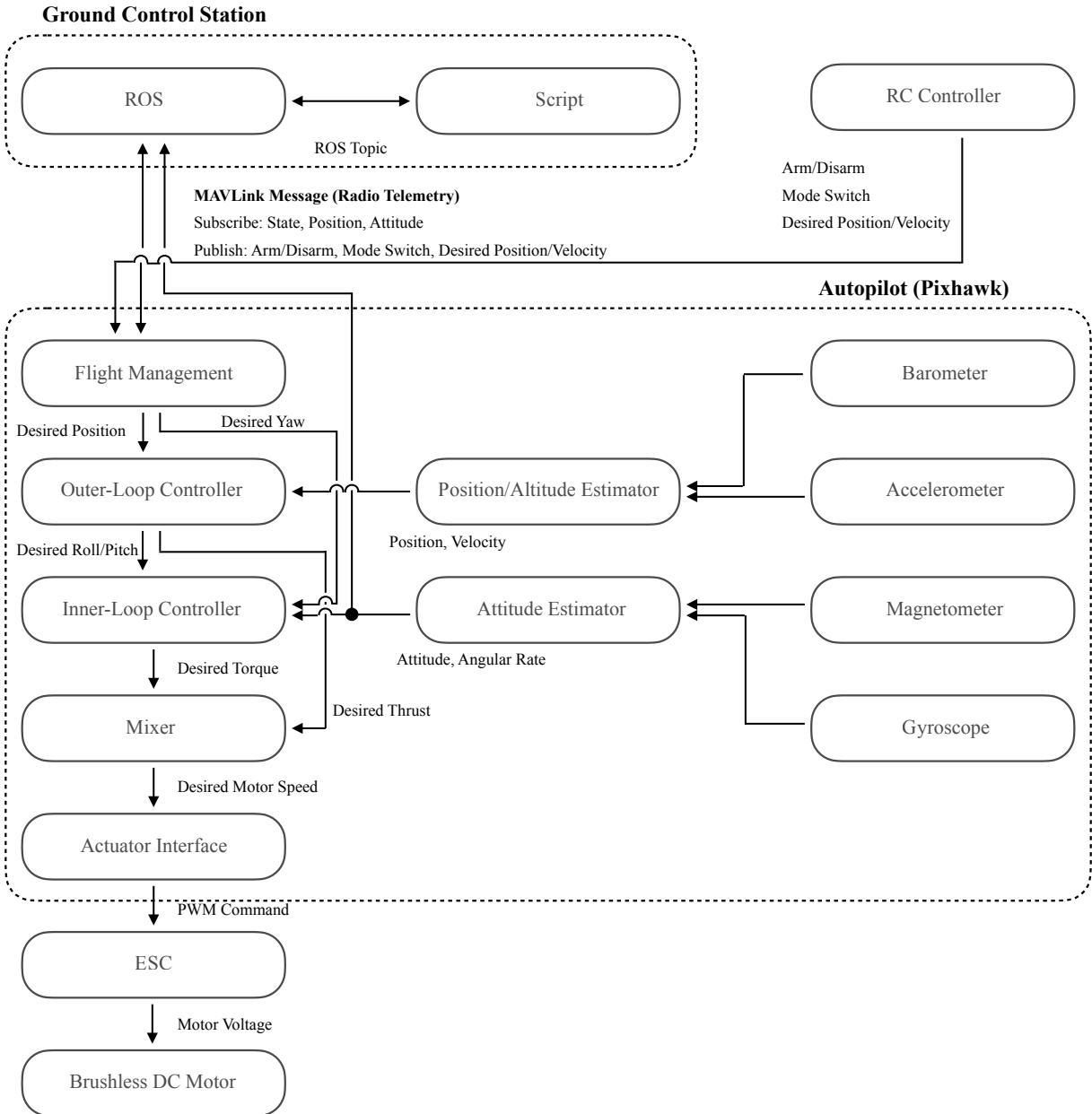


Figure 2.13: System Architecture of Offboard Control

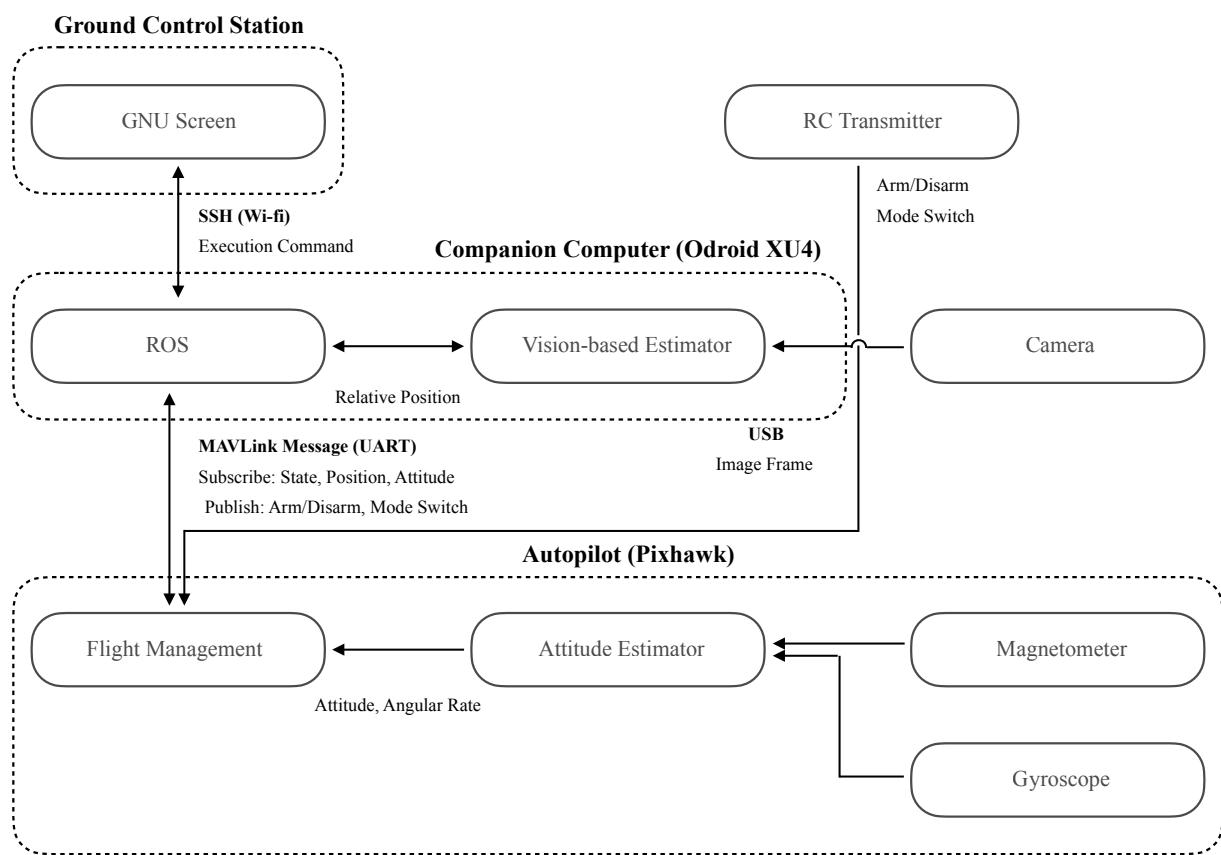


Figure 2.14: System Architecture of Onboard Control

[24]. MAVROS, a ROS package to deal with MAVLink messages, is used to process MAVLink messages [25]. In addition, in order for a human user to send command inputs to the quadrotor, the inputs are sent from the RC transmitter to the RC receiver. The RC receiver is directly connected to the autopilot controller.

On the other hand, the onboard companion computer estimates position information and sends command to the quadrotor, instead of the ground control station. In order to operate the quadrotor onboard, the onboard companion computer must be connected to the ground control station through a Wi-Fi network. Now the companion computer's console is available from the ground control station. Through the GNU SSH interface, the user can start and stop the autonomous operations of the quadrotor. When the autonomous operation starts, the companion computer accesses the autopilot controller and waits until the markers are in view. As soon as the ground facing camera captures the markers, 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 orientation is same as the orientation of the camera and its information can be measured by the magnetometer of the autopilot. MAVLink messages are used via UART serial communication to communicate between the companion computer and the autopilot. Similar to offboard control, MAVROS is used to deal with MAVLink messages.

Once MAVLink messages are transmitted to the autopilot controller, the remaining tasks are the same in both flight control modes. However, in this research, only offboard control is used for the quadrotor flight, and onboard control is used to evaluate an onboard real-time position measurement. Therefore, the quadrotor both receives maneuver commands and exchanges the quadrotor state information in offboard control, but it only exchanges the state information in onboard control.

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, and 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 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 computes desired torques. The mixer module receives 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 1.404 kg. 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 CAD model. Due to the convenience of CAD and the difficulties of installing a precise 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. The moment of inertia computed by a CAD model is given as below,

$$\begin{aligned} J &= \begin{bmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{xy} & J_{yy} & J_{yz} \\ J_{xz} & J_{yz} & J_{zz} \end{bmatrix} \\ &= \begin{bmatrix} 1.206 & -0.003318 & -0.008680 \\ -0.003318 & 1.285 & -0.001466 \\ -0.008680 & -0.001466 & 2.347 \end{bmatrix} \times 10^{-2} [kgm^2] \end{aligned} \quad (2.1)$$

The diagonal elements  $J_{xx}$ ,  $J_{yy}$ ,  $J_{zz}$  are at least 150 times greater than any cross moments of inertia, the moment of inertia  $J$  is approximated as a positive diagonal matrix.

$$J \approx \begin{bmatrix} 1.206 & 0 & 0 \\ 0 & 1.285 & 0 \\ 0 & 0 & 2.347 \end{bmatrix} \times 10^{-2} [kgm^2] \quad (2.2)$$

This implies that the quadrotor has a nearly symmetric structure about all axes. The components that have comparatively great mass, such as DC motors, are installed symmetrically. Even if the components that have small mass, such as the radio telemetry, are located asymmetrically, they have little effect on the moment of inertia. In addition, the battery is set close to the center of inertia, so its effect on the moment

of inertia is expected to be small.



Figure 2.15: CAD model of the Quadrotor (upper side)



Figure 2.16: 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 nonlinear attitude control system based on a 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 in Figure 3.1. First, the outer-loop controller receives desired position and desired attitude. Then, the outer-loop controller receives current state data from the sensors, and computes desired attitude and thrust corresponding to position errors of the quadrotor. The 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 quadrotor's attitude and angular velocity 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 angular velocity is measured by the internal gyroscope. Finally, the mixer, an open-loop motor controller, receives the desired thrust from the outer-loop controller and the desired torque from the inner-loop controller. Then, the mixer maps the desired thrust and torque into desired motor speeds, and estimates 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 state and motion representation.[26] 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, an 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

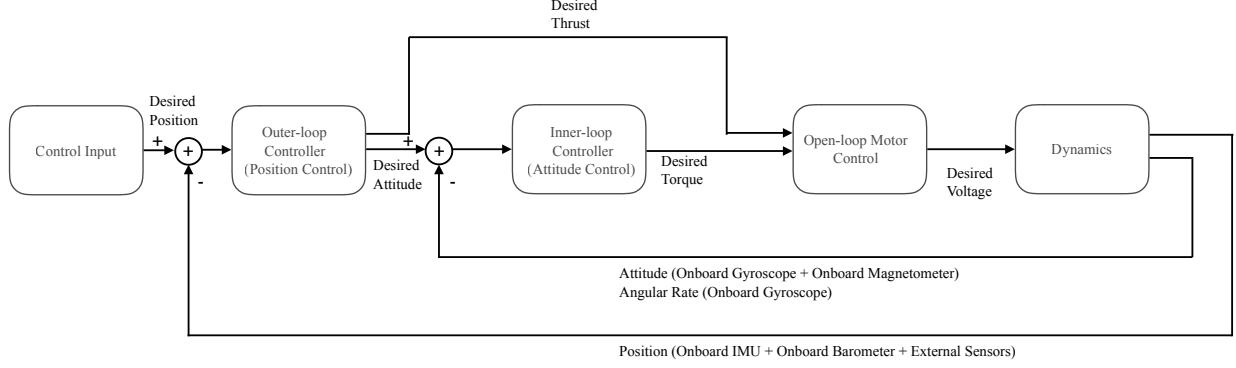


Figure 3.1: Illustration of the Control System.

the ground. The origin can be arbitrarily 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 are 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 the autopilot controller, approximately at the center of mass.

Using the above coordinate systems, the quadrotor motion is represented completely. The mapping of the origin in the body frame into the inertial frame represents the quadrotor position. The orientation of the body frame is 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 [27]. 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  $\psi$ . Rotation of the y-axis by a pitch angle  $\theta$ , and the z-axis by a roll angle  $\phi$  are followed in order. The transformation matrix of an orientation from the body frame into the inertial frame is given as,

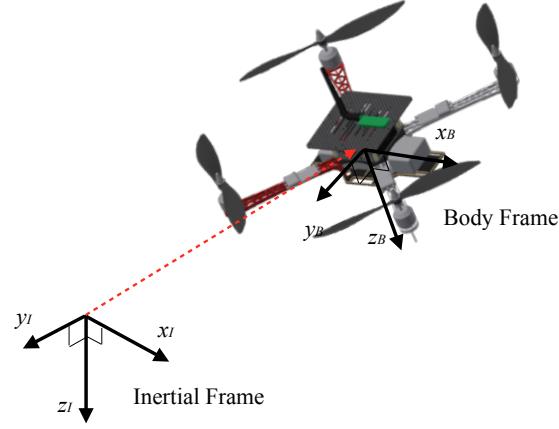


Figure 3.2: Illustration of the Inertial and Body Frames.

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

and the inverse transformation is given as,

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

### 3.3 Outer-Loop Controller

The outer-loop controller calculates desired attitude of the quadrotor, 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  $\mathbf{F} = (\bar{X}, \bar{Y}, \bar{Z})$  be the quadrotor's thrust specified in the inertial frame. Then, the equation of motion is given as,

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

where  $m$  is the mass of the quadrotor,  $\mathbf{r} = (x, y, z)$  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 the body frame into the 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  $(\ddot{x}, \ddot{y}, \ddot{z})$  is computed as,

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

The sensors are fixed on the quadrotor, and therefore, it is necessary to formulate the relation of the quadrotor 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 the below equation [28].

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

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  $\mathbf{v}_d = (v_{xd}, v_{yd}, v_{zd})$  in the inertial frame, with respect to the error of desired position  $\mathbf{r}_d = (x_d, y_d, z_d)$  and observed position  $\mathbf{r} = (x, y, z)$ . The desired velocity is generated by P control. Desired control along each axis of the inertial frame is given as,

$$\mathbf{v}_d = K_{pos}(\mathbf{r}_d - \mathbf{r}) \quad (3.9)$$

where  $K_{pos}$  is a diagonal gain matrix of the controller. In order to prevent abnormal performance of the quadrotor, we set an 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. A PID position control is given as the below equation [6][29].

$$\begin{aligned} \bar{X}_d &= m \left( \ddot{x}_d + K_{Px}(x_d - x) + K_{Dx}(\dot{x}_d - \dot{x}) + K_{Ix} \int_{t_0}^t (x_d - x) dt \right) \\ \bar{Y}_d &= m \left( \ddot{y}_d + K_{Py}(y_d - y) + K_{Dy}(\dot{y}_d - \dot{y}) + K_{Iy} \int_{t_0}^t (y_d - y) dt \right) \\ \bar{Z}_d &= m \left( \ddot{z}_d + K_{Pz}(z_d - z) + K_{Dz}(\dot{z}_d - \dot{z}) + K_{Iz} \int_{t_0}^t (z_d - z) dt \right) \end{aligned} \quad (3.10)$$

$K_{Px}$ ,  $K_{Py}$ ,  $K_{Pz}$  are the 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.

The Pixhawk autopilot has been developed to support both indoor and outdoor flight. However, outdoor flight puts priority on velocity control, and considering outdoor environments, the control law of Equation (3.10) may not work properly without precise position measurement. Therefore, a PID controller in terms of the quadrotor velocity is used alternatively in the autopilot system. The desired thrust of the alternative PID controller is given as,

$$\begin{aligned}\ddot{x} &= \frac{1}{m} \left( k_{Px}(v_{xd} - \dot{x}) + k_{Dx}(\dot{v}_{x,d} - \ddot{x}) + k_{Ix} \int_{t_0}^t (v_{xd} - \dot{x}) dt \right) \\ \ddot{y} &= \frac{1}{m} \left( k_{Py}(v_{yd} - \dot{y}) + k_{Dy}(\dot{v}_{y,d} - \ddot{y}) + k_{Iy} \int_{t_0}^t (v_{yd} - \dot{y}) dt \right) \\ \ddot{z} &= \frac{1}{m} \left( k_{Pz}(v_{zd} - \dot{z}) + k_{Dz}(\dot{v}_{z,d} - \ddot{z}) + k_{Iz} \int_{t_0}^t (v_{zd} - \dot{z}) dt \right)\end{aligned}\quad (3.11)$$

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  $\bar{X}_d$ ,  $\bar{Y}_d$ ,  $\bar{Z}_d$  can be computed as,

$$\begin{aligned}\bar{X}_d &= k_{Px}(v_{xd} - \dot{x}) + k_{Dx}(\dot{v}_{x,d} - \ddot{x}) + k_{Ix} \int_{t_0}^t (v_{xd} - \dot{x}) dt \\ \bar{Y}_d &= k_{Py}(v_{yd} - \dot{y}) + k_{Dy}(\dot{v}_{y,d} - \ddot{y}) + k_{Iy} \int_{t_0}^t (v_{yd} - \dot{y}) dt \\ \bar{Z}_d &= mg + k_{Pz}(v_{zd} - \dot{z}) + k_{Dz}(\dot{v}_{z,d} - \ddot{z}) + k_{Iz} \int_{t_0}^t (v_{zd} - \dot{z}) dt\end{aligned}\quad (3.12)$$

In this research, Equation (3.12) is used for outer-loop control, but the control law of Equation of (3.10) is more appropriate for indoor flight with the values of  $(\dot{x}_d, \dot{y}_d, \dot{z}_d)$  and  $(\ddot{x}_d, \ddot{y}_d, \ddot{z}_d)$ . In the control of Equation (3.12), 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 position, the control law becomes similar to PD control in terms of the quadrotor position.

As described above, the direction of the thrust is defined by the orientation of the quadrotor. From Equations (3.5), desired roll  $\phi_d$ , pitch  $\theta_d$ , and yaw  $\psi_d$  are computed as,

$$\begin{aligned}\phi_d &= \arcsin \frac{\bar{X}_d \sin \psi_d - \bar{Y}_d \cos \psi_d}{F_d} \\ \theta_d &= \arctan \frac{\bar{X}_d \cos \psi_d + \bar{Y}_d \sin \psi_d}{\bar{Z}_d}\end{aligned}\quad (3.13)$$

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

$$\begin{aligned}F_d &= |\mathbf{F}_d| \\ &= \sqrt{\bar{X}_d^2 + \bar{Y}_d^2 + \bar{Z}_d^2}\end{aligned}\quad (3.14)$$

$\bar{X}_d$ ,  $\bar{Y}_d$ , and  $\bar{Z}_d$  are computed from Equation (3.12). Therefore, we can find that the desired roll  $\phi_d$  and pitch  $\theta_d$  are determined by the given desired yaw  $\psi_d$ .

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

$$\mathbf{u} = \frac{d}{dt}(J\boldsymbol{\omega}) = J\dot{\boldsymbol{\omega}} - \text{Skew}(J\boldsymbol{\omega})\boldsymbol{\omega} \quad (3.15)$$

where  $J$  is the inertia matrix of the quadrotor [30].  $\text{Skew}(\mathbf{a})$  is a skew-symmetric matrix, defined as,

$$\text{Skew}(\mathbf{a}) = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \quad (3.16)$$

By considering the symmetry of the quadrotor along each axis of body 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  $J$  is approximated as a positive diagonal matrix.

$$J \approx \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} \quad (3.17)$$

Since the quadrotor attitude  $\boldsymbol{\eta} = (\phi, \theta, \psi)$  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,

$$\begin{aligned}
\begin{bmatrix} p \\ q \\ r \end{bmatrix} &= \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + R_{x,\phi}^T \left( \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R_{y,\theta}^T \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \right) \\
&= \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \left( \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \right) \quad (3.18) \\
&= \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}
\end{aligned}$$

Define a transform matrix  $Z$  from the body frame into the inertial frame with respect to the quadrotor attitude  $\eta$  as,

$$Z = Z(\eta) = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix}^{-1} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \quad (3.19)$$

Then, from the transformation of coordinate systems, the derivative of the attitude  $\dot{\eta}$  can be written with respect to  $\eta$  and  $\dot{\omega}$  as,

$$\dot{\eta} = Z\omega \quad (3.20)$$

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

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

### 3.4.2 Computation of Derivative of the Desired Attitude

The inner-loop controller receives desired attitude  $\eta_d = (\phi_d, \theta_d, \psi_d)$  from the outer-loop controller, and desired angular velocity  $\omega_d$  is computed from it. From Equation (3.20), the desired angular velocity is given from the controller as,

$$\boldsymbol{\omega}_d = Z^{-1} \dot{\boldsymbol{\eta}}_d \quad (3.22)$$

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 in Equation (3.13). From Equation (3.13),  $\dot{\boldsymbol{\eta}}_d$  is computed as,

$$\begin{aligned}\dot{\phi}_d &= \frac{1}{\sqrt{1 - c_1^2}} \left( c_2 \dot{\psi} + \frac{\sin \psi}{F_d} \dot{\bar{X}}_d - \frac{\cos \psi}{F_d} \dot{\bar{X}}_d - \frac{c_1 \dot{F}_d}{F_d^2} \right) \\ \dot{\theta}_d &= \frac{1}{1 + c_2^2} \left( -c_1 \dot{\psi} + \frac{\cos \psi}{\bar{Z}_d} \dot{\bar{X}}_d + \frac{\sin \psi}{\bar{Z}_d} \dot{\bar{X}}_d + -\frac{c_2 \dot{\bar{Z}}_d}{\bar{Z}_d^2} \right)\end{aligned} \quad (3.23)$$

where  $c_1$  and  $c_2$  are defines as,

$$\begin{aligned}c_1 &= \frac{\bar{X}_d \sin \psi - \bar{Y}_d \cos \psi}{F_d} \\ c_2 &= \frac{\bar{X}_d \cos \psi + \bar{Y}_d \sin \psi}{\bar{Z}_d}\end{aligned} \quad (3.24)$$

From geometric relation, the magnitude of the thrust,  $F_d = |\mathbf{F}_d|$ , is written as,

$$F_d = \bar{Z}_d \sec \theta \sec \psi \quad (3.25)$$

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 the below equation.

$$\dot{\phi}_d = \frac{1}{\sqrt{1 - c_1^2}} \left( c_2 \dot{\psi} + \frac{\sin \psi}{F_d} \dot{\bar{X}}_d - \frac{\cos \psi}{F_d} \dot{\bar{X}}_d + \frac{c_1}{F_d^2} \left( \bar{Z}_d (\dot{\phi} \sec \phi \tan \phi + \dot{\theta} \sec \theta \tan \theta) - \dot{\bar{Z}}_d \right) \right) \quad (3.26)$$

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 quadrotor's attitude is controlled manually, the above formulation may not be available. Therefore, an alternative computation method of desired angular velocity that uses a backward differentiation approximation can be applied. Numerically differentiated values usually have noise, and therefore, a low-pass filter is used to reduce noise. In discrete-time, let  $t_k$  be a discrete time and  $\Delta t_k = t_k - t_{k-1}$  be a time step. Let  $u(t_k)$  be a raw value, and  $y(t_k)$  be a filtered value at time  $t_k$ . Then, a first-order low-pass filter, also known as an

exponentially weighted moving average filter, is given as the below equation,

$$y(t_k) = (1 - a_k)u(t_k) - a_k y(t_{k-1}) \quad (3.27)$$

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

$$a_k = \frac{\Delta t_k}{T_f + \Delta t_k} \quad (3.28)$$

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

$$\dot{\boldsymbol{\eta}}_{d,raw}(t_k) = \frac{\boldsymbol{\eta}_d(t_k) - \boldsymbol{\eta}_d(t_{k-1})}{\Delta t_k} \quad (3.29)$$

From Equations (3.27) and (3.29), filtered derivative of the desired attitude is computed as the below equation.

$$\dot{\boldsymbol{\eta}}_d(t_k) = (1 - a_k)\dot{\boldsymbol{\eta}}_{d,raw}(t_k) - a_k \dot{\boldsymbol{\eta}}_d(t_{k-1}) \quad (3.30)$$

In the control system, Equation (3.30) is used as derivative of the desired attitude.

### 3.4.3 Computation of Desired Torque

In this research, the nonlinear attitude control developed by Bandyopadhyay and Chung is used [13]. In the control law, previous to the computation of desired torque, it is necessary to define reference angular velocity. Let  $\boldsymbol{\eta} = (\phi, \theta, \psi)$  be actual attitude of the quadrotor. Define reference attitude  $\dot{\boldsymbol{\eta}}_r$  as,

$$\dot{\boldsymbol{\eta}}_r = \dot{\boldsymbol{\eta}}_d + \Lambda(\boldsymbol{\eta}_d - \boldsymbol{\eta}) \quad (3.31)$$

where  $\Lambda$  is a positive gain matrix as,

$$\Lambda = \begin{bmatrix} \lambda_\phi & 0 & 0 \\ 0 & \lambda_\theta & 0 \\ 0 & 0 & \lambda_\psi \end{bmatrix} > 0 \quad (3.32)$$

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

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

$$\begin{aligned}\dot{\boldsymbol{\omega}}_r &= \dot{\boldsymbol{\omega}}_d + Z^{-1}\Lambda(\dot{\boldsymbol{\eta}}_d - \dot{\boldsymbol{\eta}}) - Z^{-1}\dot{Z}Z^{-1}\Lambda(\boldsymbol{\eta}_d - \boldsymbol{\eta}) \\ &= \dot{\boldsymbol{\omega}}_d + \Lambda((\boldsymbol{\omega}_d - \boldsymbol{\omega}) - Z^{-1}\dot{Z}Z^{-1}(\boldsymbol{\eta}_d - \boldsymbol{\eta}))\end{aligned}\tag{3.34}$$

The reference angular velocity defined above includes both the desired angular velocity and a position error term. Therefore, it becomes a proportional feedback term to stabilize the quadrotor attitude when the error of the angular velocity is small. Since the matrix  $Z$  is an unitary matrix,  $\dot{\boldsymbol{\omega}}_r$  can be simplified as,

$$\dot{\boldsymbol{\omega}}_r = \dot{\boldsymbol{\omega}}_d + \Lambda(\boldsymbol{\omega}_d - \boldsymbol{\omega}) - Z^{-1}\dot{Z}Z^{-1}\Lambda(\boldsymbol{\eta}_d - \boldsymbol{\eta})\tag{3.35}$$

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

According to the nonlinear attitude controller developed by Bandyopadhyay and Chung, the control law is given by the below equation [13].

$$\mathbf{u} = J\dot{\boldsymbol{\omega}}_r - \text{Skew}(J\boldsymbol{\omega})\boldsymbol{\omega}_r + K_\omega(\boldsymbol{\omega}_r - \boldsymbol{\omega})\tag{3.36}$$

where is  $K_\omega$  a positive gain matrix,

$$K_\omega = \begin{bmatrix} k_{\omega_p} & 0 & 0 \\ 0 & k_{\omega_q} & 0 \\ 0 & 0 & k_{\omega_r} \end{bmatrix} > 0\tag{3.37}$$

In order to prove stability of the nonlinear attitude controller, the Lyapunov method is used. From Equations (3.20) and (3.36), the below equations are obtained.

$$\boldsymbol{\omega}_r - \boldsymbol{\omega} = \boldsymbol{\omega}_d - \boldsymbol{\omega} + Z^{-1}\Lambda(\boldsymbol{\eta}_d - \boldsymbol{\eta})\tag{3.38}$$

$$(\dot{\boldsymbol{\eta}}_d - \dot{\boldsymbol{\eta}}) - Z(\boldsymbol{\omega}_r - \boldsymbol{\omega}) + \Lambda(\boldsymbol{\eta}_d - \boldsymbol{\eta}) = 0\tag{3.39}$$

From equations (3.15) and (3.36), the below equations can be obtained.

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

$$J(\dot{\boldsymbol{\omega}}_r - \dot{\boldsymbol{\omega}}) + (K_\omega - \text{Skew}(J\boldsymbol{\omega}))(\boldsymbol{\omega}_r - \boldsymbol{\omega}) = 0 \quad (3.41)$$

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,

$$\begin{bmatrix} J & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{s}}_1 \\ \dot{\boldsymbol{s}}_2 \end{bmatrix} + \begin{bmatrix} -\text{Skew}(J\boldsymbol{\omega}) + K_\omega & 0 \\ -Z & \Lambda \end{bmatrix} \begin{bmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (3.42)$$

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

$$V(\boldsymbol{s}_1, \boldsymbol{s}_2) = \begin{bmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \end{bmatrix}^T \begin{bmatrix} J & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \end{bmatrix} \quad (3.43)$$

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

$$V(\boldsymbol{s}_1, \boldsymbol{s}_2) = \boldsymbol{s}_1^T J \boldsymbol{s}_1 + \boldsymbol{s}_2^T I \boldsymbol{s}_2 > 0 \quad (3.44)$$

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

$$\begin{aligned} \dot{V}(\boldsymbol{s}_1, \boldsymbol{s}_2) &= \begin{bmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \end{bmatrix}^T \begin{bmatrix} J & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{s}}_1 \\ \dot{\boldsymbol{s}}_2 \end{bmatrix} + 2 \begin{bmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \end{bmatrix}^T \begin{bmatrix} J & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{s}}_1 \\ \dot{\boldsymbol{s}}_2 \end{bmatrix} \\ &= 2 \begin{bmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \end{bmatrix}^T \begin{bmatrix} J & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{s}}_1 \\ \dot{\boldsymbol{s}}_2 \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \end{bmatrix}^T \begin{bmatrix} 2\text{Skew}(J\boldsymbol{\omega}) - 2K_\omega & 0 \\ 2Z & -2\Lambda \end{bmatrix} \begin{bmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \end{bmatrix}^T \begin{bmatrix} -2K_\omega & 0 \\ 2Z & -2\Lambda \end{bmatrix} \begin{bmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \end{bmatrix} \\ &= -2\boldsymbol{s}'_1^T K_\omega \boldsymbol{s}'_1 - 2\boldsymbol{s}'_2^T \Lambda \boldsymbol{s}'_2 \\ &< 0 \end{aligned} \quad (3.45)$$

where  $\mathbf{s}'_1 = \mathbf{U}\mathbf{s}_1$ ,  $\mathbf{s}'_2 = \mathbf{U}\mathbf{s}_2$ , and  $\mathbf{U}$  is the unitary matrix such that

$$\mathbf{U}^T \begin{bmatrix} -2K_\omega & 0 \\ 2Z & -2\Lambda \end{bmatrix} \mathbf{U} = \begin{bmatrix} -2K_\omega & 0 \\ 0 & -2\Lambda \end{bmatrix} \quad (3.46)$$

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. Here,  $V(\mathbf{s}_1, \mathbf{s}_2)$  is radially unbounded.

$$\forall \mathbf{s}_1, \mathbf{s}_2 \neq \mathbf{0}, \quad c_2(\|\mathbf{s}_1\|^2 + \|\mathbf{s}_2\|^2) > V(\mathbf{s}_1, \mathbf{s}_2) \approx \mathbf{s}_1^T \mathbf{J} \mathbf{s}_1 + \mathbf{s}_2^T \mathbf{I} \mathbf{s}_2 > c_1(\|\mathbf{s}_1\|^2 + \|\mathbf{s}_2\|^2) > 0 \quad (3.47)$$

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.

$$\forall \mathbf{s}_1, \mathbf{s}_2 \neq \mathbf{0}, \quad \dot{V}(\mathbf{s}_1, \mathbf{s}_2) \leq -c_3(\|\mathbf{s}_1\|^2 + \|\mathbf{s}_2\|^2) < 0 \quad (3.48)$$

From the Lyapunov 1st method, the controller is globally exponentially stable as well without perturbation [31]. Hence it gives bounded errors that can be estimated with respect to bounded disturbance or modeling errors [13].

## 3.5 Motor Control

As described in Section 3.4, the desired torque of the quadrotor is computed by the inner-loop controller. The next step is to compute 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 are stated.

### 3.5.1 Mapping Attitude and Thrust Control into Motor Speeds

In the mixer, the magnitude of the desired thrust  $F_d$  from the outer-loop and the desired attitude control  $\boldsymbol{\tau} = (\tau_\phi, \tau_\theta, \tau_\psi)$  from the inner-loop are received. In order to map the attitude control and the desired thrust

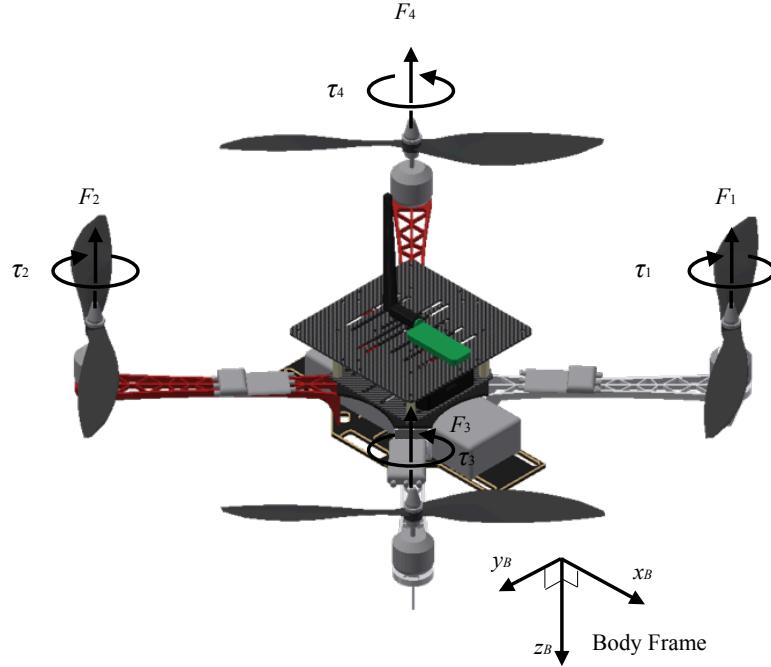


Figure 3.3: Thrust and Torque Generated by Each Rotor

into motor speed  $\mathbf{n} = (n_1, n_2, n_3, n_4)$ , an aerodynamic 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,

$$C_T(n_i) = \frac{T}{\rho n_i^2 D^4} \quad (3.49)$$

$$C_P(n_i) = \frac{P}{\rho n_i^3 D^5} \quad (3.50)$$

where  $\rho$  is the density of air,  $D$  is the propeller's diameter, and  $T, P$  are thrust and torque generated by a propeller, respectively [32]. 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 its motor speed  $n_i$ . With the experimental data of  $C_T$  and  $C_P$ , thrust  $F_i$  and torque  $\tau_i$  generated by a motor according to its motor speed  $n_i$  are given as below,

$$F_i = C_T(n_i) \rho n_i^2 D^4 \quad (3.51)$$

$$\begin{aligned}\tau_i &= \frac{P}{2\pi n_i} \\ &= \frac{C_P(n_i)\rho n_i^2 D^5}{2\pi}\end{aligned}\tag{3.52}$$

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

$$\begin{aligned}F &= F_1 + F_2 + F_3 + F_4 \\ \tau_\phi &= \frac{l}{\sqrt{2}}(-F_1 + F_2 + F_3 - F_4) \\ \tau_\theta &= \frac{l}{\sqrt{2}}(F_1 - F_2 + F_3 - F_4) \\ \tau_\psi &= \tau_1 + \tau_2 - \tau_3 - \tau_4\end{aligned}\tag{3.53}$$

where  $l$  to the length of the frame [28]. We define an aerodynamic coefficient matrix  $B(\mathbf{n})$  as,

$$B(\mathbf{n}) = \begin{bmatrix} C_T(n_1) & C_T(n_2) & C_T(n_3) & C_T(n_4) \\ -\frac{l}{\sqrt{2}}C_T(n_1) & \frac{l}{\sqrt{2}}C_T(n_2) & \frac{l}{\sqrt{2}}C_T(n_3) & -\frac{l}{\sqrt{2}}C_T(n_4) \\ \frac{l}{\sqrt{2}}C_T(n_1) & -\frac{l}{\sqrt{2}}C_T(n_2) & \frac{l}{\sqrt{2}}C_T(n_3) & -\frac{l}{\sqrt{2}}C_T(n_4) \\ \frac{D}{2\pi}C_P(n_1) & \frac{D}{2\pi}C_P(n_2) & -\frac{D}{2\pi}C_P(n_3) & -\frac{D}{2\pi}C_P(n_4) \end{bmatrix}\tag{3.54}$$

Then, the above equation can be written as,

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

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

$$\begin{bmatrix} n_1^2 \\ n_2^2 \\ n_3^2 \\ n_4^2 \end{bmatrix} = \frac{1}{\rho D^4} B^{-1}(\mathbf{n}) \begin{bmatrix} F_d \\ u_p \\ u_q \\ u_r \end{bmatrix}\tag{3.56}$$

### 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 (3.56). To solve the problem, the linear approximation in terms of  $\frac{1}{n_i^2}$  as,

$$\begin{aligned} C_T(n_i) &= C_{T0} + C_{T1} \frac{1}{n_i^2} + O\left(\frac{1}{n_i^4}\right) \\ &\approx C_{T0} + C_{T1} \frac{1}{n_i^2} \end{aligned} \quad (3.57)$$

$$\begin{aligned} C_P(n_i) &= C_{P0} + C_{P1} \frac{1}{n_i^2} + O\left(\frac{1}{n_i^4}\right) \\ &\approx C_{P0} + C_{P1} \frac{1}{n_i^2} \end{aligned} \quad (3.58)$$

can be applied. Then, Equation (3.55) can be written as,

$$\begin{bmatrix} F_d \\ u_p \\ u_q \\ u_r \end{bmatrix} \approx \rho D^4 \left( B_0 \begin{bmatrix} n_1^2 \\ n_2^2 \\ n_3^2 \\ n_4^2 \end{bmatrix} + B_1 \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \right) \quad (3.59)$$

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

$$B_0 = \begin{bmatrix} C_{T0} & C_{T0} & C_{T0} & C_{T0} \\ -\frac{l}{\sqrt{2}}C_{T0} & \frac{l}{\sqrt{2}}C_{T0} & \frac{l}{\sqrt{2}}C_{T0} & -\frac{l}{\sqrt{2}}C_{T0} \\ \frac{l}{\sqrt{2}}C_{T0} & -\frac{l}{\sqrt{2}}C_{T0} & \frac{l}{\sqrt{2}}C_{T0} & -\frac{l}{\sqrt{2}}C_{T0} \\ \frac{D}{2\pi}C_{P0} & \frac{D}{2\pi}C_{P0} & -\frac{D}{2\pi}C_{P0} & -\frac{D}{2\pi}C_{P0} \end{bmatrix} \quad (3.60)$$

$$B_1 = \begin{bmatrix} C_{T1} & C_{T1} & C_{T1} & C_{T1} \\ -\frac{l}{\sqrt{2}}C_{T1} & \frac{l}{\sqrt{2}}C_{T1} & \frac{l}{\sqrt{2}}C_{T1} & -\frac{l}{\sqrt{2}}C_{T1} \\ \frac{l}{\sqrt{2}}C_{T1} & -\frac{l}{\sqrt{2}}C_{T1} & \frac{l}{\sqrt{2}}C_{T1} & -\frac{l}{\sqrt{2}}C_{T1} \\ \frac{D}{2\pi}C_{P1} & \frac{D}{2\pi}C_{P1} & -\frac{D}{2\pi}C_{P1} & -\frac{D}{2\pi}C_{P1} \end{bmatrix} \quad (3.61)$$

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

$$\begin{bmatrix} n_1^2 \\ n_2^2 \\ n_3^2 \\ n_4^2 \end{bmatrix} \approx B_0^{-1} \left( \frac{1}{\rho D^4} \begin{bmatrix} F_d \\ u_p \\ u_q \\ u_r \end{bmatrix} - B_1 \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \right) \quad (3.62)$$

To compute the model of thrust coefficient (3.57) and power coefficient (3.58), the experimental data of the propeller APC  $10 \times 4.7$  in the UIUC Propeller Database Vol.1 are used, and the method of least mean square error (LMSE) linear regression is applied in terms of  $\frac{1}{n_i^2}$  [16]. The approximated models of  $C_T(n_i)$  and  $C_P(n_i)$  are given as Equations (3.63) and (3.64), and the experimental data and the approximated models are compared in the Figures 3.4, 3.5.

$$C_T(n_i) = 0.1627 - 73047 \times \frac{1}{n_i^2} \quad (3.63)$$

$$C_P(n_i) = 0.0791 - 49112 \times \frac{1}{n_i^2} \quad (3.64)$$

### 3.5.3 Motor Control

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, an average of constant-voltage pulsing signals actuates a motor [33]. Therefore, motor voltage is controllable by changing a pulse signal's width. Let  $d_{pwm}$  be duty cycle as,

$$d_{pwm} = H f_{pwm} : \quad 0 \leq d_{pwm} \leq 1 \quad (3.65)$$

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,

$$V_{mot} = d_{pwm} V_0 \quad (3.66)$$

where  $V_0$  is the constant voltage of signals.

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

$$V_{mot} = 2\pi k_b n + R i + L \frac{di}{dt} \quad (3.67)$$

$$k_m i = 2\pi k_f n + 2\pi I_{mot} \frac{dn}{dt} + M_R \text{sgn}(n) \quad (3.68)$$

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,  $R$  is the motor's internal resistor, and  $L$  is the motor's internal inductance [34].  $I_{mot}$  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  $n$ .

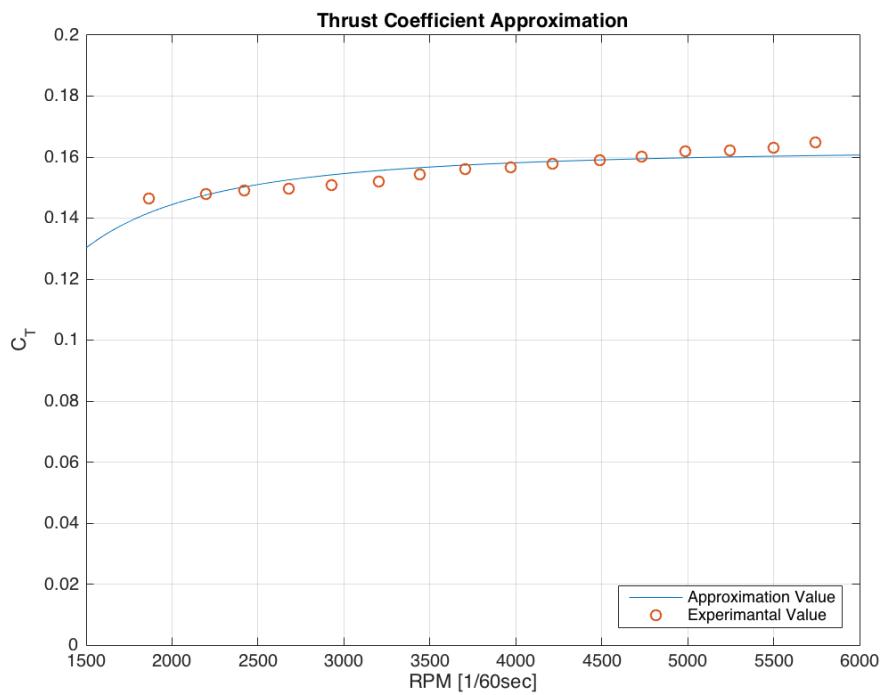


Figure 3.4: Comparison of experimental and approximated thrust coefficient

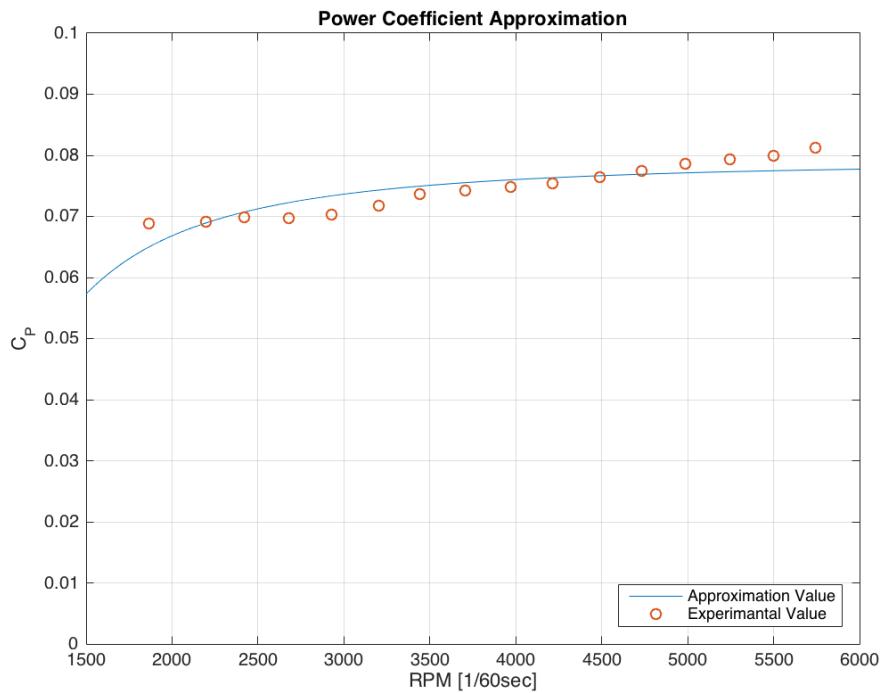


Figure 3.5: Comparison of experimental and approximated power coefficient

$$V_{mot} = 2\pi \left( (k_b + \frac{Rk_f}{k_m})n + \frac{1}{k_m}(RI_{mot} + Lk_f)\frac{dn}{dt} + \frac{LI_{mot}}{k_m}\frac{d^2n}{dt^2} \right) + RM_R \text{sgn}(n) : n \neq 0 \quad (3.69)$$

Also, from Equation (3.68), 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 (3.69) can be simplified as,

$$V_{mot} \approx 2\pi(k_b + \frac{Rk_f}{k_m})n + RM_R \text{sgn}(n) \quad (3.70)$$

Define composite coefficients of the tangent  $\alpha$ , and the intercept  $\beta$  as,

$$\alpha = \frac{k_m V_0}{2\pi(k_b k_m + Rk_f)} \quad (3.71)$$

$$\beta = \frac{k_m R M_R}{2\pi(k_b k_m + Rk_f)} \quad (3.72)$$

Then, from Equations (3.66) and (3.70), the relation between the motor speed  $n$  and the duty cycle  $d_{pwm}$  is written as the below equation.

$$n \approx \begin{cases} \alpha d_{pwm} - \beta & d_{threshold} \leq d_{pwm} \leq 1 \\ 0 & d_{pwm} \leq d_{threshold} \end{cases} \quad (3.73)$$

If the characteristic information of the motors is available, the parameters  $\alpha$ ,  $\beta$  of the model of Equation (3.73) 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  $\alpha$  and the intercept  $\beta$  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 set to be between 1000  $\mu$ sec and 2000  $\mu$ sec. The result of the experiment and the approximated model by the LMSE linear regression method are shown in Figure 3.6 and Equation (3.74).

$$n = \begin{cases} 0.1716 \times d_{pwm} - 803.4 & 1100 \leq d_{pwm} \leq 2000 \\ 0 & \text{otherwise} \end{cases} \quad (3.74)$$

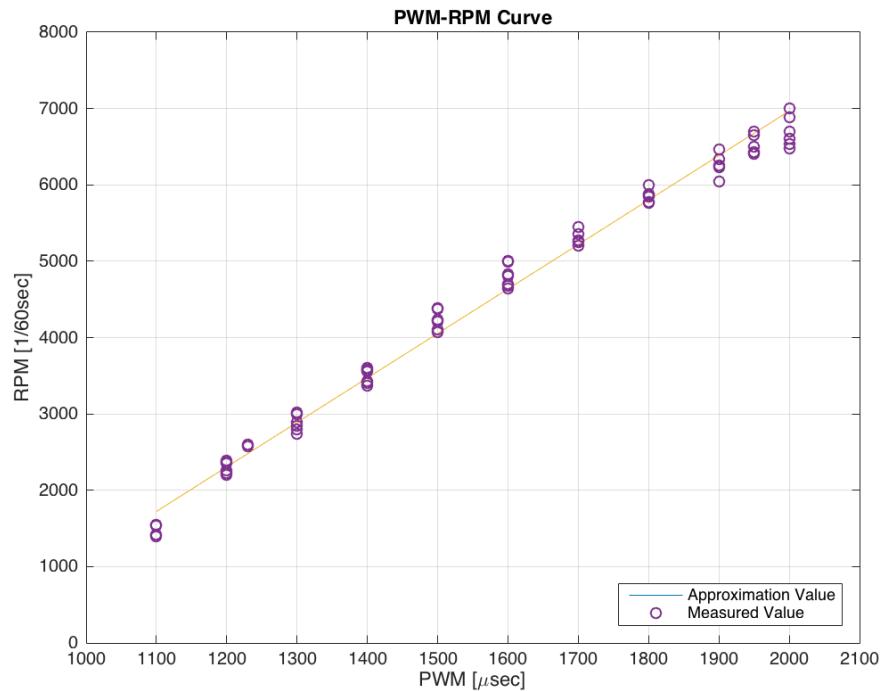


Figure 3.6: Change of RPM output corresponding to PWM command

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 (3.74) is applied to control motor speed. Also, PWM command is limited to be between  $1230 \mu\text{sec}$  and  $1950 \mu\text{sec}$  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 conventional PID attitude control are compared with each other, by simulation and flight experiments. First, the PID attitude control 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 presented. 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 nonlinear control system of Chapter 3, a conventional inner-loop controller is also tested for comparison. In the linear controller, derivative of desired attitude  $\dot{\boldsymbol{\eta}}_d$  is also computed as described in Subsection 3.4.2. 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 (3.36) of the nonlinear 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 linear model as below, instead of Equation (3.56). The matrix for

mapping the desired torques into desired motor speed is 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, such as motor dynamics, are ignored as tuning the gains of the attitude PID control.

## 4.2 Simulation Study for Inner-loop Controller

It is difficult to evaluate 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.

The PID inner-loop controller and a linear mapping model from torque into motor speeds described in Section 4.1 are often used for attitude control of a quadrotor. Let it be called "linear attitude controller". In this section, simulations of the linear and nonlinear attitude control systems are stated, and comparison of the results is discussed to validate the advantage of the control system stated in Chapter 3. First, setup for simulation is explained., and then, simulation results of the two control systems are discussed.

### 4.2.1 Simulation Setup

In simulation, the dynamic models of Equations (3.7) and (3.15) are applied. The physical quantities of the quadrotor are the same as stated in Chapter 2 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 in Section 4.1.

### 4.2.3 Discussion

In most cases, the attitude of the quadrotor converges to the desired attitude faster with the nonlinear control system than it does with the PID linear controller. With roll and pitch changes, the attitude becomes close enough to desired attitude within 0.3 sec at the nonlinear attitude control, and 0.8 sec at the linear attitude control. In the cases of yaw, the rate of convergence is approximately the same for both control systems; the attitude converges to the desired attitude within 1.5 sec. From these results, it can be concluded that the nonlinear attitude controller has the advantage of fast conversion with a guaranteed property robustness [6][35]. The improvement of the attitude control is expected from global exponential stability of the nonlinear attitude controller. In Figure 4.7, the response time of the nonlinear attitude controller is fast enough so that the attitude of the quadrotor stabilize quickly, even with rapid change of desired attitude. Therefore, the nonlinear attitude controller is more suitable for high-agility performance. The deference of response time of the nonlinear attitude control with respect to roll, pitch, and yaw may be caused by the difference of the inertia values along axises,  $I_{xx}$ ,  $I_{yy}$  and  $I_{zz}$ .

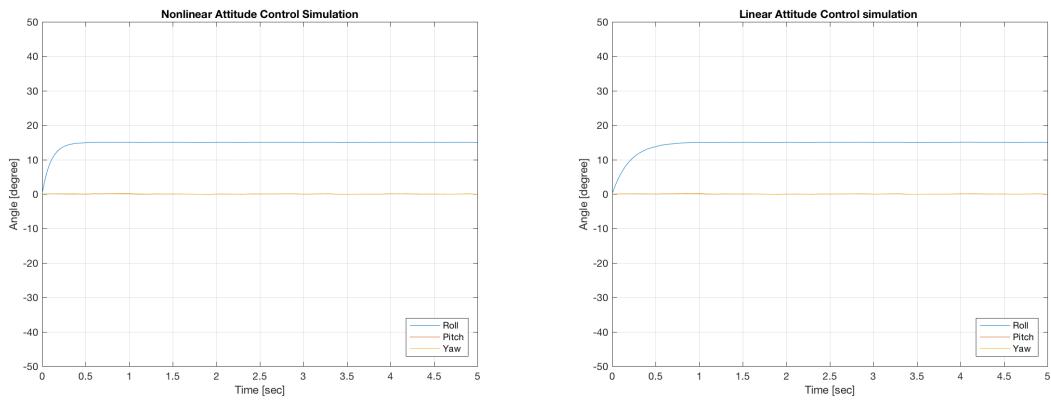


Figure 4.1: Simulation Result (Case 1)

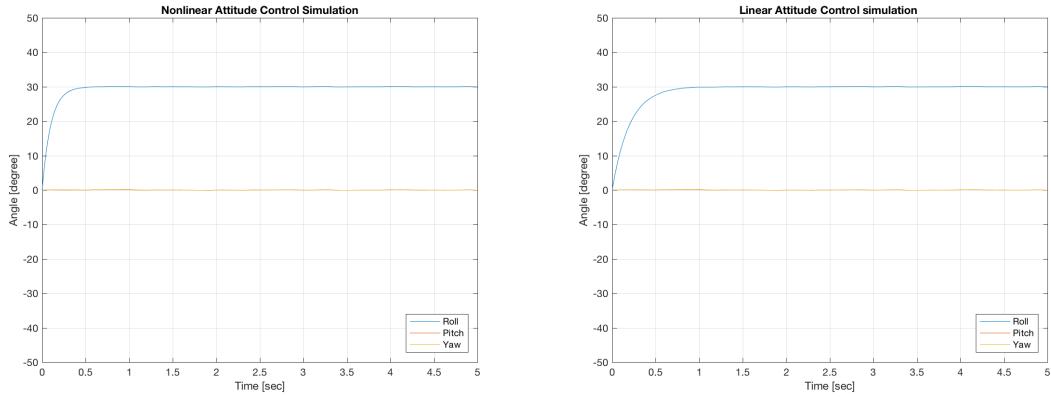


Figure 4.2: Simulation Result (Case 2)

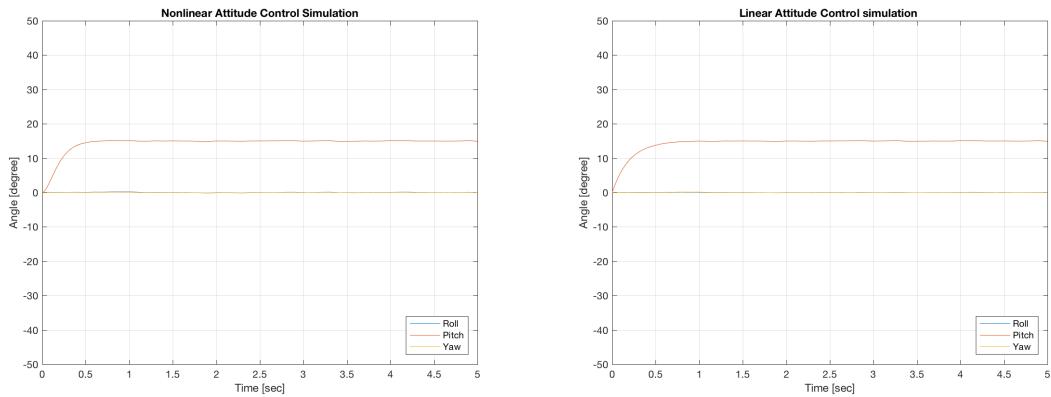


Figure 4.3: Simulation Result (Case 3)

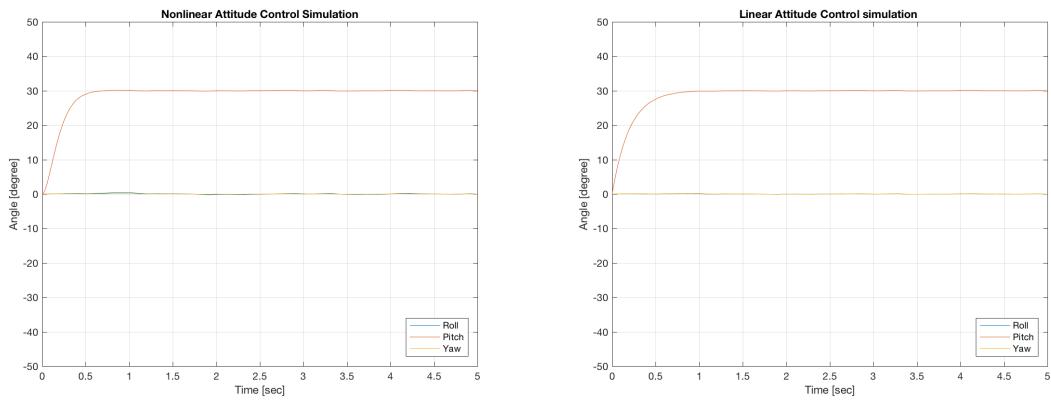


Figure 4.4: Simulation Result (Case 4)

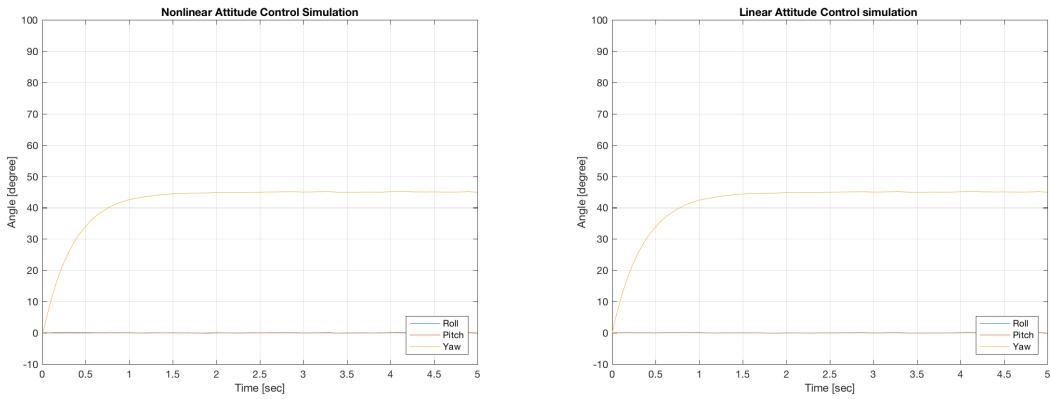


Figure 4.5: Simulation Result (Case 5)

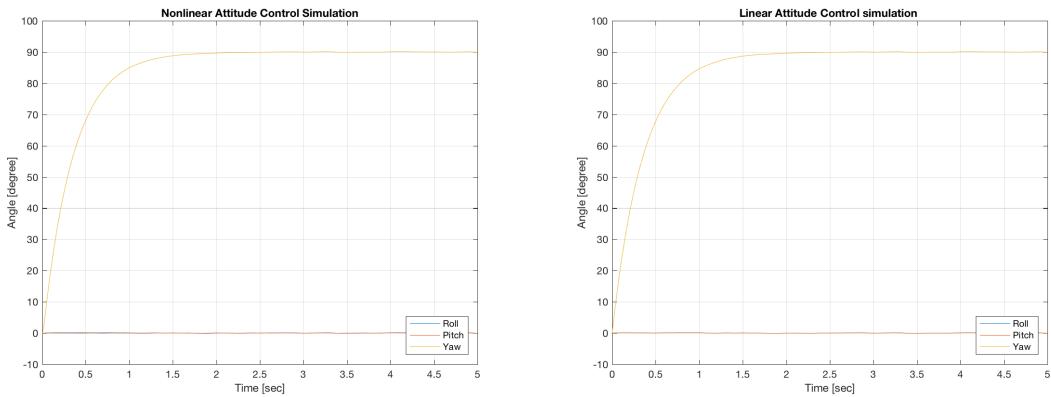


Figure 4.6: Simulation Result (Case 6)

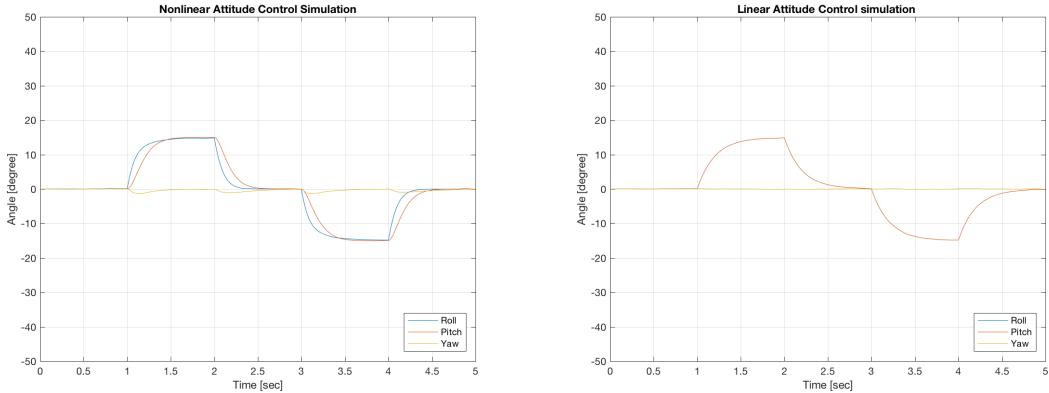


Figure 4.7: Simulation Result (Case 7)

However, the yaw error of the nonlinear attitude control is greater when desired attitude changes in Case 5, than one of the linear attitude control. This is potentially caused by dependency among each angular velocity parameter  $p$ ,  $q$ , and  $r$  at the nonlinear attitude controller. As shown at Equation (3.36), each variable cannot be decoupled, while the PID control deals each angular velocity independently. Also, the roll and pitch of the nonlinear attitude controller show different performance, while the the ones of the linear attitude controller are similar to each other. The difference of the inertia among each axis of the body frame possibly causes the asymmetry of the performance.

### 4.3 Flight Experiment

In addition to the simulation study in Section 4.2, the quadrotor system was tested by experiments to evaluate the nonlinear 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 [36]. The quadrotor was controlled with the outer-loop and inner-loop control stated in Chapter 3. The linear attitude control stated in Section 4.1 was also tested with the outer-loop control of Section 3.3 to compare with the performance of the nonlinear 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



Figure 4.8: Motion Capture Arena of the Intelligent Robotics Laboratory

not damage anything out of the arena. In order to limit the range of quadrotors' maneuver and protect them from potential damage, the arena is 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 measures motion of the object that has reflective motion capture markers on it.

### 4.3.2 Motion Capture System

IRL's motion capture system consists of 8 Vicon T-series T40S 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 [37].

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



Figure 4.9: Vicon Motion Capture Cameras

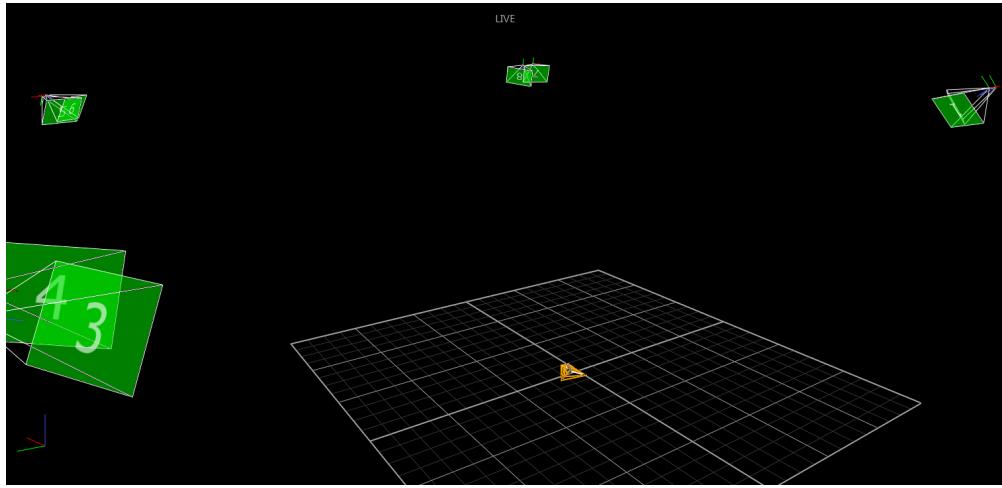


Figure 4.10: Vicon Motion Capture Interface

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 position and attitude.

### 4.3.3 Experiment Setup

Prior to experiment, we tuned the control gains of the nonlinear and linear inner-loop controllers, based on the simulated control gains in Section 4.2 and adjusted the gains by multiple flight tests.

In the experiments, the quadrotor was controlled by the PID outer-loop controller described in Section 3.3 and either the nonlinear attitude control of Chapter 3 or the linear attitude control of Section 4.1. Since it is difficult to set the same desired state of the quadrotor for its position, the position and attitude trajectories for the quadrotor to reach the desired position are used to evaluate performance of each inner-loop controller. 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 outer-loop control of the x, y-axis directions is expected to drift since the quadrotor does not include internal position measurement for x, y-axis directions.

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

the quadrotor. The trajectories and attitude changes over time are plotted in Figures 4.11 - 4.17. Full lines represent the quadrotor's actual trajectories and dashed lines represent 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, and therefore, there are drifts along the x,y-axis directions in every case. The position estimator uses integration of values from the internal accelerometer. Errors are accumulated by iteration, and therefore, position drifts are not avoidable. The errors caused by drifts are usually less severe with the nonlinear control system introduced in Chapter 3. The drift occurs especially when the quadrotor takes off and lands. Dynamic factors, such as corrosion between the quadrotor's leg and the ground, are potential reasons of drifts at taking off and landing. Also, the aerodynamic model may be different near the ground and, therefore, may cause drifts as well.

In most cases, x, y-axis position errors of the linear controller are greater than the errors of the nonlinear controller. In Case 1, the y-axis drift is less than 1 m with the nonlinear control system while the error is about up to 3 m with the linear control system, and the x-axis drifts of both controllers are similar. In Cases 4 and 5, the x, y-axis position errors with the nonlinear attitude control are bounded up to 1 m, but the errors of the linear attitude control grow up to 3 m. The x, y-axis errors show similar tendencies in Cases 2 and 3. However, compared with Cases 4 and 5, the errors with the linear control system do not exceed 2 m. In Cases 6 and 7, the x, y-axis error rates of the linear control system are twice greater than the error rates of the nonlinear control system. We were not able to further reduce the position errors of the PID attitude control by tuning the gains. From these results, we can conclude that the nonlinear control system has an advantage of superior performance, compared with the linear control system.

However, the nonlinear control system does not seem to have an advantage with respect to z-axis position over an conventional PID control system. There are increasing and decreasing of altitude while the quadrotor changes its attitude, but z-axis position errors of both controllers are less than 0.5 m 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, time for the quadrotor to reach desired position with the nonlinear control system is approximately the same as the one with the linear attitude control since there is an upper limit of desired velocity. However, as shown in Figures 4.12 - 4.15, the quadrotor with the nonlinear control system responses fast and reaches to the maximum velocity, while it moves more smoothly with the linear control system. Also,

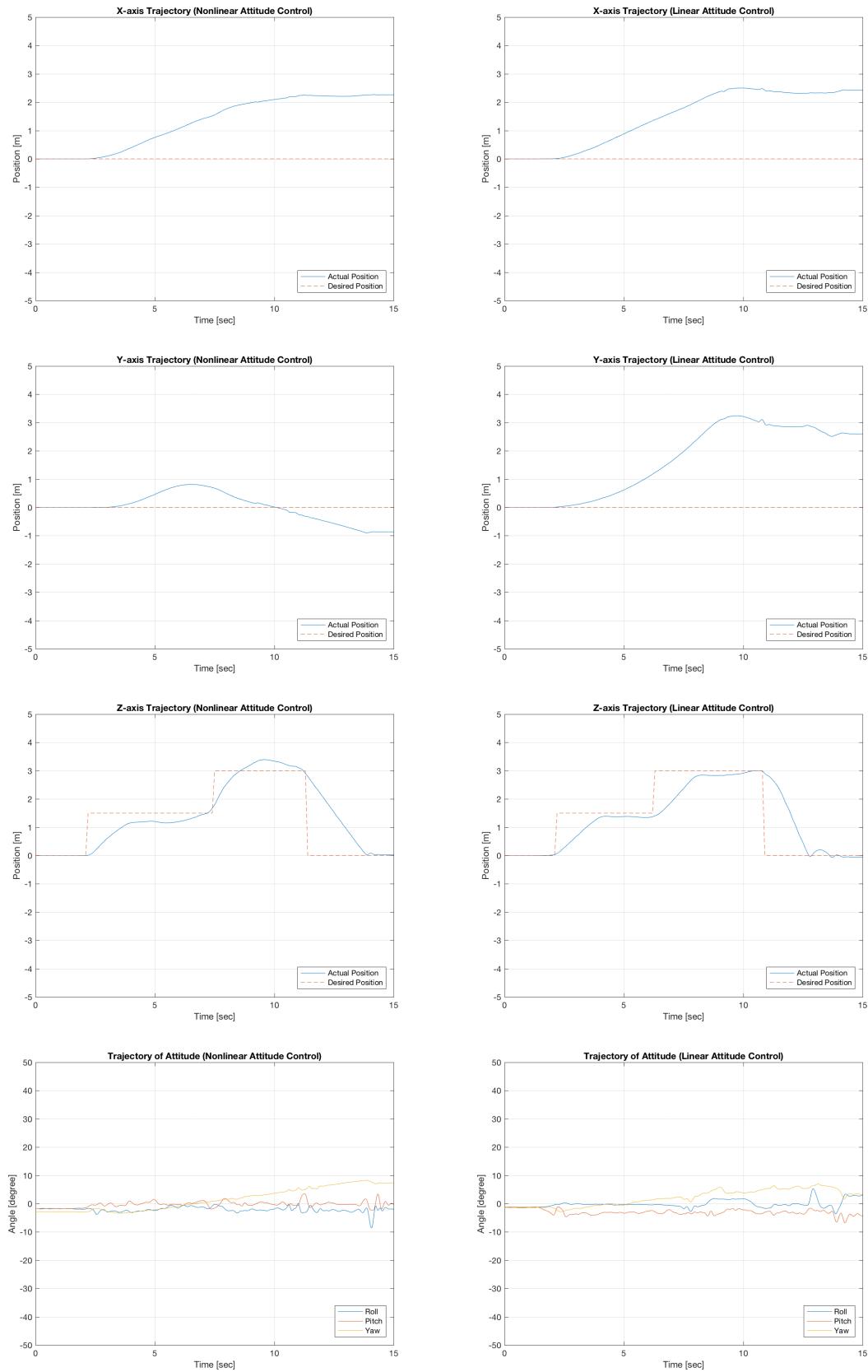


Figure 4.11: Experiment Result (Case 1)

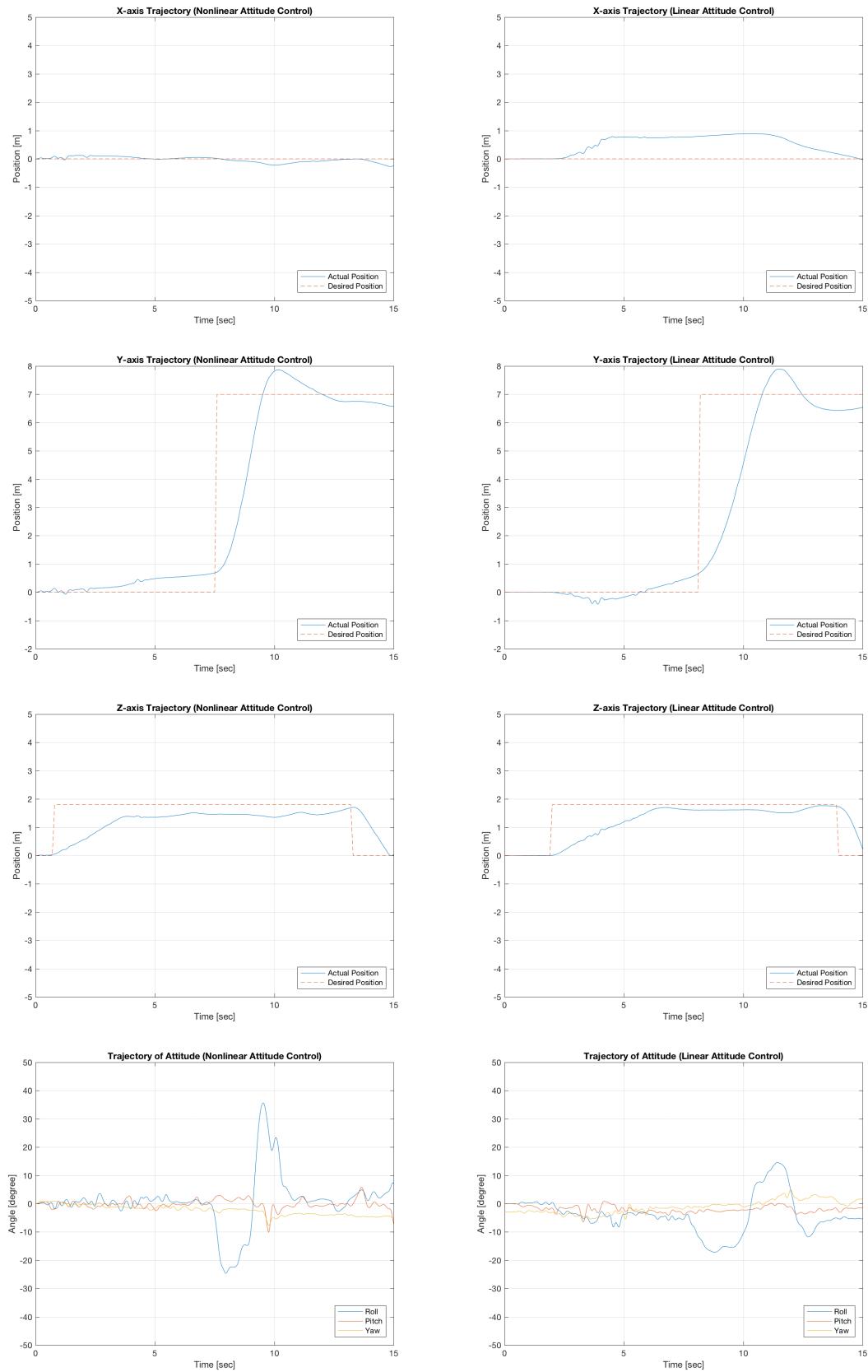


Figure 4.12: Experiment Result (Case 2)

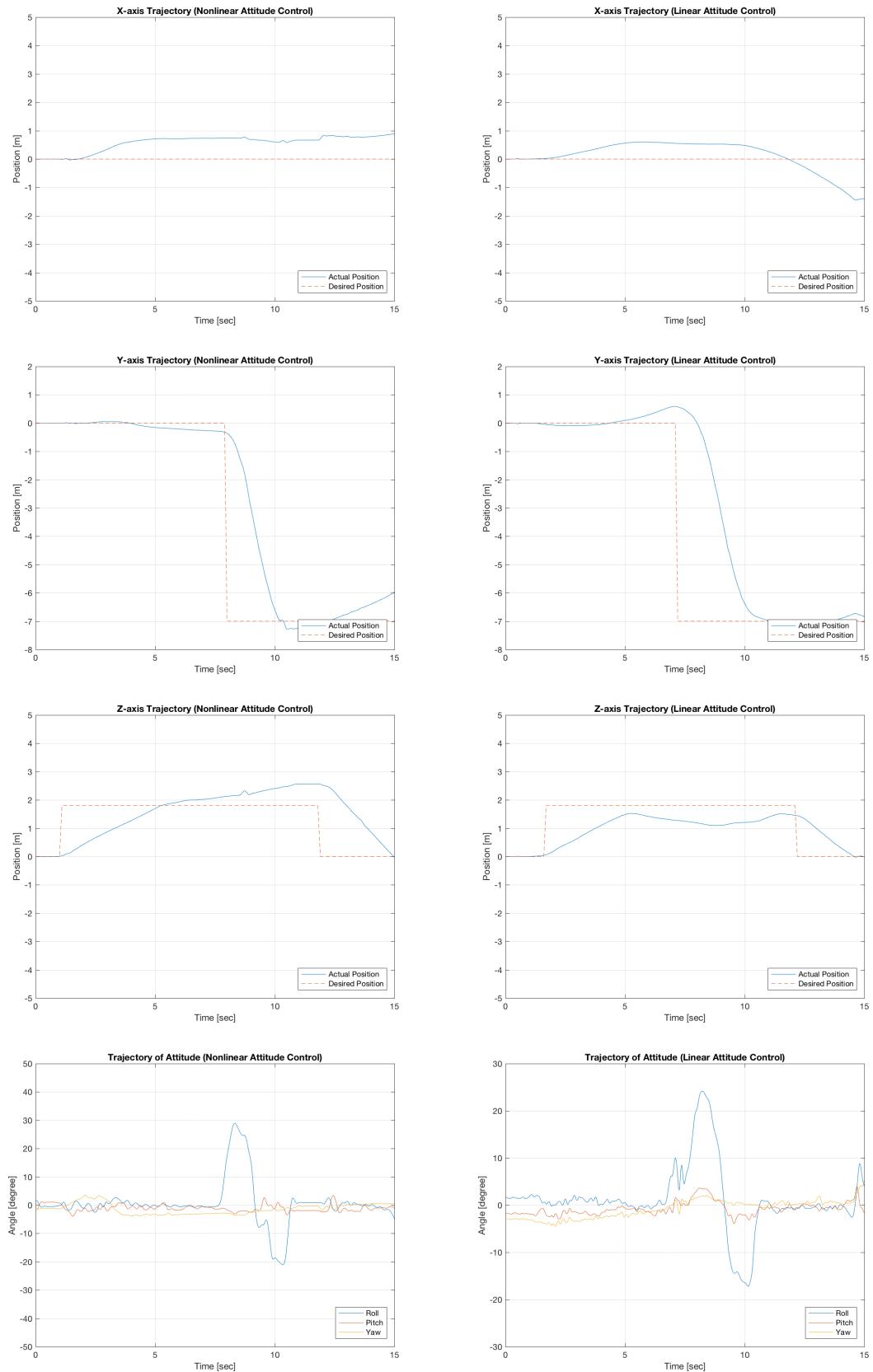


Figure 4.13: Experiment Result (Case 3)

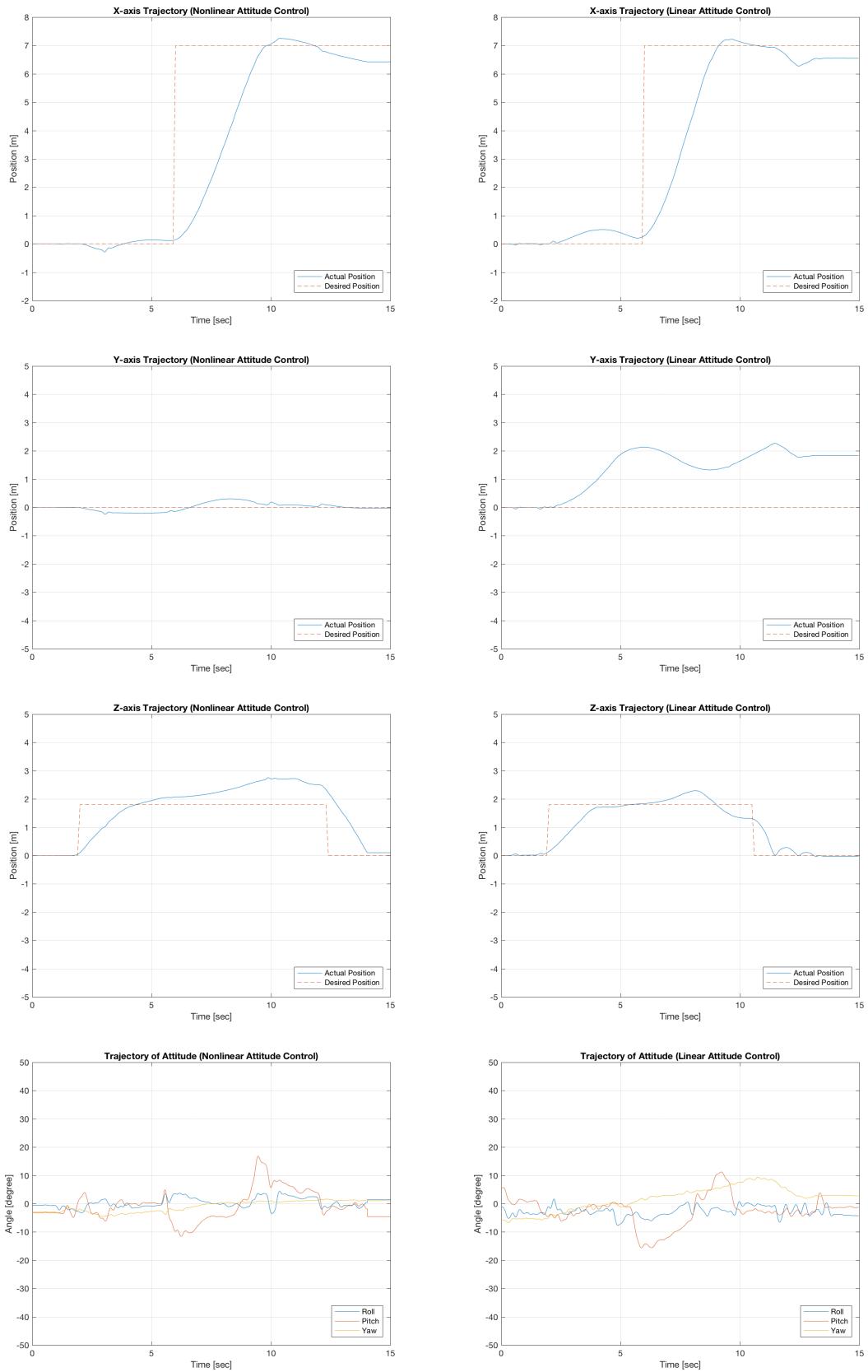


Figure 4.14: Experiment Result (Case 4)

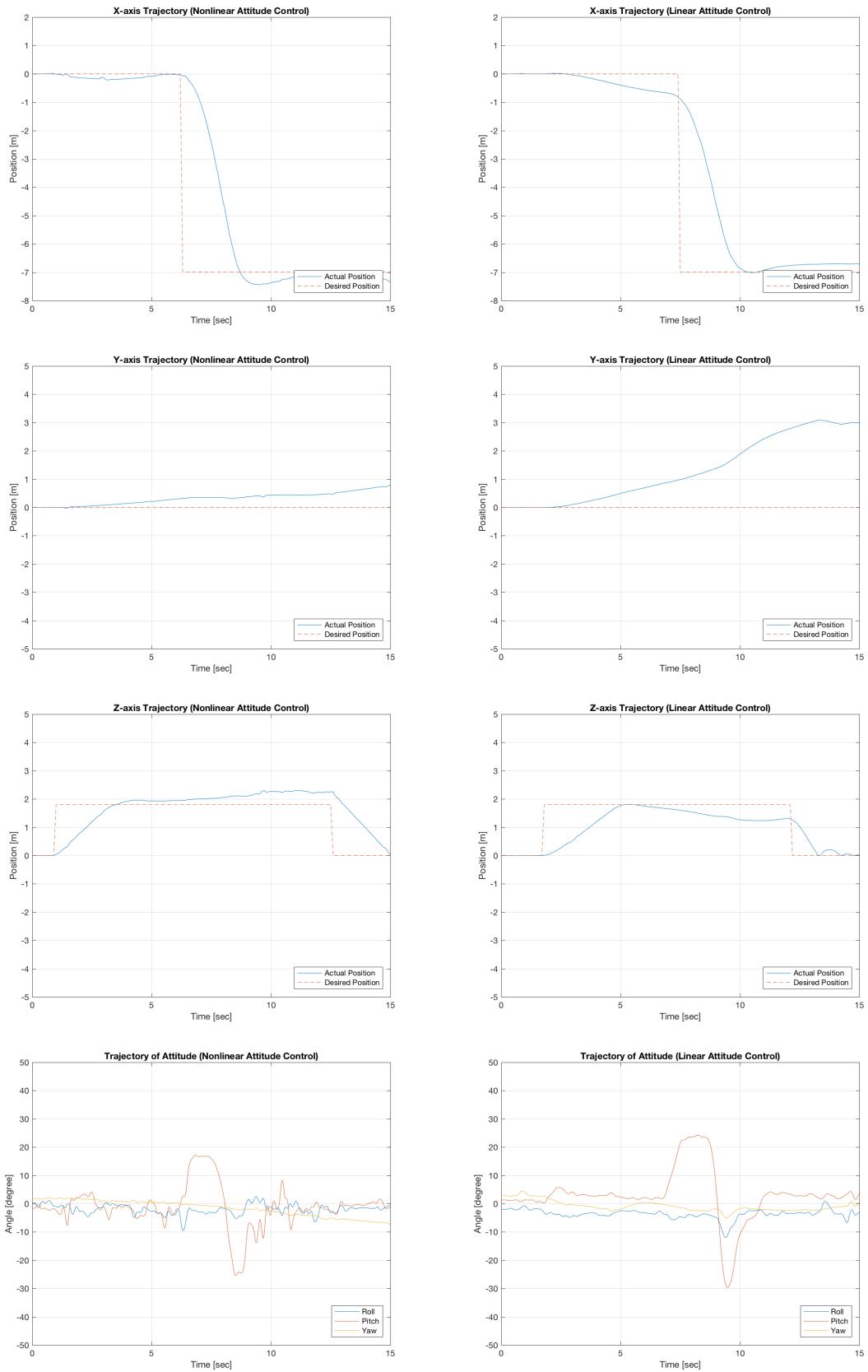


Figure 4.15: Experiment Result (Case 5)

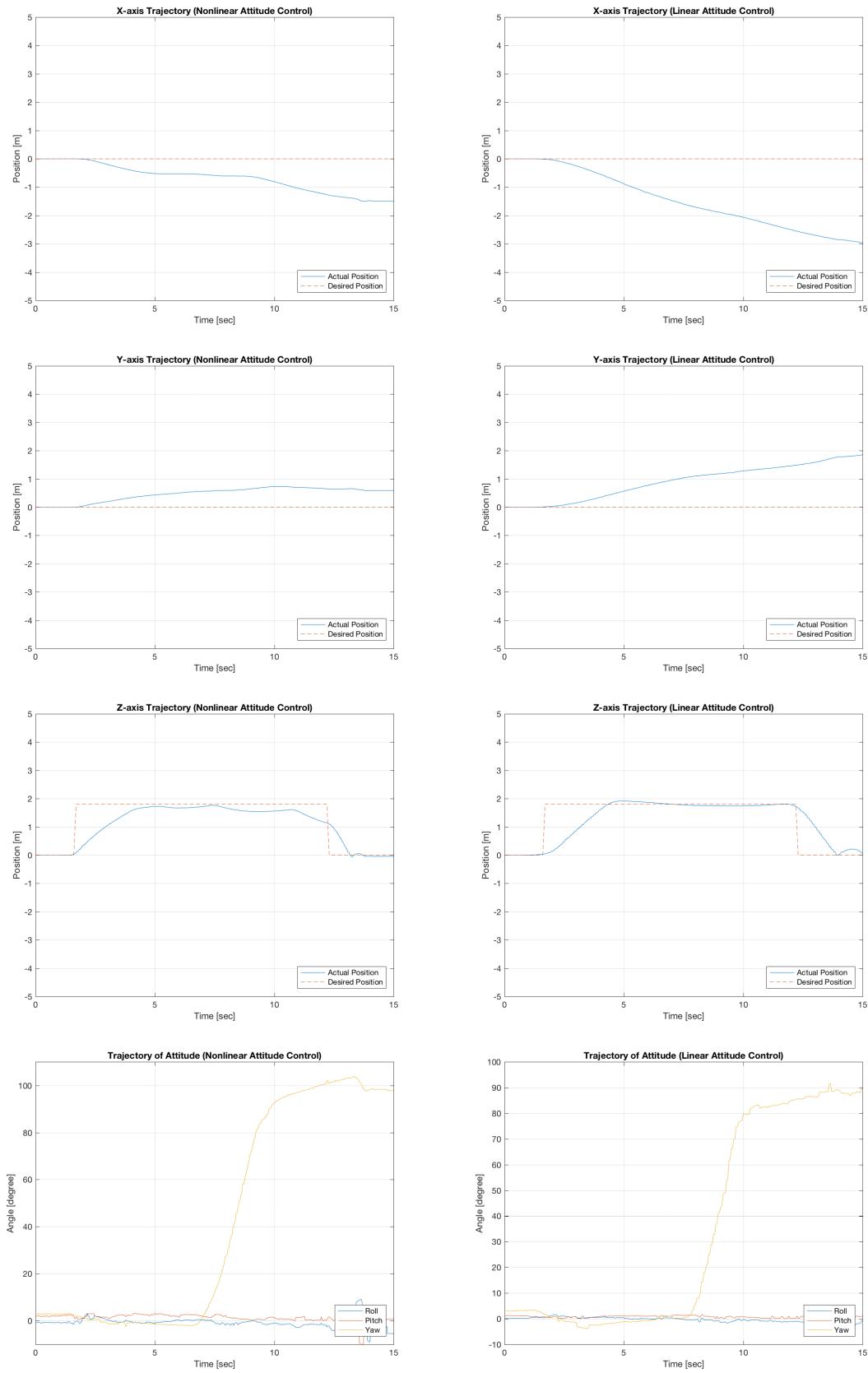


Figure 4.16: Experiment Result (Case 6)

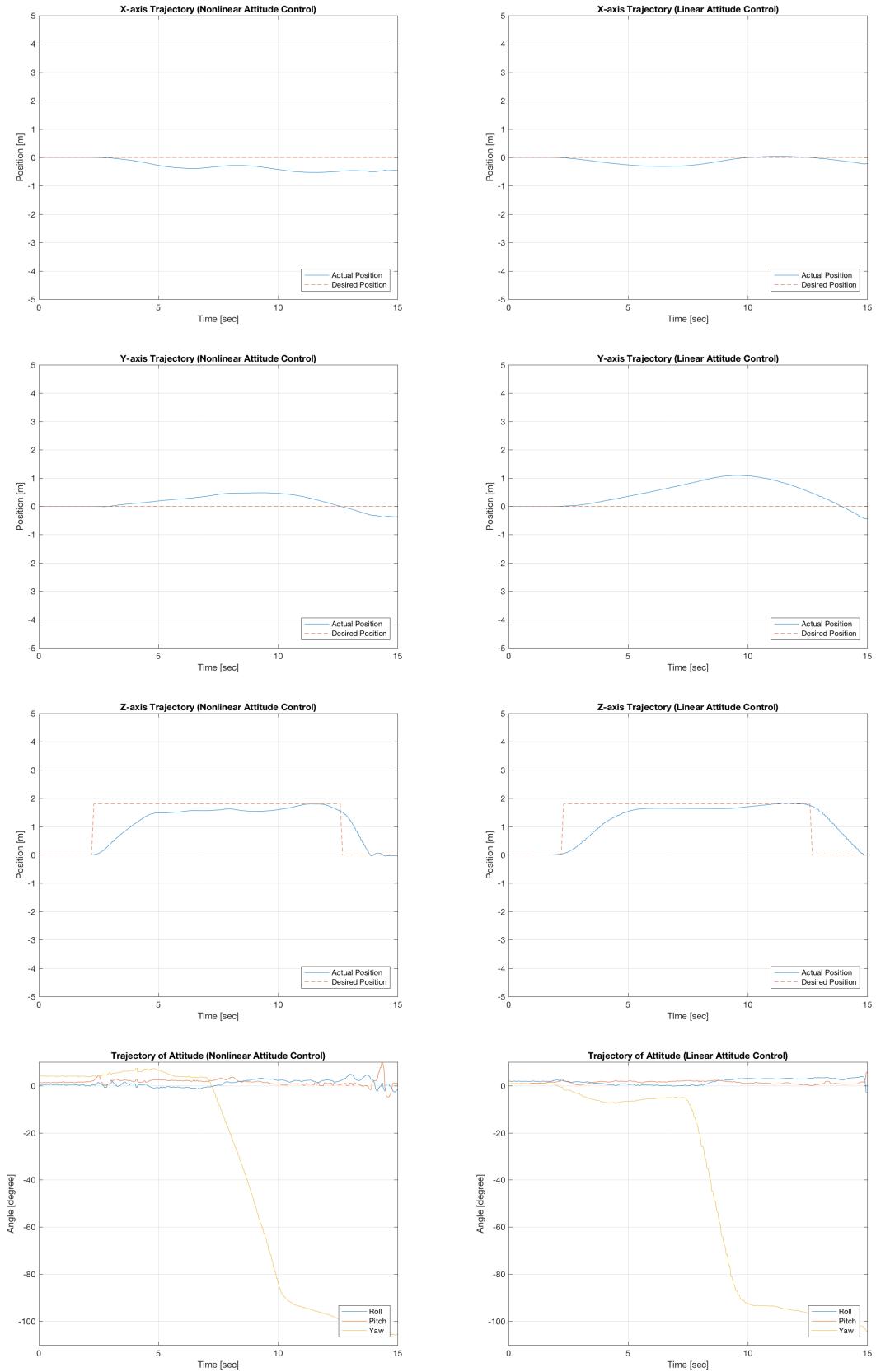


Figure 4.17: Experiment Result (Case 7)

as shown in Figures 4.16 and 4.17, yaw control is more stabilized with the nonlinear control system. These support globally exponentially stability of the nonlinear attitude control in Section 3.4 and the results of simulation in Section 4.2 that the nonlinear 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 nonlinear control system also has an advantage of its stability.

However, the  $K_\omega$  and  $\Lambda$  gains should be further tuned for the nonlinear control to draw a definitive conclusion between the nonlinear control method and the PID attitude controller method. Also, there are constant yaw drifts at both control systems in Cases 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 drifts.

# Chapter 5

## Vision-based Position Estimator

As discussed in Section 4.3, the lack 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 a video sensor, for its stable autonomous operation. A simple and light position estimator based on image processing is introduced in this chapter. We assumed the case that the quadrotor hovers above four markers located on the floor.

This chapter consists of the following contents. First, the camera distortion model used in the system is stated. Then, the image processes used 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 position is introduced.

### 5.1 Camera Calibration

An 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 video sensor and the lens. The camera used in the quadrotor system has  $120^\circ$  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 [38]. 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 [39].

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

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



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

to  $x$  and  $y$  as the below equation.

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

In Equation (5.1), 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,

$$p = K_K x_d \quad (5.3)$$

where  $K_K$  is the camera matrix, defined as,

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

$f_{c1}, f_{c2}$  are the focal distances and  $\alpha_c$  is the angle between the  $x$  and  $y$  sensor axes. Therefore, by measuring the distortion coefficients  $k_c$  and the camera matrix  $K_K$ , the undistorted coordinate  $x$  can be restored from the raw pixel coordinate  $p$ . Figure 5.1 shows the results of the marker image restored from the raw image by camera calibration.

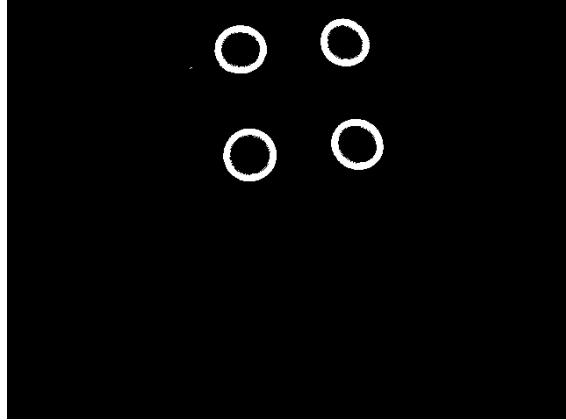


Figure 5.2: Gaussian-filtered and Thresholded Image

## 5.2 Marker Detection

Marker detection can be achieved by multiple image processes. First of all, a monochrome image is used. 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 has clean borders, the markers are detected with their contours. To be specific, the "findContours" function of OpenCV 2.4 is used for the process of finding contours [40].

From the above image processes, the markers of the image are detectable. However, due to environmental factors, such as lightness and floor state, empty coordinates possibly be detected as markers. Therefore, the following 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 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, incorrect candidates are excluded easily.

The result of each step is shown in Figure 5.2 - 5.4. The white circles in Figure 5.4 point the centers of the image markers.

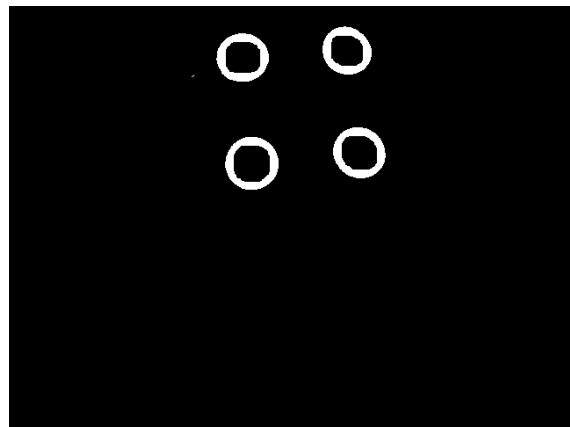


Figure 5.3: Restoration of Borders

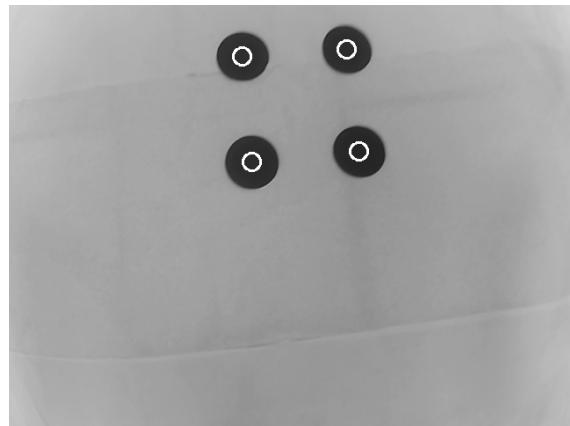


Figure 5.4: Final Result of Marker Detection

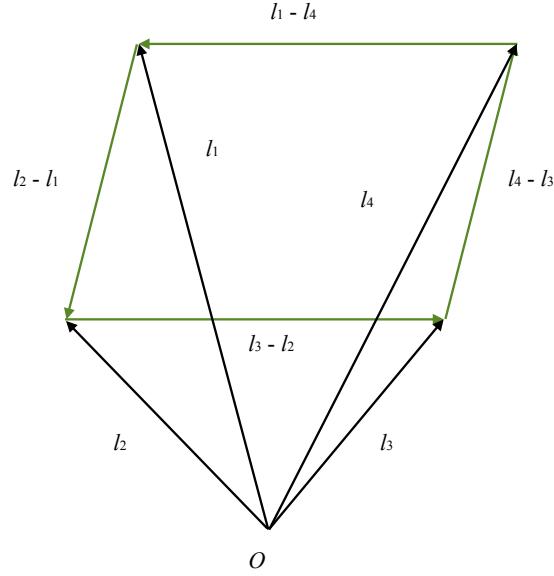


Figure 5.5: Schematic Diagram of the Position Estimation

### 5.3 Position Estimation

Let  $\vec{h}$  be the normal vector of the camera's orientation. Then, from Equation (3.21), 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  $A$  be the area of the polygon in the inertial frame, and  $\vec{l}_i$  be the relative position of a marker according to the position of the camera. Then, the projection of  $A$  is represented as below equation [41].

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

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

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

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 (5.6) can be written as,

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

where  $\Gamma$  is a 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 relative position calculated from the markers is noisy, the Kalman filter is applied to estimate and correct the value. A value computed by Equation (5.12) passes through the Kalman filter, and the filtered value is used as relative position of 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 was not armed, and the autopilot only publishes the sensor values to the onboard companion computer via UART. The algorithm was run by the companion computer in real-time. In order to evaluate the results, the Vicon motion capture system of the arena was used, and streaming data from the position estimator and the motion capture system are compared. As described at

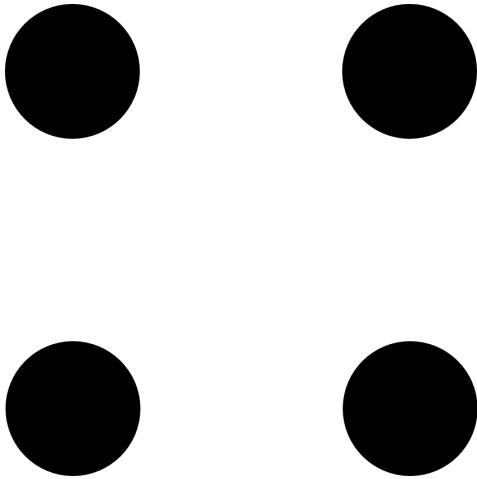


Figure 5.6: Image Markers for the Evaluation

Section 4.3, the Vicon system detects the reflective markers on the quadrotor and measures the position of the quadrotor in its own coordinate system. The image markers used in the experiment for the position estimation are shown in Figure 5.6. Each image marker is a 15cm-diameter-circle and its center is located about 45 cm away from the ones of its neighbors. The placement of the image markers forms a square. The origin of the inertial coordinate system is set to be at 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 coordinate system defined by the motion capture system. The position data from the position estimator were processed by a real-time Kalman filter. The results of the experiments are shown in Figures 5.7 - 5.9. Each figure shows the position measurements of both the above vision-based position estimator and the motion capture system.

#### 5.4.3 Discussion

As shown in Figure 5.7 - 5.9, the errors of the position estimator are about 0.3 m when the position of camera moves slowly. Therefore, it can be concluded that the position estimator works roughly correctly. Considering the experimental results of x, y-axis position errors at Subsection 4.3 are up to 3 m and the errors grow as time pasts, the vision-based estimator is expected to improve the quadrotor's navigation.

However, as Figure 5.9 shows, the position estimator is not able to reflect small change of the position

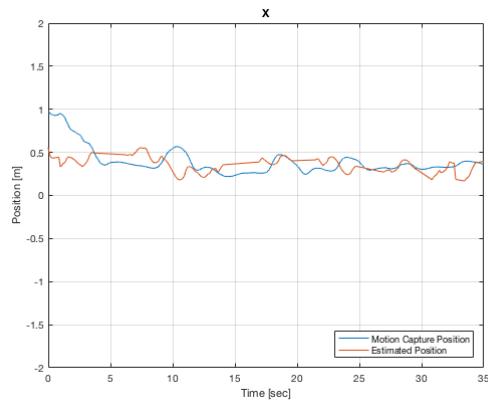


Figure 5.7: Experiment Result of the Position Estimator (x-axis Position)

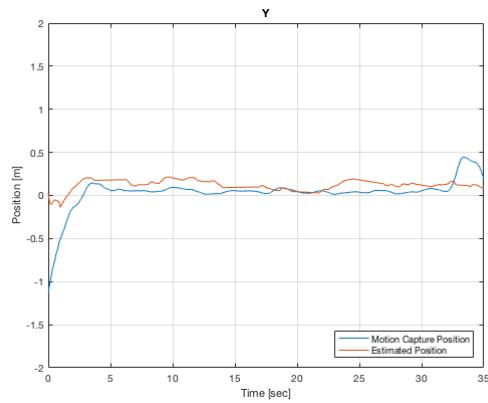


Figure 5.8: Experiment Result of the Position Estimator (y-axis Position)

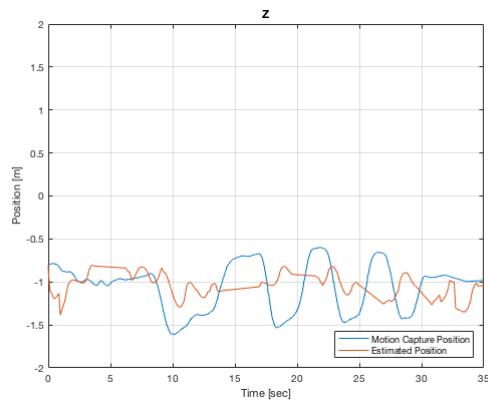


Figure 5.9: Experiment Result of the Position Estimator (z-axis Position)

among z-axis less than 0.5 m. Therefore, the position estimator is more suitable for slow flight, rather than agile performance. In addition, the camera may not capture the image markers if the camera is too close to the ground. The absence of the markers on the image frame can be a critical reason of the failure of position estimation. The problem potentially happens when the quadrotor has high roll or pitch angle, and therefore, it necessary to develop a system that deals with temporary failure of marker detection.

# Chapter 6

## Conclusion

In this research, a dynamic model-based nonlinear attitude control was developed. First, we developed a quadrotor system testbed that satisfies the requirements for potential future research. Then, we studied the dynamics of the quadrotor and developed a nonlinear inner-loop attitude controller based on the model. Further, by characterizing the motor dynamics and analyzing the propeller aerodynamics, we developed an advanced model of the quadrotor that allows more stable and agile control. Based on the model, we developed a nonlinear attitude controller and applied it to the quadrotor system.

The nonlinear control that has been developed in this research has an advantage of globally exponentially stability. Therefore, if a proper dynamic model for the quadrotor can be defined, the control law is expected to allow better performance of the quadrotor. In order to validate the maneuver advantage of the nonlinear controller, the quadrotor's performance was evaluated by both simulations and experiments. The flight performance of the quadrotor with the nonlinear attitude control is compared with the performance of a conventional PID attitude control, and it was verified that the nonlinear attitude control is superior in stability and agility, and therefore, effective for a quadrotor UAVs. By further tuning the gains of the nonlinear controller, we can expect to clearly establish the benefits of the proposed nonlinear control method.

However, in order to develop fully-autonomous system for the quadrotor, it is necessary to install position measurement on the quadrotor. Therefore, in Chapter 5, a computationally-efficient vision-based position estimator was designed. Since the position estimator is executable in real-time, it is expected to be used in the quadrotor system so that the quadrotor corrects accumulated position errors.

As for future works, synchronized with the nonlinear attitude control, the application of the motion planning and guidance is expected to further improve the quadrotor's performance [44][6][43]. Especially for path planning of rapid maneuvering, high nonlinearity of an aerial vehicle is challenging which possibly exacerbates and restricts its maneuver [42]. However, the nonlinear attitude controller developed in this research is expected to guarantee stability and allow fast maneuver. Also, in this research, the motor dynamics is approximated by experiments, but precise modeling of a motor would increase the robustness of the inner-loop control as well. The nonlinear attitude inner-loop controller has an advantage of stability

in terms of angular velocity as well as attitude, and therefore, the nonlinear quadrotor system is expected to be applied to a quadrotor's acrobatic flight that requires high response speed of angular velocity.

# References

- [1] I. C. A. O. Cir, "328 AN/190", Unmanned Aircraft Systems (UAS) Circular, 2011.
- [2] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin, "Quadrotor Helicopter Flight Dynamics and Control: Theory and Experiment?" *Proc. of the AIAA Guidance, Navigation, and Control Conference*, vol. 2. Aug 2007.
- [3] N. Michael, S. Shen, K. Mohta, Y. Mulgaonkar, V. Kumar, K. Nagatni, Y. Okada, S. Kiribayashi, K. Otake, K. Yashida, and K. Ohno "Collaborative Mapping of an Earthquake-Damaged Building via Ground and Aerial Robots," *Journal of Field Robotics*, vol. 29, no. 5, Aug. 2012, pp. 832-841.
- [4] A. Chambers, S. Achar, S. Nuske, J. Rehder, B. Kitt, L. Chamberlain, J. Haines, S. Scherer, and S. Singh, "Perception for a River Mapping Robot," *In Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, IEEE, Sep 2011, pp. 227-234.
- [5] G. V. Raffo, M. G. Ortega, and F. R. Rubio, "Backstepping/Nonlinear  $\mathcal{H}_\infty$  Control for Path Tracking of a Quadrotor Unmanned Aerial Vehicle," *In American Control Conference, 2008*, IEEE, Jun 2008, pp. 3356-3361.
- [6] D. Morgan, G. P. Subramanian, S. . Chung, and F. Y. Hadaegh, "Swarm Assignment and Trajectory Optimization Using Variable-Swarm, Distributed Auction Assignment and Sequential Convex Programming," *The International Journal of Robotics Research*, Feb. 2016. 0278364916632065.
- [7] S. Mallikarjunan, B. Nesbitt, E. Kharisov, E. Xargay, N. 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] C. Diao, B. Xian, Q. Yin, W. Zeng, H. Li, and Y. Yang, "A Nonlinear Adaptive Control Approach for Quadrotor UAVs," *In Control Conference (ASCC), 2011 8th Asian*, IEEE, May 2011, pp. 223-228.
- [9] R. Ritz, Robin, M. Hehn, S. Lupashin, and R. D'Andrea. "Quadrocopter Performance Benchmarking Using Optimal Control," *In Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, IEEE, Sep 2011, pp. 5179-5186.
- [10] D. Mellinger, N. Michael, and V. Kumar, "Trajectory Generation and Control for Precise Aggressive Maneuvers with Quadrotors," *The International Journal of Robotics Research*, Jan 2012, 0278364911434236.
- [11] T. Dierks, and J. Sarangapani, "Output Feedback Control of a Quadrotor UAV Using Neural Networks," *Neural Networks, IEEE Transactions on*, vol. 21, no. 1, 2010 Jan, pp. 50-66.
- [12] "Crazyflie" <http://www.bitcraze.io> [Retrieved 15 April 2016]
- [13] B. Saptarshi, S.-J. Chung, and F. Y. Hadaegh, "Nonlinear Attitude Control of Spacecraft with a Large Captured Object," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 4, Jan 2016 pp. 754-769.
- [14] "3D Robotics Motor AC2830-358," <http://store.3drobotics.com/products/motor-ac2830-358-850kv-1> [Retrieved 22 Sep 2015]

- [15] "3D Robotics ESC 20 Amp with SimonK," <http://store.3drobotics.com/products/esc-20-amp-SimonK-1> [Retrieved 22 Sep 2015]
- [16] "UIUC Airfoil Coordinates Database," [http://m-selig.ae.illinois.edu/ads/coord\\_database.html](http://m-selig.ae.illinois.edu/ads/coord_database.html) [Retrieved 24 Aug 2015]
- [17] "APC Propellers," <http://www.apcprop.com/ProductDetails.asp?ProductCode=LP10047SF> [Retrieved 24 Aug 2015]
- [18] "PX4," <http://dev.px4.io> [Retrieved 15 May 2016]
- [19] "3D Robotics Pixhawk," <http://store.3dr.com/products/3dr-pixhawk> [Retrieved 15 May 2016]
- [20] "Pixhawk Autopilot," <http://pixhawk.org/modules/pixhawk> [Retrieved 15 May 2016]
- [21] "Hardkernel," <http://www.hardkernel.com> [Retrieved 15 May 2016]
- [22] "3D Robotics Radio Set V2," <http://store.3dr.com/products/3dr-radio-set> [Retrieved 22 Sep 2015]
- [23] "Spektrum RC," <http://www.spektrumrc.com/Products/Default.aspx?ProdID=SPMAR610> [Retrieved 27 May 2016]
- [24] "QGround Control," <http://qgroundcontrol.org/mavlink/start> [Retrieved 27 May 2016]
- [25] "MAVROS - ROS Wiki," <http://wiki.ros.org/mavros> [Retrieved 27 May 2016]
- [26] J. J. Craig, "Spatial descriptions and transformations," *Introduction to Robotics: Mechanics and Control*, vol.3, Upper Saddle River: Pearson Prentice Hall, 2005, ch.2, pp. 19-51,
- [27] J. Diebel "Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors," *Matrix*, vol. 58 no. 15-16, Oct 2006, pp. 1-35.
- [28] B. Randal, "Quadrotor Dynamics and Control Rev 0.1," Feb 2008.
- [29] G. P. Subramanian, "Nonlinear Control Strategies for Quadrotors and Cubesats," M.S. Thesis, Department of Aerospace Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, 2015.
- [30] H. Bolandi, M. Rezaei, R. Mohsenipour, H. Nemati, and S. M. Smailzadeh, "Attitude Control of a Quadrotor with Optimized PID Controller," *Intelligent Control and Automation*, vol. 4, no. 3, Aug 2013, pp. 335-342.
- [31] H. K. Khalil, and J. W. Grizzle, "Lyapunov Stability," *Nonlinear systems*, 3rd ed. New Jersey: Prentice hall, 1996. pp. ch. 4, sec. 5, pp. 154-155.
- [32] "UIUC Propeller Data Site," <http://m-selig.ae.illinois.edu/props/propDB.html> [Retrieved 24 Aug 2015]
- [33] K. C. Pohlmann, "Pulse Modulation and Sampling Systems" *Advanced digital audio*, Sams, 1991. ch. 2, pp. 21-53.
- [34] A. A. Mahfouz, M. K. Mohammed, and F. A. Salem, "Modeling, Simulation and Dynamics Analysis Issues of Electric Motor, for Mechatronics Applications, Using Different Approaches and Verification by MATLAB/Simulink," *International Journal of Intelligent Systems and Applications*, vol. 5, no.5 May 2013 pp. 39-57.
- [35] S.-J. Chung, S. Bandyopadhyay, I. Chang, and F. Y. Hadaegh, "Phase Synchronization Control of Complex Networks of Lagrangian Systems on Adaptive Digraphs," *Automatica*, vol. 49, no. 5, May 2013, pp. 1148-1161.
- [36] "Intelligent Robotics Laboratory," <http://robotics.illinois.edu> [Retrieved 18 Apr 2016]

- [37] "Vicon Motion Capture System," <http://www.vicon.com> [Retrieved 18 Apr 2016]
- [38] V. Vezhnevets, A. Velizhev, A. Yakubenko, and N. Chetverikov, "GML C++ Camera Calibration Toolbox," <http://graphics.cs.msu.ru/en/node/909> [Retrieved 13 Jan 2016]
- [39] "Camera Calibration Toolbox for Matlab," [http://www.vision.caltech.edu/bouguetj/calib\\_doc](http://www.vision.caltech.edu/bouguetj/calib_doc) [Retrieved 15 Apr 2016]
- [40] "OpenCV 2.4.12.0 documentation," [http://docs.opencv.org/2.4/doc/tutorials/imgproc/shapedescriptors/find\\_contours/find\\_contours.html](http://docs.opencv.org/2.4/doc/tutorials/imgproc/shapedescriptors/find_contours/find_contours.html) [Retrieved 13 Jan 2016]
- [41] B. M. Haralick, C. N. Lee, K. Ottenberg, M. Nlle, "Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem," *International journal of computer vision*, vol. 1, no. 3, Dec 1994, pp. 331-356.
- [42] A. A. Paranjape, K. C. Meier, X. Shi, S.-J. Chung, and S. Hutchinson, "Motion Primitives and 3D Path Planning for Fast Flight through a Forest," *The International Journal of Robotics Research*, vol. 34, no. 3, Mar 2015, pp. 357-377
- [43] D. Morgan, S.-J. Chung, and F. Y. Hadaegh, "Model Predictive Control of Swarms of Spacecraft Using Sequential Convex Programming," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 6, Apr 2014, pp. 1725-1740.
- [44] H. M. Choset, K. M. Lynch, S. Hutchinson, S. Thrun, L. E. Kavraki, and W. Burgard, *Principles of Robot Motion: Theory, Algorithms, and Implementation*, MIT press, 2005.