Spring 2015

Real-time rss-based indoor navigation for autonomous UAV flight

Sangjun Lee
Purdue University

Graduate School Form 30
Updated 1/15/2015

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Sangjun Lee

Entitled

REAL-TIME RSS-BASED INDOOR NAVIGATION FOR AUTONOMOUS UAV FLIGHT

For the degree of Master of Science in Aeronautics and Astronautics

Is approved by the final examining committee:

Inseok Hwang
Chair

Daniel DeLaurentis

Dengfeng Sun

To the best of my knowledge and as understood by the student in the Thesis/Dissertation Agreement, Publication Delay, and Certification Disclaimer (Graduate School Form 32), this thesis/dissertation adheres to the provisions of Purdue University's "Policy of Integrity in Research" and the use of copyright material.

Approved by Major Professor(s): Inseok Hwang

Approved by: Weinong Chen 4/22/2015

Head of the Departmental Graduate Program Date

REAL-TIME RSS-BASED INDOOR NAVIGATION FOR AUTONOMOUS UAV FLIGHT

A Thesis

Submitted to the Faculty

of

Purdue University

by

Sangjun Lee

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Aeronautics and Astronautics

May 2015

Purdue University

West Lafayette, Indiana

To my family

ACKNOWLEDGMENTS

A number of people have contributions to this study in a variety of ways. First of all I would like to acknowledge faculty of School of Aeronautics and Astronautics for giving me the chance to pursue my goals. I especially wish to express my gratitude to my advisor Professor Inseok Hwang for his advice and guidance in developing this thesis. I also would like to thank other committee members, Professor Dengfeng Sun and Professor Daniel DeLaurentis. I greatly appreciate all of their comments and suggestions during my program.

I want to thank Byung-cheol Min who inspired me to be interested in the field of wireless network as a research partner. I am also most grateful to Minsu Kim and Suzy Ahn who have given me help and encouragement. I appreciate the support from all members of the Flight Dynamics and Control / Hybrid System Lab, Sangjin Lee, Kwangyeon Kim, Cheolhyun Kwon, James Goppert, and Scott Yantek, who have assisted in this research.

Finally, my parents, Jinwoo and Insook, and sister, Hyerin, have assisted me in innumerable ways, whatever I might say here cannot do full justice to the extent and the value of their contribution toward me. This work would never have become what it now is without your endless support and love.

TABLE OF CONTENTS

Page

LIST OF TABLES vi

LIST OF FIGURES vii

SYMBOLS ix

ABBREVIATIONS xi

ABSTRACT xii

1 INTRODUCTION 1
- 1.1 Motivation and Background 1
- 1.2 Literature Review 1
  - 1.2.1 Vision-based Indoor Navigation 1
  - 1.2.2 Wireless Sensor Network-based Indoor Navigation 2
  - 1.2.3 Features Comparison 3
- 1.3 Contributions 4
  - 1.3.1 Organization 6

2 INDOOR NAVIGATION SYSTEM DEVELOPMENT 7
- 2.1 Overview 7
- 2.2 RSS-based Navigation System 8
- 2.3 Path Loss Model 9
  - 2.3.1 Identification and Analysis of Sources of Uncertainty 10
  - 2.3.2 Path-loss Exponent Optimization 13
  - 2.3.3 Transmitted Signal Strength Optimization 14
- 2.4 Distance Estimation 16
- 2.5 Position Measurement 18

3 AUTONOMOUS FLIGHT SYSTEM DEVELOPMENT 21
- 3.1 Overview 21
- 3.2 Vehicle Model 22
- 3.3 State Estimation 27
  - 3.3.1 Sensor Fusion via Extended Kalman Filter 28
  - 3.3.2 System Model 29
  - 3.3.3 IMU Measurement Model 29
  - 3.3.4 RSS Measurement Model 30
- 3.4 Control 32

4 SIMULATION 35

Page

4.1 Overview . . . 35
4.2 Simulation of the Autonomous UAV Flight System . . . 36
4.2.1 Gain Tuning . . . 36
4.2.2 Scenario 1: Circular Trajectory . . . 37
4.2.3 Scenario 2: Hallway . . . 39

5 EXPERIMENT . . . 43
5.1 Overview . . . 43
5.2 Experiment Testbed . . . 43
5.2.1 Software . . . 43
5.2.2 Hardware . . . 46
5.2.3 General Setup . . . 49
5.3 RSS-based State Estimation . . . 50
5.4 Autonomous Flight Test . . . 52
5.4.1 Mission 1: Position Hold . . . 54
5.4.2 Mission 2: Straight Waypoints . . . 56
5.4.3 Mission 3: Square Spiral Waypoints . . . 60

6 CONCLUSIONS AND FUTURE WORKS . . . 63

LIST OF REFERENCES . . . 64

A Extended Kalman Filter . . . 67

B Simulation Setup . . . 69

LIST OF TABLES

Table		Page
1.1	Comparison of Autonomous Flight Systems for an Indoor Environment	4
4.1	Scenario 2: RSS-based State Estimation Error	40
5.1	Software and Hardware Used in Experiment	44
5.2	RSS-based State Estimation Error	51
5.3	Mission 1: Errors of Autonomous UAV Flight	55
5.4	Mission 2: Errors of Autonomous UAV Flight	57
5.5	Mission 3: Errors of Autonomous UAV Flight	60
B.1	Simulation Parameters	71

LIST OF FIGURES

Figure		Page
2.1	Overview of the Proposed RSS-based Indoor Navigation System	8
2.2	Path Loss Model	10
2.3	Propagation Maps of RSS Measurements	11
2.4	Characteristics of RSS measurements	12
2.5	Distance Estimates with Two Different n_i	14
2.6	Range of RSS measurements for Two Different P_T	15
2.7	Distance Estimation for Two Different P_T	16
2.8	Error Ellipse of Distance Estimation	18
3.1	Architecture of the Proposed RSS-based Indoor Navigation System	22
3.2	Vehicle Model	23
3.3	States Flow of the Proposed Autonomous System	28
3.4	xy-position Controller	33
3.5	Heading Controller	33
3.6	Altitude Controller	33
4.1	Simulation Block Diagram	35
4.2	Gain Optimization	36
4.3	Scenario 1: UAV Control	37
4.4	Scenario 1: UAV State Estimation	38
4.5	Scenario 1: State Estimaition	39
4.6	Scenario 2: UAV Position Estimation	40
4.7	Scenario 2: UAV Position Estimation(Scale-up)	41
4.8	Scenario 2: Partial Propagation Map of RSS Measurement	41
5.1	Wireless Sensor Networks Modules	46
5.2	Tracking Algorithm of MS Kinect	48

Figure Page

5.3 Object Detection and Tracking using MS Kinect 48
5.4 Experiment Testbeds 49
5.5 Schematic Overview of the Experiment 50
5.6 RSS-based UAV State Estimation 52
5.7 RSS-based UAV Position Estimation 52
5.8 Architecture of Autonomous Flight System 53
5.9 Mission 1: UAV Position 55
5.10 Mission 1: Simulation of Autonomous UAV Flight 56
5.11 Mission 1: Experiment of Autonomous UAV Flight 57
5.12 Mission 2: UAV Position 58
5.13 Mission 2: Simulation of Autonomous UAV Flight 59
5.14 Mission 2: Experiment of Autonomous UAV Flight 59
5.15 Mission 3: UAV Position 61
5.16 Mission 3: Simulation of Autonomous UAV Flight 61
5.17 Mission 3: Experiment of Autonomous UAV Flight 62
B.1 Integrated Block Diagram in Simulink 69
B.2 Subsystem of Position and Attitude Controller 70
B.3 Subsystem of Motor Controller 70

SYMBOLS

$\mathbf{L}$	position vector
$\mathbf{\Phi}$	Euler angle vector
$\mathbf{V}$	velocity vector
$\mathbf{\Gamma}$	angular velocity vector
x	East position
y	North position
z	altitude
ϕ	roll angle
θ	pitch angle
ψ	yaw angle
u or v_x	East velocity
v or v_y	North velocity
w or v_z	vertical velocity
p	roll rate
q	pitch rate
r	yaw rate
$\mathbf{b}_a$	accelerometer bias vector
$\mathbf{b}_g$	gyroscope bias vector
d_i	distance between the node i and the mobile node
$\mathbf{x}$	state vector
I_x, I_y, I_z	body moments of inertia
J_r	rotor moment of inertia
m	mass of quadrotor
g	standard gravity
Ω_i	speed of rotor i

U_i	input of motor i
E	inertial reference frame
B	body reference frame
P_T	transmission power
G_R	receiver gain
G_T	transmitter gain
n_i	path-loss exponent at static node i
$\mathbf{w}, \mathbf{v}$	Gaussian noise vector

ABBREVIATIONS

CoG	Center of Gravity
CRLB	Cramer-Rao Lower Bound
EKF	Extended Kalman Filter
GCS	Ground Control Station
GPS	Global Positioning System
IMU	Inertial Measurement Unit
KF	Kalman Filter
LOS	Line Of Sight
MLE	Maximum Likelihood Estimator
nesC	Network Embedded Systems C
OS	Operating System
PD	Proportional Derivative
RSS	Received Signal Strength
RMS	Root Mean Square
SLAM	Simultaneous Localization And Mapping
TDoA	Time Difference of Arrival
ToA	Time of Arrival
UAVs	Unmanned Aerial Vehicles
UASs	Unmanned Aerial Systems
WSNs	Wireless Sensor Networks

ABSTRACT

Lee, Sangjun M.S.A.A., Purdue University, May 2015. Real-time RSS-based Indoor Navigation for Autonomous UAV Flight. Major Professor: Inseok Hwang.

Navigation for the autonomous flight of Unmanned Aerial Vehicles (UAVs) in an indoor space has attracted much attention recently. One of the main goals of an indoor navigation system is developing an alternative method to obtain position information that can replace or complement the global positioning system. While much research has focused on vision-based indoor navigation systems, this paper aims to develop a Received Signal Strength (RSS)-based navigation system, which is a more cost effective alternative. Then, the position and attitude of a UAV can be computed by the fusion of RSS measurements and measurements from the onboard inertial measurement unit. In order to improve the estimation accuracy, we first consider a mathematical model of the RSS-based navigation system and formulate optimization problems to compute the parameter values which minimize the RSS measurement error. Using the optimal parameters, an autonomous flight system is developed whose estimator and controller components are designed to work well with the RSS-based navigation system. Simulations and experiments using a quadrotor demonstrate the feasibility and performance of the proposed RSS-based navigation system for UAVs operating in indoor environments.

1. INTRODUCTION

1.1 Motivation and Background

Over the past decade, the use of Unmanned Aerial Vehicles (UAVs) has increased in a wide range of applications such as search and rescue operations, civil and military surveillance, and tracking assets in warehouses [1]. Since the most fundamental function for a UAV in these applications is to move to a desired location, one of the key issues is to develop a navigation system which is able to precisely determine the location of the UAV. The combination of an Inertial Measurement Unit (IMU) and an external source such as the Global Positioning System (GPS) has been widely used to provide position information to the navigation system of the UAV. While GPS generally provides accuracy of a few meters, current GPS technology is not suitable for indoor applications since the reception of GPS signals is unreliable or even unavailable [2]. Therefore, an alternative method is required to replace the role of GPS in UAVs for indoor applications.

1.2 Literature Review

Extensive research has been conducted to develop alternative sensors for indoor navigation, which can be categorized into two main classes: vision-based systems and wireless sensor network-based systems.

1.2.1 Vision-based Indoor Navigation

A number of studies have been conducted for vision-based indoor navigation, such as the combination of IMU and optical flow measurements [3, 4], Simultaneous Localization And Mapping (SLAM) [5], ultrasonic-based range finding [6] and a motion

capture camera as an external vision source. Most vision sensors are light-weight and easy to implement, and these features encourage researchers to employ them for indoor navigation. While these approaches contribute to indoor navigation, they have a few limitations. For example, most vision sensors are highly dependent on light and thus impossible to operate without a source of light. Furthermore, even though the Vicon [7], a motion capture solution, provides the position of an object with millimeter accuracy, its implementation is very costly and its coverage is too restricted. SLAM also offers an accurate position estimation but requires considerable computational resources since it needs to filter out useful information from the large amount of collected images. This necessitates that the vehicle move slowly to ensure enough computation time.

1.2.2 Wireless Sensor Network-based Indoor Navigation

There are three positioning methodologies commonly used in Wireless Sensor Network (WSN)-based localization: Time of Arrival (ToA), Time Difference of Arrival (TDoA), and Received Signal Strength (RSS) [8]. First, ToA is the method that calculates the distance based on the time of transmission and the propagation speed of the signal. ToA is not widely used since measurements acquired from the propagation speed are irregular depending on temperature or humidity, which causes a disruption of exact time synchronization. Second, an estimated distance using TDoA is obtained by using a difference in arrival times of two signals with different propagation speeds originating at the same node. This means that the TDoA method requires two types of receivers and transmitters for every node, necessitating a costly and energy-intensive system. Lastly, RSS-based localization determines the position of an object by calculating the difference between the transmitted power from the transmitter and the received power at the receiver. Among these three WSN-based indoor navigation systems, RSS-based localization has been most widely used since it has the physical advantages of being low-weight, low-complexity, and low-power.

Although RSS-based localization has desirable properties, its effectiveness is questionable [9, 10] due to the following issues: raw received signals are corrupted with noise, and the strength of the signal is vulnerable to conditions of signal propagation such as signal interference, path attenuation, and transmission power. Dealing with such problems, some research employs statistical and mathematical techniques to improve the accuracy [11–15]. Also, sensor fusion algorithms that combine the RSS measurements with measurements from other sources achieves reasonable performance. However, they do not rigorously consider several factors inherent in a radio propagation model, such as transmitted signal strength and path-loss exponent, which affect the performance of the RSS-based navigation system. Thus, a detailed analysis to the radio propagation model is required to choose appropriate parameter values to enable better indoor navigation performance.

1.2.3 Features Comparison

A comparison of the accuracy, complexity, and cost of other autonomous flight systems for indoor environments is represented in Table 1.1. As shown in the table, there is a trade-off between performance and cost. This research proposes an autonomous flight system with RSS-based navigation system that is cost-effective while providing accurate enough position information for use in an indoor environment. In order to improve the estimation accuracy, we first consider a mathematical model of the RSS-based navigation system and formulate optimization problems providing optimal parameter values for the minimum estimation error. Using the optimal values, we integrate the RSS-based navigation system into the autonomous flight system to validate the effectiveness of our proposed navigation system in an indoor environment.

Table 1.1. Comparison of Autonomous Flight Systems for an Indoor Environment

Source	Vision-based				WSN-based
System	Optical flow	SLAM	Range-finder	Motion capture	RSS
References	[4]	[5]	[6]	[16]	
Accuracy	25 cm	10 cm	5 cm	0.5 mm	20 cm
Complexity	Moderate	High	High	Very high	Low
Cost	Low	Medium	Medium	Very high	Very low

1.3 Contributions

The ultimate goal of this research is to develop an autonomous UAV flight system for use in an indoor environment. To do this, this research breaks the problem into four tasks, as follows:

First, this research provides a UAV-specific RSS-based indoor navigation system which is cost effective, low power, and low complexity. Traditionally, the performance of the RSS-based localization algorithm in wireless sensor networks strongly depends on the quality of the RSS measurements, which are affected by current propagation conditions such as transmission power, time delay, path attenuation, and multi-path propagation. Unlike previous studies which used raw RSS measurements without consideration for these conditions, this research identifies sources of uncertainty and their effects on the navigation performance both physically and nonphysically. Hence, this research proposes strategies to mitigate the effects of errors, thereby improving the entire performance of the indoor navigation system by applying optimization and statistical techniques. Specifically, this research formulates an optimization problem to determine an optimal value of the path-loss exponent which describes the path attenuation of propagation conditions of the signal. Also, this research formulates an optimization problem with several constraints to determine an optimal value of the

transmission power which minimizes the difference between the expected RSS and the measured RSS. Additionally, the theoretical bound determined by the Cramer-Rao lower bound is defined to evaluate the performance of distance estimation. Based on this distance between the UAV and each WSN module, the real-time position of the UAV is determined by solving the multilateration problem. Finally, this RSS-based indoor navigation system is proposed as a solution for developing an alternative navigation system in an indoor environment.

Second, this research develops an autonomous UAV flight system based on the proposed indoor navigation system. In order to do this, this research develops a real-time feedback control system with Proportional-Derivative (PD) controllers and an estimator. The state estimator is developed by combining IMU and RSS measurements via an Extended Kalman Filter. It is novel in that it develops sensor fusion between the dynamics of the UAV as a system model and the IMU/RSS sensors as measurement models. The RSS measurement is used to correct biases of the IMU measurement using the UAV dynamics. The PD controllers corresponding to the estimator are also developed. The strategy to find an optimal gain value (which significantly affects performance) of each controller is introduced to improve the entire performance of the proposed autonomous flight system.

Third, numerical simulations are performed in order to demonstrate the feasibility of the proposed indoor navigation system and autonomous flight system. Firstly, this research shows the performance of distance estimation using the proposed RSS-based indoor navigation system. Secondly, two different simulation scenarios which contain various possible situations are considered to show the performance of the proposed autonomous UAV flight system in terms of flexibility.

Fourth, as the ultimate goal of this research, the entire performance of the proposed autonomous flight system and its reliability are demonstrated by testing three different missions using a UAV. Light-weight, cost-effective, and simply-structured WSN modules are used to provide RSS measurement using several open source software packages. Also, the external reference used to calculate state estimation errors

is introduced. Finally, the expected performance based on simulation results is compared to the actual performance based on experiment results to demonstrate the feasibility of the proposed autonomous UAV flight system.

Lastly, the experiment and simulation results support that the proposed autonomous flight system based on the proposed RSS-based indoor navigation system is a feasible and reliable solution for the autonomous flight of UAVs in a GPS-denied environment with low-cost, low-weight, low complexity, and relatively high control performance.

1.3.1 Organization

The rest of this paper begins with the development of the RSS-based navigation system in Chapter 2, including details about the system, and the optimization of parameters related to the system's performance. Chapter 3 presents a description of the autonomous flight system with the RSS-based navigation system. Simulation and flight test results are shown in Chapters 4 and 5, and Chapter 6 presents conclusions and future works.

2. INDOOR NAVIGATION SYSTEM DEVELOPMENT

2.1 Overview

The navigation system of UAVs generally acquires current position information from GPS. The performance of the navigation system is highly dependent on the quality of GPS measurement. In this research, however, RSS measurements will be used as an alternative to GPS measurements for the indoor navigation since the GPS signal in indoor environments is usually weak or totally unavailable. Therefore, this chapter focuses on the challenge of developing a RSS-based indoor navigation system to provide a reliable real-time position for UAVs. A schematic diagram of the proposed RSS-based indoor navigation system is shown in Figure 2.1. Firstly, a fundamental theory of the RSS-based system in a WSN is presented. Secondly, the effects of inherent error of the RSS-based navigation system are presented by using distance estimation with raw RSS measurements. Sources of errors are identified in both physical and nonphysical ways to clarify the relation between the performance of distance estimation and those errors. Thirdly, this chapter defines several parameters such as a path-loss exponent and a transmission power which can affect the entire system. Finally, this chapter provides a RSS-based indoor navigation system to improve the performance of position estimation by formulating and solving several optimization problems.

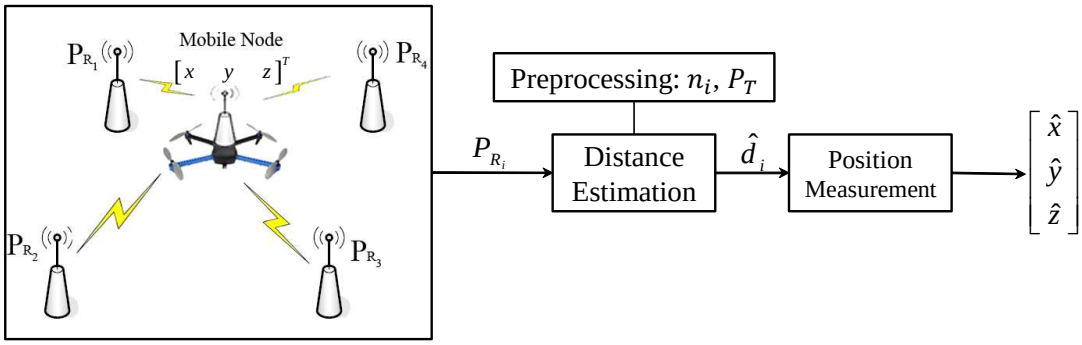


Figure 2.1. Overview of the Proposed RSS-based Indoor Navigation System

2.2 RSS-based Navigation System

In this section, a detailed description of the RSS-based navigation system is first presented, followed by a mathematical model of the RSS-based navigation system. With this model, optimization problems are formulated to find parameter values to optimize the performance of the RSS-based navigation system. The distance estimation and position measurement provide a reliable real-time position information for the UAV by using the mathematical model and the optimized parameters. The architecture of the proposed RSS-based navigation system is presented in Figure 2.1.

The leftmost block of Figure 2.1 represents the overall configuration of the RSS-based navigation system. Each node is able to both transmit and receive real-time data packets including the current battery life, received signal strength with timestamp, and transmitted signal strength with timestamp. Nodes are categorized into two different types in order to distinguish the role of each node. The first type of node is referred to as a static node, and has a fixed location which is known. The position of i-th static node is defined as a constant vector:

$$\mathbf{L}_i := \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^T \tag{2.1}$$

The second type of node is a mobile node, and is mounted on the UAV. The position of the mobile node at a given time t can be defined as

$$\mathbf{L}(t) := \begin{bmatrix} x(t) & y(t) & z(t) \end{bmatrix}^T \tag{2.2}$$

Then, the distance between the mobile node and the i-th static node at a given time t is given by

$$\begin{aligned} d_i(t) &:= \|\mathbf{L}(t) - \mathbf{L}_i\| \\ &= \sqrt{(x(t) - x_i)^2 + (y(t) - y_i)^2 + (z(t) - z_i)^2} \end{aligned} \tag{2.3}$$

Thus, the position of the mobile node can be determined from (2.3) if at least four distances are available. In order to obtain the distances, the real-time data packets between the static node and the mobile node are used as discussed in the following section.

In this paper, communication between each node follows the IEEE 802.15.4 standard, which intends to offer cost-effective communication and low power consumption without additional infrastructure [17]. These features allow that WSN modules used in our experiment operate with low complexity and low cost.

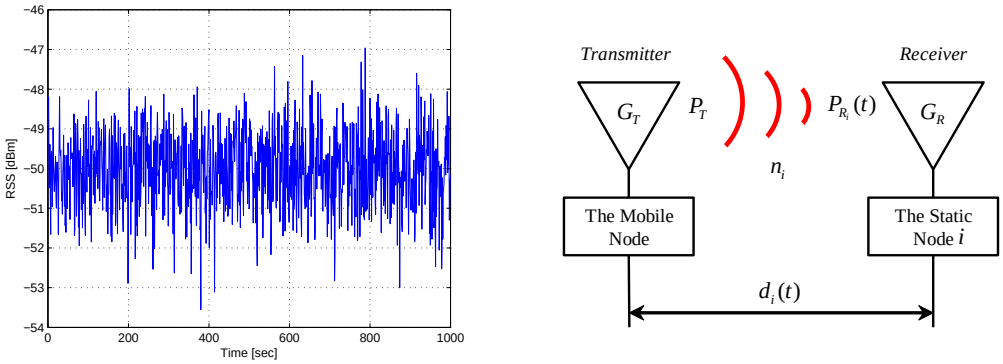
2.3 Path Loss Model

A relation between the received signal strength and the distance between two nodes can be modeled by the path loss model [8] which describes power attenuation (or loss) of a signal due to distance. At a given time t, the Received Signal Strength (RSS) at the i-th static node, $P_{R_i}(t)$, is expressed in dBm:

$$P_{R_i}(t) = P_T + G_R G_T - 10 n_i \log_{10}\{d_i(t)\} + \mathrm{w}_{\mathrm{RSS}} \tag{2.4}$$

where P_T is the transmitted signal strength at the mobile node; G_R and G_T are the receiver gain and the transmitter gain, respectively, which are given by the technical specifications of an antenna; n_i is the path-loss exponent for each static node, which represents the attenuation of the signal due to the distance traveled in a given environment and $\mathrm{w}_{\mathrm{RSS}}$ is the measurement noise due to multipath propagation caused

by uncertainties or irregularities. Figure 2.2(a) shows the time history of actual RSS measurements for a fixed distance. From this figure, the measurement noise w_{RSS} can be modeled as Gaussian white noise with zero mean. Figure 2.2(b) visually represents the path loss model.



(a) Actual RSS measurements

(b) Path Loss Model in Wireless Sensor Network

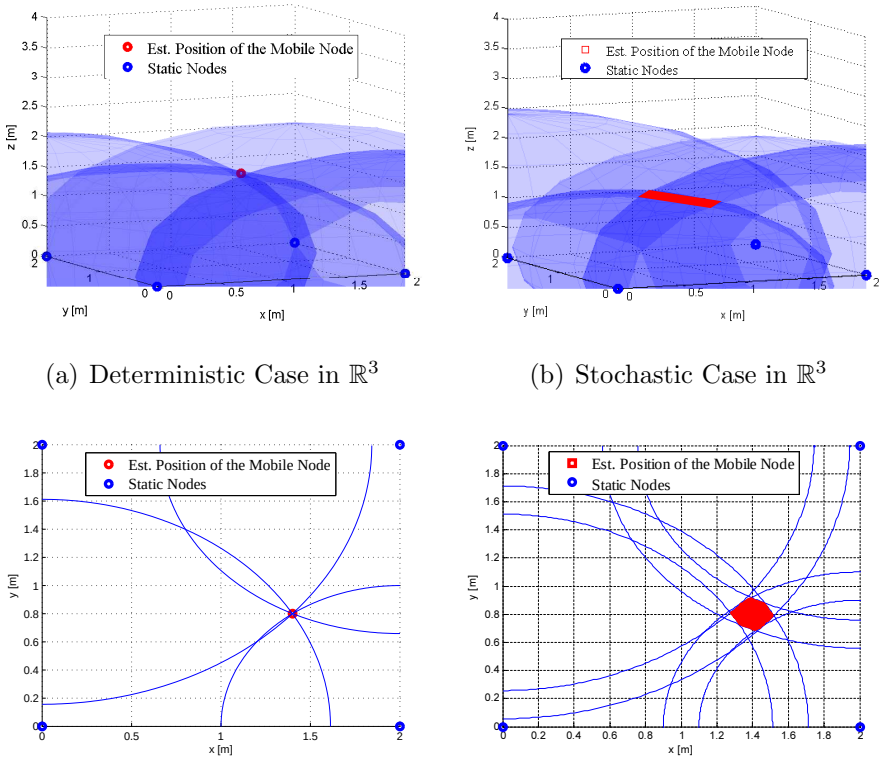
Figure 2.2. Path Loss Model

With given G_R, G_T, n_i, and P_T, $d_i(t)$ can be determined from (2.4) once $P_{R_i}(t)$ is obtained. In what follows, we formulate optimization problems to find optimal values of the other two parameters, n_i and P_T, as represented in the *Preprocessing* block in Figure 2.1.

2.3.1 Identification and Analysis of Sources of Uncertainty

In the real world, the challenge of distance estimation via the path loss model comes from the phenomenon that raw RSS measurement contains uncertainties which cause a degrading of the estimation performance. The numerical simulation result in Figure 2.3 demonstrates the impact of measurement noise on the distance estimation. While the deterministic cases, i.e., ideal case, in both three and two dimensional space produce linear quarter-sphere and quarter-circle with the single coincidence in red,

the shape of distance estimation about stochastic cases, i.e., real-world case, forms a random intersection which contains all candidates for the position of the mobile node in red as shown in Figure 2.3(b) and Figure 2.3(d). Therefore, as the intersection is getting larger, the estimation error is also getting bigger, and therefore the tracking accuracy will decreases. Parameters in (2.4) such as P_{R_i} and n have the ability to affect the performance of position estimation and these are highly dependent on propagation environments. In wireless sensor networks, several major sources affect conditions of communication environment [18] are usually introduced.



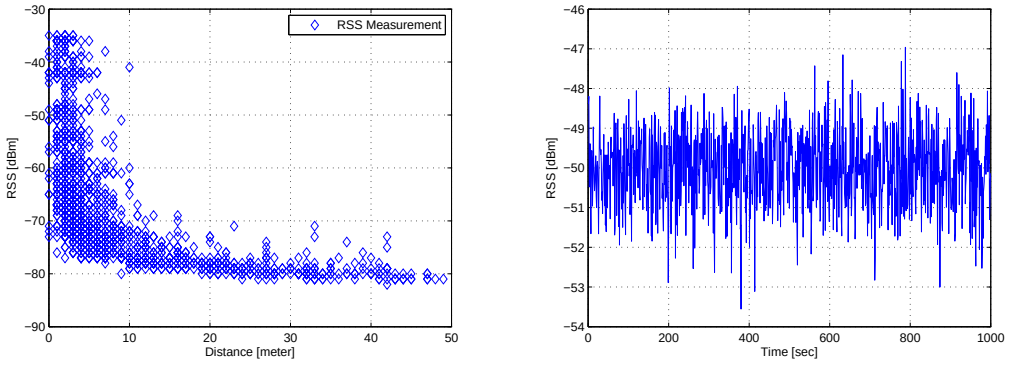
(a) Deterministic Case in $\mathbb{R}^3$ (b) Stochastic Case in $\mathbb{R}^3$

(c) Deterministic Case in $\mathbb{R}^2$ (d) Stochastic Case in $\mathbb{R}^2$

Figure 2.3. Propagation Maps of RSS Measurements

Physical sources

The effect of multipath and shadow fading are mainly concerned with physical conditions such as node configuration and the position of node. Multipath and shadow fading are random phenomenon in wireless communication and they are difficult to predict because they vary depending on a current communication circumstance. Figure 2.4(a) shows how actual RSS measurements change as a distance between the transmitter and the receiver goes far. A number of RSS measurements is dramatically decreased if the distance approximately exceeds over 10 meters due to the effect of multipath and shadow fading. Insufficient measurements lead to the low performance of estimation. Therefore, it is necessary to introduce appropriate definitions to describe a relationship between a random phenomenon and an estimation performance. A random behavior of sensor measurement is represented as Gaussian distribution. As shown in Figure 2.4(b), actual RSS measurements between two static nodes are similar to the tendency of the white noise. Also, the Line of Sight (LOS) condition between nodes is considered to make the estimation problem simple without the loss of generality. These two assumptions help the proposed indoor navigation system to acquire better position estimation.



(a) Tendency of RSS measurements (b) RSS measurements at constant distance

Figure 2.4. Characteristics of RSS measurements

Nonphysical sources

The performance of position estimation is also prone to nonphysical sources such as systematic errors, unique characteristic of devices, transmission rate, and unknown parameters of the radio propagation model. The notable difference between physical and nonphysical sources is that nonphysical sources can be determined by a user. It means that the overall performance of position estimation is improvable if it is possible to find an optimal value of parameters in (2.4) for propagation environments. In the following section, the proposed indoor navigation system shows how to mitigate these effects using proposed estimation algorithms and statistical techniques.

2.3.2 Path-loss Exponent Optimization

The goal of this section is to find the optimal value of n_i which represents the signal attenuation. A value of the path-loss exponent is typically determined from an empirical table [18, 19]. For example, the path-loss exponent is given as about 2 for an indoor environment, while it is 4 to 6 for a cluttered space. However, the value from the table might not be accurate enough for a specific environment, and this inaccuracy can degrade the performance of the distance estimation. Therefore, identifying an accurate path-loss exponent, which can be optimally adapted to unknown environments, can improve the quality of the distance estimation.

We formulate an optimization problem which minimizes the error between a measured distance and a predefined distance. Since the positions of the static nodes are given, the distance between the i-th static node and the j-th static node is defined as

$$d'_{ij} := \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \tag{2.5}$$

for $i \neq j$. Note that $d'_{ij} = d'_{ji}$. Also, all nodes can both transmit and receive a signal, which enables a static node to transmit a signal to other static nodes. Hence, the distance between the i-th static node and the j-th static node, d_{ij}, can be determined from (2.4) by regarding one node as a transmitter and the other as

a receiver. Assuming that the transmitted signal strength P_T is given, the path-loss exponent n_i is the only variable in (2.4). Then, the optimal path-loss exponent at the i-th static node can be computed by solving the following optimization problem that minimizes the square error between the distance d'_{ij} in (2.5) and the distance d_{ij} in (2.4) with l number of samples:

$$n_i^* = \arg\min_{n_i} \left\{ \sum_{k=1}^{l} \left[d'_{ij} - d_{ij}(k) \right]^2 \right\} \tag{2.6}$$

Under the assumption that several nodes in the same space have the same path-loss exponent, the path-loss exponents of the other static nodes are the same as the $n_i{}^*$.

As an example, n_i is optimized for an indoor free space. The plots of the distance estimates when $n_i = 2$ and $n_i^* = 1.78$ for a fixed distance are presented in Figure 2.5. It is clear that the error when $n_i^* = 1.78$ is smaller than the error when $n_i = 2$.

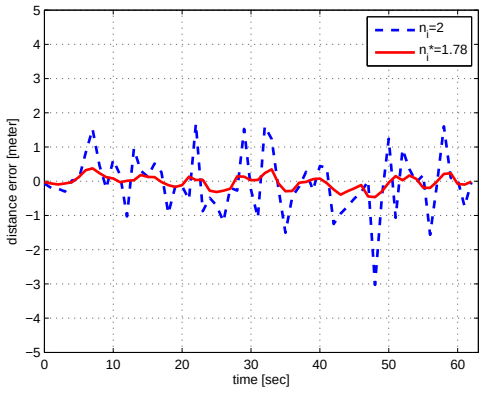


Figure 2.5. Distance Estimates with Two Different n_i

2.3.3 Transmitted Signal Strength Optimization

An optimization of the transmitted signal strength is represented in this section. Similar to the path-loss exponent, the transmitted signal strength in (2.4) can degrade the quality of the distance estimation if inappropriate transmitted signal strength is used. The transmitted signal strength is typically set to be a constant [20, 21]. As

shown in Figure 2.6, identical transmission powers for a fixed distance can result in several different values of RSS due to the effects of the measurement noise. This means that the range of RSS measurements δ, which is defined as the difference between the largest and smallest values, can vary depending on the transmitted signal strength. As an example, Figure 2.6 shows that the RSS measurements with $P_T = -25$dBm have a smaller range than $P_T = -15$ dBm.

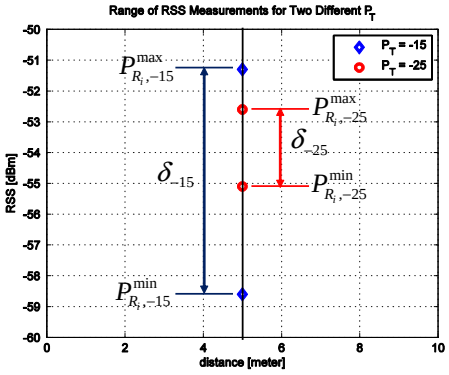


Figure 2.6. Range of RSS measurements for Two Different P_T

We now formulate an optimization problem to minimize the range. Similar to the path-loss exponent optimization, a static node sends a signal to other static nodes with the given distance to find an optimal transmitted signal strength. Then, the optimization problem can be described as

$$\textbf{minimize} \quad \left\{ \max P_{R_i}(P_T) - \min P_{R_i}(P_T) \right\}^2 \tag{2.7}$$

$$\textbf{subject to} \quad P_{T\min} \leq P_T \leq P_{T\max} \tag{2.8}$$

where the objective function is the range of the RSS measurements. The constraint is from physical conditions, which can be identified by the technical specification and the transmission range.

As an example of numerical simulation, the distance estimation using actual RSS measurements for two different P_T values is presented in Figure 2.7. It is shown that the error is reduced when $P_T^* = -31.38$ dBm compared to $P_T = -25$ dBm, which

means that the optimal parameter value with a smaller range results in the smaller error in the distance estimation. Therefore, the quality of the distance estimation improves if we obtain an optimal transmitted signal strength which minimizes the range of RSS measurements.

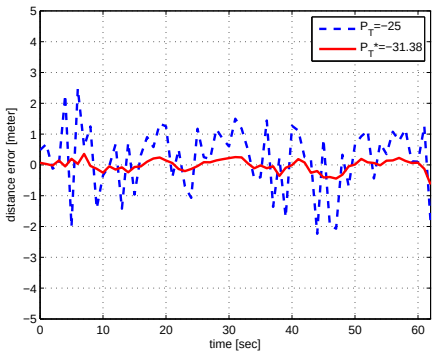


Figure 2.7. Distance Estimation for Two Different P_T

2.4 Distance Estimation

In Sections 2.3.2 and 2.3.3, the values of the optimal path-loss exponent and the transmitted signal strength are optimized. With the optimal parameter values, this section formulates the distance estimation using the Maximum Likelihood Estimator (MLE). This section is represented as the *Distance Estimation* block in Figure 2.1.

For the proposed RSS-based navigation system, let us consider a random variable P_{R_i} whose probability density function (pdf) $p(P_{R_i}; \tilde{\mathbf{d}}_i)$ depends on a measurement vector $\tilde{\mathbf{d}}_i = [\tilde{d}_i(1), ..., \tilde{d}_i(l)]^T$. As defined in Section 2.3, $P_{R_i}(1), ..., P_{R_i}(l)$ are identically and independently distributed (iid) random variables with normal distribution, $P_{R_i} \sim \mathcal{N}(\overline{P}_{R_i}, \sigma^2)$ with mean $\overline{P}_{R_i}$ and variance σ^2. Thus, the likelihood function can be defined as

$$\mathcal{L}(d_i) = \prod_{k=1}^{l} p\Big(P_{R_i}(k); \tilde{\mathbf{d}}_i(k)\Big) \tag{2.9}$$

Under the assumption that regularity conditions hold [22], we can define the MLE as

$$\hat{d}_i = \arg\max_{d_i} \ln \mathcal{L}(d_i) \tag{2.10}$$

By taking partial derivatives of $\ln \mathcal{L}(d_i)$ with respect to d_i, we obtain the estimated distance given RSS measurements as

$$\hat{d}_i = 10^{\frac{G_R G_T + P_T - \overline{P}_{R_i}}{10 n_i}} \tag{2.11}$$

Another theory of statistical inference called the CRLB which gives a lower bound on the variance of any unbiased estimator. For our estimator, the CRLB with l samples is defined as

$$\begin{aligned} \text{CRLB} &= \mathcal{F}^{-1} \\ &= \left\{ -\mathbb{E}\Big[\frac{\partial^2}{\partial d_i^2} \ln f\Big(P_{R_i}(k); \tilde{\mathbf{d}}_i(k)\Big)\Big] \right\}^{-1} \\ &= \frac{\sigma d_i}{\sqrt{l} 10 n_i \log_{10}(e)} \end{aligned} \tag{2.12}$$

where $\mathcal{F}$ is called Fishers information. The statistics $\hat{d}_i$ is an efficient estimator of d_i if and only if the variance of $\hat{d}_i$ attains the CRLB [22]. Thus, we can say that the distance estimation of the RSS-based navigation system is an efficient estimator of d_i if the variance of $\hat{d}_i$ achieves the following inequality:

$$Var(P_{R_i}) \geq \text{CRLB} \tag{2.13}$$

As an example of numerical simulation for the distance estimation, the Figure 2.8 shows two dimensional maps for the mean and the standard deviation of actual RSS measurements. In case of the RSS-based navigation system estimates the distance based on raw RSS measurements, as marked pink dots in the figure, the estimated distance is quite far away from the true value because poor measurements are also used in estimation. Therefore, we can improve the quality of the distance estimation by defining the MLE and CRLB in this section.

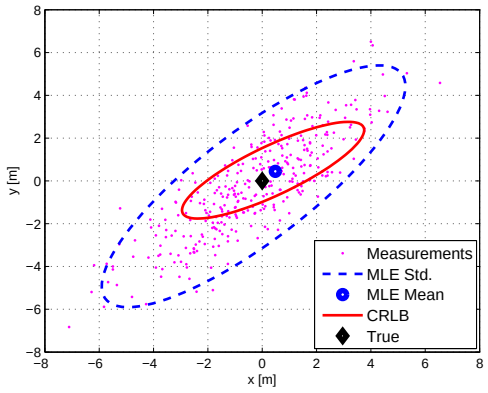


Figure 2.8. Error Ellipse of Distance Estimation

2.5 Position Measurement

In this section, we present position calculation for the mobile node using the distance estimate from Section 2.4. This section is represented as the *Position Measurement* block in Figure 2.1. We use the trilateration-based method to obtain the mobile node's position. Thus, the distance derived in (2.11) can be used to obtain the position of the mobile node at a given time k:

$$\hat{d}_i(k) = \sqrt{\left(\hat{x}(k) - x_i\right)^2 + \left(\hat{y}(k) - y_i\right)^2 + \left(\hat{z}(k) - z_i\right)^2} \tag{2.14}$$

for $i = 1, ..., N$. Note that $N \geq 4$ as dicussed in Section 2.2. By expanding and rearranging, (2.14) can be rewritten in a matrix form:

$$\mathbf{AL}(k) = \mathbf{y} \tag{2.15}$$

where

$$\mathbf{A} = 2\begin{bmatrix} x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_2 & y_3 - y_2 & z_3 - z_2 \\ \vdots & \vdots & \vdots \\ x_N - x_{N-1} & y_N - y_{N-1} & z_N - z_{N-1} \\ x_1 - x_N & y_1 - y_N & z_1 - z_N \end{bmatrix} \tag{2.16}$$

,

$$\mathbf{y} = \begin{bmatrix} (\hat{d}_1^2 - \hat{d}_2^2) - (x_1^2 - x_2^2) - (y_1^2 - y_2^2) - (z_1^2 - z_2^2) \\ (\hat{d}_2^2 - \hat{d}_3^2) - (x_2^2 - x_3^2) - (y_2^2 - y_3^2) - (z_2^2 - z_3^2) \\ \vdots \\ (\hat{d}_{N-1}^2 - \hat{d}_N^2) - (x_{N-1}^2 - x_N^2) - (y_{N-1}^2 - y_N^2) - (z_{N-1}^2 - z_N^2) \\ (\hat{d}_N^2 - \hat{d}_1^2) - (x_N^2 - x_1^2) - (y_N^2 - y_1^2) - (z_N^2 - z_1^2) \end{bmatrix}, \tag{2.17}$$

and

$$\mathbf{L}(k) = \begin{bmatrix} x(k) & y(k) & z(k) \end{bmatrix}^T \tag{2.18}$$

By solving (2.15), the position measurement of the mobile node in (2.2) is

$$\hat{\mathbf{L}}(k) = \begin{bmatrix} \hat{x}(k) & \hat{y}(k) & \hat{z}(k) \end{bmatrix}^T \tag{2.19}$$

$$= (\mathbf{A^T A})^{-1} \mathbf{A^T y} \tag{2.20}$$

Note that one of static nodes must be placed at a different altitude in order to ensure the matrix A to have full rank. This position information will be used as the measurement of an autonomous flight system in the following chapter.

3. AUTONOMOUS FLIGHT SYSTEM DEVELOPMENT

3.1 Overview

An autonomous UAV must react to any situation that may arise during a mission without human intervention. The autonomous system requirements will vary depending on the level of autonomy and the purpose of a mission. In order to attain a high degree of autonomy, UAV systems need several algorithms such as communications, path planning, trajectory generation, trajectory regulation, task allocation, cooperative tactics, and data fusion. Above all, an autonomous guidance, navigation, and a control system encompasses many of these tasks. For example, suppose a user commands the UAV to reach a desired location as shown in Figure 3.1. The vehicle starts communication with the Ground Control Station (GCS) to update the task. Next, the UAV receives its current location from its sensors, and the navigation system in the UAV generates a shortest but collision-free path based on the trajectory planning and collision avoidance algorithms. For the UAV to follow the generated trajectory, the control system sends necessary information to subsystems to change the current state such as a position, attitude, altitude, and velocity. Through these steps, the UAV eventually arrives at the desired location. In this chapter, a vehicle model of the UAV will be described, along with a model of the RSS/IMU measurements. The goal of this chapter is to develop the state estimator of the UAV using the combination of the position in (2.19) and the attitude from the IMU. Also, a control system using PD controllers for autonomous UAV flight will be defined.

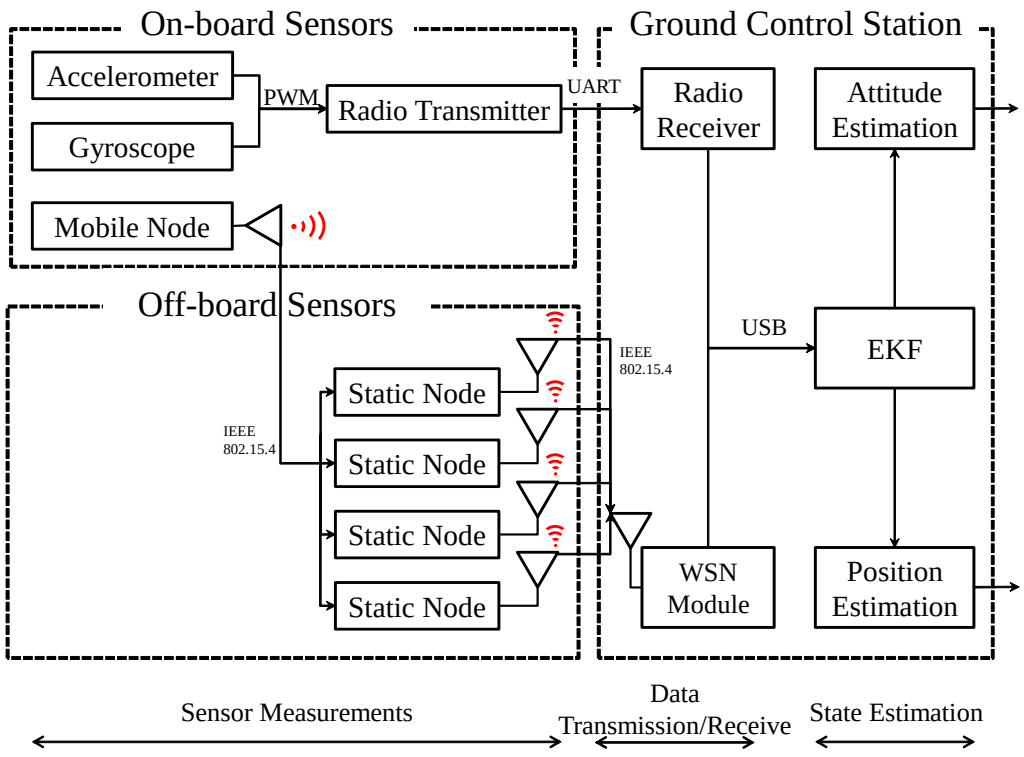
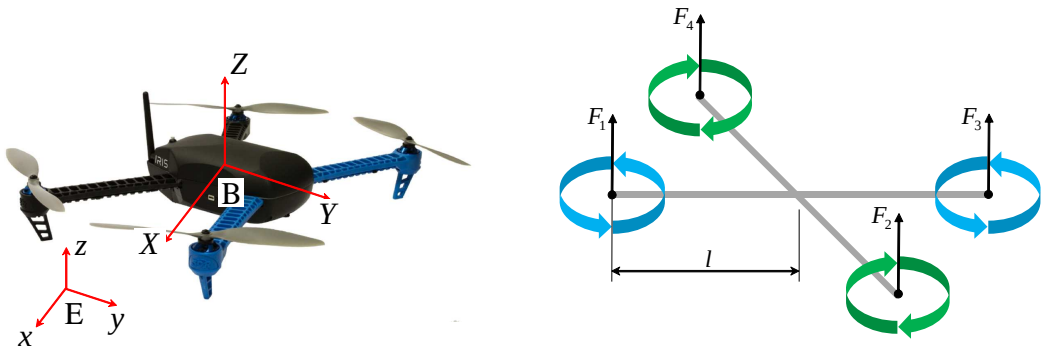


Figure 3.1. Architecture of the Proposed RSS-based Indoor Navigation System

3.2 Vehicle Model

The autonomous flight system model used in this chapter is developed based on 3DRobotics IRIS quadcopter [23], as shown in Figure 3.2(a). This quadcopter features a symmetric configuration with four Brushless DC motors and two sets of identical fixed pitched propellers which rotate clockwise and counter-clockwise as shown in Figure 3.2(b). Even the following quadcopter model is designed mainly based on the IRIS quadcopter specifically, it also follows the model which widely used for representing a quadcopter in an indoor environment as derived in [24] and [25].



(a) 3DRobotics IRIS Quadcopter (b) Configuration of Quadcopter

Figure 3.2. Vehicle Model

Reference Frame

It is necessary to define two different frames of references for depicting the dynamics and control of the quadcopter. First, on-board sensors such as the gyroscope and the accelerometer measure current state information in the body reference frame with the origin that located at the center of gravity of the quadcopter, which is denoted by $B \in \mathbb{R}^3$. Second, the RSS-based navigation system in wireless sensor networks acquires the current position of the UAV in the inertial reference frame with the origin located on the ground, which is denoted by $E \in \mathbb{R}^3$. The relationship between each frame of references is illustrated in Figure 3.2(a).

State Variables

For describing dynamics and kinematics of the quadcopter in mathematical way, state variables are established. The movement of quadcopter in three dimensional

space identified by 12 state variables is denoted as a state vector $\mathbf{x}$. State variables are defined as:

$$\mathbf{x} := \begin{bmatrix} \text{Position} & \vdots & \text{Euler angle} & \vdots & \text{Velocity} & \vdots & \text{Angular velocity} \end{bmatrix}^T \tag{3.1}$$

$$= \begin{bmatrix} \mathbf{L}_{1\times3} & \vdots & \mathbf{\Phi}_{1\times3} & \vdots & \mathbf{V}_{1\times3} & \vdots & \mathbf{\Gamma}_{1\times3} \end{bmatrix}^T \tag{3.2}$$

$$= \begin{bmatrix} x & y & z & \phi & \theta & \psi & u & v & w & p & q & r \end{bmatrix}^{\mathrm{T}} \tag{3.3}$$

Note that the position of the quadcopter is given in the inertial reference frame, the velocity and the angular velocity vector are given respect to the body reference frame, and the Euler angle vector is represented in the transformation of the body reference frame.

Rotation Matrix

The position vector $\mathbf{L} = [x, y, z]^{\mathrm{T}}$ and the velocity vector $\mathbf{V} = [u, v, w]^{\mathrm{T}}$ are defined in the different coordinate system. In order to represent the relationship between each reference frame, the following rotation matrix is defined:

$$R_1 = R_\phi R_\theta R_\psi \tag{3.4}$$

where

$$R_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \tag{3.5}$$

$$R_\theta = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \tag{3.6}$$

$$R_\psi = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.7}$$

Then,

$$R_1 = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix} \tag{3.8}$$

In a similar way, rotation matrix R_2 represents the relationship between the angular velocity vector $\mathbf{\Gamma} = [p, q, r]^{\mathrm{T}}$ and the Euler angle vector $\mathbf{\Phi} = [\phi, \theta, \psi]^{\mathrm{T}}$. This rotation matrix is given by:

$$R_2 = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\theta \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \tag{3.9}$$

Dynamics

The full dynamics of the quadcopter is given by:

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} &= \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix}^{-1} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \\ \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} &= \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\theta \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \end{aligned} \tag{3.10}$$

and

$$\begin{aligned} \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} &= \begin{bmatrix} \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ \cos\phi\cos\psi \end{bmatrix} \frac{U_1}{m} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \\ \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} &= \begin{bmatrix} (I_y - I_z)\dot{\theta}\dot{\psi} - J_r\dot{\theta}\Omega + lU_2 \\ (I_z - I_x)\dot{\phi}\dot{\psi} + J_r\dot{\phi}\Omega + lU_3 \\ (I_x - I_y)\dot{\phi}\dot{\theta} + lU_4 \end{bmatrix} \begin{bmatrix} \frac{1}{I_x} \\ \frac{1}{I_y} \\ \frac{1}{I_z} \end{bmatrix} \end{aligned} \tag{3.11}$$

where I_x, I_y, and I_z are body moments of inertia, J_r is the rotor moment of inertia, m is mass of the quadcopter, g is standard gravity, Ω is rotor speed, and U_1, U_2, U_3, and U_4 are input of each motor as given by:

$$\begin{aligned} U_1 &= b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ U_2 &= b(\Omega_4^2 - \Omega_2^2) \\ U_3 &= b(\Omega_3^2 - \Omega_1^2) \\ U_4 &= d(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \\ \Omega &= \Omega_2 + \Omega_4 - \Omega_1 - \Omega_3 \end{aligned} \tag{3.12}$$

3.3 State Estimation

In Section 2.2, the proposed indoor navigation system is developed to obtain the position of the mobile node in real-time instead of using the GPS measurement. With this position information, the UAV also requires knowledge of its attitude $\mathbf{\Phi} = [\phi\ \theta\ \psi]^{\mathrm{T}}$ and $\mathbf{V} = [u\ v\ w]^{\mathrm{T}}$ for autonomous UAV flight. Therefore, this section proposes a state estimation algorithm using both RSS measurement and IMU measurement. The overall data flow is described in Figure 3.3. This state information can significantly affect vehicle's control performance. The main challenge of state estimation is to obtain precise vehicle's state information, given sensor measurements with combination of nonlinearity and uncertainty. In the proposed indoor navigation system, the IMU which consists of three-axis accelerometers and three-axis gyroscopes will be used for estimating quadcopter's current attitude. The RSS for estimating quadrator's current position is also given by (2.19). Then, both measurements are combined via Extended Kalman Filter (EKF) to estimate the state of the UAV in real-time. The combination of these two kinds of sensors can be exploited to improve the robustness and performance of the system. . In addition to vehicle's position and attitude, state variables for representing characteristic of sensors are required. Thus, the state vector is:

$$\mathbf{x} = \begin{bmatrix} x & y & z & \phi & \theta & \psi & u & v & w & \mathbf{b}_{accel} & \mathbf{b}_{gyro} \end{bmatrix}^{\mathrm{T}} \tag{3.13}$$

where $\mathbf{b}_{accel}$ and $\mathbf{b}_{gyro}$ are gyroscope and accelerometer bias vector, respectively.

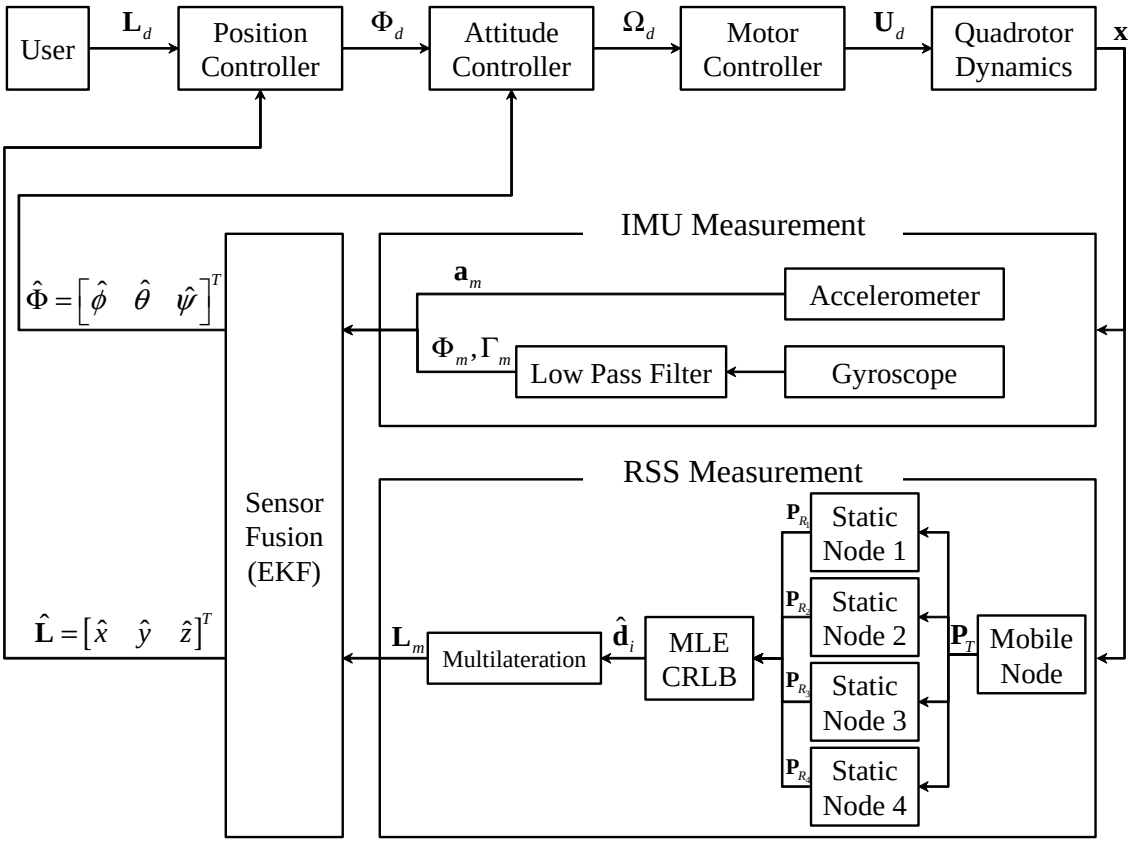


Figure 3.3. States Flow of the Proposed Autonomous System

where $\mathbf{L}_d, \mathbf{\Phi}_d, \mathbf{\Omega}_d$, *and* $\mathbf{U}_d$ *are the desired position, the desired attitude, the desired motor speed, and the desired input of the UAV, respectively. The input of the entire system is a user's command of the desired state of the UAV. The output* $\mathbf{x}$ *is the full state vector of the UAV. Note that the subscript m represents measured states and hat represents estimated states.*

3.3.1 Sensor Fusion via Extended Kalman Filter

In this analysis, the EKF follows two processing steps: the step of prediction with respect to the system model and the step of update with respect to the measurement models. The EKF recursively repeats each step in its frequency. In order to prevent the EKF from diverging, the process noise, $\mathbf{v}_k$ is assumed to be uncorrelated with the

state and the state estimate. The well-known Extended Kalman Filter is presented in Algorithm 1 by combining these two steps. The system model is given by:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \tag{3.14}$$

and measurement models are defined as:

$$\mathbf{y}_{\text{IMU}} = h_1(\mathbf{x}) \tag{3.15}$$

$$\mathbf{y}_{\text{RSS}} = h_2(\mathbf{x}) \tag{3.16}$$

Note that the IMU measures acceleration and angular rate in the body reference frame are denoted as B while the proposed navigation system and RSS measurement are represented in the inertial reference frame which are denoted as E. A relation between each frame and rotation matrices can be found in Section 3.2.

3.3.2 System Model

To implement EKF, the system model derived in Section 3.2 is used. Since the state information which provided by IMU and RSS are coupled with each other, the system model (also known as the process model) needs to be described with the full state of a vehicle. The system model is defined as:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} & \dot{\phi} & \dot{\theta} & \dot{\psi} & \dot{u} & \dot{v} & \dot{w} & \dot{\mathbf{b}}_{accel} & \dot{\mathbf{b}}_{gyro} \end{bmatrix}^{\text{T}} \tag{3.17}$$

The analytical derivation of the system model is provided in (A.3).

3.3.3 IMU Measurement Model

The IMU measurement model is designed based on typical off-the-shelf strapdown IMU which consists of ST LSM303D accelerometer [26] and ST L3GD20 gyroscope [27]. The IMU measurement model used in state estimation can be written as:

$$\mathbf{y}_{\text{IMU}} = \begin{bmatrix} \mathbf{y}_{\text{accel}} \\ \mathbf{y}_{\text{gyro}} \end{bmatrix} \tag{3.18}$$

The measurement of IMU includes the terms of bias, actual value, and noise. The accelerometer measurement model in the body reference frame can be defined as:

$$\mathbf{y}_{\text{accel}} = \begin{bmatrix} rv - qw \\ pw - ru \\ qu - pv \end{bmatrix} - g \begin{bmatrix} -\sin\theta \\ \cos\theta \sin\phi \\ \cos\theta \cos\phi \end{bmatrix} + \mathbf{b}_{\text{accel}} + \mathbf{w}_{\text{accel}} \tag{3.19}$$

where $\mathbf{b}_{\text{accel}}$ is the sensor bias and $\mathbf{w}_{\text{accel}}$ represents a Gaussian random noise with zero mean. The gyroscope measurement model represents in the body reference frame can be defined as:

$$\mathbf{y}_{\text{gyro}} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \mathbf{b}_{\text{gyro}} + \mathbf{w}_{\text{gyro}} \tag{3.20}$$

where the angular velocity $\mathbf{\Gamma}$ represents the components $[p\ q\ r]^{\text{T}}$ in the body reference frame and $\mathbf{b}_{\text{gyro}}$ is the sensor bias, and $\mathbf{w}_{\text{gyro}}$ represents the Gaussian random noise with zero mean.

3.3.4 RSS Measurement Model

The measurement model for RSS is also presented with measurement noise, and the postion given by (2.19). The RSS measurement model in the inertial reference frame is given by:

$$\mathbf{y}_{\text{RSS}} = \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} + \mathbf{w}_{\text{RSS}} \tag{3.21}$$

Algorithm 1 Algorithm of the Extended Kalman Filter (EKF)

The Extended Kalman Fliter of nonlinear dynamics system for discrete-time is given by:

$$\dot{\mathbf{x}}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{v}_k, \qquad \mathbf{v}_k \sim \mathcal{N}(0, Q_k) \tag{3.22}$$

$$\mathbf{y}_k = h(\mathbf{x}_k) + \mathbf{w}_k, \qquad \mathbf{w}_k \sim \mathcal{N}(0, R_k) \tag{3.23}$$

1. *Prediction (or Propagation)*

   State estimate:

$$\hat{\mathbf{x}}_k(-) = f\Big(\hat{\mathbf{x}}_{k-1}(+), \mathbf{u}_{k-1}\Big) \tag{3.24}$$

   Covariance:

$$\mathbf{P}_k(-) = \mathbf{F}_{k-1}\mathbf{P}_{k-1}(+)\mathbf{F}_{k-1}^{\mathrm{T}} + \mathbf{Q}_{k-1} \tag{3.25}$$

2. *Update (or Correction)*

   Kalman gain:

$$\mathbf{K}_k = \mathbf{P}_k(-)\mathbf{H}_k^{\mathrm{T}}\Big(\mathbf{H}_k\mathbf{P}_k(-)\mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k\Big)^{-1} \tag{3.26}$$

   State estimate:

$$\hat{\mathbf{x}}_k(+) = \hat{\mathbf{x}}_k(-) + \mathbf{K}_k\Big(\mathbf{y}_k - h(\hat{\mathbf{x}}_k(-))\Big) \tag{3.27}$$

   Covariance:

$$\mathbf{P}_k(+) = \Big(\mathbf{I} - \mathbf{K}_k\mathbf{H}\Big)\mathbf{P}_k(-) \tag{3.28}$$

Note that a variable with $(-)$ represents the estimated state before an update, a variable with $(-)$ represents the estimated state after an update. Jacobian matrices are given by:

$$\mathbf{F_k} = \frac{\partial f}{\partial \mathbf{x}} \tag{3.29}$$

$$\mathbf{H_k} = \frac{\partial h}{\partial \mathbf{x}} \tag{3.30}$$

More detail in the derivation of each step for Extended Kalman Filter is given in Appendix A.

3.4 Control

The goal of the control system is to minimize the error between the desired state given by the user and the estimated state from the state estimator. In order to achieve this goal, we implement a widely-used PID-type nested control loop stabilization [28]. For our system, the outputs of the controller are the three torques in the Euler angles and the thrust. They are the inputs to the motor. The inputs of the controller are the estimated states and the planned trajectory which is defined by the desired position $[x_d \ y_d \ z_d]^T$ and heading ψ_d.

The xy-position controller is shown in Figure 3.4. The inner loop controls the roll and pitch attitude of the quadrotor using a PD controller

$$\begin{bmatrix} \tau_\phi \\ \tau_\theta \end{bmatrix} = P_{\phi\theta} \left\{ \left(\begin{bmatrix} \phi_d \\ \theta_d \end{bmatrix} - \begin{bmatrix} \hat{\phi} \\ \hat{\theta} \end{bmatrix} \right) + D_{\phi\theta} \left(\begin{bmatrix} p_d \\ q_d \end{bmatrix} - \begin{bmatrix} \hat{p} \\ \hat{q} \end{bmatrix} \right) \right\} \tag{3.31}$$

where ϕ_d and θ_d are the desired roll and pitch angles given by the outer loop. The desired roll rate p_d and pitch rate q_d are each zero. The outer loop which controls the xy-position generates the desired roll and pitch angles.

$$\begin{bmatrix} \phi_d \\ \theta_d \end{bmatrix} = P_{xy} \left\{ \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix}^{-1} \left(\begin{bmatrix} x_d \\ y_d \end{bmatrix} - \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} \right) + D_{xy} \left(\begin{bmatrix} u_d \\ v_d \end{bmatrix} - \begin{bmatrix} \hat{u} \\ \hat{v} \end{bmatrix} \right) \right\} \tag{3.32}$$

where x_d and y_d are the desired xy-position and u_d and v_d are the desired velocity components which are both zero. Note that since the xy-position is in the inertial frame and the velocity components u and v are in the body frame, the position is mapped to the body frame by a rotation matrix.

As shown in Figure 3.5, heading is controlled by a PD controller

$$\tau_\psi = P_\psi \Big[(\psi_d - \psi) + D_\psi (r_d - r) \Big] \tag{3.33}$$

where ψ_d is the desired yaw angle and r_d is the desired yaw rate, which is zero.

Figure 3.6 shows the altitude controller

$$T = P_z \Big[(z_d - z) + D_z (w_d - w) \Big] + \omega_0 \tag{3.34}$$

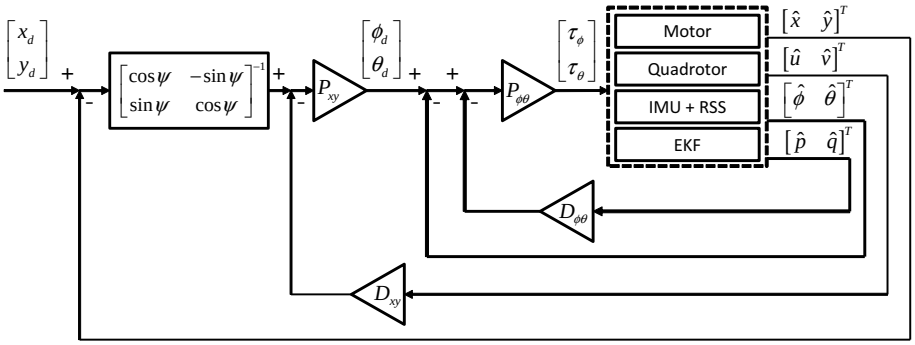


Figure 3.4. xy-position Controller

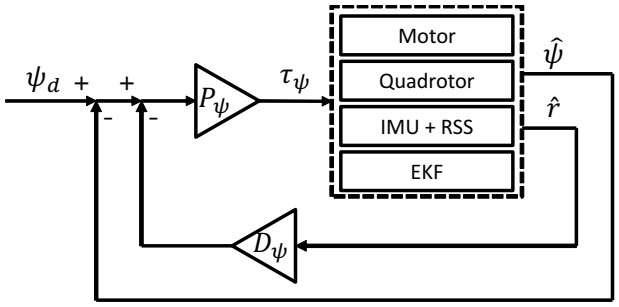


Figure 3.5. Heading Controller

where z^* is the desired position on the z-axis. Note that unlike other controllers, the altitude controller has a feedforward control ω_0 to reject a constant disturbance due to the gravity force on the vehicle [28].

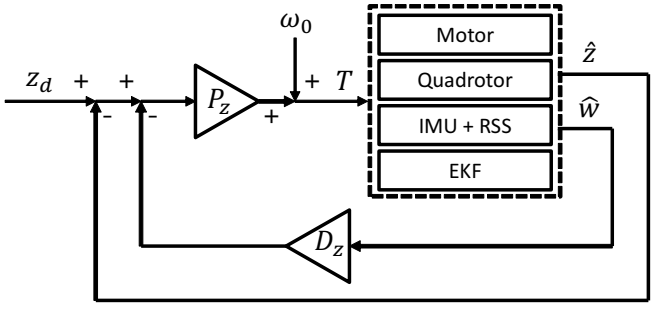


Figure 3.6. Altitude Controller

Since we use a nonlinear model of the quadrotor and there are eight gains in the control loop, a classical approach for gain tuning is not appropriate for our system. Therefore, in this paper, an optimization approach is employed to determine the values of the gains. Let the gains be the design variables:

$$\mathbf{x} = \begin{bmatrix} P_{xy} & D_{xy} & P_{\phi\theta} & D_{\phi\theta} & P_{\psi} & D_{\psi} & P_z & D_z \end{bmatrix}^T \tag{3.35}$$

These variables are constrained by inequality constraints with lower bound $\mathbf{x}_L$ and upper bound $\mathbf{x}_U$. We want to minimize the sum of the errors between the desired state and the estimated state during the entire flight. The optimization problem is then defined as

$$\min J = \int_{t_0}^{t_f} \left[\Big(x_d(t) - \hat{x}(t)\Big)^2 + \Big(y_d(t) - \hat{y}(t)\Big)^2 + \Big(z_d(t) - \hat{z}(t)\Big)^2 + \Big(\psi_d(t) - \hat{\psi}(t)\Big)^2 \right] dt$$
$$\text{subject to } \mathbf{x}_L \leq \mathbf{x} \leq \mathbf{x}_U \tag{3.36}$$

In order to solve the optimization problem, we use the coevolutionary augmented Lagrangian method [29] which can efficiently and effectively solve constrained optimization problems.

4. SIMULATION

4.1 Overview

This research has developed the algorithms which obtain the best estimate of position in real-time. It has also designed the autonomous flight system for UAVs using these position estimates. In this chapter, the results of a numerical simulation for the proposed system using Matlab or Simulink are presented for demonstrating its performance and feasibility prior to the experiment in Chapter 5. In order to do this, the proposed autonomous UAV flight system in Chapter 3 is simulated. The goal of the simulation is to evaluate the validity of the integrated system including the controllers, an indoor navigation system, and the estimator as shown in Figure 4.1. Simulation of finding an optimal gain for each controller is presented. For providing more practical situations, two simulations are performed: a simulation of following the desired trajectory in an indoor space and a simulation of following the desired waypoints in a hallway of the building.

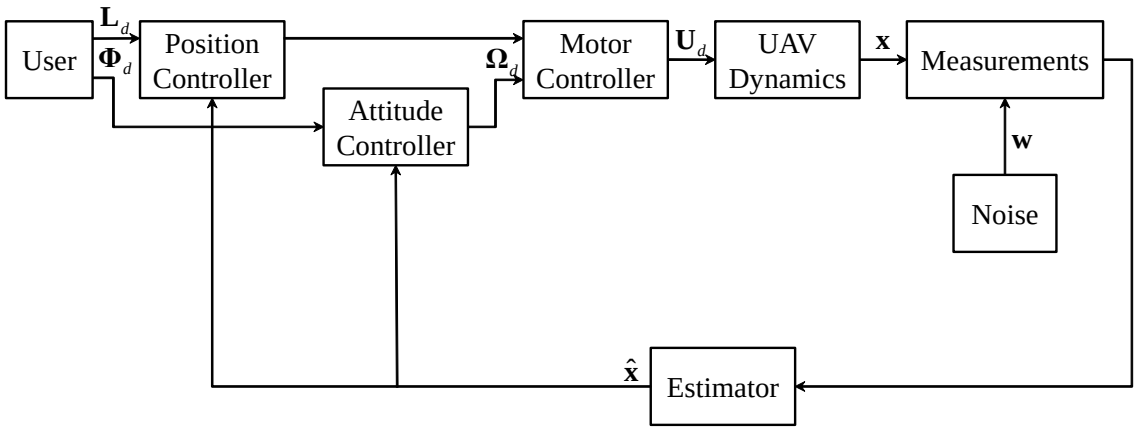


Figure 4.1. Simulation Block Diagram

4.2 Simulation of the Autonomous UAV Flight System

4.2.1 Gain Tuning

As discussed in Section 3.4, this research formulates the optimization problem which minimizes the difference between the desired states and measured states. In order to solve this optimization problem, a solver is developed and the simulation model including the controllers, navigation system, and estimator is designed in Simulink. The actual simulation model can be found in Appendix B. The desired states and measured states are stored at each sampling time from an input and from an output, respectively. The solver begins to find the optimal gain by changing the current gain iteratively from the lower bound to the upper bound. Then the solver calculates the sum of square difference between each state until it finds the optimal gain of each decision variable which yields the minimum sum of square difference. The cost function which is the sum of square difference decreases as a number of iteration increases as shown in Figure 4.2. In this simulation, the optimal gain for each controller is given by 125 iterations. The numerical value of the each optimal gain which minimizes the cost function can be found in Appendix B.

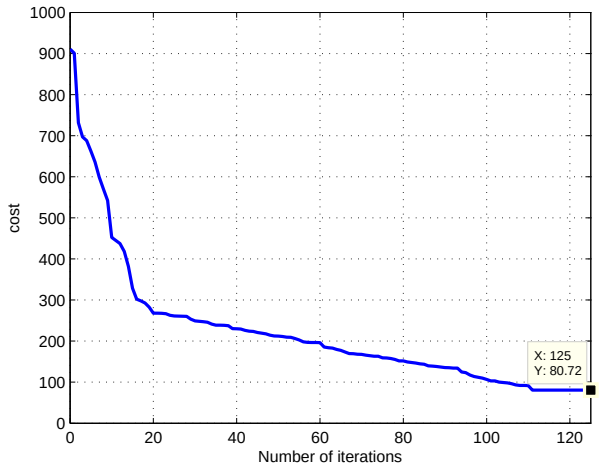


Figure 4.2. Gain Optimization

4.2.2 Scenario 1: Circular Trajectory

To make the proposed autonomous UAV flight system be more practical, the current simulation considers two different scenarios. The goal of the first scenario is to make a UAV to follow the circular trajectory, using the proposed system. The circular trajectory contains all possible movements of the UAV such as takeoff, landing, and loitering, so it is useful to observe the performance of control and estimation. In this scenario, Monte Carlo simulation runs for $n = 100$ and gains of controllers from previous section are used. Also, RSS measurement is updated in every 0.1 second. Simulation results in this scenario are built by using the mean of each state at each sampling time during 100 runs, denoted as *c_input*, *state_True*, *state_Mea*, and *state_Est* in Figure B.1.

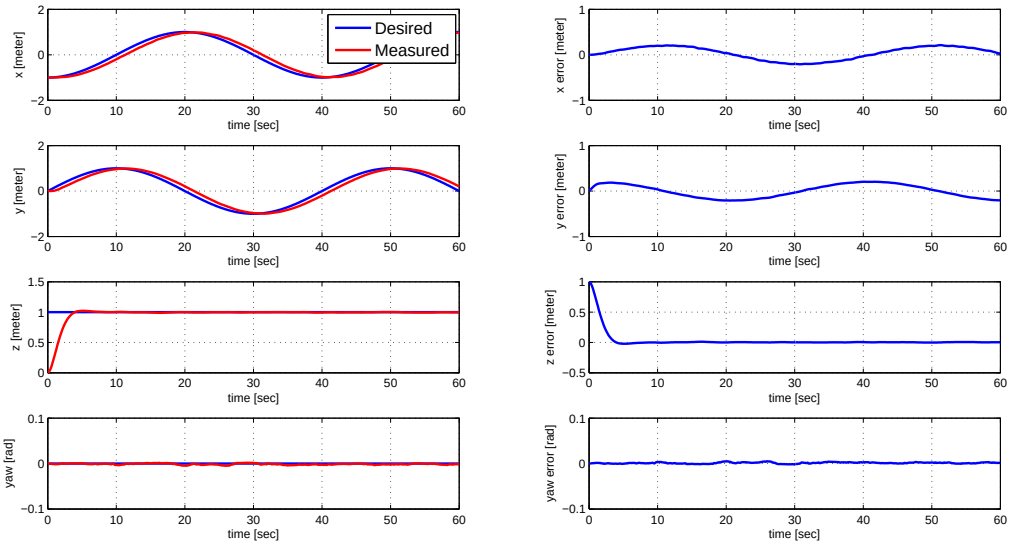


Figure 4.3. Scenario 1: UAV Control

Figure 4.3 shows the control performance by comparing the desired states to the measured states such as position and yaw. As can be seen in Figure 4.3, the measured state follows the desired state successfully. It shows that the solver has succeed to find the optimal gains. Especially, the control of the altitude, which is denoted as z,

shows a slight overshoot even though it has a different initial condition unlike other states which have zero initial condition. A maximum error of each state during the simulation stays within 20 centimeters as shown in the right side of Figure 4.3.

The simulation results demonstrating the performance of state estimation are shown in Figure 4.4, 4.5(a), and 4.5(b). Figure 4.4 shows the estimation of each state such as position and velocity by drawing the true state, a set of measurement in a run of 100 runs, estimated state, and $\pm 3\sigma$. Note that 3 standard deviations bound in the solid black line, which cover 99.7 percentages of the data, is used to show a boundary of estimation error. This boundary is calculated based on Monte Carlo simulation. A partial part of v_y in Figure 4.4 is shown in Figure 4.5(a). While some of raw measurements are out of the boundary, the estimated v_y keeps following the true state without significant errors. Therefore, simulation of scenario 1 demonstrates the feasibility of the proposed autonomous flight system and it is supported by the result figures.

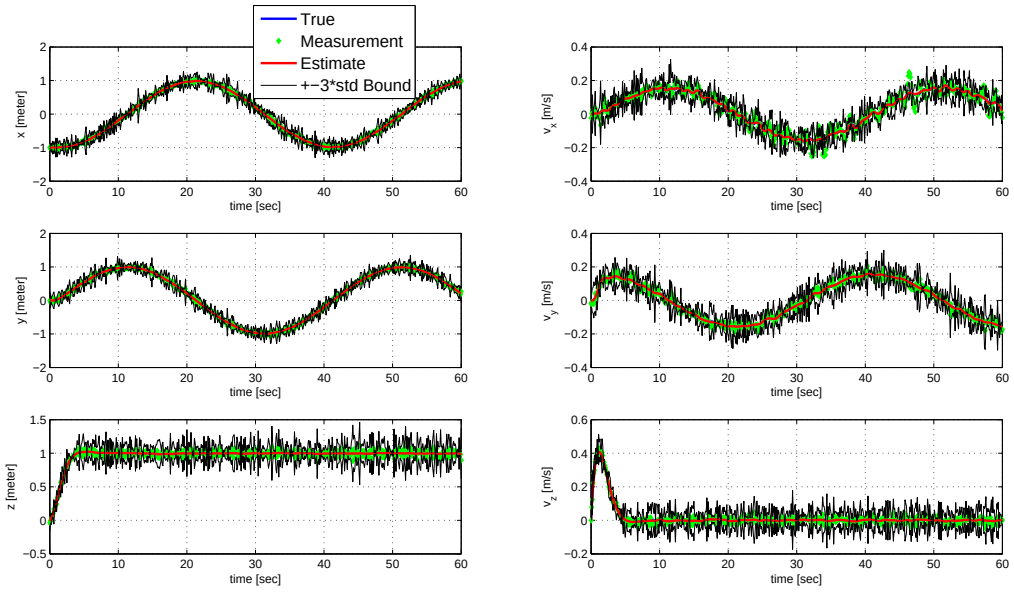
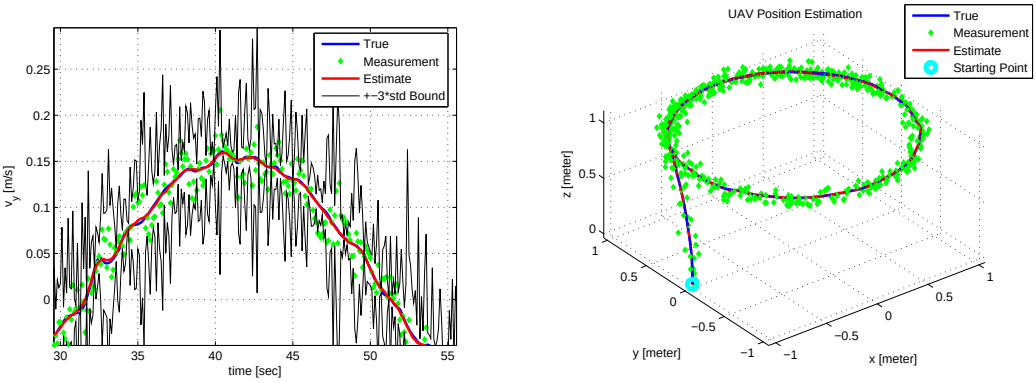


Figure 4.4. Scenario 1: UAV State Estimation



(a) UAV v_x Estimation

(b) UAV Position Estimation

Figure 4.5. Scenario 1: State Estimaition

4.2.3 Scenario 2: Hallway

Scenario 1 showed that the UAV may fly in the space where 4 static nodes are installed. However, there is a certain limitation on the range of flying due to the availability of finite number of static sensors in real world. For example, if an UAV follows waypoints through the hallway, the static node has to be installed at least at every 4 meters to provide sufficient position data stream. Therefore, more realistic situation is provided in this scenario to show the flexibility and adaptability of the proposed autonomous flight system.

Suppose that an UAV follows the desired waypoints as marked in the blue solid line in Figure 4.6, which includes the perpendicular, round corner, and the straight hallway. Assume that the UAV flies the third floor of Neil Armstrong Hall of Engineering in Purdue University and static nodes are set at every 4 meters on the walls facing each other throughout the hallway. General setup for the simulation is same as the previous scenario and simulation results are built by using the states denoted as *state_True* and *state_Est* in Figure B.1. The performance of state estimation is summarized in Table 4.1. Note that this scenario is designed for a single run while scenario 1 is designed for 100 runs. Some of the maximum estimation errors are over

30 centimeters at the corner where RSS measurement noise is higher than other location due to the effects of multipath and fading. However, it is fairly acceptable since the Root Mean Square (RMS) error of each state represents the better estimation performance.

Table 4.1. Scenario 2: RSS-based State Estimation Error

	$x[m]$	$y[m]$	$z[m]$	$v_x[m/s]$	$v_y[m/s]$	$v_z[m/s]$
RMS error	0.0906	0.0808	0.0581	0.0367	0.0359	0.0369
Max error	0.2264	0.3010	0.2308	0.1362	0.1255	0.1247

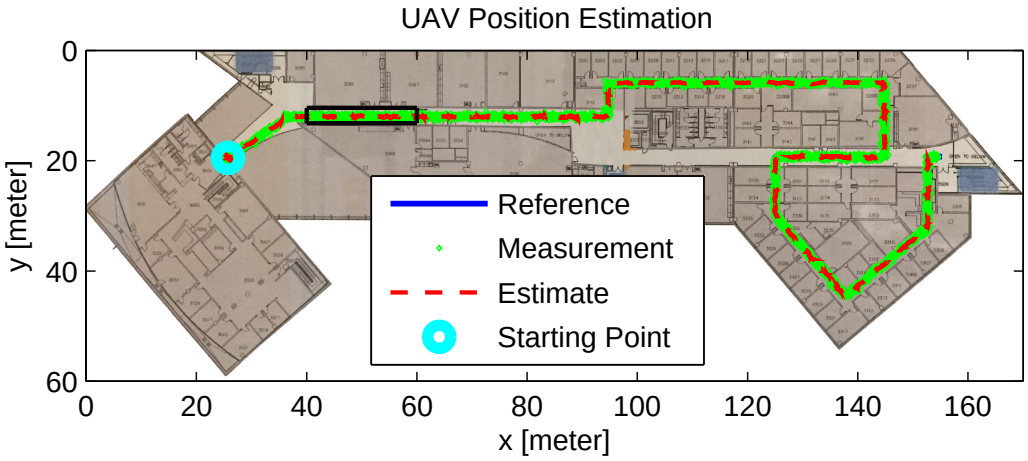


Figure 4.6. Scenario 2: UAV Position Estimation

Figure 4.7 is a close-up of the black box in Figure 4.6. The difference between the position estimate and the true position is not too much critical so that it is feasible to operate the UAV in an indoor environment. In Figure 4.8, a propagation map of RSS measurement of the black box in Figure 4.6 is shown with colors. The darker red color indicates the higher possibility of the UAV. This map is built by combining the measurements of the RSS from a set of 4 static nodes and the measurements of IMU from the UAV as developed in Chapter 3. It is clear that the dark red area and

the true position are nearly overlapped with each other. By illustrating the results of simulations, this research validates the feasibility of the proposed autonomous flight system including the proposed navigation system, controllers, and estimator. Based on this, a real flight test using a quadcopter is demonstrated in the following chapter.

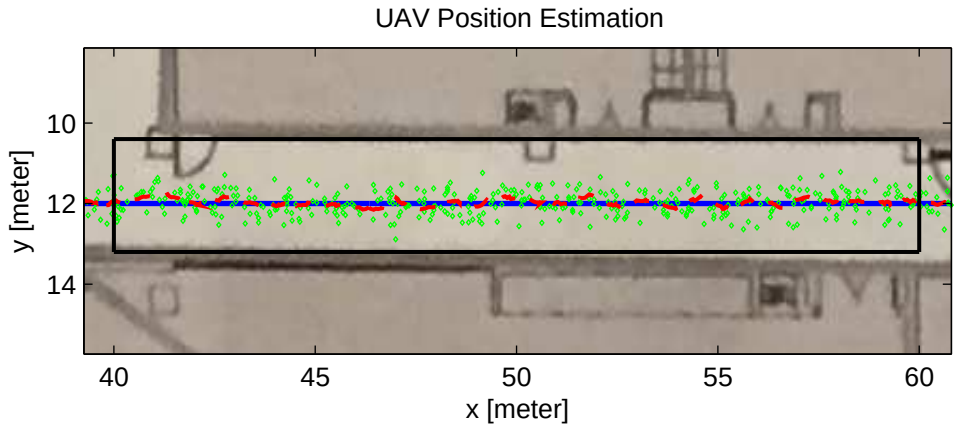


Figure 4.7. Scenario 2: UAV Position Estimation(Scale-up)

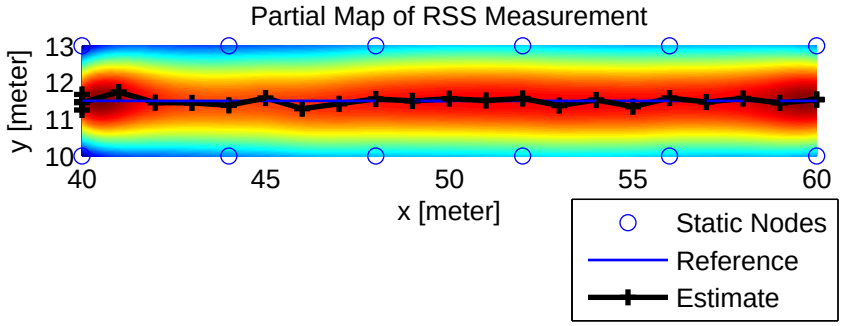


Figure 4.8. Scenario 2: Partial Propagation Map of RSS Measurement

5. EXPERIMENT

5.1 Overview

This chapter presents the validation of the proposed indoor navigation system by real flight tests based on algorithms derived in Chapters 2 and 3. The ultimate goal of this chapter is to guide an UAV to reach the desired position using the RSS-based system in real-time. In order to focus on the validation of the proposed indoor navigation system, it is assumed that there are no obstacles and walls on the vehicle's path causing a collision. In this chapter, general information about the experiment testbed, including software and hardware, is described in Table 5.1. Three different trajectories for real flight tests using an UAV and their results are also presented. Finally, this chapter evaluates reliability by comparing the real flight log with the simulation. This experiment demonstrates that the proposed indoor navigation system is feasible for guidance, navigation, and control of UAVs.

5.2 Experiment Testbed

5.2.1 Software

Most softwares are open source so that it is convenient to modify and customize them. For example, the Robot Operating System (ROS) [30] is a bunch of software libraries and toolbox to help users build robot applications. The ROS is widely used in robotics. In this experiment, the ROS runs on a companion board such as Odroid U3 to connect TinyOS and PX4. This is because the autopilot requires measurements from offboard sensors such as WSN modules as well as measurements from onboard sensors such as IMU for real-time control. More details about each software can be found in its official webpages.

Table 5.1. Software and Hardware Used in Experiment

Software	**Description**
Ubuntu* & Windows	Operating systems
TinyOS*	Operating system for wireless sensor networks
ROS*	Robot Operating System
QGroundControl*	Software for a ground control station
PX4*	Firmware for Pixhawk
OpenCV*	Software libraries for computer image processing
Hardware	**Description**
IRIS	Quadcopter platform
Pixhawk	Autopilot module
WSNs Modules	The mobile and static nodes
Kinect	Vision sensor
Laptop	Ground Control Station
Odroid U3	Companion computer

*Note that the superscript * represents open source software, and software is matched up with corresponding hardware such as TinyOS-WSNs modules, ROS-Odroid U3, QGroundControl-Laptop, PX4-Pixhawk, and OpenCV-Kinect.*

TinyOS

TinyOS [31] is an open source Linux-based operating system for low-power wireless sensor networks modules. TinyOS is an embedded operating system written in the Network Embedded Systems C (nesC) [32] programming language. TinyOS provides huge task libraries for various modules such as the mobile node and static nodes in this experiment. In this experiment, three commands are used to collect data packets including the RSS and transmitted signal strength as shown in Algorithm 2. These simple, low-complexity commands enable WSN modules to obtain data packets.

Algorithm 2 Functions in TinyOS

function **getPower**

command uint8_t **getPower**(message_t *p_msg)

% Get transmission power setting for current packet.

function **getRssi**

command int8_t **getRssi**(message_t *p_msg)

% Get rssi value for a given packet. For received packets, the value is the RSS when receiving the packet. For sent packets, the value is the RSS of an acknowledgement if the acknowledgement was received.

function **setPower**

command void **setPower**(message_t *p_msg, uint8_t power)

% Set transmission power for a given packet. The valid range is between -25 and 0 dBm.

5.2.2 Hardware

Quadcopter

Off-the-shelf and ready-to-fly IRIS quadcopter provided by 3DRobotics is used for real flight tests. The IRIS is an autonomous and fully-assembled UAV and it equips a high-performance Pixhawk autopilot hardware which include an on-board sensor IMU. More technical specifications and features can be found in [23, 33].

Wireless Sensor Networks Modules

As mentioned in Section 2.2, two different types of WSN modules are used. The first type of module, the mobile node, is an Advanticsys CM3300 [34] with an external 5 dBm SMA antenna which allows users to amplify the transmitted signal strength via TinyOS as shown in Figure 5.1(a). The second type of module, the static node, is a CM5000 [35] which collects data packets from the mobile node and then sends them to the GCS as shown in Figure 5.1(b). Both modules consist of a TI CC2420 [36] radio frequency transceiver based on the IEEE 802.15.4 standard and a MSP430F1611 [37] microcontroller unit. They are powered by two AA batteries.

(a) The Mobile Node: CM3300 (b) The Static Node: CM5000

Figure 5.1. Wireless Sensor Networks Modules

Microsoft Kinect

In this experiment, Microsoft Kinect [38] is used as reference while using an expensive tool such as a Vicon motion detecting machine. In former research [39], the error of measurement from MS Kinect has been validated to be reliable. MS Kinect has best performance within 1 to 3 meters in depth direction. The error of MS Kinect is randomly distributed from 4 centimeters at minimum to 7 centimeters at maximum. Thus, the performance of the proposed system can be validated by comparing the flight with the reference. In order to that, this research develops an object tracking algorithm for the Kinect. As shown in Figure 5.3, the Kinect provides depth information from the origin to an object which is denoted as z_k. The focal length which is denoted as f and coordinate of an object on the screen that represented as x_s and y_s are also provided. Maximum horizontal and vertical angles of view are given by technical specifications so that it is possible to calculate the position of an object which is denoted as x_k, y_k, and z_k by solving a simple geometric and rotation equation. Based on this algorithm, a software is built in the OpenCV using on `C++`. This software can automatically detect, track, and store the position of an object. The software obtains the position of an object by the following steps.

1. Search a minimum rectangular that completely covers an object on the screen.
2. Determine a center of the rectangular.
3. Determine a x and y coordinate of the center on the screen and a depth of the center and store it into a txt.
4. Import the stored data and then calculate the local position of the object in the Kinect frame.

The user interface of the developed software is shown in Figure 5.3. Therefore, the real flight trajectory from the Kinect can verify how accurate the real flight trajectory from the proposed system is by calculating the difference between them.

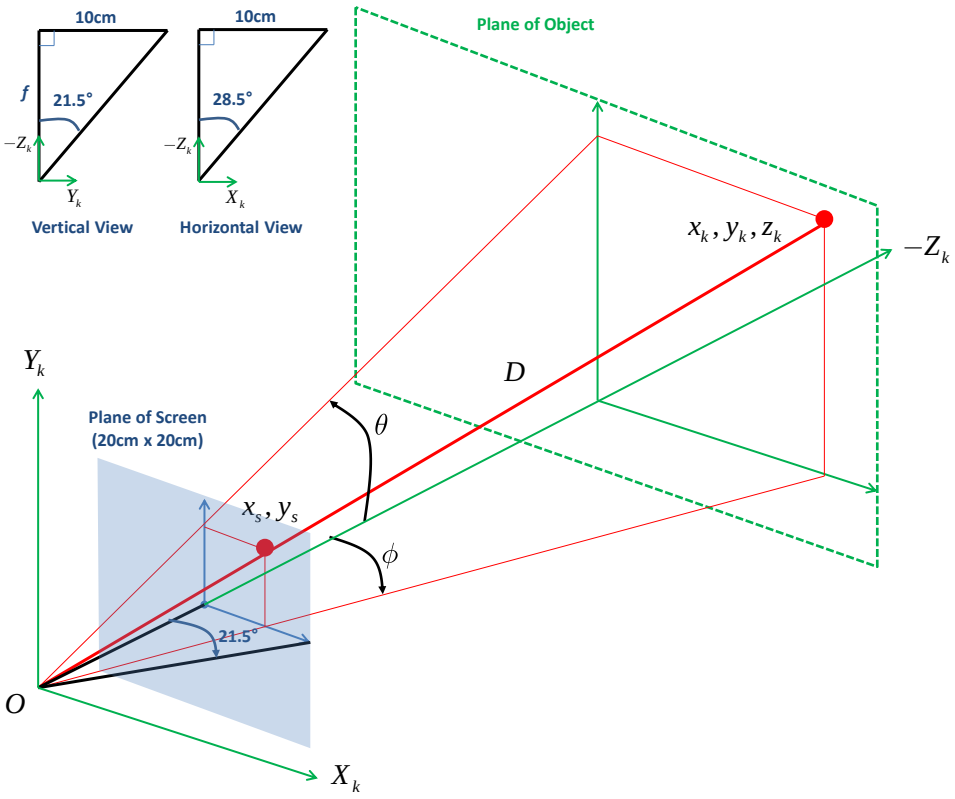


Figure 5.2. Tracking Algorithm of MS Kinect

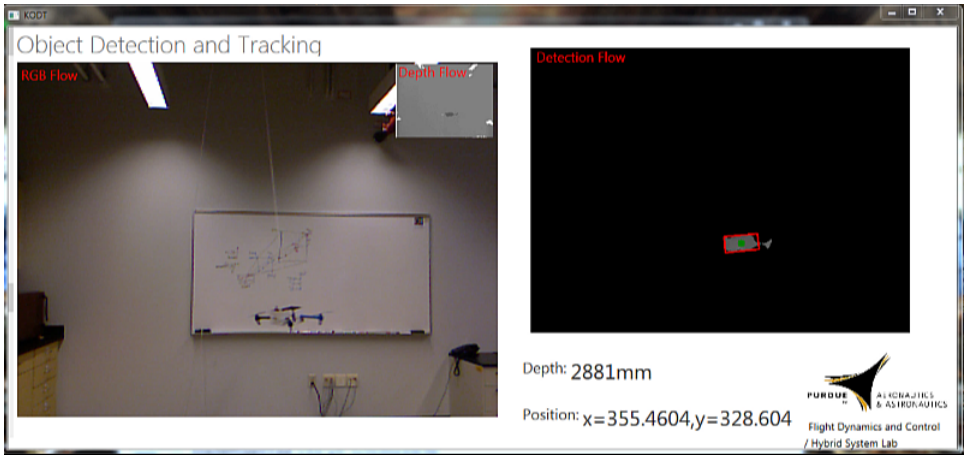


Figure 5.3. Object Detection and Tracking using MS Kinect

5.2.3 General Setup

The goal of the experiment is to show that the proposed system is reliable for navigation, guidance, and control of UAVs. The experiment is performed by the following steps. The UAV with the mobile node is placed in the area of wireless sensor networks. A user commands the UAV to reach the desired position by setting several waypoints in GCS. As derived in Sections 2.3.2 and 2.3.3, both sensor modules start to calibrate an optimal path-loss exponent and transmission power before the UAV takeoff. Based on those optimal parameters, the proposed system calculates the distance between the UAV and each static node. After that, EKF provides real-time position estimation of the UAV as a feedback until it reaches the desired position. The testbeds for the manual flight test and the autonomous flight test are represented in Figure 5.4 and an schematic overview of the experiment is shown in Figure 5.5.

(a) Testbed for the Manual Flight (b) Testbed for the Autonomous Flight

Figure 5.4. Experiment Testbeds

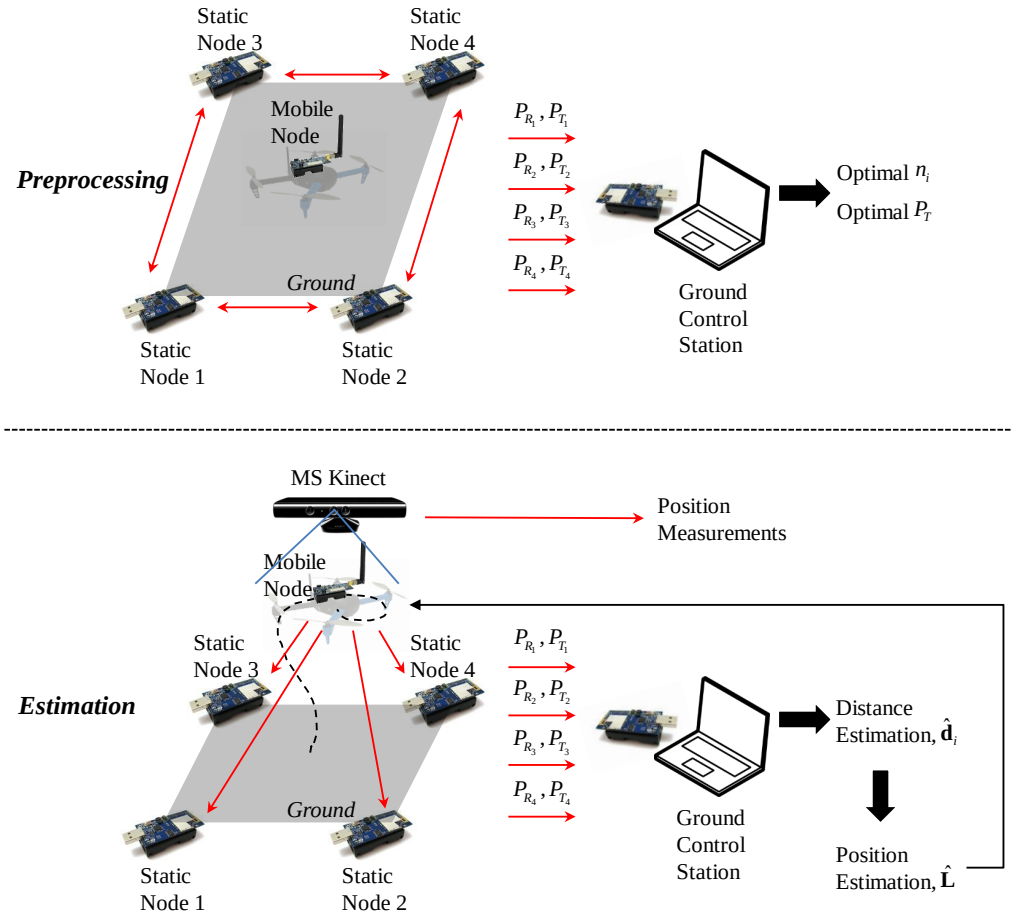


Figure 5.5. Schematic Overview of the Experiment

5.3 RSS-based State Estimation

Mission Statement

Before autonomous real flight tests, a manual real flight test is performed for verification of the estimation performance of the proposed RSS-based indoor navigation system. As discussed in Section 5.2.2, the estimation performance of the proposed indoor navigation system is validated by comparing state estimation of the proposed system to reference from the Kinect.

Result

Results of state estimation of the manual test is listed in Table 5.2. Estimation error is calculated by subtracting each state estimate from each reference state. i.e. $e = x - \hat{x}$. RMS error in the simulation is quite smaller than RMS error in the experiment. This is because the estimation error of each state in simulation is obtained by using the mean of each state of Monte Carlo simulation for 100 runs. Maximum error of each state in the experiment does not exceed 0.2 meters or meter per second.

Table 5.2. RSS-based State Estimation Error

	$x[m]$	$y[m]$	$z[m]$	$v_x[m/s]$	$v_y[m/s]$	$v_z[m/s]$
RMS error in Simulation	0.0081	0.0078	0.0078	0.0031	0.0030	0.0032
RMS error in Experiment	0.0428	0.0386	0.0385	0.0392	0.0407	0.0463
Max. error in Simulation	0.0283	0.0239	0.0256	0.0104	0.0105	0.0096
Max. error in Experiment	0.1635	0.1375	0.1073	0.1171	0.1345	0.1290

As can be seen in Figure 5.6, state estimates from raw measurements of the RSS-based system nearly coincide to the reference. While raw measurements in green dots are critically inaccurate to use for controlling the UAV, state estimation in red line is accurate enough to use as a state feedback for controlling the UAV. The difference between the trajectory of RSS-based estimation and the trajectory of reference is shown in Figure 5.7. Therefore, the proposed RSS-based system has the reliable estimation performance for controlling an UAV.

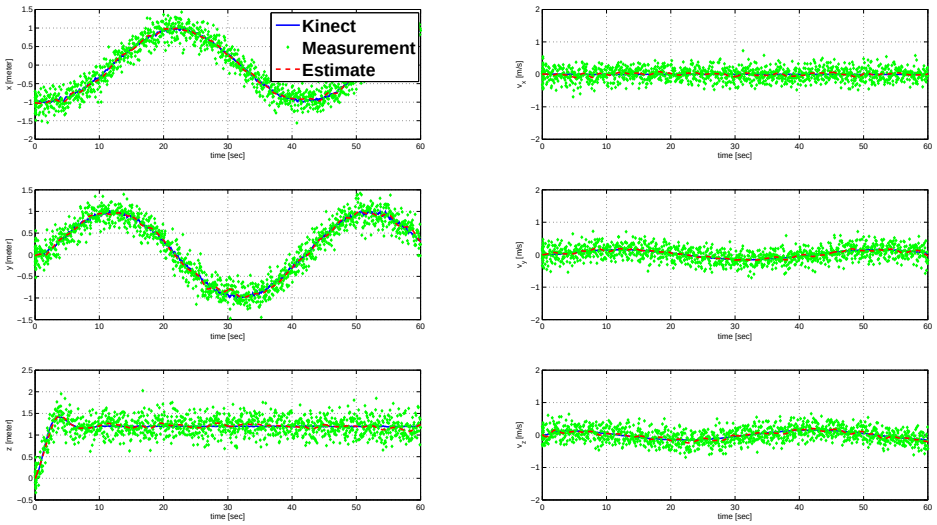


Figure 5.6. RSS-based UAV State Estimation

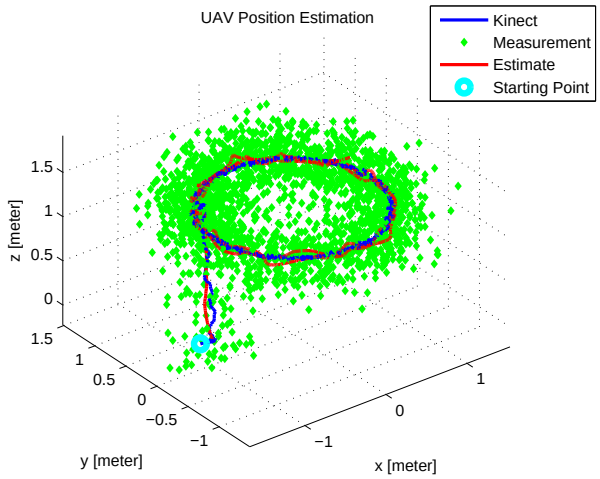


Figure 5.7. RSS-based UAV Position Estimation

5.4 Autonomous Flight Test

This research has shown the feasibility of the estimation performance of the proposed RSS-based indoor navigation system in the previous section. Now, this research demonstrates the reliability of the proposed autonomous UAV flight system based on

the proposed RSS-based indoor navigation. Testbed is shown in Figure 5.4(b) and other conditions are identical with general setup described in Section 5.2.3. The test is performed with the proposed autonomous system and the architecture of the overall system is shown in Figure 5.8. In case of an unexpected situation, the proposed autonomous flight system is built to be changed automatically from the autonomous flight mode to the manual flight mode. Once the mode is changed, the operator can retain a control of the UAV immediately.

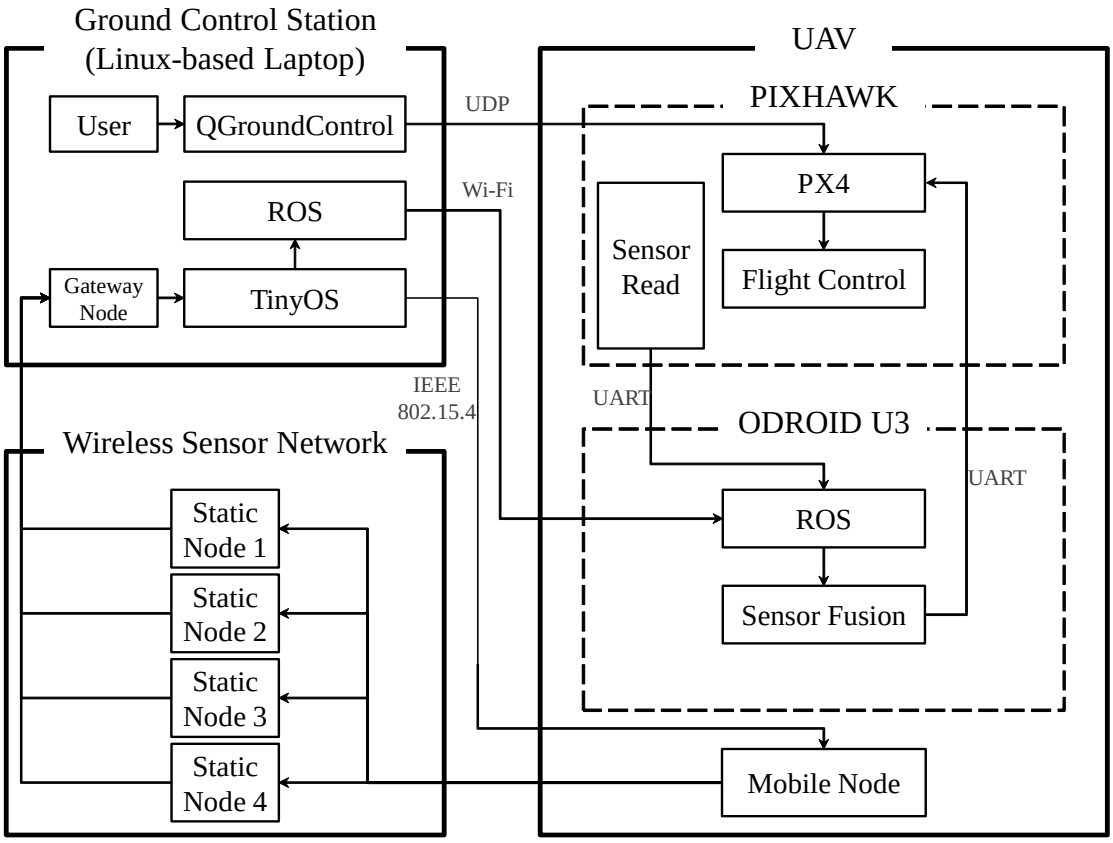


Figure 5.8. Architecture of Autonomous Flight System

The autonomous flight test is divided into three different missions. Three missions are performed under the same test environment but different desired trajectories such as holding position of the UAV at a desired waypoint, following straight waypoints,

and following square spiral waypoints. These different trajectories cover essential movements of the UAV in an indoor environment.

Test results support the validity of the proposed system in following sections by calculating the difference between the measured state and the desired state. x, y, z, and yaw are considered as desired states to determine the difference. The desired state can be defined by the operator and the measured state is obtained from the flight log data and RSS-based navigation system. Note that the RMS and maximum error in simulation is obtained by mean of each state in Monte Carlo simulation for 100 runs while the RMS and maximum error in the experiment is obtained by a single test of each mission. Since the purpose of the autonomous flight tests is to observe the control performance of the proposed autonomous system, the Kinect is not used as a reference in following tests.

5.4.1 Mission 1: Position Hold

Mission Statement

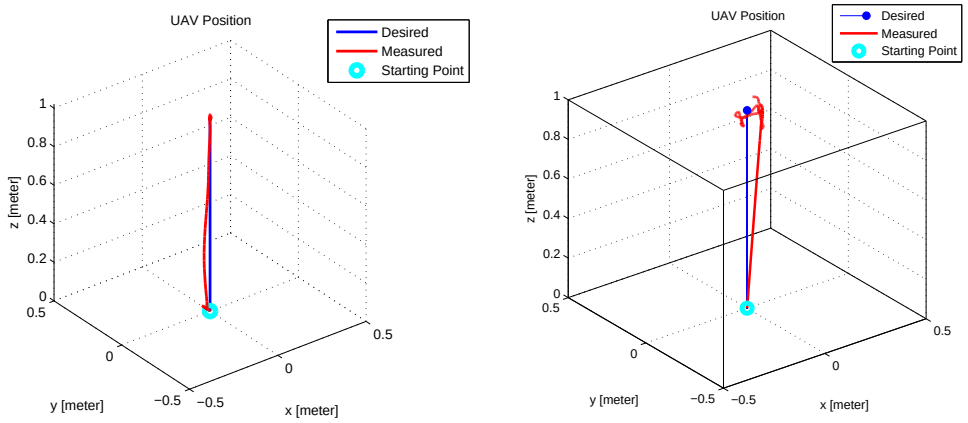
The goal of mission 1 is to make the UAV holds its position autonomously at the desired point during the flight. This mission is designed to evaluate the performance of the developed estimator with the the state feedback control. The mission of holding position will fail if the RSS-based indoor navigation system provides insufficient position data stream in real-time. The mission is tested in the classroom in Neil Armstrong Hall of Engineering at Purdue University. The desired position for the hovering is defined as $\mathbf{L}_d = [0\ 0\ 1]^T$ at the inertial reference frame.

Result

The desired flight trajectory and actual measurements in simulation and experiment are presented in Figure 5.9. Both trajectory plots show that the UAV tries to stay in its position at the desired point. RMS and maximum errors of both simulation

Table 5.3. Mission 1: Errors of Autonomous UAV Flight

	$x[m]$	$y[m]$	$z[m]$	yaw$[rad]$
RMS error in Simulation	0.0028	0.0153	0.1454	0.0024
RMS error in Experiment	0.0355	0.0318	0.0597	0.0328
Max. error in Simulation	0.0108	0.0566	1.0000	0.0064
Max. error in Experiment	0.1022	0.0528	1.0000	0.0642



(a) UAV Position in Simulation

(b) UAV Position in Experiment

Figure 5.9. Mission 1: UAV Position

and experiment are summarized in Table 5.3. The RMS error for each state in the simulation is smaller than each state in the experiment since the RMS error in simulation is obtained by the mean of Monte Carlo simulation for 100 runs and thus the smaller error of the simulation was expected. The RMS error of the state z in the experiment is relatively larger than other states such as x, y, and yaw. This can be seen in Table 5.3 where the maximum error of state z in both simulation and experiment are 1 meter. This is because the initial position and the desired position of the UAV have no huge difference between each other in x, y, and yaw, but the input for z is

the step function which has 15 seconds for the step time, 0 meter for the initial value, and 1 meter for the final value. More details about the time history of z can be found in the plot in the third row of Figure 5.10 and 5.11. In the figure, the input jumps at 15 seconds and the corresponding maximum error is observed between 15 and 20 seconds. Therefore, as can be seen in the table and figures, each RMS error and maximum error in the experiment are reasonable so that the proposed autonomous flight system based on the state estimation by the proposed indoor navigation system is feasible to control the UAV.

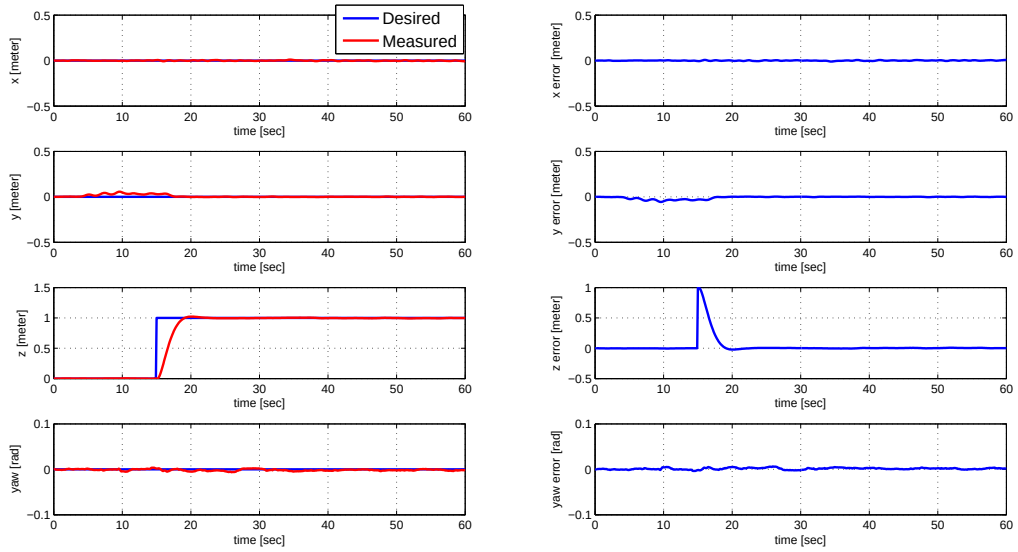


Figure 5.10. Mission 1: Simulation of Autonomous UAV Flight

5.4.2 Mission 2: Straight Waypoints

Mission Statement

The goal of mission 2 is to make the UAV follows simple straight waypoints autonomously. This mission is an extension of mission 1 which is designed to observe the entire control performace of the proposed autonomous flight system. The general

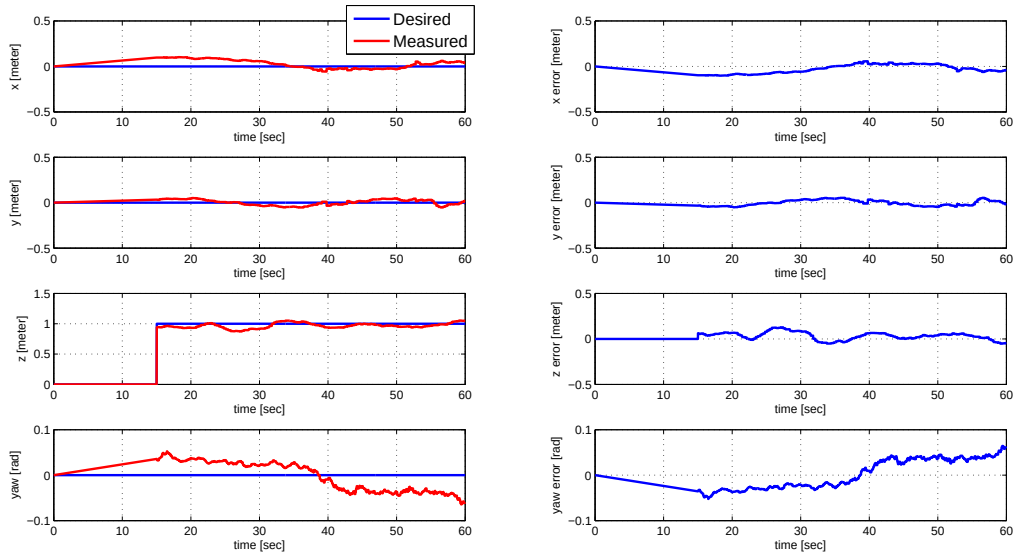


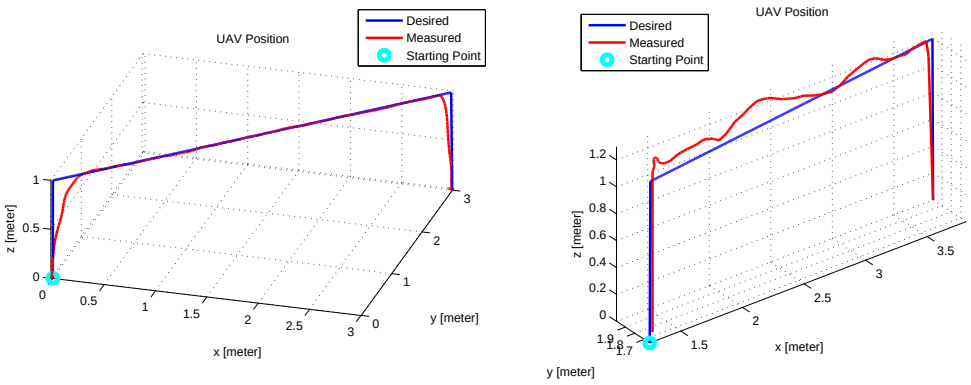
Figure 5.11. Mission 1: Experiment of Autonomous UAV Flight

setup is described in Section 5.4 which is identical to mission 1 except for the different desired waypoints. The desired trajectory is defined to takeoff at the origin and then land at $[3\ 3\ 0]^T$ with 1 meter altitude during the flight in the inertial reference frame. The desired trajectory and measured trajectory for both simulation and experiment are described in Figure 5.12.

Table 5.4. Mission 2: Errors of Autonomous UAV Flight

	$x[m]$	$y[m]$	$z[m]$	yaw$[rad]$
RMS error in Simulation	0.0804	0.0800	0.0986	0.1596
RMS error in Experiment	0.0924	0.0705	0.0415	0.0405
Max. error in Simulation	0.1160	0.1098	0.1983	1.0000
Max. error in Experiment	0.1901	0.1210	0.1127	0.1153

Result



(a) UAV Position in Simulation (b) UAV Position in Experiment

Figure 5.12. Mission 2: UAV Position

The difference between the desired state and actual measurements is presented in Table 5.4. For the same reason with the previous mission, errors in the simulation are relatively smaller than errors in the experiment. In Figure 5.13 and 5.14, the inputs for x and y are defined as increasing functions and the input for yaw is defined as a constant. The input for z is represented as the rectangular function with some point of rising and falling. It can be seen that states in the experiment still contain minor effects of measurement noise after filtering steps but the entire performance is acceptable because the maximum error in experiment does not exceed 20 centimeters. Therefore, the proposed autonomous flight system based on the state estimation by the proposed indoor navigation system is feasible for autonomous flight of a UAV.

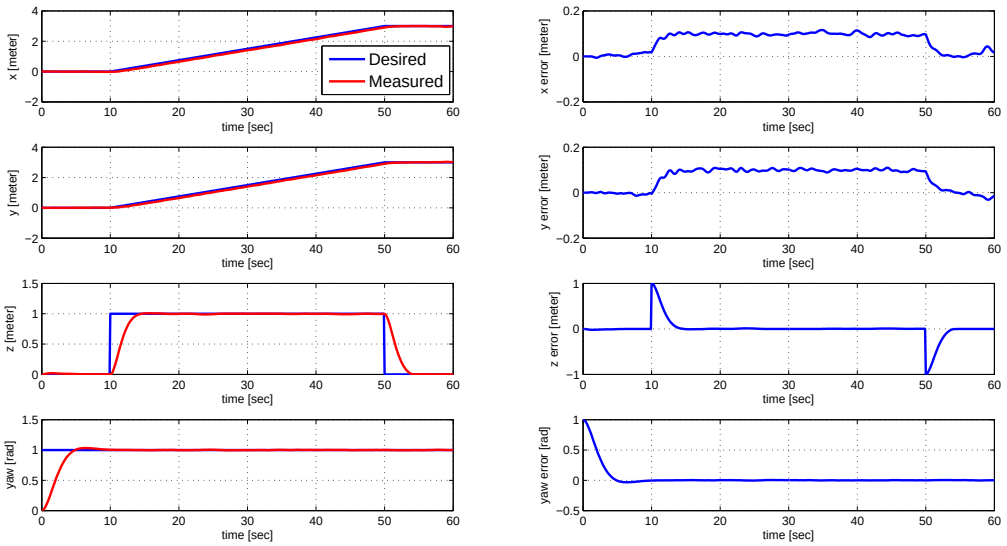


Figure 5.13. Mission 2: Simulation of Autonomous UAV Flight

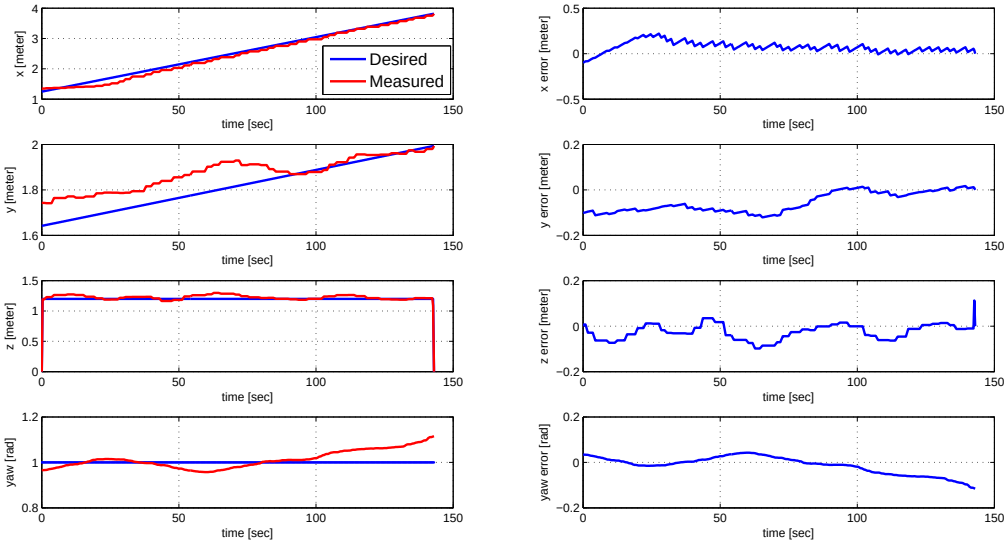


Figure 5.14. Mission 2: Experiment of Autonomous UAV Flight

5.4.3 Mission 3: Square Spiral Waypoints

Mission Statement

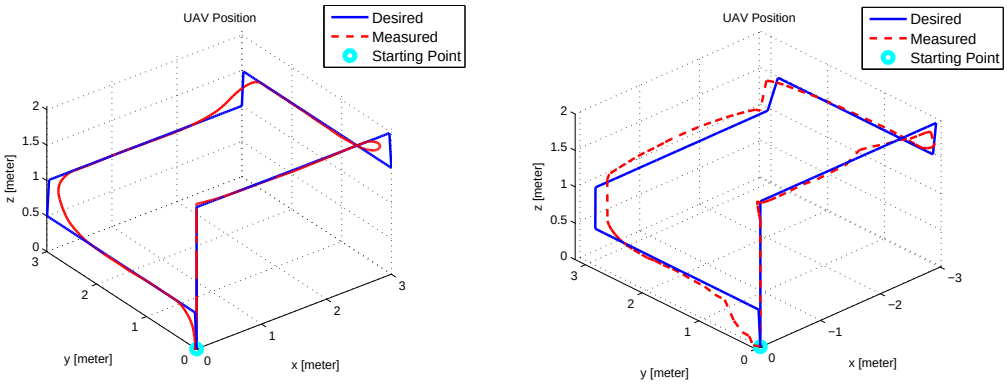
The goal of mission 3 is to make the UAV follows a square spiral-shaped trajectory. It is a more general situation for the UAV because this mission covers most fundamental motions of the UAV such as takeoff, ascending, descending, and landing. The UAV takeoff from the origin until it reaches the desired altitude and then the UAV changes its altitude to a descending square spiral at certain time for several times. The desired trajectory and measured trajectory for both simulation and experiment are described in Figure 5.15. The test setup is same with mission 1 and 2.

Table 5.5. Mission 3: Errors of Autonomous UAV Flight

	$x[m]$	$y[m]$	$z[m]$	yaw$[rad]$
RMS error in Simulation	0.2167	0.1184	0.1645	0.1595
RMS error in Experiment	0.2511	0.1631	0.2003	0.2204
Max. error in Simulation	0.4087	0.4010	0.9989	1.0000
Max. error in Experiment	0.6190	0.4360	1.0616	1.0000

Result

Note that the relatively large error of the state z is due to the initial jump from 0 to 2 meters at 5 seconds in the desired z command as shown in the third row of Figures 5.16 and 5.17. Therefore, as can be seen in the table and figures, both RMS error and maximum error in the experiment are reasonable so that the proposed autonomous flight system based on the RSS-based navigation system is feasible to control the UAV in an indoor environment.



(a) UAV Position in Simulation (b) UAV Position in Experiment

Figure 5.15. Mission 3: UAV Position

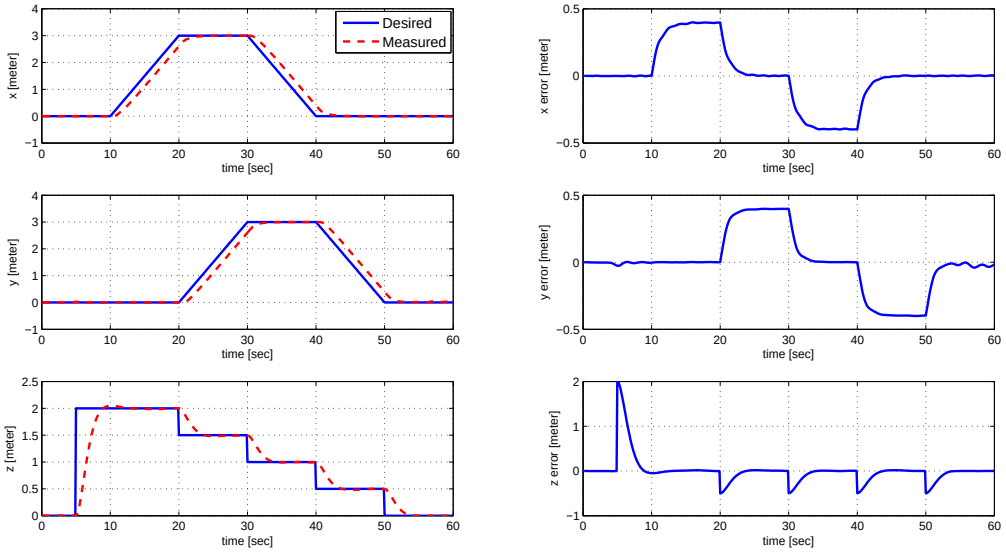


Figure 5.16. Mission 3: Simulation of Autonomous UAV Flight

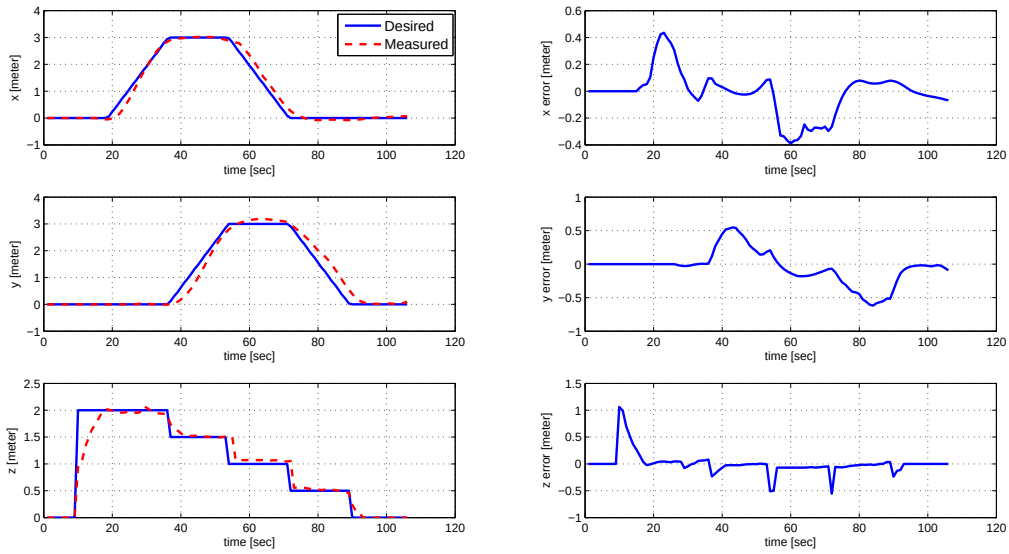


Figure 5.17. Mission 3: Experiment of Autonomous UAV Flight

6. CONCLUSIONS AND FUTURE WORKS

This paper considers autonomous Unmanned Aerial Vehicle (UAV) flight in an indoor environment. In seeking to address this problem, we have developed a Received Signal Strength(RSS)-based navigation system which can successfully replace GPS measurements which are unreliable or unavailable in indoor environments. To improve the performance of the RSS-based navigation system, we have first proposed a method to find the optimal values of two parameters, the path-loss exponent and the transmitted power, which allow for better position measurements. The RSS measurements are then fed into the extended Kalman filter where they are combined with the inertial measurement unit measurements to compute the position and attitude estimates of the UAV. Then feedback controllers which use the estimated states are designed to achieve the the desired position and heading. Finally, simulation and flight tests have been performed and their results are presented to demonstrate the performance of the autonomous flight system based on the RSS-based navigation system for indoor navigation. Note that the desired performance discussed in Table 1.1 has been achieved in terms of computational efficiency, reliability, accuracy, and cost. In the future, this work can be extended to more advanced applications such as path planning and collision avoidance for multiple UAV indoor operations.

LIST OF REFERENCES

LIST OF REFERENCES

[1] Veerachai Malyavej, Warapon Kumkeaw, and Manop Aorpimai. Indoor robot localization by rssi/imu sensor fusion. In *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2013 10th International Conference on*, pages 1–6. IEEE, 2013.

[2] Allen D Wu, Eric N Johnson, Michael Kaess, Frank Dellaert, and Girish Chowdhary. Autonomous flight in gps-denied environments using monocular vision and inertial sensors. 2013.

[3] Haiyang Chao, Yu Gu, and M. Napolitano. A survey of optical flow techniques for uav navigation applications. In *International Conference on Unmanned Aircraft Systems*, pages 710–716, May 2013.

[4] Dominik Honegger, Lorenz Meier, Petri Tanskanen, and Marc Pollefeys. An open source and open hardware embedded metric optical flow cmos camera for indoor and outdoor applications. In *IEEE International Conference on Robotics and Automation*, pages 1736–1741, 2013.

[5] Slawomir Grzonka, Giorgio Grisetti, and Wolfram Burgard. A fully autonomous indoor quadrotor. *IEEE Transactions on Robotics*, 28(1):90–100, 2012.

[6] Tomasz Krokowicz, Miguel Gasca, Holger Voos, and Dariusz Ucinski. Indoor navigation for quadrotor uavs using schematic environment maps. In *IEEE 19th International Workshop on Robotics in Alpe-Adria-Danube Region*, pages 457–462, 2010.

[7] Bonita. *A Motion Capture Solution: Bonita*. Vicon. `http://www.vicon.com/Content/FileUpload/System/BonitaOptical.pdf`.

[8] Holger Karl and Andreas Willig. *Protocols and architectures for wireless sensor networks*. John Wiley & Sons, 2007.

[9] Karel Heurtefeux and Fabrice Valois. Is rssi a good choice for localization in wireless sensor network? In *IEEE International Conference on Advanced Information Networking and Applications*, pages 732–739, 2012.

[10] Hui Liu, Houshang Darabi, Pat Banerjee, and Jing Liu. Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(6):1067–1080, 2007.

[11] Charalampos Papamanthou, Franco P Preparata, and Roberto Tamassia. Algorithms for location estimation based on rssi sampling. In *Algorithmic Aspects of Wireless Sensor Networks*, pages 72–86. Springer, 2008.

[12] Youngjune Gwon, Ravi Jain, and Toshiro Kawahara. Robust indoor location estimation of stationary and mobile users. In *INFOCOM Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 1032–1043, 2004.

[13] Heikki Laitinen, Suvi Juurakko, Timo Lahti, Risto Korhonen, and Jaakko Lahteenmaki. Experimental evaluation of location methods based on signal-strength measurements. *IEEE Transactions on Vehicular Technology*, 56(1):287–296, 2007.

[14] Fernando Seco, A Jimenez, Carlos Prieto, Javier Roa, and Katerina Koutsou. A survey of mathematical methods for indoor localization. *Intelligent Signal Processing*, pages 9–14, 2009.

[15] Xinrong Li. Rss-based location estimation with unknown pathloss model. *IEEE Transactions on Wireless Communications*, 5(12):3626–3633, 2006.

[16] Robert Mahony, Vijay Kumar, and Peter Corke. Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE Robotics & Automation Magazine*, 19(3):20–32, 2012.

[17] Jon T Adams. An introduction to ieee std 802.15. 4. In *IEEE Aerospace Conference*, pages 8–pp, 2006.

[18] Kaveh Pahlavan and Prashant Krishnamurthy. *Principles of Wireless Access and Localization*. Wiley, first edition, 2013.

[19] Theodore S. Rappaport. *Wireless Communications : Principles and Practice*. Prentice Hall PTR, second edition, 2002.

[20] Santiago Mazuelas, Alfonso Bahillo, Ruben M Lorenzo, Patricia Fernandez, Francisco A Lago, Eduardo Garcia, Juan Blas, and Evaristo J Abril. Robust indoor positioning provided by real-time rssi values in unmodified wlan networks. *IEEE Journal of Selected Topics in Signal Processing*, 3(5):821–831, 2009.

[21] Gaddi Blumrosen, Bracha Hod, Tal Anker, Danny Dolev, and Boris Rubinsky. Continuous close-proximity rssi-based tracking in wireless sensor networks. In *International Conference on Body Sensor Networks*, pages 234–239, 2010.

[22] Robert V Hogg, Joseph McKean, and Allen T Craig. *Introduction to mathematical statistics*. Pearson Education, 2005.

[23] IRIS. *3DR IRIS*. 3DRobotics. `http://3drobotics.com/iris/`.

[24] Samir Bouabdallah, Pierpaolo Murrieri, and Roland Siegwart. Design and control of an indoor micro quadrotor. In *IEEE International Conference on Robotics and Automation*, volume 5, pages 4393–4398, 2004.

[25] Leonardo Ascorti. An application of extended kalman filter to the attitude control of a quadrotor. Master's thesis, Polytechnic University of Milan, 2013.

[26] STLSM303D. *ST LSM303D Data Sheet*. ST Microelectronics. `http://www.st.com/web/en/resource/technical/document/datasheet/DM00057547.pdf`.

[27] STL3GD20. ST Microelectronics.

[28] Peter Corke. *Robotics, vision and control fundamental algorithms in MATLAB*. Springer, first edition, 2011.

[29] Min-Jea Tahk and Byung-Chan Sun. Coevolutionary augmented lagrangian methods for constrained optimization. *Evolutionary Computation, IEEE Transactions on*, 4(2):114–124, Jul 2000.

[30] *Robot Operating System*. `http://http://www.ros.org/`.

[31] *TinyOS*. `http://www.tinyos.net/`.

[32] David Gay, Philip Levis, Robert Von Behren, Matt Welsh, Eric Brewer, and David Culler. The nesc language: A holistic approach to networked embedded systems. In *Acm Sigplan Notices*, volume 38, pages 1–11. ACM, 2003.

[33] *Pixhawk Autopilot*. `https://pixhawk.org/modules/pixhawk`.

[34] CM3300. *802.15.4 Mote Module: CM3300*. Advanticsys. `http://www.advanticsys.com/wiki/index.php?title=CM3300`.

[35] CM5000. *802.15.4 Mote Module: CM5000*. Advanticsys. `http://www.advanticsys.com/wiki/index.php?title=CM5000-SMA`.

[36] CC2420. *2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver*. Texas Instruments. `http://www.ti.com.cn/cn/lit/ds/symlink/cc2420.pdf`.

[37] MSP430F1611. *MSP430F1611*. Texas Instruments. `http://www.ti.com/product/msp430f1611`.

[38] Microsoft. *MS Kinect*. Microsoft. `http://msdn.microsoft.com/en-us/library/jj131033.aspx`.

[39] Kourosh Khoshelham and Sander Oude Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437–1454, 2012.

APPENDIX

A. Extended Kalman Filter

This appendix provides all the equations necessary for developing the estimator described in 3.3. The state vector which is to be estimated is:

$$\mathbf{x} = \begin{bmatrix} x & y & z & \phi & \theta & \psi & u & v & w & \mathbf{b}_{accel} & \mathbf{b}_{gyro} \end{bmatrix}^{\mathrm{T}} \tag{A.1}$$

The system model is described as:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \tag{A.2}$$

where

$$\begin{aligned}
\dot{x} &= u\cos\theta\cos\psi + v(\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi) + w(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi) \\
\dot{y} &= u\cos\theta\sin\psi + v(\sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi) + w(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi) \\
\dot{z} &= -u\sin\theta + v\sin\phi\cos\theta + w\cos\phi\cos\theta \\
\dot{\phi} &= (y_p - b_{g_x}) + (y_q - b_{g_y})\sin\phi\tan\theta + (y_r - b_{g_z})\cos\phi\tan\theta \\
\dot{\theta} &= (y_q - b_{g_y})\cos\phi - (y_r - b_{g_z})\sin\theta \\
\dot{\psi} &= (y_q - b_{g_y})\sin\psi\sec\theta + (y_r - b_{g_z})\cos\phi\sec\theta \\
\dot{u} &= (y_r - b_{g_z})v - (y_q - b_{g_y})w + g\sin\theta - b_{a_x} \\
\dot{v} &= (y_p - b_{g_x})w - (y_r - b_{g_z})u - g\cos\theta\sin\phi - b_{a_y} \\
\dot{w} &= (y_q - b_{g_y})u - (y_p - b_{g_x})v - g\cos\theta\cos\phi - b_{a_z} \\
\dot{b}_{a_x} &= 0 \\
\dot{b}_{a_y} &= 0 \\
\dot{b}_{a_z} &= 0 \\
\dot{b}_{g_x} &= 0 \\
\dot{b}_{g_y} &= 0 \\
\dot{b}_{g_z} &= 0
\end{aligned} \tag{A.3}$$

Measurement models are defined as:

$$\mathbf{y}_{\text{IMU}} = h_1(\mathbf{x}) \tag{A.4}$$

$$\mathbf{y}_{\text{RSS}} = h_2(\mathbf{x}) \tag{A.5}$$

where

$$\begin{aligned}
y_{\text{a}_x} &= rv - qw + g\sin\theta + b_{\text{a}_x} + w_{\text{a}_x} \\
y_{\text{a}_y} &= pw - ru - g\cos\theta\sin\phi + b_{\text{a}_y} + w_{\text{a}_y} \\
y_{\text{a}_z} &= qu - pv - g\cos\theta\cos\phi + b_{\text{a}_z} + w_{\text{a}_z} \\
y_{\text{g}_x} &= p + b_{\text{g}_x} + w_{\text{g}_x} \\
y_{\text{g}_y} &= q + b_{\text{g}_y} + w_{\text{g}_y} \\
y_{\text{g}_z} &= r + b_{\text{g}_z} + w_{\text{g}_z} \\
y_{\text{RSS}_x} &= P_L - 10n_i\log_{10}\{\sqrt{(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2}\} + w_{\text{RSS}_x} \\
y_{\text{RSS}_y} &= P_L - 10n_i\log_{10}\{\sqrt{(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2}\} + w_{\text{RSS}_y} \\
y_{\text{RSS}_z} &= P_L - 10n_i\log_{10}\{\sqrt{(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2}\} + w_{\text{RSS}_z}
\end{aligned} \tag{A.6}$$

Note that first six equations are measured by IMU and next three equations are measured by RSS. Jacobian matrices are given by:

$$\mathbf{F_k} = \frac{\partial f}{\partial \mathbf{x}} \tag{A.7}$$

$$\mathbf{H_k} = \frac{\partial h}{\partial \mathbf{x}} \tag{A.8}$$

B. Simulation Setup

In this appendix, the fundamental information of the simulation is provided such as initial parameters for the numerical simulation as presented in Chapter 4.

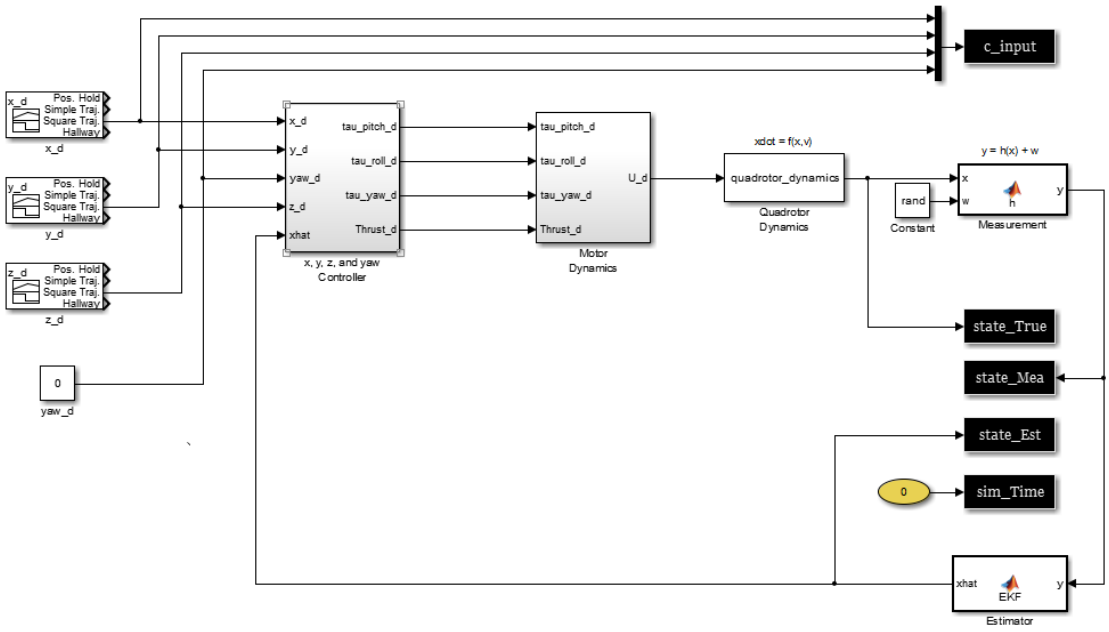


Figure B.1. Integrated Block Diagram in Simulink

Note that the simulink model of the simulation is developed base on [28]. Note also that, in Table B.1, most of mechanical properties are given by technical specifications from providers while some of parameters such as the moment of inertia and aerodynamic coefficients are calculated empirically.

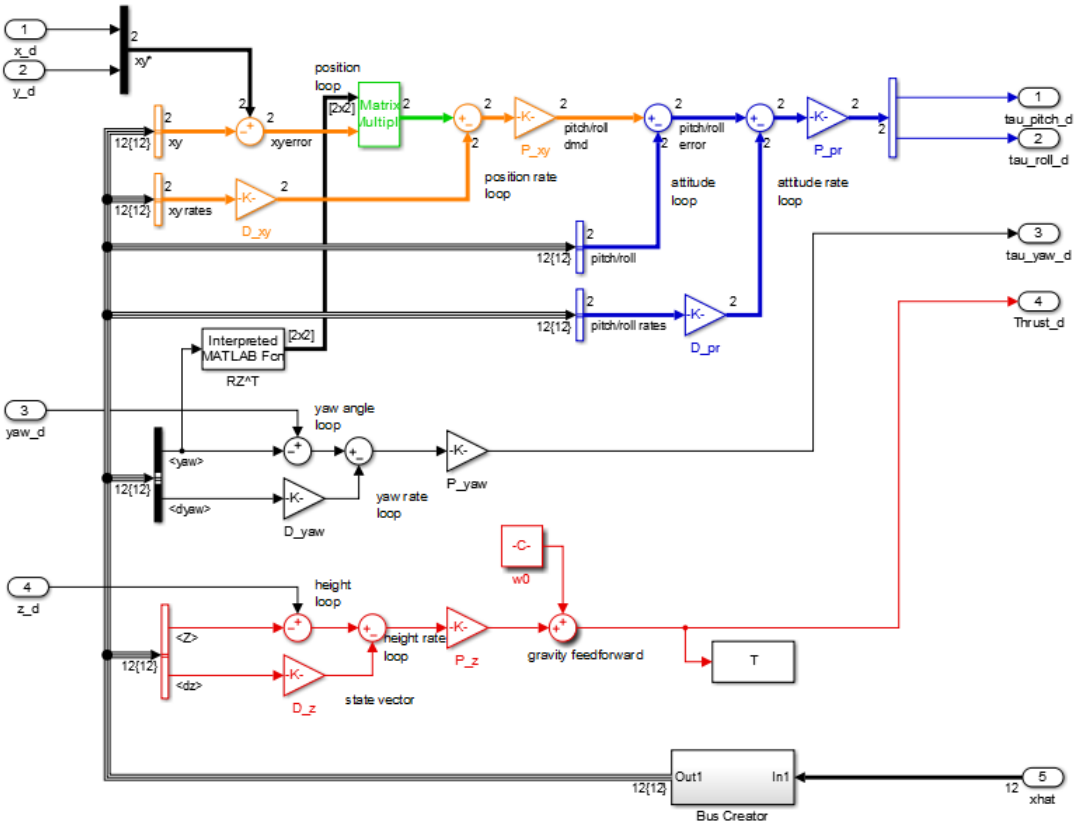


Figure B.2. Subsystem of Position and Attitude Controller

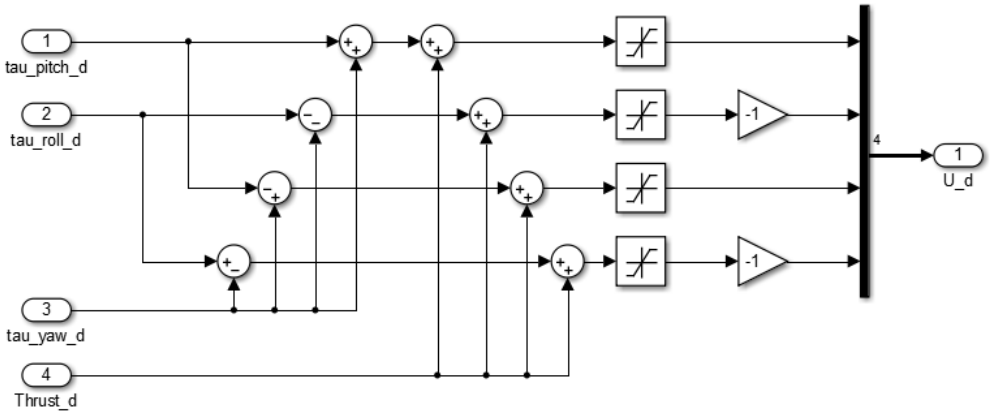


Figure B.3. Subsystem of Motor Controller

Table B.1. Simulation Parameters

Symbol	Value	Unit	Description
J	80.0098	N/A	Cost of 4.2.1
n_1	2.363	N/A	Optimal path-loss component of 1st static node
n_2	3.201	N/A	Optimal path-loss component of 2nd static node
n_3	3.385	N/A	Optimal path-loss component of 3rd static node
n_4	2.877	N/A	Optimal path-loss component of 4th static node
P_T	-19.84	dBm	Optimal transmission power of the mobile node
P_{xy}	-0.40056	N/A	Proportional gain with respect to x and y
P_{xy2}	0.21554	N/A	Proportional gain with respect to x and y
D_{xy}	1.32003	N/A	Derivative gain with respect to x and y
P_{pr}	-271.81855	N/A	Proportional gain with respect to *pitch* and *roll*
D_{pr}	0.33339	N/A	Derivative gain with respect to *pitch* and *roll*
P_{yaw}	-104.54870	N/A	Proportional gain with respect to *yaw*
D_{yaw}	2.10289	N/A	Derivative gain with respect to *yaw*
P_z	-40.43319	N/A	Proportional gain with respect to z
D_z	1.59259	N/A	Derivative gain with respect to z
g	9.81	m/s^2	Gravity
ρ	1.225	kg/m^3	Density of air
μ_v	1.5×10^{-5}	m/s	Velocity of air
m	2	kg	Mass of quadrotor
I_x	0.082	kgm^2	Moment of Inertia with respect to x-axis
I_y	0.082	kgm^2	Moment of Inertia with respect to y-axis
I_z	0.149	kgm^2	Moment of Inertia with respect to z-axis
d	0.225	m	Length of arm
r	0.127	m	Radius of blade
c	0.018	m	Chord of blade
m_b	0.005	kg	Mass of blade
c_t	0.0048	N/A	Thrust coefficient
c_q	2.35×10^{-4}	N/A	Torque coefficient