



Modeling, Simulation and Control System Design for Civil Unmanned Aerial Vehicle (UAV)

Shahriar Bagheri

**Modeling, Simulation and Control System Design
for Civil Unmanned Aerial Vehicle (UAV)**

by

Shahriar Bagheri

Submitted to the Department of Applied Physics and Electronics
in partial fulfillment of the requirements for the degree of

Master of Science in Electronics,
Specialization in Robotics and Control

at

Umeå University

November 2014

Author
Department of Applied Physics and Electronics
November 12, 2014

Certified by

Leonid B. Freidovich

Associate Professor, Umeå University

Thesis Supervisor

Certified by

Olle Hagner

Head of Research, SmartPlanes AB.

Thesis Supervisor

Assessed by

Sven Rönnbäck

Associate Professor, Umeå University

Thesis Examiner

Course title

Master's Thesis in Electronics Engineering

Specialization in Robotics and Control

30.0 Credits - 5EL205

Abstract

Unmanned aerial systems have been widely used for variety of civilian applications over the past few years. Some of these applications require accurate guidance and control. Consequently, Unmanned Aerial Vehicle (UAV) guidance and control attracted many researchers in both control theory and aerospace engineering. Flying wings, as a particular type of UAV, are considered to have one of the most efficient aerodynamic structures. It is however difficult to design robust controller for such systems. This is due to the fact that flying wings are highly sensitive to control inputs. The focus of this thesis is on modeling and control design for a UAV system. The platform under study is a flying wing developed by SmartPlanes Co. located in Skellefteå, Sweden. This UAV is particularly used for topological mapping and aerial photography.

The novel approach suggested in this thesis is to use two controllers in sequence. More precisely, Linear Quadratic Regulator (LQR) is suggested to provide robust stability, and Proportional, Integral, Derivative (PID) controller is suggested to provide reference signal regulation. The idea behind this approach is that with LQR in the loop, the system becomes more stable and less sensitive to control signals. Thus, PID controller has an easier task to do, and is only used to provide the required transient response.

The closed-loop system containing the developed controller and a UAV non-linear dynamic model was simulated in SIMULINK. Simulated controller was then tested for stability and robustness with respect to some parametric uncertainty. Obtained results revealed that the LQR successfully managed to provide robust stability, and PID provided reference signal regulation.

The thesis is mainly divided into four parts. The first part focuses on mathematical modeling and equations of motion. It explains 6 Degree of Freedom (DoF) UAV non-linear equations and linearization using small-disturbance theory. Static and dynamic stability analysis that was accomplished using DATCOM software, had been explained in part one as well. The second part presents guidelines and explains the design procedure of LQR and Proportional, Integral, Derivative controllers, used for reference signal tracking. The third part describes how to simulate the closed-loop system in SIMULINK and finally part four, presents the result of numerical simulations. Result of these simulations were studied based on system stability, reference signal regulation and robustness.

Acknowledgments

I am deeply indebted to my supervisor Assoc. Prof. Leonid Freidovich for his valuable guidance, patience and support throughout this research. Your brilliance in mathematics and passion towards control theory have contributed immeasurably to my growth as an engineer. I am forever thankful.

My sincere appreciation goes to my second supervisor Olle Hagner, who shared with me his experience and technical knowledge of UAV systems. Your enthusiasm for the field of aeronautics and aerospace was an amazing propulsive force.

My gratitude to the SmartPlanes company and all its amazing staff. First, for giving me the opportunity to conduct my Masters thesis at SmartPlanes, and second, for providing me with everything that I needed throughout the research.

I would like to thank my best friend and colleague Tural Jafarov. We collaborated together on a large portion in this research. I am grateful for all the enjoyable moments and interesting discussions. Special thanks also goes to the people at the Robotics and Control Lab (RCLab) at Umeå University, especially PhD students Daniel Ortiz Morales and Szabolcs Fodor for their helpful suggestions during the work. I would like also to acknowledge valuable comments and suggestions from Dr. Sven Rönnbäck.

Last but not least, my mother and my brother who support me no matter what. I am forever grateful and I am sure that none of this would have been possible without your support.

Umeå, November 12, 2014

Shahriar Bagheri

.....

Contents

List of Symbols	13
Acronyms	16
1 Introduction	19
1.1 Research Outline	19
1.1.1 Background	19
1.1.2 Objective	20
1.1.3 Project Requirements	21
1.1.4 Resources	21
1.1.5 Delivaries	22
1.2 The UAV Platform	22
1.3 State-of-the-art	24
1.4 Organization of Thesis	27
2 Equations of Motion	31
2.1 Reference Frames	31
2.2 Non-linear Equations of Motion	35
2.3 Linearized Equations of Motion	36
2.3.1 Small-Disturbance theory	36
2.3.2 Calculation of Stability Derivatives	39
2.3.3 State-Space Representation of the Linearized Model	43

3 Control System Design	47
3.1 Controllability and Observability	48
3.2 Linear Quadratic Regulator	50
3.2.1 Theory	50
3.2.2 Solution for the LQR problem	53
3.3 PID Controller	54
4 Simulink Models for Computer Simulation	57
4.1 UAV Non-linear Simulation Model	58
4.1.1 DATCOM	58
4.1.2 SmartFly Dynamics Simulation	60
4.2 Simulation of Control Loops	64
4.2.1 Reference Block	64
4.2.2 Longitudinal and Lateral Controllers	64
5 Results of Numerical Simulations	69
5.1 Stabilization of the System	69
5.2 Reference Signal Regulation	72
5.2.1 Altitude and Airspeed Regulation	72
5.2.2 Roll Angle Tracking	75
5.3 Controller Robustness	77
5.3.1 Variation in UAV Mass	77
5.3.2 Variation in UAV CG Location	78
6 Conclusion and Discussion	83
6.1 Discussion	83
6.2 Project Requirements and Objectives - Status	85
6.3 Implications of the Research	86
6.4 Summary	87
6.5 Ethics of the Work	88
6.6 Future Work	88

A DATCOM	91
B Meeting Minute	93
C MATLAB Script	95
Bibliography	111

CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

1-1	SmartFly 3 angle view with attached frame	22
1-2	System Overview	23
2-1	Earth surface and body reference frames	32
2-2	Forces in stability and body axis	34
3-1	Block diagram of the proposed controller for longitudinal controller	48
3-2	Generalized Plant	50
4-1	Lift Coefficient C_L Due to change in α	58
4-2	Pitching Moment C_M Due to change in α	59
4-3	Incremental Change in C_L Due to Elevator Deflection δ_e	59
4-4	Incremental Change in rolling moment coefficient C_l Due to aileron Deflection δ_a	60
4-5	Overview of the Simulation Model	61
4-6	Non-linear System	62
4-7	UAV Dynamics	62
4-8	Inside of the "Aeroblock"	63
4-9	Longitudinal Controller	66
4-10	Lateral Controller	67
5-1	Trim Condition: Airspeed	70
5-2	Trim Condition: Altitude Hold	70
5-3	Trim Condition: Roll Angle (ϕ)	71
5-4	Trim Condition: Pitch Angle (θ)	71

LIST OF FIGURES

5-5 Trim Condition: Yaw Angle (ψ)	72
5-6 Altitude Tracking: Δh	73
5-7 UAV Descent Due to Altitude Command	73
5-8 UAV Descending with Constant Airspeed	74
5-9 Airspeed tracking at $T = 50$	74
5-10 Altitude tracking at $T = 25$	75
5-11 $\Delta\Phi$ Tracking at $25 < T < 35$	75
5-12 Control signal δa	76
5-13 Changes in V_{air} while rolling at $25 < T < 35$	76
5-14 Changes in h while rolling at $25 < T < 35$	77
5-15 Comparison of Height Tracking for Different UAV Mass	78
5-16 Comparison of Airspeed Stabilization for Different UAV Mass	79
5-17 Comparison of Height Tracking for Different Center of Gravity (CG) Location . .	80
5-18 Comparison of Airspeed Stabilization for Different CG Location	80
5-19 Actual Altitude for Different CG Location	81

List of Tables

1.1	Preliminary Project Requirements List	21
1.2	Project Resources	28
1.3	List of Expected Deliveries	29
1.4	SmartFly Airframe	29
2.1	Trim Condition	37
2.2	Lateral Stability Derivatives	41
2.3	Numeric Values for Lateral Stability Derivatives	42
2.4	Aerodynamic Coefficients and Derivatives	43
2.5	Longitudinal Stability Derivatives	44
2.6	Numeric Values for Longitudinal Stability Derivatives	45
3.1	PID gains	55
6.1	Project Requirements Met	85
6.2	Deliveries to the customer	86

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

List of Symbols

α	Angle of attack
\bar{c}	Wing chord
β	Side slip angle
δ_a	Aileron deflection angle
δ_e	Elevator deflection angle
δ_T	Throttle opening
ϕ	Roll angle
ψ	Yaw angle
ρ	Air density
θ	Pitch angle
b	Wing span
C_D	Drag force coefficient
C_L	Lift force coefficient
C_l	Rolling moment coefficient
C_M	Pitching moment coefficient
C_N	Yawing moment coefficient

LIST OF TABLES

C_Y	Side force coefficient
C_X	Axial Force - Body axis
C_Z	Normal Force - Body axis
F_X	Total Force along Body X-axis
F_Y	Total Force along Body Y-axis
F_Z	Total Force along Body Z-axis
I_{xz}	Product of inertia around x and z axes
I_x	Moment of inertia around x-axis
I_y	Moment of inertia around y-axis
I_z	Moment of inertia around z-axis
L	Moment along aircraft X-axis
M	Moment along aircraft Y-axis
m	UAV mass
N	Moment along aircraft Z-axis
p	Angular velocity along aircraft X-axis
Q	Dynamic Pressure
q	Angular velocity along aircraft Y-axis
r	Angular velocity along aircraft Z-axis
S	Wing area
S_{prop}	Area swept by the propeller
u	Linear velocity along aircraft X-axis

u_c	Vector of control signals
v	Linear velocity along aircraft Y-axis
V_{air}	Airspeed
w	Linear velocity along aircraft Z-axis
X	Aerodynamic + Thrust Force - Body X-axis
X_b	Aircraft body X-axis
Y	Aerodynamic + Thrust Force - Body Y-axis
Y_b	Aircraft body Y-axis
Z	Aerodynamic + Thrust Force - Body Z-axis
Z_b	Aircraft body Z-axis

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

Acronyms

AHRS Attitude and Heading Reference System. 24

AoA Angle of Attack. 32, 33, 60, 63, 81

ARE Algebraic Riccati Equation. 53

CG Center of Gravity. 10, 20, 22, 32, 33, 35, 64, 69, 76–80, 82, 83, 85

DoF Degree of Freedom. 3, 25, 35, 62, 64, 85

HLC High Level Controller. 26, 27

IEEE Institute of Electrical and Electronics Engineers. 86

LLC Low Level Controller. 26, 27

LQR Linear Quadratic Regulator. 3, 26, 47, 48, 50, 53, 84, 85

NDI Nonlinear Dynamic Inversion. 26

PAMS Personal Aerial Mapping System. 22

PID Proportional-Integral-Derivative. 20, 47, 54, 55, 65, 84, 85

RC Radio Controller. 23

RPV Remotely Piloted Vehicles. 19

SQP Sequential Quadratic Programming. 26

Acronyms

UAV Unmanned Aerial Vehicle. 3, 4, 10, 19–22, 24–28, 31, 35, 36, 42, 44, 47, 48, 52, 57, 58, 60, 62–64, 69, 72–74, 76–78, 81, 82, 85, 86

Chapter 1

Introduction

1.1 Research Outline

1.1.1 Background

The term **Drone** is generally used by the media and public, and it refers to all types of pilotless aircraft. However, from a technical standpoint, there are significant differences between various types of unmanned airplanes. Reference [1] categorizes pilotless aircraft into three groups: Unmanned Aerial Vehicles (UAV), Remotely Piloted Vehicles (RPV) and Drones. The difference among the categories are mainly due to the type of the mission, flight envelope, size and most importantly, the level of autonomy of operation. For instance, RPVs are different from UAVs in a sense that RPVs, as the name implies, are steered and controlled from a ground station, but UAVs could operate autonomously, without any remote controller. Nevertheless, the term UAV will be used throughout this text as a generic term and includes all types of pilotless aircraft.

Since the 1990s, UAVs have found their way into the civilian market mainly due to the advancement in the field of micro-electronics. Small-size UAVs have been used since then by researchers in various disciplines, because of their operational simplicity, and affordability [2]. These systems usually employ Commercial Off The Shelf (COTS) autopilots that to most extent use traditional Proportional Integral Deriva-

tive controllers [3]. The experienced reader realizes that advantages of simplicity and relatively good performance in Proportional-Integral-Derivative (PID) controllers, are not without disadvantages. PID tuning process is time consuming and could be troublesome for an operator who most likely does not possess theoretical knowledge of the control loops. Also, since for many UAV systems, the dynamics of the UAV is either not fully understood by the control system designer or is too complex and highly nonlinear [2], often PID controllers could not guarantee high robustness which is required for some applications.

1.1.2 Objective

The main objective of this thesis project, which was done in close collaboration with a local company in Sweden (i.e. SmartPlanes AB. [4]), is to introduce a more robust controller schema and a more intuitive gain tuning procedure. Designed controllers are required to meet the following technical requirements:

- Controller must achieve stabilization of airspeed around ≈ 10 m/s. This is in fact the normal cruise speed;
- Controller must achieve stabilization of constant altitude around the normal flying altitude, which is ≈ 150 m;
- Controller must provide reference signal regulation for altitude and airspeed;
- Controller must achieve stabilization when there exists an imbalance in UAV mass, for the range between 50 and 300 grams;
- Controller must achieve stabilization when there exists a variation in CG location of maximum 4 cm (2 cm forward, and 2 cm aft of the original CG location);

These objectives were met and details are discussed in section 6.2.

1.1.3 Project Requirements

As mentioned earlier, this work was done towards a company. Consequently, certain requirements, resources and deliverable were set before the commencement of the project. Table 1.1 shows the original requirements list. The aim was to fulfill all these requirements within the project time frame. Some items in this list were not met within the project timeline. This is discussed in section 6.2.

Table 1.1: Preliminary Project Requirements List

No. Activity	Description
1. Project Planning	Preparation/approval of the project plan document
2. Preliminary Study	Problem analysis, identification of research approach
3. System Modeling	Mathematical modeling, simulation and parameter identification
4. Control System Design (Simulation)	Designing controller for longitudinal and lateral axes
5. Test (Simulation)	Performing series of tests on the simulated controller to validate its functionality
6. Control System Design (Implementation)	Hardware implementation of the designed controller
7. Result Analysis	Analyzing obtained results

1.1.4 Resources

This research is limited to computer simulations of developed algorithms. No system implementation were carried-out. Hence, project resources are limited to individuals and computer software. They are summarized in table 1.2 shown on page 28. Also, major part of this project was correlated with the work of Mr. Tural Jafarov, who was conducting research on automatic landing system for the same UAV platform [5]. Although we collaborated a lot through the process, individual thesis reports have been prepared. Separate simulation models were also developed in SIMULINK.

1.1.5 Deliveries

Expected project deliveries are tabulated in table 1.3.

1.2 The UAV Platform

The airframe used in this thesis is a small-size flying wing, called **SmartFly**, manufactured by SmartPlanes Co. [4] located in Skellefteå, Sweden. SmartPlanes has almost a decade of experience in the design and development of civilian UAVs. Their main product, Personal Aerial Mapping System (PAMS) [6], is being used worldwide for variety of applications in aerial mapping. Figure 1-1 illustrates the airframe configuration and Table 1.4 contains parametric data.

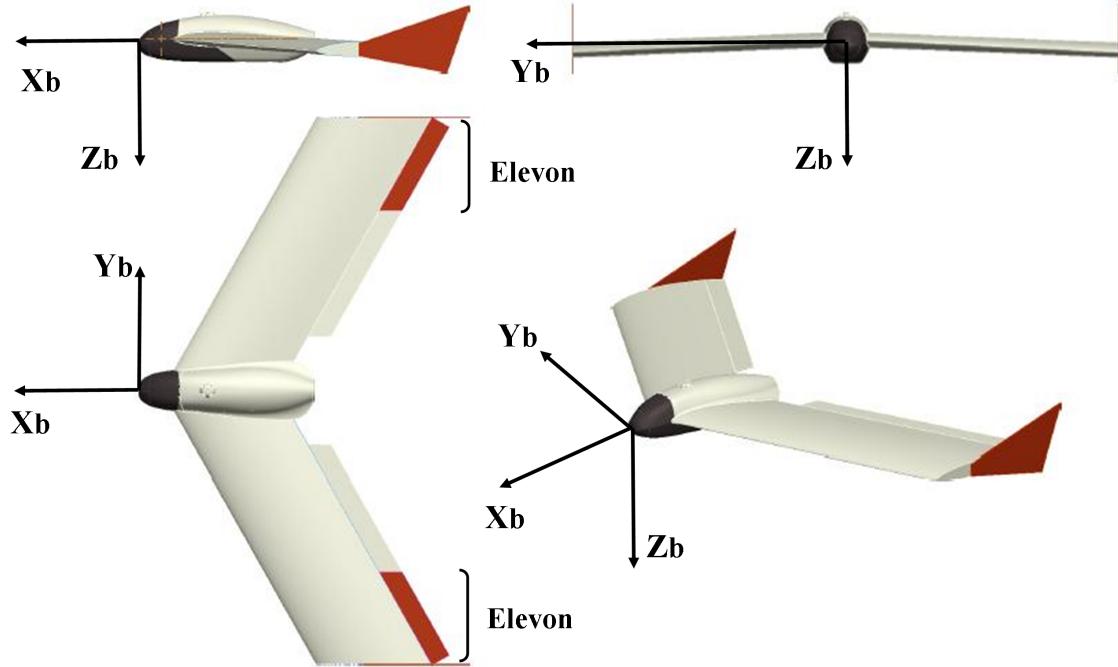


Figure 1-1: SmartFly 3 angle view with attached frame

As mentioned, SmartFly is a flying wing. Flying wings are assumed to have aerodynamically efficient airframes. This is due to the fact that the conventional stabilizers are removed [7]. Nevertheless, designing robust control system for flying wings is not an easy task. Because the distance between CG and control surface

is short, small deflections in any control surface would produce large moments. In other words, the plane is highly sensitive to control surface deflection. The aircraft is equipped with elevons as control surfaces on the trailing edge of each wing ¹. Differential movements of the right and left elevon, creates a moment around the aircraft's X_b axis and changes the roll angle, while simultaneous movement of both surfaces creates a moment around the aircraft's Y_b axis and changes the pitch angle.

SmartFly could be controlled in three different modes: **Manual**, **Assisted** and **Automatic**. The human operator could easily toggle among each of the operating modes using the switch, located on the Radio Controller (RC). An overview of the system ready to operate is shown in figure 1-2.



Figure 1-2: System Overview

¹Aircraft with conventional stabilizers are usually equipped with a set of ailerons located on each wing, and an elevator located on the horizontal stabilizer. Elevons in flying wings are ought to function as both ailerons and elevators.

According to the UAV user manual [8], in the manual mode: "*The pilot has direct control of the servos and throttle via the RC-transmitter like a conventional RC plane. The manual flight mode is mainly for diagnostic purposes*". Readings on the aircraft attitude, expressed as Euler Angles [9] (roll (ϕ), pitch (θ) and yaw (ψ)) are provided by the on-board Attitude and Heading Reference System (AHRS) and are transmitted to the ground control station at a specified rate. Information on the vehicle velocity (in m/s) and the battery capacity are also available to the pilot.

In the assisted control mode: "*The autopilot stabilizes the plane and the pilot controls the pitch, roll and throttle. This is the normal flight mode during take-off and landings since it provides the best and safest control. Control inputs are softened giving relaxing flying characteristics*".

In automatic mode: "*the aircraft flies according to the uploaded flight plan and flight commands received from the ground control station*".

1.3 State-of-the-art

This section presents an overview of methods used by other researchers for UAV modeling and control system design and intends to identify algorithms and methodologies that are suitable for this research. Due to the interdisciplinary nature of this work, two major categories of scientific literature have been reviewed: publications on UAV modeling, often tackled by researchers in the aerospace community, and research on UAV control algorithms, mostly carried out by experts in the field of control theory. Articles from both disciplines are discussed here, starting with literature on UAV modeling.

Model building is the most fundamental building block in the control system design process. In Aviation, a mathematical model of a physical platform is obtained from fundamental laws of physics and are generally derived using either Euler-Lagrange method or Newton approach. Although mathematical models are a necessary component of system modeling, they are insufficient and the engineer is required to develop a numerical model of the system as well. This is usually accomplished using either

computer software (i.e. Advance Aircraft Analysis [10], DIGITAL DATCOM [11]), or wind tunnel tests.

[12] focuses on modeling the longitudinal dynamics of a UAV system for steady-state flight condition (straight and level, un-accelerated flight). In order to do so, authors first compute the aerodynamic coefficients and derivatives using DIGITAL DATCOM [13] software. DATCOM computes numerical values for the aerodynamic coefficients (e.g. C_L , C_D , C_Y , C_l , C_M , C_N) necessary for dynamical modeling of various types of aircraft, using geometric information of the aircraft structure. [12] also presents an empirical model (with numerical values) of the UAV for the same flight condition. The values were obtained using Matlab system identification toolbox [14]. Since various sensor values were to be processed for empirical system modeling, a Kalman filter was used to reduce the effect of sensor noise. Both models were validated in a simulation environment where the steady-state flight condition was perturbed by an elevator deflection.

[15] also analyzes the dynamic of a UAV and extracts state-space model of the system using DATCOM and SIMULINK. They start by using the standard 6 Degrees of Freedom (DoF) equations of motion for aircraft (as in [9]), which are comprised of force, moment and kinematic equations. Authors then linearize equations over an equilibrium point being straight and level, un-accelerated flight, employing Small-Disturbance theory which is also fully described in chapter 3 in [9]. It should be mentioned that numerical values for aerodynamic coefficients and derivatives are computed in DATCOM, similar to [12].

[16] uses the Euler-Lagrange approach to derive equations of motion for Sky-Sailor UAV. This method uses the concepts of kinetic and potential energies to describe the dynamics of the system. It should be mentioned that since most UAVs studied herein are sub-sonic vehicles with limited flight envelopes, they can be treated as a rigid body of mass (m) in 3D space with 6 DoF, meaning that the effect of aeroservoelasticity and elasticity can be ignored. Applying Euler-Lagrange resulted in six equations (three force equations and three moment equations) which were then used (after simple modifications) for controller design purpose. There are two control problems

addressed in [16]: the path planning problem, for which a High Level Controller (HLC) was designed, and the system stability problem, for which a Low Level Controller (LLC) based on LQR had been used. The author validated the functionality of the system by implementing both controllers on Sky-Sailor UAV platform.

Unlike [16] that uses Euler-Lagrange method, [17] uses Newton method to derive the dynamics. More precisely, the authors use Newtons second law, which basically states that "*the summation of all external forces, acting on a body, is equal to the time rate of change of the momentum of the body*" [9], in order to find the relationship between forces acting on the plane (composed of aerodynamic, propulsive and gravitational forces) and motion of the vehicles center of mass. Similar to [16], the UAV is considered to be a rigid body.

[18] and [19] tackle the problem of control system design for a flying wing. In [18] the authors suggest the use of the so-called Nonlinear Dynamic Inversion (NDI), which basically uses the concepts of feedback linearization. The aim in this technique is to cancel nonlinear terms of the system equation, by introducing the inverse of the term in the controller equation. It is understood that the ability to accurately model nonlinear terms determine the reliability of term cancellation. [18] proposes an outer loop controller designed based on NDI, which generates body-axis angular velocity commands ($x_f = [p, q, r]^T$), in order to run an inner loop controller, also designed based on NDI, creating commands for effector deflection. Finally, so-called Nonlinear Control Allocation (NCA) block was designed based on Sequential Quadratic Programming (SQP) to overcome the control redundancy problem, considering the assumption that the relationship between effector deflection and moments produced by the effector is nonlinear. Although details of system modeling are not expressed explicitly in [18] and [19], the method used for controller design appears to allow fulfilling the requirements.

In [20], an optimal controller was designed based on H_2 and H_∞ control synthesis approach for flying wing UAV equipped with elevon control surfaces. Based on the relationship between elevon deflection and altitude variations, transfer function of the system was obtained and then converted into a state-space description to be used in a

controller design process. Based on the state-space model, norms of the system were calculated (both H_2 and H_∞ norms) and the controller design problem was defined to stabilize the closed-loop system, while keeping both norms minimum, at all times during the flight.

Some literature such as [21] and [22] concentrate on HLC system design for trajectory tracking. Instead of considering the relationships between control surface(s) movements and the effect on aircraft position and orientation, which requires taking into account the dynamics of the system, they define the problem as a kinematics problem. For instance, in [21] the author assumes that the airspeed and heading commands can be used as control inputs, which basically means that the LLC should function to a satisfactory level.

1.4 Organization of Thesis

The thesis is organized as follows. Chapter 1 provides an overview of the thesis work, introducing the general concepts of the problem under study and motivations. Various scientific publications had also been reviewed in chapter 1 and helped the author to create well-structured understanding of the problem. Chapter 2 focuses on derivation of a suitable dynamic model of the UAV. Both non-linear and linear equations of motion are explained in chapter 2. Chapter 3 explains the procedure followed in order to design the controller. Chapter 4 explains simulation of the UAV model and controller in MATLAB/SIMULINK. Simulation results will be later covered in chapter 5 and finally, chapter 6 summarizes the work and discusses the implication of the research and points out the main contribution along with the description of possible future extensions.

1.4. ORGANIZATION OF THESIS

Table 1.2: Project Resources

Hu- man Re- sources	Role	Task
Dr. Leonid Frei-dovich	University Supervisor	Evaluation and approval of the thesis from scientific (academic) standpoint; Consultation during the project
Olle Hagner	Industrial Supervisor	Evaluation and approval of the thesis from scientific (industrial) standpoint; Consultation during the project
Dr. Sven Rönnbäck	Thesis Examiner	Final approval of the thesis; Evaluation and feedback
Shahriar Bagheri (conducting the thesis)	Researcher	Obtaining numeric values for stability derivatives in DATCOM and aerodynamic analysis of the plane; tuning matrices Q and R for the linear state-space model (Provided by Mr.Jafarov); Integrating LQR controller into the nonlinear model; Validating controller through simulations for various flight condition; writing thesis report
Tural Jafarov	Research Partner	Analytic calculations for stability derivatives; MATLAB coding of linearized state-space model; Simulating PID controller for signal tracking
Shahriar / Tural	Joint work	Designing and coding LQR controller in MATLAB for the longitudinal and lateral axis; Developing the nonlinear simulation model for the UAV in SIMULINK;
Computer Software	Purpose	Provided by
MAT-LAB/SIMULINK	used to simulate the system and analyze the results	Umeå University - Math-work Student License
DIGI-TAL/DATCOM	used to calculate stability and control derivatives of the UAV	Open source

Table 1.3: List of Expected Deliveries

No.	Item
1.	Written report (MSc thesis)
2.	Simulation files
3.	Documented source codes

Table 1.4: SmartFly Airframe

SmartFly Airframe	Values
Wing Span	81 cm
Wing Area	2712 cm ²
Empty Weight	568 g
Payload	200 g
Electric Motor	150 watt
Cruise Speed	11.5 m/s
Max Speed	25 m/s

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 2

Equations of Motion

The aim of this chapter is to develop a proper dynamic model of the UAV, using appropriate mathematical tools. Procedure followed in this chapter is based on the materials explained in chapters 3 and 4 of [9]. This chapter is organized as follows: first, explanation of reference frames and technical terms associated with aircraft configuration are given. After that, a non-linear dynamic model of the plane will be developed. The chapter continues by linearizing the non-linear model around an equilibrium point (trim condition) using the concept of "Small Disturbance Theory". Finally, State-Space representation for the linearized model will be introduced, which forms the basis for the next chapter.

2.1 Reference Frames

In order to proceed with aircraft dynamic modeling, one needs to appropriately define number of reference frames (also known as the axis systems). There are usually three reference frames used for aircraft modeling: **Body** frame, which is attached to the aircraft body (hence the name Body axis), **Earth** frame, which is fixed to the earth surface, and **Stability** frame, which is used as a reference for aerodynamic forces.

As a general convention, most often the so-called **North-East-Down** coordinate system (NED) is used for the **Earth** frames. NED is comprised of three vectors which together form a right hand orthogonal system. **N** represents the position along the

2.1. REFERENCE FRAMES

northern axis (X_e), **E** represents position along the east axis (Y_e) and **D** represent position along the local gravity vector (Z_e). **Body** axis system is also comprised of three orthogonal vectors. The origin of the body axis frame is located at the CG. X_b points forward towards aircraft's head, Y_b points to the right side (starboard) of the vehicle and Z_b points downward.

It should be mentioned that the center of the body axis system is assumed to coincide with the aircraft CG. This assumption would help simplify the equations of motion and will be explained thoroughly later in this chapter. Figure 2-1 illustrates body and inertial axis systems along with the associated Euler angles.

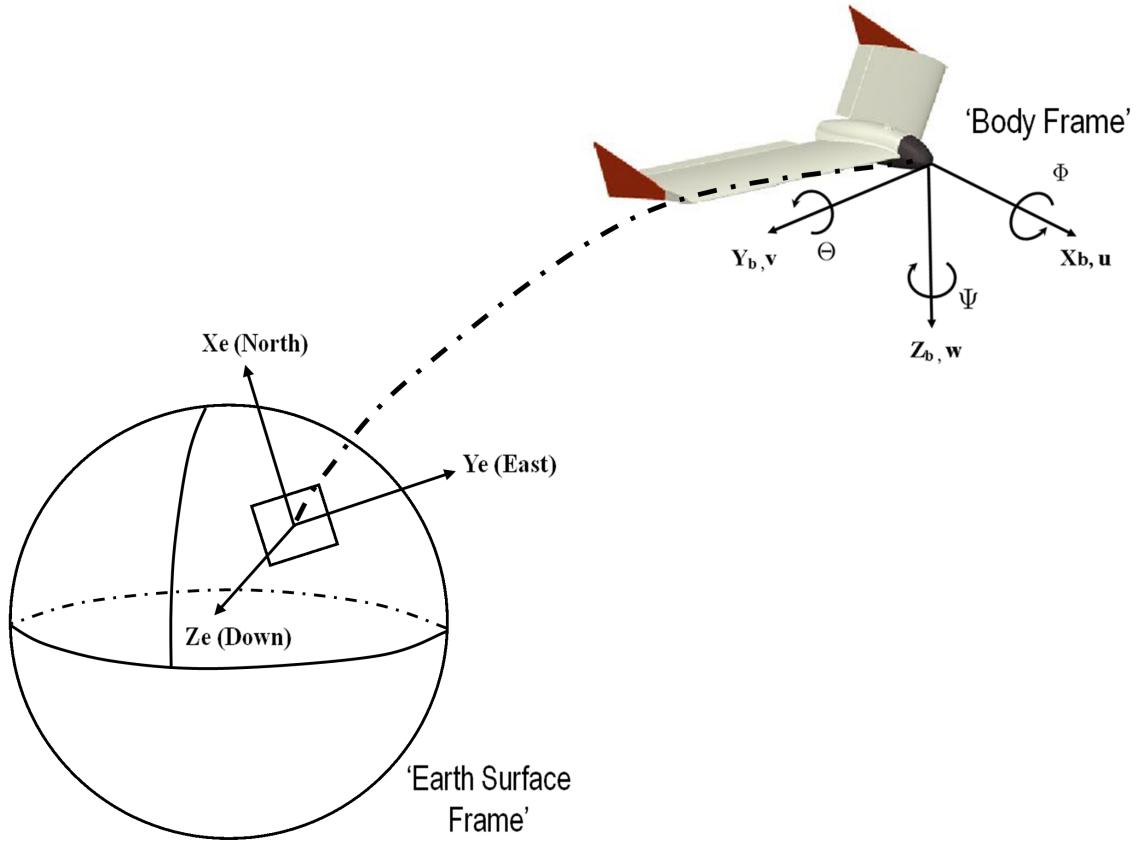


Figure 2-1: Earth surface and body reference frames

We can now calculate the so-called Angle of Attack (AoA) which is defined as the angle between the chord line of the wing (imaginary line that connects leading edge to the trailing edge of the wing) and relative velocity (from now on we denote it as airspeed V_{air}).

$$\alpha = \tan^{-1} \left(\frac{w}{u} \right) \quad (2.1)$$

$$V_{air} = \sqrt{u^2 + v^2 + w^2} \quad (2.2)$$

Body and stability axis systems coincide at AoA equal to zero ($\alpha = 0$). In many cases it is convenient to convert forces that are expressed in the stability axis, into body axis, in the beginning of all calculations. This is due to the fact that all calculations based on aerodynamic characteristics (i.e. aircraft velocity, forces and moments) are more understandable when expressed in body frame. Lift and Drag forces, which are originally expressed in stability axis, are therefore converted to body axis using equation 2.4 below. Moment equations are similar in both stability and body axis since both coordinate systems has their origin at CG. Side Force C_Y is also the same for both axis systems when β (Side Slip angle) is zero. Equation 2.3 shows how one can find β , and figure 2-2 illustrates lift, drag, normal and axial forces.

$$\beta = \tan^{-1} \left(\frac{v}{\sqrt{u^2 + w^2}} \right) \quad (2.3)$$

$$\begin{aligned} C_X &= C_L \sin(\alpha) - C_D \cos(\alpha) \\ C_Z &= -C_D \sin(\alpha) - C_L \cos(\alpha) \end{aligned} \quad (2.4)$$

In many practical applications, especially when it comes to aircraft navigation, it is required to have a systematic way of describing aircraft position and orientation with respect to a fixed coordinate system (Earth frame). Orientation of the aircraft can be fully described using Euler angles (ϕ , θ and ψ). Conversion between aircraft body axis and inertial axis can then be computed using the following rotation matrix:

$$R_b^e = \begin{bmatrix} C_\theta C_\psi & S_\phi S_\theta C_\psi - C_\phi S_\psi & C_\phi S_\theta C_\psi + S_\phi S_\psi \\ C_\theta S_\psi & S_\phi S_\theta S_\psi + C_\phi C_\psi & C_\phi S_\theta S_\psi - S_\phi C_\psi \\ -S_\theta & S_\theta C_\theta & C_\phi C_\theta \end{bmatrix} \quad (2.5)$$

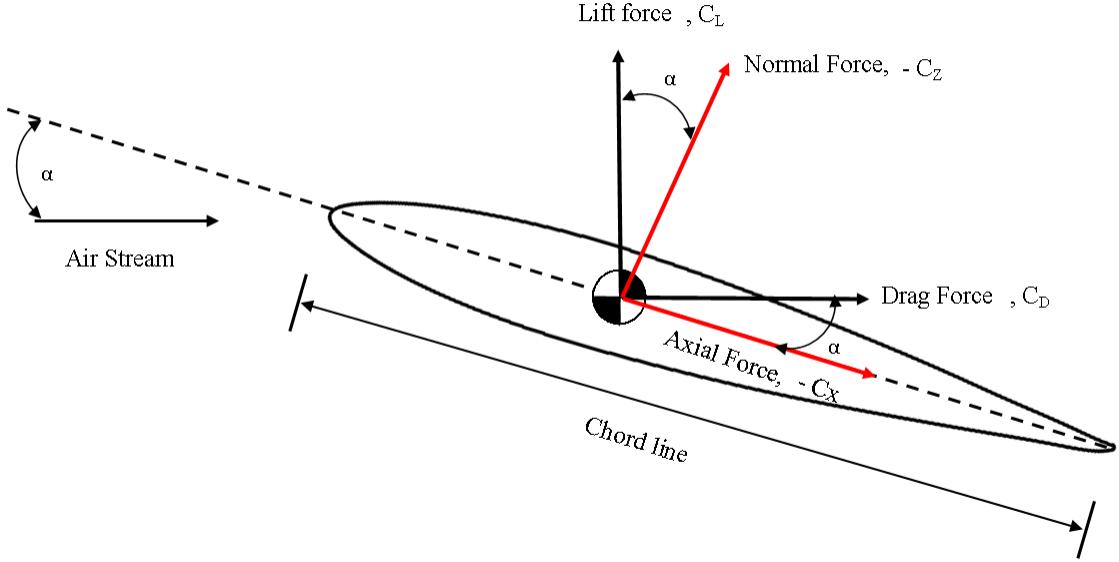


Figure 2-2: Forces in stability and body axis

One can refer to [9, p. 101,102] for details on how to derive this matrix.

Equation 2.6 below can be used to find Euler angles, that are required for frame conversion. This equation represents the relationship between aircraft angular velocities in body frame (p, q, r) and rate of change of Euler angles ($\dot{\phi}, \dot{\theta}, \dot{\psi}$). So Euler angles can simply be found by integrating the following equation:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & S_\phi \tan(\theta) & C_\phi \tan(\theta) \\ 0 & C_\phi & -S_\phi \\ 0 & S_\phi \sec(\theta) & C_\phi \sec(\theta) \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.6)$$

For example if we are interested in aircraft velocity in inertial frame, we can use the Euler angles, and equation 2.5, as shown blow:

$$\begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \\ \frac{dz}{dt} \end{bmatrix} = \begin{bmatrix} C_\theta C_\psi & S_\phi S_\theta C_\psi - C_\phi S_\psi & C_\phi S_\theta C_\psi + S_\phi S_\psi \\ C_\theta S_\psi & S_\phi S_\theta S_\psi + C_\phi C_\psi & C_\phi S_\theta S_\psi - S_\phi C_\psi \\ -S_\theta & S_\theta C_\theta & C_\phi C_\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.7)$$

2.2 Non-linear Equations of Motion

There are two commonly used techniques for deriving equations of motion in flying vehicles: Euler-Lagrange method and Newton's approach. Dynamic models used in this research are based on Newton's method as described in Chapter 3 of [9]. Before proceeding to equations, following points should be considered. In an aircraft with low speed range (as the one under study here) compressibility and elasticity effects can be ignored and the vehicle can be thought of as a rigid body with 6 DoF. Also, since the plane used in this experiment uses an electrically driven engine, mass of the plane is constant during flight. These two assumptions would allow us to simplify equations of motion.

Newton's Second Law of motion is the heart of equations of motion derivation. These equations explain relationships between forces in the body frame (F_X, F_Y, F_Z), moments (L, M, N) and aircraft linear (u, v, w) and angular (p, q, r) velocities ¹.

$$\begin{aligned} F_X &= m(\dot{u} + qw - rv) \\ F_Y &= m(\dot{v} + ru - pw) \\ F_Z &= m(\dot{w} + pv - qu) \end{aligned} \tag{2.8}$$

$$\begin{aligned} L &= I_x \dot{p} - I_{xz} \dot{r} + qr(I_z - I_y) - I_{xz} pq \\ M &= I_y \dot{q} + rp(I_x - I_z) + I_{xz}(p^2 - r^2) \\ N &= -I_{xz} \dot{p} + I_z \dot{r} + pq(I_y - I_x) + I_{xz} qr \end{aligned} \tag{2.9}$$

where m is the UAV mass, g is the gravitational acceleration and is $\approx 9.8 \text{ m/s}^2$.

Force and moment components can be further broken down into three sub-components of thrust (created by aircraft engine), gravitational (due to the earth gravity) and aerodynamic (produced due to the governing rules of aerodynamics). When expressed in the body frame, the gravitational force is a function of aircraft orientation in space. More precisely, force created due to gravity depends on pitch (θ) and roll (ϕ) angles. Since gravity acts through the UAV CG, it creates no moment. Force components

¹It is assumed that xz plane is the plane of symmetry of the aircraft.

due to gravity expressed in body frame, can be computed as:

$$\begin{aligned}(F_X)_{\text{gravity}} &= -mg \sin(\theta) \\ (F_Y)_{\text{gravity}} &= mg \cos(\theta) \sin(\phi) \\ (F_Z)_{\text{gravity}} &= mg \cos(\theta) \cos(\phi)\end{aligned}\tag{2.10}$$

θ and ϕ are aircraft pitch and roll angles, respectively.

Force due to aerodynamic is a function of various variables. They will be fully explained later in this chapter. But for now, let's simply assume that the sum of aerodynamic and thrust forces are X , Y and Z . Replacing equation 2.10 and new definitions of force due to aerodynamic and thrust back to equation 2.8, we get ²

$$\begin{aligned}X - mg \sin(\theta) &= m(\dot{u} + qw - rv) \\ Y + mg \cos(\theta) \sin(\phi) &= m(\dot{v} + ru - pw) \\ Z + mg \cos(\theta) \cos(\phi) &= m(\dot{w} + pv - qw)\end{aligned}\tag{2.11}$$

So equations 2.6, 2.9 and 2.11 represent the dynamics and kinematics of the UAV.

2.3 Linearized Equations of Motion

2.3.1 Small-Disturbance theory

The main idea in small-disturbance theory is to linearize the non-linear model developed in section 2.2, around a steady-state condition (trimmed flight) [9]. Assuming that deviations from this equilibrium state are small, a linearized model is expected to provide useful and fairly accurate representation of the non-linear system.

In this theory, each variable in the model of equations 2.9 and 2.11 is assumed to have a nominal value (at trimmed flight, indexed 0), plus a disturbance value (for instance, u_0 is the nominal value for linear velocity along body X_b axis and Δu is the small perturbation. So $u = u_0 + \Delta u$). Table 2.1 tabulates trim condition values. Using the same logic, one can calculate small changes in each force and moment

²Summary of kinematic and dynamic equations are given in chapter 3 of [9]

element ($\Delta X, \Delta Y, \Delta Z, \Delta L, \Delta M, \Delta N$) as a result of change in a particular variable (u, w, δ, \dots).

Table 2.1: Trim Condition

Variable	Value
u_0	9.7 m/s
v_0	0 m/s
w_0	1.2 m/s
p_0	0 m/s
q_0	0 m/s
r_0	0 m/s
α_0	6.9°
β_0	0°
V_{air}	9.7739 m/s
h_0	150 m
ϕ_0	0°
θ_0	6.9°
ψ_0	0°

As an example, let us consider \mathbf{X} (the force along the body X_b axis) as described in equation 2.11. By introducing the small-disturbance notation into this equation and simplifying the result as described in chapter 3 of [9], following equation will be obtained ³:

$$\Delta X - mg \Delta\theta \cos \theta_0 = m \Delta\dot{u} + mw_0\Delta q \quad (2.12)$$

Where ΔX is the change in the force along the X_b direction (contribution of aerodynamic force and thrust). It is possible to express ΔX in terms of perturbation variables using Taylor series [9]. Assuming that ΔX is dependent on $\Delta u, \Delta w, \Delta \delta_e$ and $\Delta \delta_T$, an expression for ΔX can be obtained as:

$$\Delta X = \frac{\partial X}{\partial u} \Delta u + \frac{\partial X}{\partial w} \Delta w + \frac{\partial X}{\partial \delta_e} \Delta \delta_e + \frac{\partial X}{\partial \delta_T} \Delta \delta_T \quad (2.13)$$

³One can align the x-axis, to be along the direction of the airplane's velocity vector. This would result in $w_0 = 0$ and hence the term " $mw_0\Delta q$ " will also be zero.[9]

The partial derivatives of the above equation are known as the **Stability Derivatives**.

Now, one can substitute back equation 2.13 into the force equation 2.12 to get:

$$\left(\frac{d}{dt} - X_u \right) \Delta u - X_w \Delta w + (g \cos \theta_0) \Delta \theta = X_{\delta_e} \Delta \delta_e + X_{\delta_T} \Delta \delta_T + w_0 \Delta q \quad (2.14)$$

Note that new notations X_u, X_w are just partial derivative of equation 2.13 divided by the aircraft mass ($X_u = \partial X / \partial u / m$, $X_w = \partial X / \partial w / m$ and so on).

Let us now summarize the process which was just explained for linearizing the **X** force equation, through a step-by-step procedure. This procedure can be used to linearize force and moment equations in all axes and, is as follows:

1. Replace all variables in the non-linear model of equation 2.9 and 2.11 by the notations introduced in small-disturbance theory. That is, a nominal value plus disturbance (or perturbation) value;
2. Expand the resulting equation through Taylor series considering only variables with significant influence on force or moment equation in a particular direction (The effect of some variables on force and moment in a particular direction is negligible, for instance force along X_b is hardly dependent on v , the velocity along body Y_b axis, and hence can be neglected)[23];
3. Substitute back the result of Taylor expansion into the equation obtained in item 1;
4. Divide the stability derivatives (terms with partial derivatives) with mass (for force equations) and inertia value (for moment equations) to obtain final form of the stability derivatives.

2.3.2 Calculation of Stability Derivatives

Longitudinal Axis

let us go back to the non-linear equation (2.11) one more time and consider the **X** force equation. It can be rewritten as:

$$\dot{u} = rv - qw - g \sin \theta + \frac{X}{m} \quad (2.15)$$

where X represents the combination of aerodynamic and propulsive forces (i.e. $X = F_{X_{aero}} + F_{X_{prop}}$). According to [26], assuming that $\phi = p = r = \beta = v = 0$, F_{aero} and F_{prop} can be computed as:

$$\begin{aligned} F_{X_{aero}} &= \frac{\rho(u^2 + w^2)S}{m} [C_{x_0} + C_{x_\alpha} \alpha + C_{x_{\delta_e}} \delta_e] + \frac{\rho\sqrt{u^2 + w^2}S}{m} C_{X_q} \bar{c}q \\ F_{X_{prop}} &= \frac{1}{2}\rho S_{prop}((K_{motor} \delta_T)^2 - V_{air}^2) \end{aligned} \quad (2.16)$$

where ρ is the air density, u and w are velocities along body x-axis and z-axis, respectively, C_{x_0} is the axial force coefficient, C_{x_α} is the derivative of the axial force with respect to α , $C_{x_{\delta_e}}$ is the derivative of the axial force with respect to δ_e and \bar{c} is the wing chord. $F_{X_{prop}}$ will be explained in section 4.1.2.

Substituting back equation 2.16 into 2.15 with the assumption that $\phi = p = r = \beta = v = 0$, we will get:

$$\begin{aligned} \dot{u} &= -qw - g \sin \theta + \frac{\rho(u^2 + w^2)S}{2m} [C_{x_0} + C_{x_\alpha} \alpha + C_{x_{\delta_e}} \delta_e] + \frac{\rho\sqrt{u^2 + w^2}S}{m} C_{X_q} \bar{c}q \\ &+ \frac{1}{2m}\rho S_{prop}((K_{motor} \delta_T)^2 - V_{air}^2) \end{aligned} \quad (2.17)$$

This procedure should be repeated for all non-linear forces and moments. The result (longitudinal terms) would be as follows ⁴:

⁴For further clarification please refer to [26] chapter 7, section 7.2.2

$$\begin{aligned}
 \dot{u} &= -qw - g \sin \theta + \frac{\rho(u^2 + w^2)S}{2m} [C_{x_0} + C_{x_\alpha} \alpha + C_{x_{\delta_e}} \delta_e] + \frac{\rho\sqrt{u^2 + w^2}S}{2m} C_{X_q} \bar{c}q \\
 &\quad + \frac{1}{2m} \rho S_{prop} ((K_{motor} \delta_T)^2 - V_a^2) \\
 \dot{w} &= qu + g \cos \theta \cos \phi + \frac{\rho(u^2 + w^2)S}{2m} [C_{Z_0} + C_{Z_\alpha} \alpha + C_{Z_{\delta_e}} \delta_e] + \frac{\rho\sqrt{u^2 + w^2}S}{2m} C_{Z_q} \bar{c}q \\
 \dot{q} &= \frac{1}{2I_y} \rho(u^2 + w^2) \bar{c}S [C_{m_0} + C_{m_\alpha} \alpha + C_{m_{\delta_e}} \delta_e] + \frac{1}{2I_y} \rho\sqrt{u^2 + w^2} \bar{c}S C_{m_q} \bar{c}q \\
 \dot{\theta} &= q \cos \phi - r \sin \phi = q
 \end{aligned} \tag{2.18}$$

We can now apply the Small-Disturbance theory explained in section 2.3.1 to the above equation. More precisely one should replace all variable in the equation 2.17 with a nominal value plus a disturbance value, and take linear approximations of the resulting equation, using Jacobians:

$$\begin{aligned}
 \frac{\partial f}{\partial x} &= \begin{bmatrix} \frac{\partial \dot{u}}{\partial u} & \frac{\partial \dot{u}}{\partial w} & \frac{\partial \dot{u}}{\partial q} & \frac{\partial \dot{u}}{\partial \theta} \\ \frac{\partial \dot{w}}{\partial u} & \frac{\partial \dot{w}}{\partial w} & \frac{\partial \dot{w}}{\partial q} & \frac{\partial \dot{w}}{\partial \theta} \\ \frac{\partial \dot{q}}{\partial u} & \frac{\partial \dot{q}}{\partial w} & \frac{\partial \dot{q}}{\partial q} & \frac{\partial \dot{q}}{\partial \theta} \\ \frac{\partial \dot{\theta}}{\partial u} & \frac{\partial \dot{\theta}}{\partial w} & \frac{\partial \dot{\theta}}{\partial q} & \frac{\partial \dot{\theta}}{\partial \theta} \end{bmatrix} \\
 \frac{\partial f}{\partial u_c} &= \begin{bmatrix} \frac{\partial \dot{u}}{\partial \delta_e} & \frac{\partial \dot{u}}{\partial \delta_T} \\ \frac{\partial \dot{w}}{\partial \delta_e} & \frac{\partial \dot{w}}{\partial \delta_T} \\ \frac{\partial \dot{q}}{\partial \delta_e} & \frac{\partial \dot{q}}{\partial \delta_T} \\ \frac{\partial \dot{\theta}}{\partial \delta_e} & \frac{\partial \dot{\theta}}{\partial \delta_T} \end{bmatrix}
 \end{aligned} \tag{2.19}$$

Note that u_c is a vector of control signals ($u_c = [\delta_e \ \delta_T]^T$), but u is a scalar and represents linear velocity along X_b . Finally, longitudinal linearized model will be:

$$\begin{aligned}
 \Delta \dot{u} &= X_u \Delta u + X_w \Delta w + X_q \Delta q - g \cos \theta \Delta \theta + X_{\delta_e} \Delta \delta_e + X_{\delta_T} \Delta \delta_T + w_0 \Delta q \\
 \Delta \dot{w} &= Z_u \Delta u + Z_w \Delta w + Z_q \Delta q - g \sin \theta_0 \Delta \theta + Z_{\delta_e} \Delta \delta_e - w_0 \Delta q \\
 \Delta \dot{q} &= M_u \Delta u + M_w \Delta w + M_q \Delta q + M_{\delta_e} \Delta \delta_e \\
 \Delta \dot{\theta} &= \Delta q
 \end{aligned} \tag{2.20}$$

Lateral Axis

Lateral axis equations will be obtained in the same way as the longitudinal axis. Formulas for calculating lateral stability derivatives are tabulated in table 2.2.

$$\begin{aligned}
 \Delta\dot{\beta} &= \frac{Y_\beta}{u_0}\Delta\beta + \frac{Y_p}{u_0}\Delta p - (1 - \frac{Y_r}{u_0})\Delta r + g \cos \theta_0 \Delta\phi \\
 \Delta\dot{p} &= L_\beta\Delta\beta + L_p\Delta p + L_r\Delta r + L_{\delta_a}\Delta\delta_a \\
 \Delta\dot{r} &= N_\beta\Delta\beta + N_p\Delta p + N_r\Delta r + N_{\delta_a}\Delta\delta_a \\
 \Delta\dot{\phi} &= \Delta p + (\cos \phi_0 \tan \theta_0)\Delta r + (q_0 \cos \phi_0 \tan \theta_0 - r_0 \sin \phi_0 \tan \theta_0)\Delta\phi
 \end{aligned} \tag{2.21}$$

Notice that it is more convenient to use $\Delta\beta$ angle instead of side velocity Δv ($\Delta\beta \approx \frac{\Delta v}{u_0}$). Also, according to the trim condition of table 2.1, $\phi_0 = q_0 = r_0 = 0$, therefore $\Delta\dot{\phi} \approx \Delta p + 0.121\Delta r$.

Table 2.2: Lateral Stability Derivatives

Lateral Derivatives	Formula
Y_β	$\frac{QS}{m}C_{Y_b}$
Y_p	$\frac{QSb}{2u_0m}C_{Y_p}$
Y_r	$\frac{QSb}{2u_0m}C_{Y_r}$
Y_{δ_a}	$\frac{QS}{m}C_{Y_{\delta_a}}$
N_β	$\frac{QSb}{I_z}C_{N_b}$
N_p	$\frac{QSb^2}{2u_0I_x}C_{N_p}$
N_r	$\frac{QSb^2}{2u_0I_x}C_{N_r}$
N_{δ_a}	$\frac{QSb}{I_z}C_{N_{\delta_a}}$
L_β	$\frac{QSb}{I_x}C_{l_b}$
L_p	$\frac{QSb^2}{2u_0I_x}C_{l_p}$
L_r	$\frac{QSb^2}{2u_0I_x}C_{l_r}$
L_{δ_a}	$\frac{QSb}{I_x}C_{l_{\delta_a}}$

In which $Q = \frac{1}{2}\rho V_{air}^2$ is the dynamic pressure, m is the aircraft mass in kg , S is

2.3. LINEARIZED EQUATIONS OF MOTION

the wing Area, b is the wing span.

Table 2.3: Numeric Values for Lateral Stability Derivatives

Vari- able	Y-Force Derivative	L-Moment Derivative	N-Moment Derivative
β	$Y_\beta = -2.6648$	$L_\beta = -11.1699$	$N_\beta = 0.0664$
p	$Y_p = 0.0730$	$L_p = -1.0893$	$N_p = 0.3634$
r	$Y_r = 0$	$L_r = 0.2319$	$N_r = -0.0655$
δ_a	$Y_{\delta_a} = 0$	$L_{\delta_a} = 15.830$	$N_{\delta_a} = -0.7353$

Stability Derivatives obtained above are functions of static and dynamic **stability coefficients**. These coefficients depend on the aerodynamic characteristic of a particular UAV. Among various methods available for finding aerodynamic coefficients, computer software was a reasonable choice for this research, taking into account affordability and accessibility. One of the most widely used computer programs for airplane stability and control analysis is the USAF DIGITAL DATCOM [13].

DIGITAL DATCOM provides aerodynamic analysis for fixed-wing airplanes. The output, among many other useful information, particularly contains static and dynamic stability and control derivatives. The input file for the program should contain geometric information of the plane (that includes body shape and dimension, airfoil data for wings and etc.), flight condition (i.e. velocity and Mach number, altitude) and control surfaces movements (for dynamic analysis). The so-called "DATCOM input card" defined specifically for this research is available in Appendix A. Stability and control coefficients obtained from DATCOM are tabulated in table 2.4. Some of these values are also used to simulate the non-linear UAV model in SIMULINK. Simulation procedure will be explained in Chapter 4.

One would realize that the numeric values for the lift force coefficient C_L and drag force coefficient C_D are expressed in the aircraft stability axis. It would be convenient to convert them to body axis prior to calculation. In order to do so, equation 2-1 is used to convert lift and drag force to normal and axial force (C_Z and C_X).

Relationship between stability coefficients and stability derivatives introduced in section 2.3.1, are tabulated in tables 2.5 and 2.2. Formulas of this table are pre-

Table 2.4: Aerodynamic Coefficients and Derivatives

Longitudinal Coefficients	Value	Lateral Coefficients	Value
C_{L_0}	0.2615	C_{Y_0}	0
C_{D_0}	0.029	C_{l_0}	0
C_{M_0}	0	C_{n_0}	0
C_{L_α}	3.4424	C_{Y_β}	-0.06972
C_{D_α}	0.8477	C_{l_β}	-0.093
C_{M_α}	-0.4438	C_{n_β}	0.001002
$C_{L_{\delta_e}}$	0.7450	$C_{Y_{\delta_a}}$	0
$C_{D_{\delta_e}}$	0	$C_{l_{\delta_a}}$	0.1318
$C_{M_{\delta_e}}$	-0.3438	$C_{n_{\delta_a}}$	-0.0069
		C_{Y_p}	0.0539
		C_{l_p}	-0.256
		C_{n_p}	0.0854

sented without proof. One can refer to chapter 3 of [9] and chapter 6 of [26] for detailed mathematical explanation. These calculations were carried out in MATLAB (Appendix C).

Above equations can be converted to state-space representation. This will be explained in the following section.

2.3.3 State-Space Representation of the Linearized Model

Before continuing, it might be helpful to summarize equations of motion, and present an intuitive and understandable representation of the linearized equations developed in section 2.3.1.

Non-linear equations of motion which had been derived based on Newton's second law in section 2.2, are difficult to be used for control system design purpose: apart from the fact that they are highly non-linear, terms describing longitudinal and lateral movement of the system are coupled. Therefore, using small-disturbance theory, linearized form of dynamic equations had been calculated (2.20 and 2.21). These linearized equations are based on the stability derivatives of section 2.3.2.

So, now that we have the decoupled, simple linear ordinary differential equations

Table 2.5: Longitudinal Stability Derivatives

Longitudinal Derivatives	Formula
X_u	$\frac{u\rho S}{m} [C_{x_0} + C_{x_\alpha} \alpha + C_{x_{\delta_e}} \delta_e] - \frac{\rho S w C_{x_\alpha}}{2m} - \frac{\rho S_{prop} C_{prop} u}{m}$
X_w	$\frac{w\rho S}{m} [C_{x_0} + C_{x_\alpha} \alpha + C_{x_{\delta_e}} \delta_e] - \frac{\rho S u C_{x_\alpha}}{2m} - \frac{\rho S_{prop} C_{prop} u}{m}$
X_q	$-w + \frac{\rho V_a S C_{X_q} c}{2m}$
X_{δ_e}	$\frac{\rho V_{air}^2 S C_{X_{\delta_e}}}{2m}$
X_{δ_T}	$\frac{\rho S_{prop} C_{prop} K \delta_T}{m}$
Z_u	$\frac{u\rho S}{m} [C_{Z_0} + C_{Z_\alpha} \alpha + C_{Z_{\delta_e}} \delta_e] - \frac{\rho S w C_{Z_\alpha}}{2m}$
Z_w	$\frac{w\rho S}{m} [C_{Z_0} + C_{Z_\alpha} \alpha + C_{Z_{\delta_e}} \delta_e] - \frac{\rho S u C_{Z_\alpha}}{2m}$
Z_q	$u + \frac{\rho V_a S C_{Z_q} C}{2m}$
Z_{δ_e}	$\frac{\rho V_{air}^2 S C_{Z_{\delta_e}}}{2m}$
M_u	$\frac{u\rho Sc}{J_y} [C_{M_0} + C_{M_\alpha} \alpha + C_{M_{\delta_e}} \delta_e] - \frac{\rho Sc C_{M_\alpha} w}{2J_y}$
M_w	$\frac{w\rho Sc}{J_y} [C_{M_0} + C_{M_\alpha} \alpha + C_{M_{\delta_e}} \delta_e] - \frac{\rho Sc C_{M_\alpha} u}{2J_y}$
M_q	$\frac{\rho V_{air} S C^2 C_{M_q}}{2J_y}$
M_{δ_e}	$\frac{\rho V_{air}^2 S c C_{M_{\delta_e}}}{2J_y}$

with constant coefficients (for small deviations from the trimmed flight condition) it is possible to write them as a set of first-order differential equations in the form of a **State-Space Model** [9]. Three equations describing UAV longitudinal movement (equation 2.20) can now be written in the form of (2.22).

$$\begin{bmatrix} \Delta \dot{u} \\ \Delta \dot{w} \\ \Delta \dot{q} \\ \Delta \dot{\theta} \end{bmatrix} = \begin{bmatrix} X_u & X_w & X_q + w_0 & -g \cos \theta_0 \\ Z_u & Z_w & Z_q - w_0 & -g \sin \theta_0 \\ M_u & M_w & M_q & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta w \\ \Delta q \\ \Delta \theta \end{bmatrix} + \begin{bmatrix} X_{\delta_e} & X_{\delta_T} \\ Z_{\delta_e} & 0 \\ M_{\delta_e} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta \delta_e \\ \Delta \delta_T \end{bmatrix} \quad (2.22)$$

In a similar fashion, State-Space representation for linearized lateral equations is given in (2.23).

Table 2.6: Numeric Values for Longitudinal Stability Derivatives

Vari- able	X-Force Derivative	Z-Force Derivative	M-Moment Derivative
u	$X_u = -0.0543$	$Z_u = -2.7791$	$M_u = -0.3403$
w	$X_w = -0.5332$	$Z_w = -10.3435$	$M_w = -2.0302$
δ_e	$X_{\delta_e} = 2.4224$	$Z_{\delta_e} = -20.2054$	$M_{\delta_e} = -18.4384$
δ_T	$X_{\delta_T} = 0.0224$	$Z_{\delta_T} = -0.0020$	$M_{\delta_T} = 0$

$$\begin{bmatrix} \Delta\dot{\beta} \\ \Delta\dot{p} \\ \Delta\dot{r} \\ \Delta\dot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{Y_\beta}{u_0} & \frac{Y_p}{u_0} & -\left(1 - \frac{Y_r}{u_0}\right) & \frac{g\cos\theta_0}{u_0} \\ L_\beta & L_p & L_r & 0 \\ N_\beta & N_p & N_r & 0 \\ 0 & 1 & \tan\theta_0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\beta \\ \Delta p \\ \Delta r \\ \Delta\phi \end{bmatrix} + \begin{bmatrix} 0 \\ L_{\delta_a} \\ N_{\delta_a} \\ 0 \end{bmatrix} \begin{bmatrix} \Delta\delta_a \end{bmatrix} \quad (2.23)$$

Where the zero index denote the trim flight condition (θ_0 is the pitch angle for trim condition, u_0 is the linear velocity along body X_b axis etc.). All the notation used are available in the "LIST OF SYMBOLS" section in the beginning of the thesis. What we had obtained in equations 2.22 and 2.23 gives us the relationship between the state vector and control vector ($\dot{X} = AX + Bu$, $y = CX + Du$)⁵. By substituting numerical values of the stability derivatives of section 2.3.2 into matrices **A** and **B** of equations 2.22 and 2.23 following state, and input matrices had been obtained (See Appendix C, where all the used numerical values are given). Also, assuming that we can measure all states, **C** matrix is identity for both longitudinal and lateral axes. **D** matrix is assumed to be zero.

⁵Matrix 'A' is known as the state (or system) matrix, and matrix 'B' is known as the control matrix.

$$A_{long} = \begin{bmatrix} -0.0543 & -0.5332 & 0 & -9.7295 \\ -2.7791 & -10.3435 & 8.5100 & -1.1732 \\ -0.3403 & -2.0302 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad B_{long} = \begin{bmatrix} 2.4224 & 0.0224 \\ -20.2054 & 0 \\ -18.4384 & 0 \\ 0 & 0 \end{bmatrix} \quad (2.24)$$

$$C_{long} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad D_{long} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (2.25)$$

$$A_{lat} = \begin{bmatrix} -0.2331 & 0.0730 & -11.4320 & 9.7627 \\ -0.9771 & -1.0893 & 0.2319 & 0 \\ 0.0058 & 0.3634 & -0.0655 & 0 \\ 0 & 1 & 0.121 & 0 \end{bmatrix}, \quad B_{lat} = \begin{bmatrix} 0 \\ 15.8300 \\ -0.4571 \\ 0 \end{bmatrix} \quad (2.26)$$

$$C_{lat} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad D_{lat} = 0 \quad (2.27)$$

Chapter 3

Control System Design

Inherent instability and non-linearity associated with flying wings add to the difficulty of designing a robust controller for this type of UAV. Linearizing the model would simplify the task of controller design and enables the control engineer to benefit from the tools of linear system theory. Nonetheless, the downside of using linear model and linear control theory is that the performance can not be guaranteed for all the flight regime. However, in the case of this experiment, since the UAV flight profile is somewhat limited (speed range, service ceiling, weight and etc.), linear controller might be a reasonable solution.

Therefore, linear control strategies were used to control both longitudinal and lateral movements of the plane. Specifically, Linear Quadratic Regulator (LQR) method along with PID controller were used to stabilize the system and provide reference tracking for height, velocity and roll angle. The main idea behind using two controllers in sequence for a single system is that LQR, as an optimal state-feedback controller, stabilizes the system by appropriately positioning poles of the closed loop system. This would make the system easier to control with PID. Afterwards, PID controller could be designed for the stable system to track reference input signals. Block diagram shown in figure 3-1 illustrates the concept of the proposed architecture.

Focus of this chapter is on designing LQR controller. The chapter begins by introducing notions of **Controllability** and **Observability**, which are two of the important properties of linear control systems. Explanation of theory and mathe-

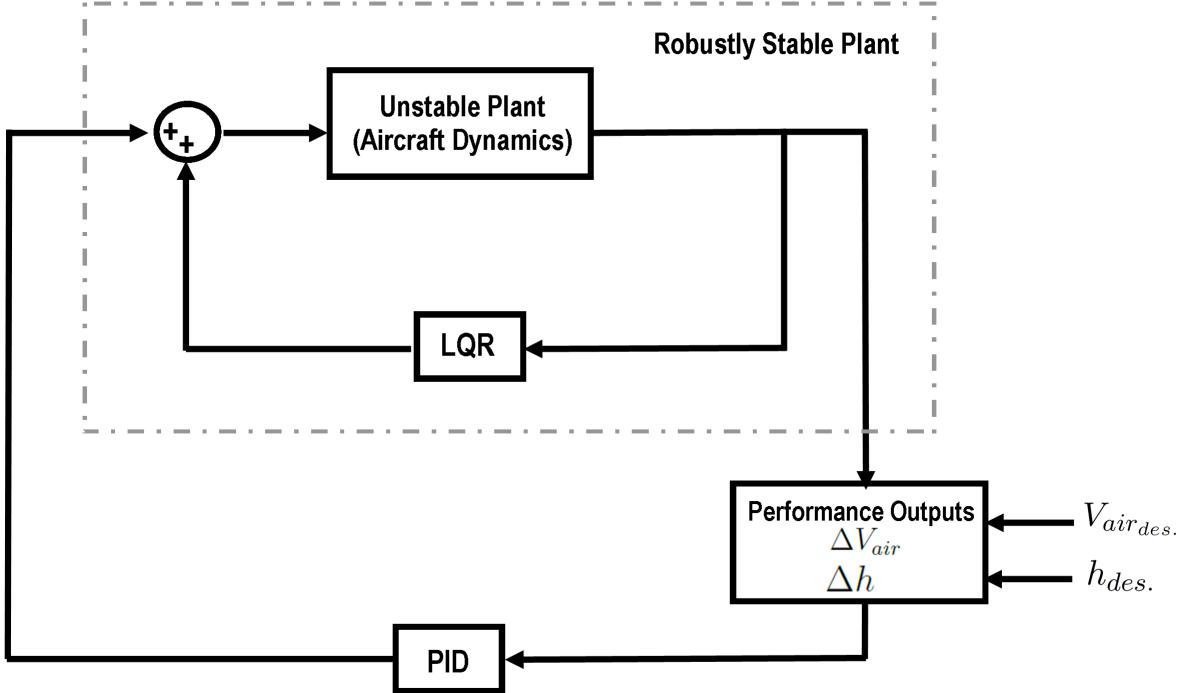


Figure 3-1: Block diagram of the proposed controller for longitudinal controller

matics of LQR is also covered in this chapter. Furthermore, logic behind specific choice of weighting matrices \mathbf{Q} and \mathbf{R} is presented. Final section briefly explains PID controllers and specific selection of gains for our work.

3.1 Controllability and Observability

In the previous chapter, more precisely section 2.3.3 we agreed to define the dynamics of our UAV system using State-Space representation of the following form:

$$\begin{aligned} \frac{d}{dt}x(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{3.1}$$

In which $x(t) \in \mathbb{R}^n$ is the so called **state** of the system, matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is the system matrix, $u(t) \in \mathbb{R}^m$ is the input to the system, $\mathbf{B} \in \mathbb{R}^{n \times m}$ is the input matrix, $y(t) \in \mathbb{R}^p$ is the system output, $\mathbf{C} \in \mathbb{R}^{p \times n}$ is the output matrix and finally $\mathbf{D} \in \mathbb{R}^{p \times m}$ is called the feedthrough matrix. [27]

An advantages of writing the system in State-Space format, is the possibility to easily determine some important characteristics of the system. Two of these key characteristics are **Controllability** and **Observability**. Controllability determines whether it is possible to make system states ' $x(t)$ ' evolve in time the way we want, using input ' $u(t)$ '. To check whether particular system is controllable, matrix 3.2 could be used. If this matrix is full rank, then the system is said to be controllable.

$$\mathcal{C} = [B, AB, A^2B, \dots, A^{n-1}B] \quad (3.2)$$

Observability on the other hand, is to see if it is possible to determine all the states ' $x(t)$ ' from output ' y ' and input signal ' u '. The system is said to be observable if the matrix 3.3 is full rank. These matrices are presented here without proof (For further clarification refer to chapter 1 of [27] or chapter 3 of [28]).

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \quad (3.3)$$

MATLAB functions `ctrb(A, B)` and `obsv(A, C)` computes controllability and observability matrices, respectively. In the case of this experiment, both longitudinal and lateral axes (2.27 and 2.26) are controllable and observable.

Above two definition give rise to the following argument: when the system is controllable then it is possible to place the closed-loop system eigenvalues in a desired location in order to achieve certain closed-loop poles, which define performance measures (settling time, overshoot level and etc.). This can be done using feedback gain. Although this statement is theoretically correct, it would not be so useful for practical applications. This is because achieving a certain performance characteristic in a controllable system, although theoretically possible, might require a control signal that is too large for a physical actuator to produce. Such restrictions in which certain choice of the feedback gain would not be feasible from implementation point

of view require a more systematic method of calculating feedback gain. **LQR** design introduces a method to partially overcome such problems by finding a trade-off between performance and control signal magnitude. The theory of such design will be explained in the following section.

3.2 Linear Quadratic Regulator

3.2.1 Theory

The Linear Quadratic Regulator (LQR) is a method that generates static feedback gain matrix ' \mathbf{K} '. The closed-loop system dynamics formed with ' \mathbf{K} ' is given by:

$$\dot{x}(t) = (A - BK)x(t) \quad (3.4)$$

The feedback gain calculated using LQR, is supposed to be optimal for a performance output (sometimes denoted as ' \mathbf{z} ') in a sense that it would minimize the energy. Consider the following schematic diagram, commonly referred to as the **Generalized Plant**. [27]

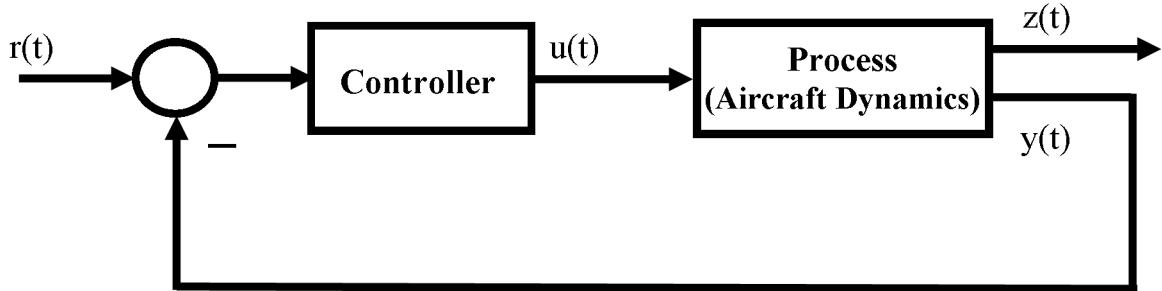


Figure 3-2: Generalized Plant

In this diagram, the *process* block illustrates the system to be controlled. The state-space model of the process in the generalized plant of figure 3-2 is as follows:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= C_1x(t) \\ z(t) &= \begin{bmatrix} C_2 & x(t) \\ D & u(t) \end{bmatrix}\end{aligned}\tag{3.5}$$

$'y(t)'$ denotes the *measurable output* of the system and $'z(t)'$ is the so-called *controlled output* that specifies some sort of performance index. Here, in the case of the longitudinal axis, measurable outputs are Δu , Δw , Δq and $\Delta\theta$, so C_1 is the identity matrix. However, controlled outputs are airspeed ΔV_{air} and aircraft's height Δh . They can be calculated from the measurable outputs (states of the longitudinal axis), using following equations:

$$\begin{aligned}V_{air} &= \sqrt{u^2 + v^2 + w^2}, \quad Z = \begin{bmatrix} \Delta h \\ \Delta V_{air} \end{bmatrix} \\ \Delta V_{air} &\approx \cos \theta_0 \Delta u + \sin \theta_0 \Delta w \\ \Delta h &= \sin \theta_0 \Delta u - \cos \theta_0 \Delta w + (u_0 \cos \theta_0 + w_0 \sin \theta_0) \Delta \theta\end{aligned}\tag{3.6}$$

Matrix C_2 can then be found for the trim flight condition of table 2.1, using equation 3.6:

$$C_2 = \begin{bmatrix} -0.1193 & 0.9929 & 0 & -9.7350 \\ 0.9929 & 0.1193 & 0 & 0 \end{bmatrix}\tag{3.7}$$

For the lateral axis, the roll angle $\Delta\phi$ the only controlled output: $y(t)_{lat} = \Delta\phi$. Thus $C_{2lat} = [0 \ 0 \ 0 \ 1]$, while C_{1lat} is the identity matrix.

The optimal control problem can now be formulated as the problem of finding a controller that minimizes the **cost function** ' \mathcal{J} ', which is expressed as follows:

$$\mathcal{J} = \int_0^\infty \|z(t)\|^2 = \int_0^\infty (x^T \mathcal{Q} \ x + u^T \mathcal{R} \ u) dt\tag{3.8}$$

One would notice that \mathcal{J} is the energy of the performance output ' z '. Therefore we have:

$$\|z(t)\|^2 = \begin{bmatrix} C_2 & x \\ D & u \end{bmatrix}^T \begin{bmatrix} C_2 & x \\ D & u \end{bmatrix} = x^T C_2^T C_2 x + u^T D^T D u \quad (3.9)$$

So $\mathcal{Q} = C_2^T C_2$ and $\mathcal{R} = D^T D$. Hence, we can find matrix \mathcal{Q} for both longitudinal and lateral axes based on C_2 presented earlier:

$$\mathcal{Q}_{long} = \begin{bmatrix} 1.0001 & 0 & 0 & 1.1614 \\ 0 & 1.0001 & 0 & -9.6659 \\ 0 & 0 & 0 & 0 \\ 1.1614 & -9.6659 & 0 & 94.7702 \end{bmatrix}, \quad \mathcal{Q}_{lat} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.10)$$

Now we need to find \mathcal{R} . In order to explain how one could tune \mathcal{R} , consider the longitudinal movement of the UAV as was described by equation (2.22). In that equation, the system had four states: $\Delta u, \Delta w, \Delta q, \Delta \theta$ ($x(t) \in \mathbb{R}^4$) and two control inputs, being $\Delta \delta_e, \Delta \delta_T$ ($u(t) \in \mathbb{R}^2$). Accordingly, \mathcal{R} should be a 2×2 matrix. We end up with the following form of cost function ' \mathcal{J} ' for the longitudinal equations:

$$\mathcal{J} = \int_0^\infty \left(\begin{bmatrix} \Delta u \\ \Delta w \\ \Delta q \\ \Delta \theta \end{bmatrix}^T \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ q_{14} & q_{24} & q_{34} & q_{44} \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta w \\ \Delta q \\ \Delta \theta \end{bmatrix} + \begin{bmatrix} \Delta \delta_e \\ \Delta \delta_T \end{bmatrix}^T \begin{bmatrix} r_{11} & r_{12} \\ r_{12} & r_{22} \end{bmatrix} \begin{bmatrix} \Delta \delta_e \\ \Delta \delta_T \end{bmatrix} \right) dt \quad (3.11)$$

Now, one would notice from equation 3.11 that the diagonal elements of matrix \mathcal{R} effect individual control signals and off-diagonal elements effect combination of control inputs. So if there are restrictions on the magnitude of control signals, one can put a large number on the corresponding element in \mathcal{R} to penalize that signal more. With this in mind, we obtained the following \mathcal{R} matrices. Trial and error was also involved in the process of selecting appropriate \mathcal{R} .

$$\mathcal{R}_{long} = \begin{bmatrix} 5 & 0 \\ 0 & 0.1 \end{bmatrix}, \quad \mathcal{R}_{lat} = 5000 \quad (3.12)$$

3.2.2 Solution for the LQR problem

So far, we have seen how to formulate LQR problem and presented an approach for choosing weighting matrices \mathcal{Q} and \mathcal{R} . But the original problem statement was to find the feedback gain matrix ' \mathbf{K} ', that is optimal when it comes to the energy of the performance index in the system (In this case the performance index is the energy of the performance output ' $z(t)$ ' that is expressed as the cost function of equation (3.8)). This section explains how ' \mathbf{K} ' is actually calculated.

Chapter 12 in reference [28] presents a method for the solution of LQR using the so-called Algebraic Riccati Equation (ARE). Detailed mathematical description is available in the mentioned reference but the result is presented below:

$$u = -Kx = -R^{-1}B^TP \quad (3.13)$$

Where ' \mathbf{P} ' is the stabilizing solution to the ARE of the following form:

$$Q + PA + A^T P = PBR^{-1}B^TP \quad (3.14)$$

This equation takes in the system matrix ' \mathbf{A} ', the control matrix ' \mathbf{B} ', matrices \mathcal{Q} and \mathcal{R} and then you can solve it and find ' \mathbf{P} '. MATLAB function `lqr(A, B, Q, R)` calculates the solution to the ARE and also returns the optimal gain matrix ' \mathbf{K} '. In the case of this experiment, two separate MATLAB script were used: one for longitudinal and one for lateral movement. MATLAB scripts are available in Appendix C.

The computed feedback gain matrices \mathbf{K} for longitudinal and lateral axes are given below:

$$\mathbb{K}_{long} = \begin{bmatrix} 0.5560 & 0.0331 & -0.7552 & -4.7789 \\ 0.3925 & -0.0375 & 0.0589 & -0.1932 \end{bmatrix} \quad (3.15)$$

$$\mathbb{K}_{lat} = \begin{bmatrix} -0.0006 & 0.1087 & -0.4669 & 0.4282 \end{bmatrix} \quad (3.16)$$

Let us end this section by a step-by-step procedure on how one can design a linear quadratic regulator for a typical linear dynamical system:

1. Describe the dynamics of a linear control system using state-space representation and identify system states, control signal and measurable outputs;
2. Analyze the system to understand whether it is controllable and/or observable using matrices 3.2 and 3.3;
3. Form matrices \mathcal{Q} and \mathcal{R} considering limitations of the available control signals;
4. Form the Algebraic Riccati Equation as in 3.14 and solve for ' \mathbf{P} ';
5. Find the feedback gain matrix ' \mathbf{K} ' using equation 3.13 and form the closed-loop system.

3.3 PID Controller

Proportional, Integral, Derivative (PID) Controllers are widely used in many industrial applications. These types of controllers are relatively easier to design and implement, compared to other type of controllers. As [29, p. 160] explains, **Proportional action** "provides a contribution which depends on the instantaneous value of the control error". Simply put, the bigger the error, the bigger the control signal (error is expressed as the difference between the reference signal and the output: $E(s) = R(s) - Y(s)$). **Integral action** as [29] explain : "Gives a controller output that is proportional to the accumulated error"; and the **Derivative action**, referring to the same text: "acts on the rate of change of the control error".

PID controller is represented in the Laplace domain as:

$$C(s) = \frac{(K_d + \frac{K_p}{N})s^2 + (K_p + \frac{K_i}{N})s + K_i}{s(1 + \frac{s}{N})} \quad (3.17)$$

The question then becomes how one could choose the so-called PID gains: namely K_P , K_I , K_D and N (filter coefficient). There are number of methods for finding PID gains (Pole-placement, Ziegler-Nichols (Z-N) oscillation method, etc.). In this experiment however, PID controller was tuned using automatic PID tuner, which is available as part of the control system toolbox in SIMULINK. As it will be seen in chapter 4, one PID controller was used for the lateral axis, in order to provide reference tracking of roll angle. One PI controller was used for altitude tracking, and an integrator provide airspeed tracking. Following table contain gains for each controller:

Table 3.1: PID gains

Axis	Reference Signal	K_P	K_I	K_D	N
Longitudinal	Altitude (h)	0.8645	0.0114	0.0199	15.92
	Airspeed (V_{air})	0	3.2674	0	0
Lateral	Roll angle (Φ)	5.3301	0.800	2.004	38.7852

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 4

Simulink Models for Computer Simulation

Equations presented in chapter 2 are mathematical description of the UAV system. Mathematical models are approximation of the actual system. This means that they are capable of presenting a physical platform with certain level of uncertainty [3]. One needs to determine whether the model developed earlier, sufficiently and accurately describes the actual system. Also, it would be reasonable to minimize the risk associated with control system implementation, by running it through simulation and analyzing its behavior. MATLAB/SIMULINK provide a rather powerful tool for engineers in various discipline to simulate and analyze mathematical models and control systems [24]. Special set of blocks and functions are available as part of the 'MATLAB Aerospace Toolbox/Blockset', 'MATLAB Control System Toolbox' [25]. They are widely used by engineers to simulate aircraft dynamics and control. The focus of this chapter is on simulating UAV model and controller developed in chapters 2 and 3. Simulation results will be covered in the next chapter.

4.1 UAV Non-linear Simulation Model

4.1.1 Datcom

In order to model an aircraft using aerospace toolbox, preliminary knowledge of aircraft geometry, static and dynamic stability characteristics are required. These information has already been used once in Chapter 2 section 2.3.2 for linearization of UAV model. Here, we focus on importing DATCOM output file into MATLAB, and use it to simulate SmartFly in SIMULINK.

Appendix C contains the MATLAB script written to import DATCOM file into the MATLAB, using `datcomimport` function. This function stores all the information from DATCOM in a MATLAB cell array (called SmartFly). This array could contain various structure elements (in this case only two) that further contain information from various DATCOM analysis (static, dynamic, different Mach number and etc.). The script also extracts useful cell arrays from DATCOM and stores them into normal vectors and matrices (of type double) in MATLAB workspace, for easier access. It is sometimes reasonable to plot some of the aerodynamic characteristics of the UAV in MATLAB. For example, the so-called lift curve is illustrated in figure 4-1. Some other important parameters are also shown in figure 4-2 through 4-4.

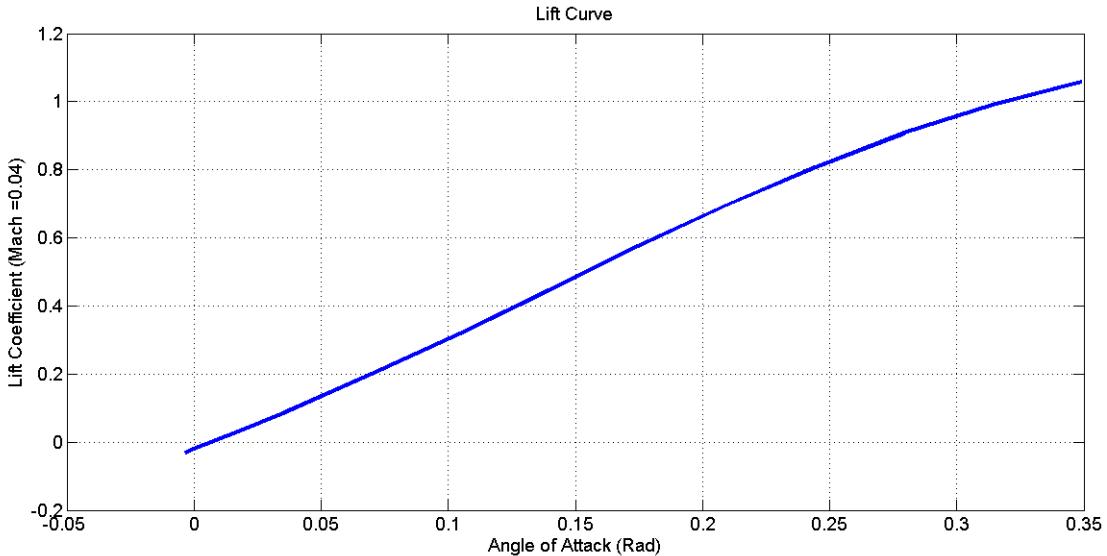
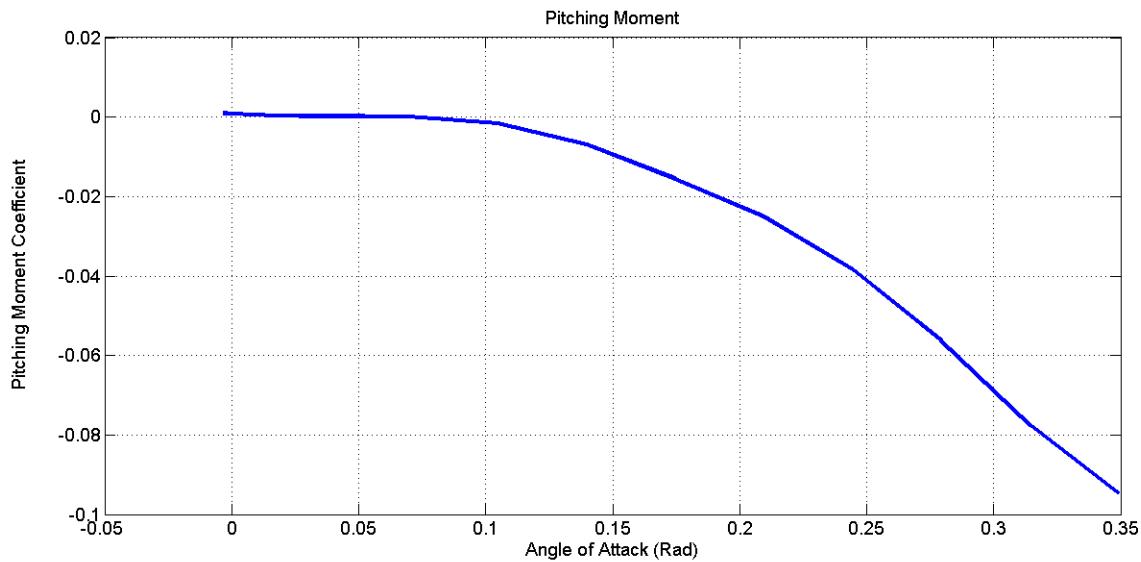
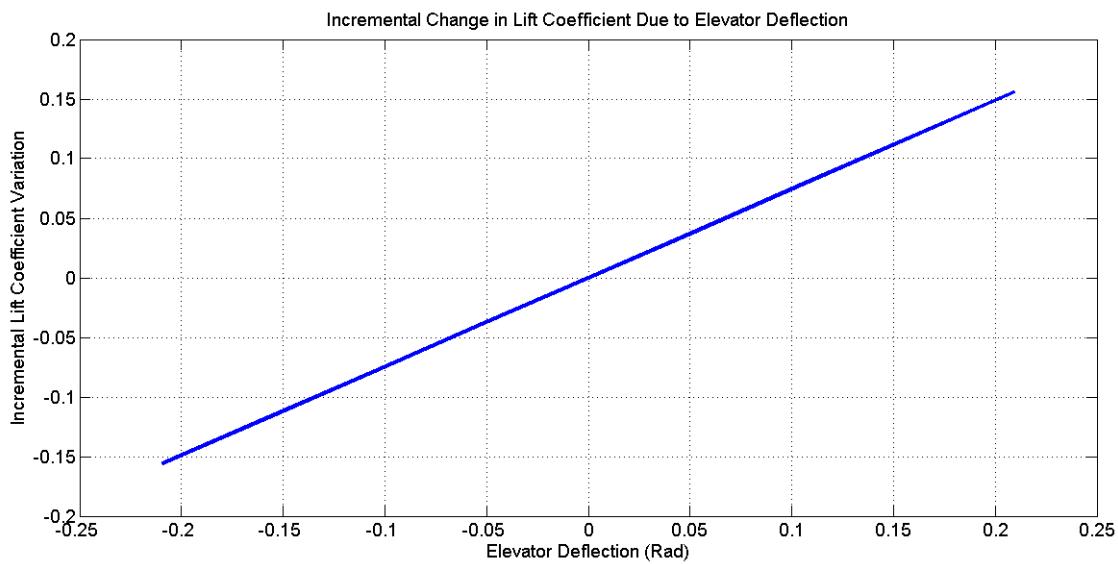


Figure 4-1: Lift Coefficient C_L Due to change in α

Figure 4-2: Pitching Moment C_M Due to change in α Figure 4-3: Incremental Change in C_L Due to Elevator Deflection δ_e

4.1. UAV NON-LINEAR SIMULATION MODEL

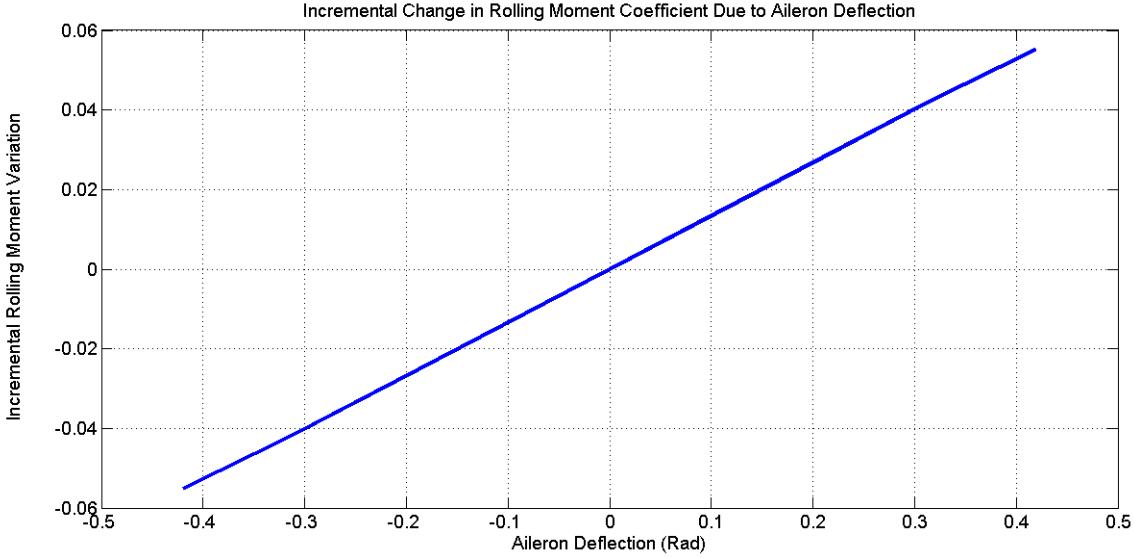


Figure 4-4: Incremental Change in rolling moment coefficient C_l Due to aileron Deflection δ_a

4.1.2 SmartFly Dynamics Simulation

After importing DATCOM information into MATLAB workspace, we can proceed by simulating system dynamics in SIMULINK. An overview of the simulation model, containing controller and dynamics is illustrated in figure 4-5.

Simulation block developed for UAV dynamics is titled **Non-linear System** and is marked with the picture of SmartOne. Non-linear system block is comprised of two main sub blocks: **Flight Condition** and **UAV Dynamics**. This is shown in figure 4-6. The orange block simulates the flight condition. Basically it pre-processes and calculates required parameters for Dynamics block. These parameters include the **Mach** number, which is calculated based on aircraft velocity and altitude, **Dynamic pressure** (denoted Q), AoA (α) and simple unit conversions (degrees to radiance for angles).

UAV Dynamics block is the heart of the simulation model. This block receives two major set of information: flight condition parameters along with DATCOM data, and control signals from longitudinal and lateral controller blocks. The purpose of such block is to simulate non-linear dynamics of the UAV. There are four major sub blocks inside the **UAV Dynamics** (figure 4-7): "Thrust Force Calculation", "Aero-dynamic Force/Moment Calculation" (also known as the 'Aeroblock'), "Gravitational

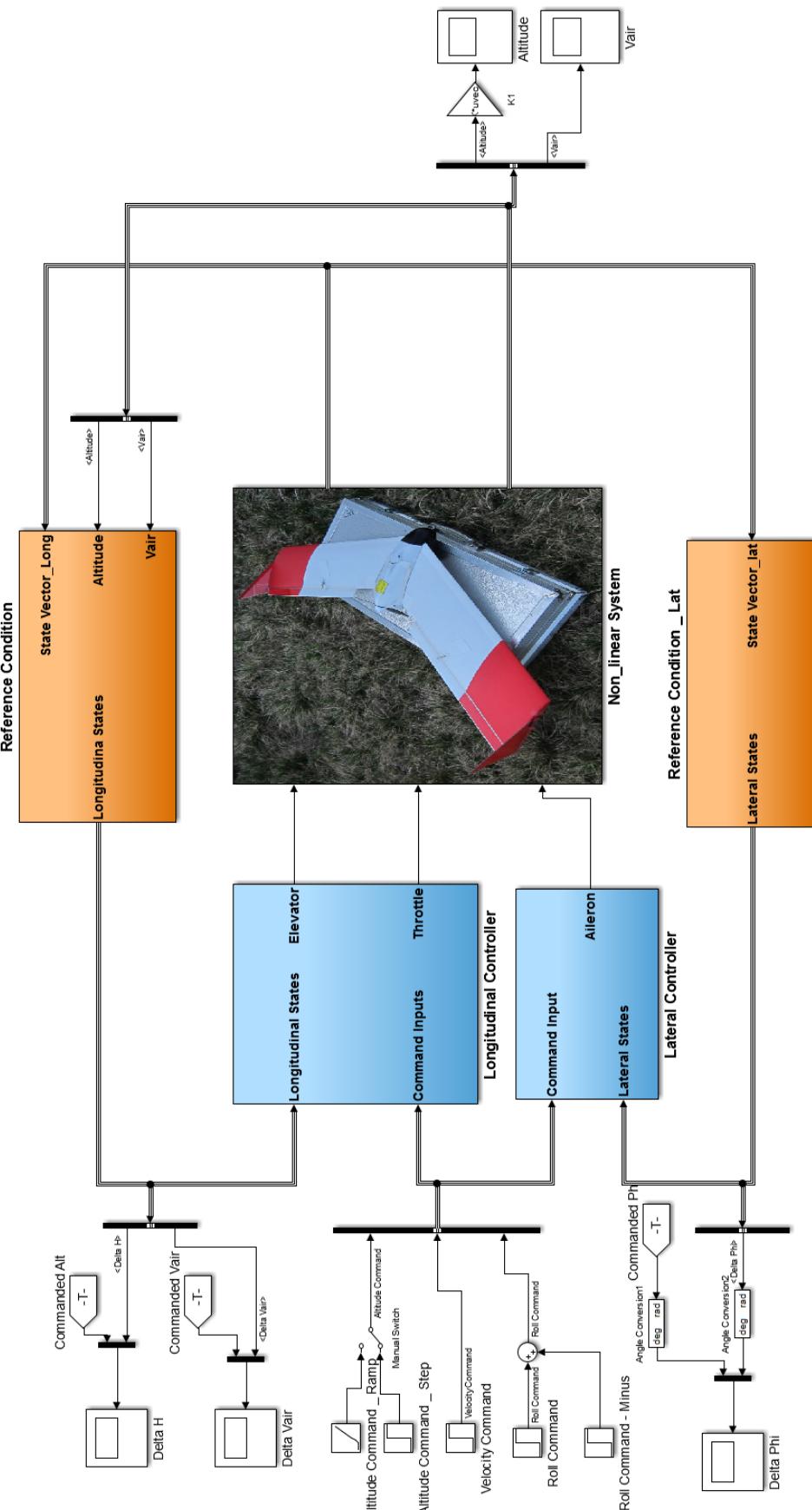


Figure 4-5: Overview of the Simulation Model

4.1. UAV NON-LINEAR SIMULATION MODEL

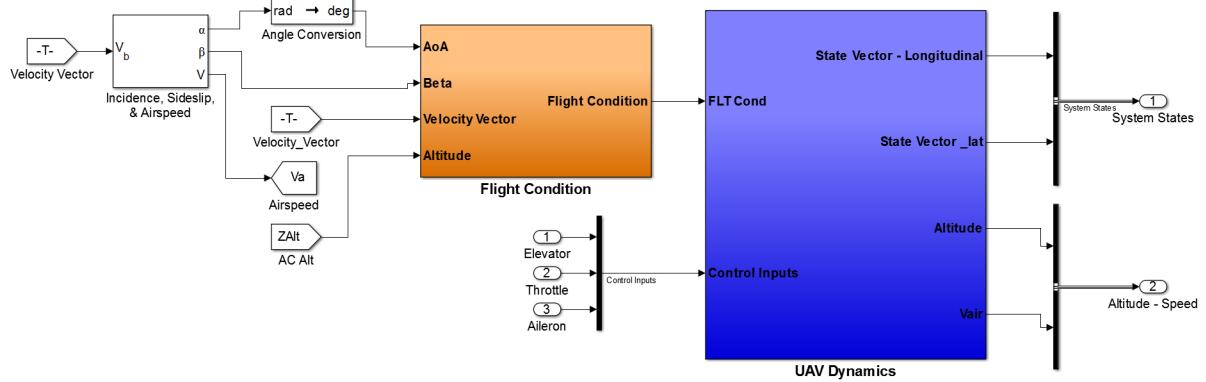


Figure 4-6: Non-linear System

Force” block and ”6 DoF Euler angles” block. Lets explain each block individually, starting with thrust force calculation.

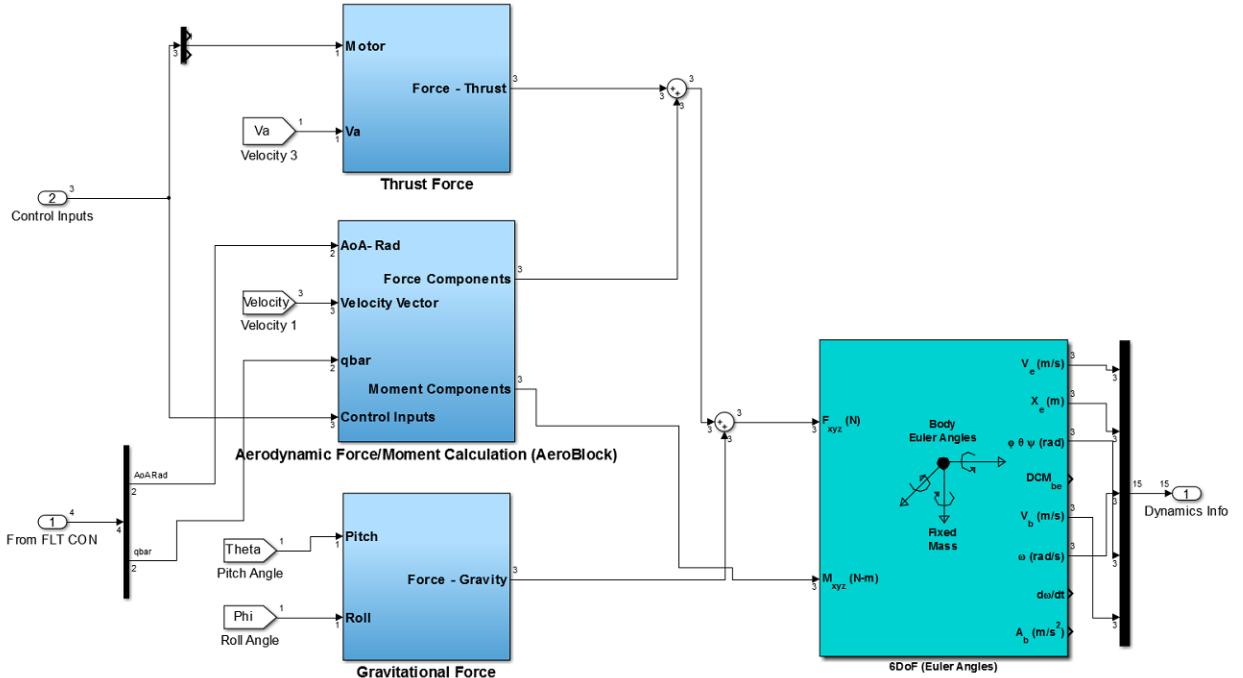


Figure 4-7: UAV Dynamics

One would recall from chapter 2 that the total force acting on an aircraft is comprised of three elements: Thrust, Gravitational and Aerodynamic force. Thrust force is directly produced by the aircraft propulsion system. The task of this block is to compute contribution of thrust force in the UAV total force and moment. A reasonable model for UAV propulsion force is presented in chapter 4 of [26]. In that

model, thrust force generated by propeller is assumed to act directly along the UAV body x-axis (hence generating no moment) and is calculated as follows:

$$F_{prop} = \frac{1}{2} \rho S_{prop} ((K_{motor} \delta_T)^2 - V_{air}^2) \quad (4.1)$$

In which ρ is the air density, S_{prop} is the area swept by the propeller, K_{motor} is the engine constant, δ_T is the throttle opening, and V_{air}^2 is the square of airspeed.

The task of the **Aeroblock** is to calculate contribution of aerodynamic forces and moments. This is accomplished using aerodynamic characteristic of the UAV. Inside view of the Aeroblock is shown in figure 4-8.

Aerodynamic forces and moments are functions of aerodynamic coefficients. These coefficients are obtained from DATCOM for different AoA and control surface deflections. In other words, aerodynamic coefficients should be evaluated for given α , δ_e and δ_a . This is accomplished inside the Aeroblock using **pre-lookup** tables and **interpolation** block.

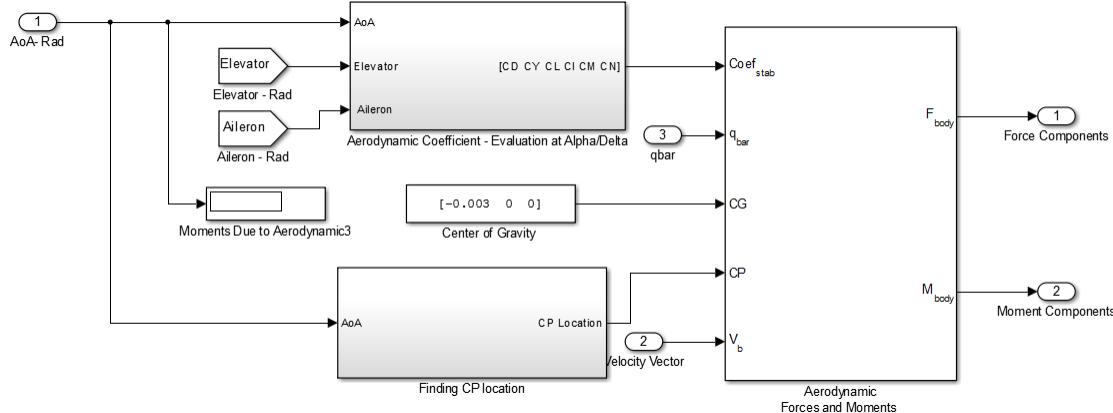


Figure 4-8: Inside of the "Aeroblock"

After evaluating coefficients at α , δ_e and δ_a , SIMULINK block entitled "Aerodynamic Forces and Moments" receives these coefficients and calculate forces and moments due to aerodynamics. Notice that coefficients are defined in the aircraft stability axis, as explained in section 2.1, but forces and moments are expressed in the body frame. Conversion is accomplished inside the block by selecting output axis to be body axis.

Gravitational force is calculated based on equations 2.10. Gravity vector is positive in body positive **Z** axis (Z_B). Resultant force is obtained by subtracting the gravitational force from aerodynamic force. As mentioned before, since gravity acts upon aircraft CG, it does not affect moment.

Finally, there is the "6 DoF Euler angle" block. This is a built in SIMULINK block (part of aerospace blockset) that receives forces and moments and returns 6 DoF Euler angle representation of equations of motion. It basically provides information on aircraft position (X_e, Y_e, Z_e), orientation (ϕ, θ, ψ) and velocity (u, v, w, p, q, r).

4.2 Simulation of Control Loops

4.2.1 Reference Block

Controllers developed in chapter 3 was simulated in blue-colored blocks of figure 4-5. Before exploring these two blocks let us examine the orange blocks entitled "Reference Block".

In section 2.3, it was explained that the linearized UAV model is valid when the perturbation is in the vicinity of the steady-state flight condition. It was also discussed that the resulting state-space representation are in terms of deviation from that steady-state condition, and that we control small deviations from the steady-state value. The function of "Reference Block" is to find deviation from the steady-state condition for each state in equation 2.22 and 2.23. This was simply done by subtracting the variable from its reference value: $\Delta u = u(t) - u_0$, $\Delta w = w(t) - w_0$, so on and so forth.

4.2.2 Longitudinal and Lateral Controllers

Now let us examine the longitudinal controller block. Overview of the simulated longitudinal controller is illustrated in figure 4-9. The block takes two set of inputs: Input Commands and Longitudinal States. Remember that states are in fact deviations from trim condition. For the longitudinal axis we have: Δu , Δw , Δq , $\Delta \theta$, ΔV_{air}

and Δh .

Altitude command is subtracted from the current altitude to produce an error signal. This error signal is fed through the PI controller, with gains demonstrated in table 3.1. Same goes with airspeed command and the integrator. Output of both controllers are subtracted from control signals generated through **LQR** feedback. This produces final, combined control signals that are further passed through saturation blocks which represent limitation in aileron deflection angle ($-12 \text{ deg} < \delta_e < 12 \text{ deg}$) and throttle opening ($0 < \delta_T < 1$).

Lateral Controller block is illustrated in figure 4-10. Same as before, the error signal (in this case the difference between commanded roll angle and actual roll angle) is fed through the PID controller and is subtracted from lateral states to create control command δ_a .

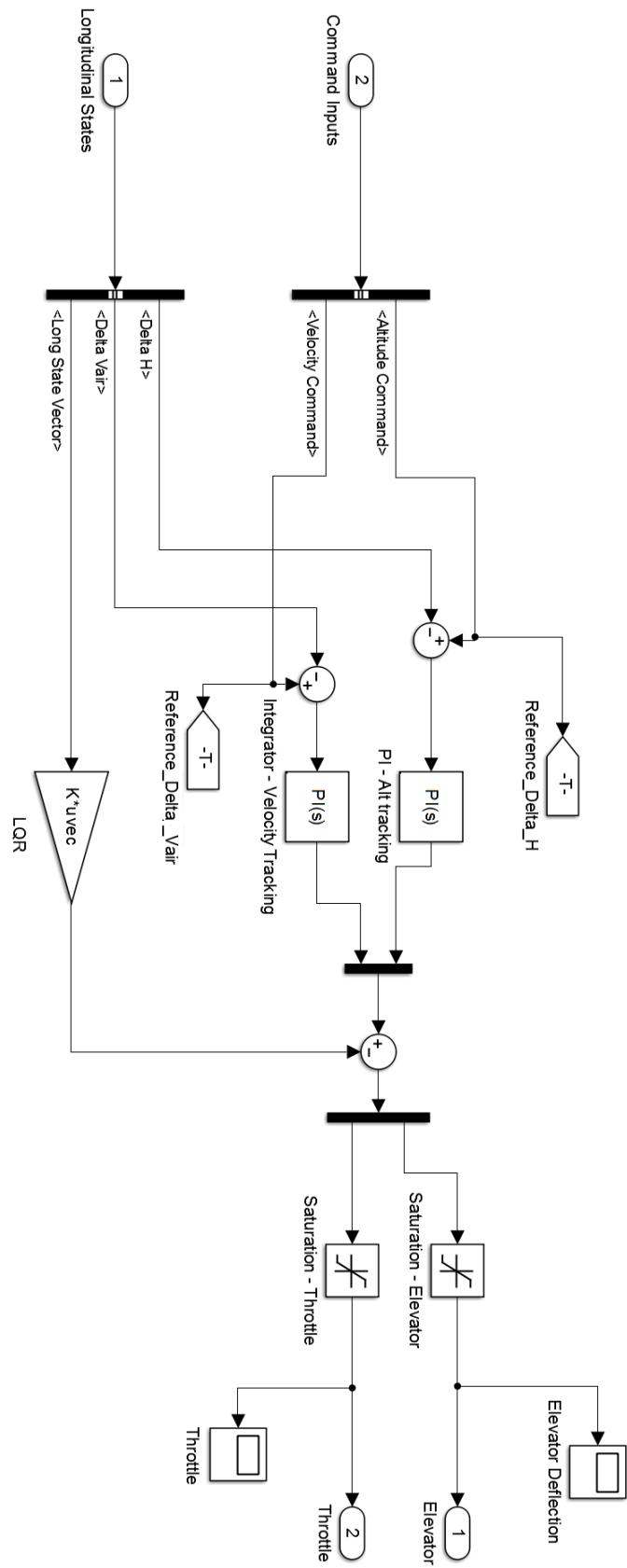


Figure 4-9: Longitudinal Controller

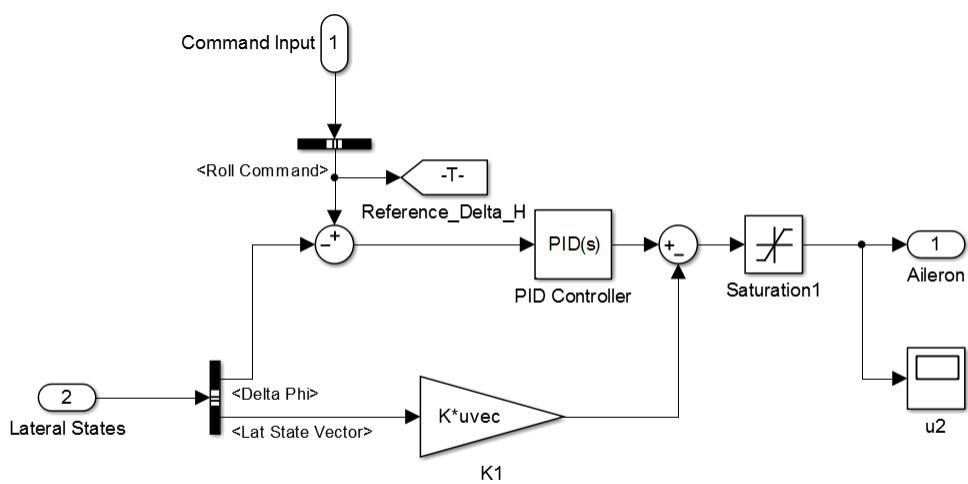


Figure 4-10: Lateral Controller

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 5

Results of Numerical Simulations

So far we have discussed how the UAV was modeled and simulated in SIMULINK. We also talked about how to design and simulate a controller for the system. It is now time to verify functionality and robustness of the designed controller. In this section we present results of series of simulations and evaluate the behavior of the designed control system. Discussion on these results will be presented in the next chapter.

First thing to check is to see whether the controller is able to stabilize the system. We will then move forward and check if we can track constant reference altitude, airspeed and roll angle. Finally, we need to check the system for robustness. Two scenarios will be considered: variations in UAV mass, and variation in CG location.

5.1 Stabilization of the System

Considering inherent instability of the UAV, first we need to check whether control loops are able to stabilize the system around trim condition. In order to do so, all input commands (airspeed, altitude, and roll angle) are set to zero. It is expected to see variables converge to trim condition values of table 2.1.

Figure 5-1 illustrates UAV airspeed at trim condition and figure 5-2 shows trim condition altitude. Trim condition Euler angles (ϕ , θ and ψ) are shown in figures 5-3 through 5-5. θ is equal to around 6.9 deg as expected.

Let us further investigate step tracking of altitude, airspeed and roll angle.

5.1. STABILIZATION OF THE SYSTEM

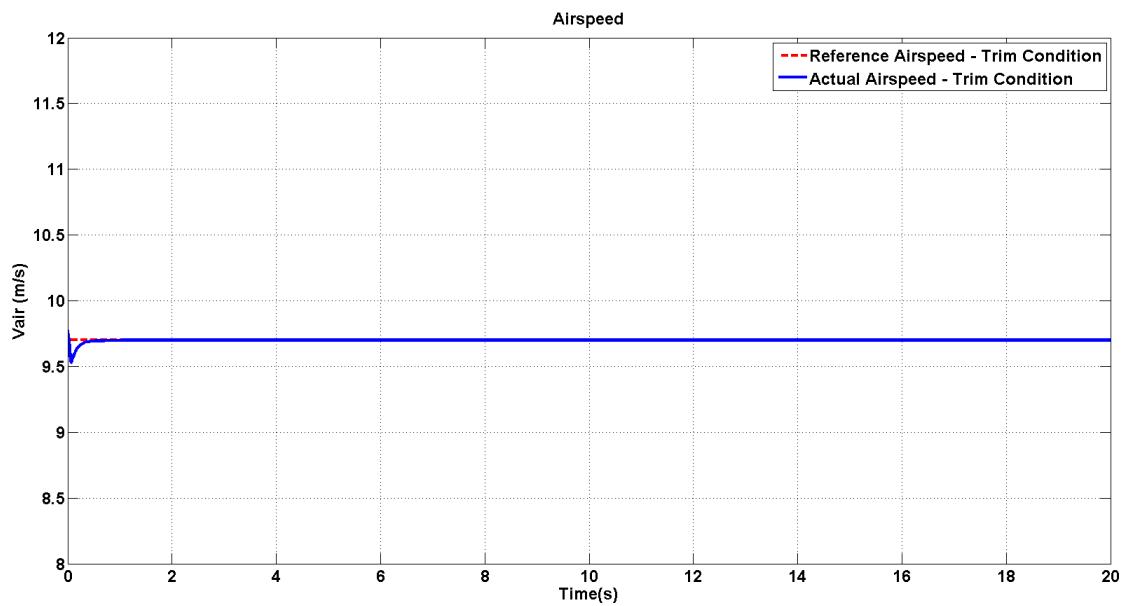


Figure 5-1: Trim Condition: Airspeed

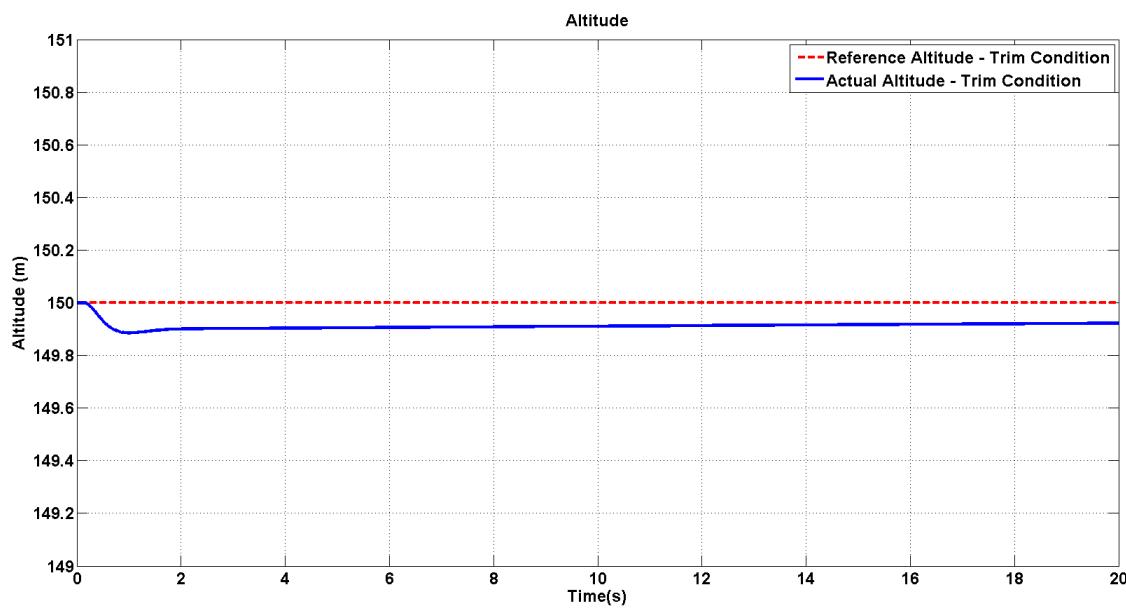


Figure 5-2: Trim Condition: Altitude Hold

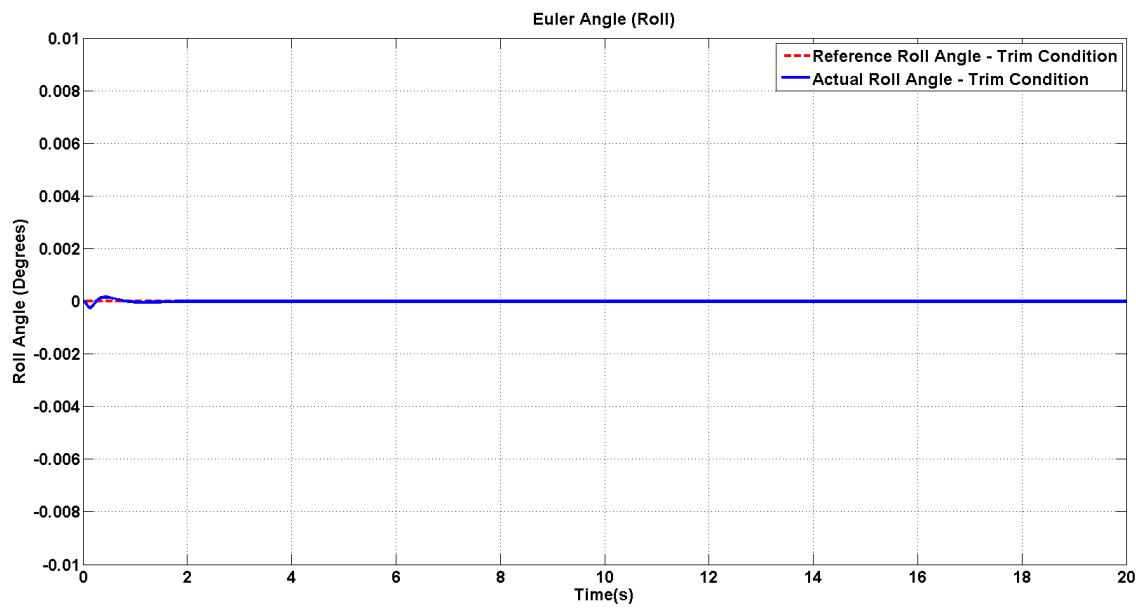


Figure 5-3: Trim Condition: Roll Angle (ϕ)

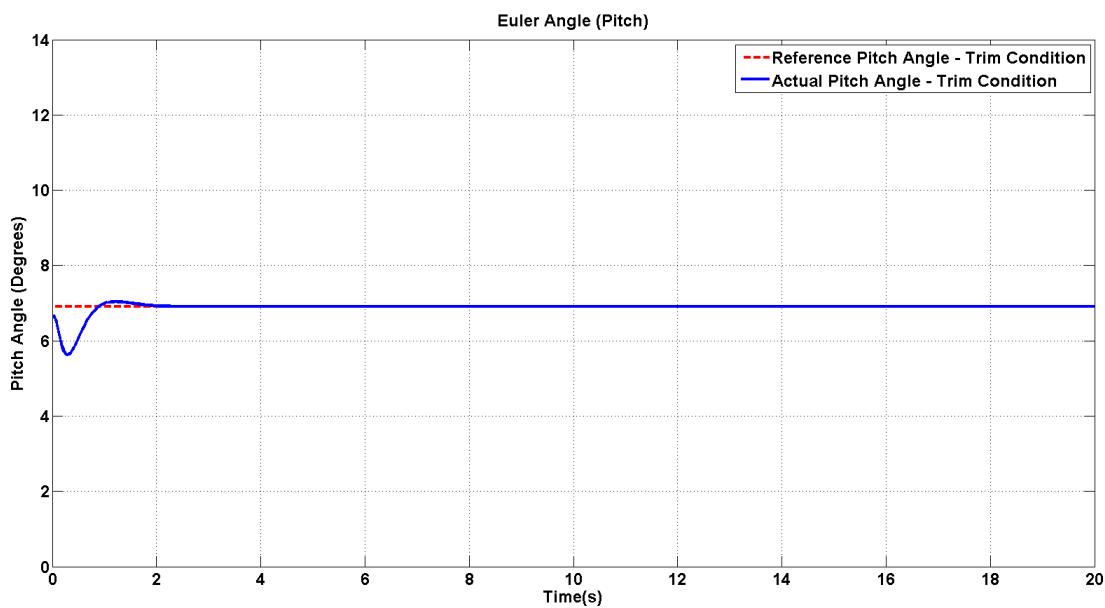
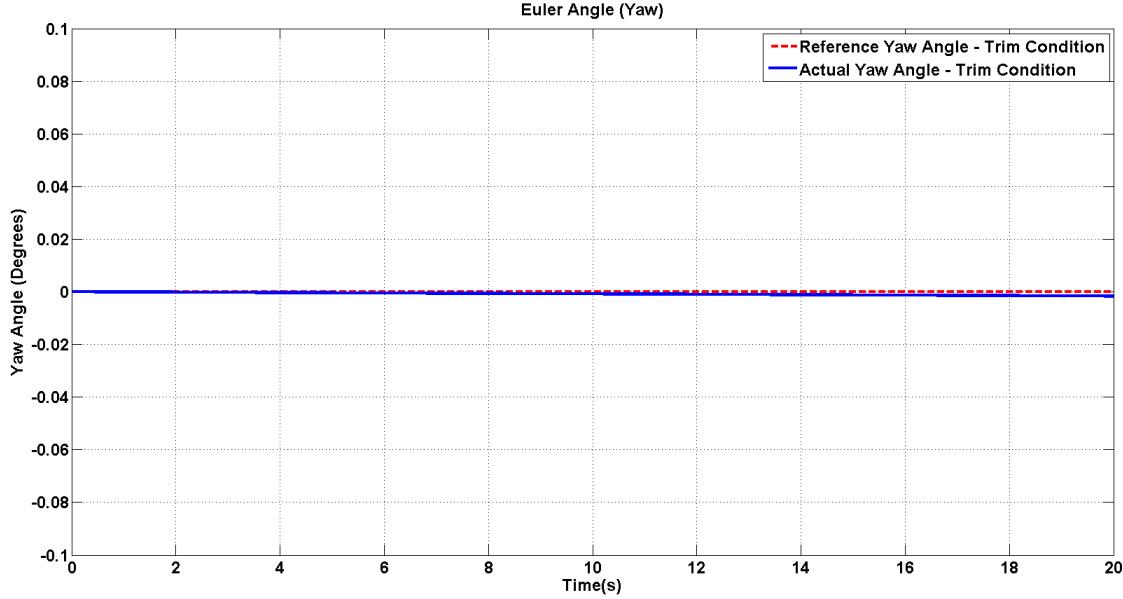


Figure 5-4: Trim Condition: Pitch Angle (θ)


 Figure 5-5: Trim Condition: Yaw Angle (ψ)

5.2 Reference Signal Regulation

5.2.1 Altitude and Airspeed Regulation

Longitudinal controller is supposed to provide tracking of step changes for $h = h_0 = 150m$ and $V_{air} = V_{air_0} = 9.77$. Let us first consider the case where we are only interested in changing altitude. Figure 5-6 depicts this scenario. A step input with amplitude 1, representing one meter altitude change is applied to the system at $T = 25$. Longitudinal controller then generates appropriate control signal δ_e and tracks the step input. It should be mentioned that according to the coordinate system presented in section 2.1, positive value for altitude corresponds to UAV descend. Hence the plane is expected to lose 1 meter height. Figure 5-7 shows that it is indeed correct. Also, the controller keeps V_{air} steady by changing δ_T while UAV is descending¹. This is shown in figure 5-8.

Further, let us investigate the case when both Δh and ΔV_{air} command are present: altitude command change is applied at $T = 25$, and airspeed command change is applied at $T = 50$. It is shown in figure 5-9 and 5-10 that both inputs have been

¹In an open loop system loosing altitude causes V_{air} to increase rapidly.

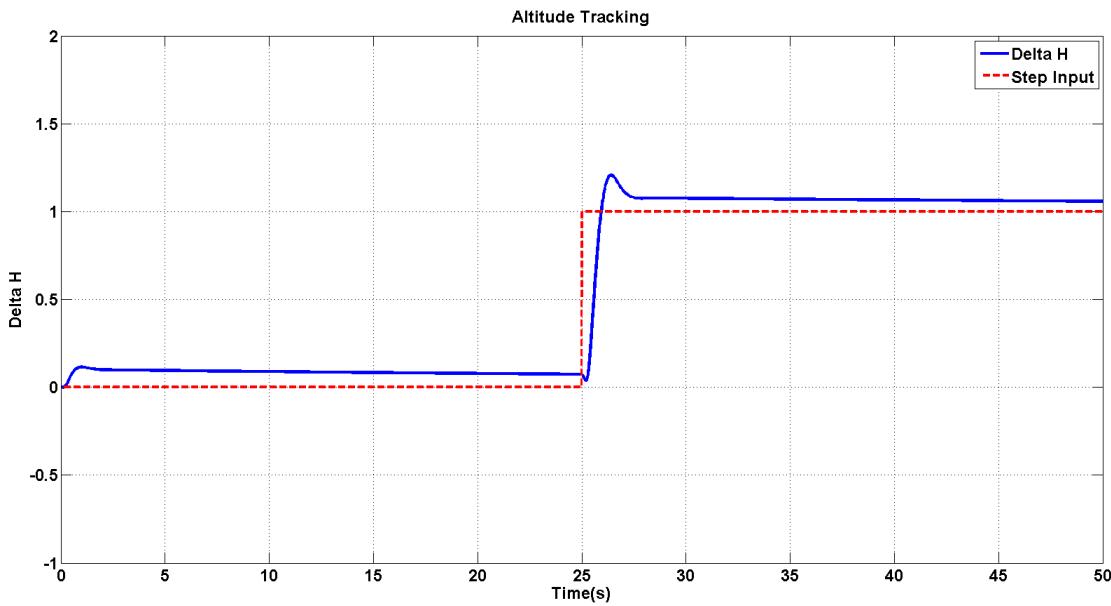
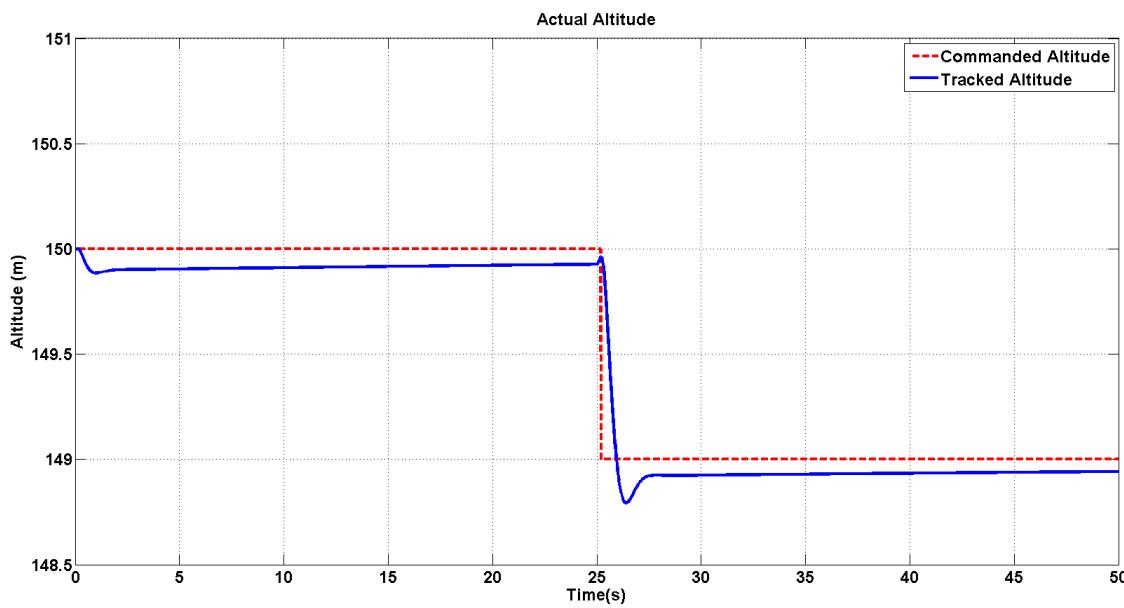
Figure 5-6: Altitude Tracking: Δh 

Figure 5-7: UAV Descent Due to Altitude Command

5.2. REFERENCE SIGNAL REGULATION

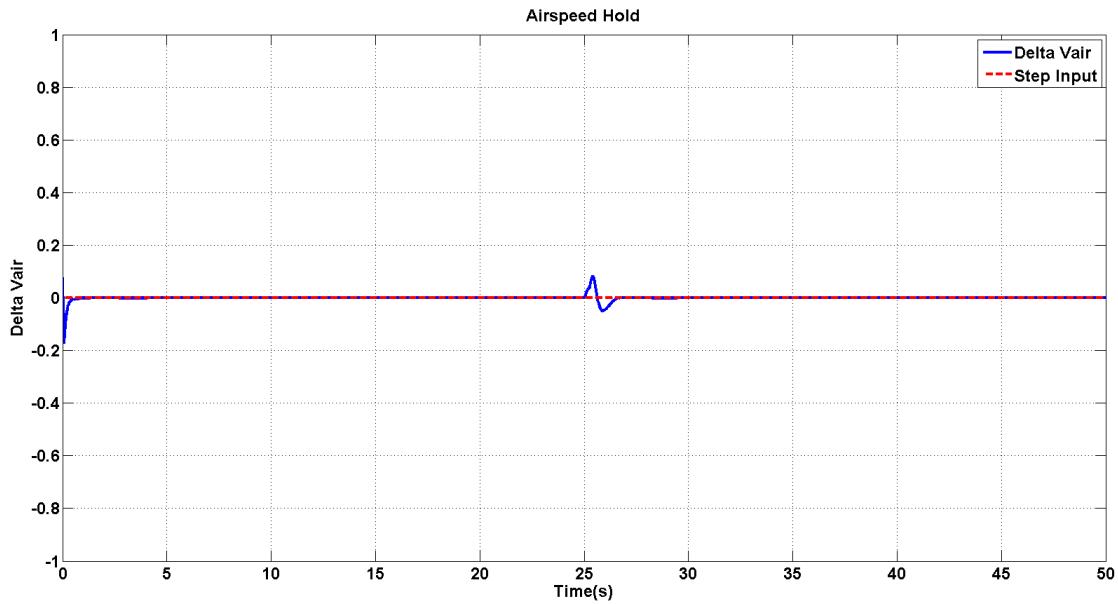


Figure 5-8: UAV Descending with Constant Airspeed

appropriately tracked.

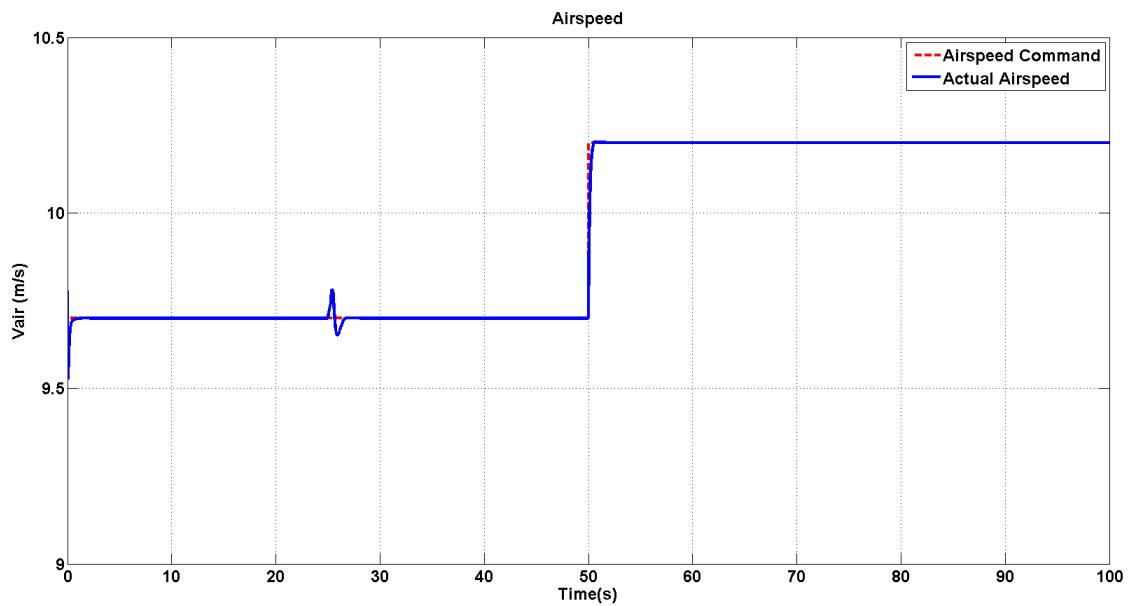
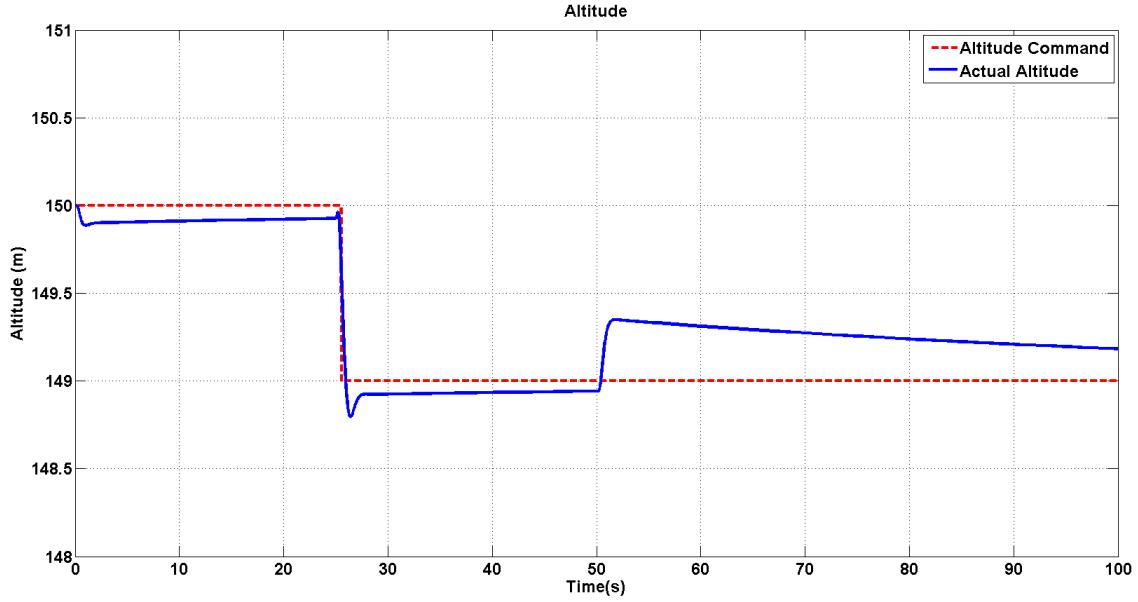
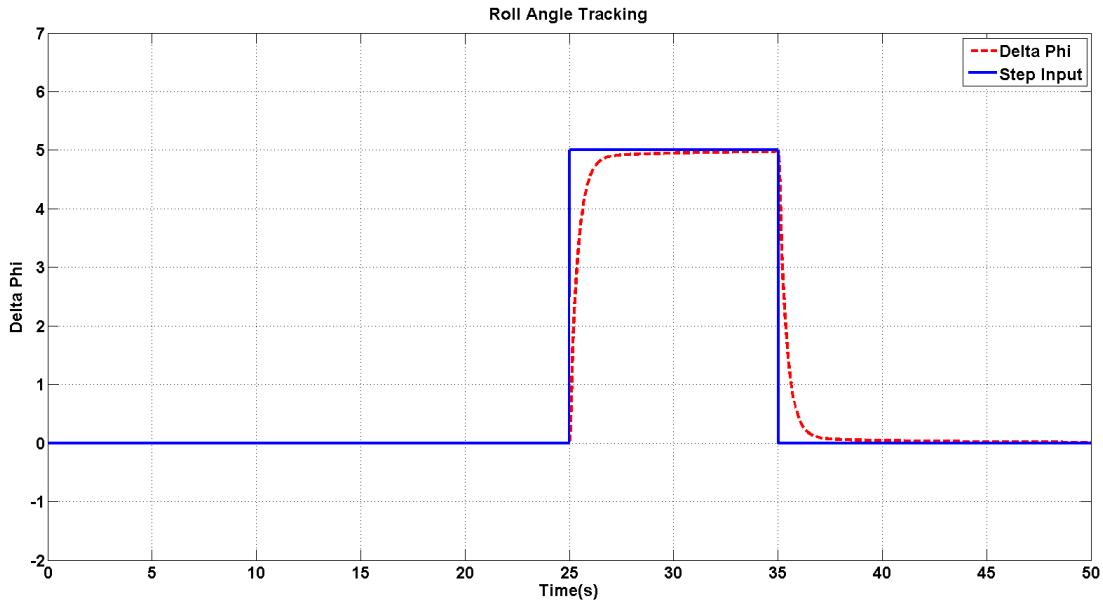


Figure 5-9: Airspeed tracking at $T = 50$


 Figure 5-10: Altitude tracking at $T = 25$

5.2.2 Roll Angle Tracking

Lateral controller provides roll angle $\Delta\Phi$ tracking using aileron deflection (δ_a). Figure 5-11 shows 5 deg turn (Positive angle corresponds to right turn) for $25 < T < 35$. Figure 5-12 illustrates aileron deflection (control signal) for roll angle tracking.


 Figure 5-11: $\Delta\Phi$ Tracking at $25 < T < 35$

5.2. REFERENCE SIGNAL REGULATION

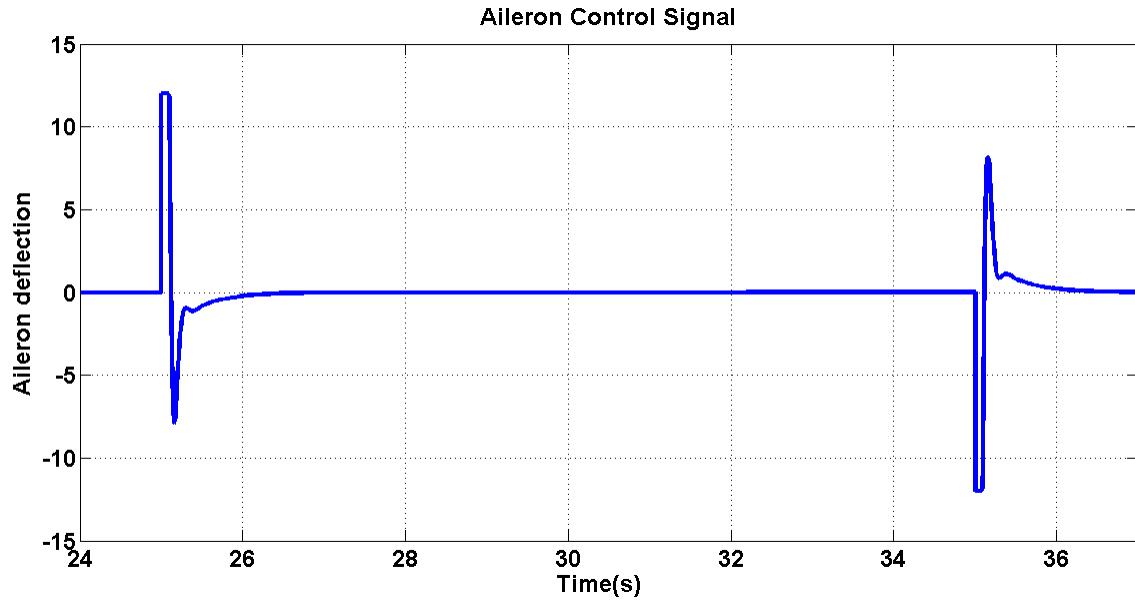


Figure 5-12: Control signal δa

Intuitively, when aircraft starts to roll, velocity and altitude will change. This is shown in figures 5-13 and 5-14.

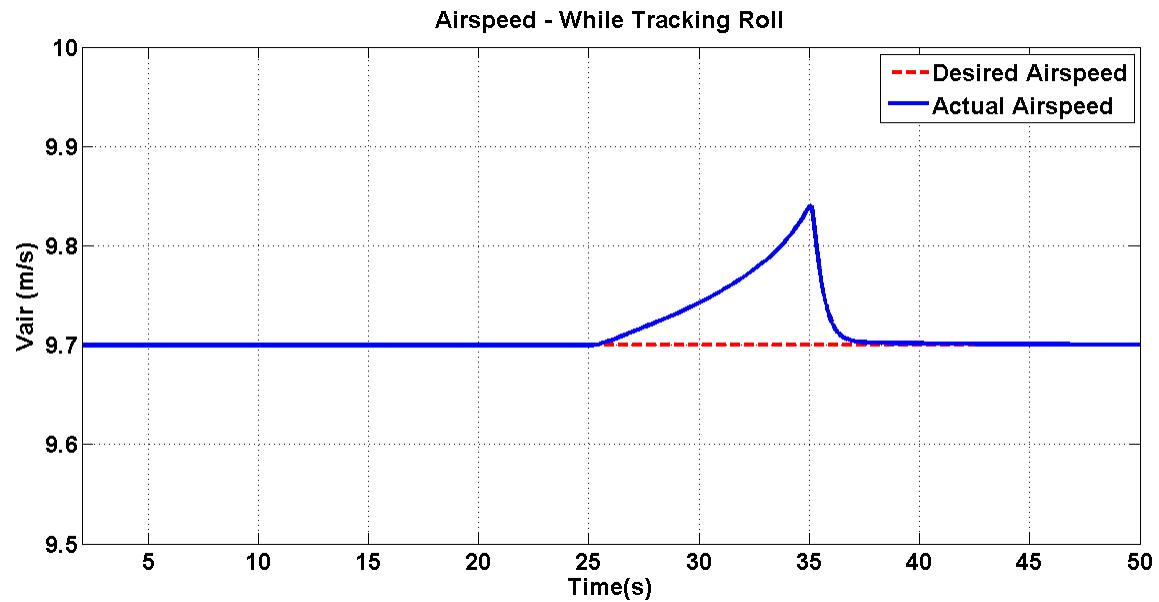


Figure 5-13: Changes in V_{air} while rolling at $25 < T < 35$

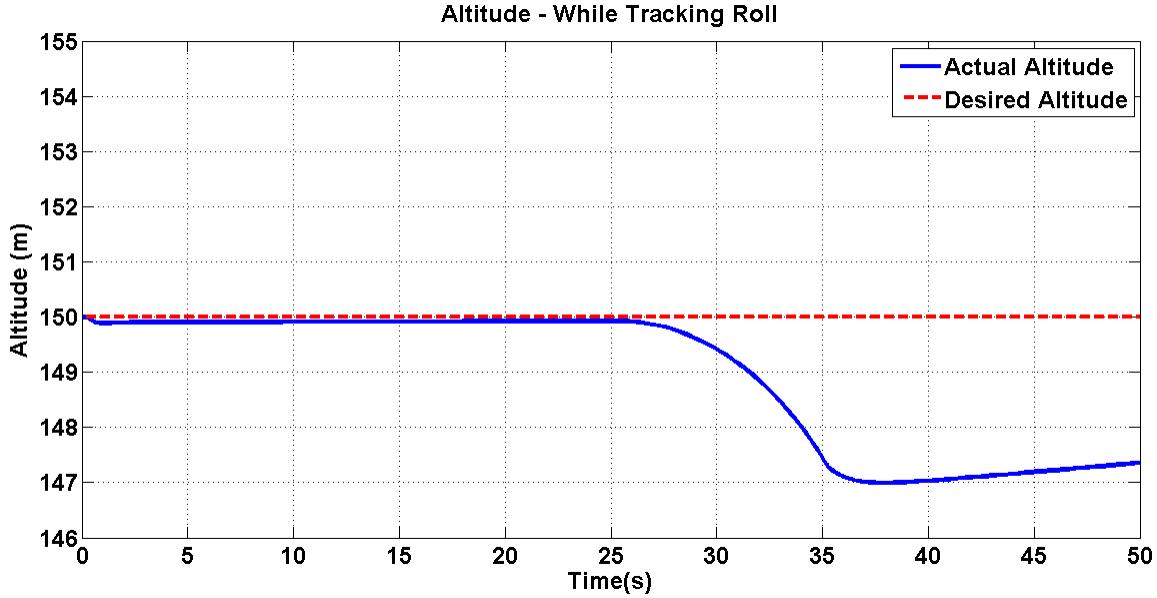


Figure 5-14: Changes in h while rolling at $25 < T < 35$

5.3 Controller Robustness

Thus far, both longitudinal and lateral controllers have been tested through series of simulations. In this section, the goal is to evaluate them in terms of robustness. That is achieved by running simulations on model with parameter uncertainty. Two different scenarios will be considered: variations in the UAV mass; and drift in the UAV CG location (forward and aft along x axis²). Let us begin by simulating the first scenario.

5.3.1 Variation in UAV Mass

Figure 5-15 shows altitude tracking in presence of uncertainty in UAV mass. According to the objective discussed in section 1.1.2, changes in the mass is between 0 to 300 grams. Therefore, we tested the controller for $m = 0.617$ kg (50 grams added mass) and $m = 1.067$ kg (500 grams added mass).

In order to modify the simulation model and simulate variations in UAV mass, two steps were followed:

²Based on the original problem statement, only variations in CG along x axis is studied here. No data is available for changes in CG location in y or z axis.

5.3. CONTROLLER ROBUSTNESS

1. change aircraft mass in DATCOM input card (Appendix A);
2. change the input parameter for airplane mass in the "Aerodynamic force and moment" block shown in figure 4-8.

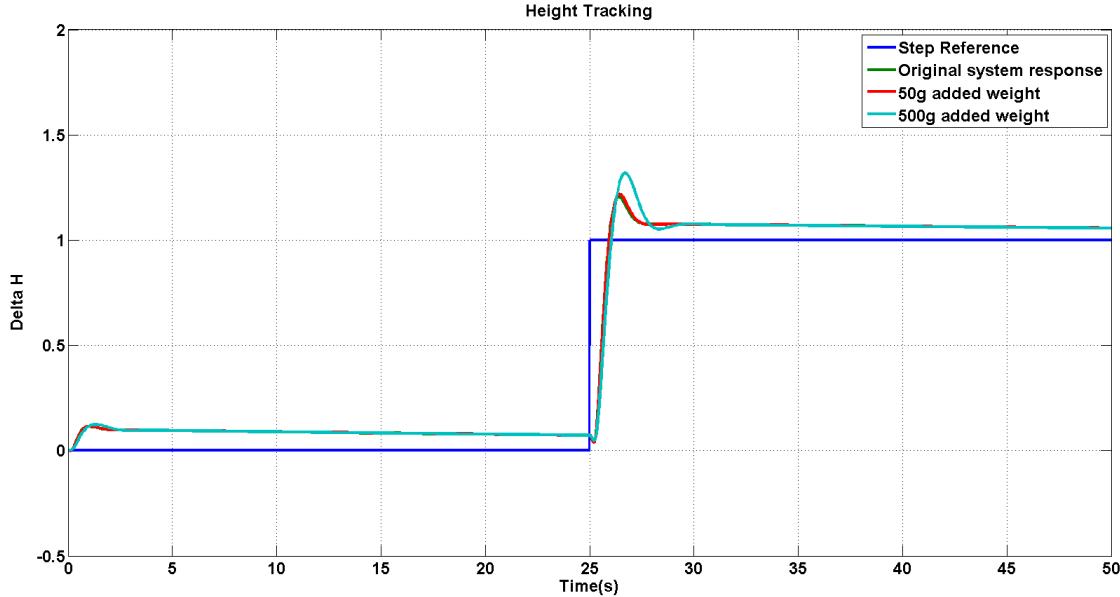


Figure 5-15: Comparison of Height Tracking for Different UAV Mass

Airspeed stabilization for different UAV mass is shown in figure 5-16.

5.3.2 Variation in UAV CG Location

Variations in CG location would greatly effect performance of small-sized flying vehicles. Moments acting on the plane are affected directly by the CG location. In this part of simulation, CG location was intentionally moved forward and back to analyze whether control loops are still capable of fulfilling their task. It should be noted that this test was indeed one of the objectives presented in section 1.1.2.

Same as the case for change in mass, two steps were followed to impose additional mass to the simulation model:

1. change CG location in DATCOM input card (Appendix A);
2. change the CG location input vector in the "Aerodynamic force and moment" block shown in figure 4-8

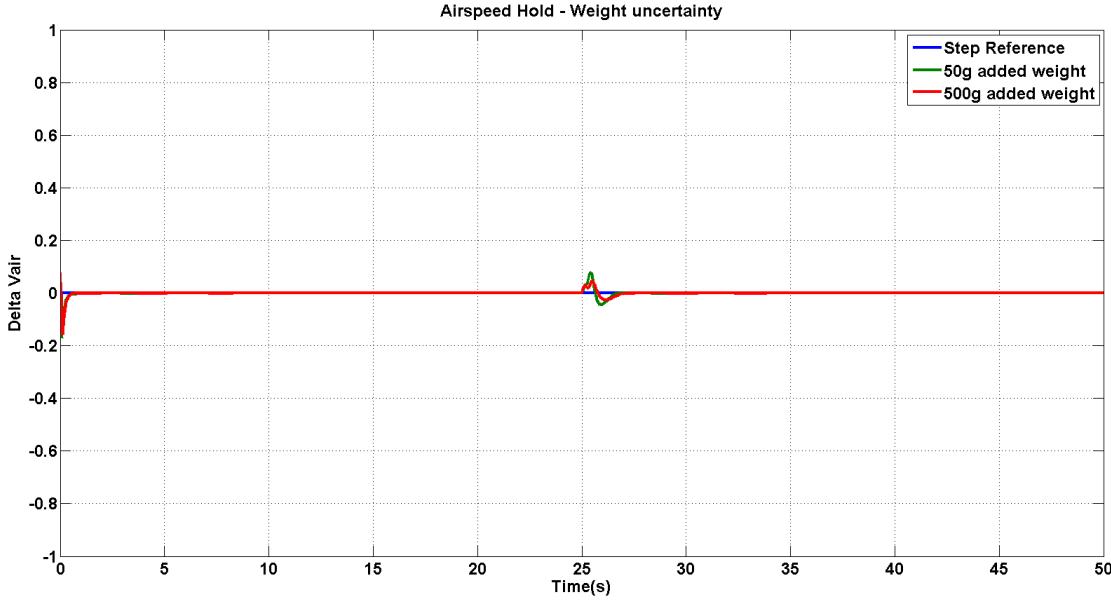


Figure 5-16: Comparison of Airspeed Stabilization for Different UAV Mass

Four different locations (two forward and two aft) were compared against the original CG location. Figure 5-17 illustrates altitude tracking (Δh) when CG is moved, while figure 5-18 shows airspeed stabilization (ΔV_{air}). Actual UAV altitude is illustrated in figure 5-19. One should note that according to the specific choice of axis system presented in section 2.1, 1 meter change in altitude (positive number) corresponds to the aircraft descent. This is because the Z_b (body z axis) is assumed to be positive pointing downward. Discussion on these plots are given in section 6.1.

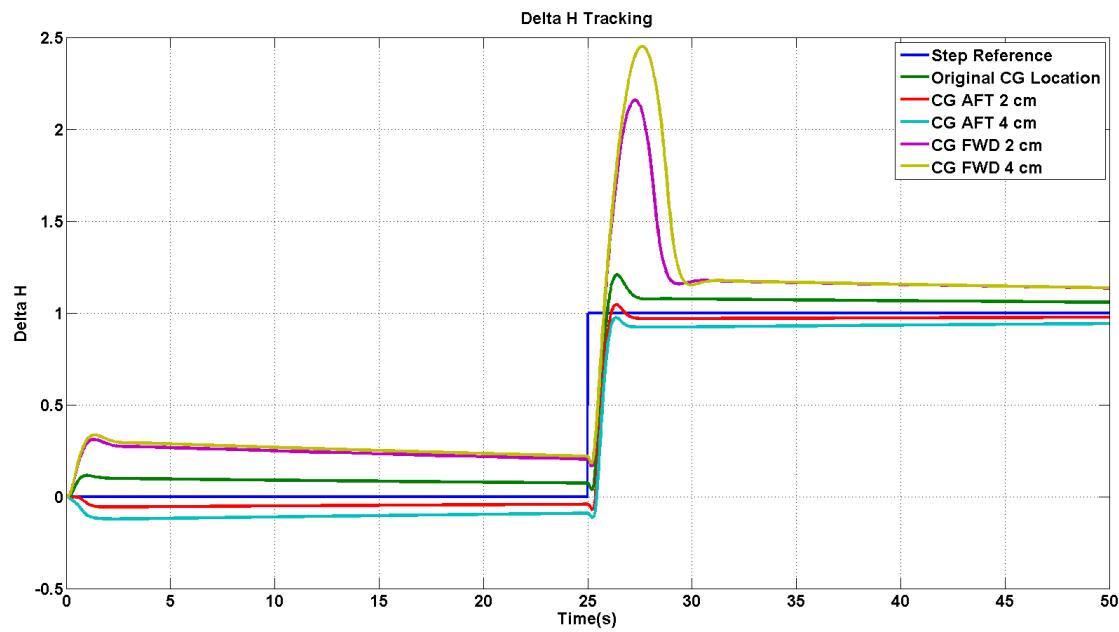


Figure 5-17: Comparison of Height Tracking for Different CG Location

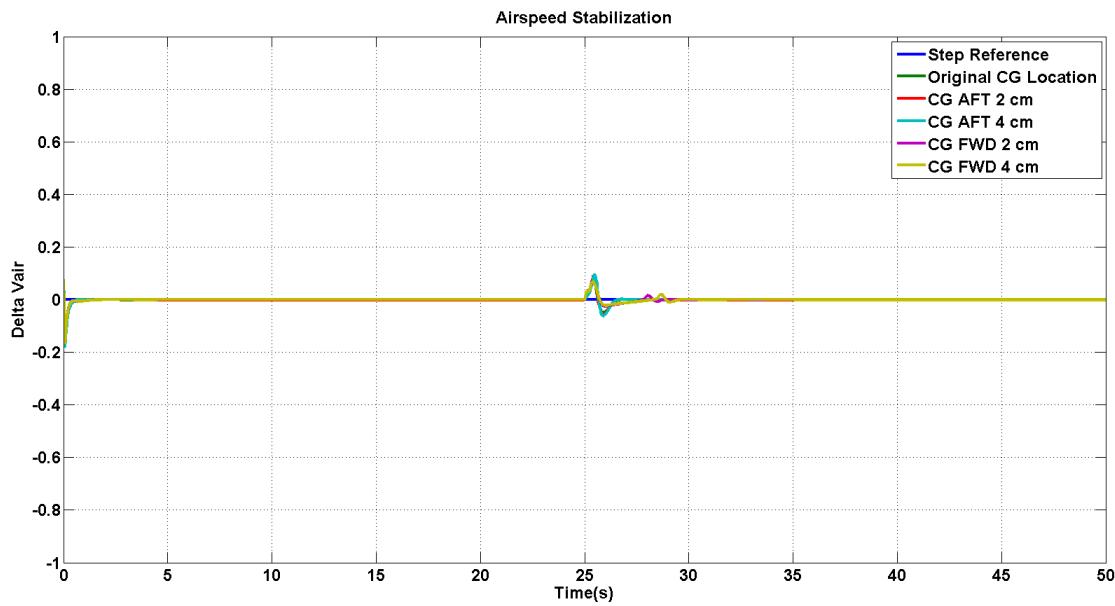


Figure 5-18: Comparison of Airspeed Stabilization for Different CG Location

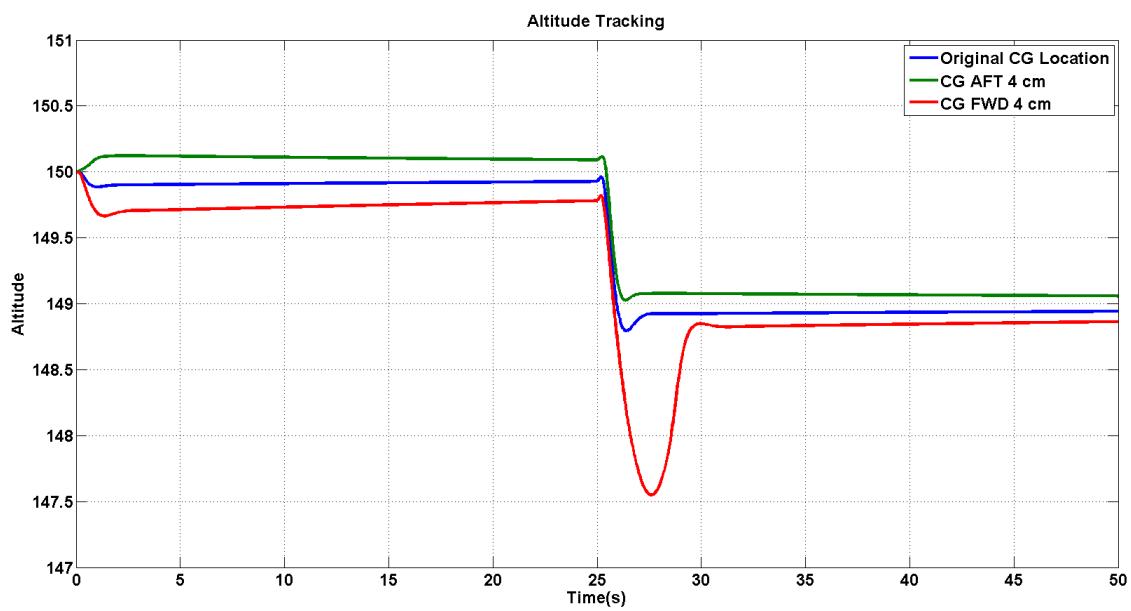


Figure 5-19: Actual Altitude for Different CG Location

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 6

Conclusion and Discussion

This chapter is dedicated to the discussion around the results of numerical simulations presented in the previous chapter. Whether project requirements and objectives were met are also discussed here.

6.1 Discussion

Results presented in the previous chapter showed that the controller architecture presented in chapter 3 provide both system stability and robustness for the non-linear UAV simulation model.

According to figure 5-1 and 5-2, longitudinal controller described in section 4.2.2 managed to stabilize the system. Produced throttle control signal (δ_T) stabilized ΔV_{air} and elevator deflection (δ_e) stabilized Δh . Small steady state error for altitude stabilization is observed in figure 5-2. We believe that one reason for this steady state error is the mathematical approximation used for finding Δh (equation 3.6). The error however, is rather small (in scale of 0.1 meter) and can be neglected.

Another interesting observation arises when both altitude and airspeed reference signal are present. This was shown in figure 5-9 and 5-10. We think that slower response of the controller after velocity command at $T = 50$, is due to the fact that changing V_{air} would effect many variables in the system (i.e. AoA, altitude). This in turn would cause the trim condition to change. Therefore, the controller which was

6.1. DISCUSSION

designed according to specific trim condition, would no longer function as good as it would for the original trim condition.

Regarding the lateral controller presented in section 5.2.2, it should be mentioned that roll angle tracking is functional for short time intervals. The reason could be the fact that steep turns, for periods longer than ≈ 10 seconds, would cause the system to deviate from trim condition too much, and the lateral controller is no longer able to stabilize the system. As it was observed in figures 5-13 and 5-14, both airspeed and altitude will change, when UAV starts to turn.

Let us further discuss two cases studied in chapter 5 for robustness: changes in the UAV CG location, presented in section 5.3.2, and variations in UAV mass, discussed in section 5.3.1. Both of these cases are very common in practical applications. Variation in mass could occur when there is a physical modification (changing the on board camera, propulsion system or etc.) and CG location drift could be the result of improper trimming or maintenance.

Changes in the CG location effect the transient response (steady-state error, overshoot and undershoot). Common sense intuition is that when CG is moved forward of its original location, UAV would experience a nose heavy situation. This creates a tendency for pitch down which results in aircraft descend. This was observed in our nonlinear simulation model as well. According to figure 5-19, when CG is placed 4 cm forward of its original location (the red line in the figure), the aircraft starts to descend in the first 2 seconds. The controller then reacts and the plane starts to climb and gains altitude. Reverse situation occurs for aft CG. That is, aftward CG location creates pitch up tendency and causes the plane to climb. This is further evidence that the nonlinear simulation model is a convenient approximation of the actual system. It should also be mentioned that the controller failed to handle changes outside the range of 8 cm. For the case of change in UAV mass, controller successfully stabilized the aircraft and provided reference signal regulation for the corresponding masses.

6.2 Project Requirements and Objectives - Status

This Master's thesis presented modeling, simulation and control system design procedure for small-size flying wing. The main objective of the research was to develop robust flight control system, considering practical limitations of this particular platform. Table 6.1 tabulates the preliminary project requirement list. These requirements however were slightly modified throughout the research, mainly due to time constraints. More precisely, item 6 of table 6.1 was removed from the list after negotiation with the customer. According to the new project plan document that was submitted in June 2014, all items have been successfully fulfilled.

Table 6.1: Project Requirements Met

No. Activity	Description	Status
1. Project Planning	Preparation/approval of the project plan document	✓
2. Preliminary Study	Problem analysis, identification of research approach	✓
3. System Modeling	Mathematical modeling, simulation and parameter identification	✓
4. Control System Design (Simulation)	Designing controller for longitudinal and lateral axes	✓
5. Test (Simulation)	Performing series of tests on the simulated controller to validate its functionality	✓
6. Control System Design (Implementation)	Hardware implementation of the designed controller	✗
7. Result Analysis	Analyzing obtained results	✓

Throughout the project, there were various meetings between the company and university, evaluating the progress of the work. Meeting minute kept at one of these meetings is available in Appendix B.

Based on the results presented in chapter 5, it can be seen that the list of objectives given in section 1.1.2, are met. In fact, the controller proved to provide robust stability for ranges beyond the required limit. For instance, in the case of CG variation, the controller was able to provide system stability within the range of 8 cm.

Project Deliveries tabulated in table 1.3 were delivered to the customer in the

meeting on August 29th, 2014 (see table 6.2). The meeting minute for the mentioned meeting is available in Appendix B.

Table 6.2: Deliveries to the customer

No.	Item	Delivered
1.	Written report (MSc thesis)	✓
2.	Simulation files	✓
3.	Documented source codes	✓

6.3 Implications of the Research

Let us point out the novel idea behind the methodology used for control system design one last time. As one would recall from chapter 3, we approached the control system design problem by separating the problem into two parts: system stabilization, and fulfillment of a certain performance criteria. We claimed that if one is able to appropriately define elements of matrix \mathcal{R} , then LQR feedback gains would not only regulate the system, but also provide reasonably good robustness. So, thus far, LQR made a "bad" system, (unstable, sensitive to control signal), slightly "better" (stable, relatively robust).

Afterwards, considering that handling specific performance criteria using LQR is hard, we made use of the conventional PID controllers. The advantage is that in this scenario, with LQR taking care of robustness and stability, PID provide reference signal regulation. PID does not need to provide system stability simultaneously with transient performance. All PID has to do is to take care of the performance criteria (in this case signal tracking performance) in a system that is already robustly stable.

From implementation stand point, we believe that after having the LQR in the loop, it would be even easier to tune PID because the system is more stable compared to the case where the PID involves system stabilization also. It should however be mentioned that finding appropriate LQR gains are not an easy task and it requires a little bit of trial and error also.

6.4 Summary

This thesis brings together knowledge and advancements in many scientific areas and engineering disciplines, namely: mathematical modeling, aerospace engineering, control system theory and computer science. Large part of the work focused on modeling the dynamics of the system. A 6 DoF nonlinear model was developed and simulated in SIMULINK. Static and dynamic stability analysis had also been accomplished in DATCOM, which is one of the most widely used software for aircraft aerodynamic analysis.

Analytical linearization of the model was presented to further simplify controller design process. Furthermore, various control loops had been developed to stabilize the system and provide reference signal tracking in both longitudinal and lateral axis. Particularly, LQR controller had been designed to work in collaboration with traditional PID.

Various simulations were accomplished to analyze performance of controller under system uncertainty. Particularly, the influence of variations in the UAV mass and CG location were studied. Based on the results obtained, we can conclude that the controller functions properly with acceptable robustness. This further implies that linearization of the model was somewhat accurate: because the controller was designed based on the linear system, but was tested against the non-linear simulation model.

We can also conclude that LQR controller is a reasonable choice for small-sized UAV. The reason is that flying-wings are usually equipped with commercial-off-the-shelf processors, with limited processing power. This would limit the choice of flight control system. Since LQR is simply a matrix of static feedback gains, it is easy to implement and does not require high level of processing power.

6.5 Ethics of the Work

Recently, there has been active ethical debates and discussions in UAV communities concerning the use of military UAVs as war drones [31]. However, as civilian UAVs find their path through the industrial market, many environmental activists and supporters of the green space acclaim the use of civilian UAV systems¹ [32]. First, we would like to point out that the UAV system used for this research is designed specifically for civilian applications. Neither humans, nor animals are to be harmed by the use of such system. We would also hope that the designed control system promote the level of safety, both for the operator, and other civilians in the vicinity of operation.

We were devoted to consider guidelines, codes and instructions regarding ethics of research in engineering (i.e. Institute of Electrical and Electronics Engineers (IEEE) code of ethics) at all times during the work. According to [33], principle item number 2, robotics engineer is obligated to consider and respect human rights in his/her work. We should therefore mention that the operator is required to comply with the local regulations regarding UAV Unmanned Aerial Systems (UAS) flight and operation. These regulations consider privacy issues of operating such system in specific area.

6.6 Future Work

An interesting recommendation for future work would be to use empirical data in the simulation model developed in this work. This can be done using MATLAB 'System Identification Toolbox' to record actual in-flight system parameter. Also, implementing developed control loops on an actual platform could be really interesting.

Another interesting point for future research is the following: the response of the controller for altitude tracking is slower in the case when we applied airspeed input (refer to figure 5-10). To understand why this actually happens could be another research question.

¹Civilian UAVs are used for applications such as topographic mapping, glacier and ice sheet dynamics and thunderstorm research. They can also be used for surveillance missions to identify and reduce poaching or illegal logging.

Finally, although lateral controller was developed here, there are limitations on the time period that the plane can roll. To somehow overcome this problem would be another open area for further research.

6.6. FUTURE WORK

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix A

DATCOM

```
$FLTCON NMACH=1.0, MACH=0.04,
  NALPHA=13.0, ALSCHD(1)=-0.2, 0.0, 1.0, 2.0, 4.0, 6.0, 8.0, 10.0, 12.0, 14.0,
  16.0, 18.0, 20.0,
  NALT=1.0, ALT=150.0, WT=5.5701$
$OPTINS SREF=0.2712, CBARR=0.393, BLREF=0.81$
$SYNTHS XCG=0.175, ZCG=0.00098, XW=-0.015, ZW=0.0, ALIW=0.0,
  ALIH=0.0, XV=0.25, ZV=0.017, VERTUP=.TRUE.$
$WGPNF CHRDTP=0.33, SSPNE=0.355, SSPN=0.405,
  CHRDR=0.45, SAVSI=27.41,
  TWISTA=-1.0, DHDADI=0.0, TYPE=1.0$
NACA-W-4-0009
$BODY NX=6.0, X(1)=-0.03, -0.02, 0.0, 0.45, 0.47, 0.48,
  S(1)=0.0, 0.0005, 0.001062, 0.001062, 0.0005, 0.0, BNOSE=1.0, BLN=0.0259,
  BLA=0.0367$
$VTPLNF CHRDTP=0.05, SSPNE=0.11, SSPN=0.12, CHRDR=0.15, TYPE=1.0,
  SAVSI=40.1,
  SWAFP=0.0, TWISTA=0.0$
NACA-V-4-0005
$SYMFLP FTYPE=1.0, NDELTA=9.0, DELTA(1)=-12.0, -9.0, -6.0, -3.0, 0.0,
  3.0, 6.0, 9.0, 12.0,
  CHRDFO=0.05, CHRDFO=0.05, SPANFI=0.11, SPANFO=0.405$
DIM M
DAMP
DERIV RAD
SAVE
CASEID ELEVATOR ANALYSIS, SMARTFLY, CASE 1
NEXT CASE
$ASYFLP STYPE=4.0, NDELTA=9.0, SPANFI=0.11, SPANFO=0.405, CHRDFO=0.5,
  CHRDFO=0.05, DELTAL(1)=-12.0, -9.0, -6.0, -3.0, 0.0,
  3.0, 6.0, 9.0, 12.0, DELTAR(1)=12.0, 9.0, 6.0, 3.0, 0.0,
  -3.0, -6.0, -9.0, -12.0,$
DERIV RAD
CASEID AILERON ANAYLSIS, CASE 2
NEXT CASE
```

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix B

Meeting Minute



Minute kept at the meeting on Aug. 29th 2014. Meeting time 13.15–15:00

Attendees: Olle Hagner, Leonid Freidovich, Sven Rönnbäck, Shahriar Bagheri

Subject: This was the final meeting regarding the Master's thesis project, entitled: "Modeling and Control System Design for Civil UAV" which had officially been started on January 27th, 2014. Details of the meeting are as follows:

1. Opening of the meeting

Meeting declared open at 13:15

2. Assignments carried out

- Discussion about the work. Mainly the results obtained from the simulation model developed in Simulink;
- Deviations from the original project plan:
 - Original project plan was revised on June 18th 2014. It was agreed by the customer that no system implementation will be carried out and the thesis will focus only on computer simulations. In this meeting it was agreed by the customer that the work includes and meets all requirements, defined in the project plan **Version 2**.
- Comments/feedbacks/corrections regarding the current draft of the thesis. Both supervisors gave feedback on the thesis document.
- Suggestions for the defense session date. Following dates were suggested by the customer for the defense session:
 - Friday Sep. 19th,
 - Week 39 (between Sep. 22nd and 26th).

3. Coming assignments

- Olle is supposed to provide the current version of the SmartOne user manual, along with some pictures of SmartFly to Shahriar;
- Shahriar is supposed to revise the thesis according to the feedbacks from supervisors and resubmit it for further evaluation. Final document, after being approved by both supervisors will be submitted to the examiner (Sven) by the end of week 36 (before Friday the 5th 2014).
- Confirmation of the exact date, time and location for the defense session. This is indeed after the examiner reviews the work.

4. Meeting finish

- Meeting declared closed at 15:00

Signature
Shahriar Bagheri

Appendix C

MATLAB Script

```
%%%%%%%%%%%%%%
%Importing DATCOM Output into MATLAB
%%%%%%%%%%%%%
clear all;
close all;
clc;
% Storing all values from Datcom Into the Structure "SmartFly"
%
SmartFly=datcomimport('datcomRad.out', true, 0);
data=SmartFly{1}; % Just to have quick access to Data

% Since some values are missing for some conditions, we
% should replace them with the values that are available.
% This can be done by the
% following 'For loop'

aerotab = {'cyb' 'cnb' 'clq' 'cmq'};

for k = 1:length(aerotab)
```

```

for m = 1: data . nmach
    for h = 1: data . nalt
        data . ( aerotab {k} ) ( : ,m,h ) = data . ( aerotab {k} ) ( 1 ,m,
            h );
    end
end

%%%%%
%Plotting
%%%%%

% Lift Curve: Lift Vs. AoA
h1 = figure;
figtitle = { 'Lift Curve' ' ' };

plot ( data . alpha , data . cl ( : ,1 ) )
grid
ylabel ( [ 'Lift Coefficient (Mach =' num2str ( data . mach (1) )
    ') ' ] )
title ( figtitle {1} );

xlabel ( 'Angle of Attack (deg)' )

% Drag polar: Lift Vs. Drag
h2 = figure;
figtitle = { 'Drag Polar' ' ' };
plot ( data . cd ( : ,1 ) , data . cl ( : ,1 ) )
grid
ylabel ( [ 'Lift Coefficient (Mach =' num2str ( data . mach (1) )
    ') ' ] )

```

```
title( figtitle{1})  
xlabel( 'Drag Coefficient')  
  
%% Pitching Moment Plot: Mach No. Vs. Pitching moment  
h3 = figure;  
figtitle = {'Pitching Moment', ''};  
plot(data.cm(:,1),data.cl(:,1))  
grid  
ylabel(['Lift Coefficient (Mach =' num2str(data.mach(1))  
)'])  
title( figtitle{1})  
xlabel('Pitching Moment Coefficient')  
  
%% Pitching Moment Plot: AoA. Vs. Pitching moment  
h4 = figure;  
figtitle = {'Pitching Moment', ''};  
plot(data.alpha,data.cm(:,1))  
grid  
ylabel(['Lift Coefficient (Mach =' num2str(data.mach(1))  
)'])  
title( figtitle{1})  
xlabel('Pitching Moment Coefficient')  
  
%%%%%  
%% TO BE USED IN SIMULINK MODEL  
%%%%%
```

*% To Activate and test prelookup interpolation block in simulink I will create a matrix 13*2 one col. has alpha values , and one has Cl values monotonically increasing*

```

Alpha=data.alpha';
Alpha=Alpha*(pi/180); % Alpha in radian
Cl=data.cl;
CD=data.cd;
CM=data.cm;
Delta=data.delta';
Delta=Delta*(pi/180); % Delta in radian

Delta_r=data.deltar;
Delta_r=Delta_r*(pi/180);
Delta_l=data.deltal;
Delta_l=Delta_l*(pi/180);
DCl=data.dcl_sym;
DCm=data.dcm_sym;
DCd=data.dcdmin_sym;
Xcp=data.xcp;

CLroll=[-5.5056E-02 -4.2007E-02 -2.8005E-02 -1.4002E-02
0.0000E+00 ...
1.4002E-02 2.8005E-02 4.2007E-02 5.5056E-02]';

CN=[-4.549E-04 -3.471E-04 -2.314E-04 -1.157E-04 0.000E
+00 ...
1.157E-04 2.314E-04 3.471E-04 4.549E-04;
-3.037E-04 -2.317E-04 -1.545E-04 -7.724E-05 0.000E+00
...
7.724E-05 1.545E-04 2.317E-04 3.037E-04;
4.450E-04 3.395E-04 2.263E-04 1.132E-04 -0.000E+00 ...
-1.132E-04 -2.263E-04 -3.395E-04 -4.450E-04;

```

1.218E-03	9.292E-04	6.195E-04	3.097E-04	-0.000E+00	...
-3.097E-04	-6.195E-04	-9.292E-04	-1.218E-03;		
2.844E-03	2.170E-03	1.447E-03	7.234E-04	-0.000E+00	...
-7.234E-04	-1.447E-03	-2.170E-03	-2.844E-03;		
4.566E-03	3.484E-03	2.323E-03	1.161E-03	-0.000E+00	...
-1.161E-03	-2.323E-03	-3.484E-03	-4.566E-03;		
6.368E-03	4.859E-03	3.239E-03	1.620E-03	-0.000E+00	...
-1.620E-03	-3.239E-03	-4.859E-03	-6.368E-03;		
8.202E-03	6.258E-03	4.172E-03	2.086E-03	-0.000E+00	...
-2.086E-03	-4.172E-03	-6.258E-03	-8.202E-03;		
9.925E-03	7.573E-03	5.049E-03	2.524E-03	-0.000E+00	...
-2.524E-03	-5.049E-03	-7.573E-03	-9.925E-03;		
1.152E-02	8.789E-03	5.859E-03	2.930E-03	-0.000E+00	...
-2.930E-03	-5.859E-03	-8.789E-03	-1.152E-02;		
1.295E-02	9.879E-03	6.586E-03	3.293E-03	-0.000E+00	...
-3.293E-03	-6.586E-03	-9.879E-03	-1.295E-02;		
1.416E-02	1.081E-02	7.205E-03	3.602E-03	-0.000E+00	...
-3.602E-03	-7.205E-03	-1.081E-02	-1.416E-02;		
1.512E-02	1.154E-02	7.693E-03	3.847E-03	-0.000E+00	...
-3.847E-03	-7.693E-03	-1.154E-02	-1.512E-02];		

```

Delta_aileron=(-24:6:24);
Delta_aileron=Delta_aileron';
Delta_aileron=Delta_aileron*(pi/180);

```

```

Inertia=[0.14641 -0.000149 0; -0.000149 0.11995 0;0 0
0.26547]; % UAV Inertia Matrix

```

```

CN_trim=[4.566E-03 3.484E-03 2.323E-03 1.161E-03
-0.000E+00 ...]

```

-1.161E-03 -2.323E-03 -3.484E-03 -4.566E-03];

```
% Calculating Stability derivatives - longitudinal axis  
  
% Defining UAV Physical Parameters  
  
u = 9.67; % Velocity body x-axis  
w = 1.16; % Velocity body z-axis  
Ro = 1.225; % Air density  
S = 0.2712; % Wing Area / m^2  
alpha = 0.12; % AoA in Radiane  
theta = 0.12; % Pitch Angle in Radiane  
delta_e = 0.0279; % Trim condition elevator deflection  
delta_t = 0.57; % Trim condition Throttle opening  
m = 0.568; % Aircraft Mass /Kg  
Va = 9.74; % Trim condition Airspeed  
c = 0.33; % Chord  
Iy = 0.11995; % Inertia  
S_prop = 0.0314;  
C_prop=0.029;  
k = 20; % Engine constant  
g = 9.8; % Gravity  
  
% Finding required Aerodynamic derivatives  
  
Cm0 = 0;  
Cm_alpha = -0.4438;  
Cm_delta_e = -0.4253;  
Cm_q=0;
```

```

Cd0 = 0.029;
Cd_alpha=0.6;
Cd_q =0;
Cd_delta_e = 0.0005;

Cl0 = 0.26;
Cl_alpha = 3.4424;
Cl_q=0;
Cl_delta_e = 0.7335;

Cx0 = -Cd0*cos(alpha)+Cl0*sin(alpha);
Cx_alpha = -Cd_alpha*cos(alpha)+Cl_alpha*sin(alpha);
Cx_q = -Cd_q*cos(alpha)+Cl_q*sin(alpha);
Cx_delta_e = -Cd_delta_e*cos(alpha)+Cl_delta_e*sin(alpha);

Cz0 = -Cd0*sin(alpha)-Cl0*cos(alpha);
Cz_alpha = -Cd_alpha*sin(alpha)-Cl_alpha*cos(alpha);
Cz_q = -Cd_q*sin(alpha)-Cl_q*cos(alpha);
Cz_delta_e = -Cd_delta_e*sin(alpha)-Cl_delta_e*cos(alpha);

%%%%%%%%%%%%%
%Stability Derivatives for long. axis
%%%%%%%%%%%%%

Xu = (u*Ro*S)*(Cx0+Cx_alpha*alpha+Cx_delta_e*delta_e)/m - ...
      Ro*S*w*Cx_alpha/(2*m) - (Ro*S_prop*C_prop*u)/m ;

Xw = (w*Ro*S)*(Cx0+Cx_alpha*alpha+Cx_delta_e*delta_e)/m + ...
      Ro*S*Cx_alpha*u/(2*m)-(Ro*S_prop*C_prop*w)/m ;

Xq = -w + (Ro*Va*S*Cx_q*c)/(2*m) ;

```

```
Xdelta_e = (Ro*(Va^2)*S*Cx_delta_e)/(2*m) ;
Xdelta_t = (Ro*S_prop*C_prop*k*delta_t)/m ;

Zu = (u*Ro*S*(Cz0+Cz_alpha*alpha+Cz_delta_e*delta_e))/m - ...
      Ro*S*w*Cz_alpha/(2*m) ;
Zw = (w*Ro*S*(Cz0+Cz_alpha*alpha+Cz_delta_e*delta_e))/m + ...
      (Ro*S*Cz_alpha*u)/(2*m) ;
Zq = u + (Ro*Va*S*Cz_q*c)/(2*m) ;
Zdelta_e = (Ro*(Va^2)*S*Cz_delta_e)/(2*m) ;

Mu = (u*Ro*S*c*(Cm0+Cm_alpha*alpha+Cm_delta_e*delta_e))/Jy -
      ...
      (Ro*S*c*Cm_alpha*w)/(2*Jy) ;
Mw = (w*Ro*S*c*(Cm0+Cm_alpha*alpha+Cm_delta_e*delta_e))/Jy +
      ...
      (Ro*S*c*Cm_alpha*u)/(2*Jy) ;
Mq = (Ro*Va*S*(c^2)*Cm_q)/(2*Jy) ;
Mdelta_e = (Ro*(Va^2)*S*c*Cm_delta_e)/(2*Jy) ;

%%%%%%%%%%%%%
%State-Space Representation - Long.
%%%%%%%%%%%%%

A_long = [Xu Xw Xq+w -g*cos(theta) ;
           Zu Zw Zq-w -g*sin(theta) ;
           Mu Mw Mq      0          ;
           0     0     1      0          ] ;

B_long = [ Xdelta_e Xdelta_t ;
           Zdelta_e      0          ;
```

```

Mdelta_e      0      ;
               0      ] ;

C_long=eye(4) ;

D_long=zeros(4,2) ;

%%%%%%%%%%%%%
%%Transfer Functions %%
%%%%%%%%%%%%%

%%%%%%%%%%%%%
%Transfer function delta_e to input(s)
%%%%%%%%%%%%%
[Num,Den]=ss2tf(A_long,B_long,C_long,D_long,1) ;
TF_u_delta_e = tf(Num(1,:),Den) ;
TF_w_delta_e = tf(Num(2,:),Den) ;
TF_q_delta_e = tf(Num(3,:),Den) ;
TF_theta_delta_e = tf(Num(4,:),Den) ;

%%%%%%%%%%%%%
%Transfer function delta_t to input(s)
%%%%%%%%%%%%%
[Num_t,Den_t]=ss2tf(A_long,B_long,C_long,D_long,2) ;
TF_u_delta_t = tf(Num_t(1,:),Den_t) ;
TF_w_delta_t = tf(Num_t(2,:),Den_t) ;
TF_q_delta_t = tf(Num_t(3,:),Den_t) ;
TF_theta_delta_t = tf(Num_t(4,:),Den_t) ;

```

```
%%%%%%  
%Calculating Stability derivatives - Lateral axis  
%%%%%
```

```
m=0.568; % Aircraft Mass in Kg
```

```
Q=80.05; % Dynamic Pressure
```

```
S=0.2712; % Wing Area
```

```
Iy=0.11995; % Inertia
```

```
Iz=0.26547;
```

```
Ix=0.14641;
```

```
C_bar=0.39; % Mean Chrod Length
```

```
u0=11.432; % Aircraft Velocity
```

```
g=9.8;
```

```
Sw_over_S = 0.1087;
```

```
e=0.95;
```

```
AR=2.42;
```

```
tau=0.3;
```

```
b= 0.81;
```

```
theta0=5*pi/180;
```

```
%%%%%%
```

```
%Finding required Aerodynamic derivatives - Nelson Ch.3
```

```
%%%%%%
```

```
Cy_beta=-0.06972; % /rad
```

```
Cy_p=0.0539; %/rad
```

```
Cy_r=0;
```

```
Cy_delta_a=0;
```

```
Cn_beta=0.001002; % /rad
```

```
Cn_p= 0.0854; %/rad
```

Cn_r = -0.0154; %/rad
Cn_delta_a = -0.0069; % /rad

Cl_beta = -0.093; %/rad
Cl_p = -0.256; %/rad
Cl_r = 0.0545; % /rad
Cl_delta_a = 0.1318; %/rad

%%%%%%
%Stability Derivatives for lateral axis
%%%%%%%%%

Y_beta = (Q*S*Cy_beta) /m;
Y_p = (Q*S*b*Cy_p) / (2*u0*m) ;
Y_r = (Q*S*b*Cy_r) / (2*u0*m) ;
Y_delta_a = (Q*S*Cy_delta_a) /m;
Y_delta_r = 0;

N_beta = (Q*S*b*Cn_beta) /Iz ;
N_p = (Q*S*b*b*Cn_p) / (2*u0*Ix) ;
N_r = (Q*S*b*b*Cn_r) / (2*u0*Ix) ;
N_delta_a = (Q*S*b*Cn_delta_a) /Iz ;
N_delta_r = 0;

L_beta = (Q*S*b*Cl_beta) /Ix ;
L_p = (Q*S*b*b*Cl_p) / (2*u0*Ix) ;
L_r = (Q*S*b*b*Cl_r) / (2*u0*Ix) ;
L_delta_a = (Q*S*b*Cl_delta_a) /Ix ;
L_delta_r = 0;

```
%%%%%%
%State-Space Representation - Lateral
%%%%%

A_lat =[Y_beta/u0      Y_p -(u0-Y_r)      g*cos(theta0);
         L_beta/u0      L_p      L_r      0;
         N_beta/u0      N_p      N_r      0;
         0              1        0.121     0];

B_lat=[      0      Y_delta_r ;
            L_delta_a  L_delta_r ;
            N_delta_a  N_delta_r ;
            0          0      ] ;

C_lat=eye(4);

D_lat=zeros(4,2);

%%%%%
%Transfer Functions %
%%%%%

%%%%%
%Transfer function delta_a to input(s)
%%%%%

[Num_lat_1,Den_lat_1]=ss2tf(A_lat,B_lat,C_lat,D_lat,1);
TF_beta_delta_a = tf(Num_lat_1(1,:),Den_lat_1);
TF_p_delta_a = tf(Num_lat_1(2,:),Den_lat_1);
TF_r_delta_a = tf(Num_lat_1(3,:),Den_lat_1);
```

```
TF_phi_delta_a = tf(Num_lat_1(4,:),Den_lat_1);  
%  
% Transfer function delta_r to input(s)  
%  
[Num_lat_2,Den_lat_2]=ss2tf(A_lat,B_lat,C_lat,D_lat,2);  
TF_beta_delta_r = tf(Num_lat_2(1,:),Den_lat_2);  
TF_p_delta_r = tf(Num_lat_2(2,:),Den_lat_2);  
TF_r_delta_r = tf(Num_lat_2(3,:),Den_lat_2);  
TF_phi_delta_r = tf(Num_lat_2(4,:),Den_lat_2);
```

```
%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%
%Linear Quadratic Regulator - longitudinal axis
%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%
% Defining system matrix

A_long = [ -0.0543    -0.5332      0     -9.7295
           -2.7791   -10.3435     8.5100    -1.1732
          -0.3403    -2.0302       0         0
            0         0      1.0000      0 ];

B_long = [ 2.4224      0.0224
           -20.2054      0
          -18.4384      0
            0         0 ];

C_long = [-0.1193      0.9929       0     -9.7350;
           0.9929      0.1193       0         0 ];

D_long = zeros(2,2);

sys_long=ss(A_long,B_long,C_long,D_long);

Q_long = C_long'*C_long;
R_long=diag([5  0.1]);

[K_long,S,e] = lqr(A_long,B_long,Q_long,R_long);
Kf_long=K_long(1:2,1:4);
```

```
%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%
%Linear Quadratic Regulator - Lateral axis
%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%0%
% Defining system matrix

A_lat = [ -0.2331      0.0730    -11.4320      9.7627
          -0.9771     -1.0893      0.2319          0
          0.0058      0.3634     -0.0655          0
          0        1.0000       0.121          0 ];

B_lat = [0  15.8300 -0.4571  0] ';

C_lat = [0  0  0  1];

D_lat = zeros(1,1);

% LQR Design

Q_lat=diag([0  0  0  1]);
R_lat= 5000;

[K_lat,S,e] = lqr(sys_lat,Q_lat,R_lat);
Kf_lat=K_lat(1:4)
```

Bibliography

- [1] Fahlstrom, P.G. and Gleason, T. J. (2012), “*Introduction to UAV systems*”, JOHN WILEY.
- [2] Chao, H. Cao, Y. Chen, Y. Q. (2007), “*Autopilots for Small Fixed-Wing Unmanned Air Vehicles: A Survey*”, *Conference Proceedings*, IEEE International Conference on Mechatronics and Automation, Harbin, China, August, pp: 3144-3149.
- [3] Paw, Y. C. (2009), “*Synthesis and Validation of Flight Control for UAV*”, *PhD Dissertation*, Faculty of the Graduate School, University of Minnesota December.
- [4] “*SmartPlanes Company - Official web-page*”, Available from: <http://www.smartplanes.se/>. Date: Oct. 1st - 2014
- [5] Jafarov, T. (2014), “*Modeling and Landing Autopilot Design for Unmanned Aerial Vehicle (UAV)*”, *Master’s Thesis*, Department of Applied Physics and Electronics, Umeå University, to be submitted.
- [6] “*The Personal Aerial Mapping System - SmartPlanes Co.*”, Available from: <http://www.smartplanes.se/products/the-personal-aerial-mapping-system-pams/>. Date: Oct. 1st - 2014
- [7] Xue, C. Ganglin, W. Zhe, W. (2010), “*Flight Control System Design for the Flying-wing Aircraft by Using Intelligent Optimal Seeking Method*”, *Conference Proceedings*, 2nd International Asia Conference on Informatics in Control, Automation and Robotics, Wuhan, China, March. pp: 429-432.

BIBLIOGRAPHY

- [8] “*SmartOne User Manual*”, Prepared by: SmartPlanes AB. (2014),.
- [9] Nelson, R. C. (1998), “*Flight Stability and Automatic Control*”, 2nd Edition, McGraw Hill.
- [10] “*Advanced Aircraft Analysis - DAR Corporation.*”, Available from: <http://www.darcorp.com/Software/AAA/>. Date: Sep. 15th - 2014
- [11] “*Digital DATCOM*”, Available from: <http://www.pdas.com/datcom.html>.
- [12] Triputra, F. R. Trilaksono, B. R. Sasongko, R. A. Dahsyat, M. (2012), “*Longitudinal Dynamic System Modeling of a Fixed-Wing UAV towards Autonomous Flight Control System Development A Case Study of BPPT Wulung UAV Platform* ”, Conference Proceedings, International Conference on System Engineering and Technology, Bandung, Indonesia, September. pp: 1-6.
- [13] “*The USAF Stability and Control DATCOM*”, Volume 1, Users Manual, McDonnell Douglas Astronautics Company, April. 1979.
- [14] “*System Identification Toolbox - Mathworks MATLAB*”, Available from: <http://www.mathworks.se/products/sysid/>. Date: Oct. 17st - 2014
- [15] Rauf, A. Zafar, M. A. Ashraf, Z. Akhtar, H. (2011), “*Aerodynamic modeling and State-Space model extraction of a UAV using DATCOM and Simulink*”, Conference Proceedings, Conference on Computer Research and Development (ICCRD), Shanghai, March, pp: 88-92.
- [16] Mattio, A. (2006), “*Modeling and Control of the UAV Sky-Sailor*”, Master’s Project report, Swiss Federal Institute of Technology - Lausanne.
- [17] Zhiping, L. Fang, T. (2013), “*Model Derivation and Control System Simulation for Unmanned Aerial Vehicle*”, Conference Proceedings, 25th Chinese Control and Decision Conference (CCDC), Guiyang, China, May. pp: 4053-4057.

- [18] Mingxing, X. Xiaoping, Z. Zhou, Z. Bo, Z. (2013), “*Flight Control System Design for a Flying-Wing Aircraft*”, *Conference Proceedings*, TENCON - 2013, IEEE Region 10 Conference, Xi'an, China, October. pp: 1-4.
- [19] Lei, W. Lixin, W. (2012), “*Nonlinear Flight Control Design of a Combat Flying Wing with High Aspect-ratio*”, *Conference Proceedings*, 2nd IEEE International Conference on Computer Application and System Modeling.
- [20] Hussain, S. A. Kadri, M. B. (2013), “*Optimal Control Synthesis for UAV*”, *Conference Proceedings*, IEEE International Conference on Computer, Control and Communication(IC4), Karachi, Pakistan, September. pp: 1-6.
- [21] Low, C. B. (2010), “*A Trajectory Tracking Control Design for Fixed-wing Unmanned Aerial Vehicles*”, *Conference Proceedings*, IEEE International Conference on Control Applications, Part of 2010 IEEE Multi-Conference on Systems and Control Yokohama, Japan, September. pp: 2118-2123.
- [22] Ren, W. and Beard, R. W. (2004), “*A Trajectory Tracking Control Design for Fixed-wing Unmanned Aerial Vehicles*”, *IEEE Transaction on Control Systems Technology*, Vol. 12, No. 5, pp: 706-716.
- [23] Hostmark, J. B. (2007), “*Modeling Simulation and Control of Fixed-wing UAV: CyberSwan*”, Master of Science Thesis, Norwegian University of Science and Technology (NTNU), Trondheim, Norway.
- [24] “*MATLAB AND SIMULINK - Mathworks*”, Available from:
<http://www.mathworks.se/>. Date: Oct. 17th - 2014
- [25] “*Aerospace Blockset - MATLAB SIMULINK*”, Available from:
<http://www.mathworks.se/products/aeroblks/>. Date: Oct. 17th - 2014
- [26] Beard, R. W. McLain, T. W. (2012), “*Small Unmanned Aircraft: Theory and Practice*”, Princeton University Press.

BIBLIOGRAPHY

- [27] Freidovich, L. B. (2013), “*Optimal Control For Linear Systems: Lecture Notes*”, , National Research University (NRU) of Information Technologies, Mechanics and Optics (ITMO), St. Petersburg, Russia.
- [28] Zhou, K. Doyle, J. C. (1997), “*Essentials of Robust Control*”, Prentice Hall.
- [29] Goodwin, G. C. Graebe, S. F. Salgado, S. E. (2001), “*Control System Design*”, PRENTICE HALL.
- [30] “*IEEE Code of Ethics*”, Available from: <http://www.ieee.org/about/corporate/governance/p7-8.html>, Date: Oct. 7th - 2014
- [31] Lin, P. Bekey, G. Abney, K. (2009), “*Robots in War: Issues of Risk and Ethics*”, Ethics and Robotics.
- [32] Association for Unmanned Vehicle Systems International (AUVSI)(2012), “*The Benefits of Unmanned Aircraft Systems: Saving Time, Saving Money, Saving Lives*”.
- [33] Ingram, B. Jones, D. Lewis, A. Richards, M. Rich, C. Schachterle, L. (2010), “*A Code of Ethics for Robotics Engineers*”, ACM/IEEE International Conference on Human-Robot Interaction (HRI), Osaka, Japan. PRENTICE HALL.