DEVELOPMENT OF AN AUTOPILOT FOR AUTOMATIC LANDING OF AN UNMANNED AERIAL VEHICLE

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SEÇKİN ARIBAL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

JUNE 2011

Approval of the thesis:

DEVELOPMENT OF AN AUTOPILOT FOR AUTOMATIC LANDING OF AN UNMANNED AERIAL VEHICLE

submitted by **SEÇKİN ARIBAL** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen ___________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. İsmet Erkmen ___________
Head of Department, **Electrical and Electronics Engineering**

Prof. Dr. Kemal Leblebicioğlu ___________
Supervisor, **Electrical and Electronics Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Aydan Erkmen _____________
Electrical and Electronics Engineering Dept., METU

Prof. Dr. Kemal Leblebicioğlu _____________
Electrical and Electronics Engineering Dept., METU

Prof. Dr. Kemal Özgören _____________
Mechanical Engineering Dept., METU

Asst. Prof. Dr. Afşar Saranlı _____________
Electrical and Electronics Engineering Dept., METU

Asst. Prof. Dr. İlkay Yavrucuk _____________
Aerospace Engineering Dept., METU

Date: _____________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Seçkin Arıbal

Signature :

ABSTRACT

DEVELOPMENT OF AN AUTOPILOT FOR AUTOMATIC LANDING OF AN UNMANNED AERIAL VEHICLE

Arıbal, Seçkin
M.S., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Kemal Leblebicioğlu

June 2011, 157 Pages

This thesis presents the design of an autopilot and guidance system for an unmanned aerial vehicle. Classical (PID) and modern control (LQT, Sliding Mode) methods for autonomous navigation and landing in adverse weather conditions are implemented. Two different guidance systems are designed in order to navigate through waypoints during normal and/or emergency flight. The nonlinear Pioneer UAV model is used in controller development and simulations.

Aircraft is linearized at different trim points and total airspeed, altitude, roll and yaw autopilots are designed using Matlab/Simulink environment for lateral and longitudinal control of the aircraft. Gain scheduling is used to combine controllers designed for different trim points. An optimal landing trajectory is determined using “Steepest Descent” Algorithm according to the dynamic characteristics of the aircraft. Optimal altitude trajectory is used together with a lateral guidance against cross-wind disturbance.

Finally, simulations including landing under crosswind, tailwind, etc., are run and the results are analyzed in order to demonstrate the performance and effectiveness of the controllers.

Keywords: UAV, Autonomous Landing, Autopilot, Guidance.

ÖZ

İNSANSIZ BİR HAVA ARACININ OTOMATİK İNİŞİ İÇİN OTOPİLOT TASARIMI

Arıbal, Seçkin
Yüksek Lisans., Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Kemal Leblebicioğlu

Haziran 2011, 157 Sayfa

Bu tez çalışması, insansız bir hava aracı için otopilot ve güdüm sistemi tasarımını anlatmaktadır. Otonom iniş ve seyrüsefer amaçlı klasik (OİT) ve modern (KKD, DİDT) kontrol yöntemlerinin uygulamaları üzerinde çalışılmıştır. Nirengi noktalarını kullanarak, normal ve/veya acil durum seyrüseferi gerçekleştirmek için 2 farklı güdüm sistemi tasarlanmıştır. Kontrolcü geliştirme ve simulasyon çalışmalarında insansız bir hava aracı olan Pioneer'ın modeli kullanılmıştır.

Hava aracı ilk olarak farklı denge-uçuş noktalarında doğrusallaştırılmış, sonrasında hava aracının kontrolü için Matlab/Simulink ortamında hız, irtifa, yatış ve yön otopilotları tasarlanmıştır. Son olarak kazanç-tablolama yöntemi kullanılarak farklı denge-uçuş noktaları için geliştirilen kontrolcüler bir araya getirilmiştir. Hava aracının dinamik özelliklerine uygun olarak, "En Hızlı İniş" Yöntemi kullanılarak optimize edilmiş bir iniş güzergahı oluşturulmuştur. Optimize edilmiş irtifa komutları, yanal güdümcü ile beraber yan rüzgar koşullarına karşı kullanılmıştır.

Son olarak, yan rüzgar, kuyruk rüzgarı gibi zorlu hava koşulları altında iniş simulasyonları yapılmış ve kontrolcülerin performans ve etkinliği değerlendirilmiştir.

Anahtar Kelimeler: İHA, Otomatik İniş, Otopilot, Güdüm.

To My Wife,

To My Parents

ACKNOWLEDGMENTS

First and foremost, I would like to thank my beloved wife Gizem for her invaluable support throughout this process. Without her, this would have been much harder.

I would like to express my special thanks and appreciation to my supervisor Prof. Dr. Kemal Leblebicioğlu for his valuable comments, and expertly guidance throughout all stages of this study. Thanks are also due to Merve Hanköylü, for her contributions in this thesis.

I am grateful to Hakan Tiftikçi, Deniz Karakaş, Volkan Kargın, Burak Durmaz, Nebi Çallı and Yavuz Öztürk at TAI for their discussions and very helpful advices on the subject that guided me through. I would like to extend my thanks to Aykut Cihangir, Murat Özceylan, Selim Kocabay, Mehmet Karakaş, Salim Özgün, Çağrı Camcıoğlu, Mehmetcan Apaydın and Bahar Haser for their valuable friendship and support during this thesis.

Finally, I would like to thank to my parents for their life-long love, support and understanding.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

LIST OF SYMBOLS

X, Y, Z	Axes of earth fixed reference frame
x, y, z	Axes of body fixed reference frame
ϕ, θ, ψ	Euler angles (in roll, pitch and yaw planes)
u, v, w	Velocity components in body fixed reference frame
p, q, r	Components of angular velocity in the body fixed reference frame with respect to the earth fixed reference frame
$\vec{F}$	Sum of all externally applied forces to the aırcraft
$\vec{V}$	Total aircraft velocity
m	Aircraft mass
F_x, F_y, F_z	Components of total force acting on the aircraft expressed in the body frame
M_x, M_y, M_z	Components of moment acting on the aircraft expressed in the body frame
$\delta_a, \delta_e, \delta_r$	Control surface deflections (aileron, elevator, rudder)
α	Angle of attack
β	Sideslip angle
C_x	Drag force coefficient
C_y	Side force coefficient
C_z	Lift force coefficient
C_l	Rolling moment coefficient
C_m	Pitching moment coefficient
C_n	Yawing moment coefficient
I	Inertia matrix
I_{xy}, I_{xz}, I_{yz}	Cross inertia terms
I_x, I_x, I_y	Elements of inertia matrix
Ω	Angular velocity

CHAPTER 1

INTRODUCTION

According to operational experiences, most of the UAV mishaps occur due to human errors during manual landing and takeoff. UAV's include some expensive payloads, which constitute 50-80% of their total cost and besides that; it takes many years for the pilots to reach the adequate experience level. Here comes the importance of an automatic landing system. In addition to increasing the safety of flight, it reduces the workload of the pilot significantly during landing. At the same time, by improving the aircraft's wind limits it provides safe landing under strong winds.

Main functionalities of the autopilots are: making an unstable system stable and to shape the response of the system. In modern control theory, control methods like adaptive control, robust control and fuzzy-logic have emerged in order to cope with the uncertainties in nonlinear systems.

Landing autopilots mainly consist of an inner loop and an outer loop. The dynamics are controlled in inner loops and position and velocity are controlled in outer loop. The parameters used in these loops are obtained using inertial (gyro, accelerometer) and navigation (GPS) sensors. low-level control functional division can be made as follows:

- Roll rate control,
- Pitch rate control,
- Velocity control,
- Yaw/Steering rate control.

Decisions of in-air and on-ground modes are made using the signal coming from a weight-on-wheel (WOW) sensor.

1.1 Literature Survey

There are many studies in the literature related to automatic landing. Some of the studies have concentrated on traditional control methods, while some of them have concentrated on modern control methods. In the thesis submitted by Kargin [1]; the design of a lateral and longitudinal autopilot for the METU tactical UAV is described. Its purpose is to design an autopilot that can land the aircraft under adverse weather conditions without exceeding the desired limits. Here, the traditional proportional-integral (PI) and proportional-integral-derivative (PID) controllers are designed and tested. Cho et al. [2], tested completely automatic landing using Linear Quadratic Regulator (LQR) Controller with only 1 navigation sensor (Differential GPS). The results show that for a more accurate landing inertial sensors are also needed.

The automatic takeoff and landing system used in Heron UAV of Israel Aircraft Industries (IAI) is developed based on the accumulated experience collected in many years [3]. The primary sensor in this system is a DGPS augmented by a radar altimeter. The secondary sensor is a laser tracker that measures the altitude, azimuth and distance of the UAV. By combining the data obtained from aircraft and ground based sensors with the flight control algorithms, the lateral and vertical deviations are calculated using glide slope and flight control laws are applied. At a certain altitude, the UAV accomplishes the approach and pitch maneuvers. In lateral axes, decrab, deroll and runway steering maneuvers are performed until the UAV stops completely.

One of the most popular modern control method used in landing autopilots is the neural networks. In [4], genetic algorithms are used with a recurrent neural network to enhance the traditional automatic landing systems. It states that genetic algorithm makes the adaptation of the controller to different environments easy, and neural network is preferred for its adaptability, robustness and its applicability to hardware.

The research is mostly focused on making the landing autopilots smarter and more robust. For this purpose, fuzzy neural systems are developed [5]. The reason of applying modern control methods to landing autopilots is that, the landing operations are very challenging in terms of flight safety and it is aimed to bring an autopilot the ability to cope with uncertainties during landing. In addition to strong wind conditions, serious problems like sticking control surfaces can be encountered during landing. The traditional landing systems can be unstable or their performance can degrade in these situations. This creates dangerous occasions, during a flight phase that the aircraft has to strictly follow a trajectory. Liguni et al. [6] have designed an automatic landing system based on the real pilot responses / inputs. In order to train the system, an experienced pilot's inputs logged on a database are used and the developed model has generated responses for the aircraft's current state using these inputs. This technique requires a huge database. Rong et al. [7] have designed a dynamic fuzzy system called "Sequential Adaptive Fuzzy Inference System (SAFIS)". The system executes an online learning by adding or removing rules from the main control rules using the sensor data. Fuzzy logic based controllers are very strong tools for the landing under strong wind conditions.

1.2 Problem Statement

In this thesis, different control and guidance methods are studied and applied to a UAV for automatic landing. Main intention in designing the controllers is to be able to land a UAV safely without pilot commands. Landing maneuver can be studied in two different phases. Initial phase is the "Approach Phase", in which the aircraft descends with a constant glide slope. Second phase is the "Flare Phase", which is the final maneuver before the touchdown that the aircraft completes the transition from the straight approach path to the horizontal ground roll. In the first phase, the objective is to keep the aircraft on a straight line with a constant flight path angle. Longitudinal guidance and control of the aircraft will be in great importance here. However, in cases when a crosswind is present in the environment lateral guidance and control will be needed to keep the aircraft on track. In the second phase, importance of the longitudinal controller and guidance system increases with a maneuver that needs accuracy. As the height of the aircraft is low, the responses of the aircraft should be faster and adaptation to disturbances should be quicker.

In order to define a landing as safe, the safety margins for landing maneuver should be defined. These margins define the aircraft's allowed states during landing. Automatic landing systems of the UAVs are mostly based on DGPS, radar or laser to provide accurate positioning and correction data to the aircraft. In this thesis, landing zone is defined assuming that there is radar on ground sending relative position information to aircraft, and the radar has a limited line of sight. A line of sight angle is defined by taking the aircraft maneuvering capability into account, and it is used in determining a landing cone.

One important problem is to bring the aircraft to a position inside the cone with the desired states. Waypoints can be used to guide the aircraft to the desired landing cone. Waypoints can be defined to include altitude, velocity and/or position information of the aircraft at specified locations. Defining waypoints brings the necessity of coordinating waypoint transitions or aircraft flight modes.

1.3 Work Done

First of all, a nonlinear aircraft model is obtained. Autopilot design is accomplished by linearization of the nonlinear aircraft model around various equilibrium flight conditions. Some trivial assumptions are made during the application of this procedure. Gain scheduling is applied since the airspeed variations lead to performance degradation in autopilots. Then, autonomous landing is studied using the developed controllers. A landing trajectory is necessary for this purpose and it is generated by using a constant glide slope. An exponential flare maneuver is added to the end of the trajectory to get a smooth touchdown. Finally, landing under wind and emergency situations are studied. Flight zones for the safe flight and landing are designated, and a high-level logic is used to decide whether to land or abort landing.

Linearization, PID controller design, optimal and lateral track guidance design studies of this thesis are conducted in collaboration with Merve Hanköylü [30]. Same nonlinear aircraft model is used and the linearization procedures and tools are the same. Similar lateral and longitudinal PID controller architectures are selected; however, autopilot gains are re-calculated using Matlab/Simulink's Optimization Toolbox in this thesis to obtain better performance. As the aircrafts used are the same, the landing trajectory and the optimal guidance algorithm designs are developed in collaboration and the results are presented in both studies. Lateral track guidance algorithm in this thesis is modified such that it uses track axes, which is defined according to the desired aircraft track. This improved the wind performance of the system, since the ground speed is taken into account.

1.4 Organization of the Thesis

This thesis is organized as follows: Chapter 2 describes the nonlinear 6-DOF aircraft dynamics, trimming and linearization processes. The nonlinear aircraft model is obtained in Matlab/Simulink and in order to design the controller linearized at calculated trim points. Chapter 3 presents the autopilot design details. Four different autopilot designs are presented in this chapter. Chapter 4 addresses the guidance system design. One longitudinal and two lateral guidance systems are designed. In Chapter 5, different wind disturbance scenarios are generated during landing and controller performances are analyzed. Chapter 6 presents conclusions and recommendations for future work.

CHAPTER 2

AIRCRAFT DYNAMICS

2.1 Platform

The purpose of this thesis is to design an autopilot and guidance system for the unmanned aerial vehicle (UAV) AAI RQ-2 Pioneer (will be mentioned as Pioneer for the rest of this thesis). Pioneer had been used by the United States Navy, Marine Corps, and Army, where it is deployed at sea and on land from 1986 until 2007. At first, it is placed aboard battleships to provide gunnery spotting, but then its mission evolved into reconnaissance and surveillance, primarily for amphibious forces [8]. It is powered with a two-cylinder two-stroke engine, which is outfitted with a 29-inch propeller. It can be launched by rocket assist, by catapult, or from a runway. It flies with a gimbaled EO/IR sensor, relaying analog video in real time via a C-band line-of-sight (LOS) data link. This thesis is based on the runway landing capability of the Pioneer.

Figure 1 - AAI RQ-2 Pioneer

The specifications of the Pioneer are shown in Table 1.

Table 1 - Pioneer Specifications

GVW	451.9 pounds
Fuel	47 liters 100 Octane AVGAS
Length	14 feet (4.26m)
Width	16.9 feet (5.15m)
Height	3.3 feet (1m)
Engine	26-HP magneto ignition, crankcase scavenged, horizontally opposed, simultaneous firing two-stroke directly coupled to a 29-inch fixed 18 degree pitch wooden laminate propeller.
Service Ceiling	12,000 feet
Absolute Ceiling	15,000 feet
Maximum Range	185+ KM
Maximum Endurance	5+ hours
Maximum Authorized Airspeed	110 KIAS (Knots Indicated Airspeed)
Minimum Speed	55 KIAS (Still Air) 60 KIAS (Rough Air) 65 KIAS (MIAG Autopilot software limit)
Stall Speed	40-45 KIAS
Cruise Speed	70 KIAS

2.2 Reference Coordinate Frames

Different reference frames can be used to deal with the motion of an aircraft, such as stability coordinate frame, wind coordinate frame, earth fixed coordinate frame (inertial frame) and body coordinate frame. In the Pioneer model, coordinate frame transformation is done and all the results are calculated in body coordinate frame, except x, y and z positions of the aircraft which are represented in inertial exes. The axes of inertial reference frame are represented by X, Y and Z. Here X axis points towards north, Z axis points downwards to earth's center, and Y axis is the complementing orthogonal axis found by the right hand rule. Body fixed reference frame has its origin at the aircraft's center of mass or geometric center and its axes are x, y and z. Earth and body reference coordinate frames are represented as in Figure 2.

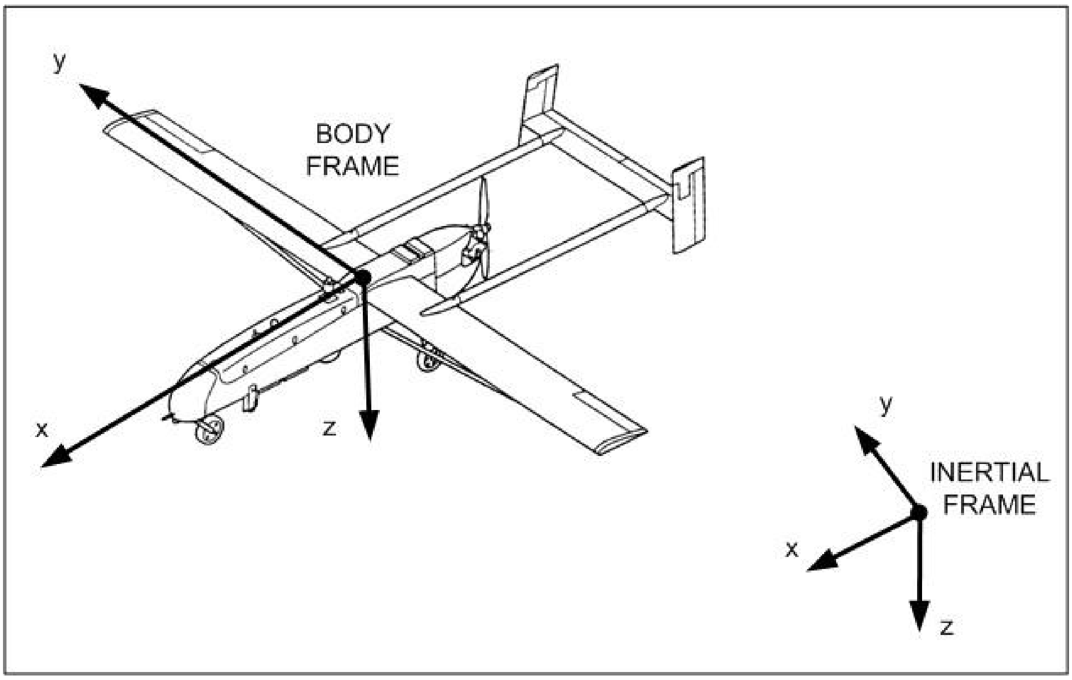

Figure 2 Earth and Body Axes

2.3 Aerodynamic Model

The first step in designing a controller for any physical system is to characterize the dynamics of that system [9]. Dynamics of the Pioneer taken from an open-source resource is used in this thesis, which is mostly based on FDC toolbox [10], [11].

The dynamics can be decoupled as lateral dynamics and longitudinal dynamics, where lateral dynamics are the aircraft's response along roll and yaw axis and longitudinal dynamics are the response of aircraft along pitch axis. Lateral modes are generally excited with aileron and rudder inputs, while longitudinal modes are excited with elevator and throttle. Generalized body dynamics block diagram is given in Figure 3 [11].

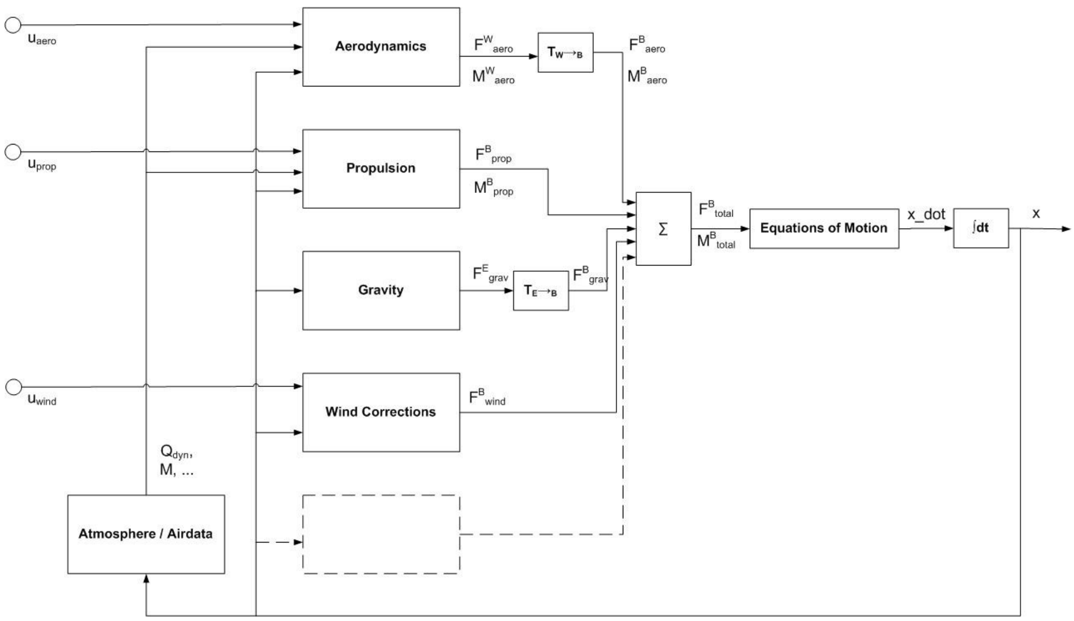

Figure 3 - Block Diagram of Generalized Body Dynamics [11]

The aerodynamic model expresses the aerodynamic force and moment coefficients in terms of non-linear polynomial functions of the state variables and aerodynamic control inputs. Aerodynamic force and moment coefficients measured in the body-fixed frame defined in [11] as follows:

$$C_D = C_{X_0} + C_{X_\alpha}\alpha + C_{X_{\alpha^2}}\alpha^2 + C_{X_{\alpha^3}}\alpha^3 + C_{X_q}\frac{q\bar{c}}{V} + C_{X_{\delta_r}}\delta_r + C_{X_{\delta_f}}\delta_f + C_{X_{\alpha\delta_f}}\alpha\delta_f \tag{2.1}$$

$$C_S = C_{Y_0} + C_{Y_\beta}\beta + C_{Y_p}\frac{pb}{2V} + C_{Y_r}\frac{rb}{2V} + C_{Y_{\delta_a}}\delta_a + C_{Y_{\delta_r}}\delta_r + C_{Y_{\alpha\delta_r}}\alpha\delta_r \tag{2.2}$$

$$C_L = C_{Z_0} + C_{Z_\alpha}\alpha + C_{Z_{\alpha^3}}\alpha^3 + C_{Z_q}\frac{q\bar{\bar{c}}}{V} + C_{Z_{\delta_e}}\delta_e + C_{X_{\delta_e\beta^2}}\delta_e\beta^2 + C_{Z_{\delta_f}}\delta_f + C_{Z_{\alpha\delta_f}}\alpha\delta_f \tag{2.3}$$

$$C_l = C_{l_0} + C_{l_\beta}\beta + C_{l_p}\frac{pb}{2V} + C_{l_r}\frac{rb}{2V} + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r + C_{l_{\alpha\delta_a}}\alpha\delta_a \tag{2.4}$$

$$C_m = C_{m_0} + C_{m_\alpha}\alpha + C_{m_{\alpha^2}}\alpha^2 + C_{m_q}\frac{q\bar{c}}{V} + C_{m_{\delta_e}}\delta_e + C_{m_{\beta^2}}\beta^2 + C_{m_r}\frac{rb}{2V} + C_{m_{\delta_f}}\delta_f \tag{2.5}$$

$$C_n = C_{n_0} + C_{n_\beta}\beta + C_{n_p}\frac{pb}{2V} + C_{n_r}\frac{rb}{2V} + C_{n_{\delta_a}}\delta_a + C_{n_{\delta_r}}\delta_r + C_{n_q}\frac{q\bar{c}}{V} + C_{n_{\beta^3}}\beta^3 \tag{2.6}$$

The subscripts a, e, r of the δ terms denote the aileron, elevator and rudder deflections, respectively.

The dimensionless C_i coefficients are defined as:

C_D : Drag force coefficient

C_S : Side force coefficient

C_L : Lift force coefficient

C_l : Rolling moment coefficient

C_m : Pitching moment coefficient

C_n : Yawing moment coefficient

2.4 Nonlinear Equations of Motion

The aircraft equations of motion can be derived from basic Newtonian mechanics. During the derivation of translational and rotational dynamic equations of motion, mass and inertia terms will be taken as constant, because mass and moment of inertia changes more slowly when compared with the translational and rotational velocities [12].

For the derivation of translational dynamic equations, following equality is used:

$$F = m(\frac{\partial V}{\partial t} + \Omega x V) \qquad (2.7)$$

Here, $F = [F_x\ F_y\ F_z]^T$ represents the total external applied forces vector. The external forces are defined as aerodynamic forces, gravity forces, engine thrust, landing gear (ground reaction) and wind disturbance. $V = [u\ v\ w]^T$ is the total aircraft velocity vector and m is the aircraft mass. $\Omega = [\mathrm{p\ q\ r}]^{\mathrm{T}}$ is the angular velocity vector about the c.g.

The rotational dynamic equations is derived from the following equation:

$$M = \frac{\partial (I.\Omega)}{\partial t} + \Omega \times (\mathrm{I} \cdot \Omega) \qquad (2.8)$$

Here, M = [L M N]^T represents the sum of all external moments vector. External moments are defined as engine, landing gear (ground reaction) moment. *I* is the inertia tensor, which is defined as:

$$I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \tag{2.9}$$

Using (2.7) and (2.8), translational and rotational dynamics can be found:

$$u_e = \frac{F_x}{m} - qw_e + rv_e \tag{2.10}$$

$$v_e = \frac{F_y}{m} + pw_e - ru_e \tag{2.11}$$

$$w_e = \frac{F_z}{m} - pv_e + qu_e \tag{2.12}$$

$$\dot{p} = P_{pp}p^2 + P_{pq}pq + P_{pr}pr + P_{qq}q^2 + P_{qr}qr + P_{rr}r^2 + P_lL + P_mM + P_nN \tag{2.13}$$

$$\dot{q} = Q_{pp}p^2 + Q_{pq}pq + Q_{pr}pr + Q_{qq}q^2 + Q_{qr}qr + Q_{rr}r^2 + Q_lL + Q_mM + Q_nN \tag{2.14}$$

$$\dot{r} = R_{pp}p^2 + R_{pq}pq + R_{pr}pr + R_{qq}q^2 + R_{qr}qr + R_{rr}r^2 + R_lL + R_mM + R_nN \tag{2.15}$$

where P_{pp}, P_{pq}, … R_n are inertia coefficients derived from the matrix multiplications involving the inertia tensor I [11], which have been given in Figure 4.

symbol	definition
$\|I\|$	$I_{xx}I_{yy}I_{zz} - 2J_{xy}J_{xz}J_{yz} - I_{xx}{J_{yz}}^2 - I_{yy}{J_{xz}}^2 - I_{zz}{J_{xy}}^2$
I_1	$I_{yy}I_{zz} - {J_{yz}}^2$
I_2	$J_{xy}I_{zz} + J_{yz}J_{xz}$
I_3	$J_{xy}J_{yz} + I_{yy}J_{xz}$
I_4	$I_{xx}I_{zz} - {J_{xz}}^2$
I_5	$I_{xx}J_{yz} + J_{xy}J_{xz}$
I_6	$I_{xx}I_{yy} - {J_{xy}}^2$
P_l	$I_1 \ / \ \|I\|$
P_m	$I_2 \ / \ \|I\|$
P_n	$I_3 \ / \ \|I\|$
P_{pp}	$-(J_{xz}I_2 - J_{xy}I_3) \ / \ \|I\|$
P_{pq}	$(J_{xz}I_1 - J_{yz}I_2 - (I_{yy} - I_{xx})I_3) \ / \ \|I\|$
P_{pr}	$-(J_{xy}I_1 + (I_{xx} - I_{zz})I_2 - J_{yz}I_3) \ / \ \|I\|$
P_{qq}	$(J_{yz}I_1 - J_{xy}I_3) \ / \ \|I\|$
P_{qr}	$-((I_{zz} - I_{yy})I_1 - J_{xy}I_2 + J_{xz}I_3) \ / \ \|I\|$
P_{rr}	$-(J_{yz}I_1 - J_{xz}I_2) \ / \ \|I\|$
Q_l	$I_2 \ / \ \|I\|$
Q_m	$I_4 \ / \ \|I\|$
Q_n	$I_5 \ / \ \|I\|$
Q_{pp}	$-(J_{xz}I_4 - J_{xy}I_5) \ / \ \|I\|$
Q_{pq}	$(J_{xz}I_2 - J_{yz}I_4 - (I_{yy} - I_{xx})I_5) \ / \ \|I\|$
Q_{pr}	$-(J_{xy}I_2 + (I_{xx} - I_{zz})I_4 - J_{yz}I_5) \ / \ \|I\|$
Q_{qq}	$(J_{yz}I_2 - J_{xy}I_5) \ / \ \|I\|$
Q_{qr}	$-((I_{zz} - I_{yy})I_2 - J_{xy}I_4 + J_{xz}I_5) \ / \ \|I\|$
Q_{rr}	$-(J_{yz}I_2 - J_{xz}I_4) \ / \ \|I\|$
R_l	$I_3 \ / \ \|I\|$
R_m	$I_5 \ / \ \|I\|$
R_n	$I_6 \ / \ \|I\|$
R_{pp}	$-(J_{xz}I_5 - J_{xy}I_6) \ / \ \|I\|$
R_{pq}	$(J_{xz}I_3 - J_{yz}I_5 - (I_{yy} - I_{xx})I_6) \ / \ \|I\|$
R_{pr}	$-(J_{xy}I_3 + (I_{xx} - I_{zz})I_5 - J_{yz}I_6) \ / \ \|I\|$
R_{qq}	$(J_{yz}I_3 - J_{xy}I_6) \ / \ \|I\|$
R_{qr}	$-((I_{zz} - I_{yy})I_3 - J_{xy}I_5 + J_{xz}I_6) \ / \ \|I\|$
R_{rr}	$-(J_{yz}I_3 - J_{xz}I_5) \ / \ \|I\|$

Figure 4 - Definition of Inertia Coefficients [11]

2.5 Control Surface Actuators System Model

The dynamics of the control surface actuator system is also included in the overall simulation model. Its dynamics is expressed as a first order system as shown with (2.14):

$$\frac{\delta(s)}{\delta_{cmd}(s)} = \frac{a}{s+a} \tag{2.14}$$

Deflection rate of the actuators are limited, and also there is physical limitation for the deflection amounts. The deflection rates for all control surface actuators are taken as $\mp 150°/sec$, and the physical deflection limit is taken as $\mp 30°$. The modeling of the control surfaces is given in Figure 5.

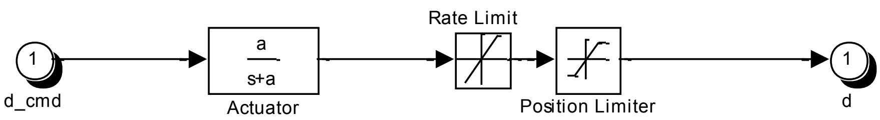

Figure 5 Control Surface Actuator Block Diagram

2.6 Linearization

The controller design methodology in this thesis is first linearizing the nonlinear system and then designing the controller for the linear model, then applying the controller to nonlinear model with the necessary modifications. Aircraft is linearized at a trim point. A trim point is an equilibrium point that the total forces and moments acting on an aircraft is zero. Trim points of Pioneer is found using "trim" function of Matlab. It is observed that the airspeed parameter is the most dominant parameter among others for Pioneer aircraft. For this reason, Pioneer is linearized at two different airspeed conditions suitable for landing, which are V = 30 m/s and V = 60 m/s. The altitude is selected as 60 m. and the gamma is selected as zero. The first trim condition is used for landing phase of the flight, and the second trim condition is used for the cruise phase. It is possible to choose more trim points; however, the controller's performance is checked after applying gain scheduling, and it turned up to be satisfactory for the airspeed values in between.

In order to test the trim points, aircraft states are set to the trim values, and trim inputs are applied from the input channels. It is expected that the roll, pitch and yaw rates (p,q,r) are zero or very close to zero. Figure 6, Figure 7 and Figure 8 prove us that the aircraft is well trimmed.

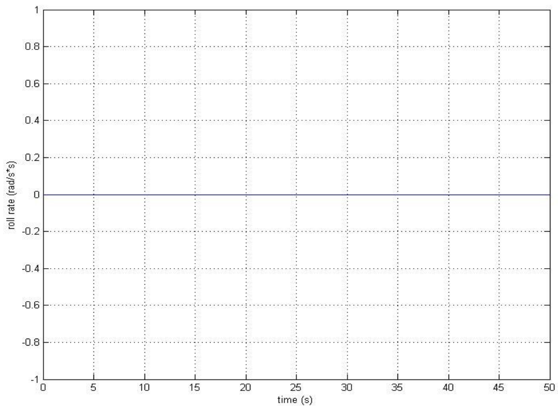

Figure 6 - Roll Rate for Trim Input

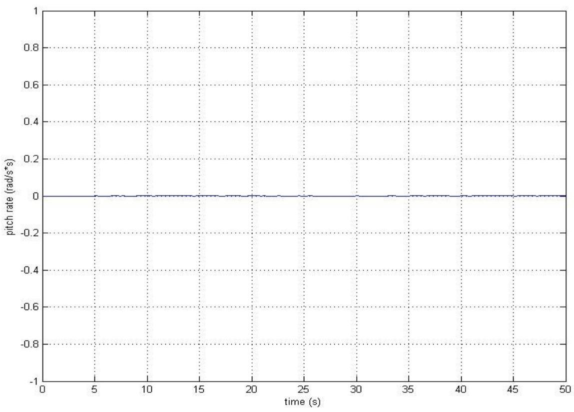

Figure 7 - Pitch Rate for Trim Input

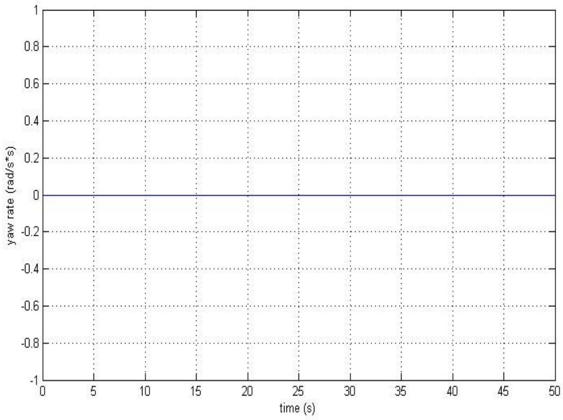

Figure 8 - Yaw Rate for Trim Input

After trimming the aircraft, Matlab's "Linear Analysis" utility is used to linearize the aircraft at the trim points. Linearization can be defined finding the linear approximation to a function at a given point, in our case trim point.

2.7 Linear Aircraft Model

In modeling and simulation of the aircraft, the state space representation is used in order to apply the modern control theory principles.

$$\dot{x} = Ax + Bu \tag{2.15}$$
$$y = Cx + Du$$

The selected states (x) for the autopilot design are given below:

$$x = [V \quad \alpha \quad \beta \quad p \quad q \quad r \quad \psi \quad \theta \quad \phi \quad x \quad y \quad z]^T \tag{2.16}$$

The states are total airspeed, angle of attack (alpha), angle of sideslip (beta), roll rate, pitch rate, yaw rate, yaw angle (psi), pitch angle (theta), roll angle (phi), x-axis position of the aircraft in flat-earth axes, y-axis position of the aircraft in flat-earth axes and z-axis position of the aircraft in flat-earth axes, respectively. Trim states and inputs of the aircraft are found as:

$X0 = [60\ \text{-}0.0293682\ 0\ \ \ 0\ 0\ 0\ \ \ 0\ \text{-}0.0293682\ 0\ \ \ 0\ 0\ 60]$ (2.17)

$U0 = [309.8742\ 0\ 0\ 0\ 0\ 0\ 0.145603\ 0\ 0\ 0]$

$X0 = [30\ 0.1753\ 0\ \ \ 0\ 0\ 0\ \ \ 0\ 0.1753\ 0\ \ \ 0\ 0\ 60]$ (2.18)

$U0 = [210.104\ 0\ 0\ 0\ 0\ 0\ \text{-}0.1009\ 0\ 0\ 0]$

The outputs are the same as states, with one addition, gamma.

$$y = [V \quad \alpha \quad \beta \quad p \quad q \quad r \quad \psi \quad \theta \quad \phi \quad x \quad y \quad z \quad \gamma]^T \tag{2.19}$$

The A, B, C and D matrices are found as stated in previous chapter using Matlab's "Linear Analysis" utility. The state space matrices are as follows:

1. For the case; V = 60 m/s and h = 60 m:

$$A = \begin{bmatrix} -0.05 & -4{,}13 & 0 & 0 & 8e^{-14} & 0 & 0 & -9.8 & 0 & 0 & 0 & 1.5e^{-4} \\ -0.005 & -2.61 & 0 & 0 & 0.98 & 0 & 0 & -8e^{-13} & 0 & 0 & 0 & 1.57e^{-5} \\ 0 & 0 & -0.44 & -0.02 & 0 & -0.99 & 0 & 0 & 0.16 & 0 & 0 & 0 \\ 0 & 0 & -20.10 & -12.97 & 0 & 8.1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5e^{-6} & -79.2 & 0 & 0 & -6.25 & 0 & 0 & 0 & 0 & 0 & 0 & -1.4e^{-8} \\ 0 & 0 & 32.4 & -0.57 & 0 & -2.94 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -0.02 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1e^{-9} & 0 & 0 & 0 & 0 & 0 & 1e^{-10} & 0 & 0 & 0 & 0 \\ 0 & 0 & 60 & 0 & 0 & 0 & 60 & 0 & 1.76 & 0 & 0 & 0 \\ -1e^{-13} & -60 & 0 & 0 & 0 & 0 & 0 & 60 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.005 & 0 & -1.54e^{-4} & 0 & 0 & 0 & -0.5854 & 0 & 0 & 0 \\ 2.56e^{-6} & 0 & 8.74e^{-5} & 0 & 0 & 0 & -0.2174 & 0 & 0 & 0 \\ 0 & 8.74e^{-5} & 0 & 0 & 0 & 0 & 0 & 0.1035 & 0 & 0 \\ 0 & 0 & 0 & 0.021 & 0 & -0.0013 & 0 & -110.53 & 2.165 & 0 \\ 0 & 0 & 0 & 0 & 0.011 & 0 & -65.77 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.0013 & 0 & 0.009 & 0 & 12.31 & -26.38 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$D = [0]$$

2. For the case; V = 30 m/s and h = 60 m:

$$A = \begin{bmatrix} -0.07 & 6.31 & 0 & 0 & 4e^{-12} & 0 & 0 & -9.8 & 0 & 0 & 0 & 1.05e^{-4} \\ -0.02 & -1.29 & 0 & 0 & 0.98 & 0 & 0 & -5e^{-13} & 0 & 0 & 0 & 3.17e^{-5} \\ 0 & 0 & -0.18 & 0.17 & 0 & -0.98 & 0 & 0 & 0.32 & 0 & 0 & 0 \\ 0 & 0 & -5.02 & -6.48 & 0 & 4.05 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.11 & -19.8 & 0 & 0 & -3.12 & 0 & 0 & 0 & 0 & 0 & 0 & 1.59e^{-4} \\ 0 & 0 & 8.10 & -0.28 & 0 & -1.47 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.01 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0.17 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -8e^{-10} & 0 & 0 & 0 & 0 & 0 & 8e^{-11} & 0 & 0 & 0 & 0 \\ 0 & 0 & 30 & 0 & 0 & 0 & 30 & 0 & -5.23 & 0 & 0 & 0 \\ -7e^{-13} & -30 & 0 & 0 & 0 & 0 & 0 & 30 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.005 & 0 & 9e^{-4} & 0 & 0 & 0 & -0.1463 & 0 & 0 & 0 \\ -3e^{-5} & 0 & 1e^{-4} & 0 & 0 & 0 & -0.1087 & 0 & 0 & 0 \\ 0 & 1e^{-4} & 0 & 0 & 0 & 0 & 0 & 0.0518 & 0 & 0 \\ 0 & 0 & 0 & 0.02 & 0 & -0.0013 & 0 & -27.63 & 0.541 & 0 \\ 0 & 0 & 0 & 0 & 0.011 & 0 & -16.44 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.001 & 0 & 0.009 & 0 & 3.079 & -6.59 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$D = [0]$$

The result is verified by comparing the angular rate and Euler angle outputs of the linear and nonlinear models for aileron doublet input.

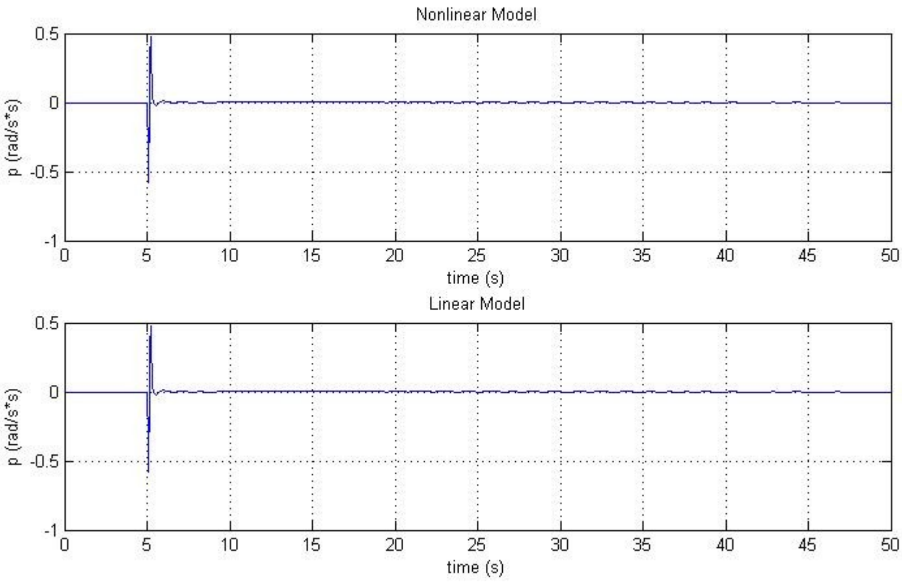

Figure 9 - Roll Rate Comparison

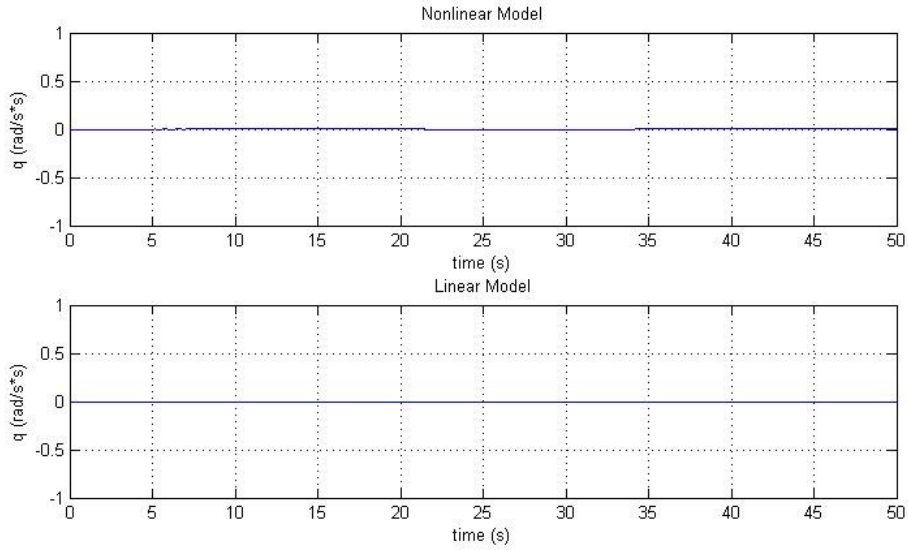

Figure 10 – Pitch Rate Comparison

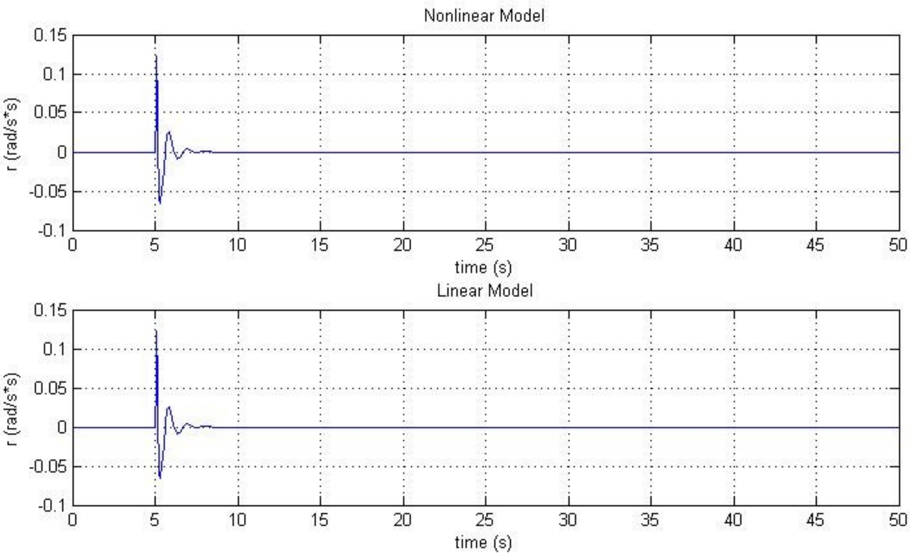

Figure 11 - Yaw Rate Comparison

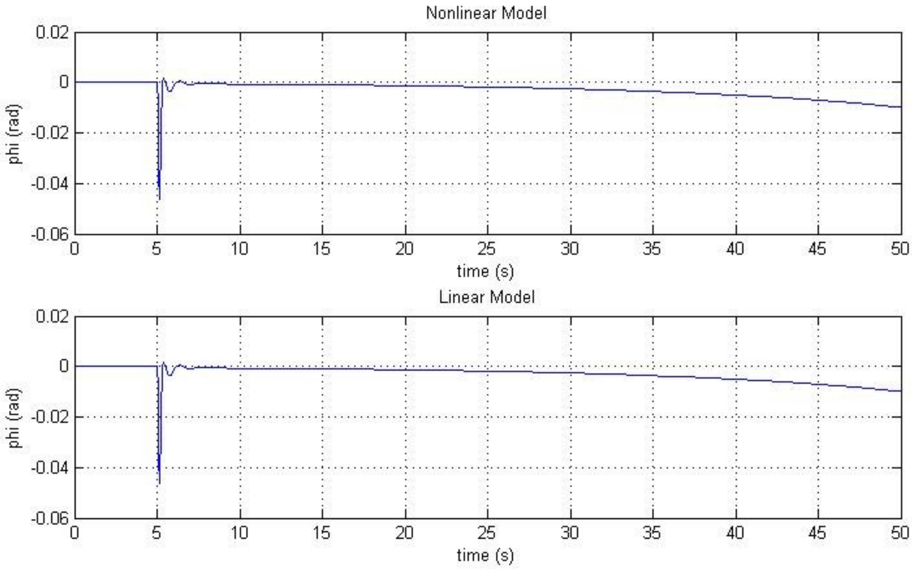

Figure 12 - Roll Angle Comparison

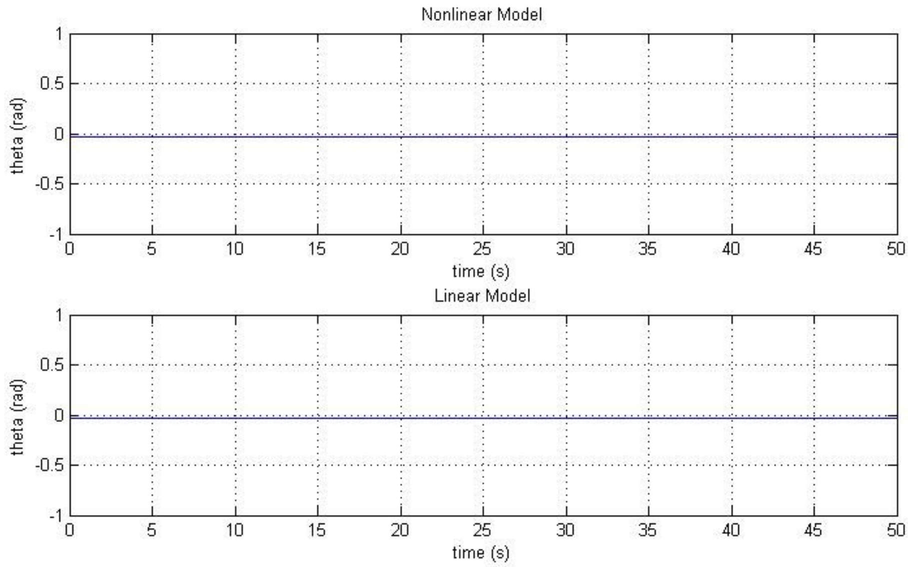

Figure 13 - Pitch Angle Comparison

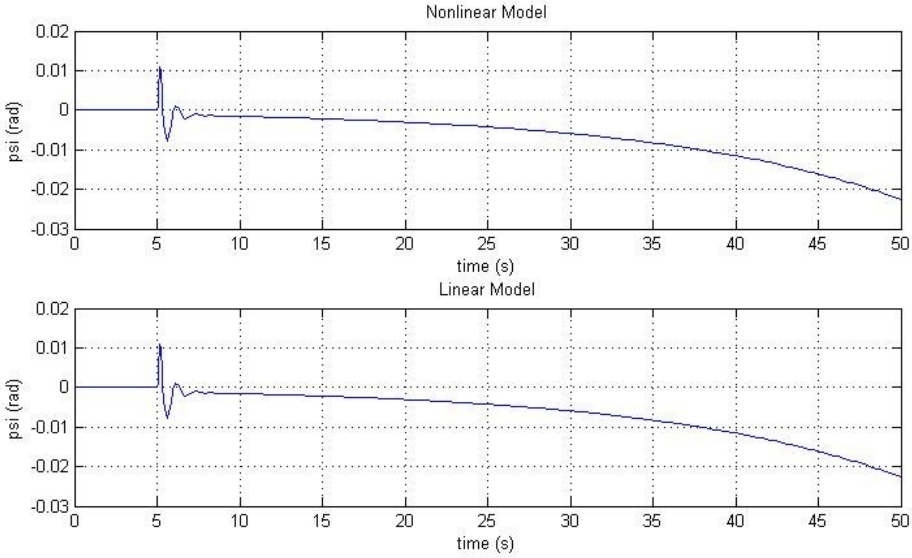

Figure 14 - Yaw Angle Comparison

The comparison shows that the linear and nonlinear aircraft angular rates and Euler angles are the same under same input. This proves that the linearization procedure is successful.

2.7.1 Longitudinal Dynamics

The longitudinal states are

$$x = \begin{bmatrix} u \\ \alpha \\ q \\ \theta \end{bmatrix} \tag{2.20}$$

Longitudinal matrix A_{long} is

$$A_{long} = \begin{bmatrix} -0.054 & -4.13 & 8.57e-14 & -9.8 \\ -0.0055 & -2.61 & 0.98 & -8.6e-13 \\ 5.18e-6 & -79.23 & -6.253 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

The longitudinal motion is described by two different oscillatory modes, namely the short period mode and the phugoid mode. Eigenvalues of the $\mathrm{A_{long}}$ matrix contain information about the longitudinal modes of the system. Both modes are stable since the real parts of their roots are negative.

Table 2 - Longitudinal Mode Characteristics

	Root Location	Natural Frequency (rad/s)	Damping Ratio
Short Period	-4.4385 ± 8.6199i	9.7	0.458
Phugoid	-0.0242 ± 0.2113i	0.213	0.114

2.7.2 Lateral Dynamics

The lateral states are

$$x = \begin{bmatrix} \beta \\ p \\ r \\ \varphi \end{bmatrix}$$

Lateral matrix A_{lat} is

$$A_{lat} = \begin{bmatrix} -0.443 & -0.0294 & -0.999 & 0.1634 \\ -20.103 & -12.97 & 8.101 & 0 \\ 32.4047 & -0.5784 & -2.94 & 0 \\ 0 & 1 & -0.029 & 0 \end{bmatrix}$$

The lateral modes are the roll mode, spiral mode and the Dutch-roll mode. Eigenvalues of the A_{lat} matrix contain information about the lateral modes of the system.

Dutch roll and roll modes are stable since the real parts of their roots are negative. Spiral mode is unstable. The spiral mode consists mainly of yawing at almost zero sideslip with some rolling.

Table 3 - Lateral Mode Characteristics

	Root Location	Natural Frequency (rad/s)	Damping Ratio
Dutch Roll	-1.81 ± 5.75i	6.03	0.3
Roll	-12.8		
Spiral	0.0669		

Figure 15 shows the effect of rudder on psi angle of the aircraft. To address these instabilities, the aircraft will require stabilization in the yaw axis.

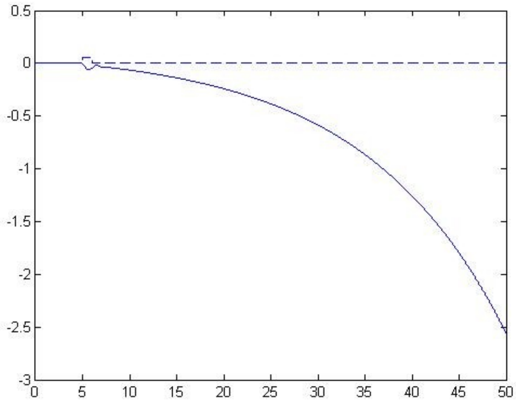

Figure 15 - Step Response: Psi (solid) to Rudder (dashed)

The autopilot design process can begin after the successful verification of the state-space form of the system.

CHAPTER 3

AUTOPILOT DESIGN

The autopilot is designed using the state-space model obtained in the previous section. The closed loop system including the autopilot can be presented in block diagram representation as in Figure 16.

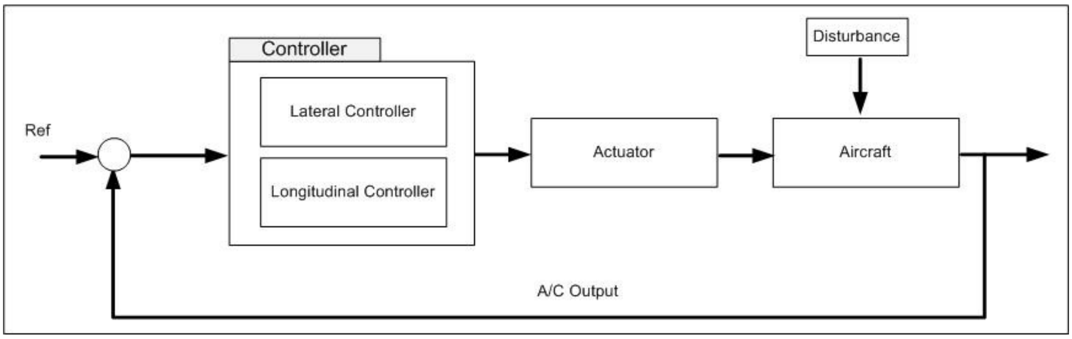

Figure 16 Closed Loop System Representation

Autopilot block in itself is designed as lateral and longitudinal autopilots. Lateral autopilot is dealing with yaw and roll control of aircraft, whereas the longitudinal autopilot is dealing with airspeed and pitch control of the aircraft. The lateral autopilot uses the aileron and rudder and the longitudinal uses the elevator and the engine thrust as a means of control. As it can also be observed from Figure 16 the autopilot commands are fed to actuator block and then to the aircraft plant. In the design of the controllers, it is assumed that all of the states are available and no noise exists on the sensors.

In this chapter, the design of the Proportional-Integral-Derivative (PID) controller for different trim points will be presented. This will bring in the concept of gain scheduling and its effects on the system. After that, Linear Quadratic Tracker (LQT) and Sliding Mode Controller (SMC) designs are presented.

3.1 Gain Scheduling

Gain scheduling is an approach so as to enable well established linear design methods to apply to nonlinear problems. Gain-scheduling design typically employs a divide and conquer approach whereby the nonlinear design task is decomposed into a number of linear sub-tasks. Such a decomposition depends on establishing a relationship between a nonlinear system and a family of linear systems.

The aircraft is needed to be in specific states for landing. In our case, these states mainly depend on heading, altitude and airspeed. The aircraft enters the landing phase with an initial altitude, which decreases until the aircraft touches down, and the airspeed and the heading of the aircraft is needed to be in a predefined range. Trimming the aircraft at one point, linearizing aircraft model and designing controllers according to that linear model can sometimes be not enough for different aircraft states. This means that the controller developed on linear model cannot be used for nonlinear model for another state of the aircraft.

Gain scheduling is used to overcome this situation. As the controller with existing gains starts not to control the aircraft beyond a specific state the linearizing procedure is repeated, and new gain values are obtained.

One or more observable variables, called the scheduling variables, are used to determine what operating region the system is currently in and to enable the appropriate linear controller. In this study, only the airspeed is selected as the scheduling variable, because of its dominance with respect to other parameters.

3.2 Proportional-Integral-Derivative (PID) Controller

Aim of a control system is to obtain a desired response for a given system. Feedback control is a way of controlling a system, where the controller determines the input signal to the process by using the measurement of the output signal [13]. The purpose of the feedback loop is to keep the process variable close to the desired value in spite of disturbances.

The PID controller has several important functions such as providing feedback, eliminating steady state offsets via integral action, and anticipating future through derivative action [14]. PID control law consists of three types of control actions: a proportional action, an integral action and derivative action.

Proportional action is given by the following expression:

$$u\,(t) = K_p e(t) = K_p(r(t) - y(t)) \qquad (3.1)$$

where K_p is the proportional gain. It simply increases the control variable when the control error is large. However, it introduces the system a steady state error. Here the integral action is presented to the system.

$$u(t) = K_i \int_0^t e(\tau)d\tau \qquad (3.2)$$

where K_i is the integral gain. While proportional action deals with the current value of the error, and the integral action deals with the past values, the derivative action is based on the predictions of the future values of the error. The expression for the derivative action is:

$$u(t) = K_d \frac{de(t)}{dt} \qquad (3.3)$$

where K_d is the derivative gain. The combination of proportional, derivative and integral actions is described with the following transfer function:

$$C_i(s) = K_p(1 + \frac{1}{T_i s} + T_d s) \qquad (3.4)$$

where K_p is the proportional gain, Ti is the intergral action time constant and T_d is the derivative action time constant.

3.2.1 PID Controller Design Details

This part of the thesis will detail the design of a control system that will stabilize the Pioneer aircraft. Inner and outer loop structure will be discussed, and finally the closed loop simulation results will be given.

The lateral modes are excited with the aileron and rudder inputs, and the longitudinal modes are as a result of the elevator inputs and thrust. For the aircraft to be navigated autonomously, it must be capable of navigating through waypoints or a predefined path. This requires that the altitude, heading and airspeed should be controlled. Following figure shows the block diagram for the control architecture:

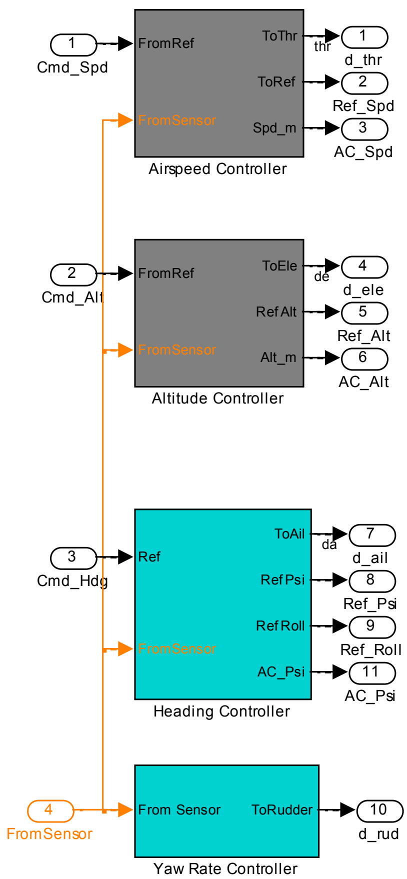

Figure 17 - Control Architecture

The aileron, rudder and elevator commands are controlled using inner PID structures that stabilize roll, and yaw. The autopilot is divided into two: Longitudinal controller and lateral controller.

Lateral Control

The lateral controller is used to control yaw rate, heading and roll angle. Three inner and one outer loop is used for this purpose. The inner loops are:

- Aileron Command from Roll Angle: This loop generates aileron command from the roll angle error with a PI controller.

- Aileron Command from Roll Rate: This loop generates aileron command from the roll rate error.

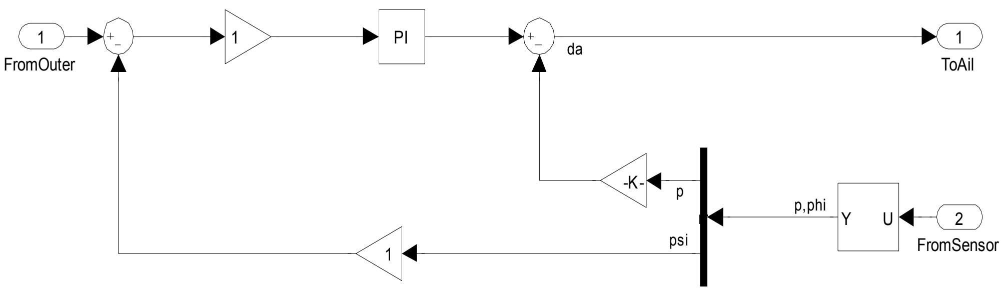

Figure 18 - Inner Loop: Roll and Roll Rate Controller

Aircraft roll angle with respect to a step input from aileron channel is given in Figure 19.

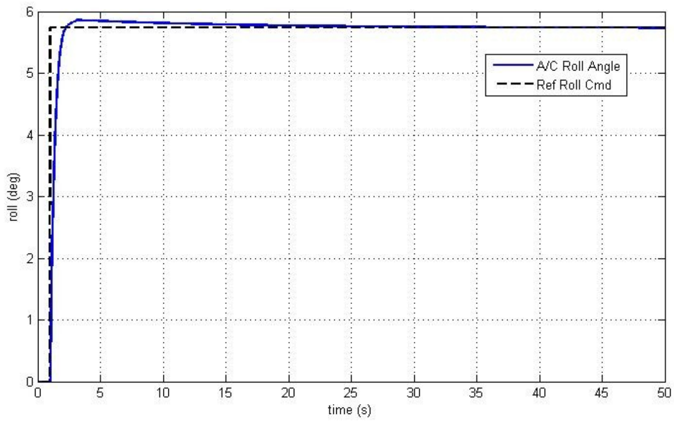

Figure 19 – Roll Angle wrt Step Input from Aileron Channel (PID)

As the roll controller is in the inner loop, a fast and accurate performance is expected. Steady state error and overshoot must be low. There are no specific requirements are defined for overshoot, undershoot or settling time performance for the controllers; however, during the gain selection process Matlab/Simulink's "Design Optimization Toolbox" is used. The desired overshoot value is set to 10%, and settling time is set to 7.5 seconds for step input with amplitude 1 and the objective is met in most of the controller responses. Roll controller performance is important especially under crosswind, therefore the performance of the roll controller is observed under 10 m/s constant crosswind. Constant roll command of 15 degrees is given as an input to the controller. Tracking performance is important here, and tracking error is expected to be low. The result shows that controller is successful in tracking the desired reference command even under strong wind.

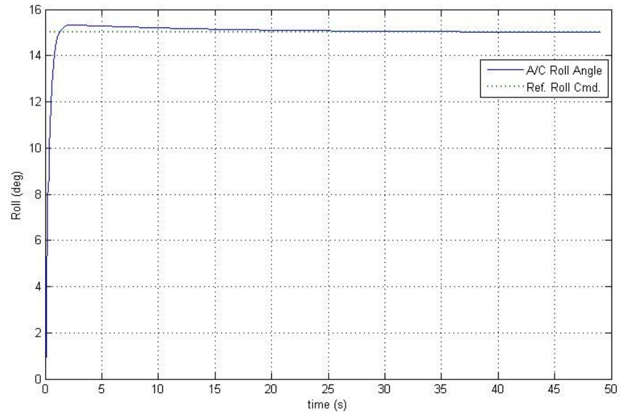

Figure 20 - Roll Angle wrt Step Input from Aileron Channel (10 m/s Crosswind - PID)

- Yaw Damper: This loop generates a rudder command from yaw rate error. Usually a yaw damper does not remove the effect of the initial disturbance in yaw rate, since there are nonzero steady state values. In addition, such a system tends to reject and oppose any change in yaw rate so as to change the aircraft's heading. To avoid such an opposition, the feedback signal is first passed through a wash-out block [15]. Rudder command is not used to make turns in this thesis. For this reason, yaw rate is commanded as 0° to the controller at all times.

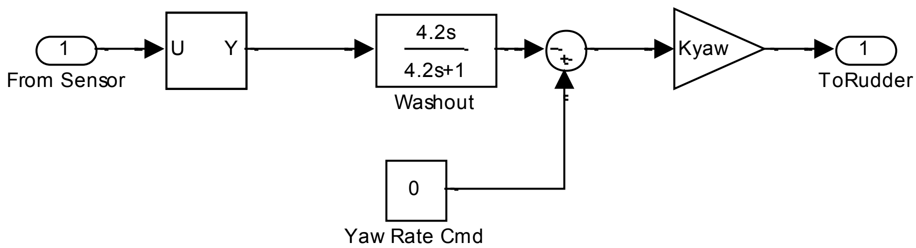

Figure 21 - Inner Loop: Yaw Rate Controller

The washout block is defined as:

$$W/O = \frac{s}{s+\tau} \qquad (3.5)$$

The constant τ is chosen by trial and error method as '10'. Applying the same roll angle step input to aileron channel, the psi angle response of the aircraft due to coupling is given below:

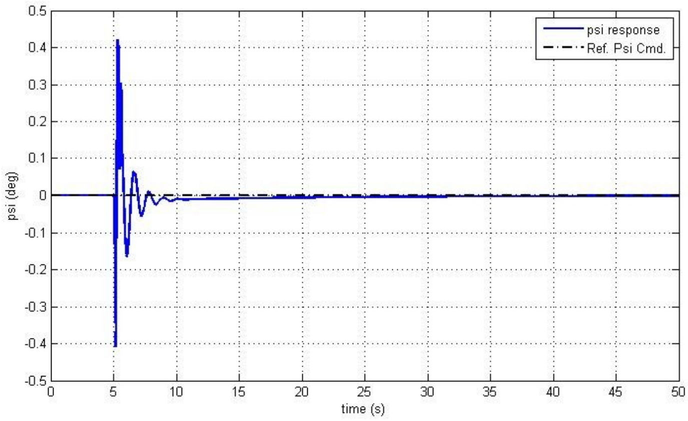

Figure 22 - Psi Angle wrt Step Input from Aileron Channel and $0°$ Psi Command (PID)

The change in the psi angle is observed during roll maneuver, which is at acceptable levels, and after the maneuver is complete the psi angle reaches zero as commanded.

The gain values obtained for the lateral controller inner loops are given below:

Table 4 - Lateral Controller Inner Loop Gain Values

	Controller Gains
Kpr	-2.986
Kir	-0,1955
KPr	-0,9
Kyaw	0.0526

The outer loop is:

- Roll command from heading: A roll command to the inner loop is generated from the heading error using a PI controller. The derivative term here is added to the heading command in order to speed up the response. The saturation block limits the roll command that can be generated by the heading controller to 30°. Yaw angle of the aircraft could also be controlled through the rudder channel; however, it is observed that the aircraft's responses are slower over rudder channel, and aileron is selected as the control surface for lateral control.

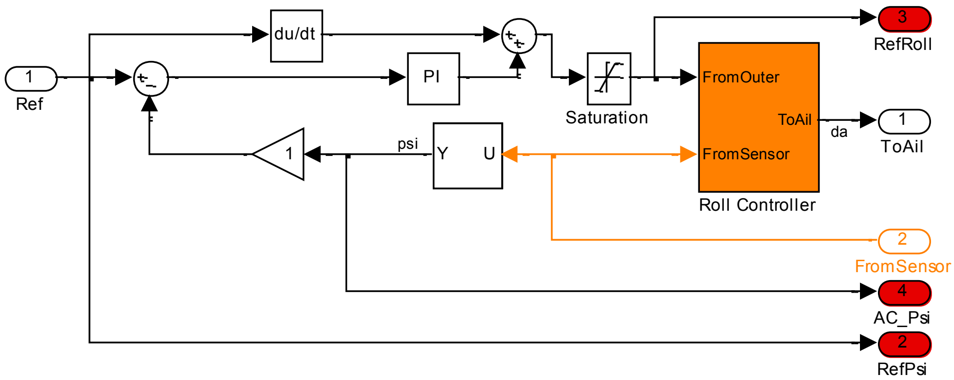

Figure 23 - Outer Loop: Heading Controller

Aircraft Psi angle with respect to a smooth step input from aileron channel is given below:

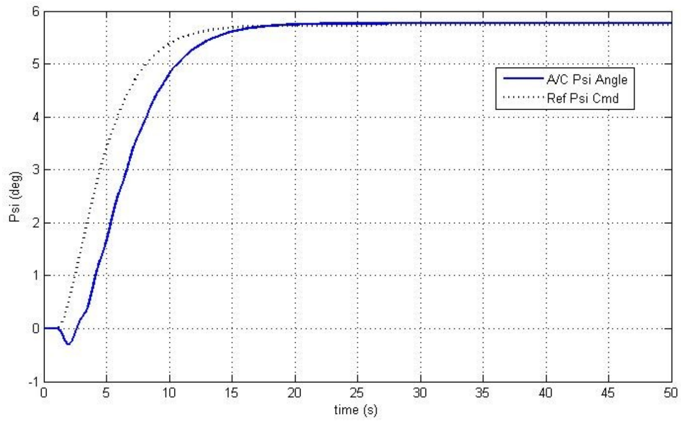

Figure 24 - Psi Angle wrt Step Input from Aileron Channel (PID)

Aircraft's psi response to aileron commands is observed to be slow, which is also related with the adverse yawing effect due to aileron deflection. This response can be acceptable for the cruise scenarios that the waypoints are far from each other. For landing scenarios, controller architecture is modified such that the guidance directly produces roll commands to controller for SM and LQT Controllers. Detailed simulation results and analysis are given in Chapter 5.

The gain values obtained for the lateral controller outer loop are as follows:

Table 5 - Lateral Controller Outer Loop Gain Values

	Controller Gains
Kph	2.3901
Kih	0.0092

Longitudinal Control

The longitudinal controller is for controlling the aircraft total airspeed and altitude. Three inner loops and one outer loop are used for the longitudinal control. The inner loops are:

- Elevator command from pitch angle: An elevator command is generated from the pitch angle error.
- Elevator command from pitch rate: An elevator command is generated from the pitch rate error.

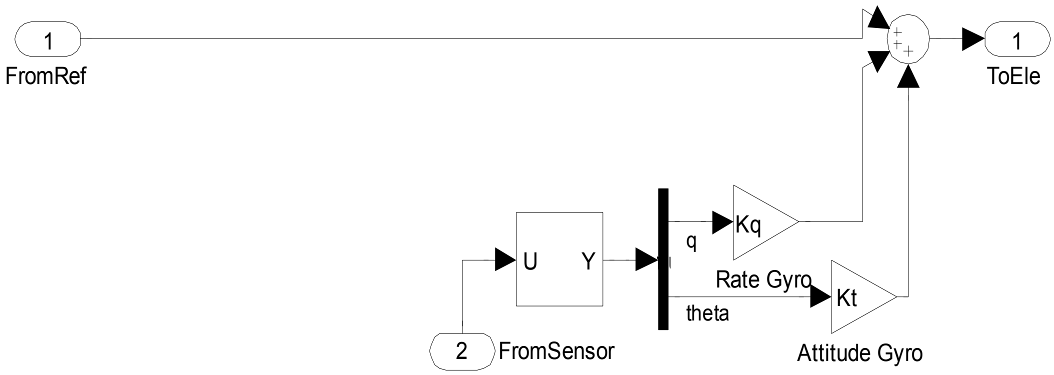

Figure 25 - Inner Loop: Pitch Angle and Pitch Rate Controller

- Throttle(Thrust) command from airspeed: A thrust command is generated from the airspeed error using a PID controller.

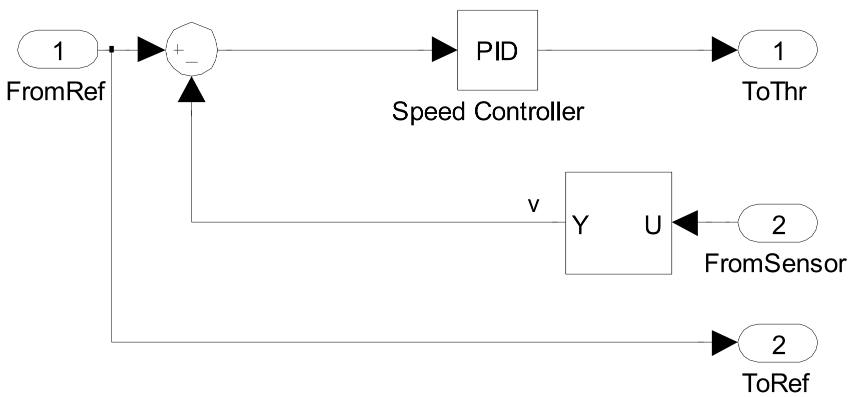

Figure 26 - Inner Loop: Airspeed controller

In classical controller architecture, airspeed (or velocity) is controlled by the elevator commands, and altitude is controlled by the throttle commands. The architecture that is used in this thesis is different; however, this architecture works well in landing. This also provides independent velocity control using throttle without changing the altitude. A reference airspeed step input is applied from thrust channel and the aircraft airspeed response is given in Figure 27. The airspeed command is chosen to be changing very fast to generate a condition similar to the case of waypoint navigation. It is observed that the aircraft's response is much slower than the input; however, this kind of an input is not used in landing, and this will be the case in normal flight. If the waypoint frequency is set according to this performance, unsafe situations will not be encountered.

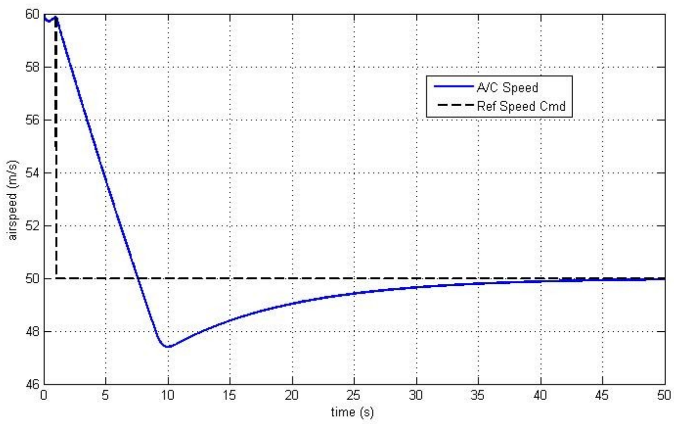

Figure 27 - Aircraft Airspeed Response to Reference Airspeed Input (PID)

The gain values obtained for the longitudinal controller inner loop are given below:

Table 6 - Longitudinal Controller Inner Loop Gain Values

	Controller Gains
Kps	300
Kis	28.5
Kds	14.7925
Kq	1
Kt	1

The outer loop is:

- Pitch from altitude: This loop produces a pitch command for the inner loop from the altitude error. Pitch command that can be generated by the altitude controller is limited to ∓10°.

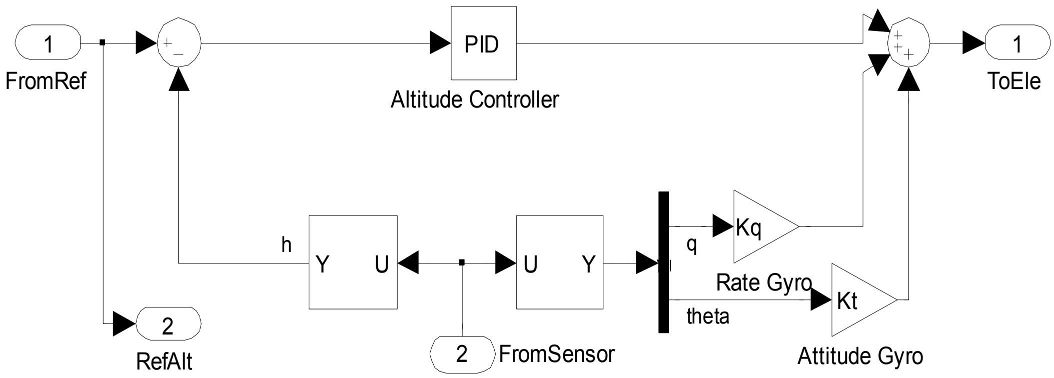

Figure 28 - Outer Loop: Altitude Controller

Step input is applied to elevator channel and the corresponding aircraft response is given in Figure 29. The reference input is selected considering the waypoint navigation.

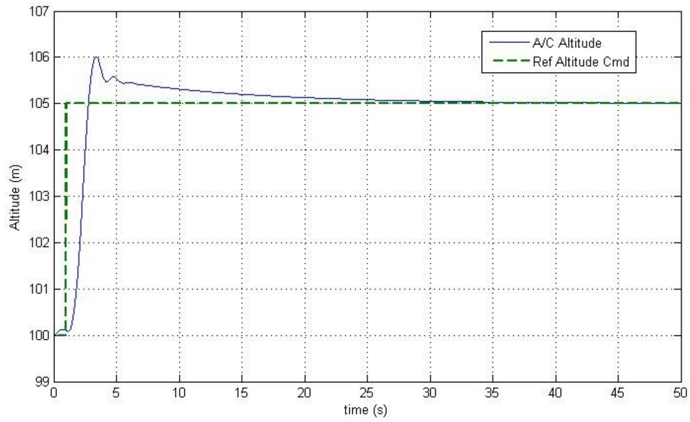

Figure 29 - Aircraft Altitude wrt Altitude Reference Command on Elevator Channel (PID)

Aircraft's pitch attitude in order to reach the desired altitude is given below:

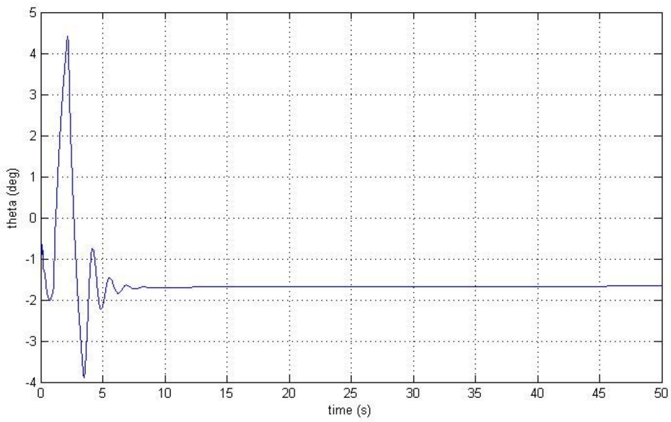

Figure 30 - Aircraft Pitch Attitude Response to Altitude Step Input (PID)

Altitude response will be crucial during landing, for this reason a reference input same as the landing trajectory is applied, and the performance is observed in Figure 31.

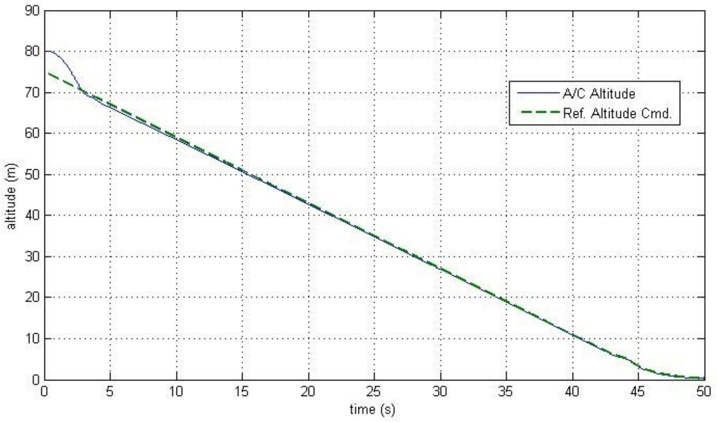

Figure 31 - Altitude Controller Response for Landing Trajectory as the Input (PID)

The gain values obtained for the longitudinal controller outer loop are given in Table 7, here for the altitude channel, the controller could not control the system with the same gain values for different trim conditions. For this reason, gain scheduling is applied and the required gain tuning is made.

Table 7 - Longitudinal Controller Outer Loop Gain Values

Controller Gains	**V = 30 m/s**	**V = 60 m/s**
Kpa	-0.1781	-0.2935
Kda	-0.5174	-0.1895
Kia	-0.0943	-0.0242

The results show that 2 trim points are enough. The values chosen as trim points are the minimum and maximum airspeed of the aircraft, the points between these numbers are found by interpolation. Altitude controller gains changing with airspeed is shown in Figure 32, Figure 33 and Figure 34.

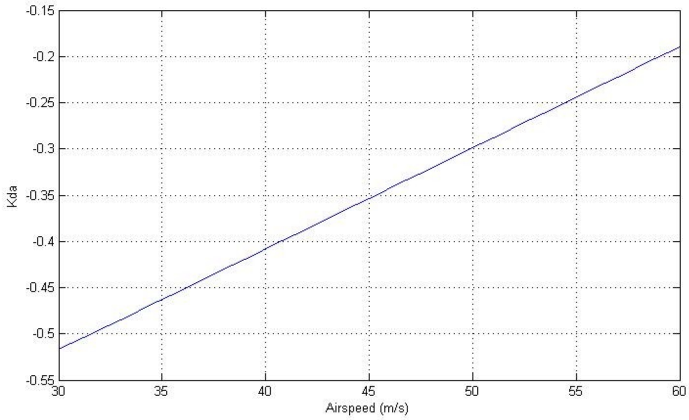

Figure 32 - Kda vs Airspeed

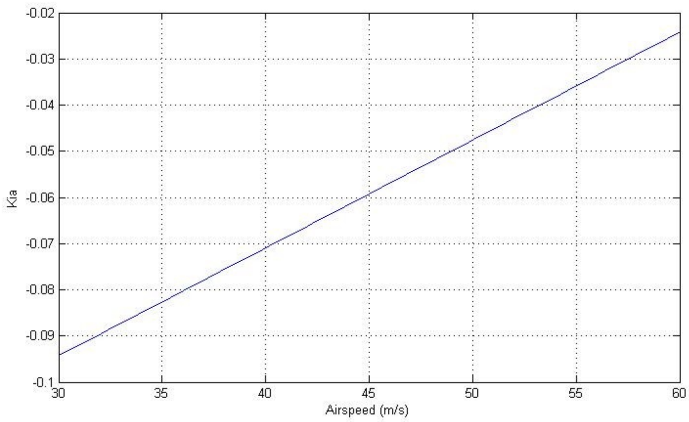

Figure 33 - Kia vs Airspeed

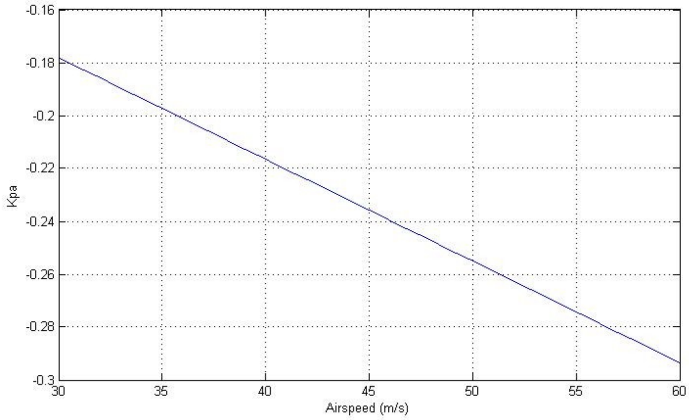

Figure 34 - Kpa vs Airspeed

3.3 Linear Quadratic Tracker (LQT)

LQT design procedures have a usage and philosophy that are very similar to the Linear Quadratic Regulator (LQR) methodology. The theory is simply to find a controller that provides the best possible performance with respect to some given measure of performance. Due to its robustness to disturbance against the PID controllers, LQT control method is selected in this thesis. Choosing the gains using a methodology that is known to be optimum in linear case is also preferable and beneficial with respect to choosing the gains using trial and error as in PID control. Robustness property is what is needed for the landing problem, as the problem includes wind disturbance. Not much difference is expected for the cases without wind disturbance; however, under wind disturbance the LQT controller's performance is expected to be superior to PID in response time and settle time. LQ controllers are also known for their appropriateness to MIMO systems; however, for this study SISO approach is chosen for its simplicity in application. MIMO approach is considered as a future work for this thesis. The case where the system dynamics are described by a set of linear differential equations and the cost is described by a quadratic functional is called the LQ problem. The LQR problem can be stated for a continuous linear time-invariant system defined by:

$$\dot{x}(t) = Ax(t) + Bu(t); \qquad x(t=0) = x_0 \tag{3.6}$$

with x(t) ϵ $\boldsymbol{R^n}$ and u(t) ϵ $\boldsymbol{R^m}$. A quadratic cost functional is defined as:

$$J = \frac{1}{2}\int_0^{\infty}[x^TQx + u^TRu]dt \tag{3.7}$$

Then a control law can be defined to minimize the cost functional. If certain conditions are met, LQR guarantees nominally stable closed loop systems and achieves guaranteed levels of stability robustness. The conditions for achieving a stable LQ system are as follows [16]:

- R > 0, Q ≥ 0
- (A,B) stabilizable
- (A,C) observable, where Q = C^TC

LQR controller is designed to drive the states to zero; however, in this study the controllers are needed to track a reference input. This requirement comes from the landing system architecture. In landing system architecture, a higher-level system block generates the required reference altitude, total airspeed and roll angles for the aircraft to keep track. However, the aircraft needs the angles for the control surfaces to change attitude. This conversion is done by the controller block, and the reference commands should be tracked accurately in order to follow the desired landing trajectory. For this purpose, the cost functional is modified such that it contains the tracking error, not the states. The cost functional and the optimal control are obtained in [16] as follows:

$$J = \frac{1}{2}(y(t_f) - r(t_f))^TP(y(t_f) - r(t_f)) + \frac{1}{2}\int_0^{t_f}[(y(t) - r(t))^TQ(y(t) - r(t)) + u^TRu]dt \tag{3.8}$$

where r(t) is the reference command and the system given in (3.6) with the output equation:

$$y(t) = Cx(t) \tag{3.9}$$

where y(t) is the output of the system and C is the output matrix (Q ≥ 0, R > 0 and P ≥ 0). Q matrix is formed as $C^T Q_y C$, by selecting a Q_y first, and then normalizing it in order to add the effect of the output.

The optimal control is given by:

$$u(t) = -K(t)x(t) + R^{-1}B^T v(t) \tag{3.10}$$

where $K(t) = R^{-1}B^T S(t)$ (3.11)

and S(t) is the solution to Riccati equation:

$$-\dot{S} = A^T S + SA - SBR^{-1}B^T S + C^T Q_y C \tag{3.12}$$
$$S(t_f) = C^T PC$$

with

$$-\dot{v} = (A - BK)^T v + C^T Q_y r \tag{3.13}$$
$$v(t_f) = C^T S(t_f) r(t_f)$$

In a particular case that $t_f \to \infty$, and the reference input (r) is constant, then we have S constant and given by the solution of the Riccati equation:

$$A^T S + SA - SBR^{-1}B^T S + C^T Q_y C = 0 \tag{3.14}$$

and v is also constant and given as:

$$v = -[(A - BK)^T]^{-1} C^T Q_y r \tag{3.15}$$

Using (3.15) in (3.10), the optimal control can now be expressed as:

$$u(t) = -Kx(t) + Fr \tag{3.16}$$

where

$$F = -R^{-1}B^T[(A - BK)^T]^{-1}C^T Q_y \tag{3.17}$$

In real systems, the states are measured using sensors, which introduce delay and noise to the system. In addition, sometimes some states cannot be measured if the necessary sensors are not installed to the platform or the installed sensors do not provide the necessary parameters. Those missing parameters are obtained via calculation (observer) using other states, which means additional delay. Since, this work is focused on a simulation environment instead of a real environment, all the necessary states are assumed to be measured without noise and delay. The system with real sensor behavior is left as a future work. Hence, the block diagram of LQT control is given in Figure 35.

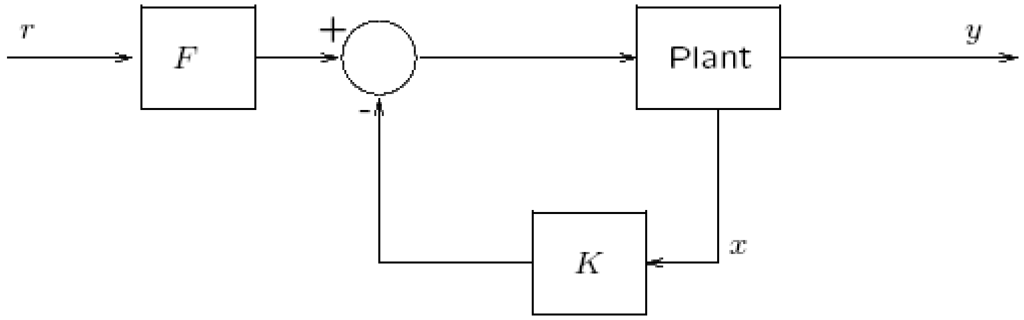

Figure 35 - LQT Control Architecture

3.3.1 LQT Design Details

LQT controller is designed for controlling altitude, total airspeed and roll attitude of the Pioneer. Different linear dynamic models are generated, consisting of the corresponding states and outputs for each control problem. The system actually is a multi-output multi-input (MIMO) system; however, it is divided into separate single-input single output (SISO) systems to design the controllers. Then all the controllers are combined in a single model consisting of all the 12 states and outputs.

After obtaining the linear models, the next thing to do is to choose the quadratic weight selection. At first two weighting matrices Q_y and R are selected. Closed loop system will exhibit different behaviors for different values of these matrices. Selecting a large Q matrix means that, to keep J small, states $x(t)$ must be smaller. On the other hand selecting a large R means that control input $u(t)$ must be smaller to keep J small. This means that larger values of Q generally result in the poles of the closed-loop system matrix being further left in the s-plane so that the state decays faster to zero. On the other hand, larger R means that less control effort is used, so that the poles generally move to the right in the s-plane, resulting in larger values of the state $x(t)$ [12]. The tracking error is defined as the state in the cost function, which is needed to be zero at all times. As stated in the Introduction chapter, the rate at which the speed of the tracking error decreases is one of the most important parameters in landing. In the weighting matrices selection process, this requirement is taken into account with higher priority. The values of the weighting matrices are chosen by an iterative trial and error methodology while examining the transient response of the system with respect to the requirement.

Then the controller optimal gain matrix is found by solving (3.8). Optimal gain can also be found using the Matlab's 'lqr' function.

$$[K\ S] = lqr(A, B, Q, R) \tag{3.18}$$

where K is the optimal gain matrix given in (3.11) and S is the solution of the Riccati equation. According to the Matlab Documentation "lqr" function has limitations such that the problem data must satisfy stabilizability and observability conditions as well as the Q and R limits. The optimal gain values are obtained using the "lqr" function, which also means that the conditions for achieving a stable LQ system is satisfied for all SISO systems. Therefore, no additional stability or observability analysis is presented in this study. Then the other gain matrix F is found using the formula given by (3.17).

For this controller and the other SISO controllers, the states are selected according to their effect on the controlled state. For roll attitude control, the states are selected as:

$$x = \begin{bmatrix} \beta \\ p \\ r \\ \emptyset \end{bmatrix} \qquad (3.19)$$

The output is the roll angle (∅). Block diagram for the roll controller is given below:

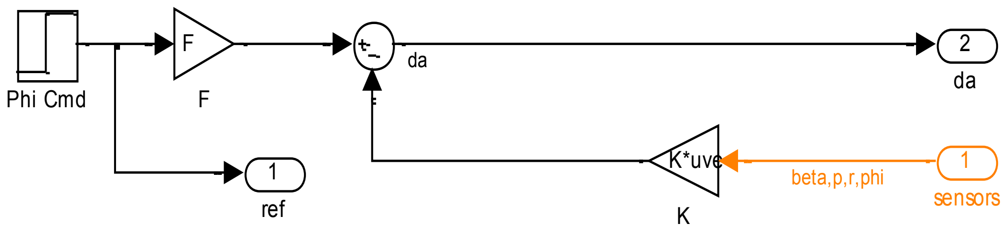

Figure 36 - LQT Roll Controller Block Diagram

Step input is applied from the aileron channel, and the roll attitude of the linear aircraft model is given below:

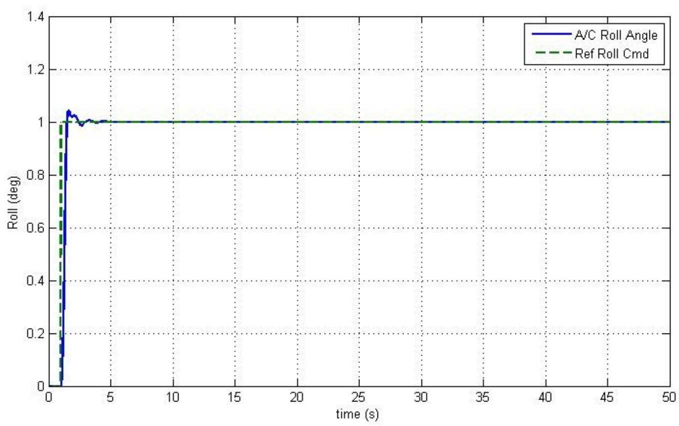

Figure 37 - Roll Angle wrt Step Input from Aileron Channel (LQT)

The overshoot and settling time performances are within acceptable limits for landing maneuver. Overshoot is less than 10%, which is better than the requirements.

Second controller is the altitude controller. The states are selected accordingly as:

$$x = \begin{bmatrix} \alpha \\ q \\ \theta \\ h \end{bmatrix} \tag{3.20}$$

The output is the altitude of the aircraft and the elevator control input to the aircraft is generated from the controller. Altitude controller block diagram is the same as the roll controller except the selected states.

A step input is given to the input of the controller and the altitude of the aircraft is observed. In order to obtain a fast and accurate tracking performance, different Q and R values are examined, and the ones that meeting the requirements are selected. Below is the step response of the linear system:

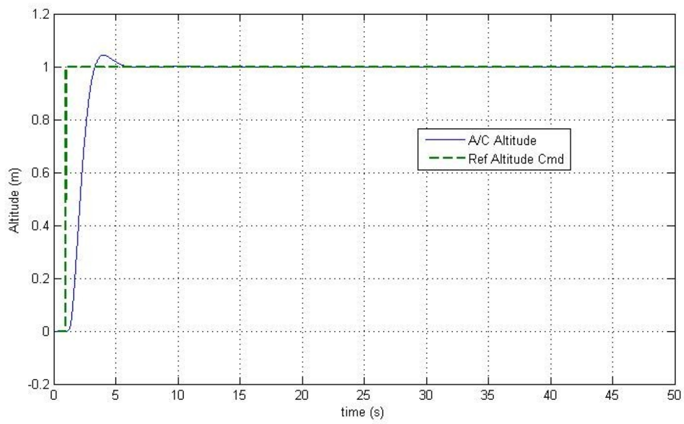

Figure 38 - Altitude wrt Step Input from Elevator Channel (LQT)

The third controller is the airspeed controller. The following longitudinal states are selected to control airspeed:

$$x = \begin{bmatrix} V \\ \alpha \\ q \\ \theta \end{bmatrix} \tag{3.21}$$

The input is applied as thrust to the aircraft and the output is the airspeed. Again a step input is applied, and the output is observed as below:

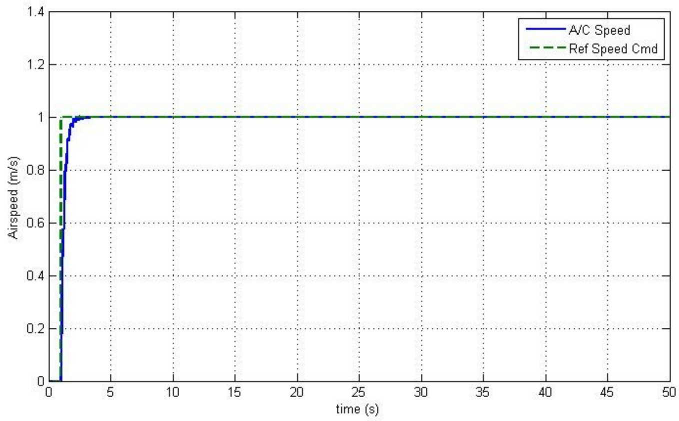

Figure 39 - Airspeed wrt Step Input from Thrust Channel (LQT)

LQT performance on linearized system is very good, with low overshoot and fast settling time; however, linear model performance is not a criterion for this study. For this reason, controller performance on nonlinear model is observed. Following are the results from LQT applied to nonlinear model:

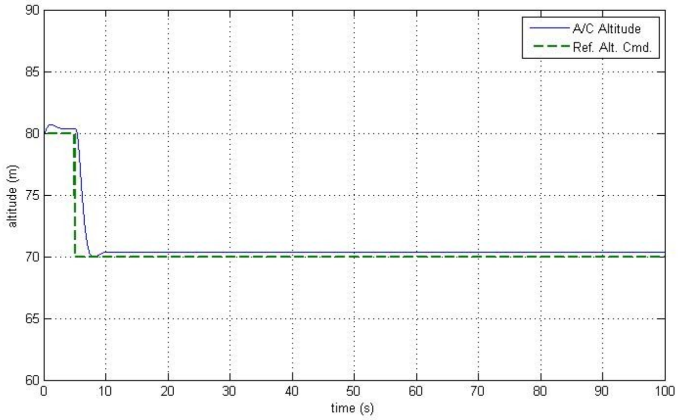

Figure 40 - LQT Controller on Nonlinear Model (Altitude Plot)

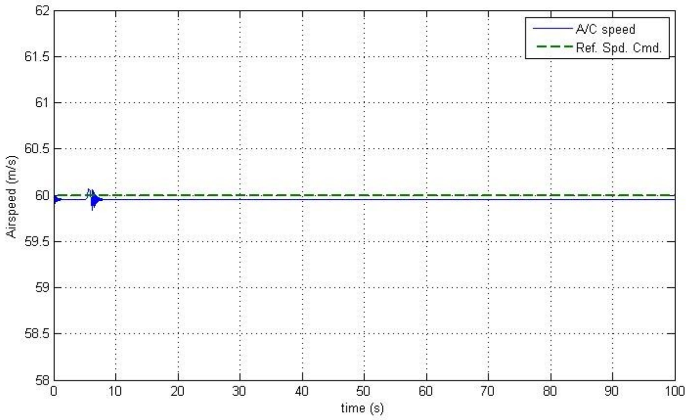

Figure 41 - LQT Controller on Nonlinear Model (Airspeed Plot)

A small steady state error is observed in altitude and airspeed controller's responses. It is possible to get rid of the error using integrator, but this will bring additional problems such as error propagation. The steady state error is not growing and the amplitude is small enough such that it is considered as acceptable for the scope of this thesis. This consideration is verified by testing the controller with inputs that is used in landing simulations. Figure 42 shows the controller's altitude performance without disturbance. It can be stated that the offset in the altitude affects only the touchdown time of the aircraft. The landing maneuvers are not affected.

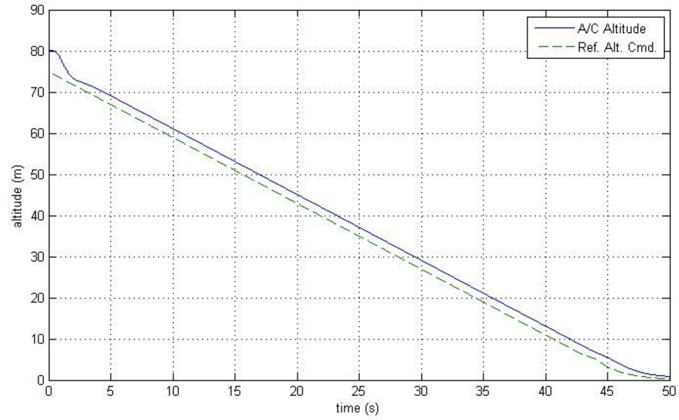

Figure 42 - LQT Landing Trajectory Performance

A cross-track reference step input of 10 meters is applied to the system and y-axis position is observed:

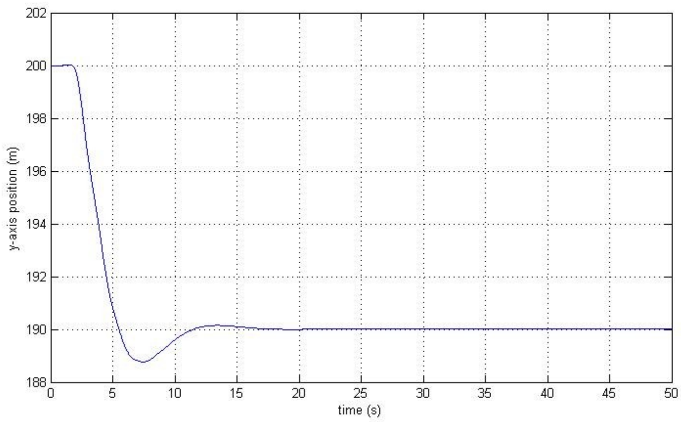

Figure 43 - LQT Controller on Nonlinear Model (y-Axis Position Plot)

3.4 Sliding Mode Controller (SMC) – Regular Form Approach

3.4.1 Problem Statement

In control theory, sliding mode control, or SMC, is a form of variable structure control (VSC). It is a nonlinear control method that alters the dynamics of a nonlinear system by application of a high frequency switching control. This VSC law provides an effective and robust means of controlling nonlinear plants. SMC Controller is invariant to matched uncertainty and disturbances [17]. Its variable structure allows SMC to adopt to parameter disturbances “instantaneously.”

In variable structure systems, the control is allowed to change its structure, that is, to switch at any instant from one to another member of a set of possible continuous functions of the state. The variable structure design problem is then to select the parameters of each of the structures and to define the switching logic [18].

The multiple control structures are designed so that trajectories always move toward a switching condition, and so the ultimate trajectory will not exist entirely within one control structure. Instead, the ultimate trajectory will slide along the boundaries of the control structures. The motion of the system as it slides along these boundaries is called a sliding mode and the geometrical locus consisting of the boundaries is called the sliding (hyper) surface. Figure 44 shows the phases of SMC.

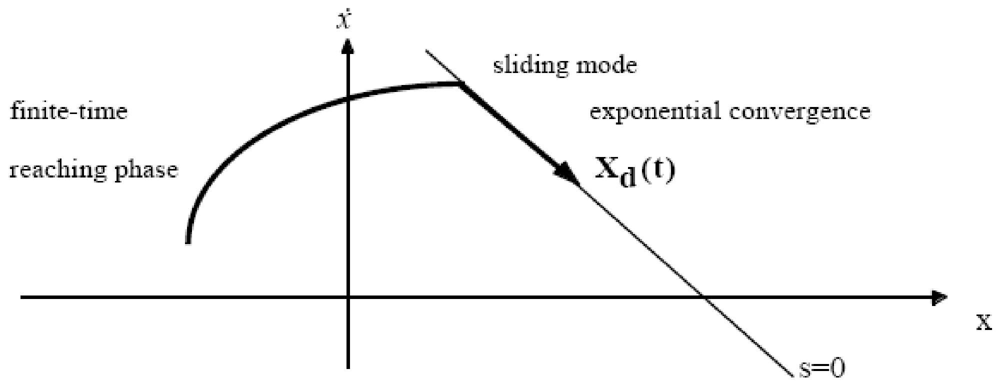

Figure 44 Phases of SMC

The plant dynamics restricted to this surface represent the controlled system's behavior. By proper design of the sliding surface, VSC attains the conventional goals of control such as stabilization, tracking, regulation, disturbance rejection etc., [19].

Consider the uncertain with m inputs and n states given by:

$$\dot{x}(t) = f(x,t) + B(x,t)u(x,t) \tag{3.22}$$

where $x \in R^n$ and $u \in R^m$. The objective is to define:

- m switching functions, represented in vector form as $s(x) = [s_1(x), \ldots, s_m(x)]^T$ with the desired state trajectories.
- a variable structure control $u(x,t)$ such that any state outside the switching surface is driven to the surface in finite time and remains on this surface for all subsequent time.

3.4.2 Sliding Surface Design

The purpose of the switching control law is to drive the state trajectory of the system onto a pre-specified (user-chosen) surface in the state space and to maintain the state trajectory on this surface for the subsequent time interval. This surface is called a switching surface. When the state trajectory is “above” the surface, the feedback controller uses one gain and a different gain if the state trajectory is “below” the surface. This surface defines the rule for a proper switching. This surface is also called a sliding surface (sliding manifold).

Although general nonlinear sliding surfaces are possible, linear ones are more prevalent in design [31], [32]. Moreover, for a large class of systems, design of linear sliding surfaces proves amenable to classical linear controller techniques. Thus for clarity, convenience, and simplicity of exposition, here focus will be on linear sliding surfaces of the form

$$s(x) = S\,x(t) \tag{3.23}$$

where S is a $m \times n$ matrix. Since the control action in the ideal sliding mode is discontinuous, the resulting differential equation is cannot be analyzed with traditional methods. A formal technique named equivalent control method will be used for finding equations of ideal sliding modes. The equivalent control is the continuous control action needed to maintain the ideal sliding motion. In this technique, the time derivative of the vector $s(x)$ along the system trajectory (3.2) is set equal to zero and resulting algebraic system is solved for control vector. This equivalent control (if it exists) is substituted into the original system. The resulting equations are the equations of ideal sliding mode.

An ideal sliding mode exists only when the state trajectory $x(t)$ of the controlled plant satisfies $s(x(t)) = 0$ at every $\mathrm{t} \geq \mathrm{t}_0$ for some t_0. This requires infinitely fast switching. In actual systems, all facilities responsible for the switching control function have imperfections such as delay, hysteresis, etc., which force switching to occur at a finite frequency. The representative point then oscillates within a neighborhood of the switching surface. This oscillation is called chattering. If the frequency of the switching is very high compared to the dynamic response of the system, the imperfections and the finite switching frequencies are often but not always negligible [19].

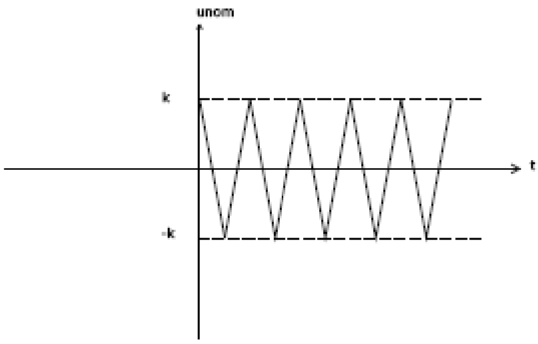

Figure 45 - Chattering Phenomenon

Consider the nominal linear model of an uncertain system, given by

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{3.24}$$

where state vector $x(t) \in \Re^n$, control input $u(t) \in \Re^m$, $rank(B) = m$ and (A, B) is a controllable pair.

The system (3.24) can be transformed into regular form via a change of coordinates defined by an orthogonal matrix T_r such that [20]

$$z(t) = T_r x(t) \tag{3.25}$$

where T_r is found by a QR decomposition of the input distribution matrix, that is

$$T_r B = \begin{bmatrix} 0 \\ B_2 \end{bmatrix} \tag{3.26}$$

Then, defining

$$T_r A T_r^T = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \tag{3.27}$$

and

$$S T_r^{T} = \begin{bmatrix} S_1 & S_2 \end{bmatrix} \tag{3.28}$$

The system can be expressed in the well-known regular form as

$$\begin{aligned} \dot{z}_1(t) &= A_{11}z_1(t) + A_{12}z_2(t) \\ \dot{z}_2(t) &= A_{21}z_1(t) + A_{22}z_2(t) + B_2u(t) \end{aligned} \tag{3.29}$$

and

$$s(t) = S_1 z_1(t) + S_2 z_2(t) \tag{3.30}$$

During the sliding motion, the switching function must be identically zero, so

$$S_1 z_1(t) + S_2 z_2(t) = 0 \tag{3.31}$$

It can be shown that S_2 is non-singular, so z_2 can be solved for on the sliding mode

$$z_2(t) = -S_2^{-1} S_1\, z_1(t) = -M\, z_1(t) \tag{3.32}$$

where $M \in \Re^{m\times(n-m)}$. The sliding mode is then governed by

$$\begin{aligned} \dot{z}_1(t) &= A_{11}z_1(t) + A_{12}z_2(t) \\ z_2(t) &= -Mz_1(t) \end{aligned} \tag{3.33}$$

This is an $(n-m)^{th}$ order system in which z_2 acts in the role of a linear full-state feedback control signal. Closing this loop gives the free motion of the system

$$\dot{z}_1(t) = (A_{11} + A_{12}M)z_1(t) \tag{3.34}$$

By using quadratic minimization approach and performance index weightings on the states, the hyperplane matrix S will be determined from M by letting $S_2 = I_m$, giving

$$S T_r^T = [M \quad I_m] \tag{3.35}$$

This approach minimizes the preceding calculations from M to S and so reduces the possibility of numerical errors [12].

3.4.3 The Control Law

After the sliding surface is determined, the next stage of controller design is to determine the actual control vector **u** so that the state of the system is forced toward the sliding surface. In other words, **u** must be determined so that s is driven to zero and kept so despite the disturbances. This purpose can be achieved by using the second stability theorem of Lyapunov as described below.

Let $V = \frac{1}{2} s^T s$ be defined as the Lyapunov function. Then, the following condition must be satisfied to force s to be zero:

$$\dot{V} = s^T \dot{s} < 0 \tag{3.36}$$

By using the condition (3.36), SMC can be obtained by adding two different controllers.

An associated switching function such as given in (3.23) is defined and the equivalent control for (3.24) is obtained as given in [18]:

$$u_{eq}(t) = -(SB)^{-1}\big(SAx(t)\big) \tag{3.37}$$

If the plant model is correct (i.e., there are no unmodeled dynamics or disturbances), u_{eq} will maintain the sliding mode assuming the switching manifold has been reached.

However, the discontinuous control function is required to maintain the sliding mode in the presence of model uncertainties and disturbances. It is common to see the equivalent control included in the control law, as follows:

$$u(x,t) = u_{eq} + \rho sgn(s) \tag{3.38}$$

In this study, instead of $\mathrm{sgn}(s)$, an approximate function tanh(s) is used.

3.4.4 SMC Design Details

SMC is designed using the regular form approach, and following the design steps given below [20]:

1. Design the sliding surface,

The sliding surface is selected as the tracking error of the system:

$$s = S_i(x - x_d) \tag{3.39}$$

2. Change the coordinates of the system to regular form,
3. Perform QR decomposition to get Tr,
4. Obtain Areg and Breg using Tr,
5. Obtain matrix sub-blocks in the regular form equations,
6. Obtain the switching function matrix coefficients,
7. Transform weighting matrix to regular form coordinates,

$$Q_T = T_r\, Q T_r^{\,T} \tag{3.40}$$

8. Partition weighting matrix with regular form description,

$$Q_T = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \quad (3.41)$$

9. Form reduced order system description and associated weighting matrix,

10. Solve the LQR problem,

11. Obtain the switching function matrix in terms of the original coordinates,

$$M = Q_{22}^{-1} \left(A_{12}^{T} P_1 + Q_{21} \right)$$
$$S = \begin{bmatrix} M & I_m \end{bmatrix} T_r \quad (3.42)$$

Controller is initially designed over linearized (Trimmed at h=60m, V=60m/s) Pioneer UAV model. The linear model is simplified to a SISO model. Control surface dynamics are also added to the linear model in order to keep the behavior between linear and nonlinear model similar to each other. The design environment is Matlab/Simulink. Four separate controllers are developed using relevant state-space configurations:

- Roll,
- Total Airspeed,
- Altitude.

The required input channel:

- For roll control is the aileron input, and the output is selected as phi angle.
- For airspeed control is the thrust input, and the output is selected as aircraft airspeed.
- For altitude control is the elevator input, and the output is selected as aircraft altitude.

The following are the selected states of the system for roll model:

$$x = \begin{bmatrix} \beta \\ p \\ r \\ \emptyset \end{bmatrix} \tag{3.43}$$

Sliding surface is selected as:

$$S = S_1\beta + S_2p+S_3r+S_4(\emptyset - \emptyset_d) \tag{3.44}$$

The following are the selected states of the system for airspeed control model:

$$x = \begin{bmatrix} V \\ \alpha \\ q \\ \theta \end{bmatrix} \tag{3.45}$$

Sliding surface is selected as:

$$S = S_1(V-V_d) + S_2\,\alpha + S_3q+S_4\,\theta \tag{3.46}$$

The following are the selected states of the system for altitude model:

$$x = \begin{bmatrix} \alpha \\ q \\ \theta \\ h \end{bmatrix} \tag{3.47}$$

Sliding surface is selected as:

$$S = S_1\,\alpha + S_2q+S_3\,\theta + S_4(h-h_d) \tag{3.48}$$

3.4.5 Simulations

3.4.5.1 Simulations with Linear Model

For simulation purposes, step commands are given to the autopilots and the responses are observed. Following three figures are the simulation results for the linear model, and the performance of the controllers can be misguiding for the final landing scenario with the nonlinear model. The overshoot, settling time and the steady state error values are expected to be close to ideal, for this reason they are used just to give an idea about the controller performance.

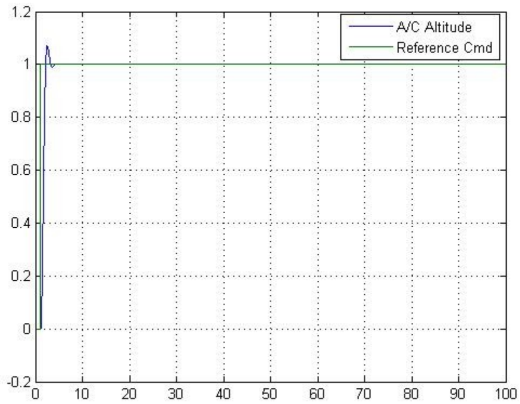

Figure 46 - Altitude Channel Step Response (Linear Model)

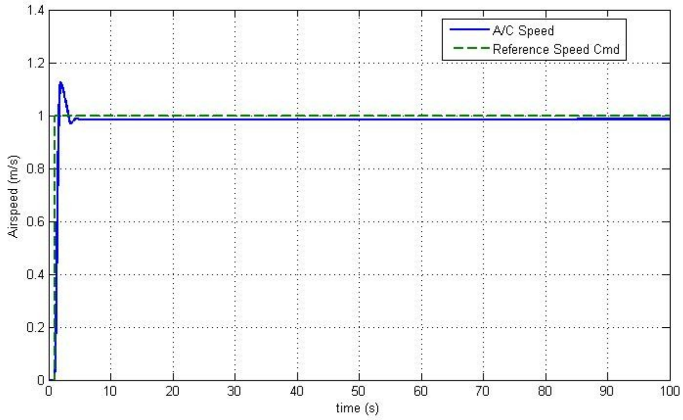

Figure 47 - Airspeed Channel Step Response (Linear Model)

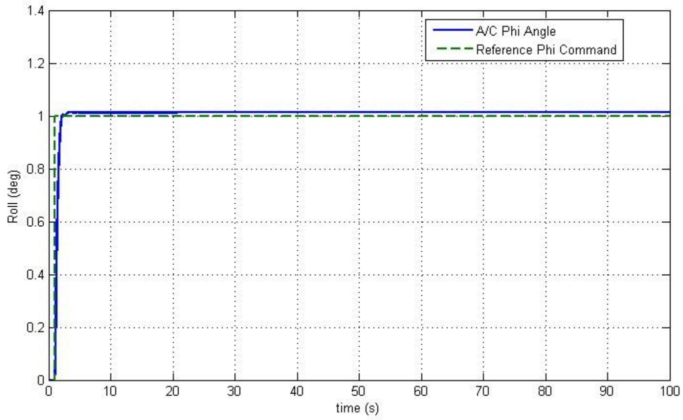

Figure 48 - Phi Channel Step Response (Linear Model)

3.4.5.2 Simulations with Nonlinear Model

After the design of the controller for the linear model, it is modified and embedded to the original nonlinear model. The switching controller coefficient (ρ) and weighting matrices (Q, R) are need to be revised. Required values are found by trial and error methodology.

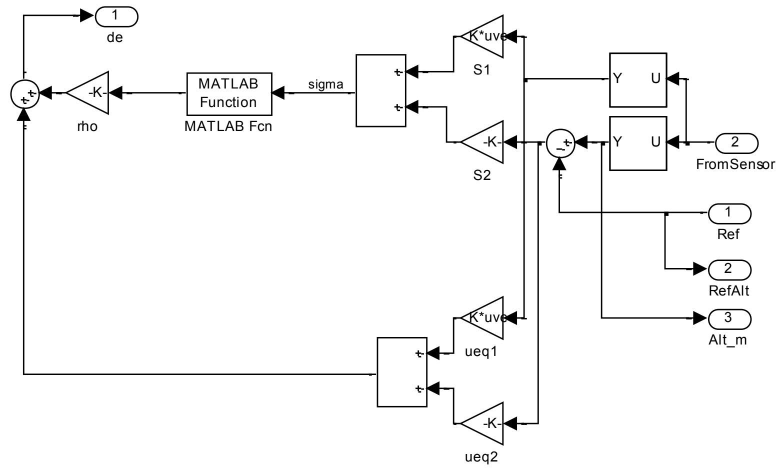

Figure 49 - SMC Altitude Controller Block Diagram

Table 8 - SMC Altitude Controller Parameters

S	5,627	1,018	11,783	0,316
U_{eq}	-1,291	-0,0015	0,288	-1,346e-06
ρ	-0.5			

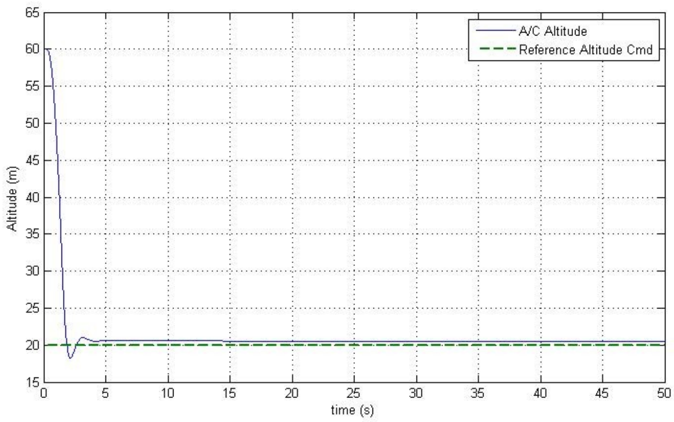

Figure 50 - Altitude Response to Constant Input (Nonlinear Model)

A constant input is applied as an altitude reference for the controller, which is also different from the current altitude of the aircraft. Speed and accuracy of the controller is acceptable for the landing maneuver, because the altitude maneuver is in landing is slower than this scenario with no disturbance.

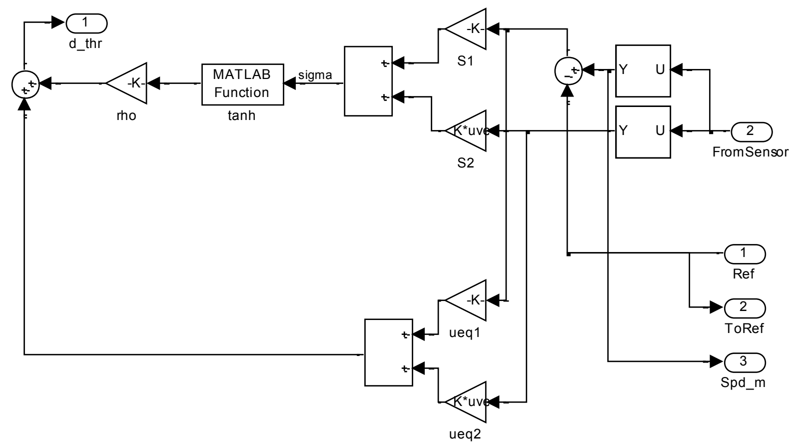

Figure 51 - SMC Airspeed Controller Block Diagram

Table 9 - SMC Airspeed Controller Parameters

S	-1	0,052	0,0038	0,00026
U_{eq}	10,275	702,56	5,035	1869,102
ρ	800			

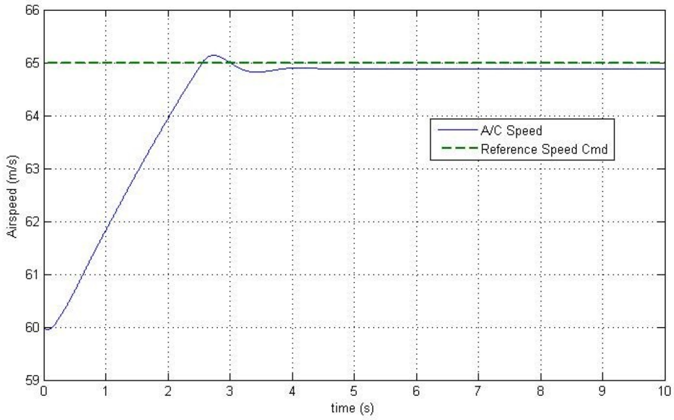

Figure 52 - Airspeed Response to Constant Input (Nonlinear Model)

A small steady state error is observed, as in LQT in airspeed values. This error is also neglected. In real systems, sensor noise or measurement errors/inaccuracies exist, and the amount of the error is higher. This kind of error sources are not used in this simulation, therefore this small error can be tolerated by the aircraft.

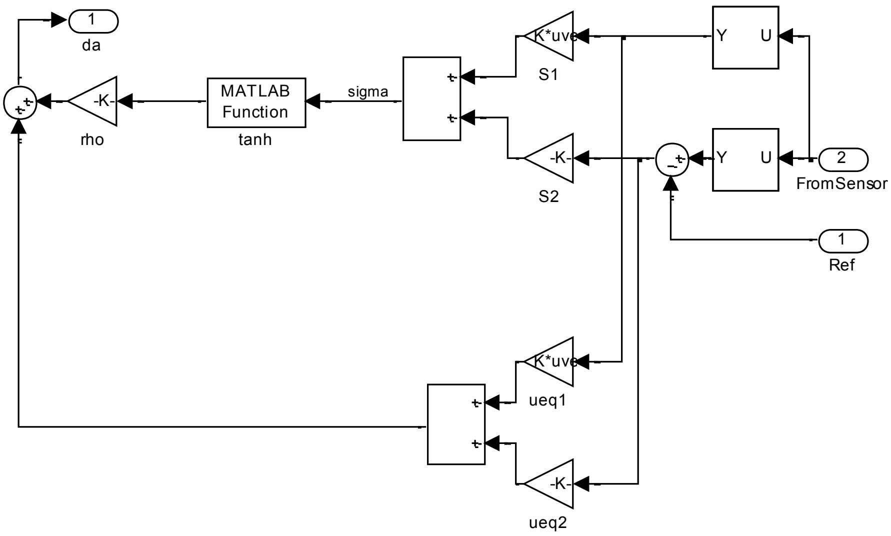

Figure 53 - SMC Phi Controller Block Diagram

Table 10 - SMC Phi Controller Parameters

S	1,022	0,970	-0,332	3,203
U_{eq}	-0,276	-0,082	0,069	0,0015
ρ	0.5			

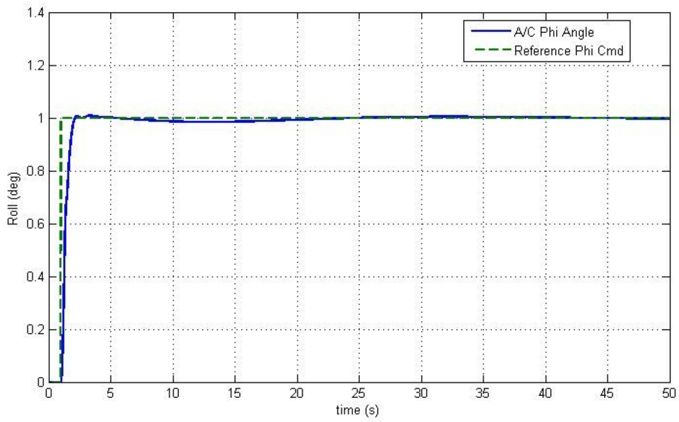

Figure 54 - Phi Channel Step Response (Nonlinear Model)

Almost no overshoot and steady state error is observed on roll response of the controller. This response is critical in landing especially under crosswind.

3.5 Sliding Mode Controller (SMC) – Nonlinear Approach

The nonlinear system used in this thesis is in the following form

$$\dot{x} = f(t, x, u) \tag{3.49}$$

For the nonlinear approach, different than the regular form approach it is assumed that the right-hand side of the equation is linear with respect to control u. A nonlinear affine dynamical system modeled as (3.50) is considered [21].

$$\dot{x} = f(t, x) + g(t, x)u \tag{3.50}$$

where $x(t) \in R^n$, $f(x, t) \in R^n$, $g(x, t) \in R^{nxm}$ and $u \in R^m$. Here f and g are continuous functions of their arguments. First thing to do here is to obtain the functions f and g. For this purpose, Pioneer dynamics are recalculated using Maple Software. Maple is a general-purpose computer algebra system. There is extensive support for numeric computations, to arbitrary precision, as well as symbolic computation and visualization [29]. All the calculations are done in symbolic form for this thesis. Engine force and moments, wind disturbance, aileron deflection, rudder deflection, elevator deflection are defined as inputs to the system. States are also kept the same as the previous designs. First aerodynamic and propulsive forces and moments are calculated then gravitational forces are added to obtain the total forces and moments. Then the state derivatives are obtained using the forces, Euler angles, inputs and states. After obtaining the dynamics in the form of (3.50), $f(t, x)$ is found by defining $u = 0$.

$$f(t, x) = f(t, x, u = 0) \tag{3.51}$$

Then $g(t, x)$ is found by subtracting the jacobian of $f(t, x, u)$ wrt u from $f(t, x, u)$ as following:

$$g(t, x) = f(t, x, u) - \frac{\partial f(t,x,u)}{\partial u} \tag{3.52}$$

The functions $f(t, x)$ and $g(t, x)$ are then exported as Matlab m-file format, and embedded into a ‘Embedded Matlab Function’ block of Simulink. A linear sliding surface is selected same as (3.23). Then, by following the Utkin’s method so-called Equivalent Control Method the sliding mode equation with a unique solution may be derived for the nonsingular matrix [22].

$$S(x)g(x, t), S(x) = \left\{\frac{\partial s}{\partial x}\right\}, \ \det(Sg) \neq 0 \tag{3.53}$$

An equivalent control is found for the system as the solution to the equation $\dot{s} = 0$ on the system trajectories (S and $(SB)^{-1}$ are assumed to exist):

$$\dot{s} = S\dot{x} = Sf + Sgu_{eq} = 0 \quad , \qquad u_{eq} = -(Sg)^{-1}Sf \tag{3.54}$$

Then the control is substituted into (3.50) as following:

$$\dot{x} = f - g(Sg)^{-1}Sf \tag{3.55}$$

Equation (3.54) is the sliding mode equation with initial conditions $s(x(0), 0) = 0$. Equivalent control is included in the control law as in (3.56).

$$u(x, t) = u_{eq} + \rho \tanh{(s)} \tag{3.56}$$

This has the advantage of reducing the activity of the discontinuous portion of the control. Block diagram of the SMC is generated in Simulink as given below:

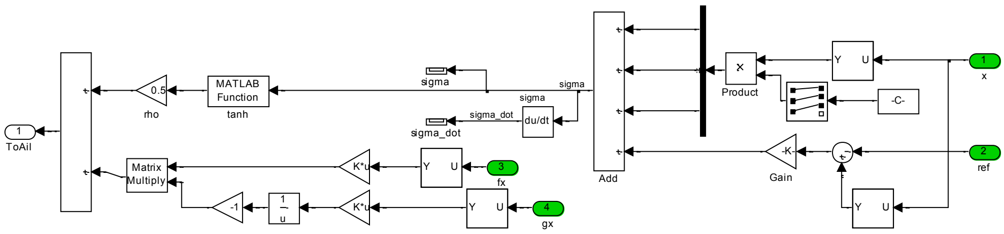

Figure 55 - Nonlinear SMC Approach Controller Block Diagram

As in the linear case, controllers for roll attitude, altitude and airspeed of the aircraft are designed, and an additional theta controller is designed. Theta controller is not used in these scenarios; however, it is planned to be used in future studies that touchdown will be analyzed. Step input is applied to roll controller. Roll reference is provided from the guidance block to the controller, and it is expected to be changing very quick. Step response is very important in that sense.

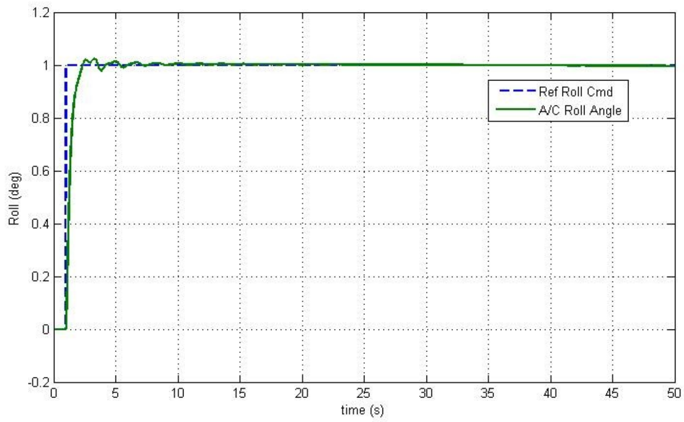

Figure 56 - NL SMC Approach Roll Step Response

Overshoot is less than 10% and settling time is lower than 5 seconds, which means that the controller satisfies requirements.
Similarly, reference airspeed and altitude inputs are applied and controller responses are plotted below:

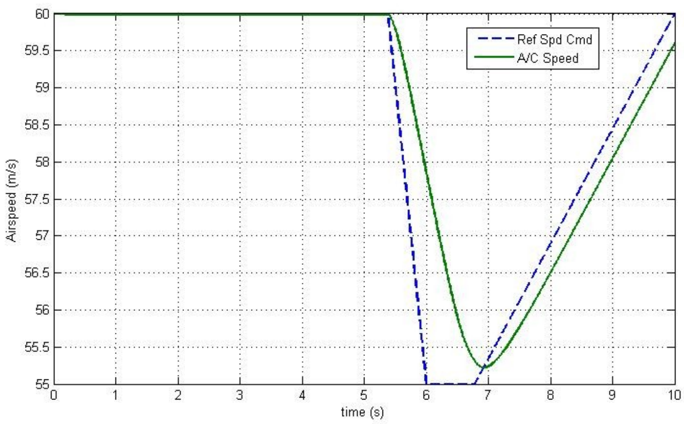

Figure 57 - NL SMC Approach Airspeed Response

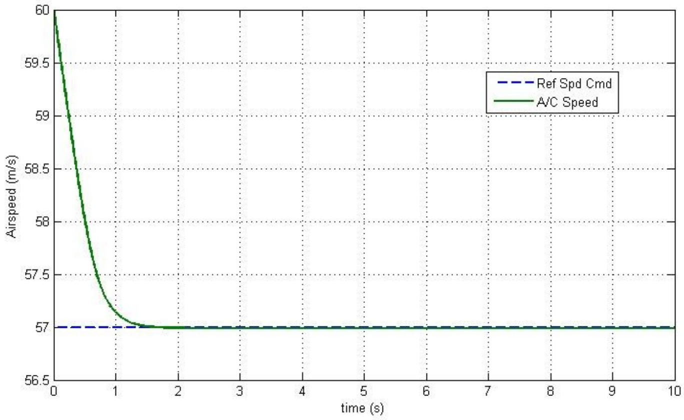

Figure 58 - NL SMC Approach Constant Airspeed Command Response

First test case is similar to the cruise phase commands for the controller, which is done via waypoint navigation. Second test case is similar to the landing phase, where constant airspeed input (possibly different from the current airspeed) is applied throughout the trajectory. Following is the altitude response of the controller, where a tracking error is observed as the reference changes faster than the controller can track. This also shows that the controller bandwidth is not high enough to tolerate this rate of change in the reference. This problem can be omitted by defining waypoints according to the controller bandwidth. In landing, this is not expected to produce an effect, because the rate of change of reference command will be constant during most of the maneuver.

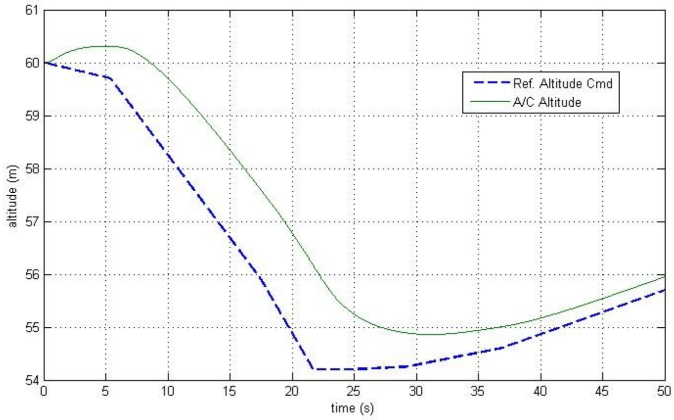

Figure 59 - NL SMC Approach Altitude Response

After designing and embedding the controller, an external disturbance of wind is added to the model. Wind speed components are input to the system in earth-axes, then converted to the body-axes and added to the aircraft airspeed's body-axes components.

In order the aircraft to keep track on cross-wind, a Cross Track Guidance is included to the model. It is a simple PID controller based on the aircraft's distance to the desired y position (cross-track error). A roll command is generated using the following formula:

$$\emptyset_{cmd} = K_p(xt_d - xt_{ac}) + K_i \int (xt_d - xt_{ac}) + K_d \frac{d}{dt}(xt_d - xt_{ac}) \quad (3.57)$$

where, xt_d is the desired (reference) cross-track, xt_{ac} is the current aircraft cross-track and $\emptyset_{cmd}$ is the roll angle commanded to the roll autopilot.

The controller is then tested under different wind conditions. First, under no wind conditions a cross-track command of 5 meters is applied:

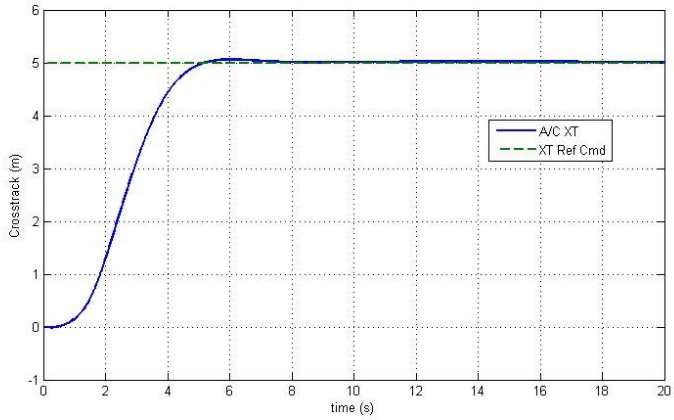

Figure 60 - XT Response (No Wind)

For the cross-wind cases, the desired cross-track is set as 0, which means that the aircraft should hold its initial heading. The aircraft's initial heading is set to 0 (North). Then different cross-wind conditions are generated as 1m/s, 5m/s and 10m/s, respectively.

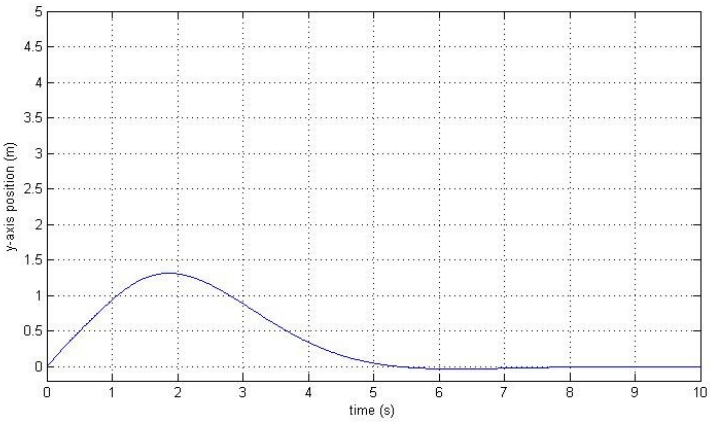

Figure 61 – y-Axis Position of the Aircraft (Wind = 1m/s)

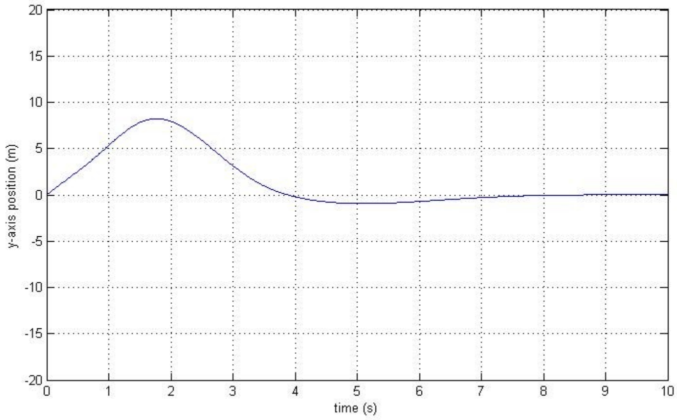

Figure 62 – y-Axis Position of the Aircraft (Wind = 5m/s)

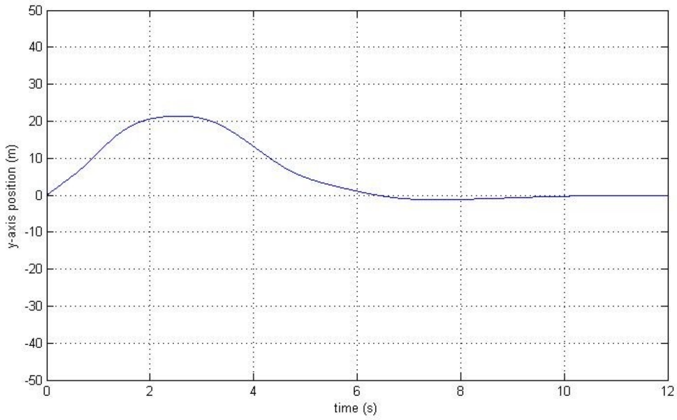

Figure 63 – y-Axis Position of the Aircraft (Wind = 10m/s)

The results show that; as the aircraft is first exposed to wind, it is drifted to the direction of the wind, then the controller generates an opposing roll command and beats the wind. Other than cross-wind, many other wind scenarios including different wind speeds and directions are run, and the detailed results are given in Chapter 5. The controller command versus the position of the aircraft for the 10m/s cross-wind condition can be seen in Figure 64:

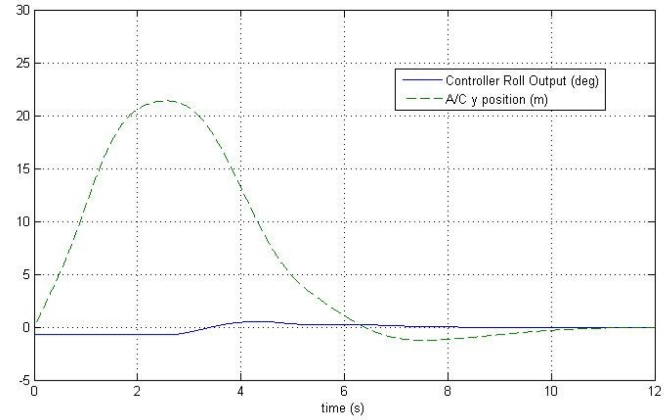

Figure 64 - Controller Roll Output vs. Aircraft y-Position

CHAPTER 4

GUIDANCE AND NAVIGATION APPROACH

Guidance is used to navigate the aircraft from one state to another. It is the “Pilot” of the aircraft. It takes input from the sensors (where am I) and uses the targeting information (where do I want to go) to send signals to the controller that will allow the aircraft to reach its destination.

Type of guidance system used depends on the design of the aircraft and the accuracy requirements. The guidance systems are divided into two categories: online and offline. In offline case, the autopilot commands are generated before the flight considering the possible aircraft path. In online case; guidance system uses a previously defined mission/flight plan, which includes destination points called waypoints. Waypoints are basically state vectors containing destination position data, velocity data, etc. In any case, the guidance output serves as the input to the control system. The guidance output may be in the form of an attitude command, an airspeed command, and a pitch or yaw command. In real systems motion related parameters of the aircraft can be obtained by the inertial navigation system (INS) or the global positioning system (GPS).

In this thesis, longitudinal and lateral autopilots designed are driven by airspeed, altitude and heading reference commands, respectively. For this reason, the guidance block that will be added to the system must output these commands. As stated above, the guidance block should know about the aircraft’s current state such as position and velocity in order to guide the aircraft to the desired state. Main objective of this thesis is to design an automatic landing system for the Pioneer aircraft, and for this reason, a landing trajectory or a landing corridor for the aircraft must be designed. Landing trajectory of an aircraft is a combination of many parameters, such as altitude, distance from runway centerline, airspeed, etc. It is possible to land an aircraft from different initial points; for this reason there can be more than one landing trajectory. This study aims to generate a landing corridor that the aircraft can be safely landed autonomously. It can be done by selecting a corridor that is suitable for the aircraft's capability to follow. First thing to do is to define the ideal landing path, and then to define the borders. The glide slope for the descent phase is chosen as -3°; however, the angle must be changed to a positive value which is suitable for touchdown, in other words, during this maneuver (the flare maneuver) the control system must control the height of the aircraft's c.g. and its rate of change such that the resulting trajectory corresponds as nearly as possible to the idealized exponential path shown in Figure 65. The equation which governs the idealized, exponential flare trajectory is defined in (4.1) [15].

$$h = h_0 e^{-t/\tau} \tag{4.1}$$

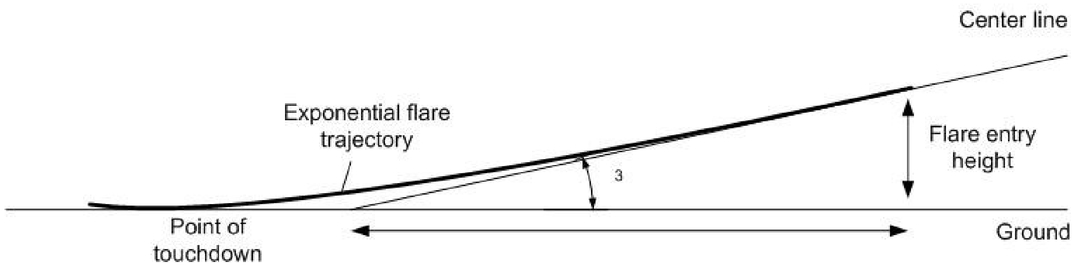

Figure 65 - Exponential Flare Trajectory

We know that

$$\dot{h} = U_o \sin(\gamma) \tag{4.2}$$

and using it together with (4.1), the flare height entry can be found as 5,2 m. The desired flare trajectory can now be plotted as:

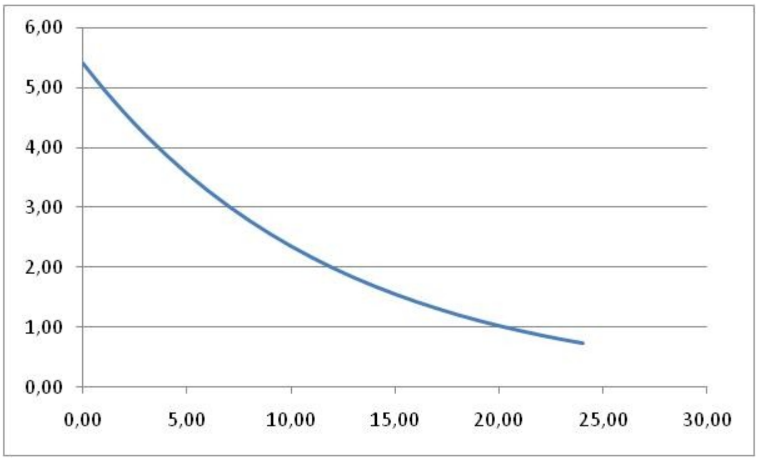

Figure 66 - The Desired Flare Trajectory

After defining the trajectory, the landing corridor is generated by first adding some strict tolerances to it.

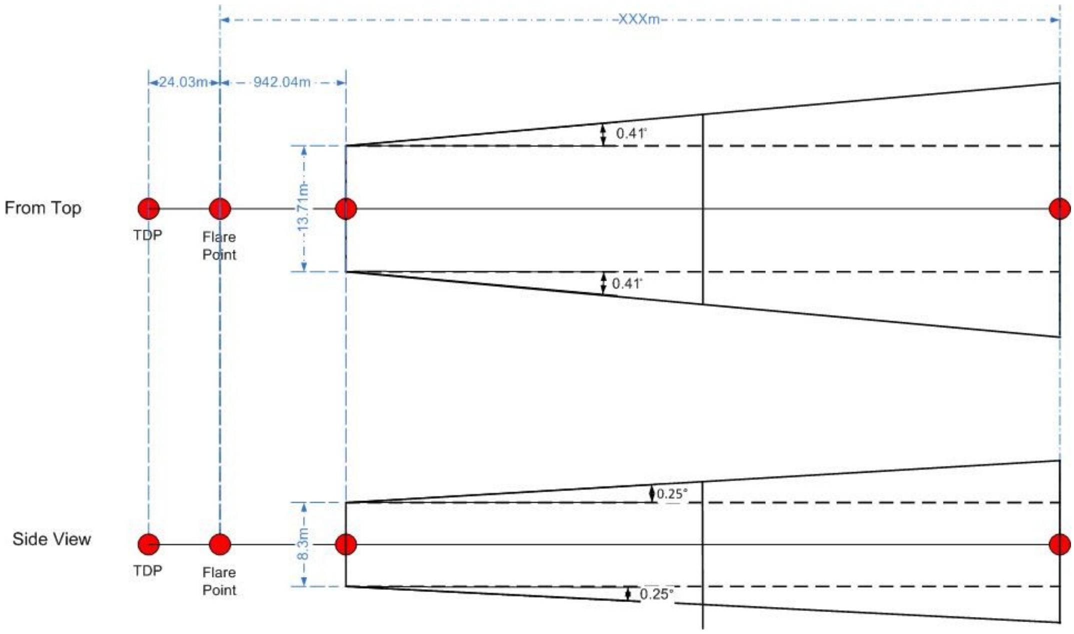

Figure 67 - Strict Trajectory Tolerances

Then, by increasing the error tolerances while staying in the safe side, the final borders of the corridor are defined as the following:

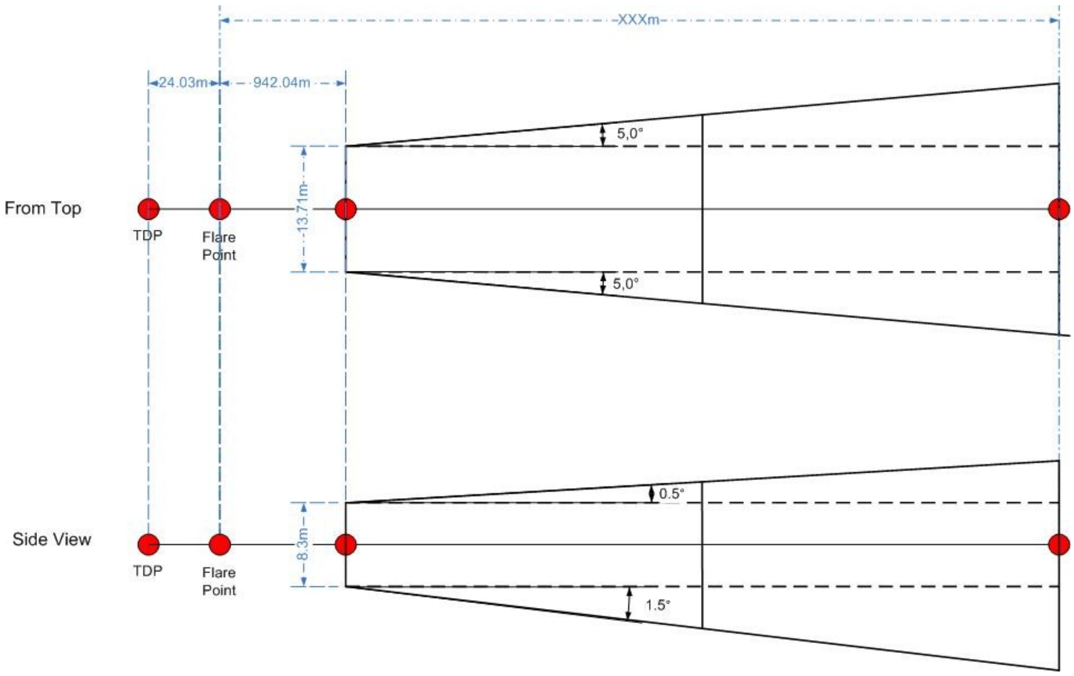

Figure 68 - Large Trajectory Tolerances

$$Corridor\ (X, Y, Z) = (X, 2Xtan(5°) + 13.71, Xtan(0.5°) + Xtan(1.5°) + 8.3) \qquad (4.2)$$

Combining the ideal trajectory with the tolerances above according to the equation (4.2) gives us the desired final landing corridor as given in Figure 69. The landing corridor definition is considered together with the controller design specifications. The corridor should be defined such that it allows safe landing margins to the aircraft even under wind disturbance, and it is the controller's duty to keep the aircraft in corridor. If the borders of the corridor was selected wider, the controller design would also be loose, and under wind disturbance this would yield the aircraft getting out of the corridor without intention. Considering this fact, the final landing corridor is designed narrow enough for the aircraft to be prevented to get out, and wide enough to allow the required maneuvers.

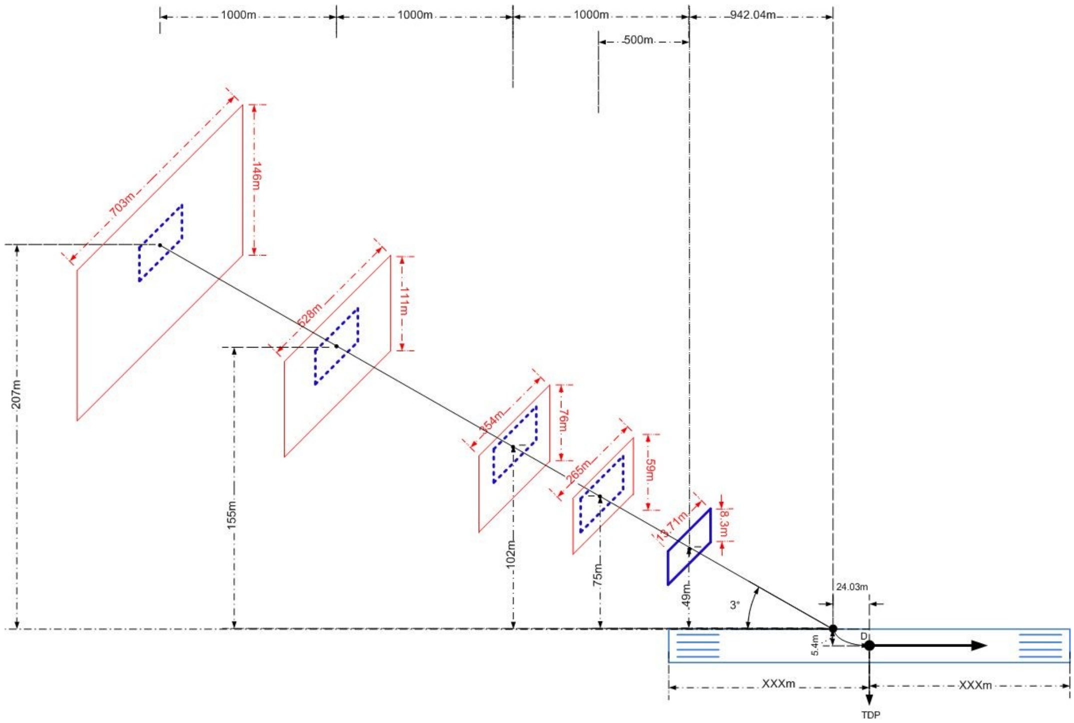

Figure 69 - The Landing Trajectory

Generalized guidance-navigation-control (GNC) loop with all other system blocks can be shown as in the block diagram representation in Figure 70.

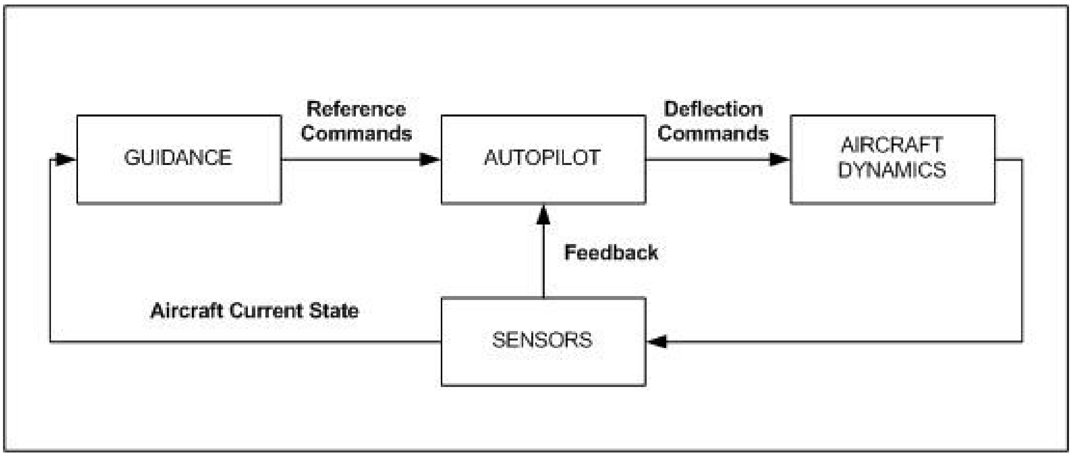

Figure 70 Guidance-Navigation-Control Loop

In this chapter, the guidance algorithms implemented for the landing maneuver in the thesis are introduced. An Optimal Guidance (OG) approach, Lateral Track Guidance and the Cross Track Guidance laws are developed.

4.1 An Optimal Guidance (OG) Approach

Optimal linear quadratic tracker approach can be used in guidance applications. The performance index that will be used for tracking design is stated as follows

$$J = \propto \sum_{i=0}^{t_f} \left(\bar{x}(i) - \bar{x}_d(i)\right)^T Q \left(\bar{x}(i) - \bar{x}_d(i)\right) + \frac{\beta}{2} \sum_{0}^{t_f} \bar{u}^T(i) R\, \bar{u}(i) \qquad (4.3)$$

where $Q \geq 0,\ R > 0$ are symmetric matrices.

Reference signal $\bar{x}_d$ will be taken as predefined states of the aircraft at the next time instant.

The states are chosen as altitude, airspeed and heading., whereas control channels are elevator, thrust and aileron respectively. The approach can be stated as:

1.

$$\left.\begin{matrix} \alpha \\ \beta/2 \\ \bar{u} \\ Q \\ R \\ x_d \end{matrix}\right\} \text{are known} \rightarrow \text{Then (Estimate) } x(0)\text{, then solve (4.3) and } J(0)$$

2.

Then use the Steepest Descent Algorithm to find $u(1)$ by minimizing $J(1)$. Then replace $u(0)$ with $u(1)$ and repeat the first step above. Main goal is to find the input vectors (u) for altitude, airspeed and heading that keeps the aircraft on the desired states. The desired airspeed and heading of the aircraft is defined constant as 30 m/s and 0° (same as runway heading), respectively. The only varying parameter is the desired altitude.

The minimization problem can be written as:

Minimize

$$J = \propto \sum_{i=0}^{t_f} \left(\bar{x}(i) - \bar{x}_d(i)\right)^T Q\left(\bar{x}(i) - \bar{x}_d(i)\right) + \frac{\beta}{2} \sum_{0}^{t_f} \bar{u}^T(i) R\, \bar{u}(i)$$

Subject to

$$\bar{u}_{altitude} \geq 0$$

$$60 \geq \bar{u}_{speed} \geq 26$$

$$1 \geq \bar{u}_{heading} \geq -1$$

The steepest descent method (also known as the gradient descent) is used for minimizing the cost function, which is one of the most widely known methods for minimizing a function of several variables [23].

4.1.1 The Steepest Descent Method (SDM)

The basis for the method is a simple observation that a continuous function should decrease, at least initially, if one takes a step along the direction of the negative gradient. The only difficulty is deciding how to choose the length of the step one should take [24].

Suppose, the minimization function is $f(x), x \in R^n$ and $f: R^n \rightarrow R$. The gradient of f is denoted by $g_k = g(x_k) = \nabla f(x_k)$. The following iterative algorithm defines the method of steepest descent:

$$x_{k+1} = x_k - \alpha_k g_k \qquad (4.4)$$

where the step length α_k, a nonnegative scalar, is chosen such that

$$\alpha_k = \arg min_{\alpha}\ f(x_k - \alpha_k g_k) \qquad (4.5)$$

The steepest descent method has two main computational advantages such that: ease with which a computer algorithm can be implemented and the low storage requirements. The main work is composed of a line search required to compute the step length, α_k and the computation of the gradient.

Algorithm 1 – Steepest Descent Method

Given an initial $-g_0$, and a convergence tolerance *tol*
for *k=0 to maxiter* ***do***
Set $\alpha_k = argmin\ \emptyset(\alpha) = f(x_k) - \alpha g_k$
$x_{k+1} = x_k - \alpha_k g_k$
Compute $g_{k+1} = \nabla f(x_{k+1})$
if $\|g_{k+1}\|_2 \leq tol$ ***then***
converged
end if
end for

In order to find the step size, line search algorithms are used. It is basically the process of determining the minimum point on a given line. In most problems; however, it can be safely assumed that the function being investigated, as well as being unimodal, possesses a certain degree of smoothness. Techniques exploiting this smoothness are usually based on curve fitting procedures where a smooth curve is passed through the previously measured points in order to determine an estimate of the minimum point. The parabolic fitting procedures have been applied in this thesis.

The simulation is done by using 'sim' function of Matlab. The optimization algorithm is run on an m-file while calling the nonlinear simulink model. The simulation block diagram is given below:

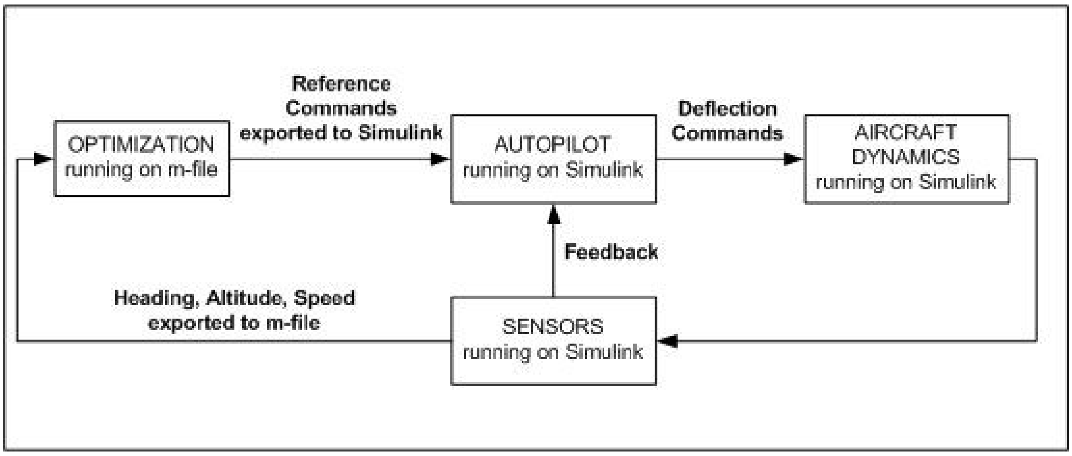

Figure 71 - Optimization Simulation Block Diagram

Optimization algorithm is run for z (altitude) and y (heading) axes, airspeed is added as a punishment to the algorithm, such that when it exceeds the desired boundaries the algorithm terminates or cost is multiplied by a huge number. Cost function is selected such that it minimizes the cross track error with the desired altitude and distance from runway centerline, and the tracking error.

Algorithm is run for the following values:

Table 11 - SDM Iteration Numbers and Initial Conditions

Initial Altitude	Initial Airspeed	Initial y position	SDM iteration	Parabolic Fit iteration
120	30 (Const)	0 (Const)	2	20
140	30 (Const)	0 (Const)	2	20
60 (Const)	30 (Const)	100	10	50
60 (Const)	30 (Const)	144	10	50

The results for the iterations and initial conditions given in Table 11 are given below:

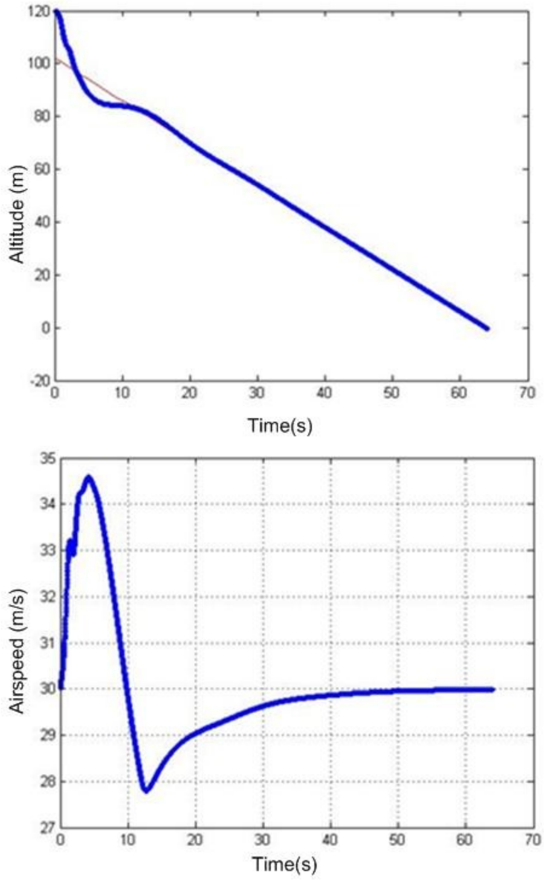

Figure 72 - Initial Altitude = 120 (Altitude (above), Airspeed (below), Thin line: Desired, Thick Line: A/C)

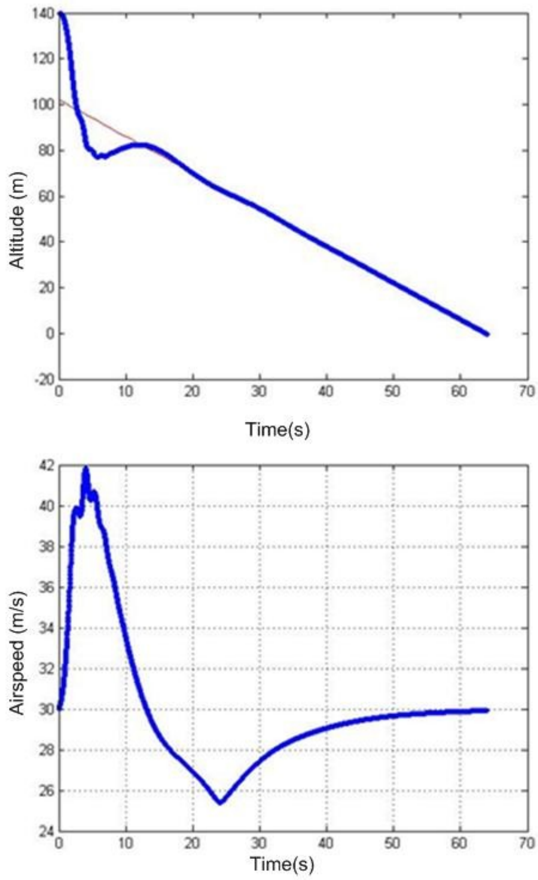

Figure 73 - Initial Altitude = 140 (Altitude (above), Airspeed (below), Thin Line: Desired, Thick Line: A/C)

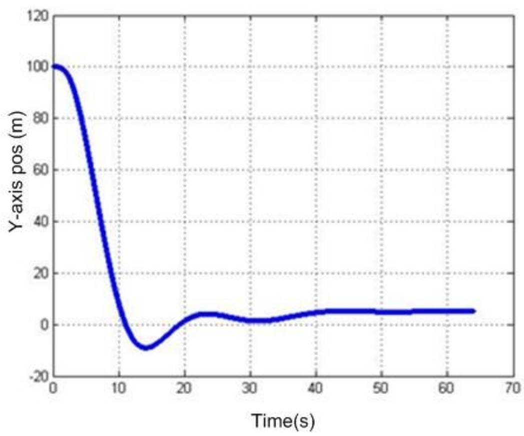

Figure 74 - Initial y-Position = 100

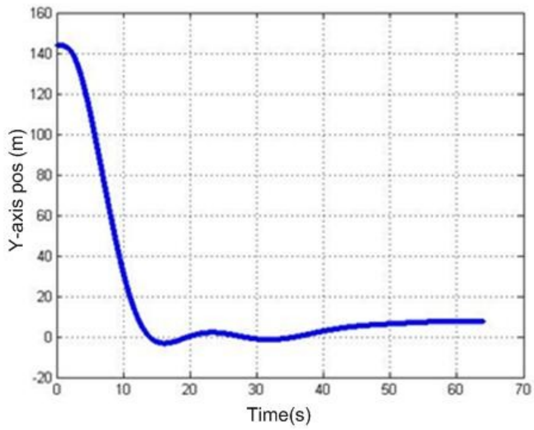

Figure 75 - Initial y-Position = 144

As a result of this algorithm some optimal reference command vectors are obtained, which include different command sets suitable for different scenarios. Therefore there is a need for using the correct command set for the correct scenario. First, all the generated data are collected in different vectors according to their use, called AltitudeData and yAxisData which are used for altitude commands and heading commands, respectively. A reference command selection block is implemented in Simulink using prelookup tables.

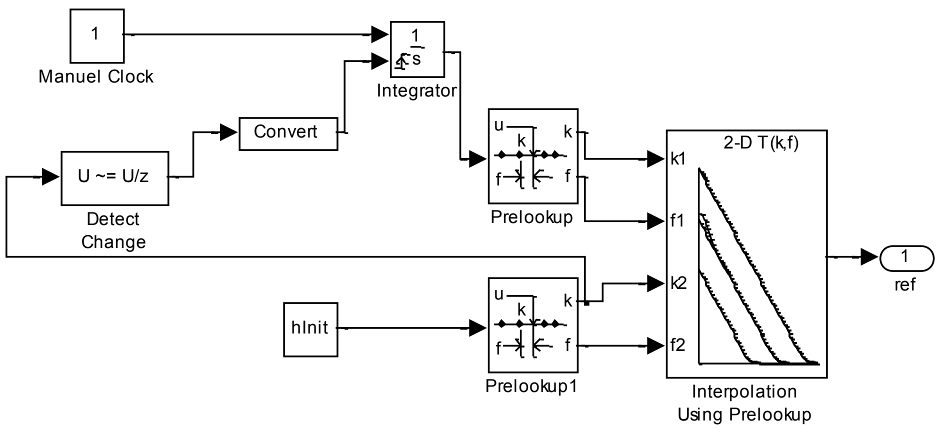

Figure 76 - Reference Command Selection Block for Altitude

Basically, the algorithm takes altitude or y axis data from the aircraft and uses it to decide which reference command set will be used according to the time index. Prelookup tables are used to pick the related row and column, and then interpolation block is used to output the data. The part before the prelookup tables is the manual clock (or index) used to select the specific reference command. Input of this block is the initial altitude or y axis data, which is entered by the user.

4.2 Lateral Track Control Law

A nonlinear track guidance algorithm for straight paths is implemented into the model [25], [1]. Lateral track controller is added to the outer loop, and desired psi commands are generated based on the so-called cross track error, which is the lateral position error between the commanded trajectory and the aircraft position. The purpose of the control is to produce rudder and aileron commands when there is cross track error, and level the aircraft when the error is zero. This guidance law requires waypoints (mission planning block will be mentioned in detail in Chapter 4.4) to guide the aircraft along the desired trajectory. Guidance law produces the commands necessary for the aircraft to reach the waypoints.

In this approach, the guidance law will try to align the aircraft velocity vector with the imaginary straight line between the aircraft and the selected point. The necessary psi command is found using the following equation:

$$\dot{\varphi}_{com} = K_R(kx_e\dot{y}_t - y_e\dot{x}_t) \tag{4.6}$$

where, x_e and y_e are the cross track errors, $\dot{x}_t$ and $\dot{y}_t$ are the velocity components parallel and normal to the desired track. The proportional gain K_R is obtained iteratively through simulations. Assume that Pioneer is flying at an arbitrary position relative to the track line between waypoints wp_1 and wp_2, and on an arbitrary heading ψ. Following notations are adopted: $\psi = \angle(\overline{U}, y_{North})$, $\psi_W = \angle(\overline{W}, y_{North})$ and $\psi_{12} = \angle(< wp_1, wp_2 >, y_{North})$. Given the Pioneer's velocity and position in the North-East reference frame, we are interested to find the velocity and position components in Track reference ($< x_t, y_t >$) frame. A transformation is applied as a rotation by an angle $(\psi_{12} - \pi/_2)$ and the resulting rotation matrix is given by

$$T_{\psi} = \begin{bmatrix} \cos(\psi_{12} - \pi/_2) & -\sin(\psi_{12} - \pi/_2) \\ sin(\psi_{12} - \pi/_2) & \cos(\psi_{12} - \pi/_2) \end{bmatrix} \quad (4.7)$$

Finally, along-track and cross-track velocities are obtained as

$$\dot{x}_t = U_{tx} + W_{tx} \quad (4.8)$$
$$\dot{y}_t = U_{ty} + W_{ty}$$

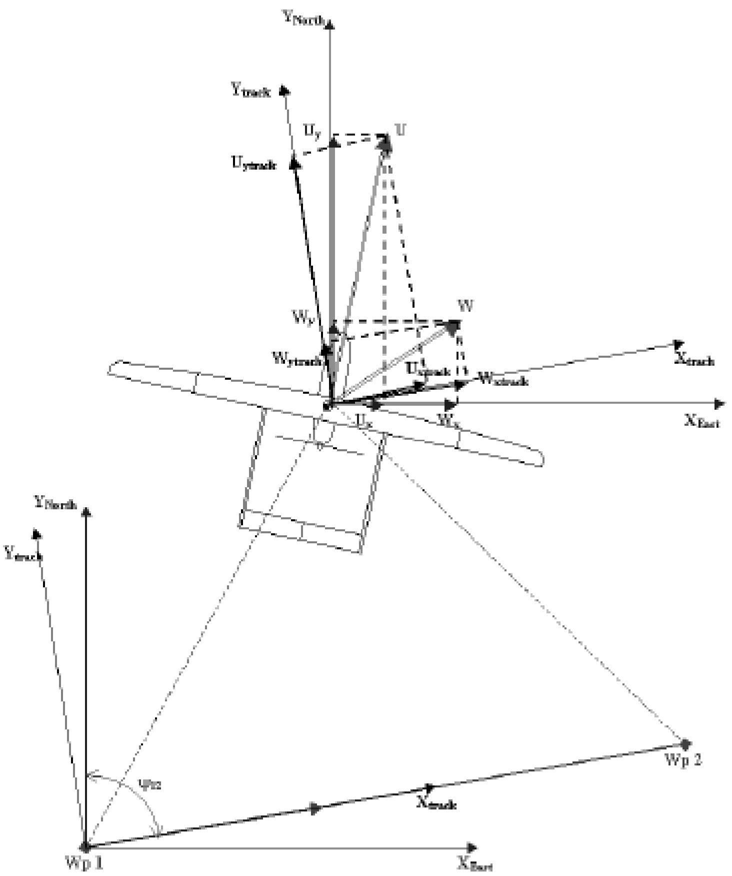

Figure 77 - Lateral Guidance Law Introduced by [25]

The point that the aircraft crosses the desired track is defined by the parameter k in the equation. k = 1 means that the aircraft directly goes to the desired waypoint.

However, it is not usually possible to achieve such a close tracking in the presence of uncertain conditions. Therefore, it is desired to use k < 1.

K_R is selected as -0.00001 for the simulations under no wind. Simulations run for different k values. The results are given below:

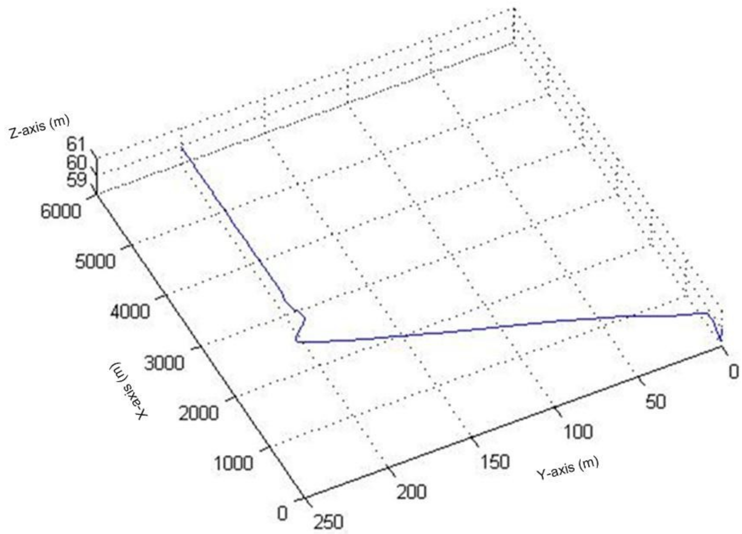

Figure 78 - k = 0.2

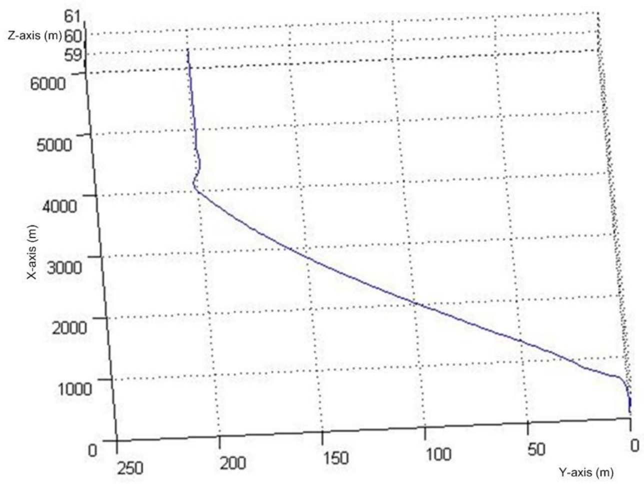

Figure 79 - k = 0.5

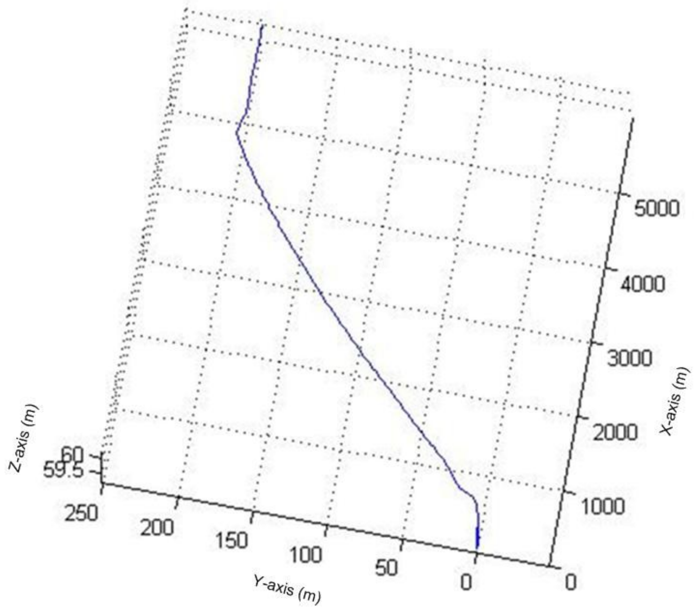

Figure 80 - k = 0.7

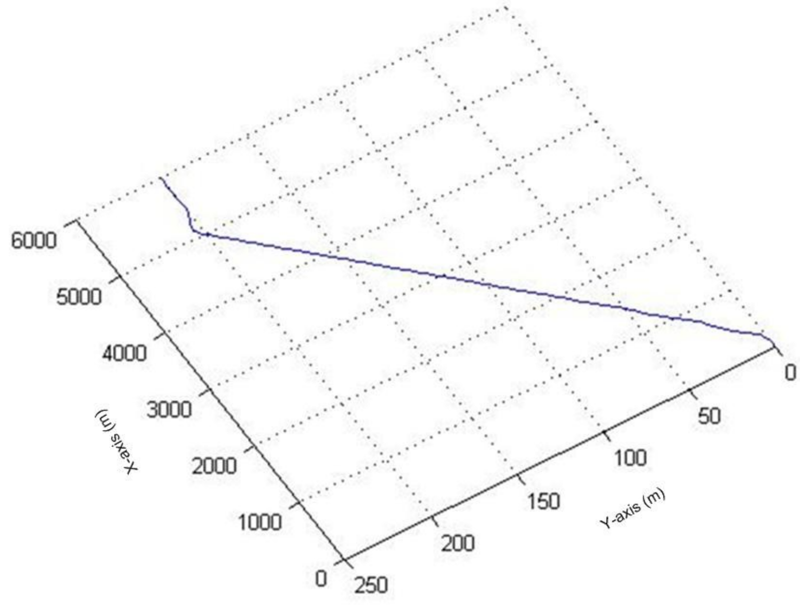

Figure 81 - k = 1

It can be observed that as k gets bigger the aircraft tends to go to the waypoint directly.

4.3 Cross Track Controller

A proportional-integral (PI) guidance with wind correction is implemented to the model in order to control the cross-track error of the aircraft while landing under crosswind.

The distance of the aircraft from the desired track is calculated and the cross-track error is obtained by subtracting it from the desired cross-track, which is 0. Then a ψ command is generated using the following formula:

$$\psi_{cmd} = K_p(xt_d - xt_{a/c}) + K_i \int (xt_d - xt_{a/c}) + \psi_{WCA} \tag{4.9}$$

Wind correction angle ψ_{WCA} is obtained using 4 values as input for calculation. (Wind direction, Wind speed, Airspeed and Desired Course Angle). This is the minimum required to define the triangle and orient it in relative to North. Here are the essential equations used to calculate the wind correction angle [26].

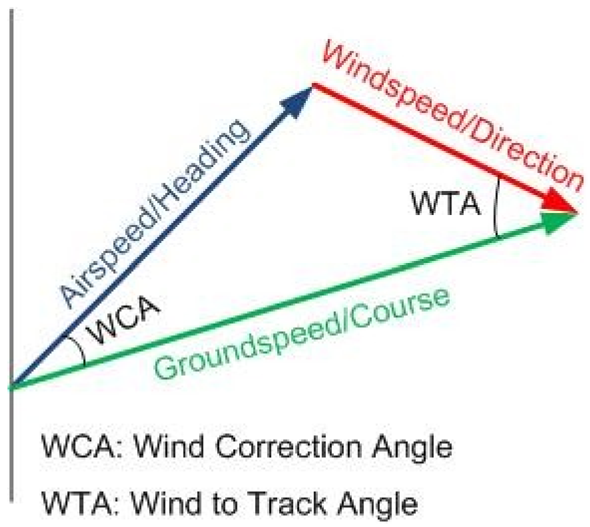

Figure 82 - Wind Triangle

$$windDir = InputWindDirection + 180° \tag{4.10}$$

Since wind directions are given as "From" direction rather than "To", 180 degrees will be added to get the vector point in the correct (downwind) direction.

$$WTAngle = DesiredCourse - windDir \tag{4.11}$$

$$SinWCA = windspeed * \frac{\sin(WTAngle)}{Airspeed} \tag{4.12}$$

$$WCA = sin^{-1}(sinWCA) \tag{4.13}$$

Wind Correction Angle (WCA) is defined to be the angle that the heading must deviate from the desired course to correct for the effect the wind is having on the aircraft.

4.4 Flight/Landing/Emergency Coordinator

Flight/Landing/Emergency coordinator is a simple flight/mission plan block that manages the waypoints for different scenarios. The block has three main functionalities:

1. Obtaining the next waypoint from the database according to the predefined conditions,
2. Deciding the aircraft's flight mode,
3. Deciding when to switch to the emergency plan during landing.

Waypoints are the database vectors holding the necessary flight information/trajectory for the aircraft. Waypoints can consist of several parameters serving for different purposes. In this thesis, x, y coordinates, altitude and the airspeed of the aircraft is defined in a waypoint. The aircraft autonomously sense that a waypoint is reached and activate the next waypoint. The first functionality parses the current waypoint into related aircraft desired flight information, and when the aircraft reaches the next waypoint it updates the current waypoint information with that of the next waypoint.

There are two mission plans uploaded on the aircraft, one of them is emergency plan and the other is the normal flight plan. All wp's are defined as fly-by as defined in the Figure 83.

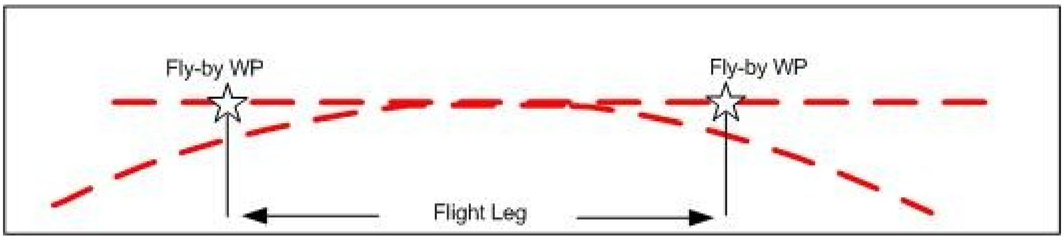

Figure 83 - Fly-by Waypoint

The waypoint update criterion is defined as:

If

- $aircraft\ is\ in\ 300m\ radius\ of\ x\ and\ y\ axis\ of\ next\ wp$ **AND**
- $altitude\ of\ next\ wp\ is\ in\ \pm 10m$ **AND**
- $speed\ of\ the\ next\ wp\ is\ in\ \pm 1m/s$

A waypoint navigation example is given below:

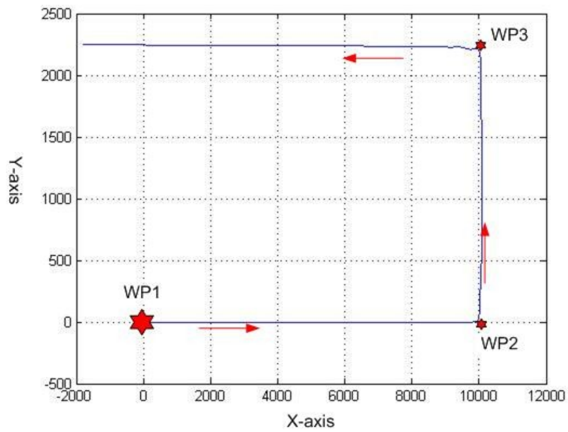

Figure 84 - WP Navigation Example

The aircraft has 3 main modes: Normal Flight, Landing, Emergency. The flight mode is decided using the predefined definitions for flight zones. Normal flight mode is the mode when the aircraft is outside landing zone. As the aircraft reaches the landing zone, it has to decide whether to land or abort landing due to its position. For the aircraft to decide landing, it must be in landing cone and satisfying predefined conditions. If the aircraft is in landing zone, but not in the cone, then it will change its mode as Emergency.

In Emergency Mode, aircraft will ascend to a predefined waypoint outside the landing zone and then try to enter the cone again. When the aircraft enters the Emergency Mode, it will stay in it until the Emergency Plan is completed. Then the aircraft decides its mode according to the zone that it is in. Aircraft zones are defined in Figure 85. Zone 1 is defined as the landing zone, which resides in the landing cone. Zone 2 is the normal flight zone, which the aircraft navigates through waypoints. Zone 3 is a critical zone, which means that the aircraft is in landing zone, but not in the landing cone. In landing zone, aircraft is expected to be in the landing cone to accomplish a safe landing maneuver, and if it is not, emergency plan is executed. After the final window, the landing cannot be aborted as the altitude is very low, and there is a risk to crash the aircraft while aborting the landing maneuver.

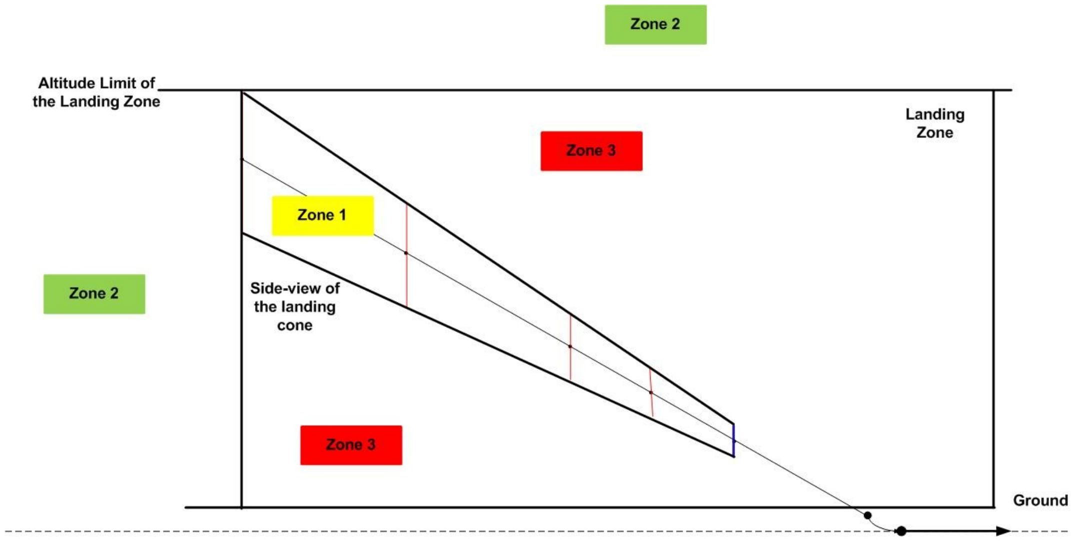

Figure 85 - Flight Zones (Zone 1 = Land, Zone 2 = Normal Flight, Zone 3 = Emergency)

Different emergency scenarios are run and in all of them mode transitions and tracking accuracies are satisfactory. The runway starting coordinates are taken as (x = 10000, y = 200, z = 0). It is assumed that the runway is 3000 m. long and 30 m. width to make it similar to standard commercial runways. An emergency waypoint is also defined for this runway at x = 15000, y = 200, z = 300. It is obvious that this emergency waypoint would possess different properties at different runways.

- **Scenario 1**: Aircraft is in landing zone but below the landing cone (xi = 9000, yi = 200, zi = 35). In this scenario, it is assumed that the aircraft could not meet the corridor entrance plane at first trial, and stayed below the cone. It is expected that the aircraft defines its mode as emergency and navigates to the emergency waypoint. This is a challenging situation in terms of algorithm, because the aircraft enters the landing cone during its flight to the emergency waypoint. At this point, aircraft should not change its mode, because it may be close to the runway and may not have enough time and space to accomplish its maneuver, and safely land the aircraft.

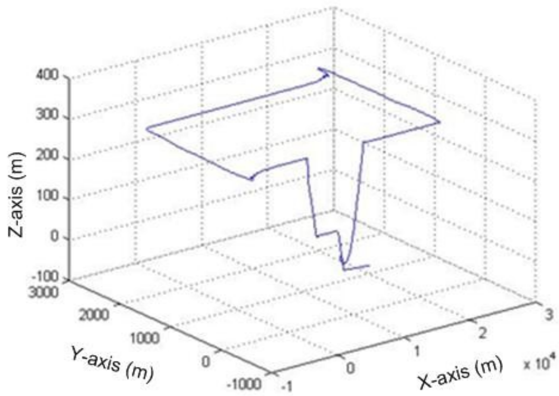

Figure 86 – Emergency Test – Scenario 1

- **Scenario 2:** Aircraft is in landing zone but on below left side of the cone (xi = 9000, yi = 350, zi = 50). In this scenario, it is assumed that the aircraft could not meet the corridor entrance plane at first trial, and stayed at the east of the cone. Again, it is expected that the aircraft defines its mode as emergency and navigates to the emergency waypoint. However, this time a lateral maneuver should be made, since the emergency waypoint is on the left of the initial y-axis position.

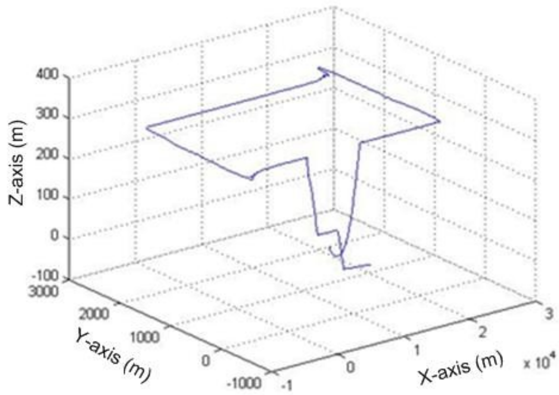

Figure 87 - Emergency Test – Scenario 2

- **Scenario 3:** Aircraft is in landing zone but above the cone (xi = 9000, yi = 200, zi = 150). In third scenario, it is assumed that the aircraft could not meet the corridor entrance plane at first trial, and stayed above the cone. It is expected that the aircraft defines its mode as emergency and navigates to the emergency waypoint like in scenario 1, but this is a simpler scenario such that the aircraft does not enter the cone to reach the emergency waypoint.

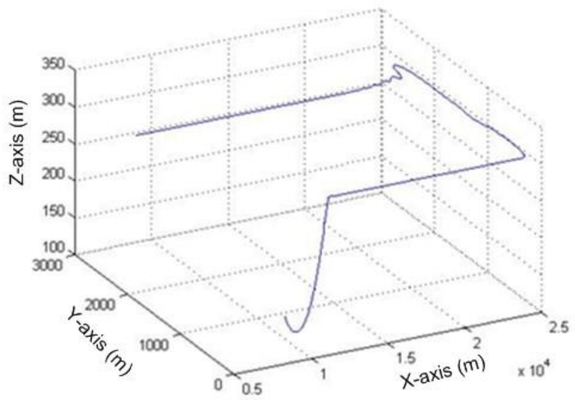

Figure 88 - Emergency Test – Scenario 3

In first two scenarios, the aircraft detected as it is in landing zone but not in landing cone, so that it changed its mode as Emergency and applied the Emergency plan. As the Emergency Plan is accomplished, the aircraft detects that it is in landing cone and land successfully. In third scenario, landing is not simulated due to its similarity to other scenarios, but a successful change of aircraft mode is observed.

- **Scenario 4:** Aircraft is in landing zone but below the cone (xi = 9950, yi = 200, zi = 35). In this scenario, it is assumed that the aircraft leaves the cone from below at a point very close to the runway (50 m), which could endanger the landing. This point resides at the end of the cone, and it is the last chance of the aircraft to keep the final track. It can be observed from the simulation results that the aircraft manages to keep the desired track and completes the landing maneuver after touring above runway and re-entering the cone.

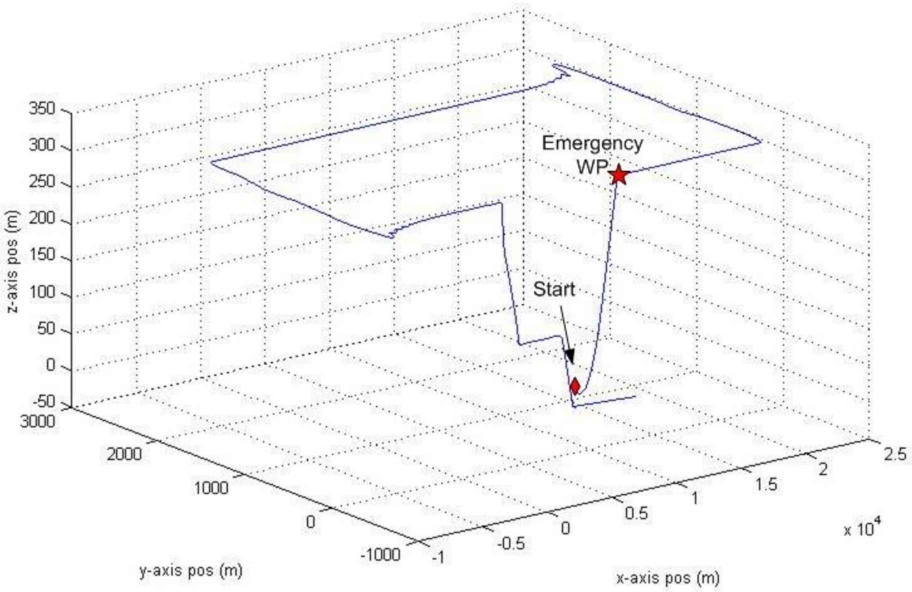

Figure 89 - Emergency Test – Scenario 4

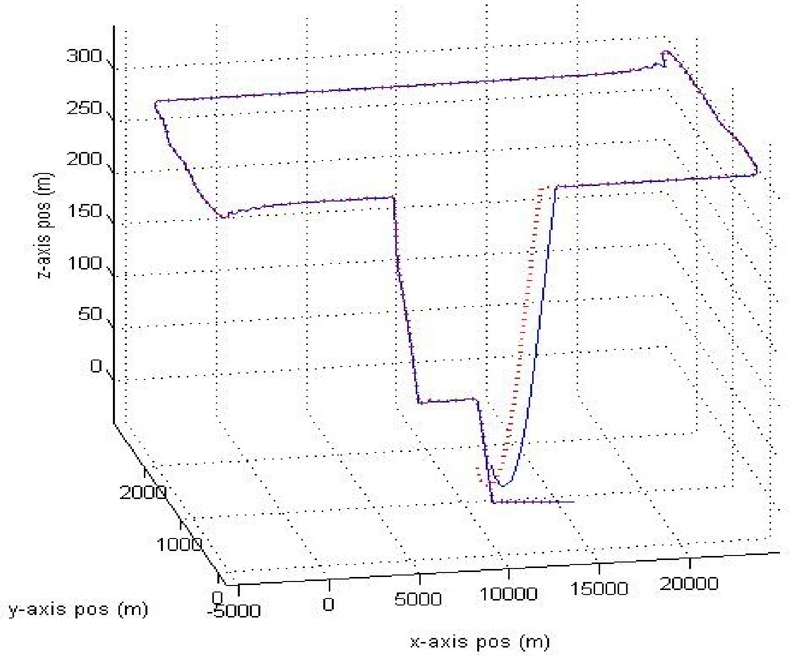

Figure 90 - Emergency Test – Scenario1 (dotted) vs. Scenario4 (solid)

The simulation results of Scenario 1 and Scenario 4 are compared in Figure 90. The behavior of the aircraft is similar in both cases. The aircraft loses altitude before it starts to ascend to the emergency waypoint as expected. This is emanating from the fact that the aircraft's flight path angle is negative at the beginning of the maneuver. The Scenario 4 is critical in the sense that it is close to the runway, and the decision speed is critical. Fast, but true decision is the only requirement from the emergency coordinator block, and no unsafe situations are observed during simulations.

These simulations are run assuming that there is no wind disturbance present in the environment, in order to show the decision mechanism works well. Wind effects on the aircraft and the aircraft's attitude in the cone is discussed in Chapter 5.

CHAPTER 5

SIMULATION RESULTS

Here, different scenarios are generated for testing the controllers and guidance mechanisms designed on previous chapters. The aim is to create a challenging situation for the controller during landing and test its robustness. Wind is applied as an external disturbance to the system and the scenarios are generated based on various wind speeds and directions. The following three controllers are tested:

1. Controller 1: PID Controller with Lateral Track Guidance and Cross-track Guidance
2. Controller 2: Sliding Mode Controller (Linear Method) with Cross-track Guidance
3. Controller 3: Sliding Mode Controller (Nonlinear Method) with Cross-track Guidance
4. Controller 4: LQT with Cross-track Guidance

Emergency coordinator is not activated in the simulations, in order to observe the complete behavior of the aircraft under wind.

5.1 Scenarios

5.1.1 Scenario 1: Constant Wind Speed, Same Direction

For this case, a constant wind is applied from the same direction to the aircraft during landing. The simulation is repeated for different wind speeds and directions. This is a challenging situation since the controller has to cope with a non-ceasing disturbance. The landing trajectory for altitude is defined as following:

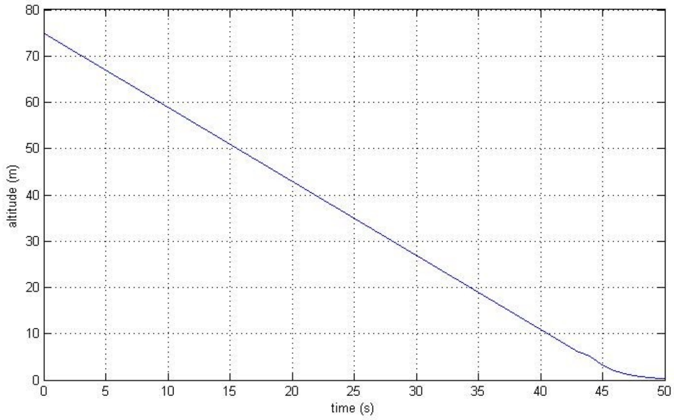

Figure 91 - Landing Trajectory Defined for Simulations

Airspeed and y-axis position is taken as 60 m/s and 200, respectively.

- **Scenario 1.1: Windspeed = 1, 5, 10 m/s, Wind Direction = 90° (From East)**

Controller 1 with lateral track guidance is analyzed.

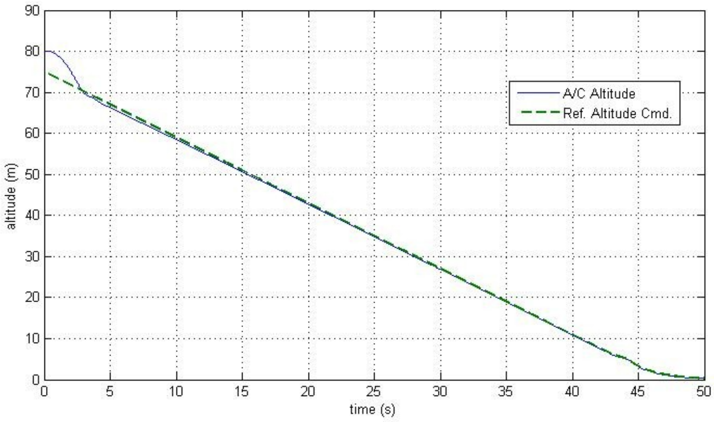

Figure 92 - Scn 1.1 Altitude Plot for Controller 1 with LTG

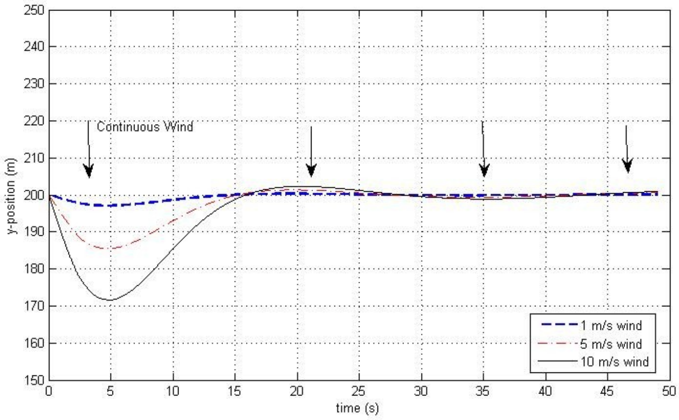

Figure 93 - Scn 1.1 y-position Plot for Controller 1 with LTG

It can be observed from the figures that the altitude controller is not affected by the disturbance, but there is a slight effect on lateral control. Then, the controller beats the disturbance effect and starts following the trajectory.

Controller 1 with Cross-Track guidance is analyzed:

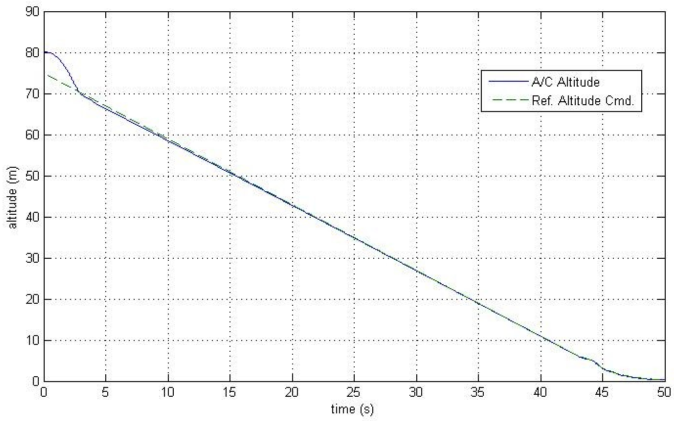

Figure 94 - Scn 1.1 Altitude Plot for Controller 1 with CTG

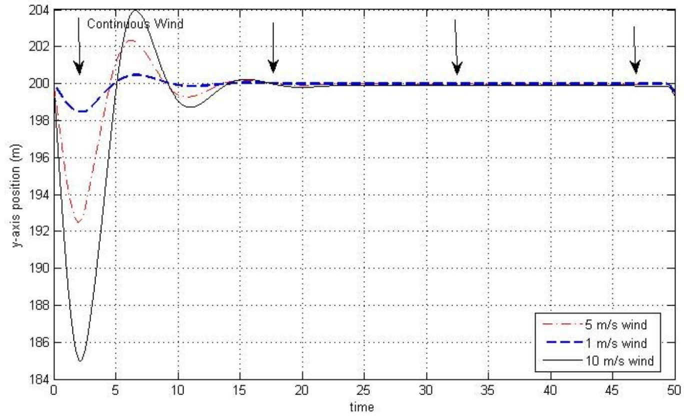

Figure 95 - Scn 1.1 y-Position Plot for Controller 1 with CTG

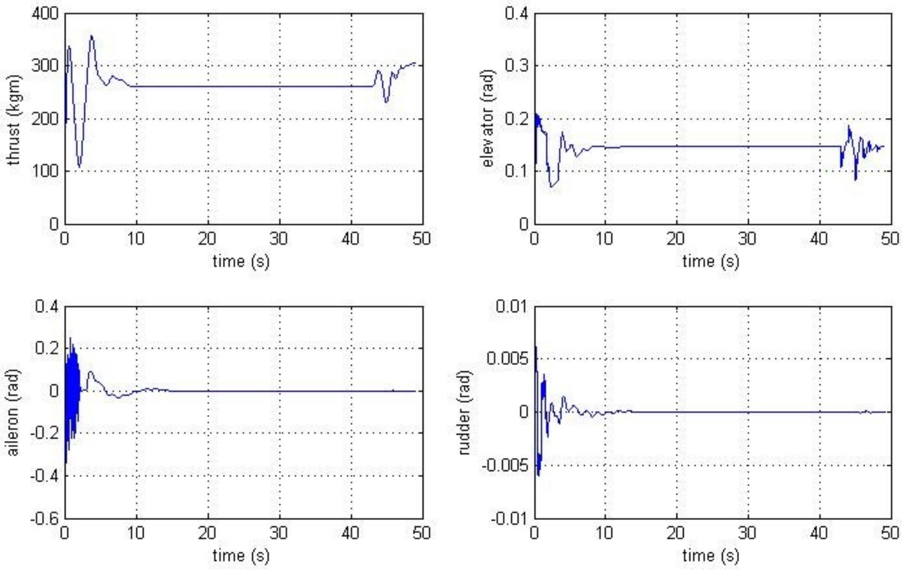

Figure 96 - Scn 1.1 Control Surfaces Plot for Controller 1 with CTG (10 m/s wind)

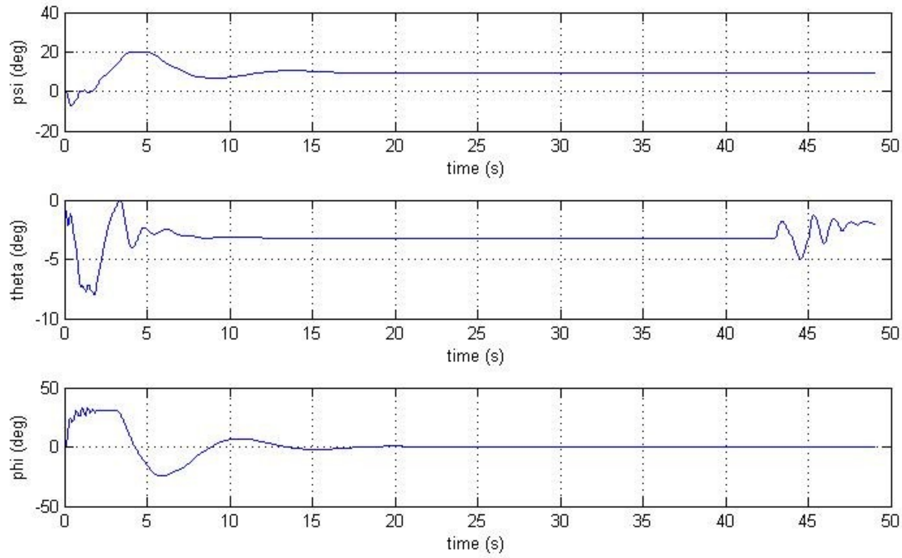

Figure 97 - Scn 1.1 Euler Angles for Controller 1 with CTG (10 m/s wind)

Altitude controller behavior is almost the same as LTG; however, the lateral controller response is faster. It can be seen that there is an over-shoot on the response. In order to return the aircraft to the track again, Controller 1 uses aileron extensively. After reaching the track frequency of the aileron commands reduces to normal levels. Pitch angle of the aircraft is -2 degrees at the moment of touchdown, which is an acceptable value.

Then, the Controller 2 is simulated using scenario 1.1.

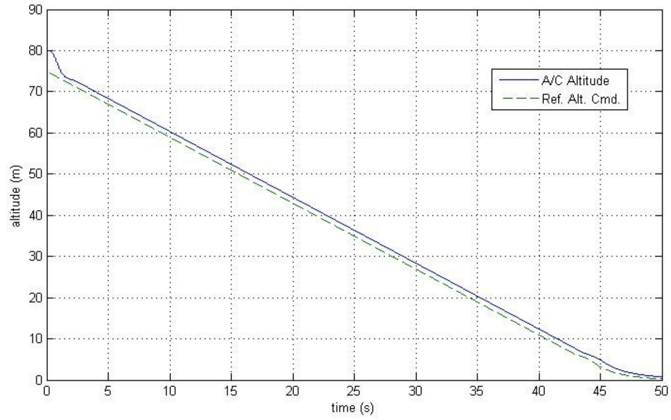

Figure 98 - Scn 1.1 Altitude Plot for Controller 2

The tracking performance of the controller is very good; however, a small offset is observed. The same response is observed under every wind condition.

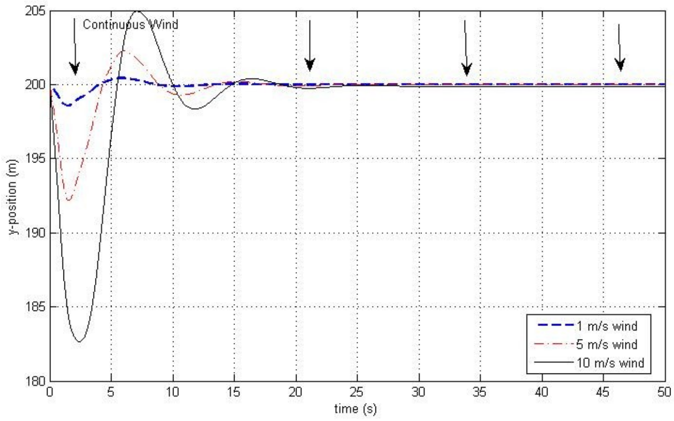

Figure 99 - Scn 1.1 y-Position Plot for Controller 2

The lateral controller response is very fast. The overshoot effect is also observed here.

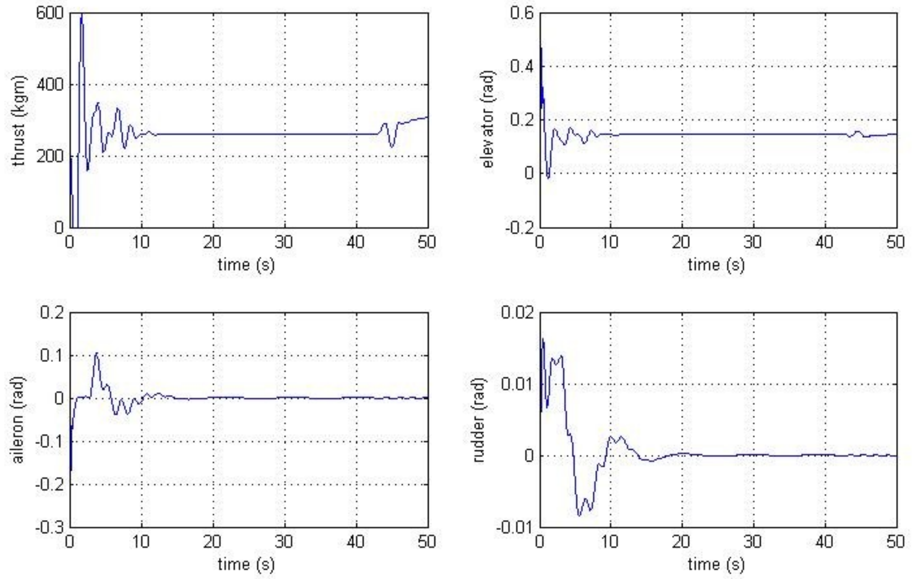

Figure 100 - Scn 1.1 Control Surfaces Plot for Controller 2 (10 m/s wind)

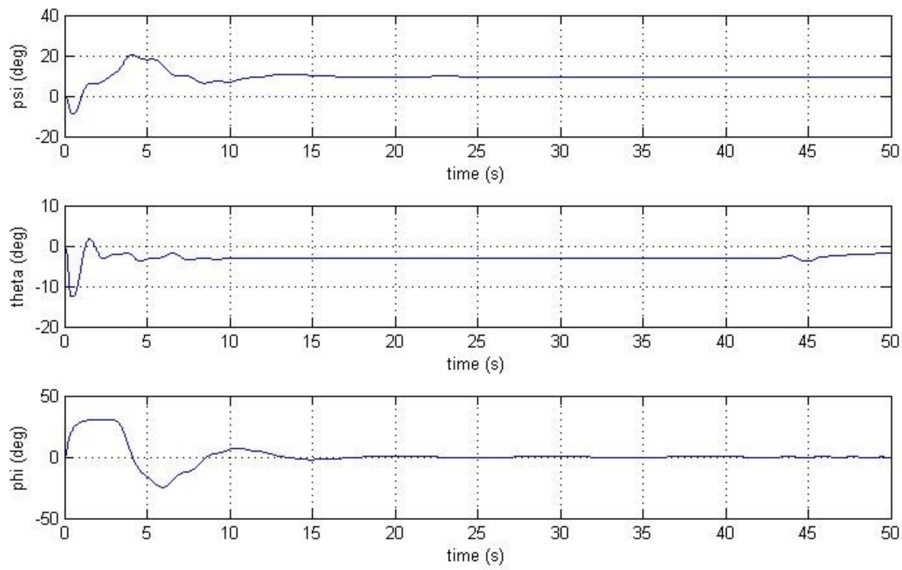

Figure 101 - Scn 1.1 Euler Angle Plot for Controller 2 (10 m/s wind)

A noticeable improvement in control surface deflections is observed with respect to the Controller 1. Aileron deflection frequency is very low even during the lateral maneuver of the aircraft.

Controller 3 is analyzed using the same scenario.

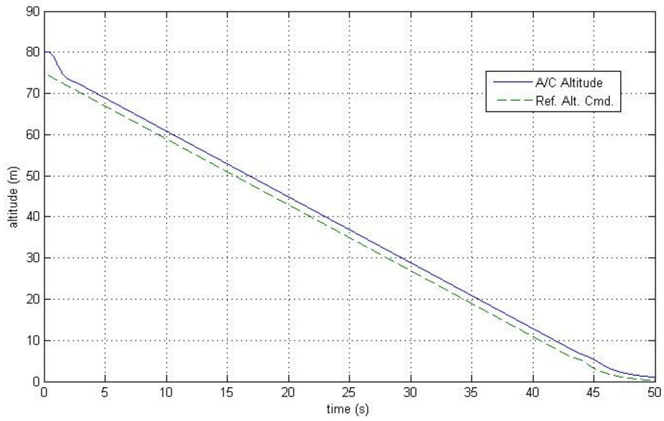

Figure 102 - Scn 1.1 Altitude Plot for Controller 3

Altitude controller response for the Controller 3 is very similar to the Controller 2. There is a small offset, which does not have an effect on landing. Altitude response is almost the same on every wind speed.

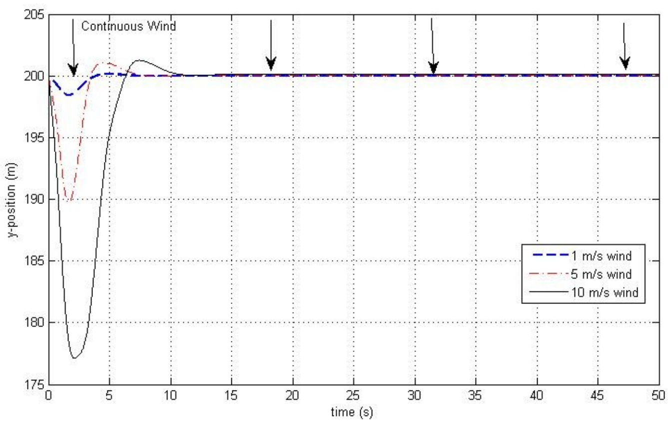

Figure 103 - Scn 1.1 y-Position Plot for Controller 3

The lateral response of the Controller 3 is very fast and the overshoot is very small. The response of the Controller 3 in lateral axes is the best among all other controllers. It converges fast enough to get on track before touch down if the range is suitable. However, the results show that the cross-wind speeds higher than 5 m/s emerging at close ranges (ranges smaller than 5 m.) put the landing maneuver into danger.

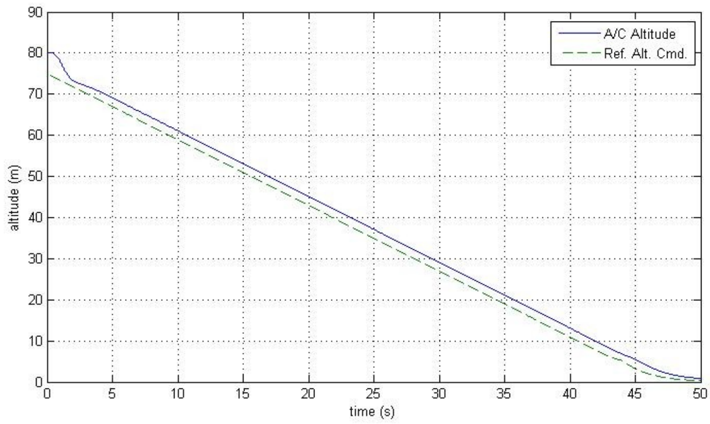

Figure 104 - Scn 1.1 Altitude Plot for Controller 4

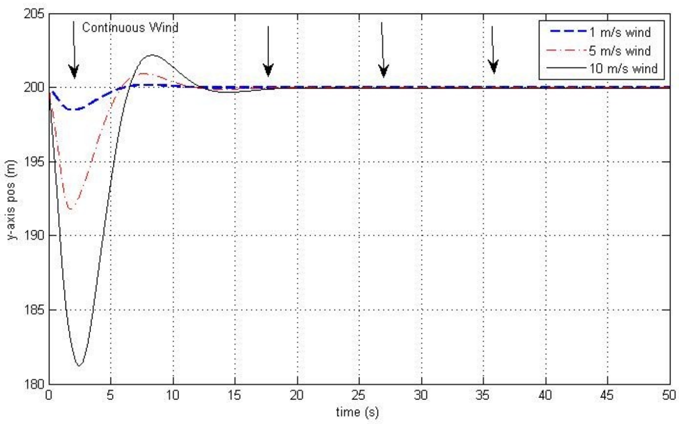

Figure 105 - Scn 1.1 y-Position Plot for Controller 4

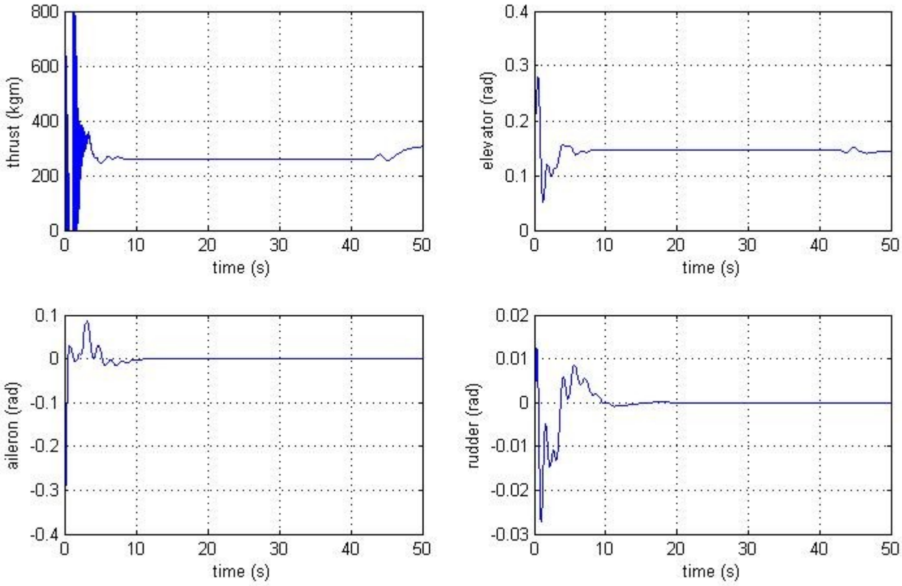

Figure 106 - Scn 1.1 Control Surfaces Plot for Controller 4 (10 m/s wind)

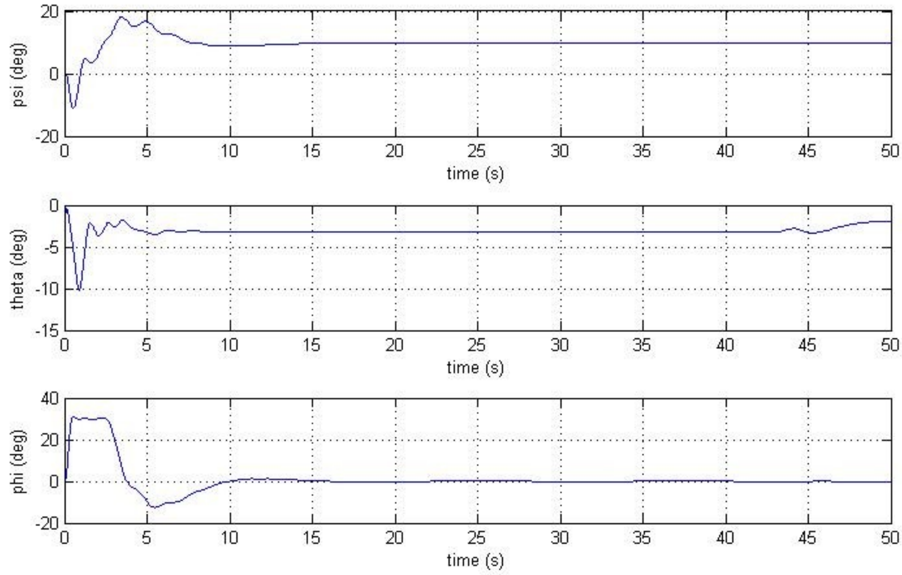

Figure 107 - Scn 1.1 Euler Angles Plot for Controller 4 (10 m/s wind)

Y-axis position of the aircraft is plotted for all controllers under 10 m/s crosswind. It shows the performances of the controllers with respect to each other.

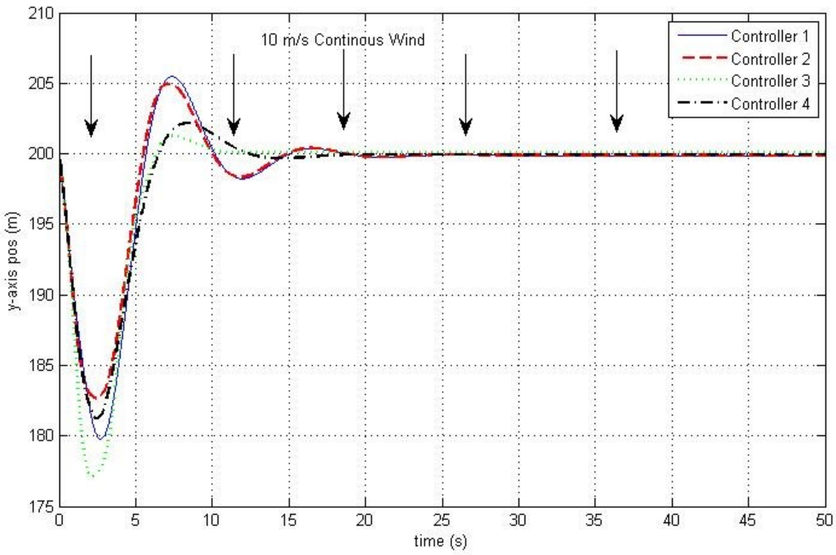

Figure 108 – Scn 1.1 y-Axis position for all Controllers (10 m/s wind)

Initial lateral response of the controller to the disturbance is fast according to the Controller 3, but slower than the Controller 2. Time for the aircraft to settle down to the track is in between the two other controllers. Nevertheless, it is superior to Controller 1 in all cases. Control surface deflections, and Euler angles are in acceptable limits, such that maximum aileron deflection is 17 degrees. A yaw angle different than zero is observed at the moment of touchdown, which is a result of the continuous crosswind. This can be eliminated by a de-crab maneuver just before touchdown. Roll angle of the aircraft reaches to 30 degrees during the maneuver to reach the desired y-axis position. It is the maximum value that the controller can generate, because of the envelope protection applied to controller commands. Controller would definitely be generating higher commands, if there were no envelope protections. Changing the envelope limits will affect the time that the tracking error reduces to zero; however, it will also affect the safety limits of the aircraft.

❖ **Scenario 1.2: Windspeed = 1, 5, 10 m/s, Wind Direction = 180° (From South)**

In this scenario, wind direction is changed such that it is coming from south (tail-wind), and the controllers' response is analyzed. Here, the y-axis position is not expected to change; for this reason only altitude and airspeed is plotted.

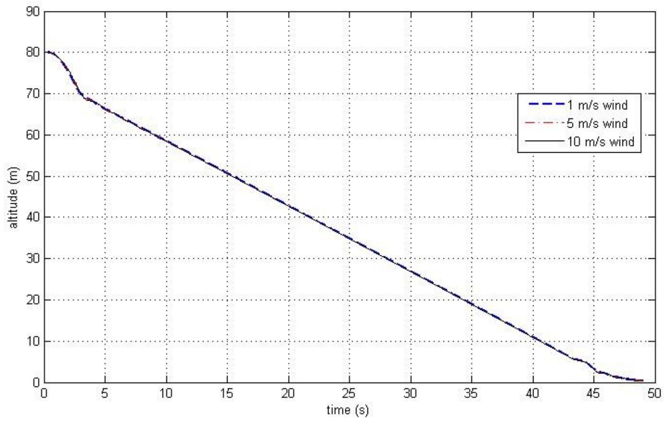

Figure 109 - Scn 1.2 Altitude Plot for Controller 1

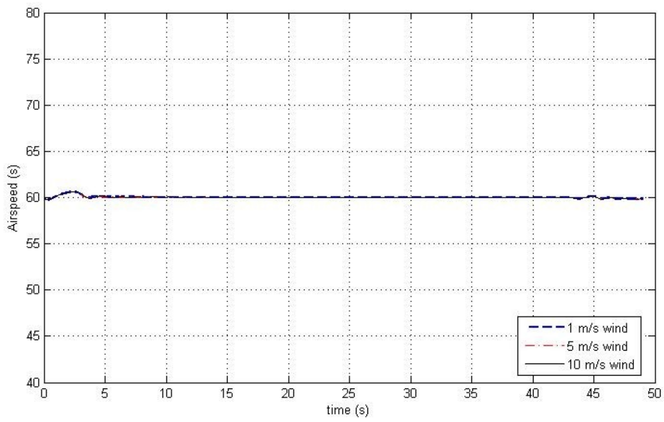

Figure 110 - Scn 1.2 Airspeed Plot for Controller 1

The results show that the altitude controller is not affected from the disturbance, and the lateral controller is slightly affected. As the wind speed gets higher airspeed starts to oscillate, but the controller manages to track the reference. The reason for the initial and final deviations in the airspeed is the longitudinal maneuvers of the aircraft. The first maneuver is for matching the altitude to the reference command, and the final is the flare maneuver.

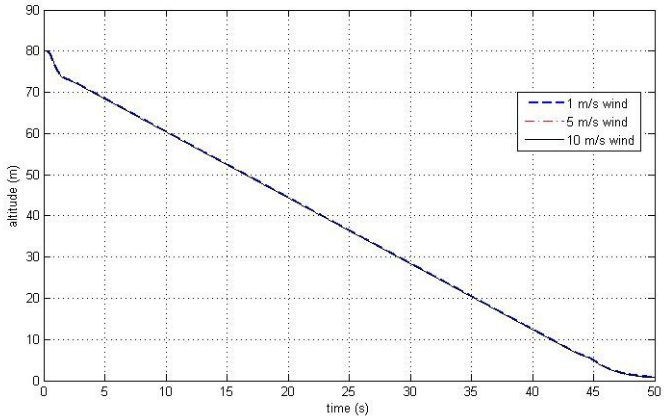

Figure 111 - Scn 1.2 Altitude Plot for Controller 2

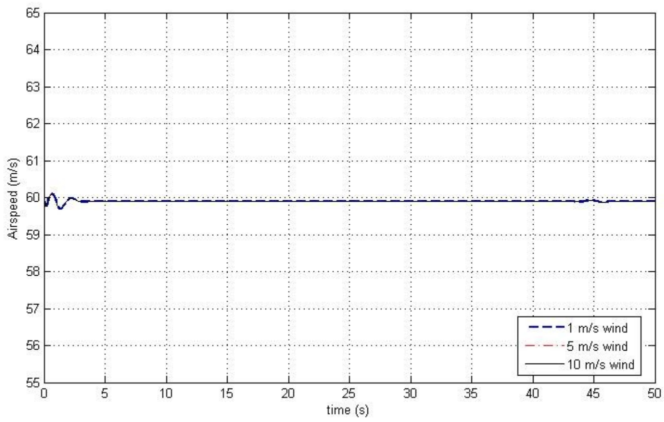

Figure 112 - Scn 1.2 Airspeed Plot for Controller 2

Figure 111 and Figure 112 tell us that Controller 3 is not affected by the tail winds up to 10 m/s.

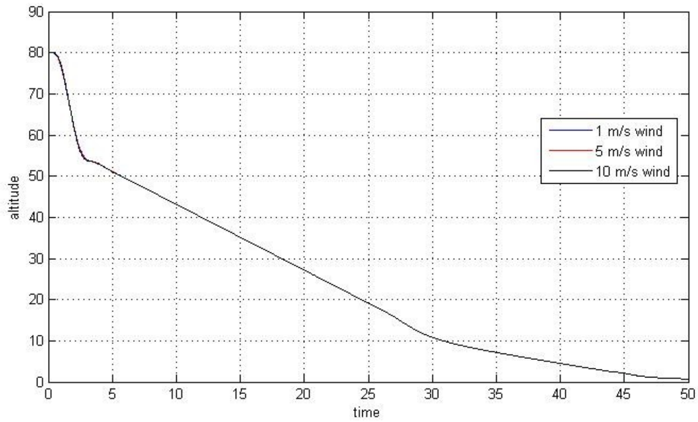

Figure 113 - Scn 1.2 Altitude Plot for Controller 3

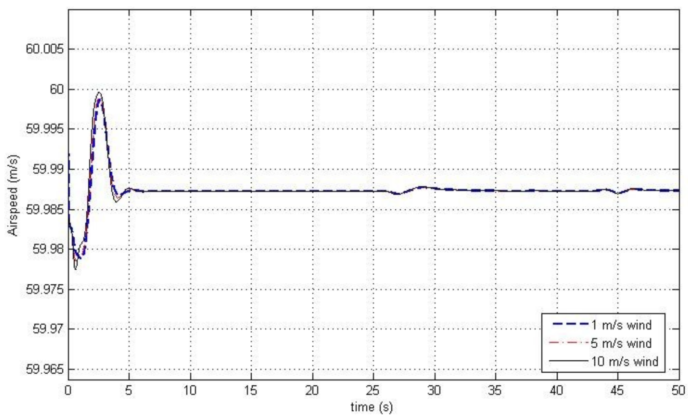

Figure 114 - Scn 1.2 Airspeed Plot for Controller 3

Controller 3 is also not affected tail winds up to 10 m/s. Small variations occur in the airspeed due to the wind but it only changes the touch-down point a few centimeters.

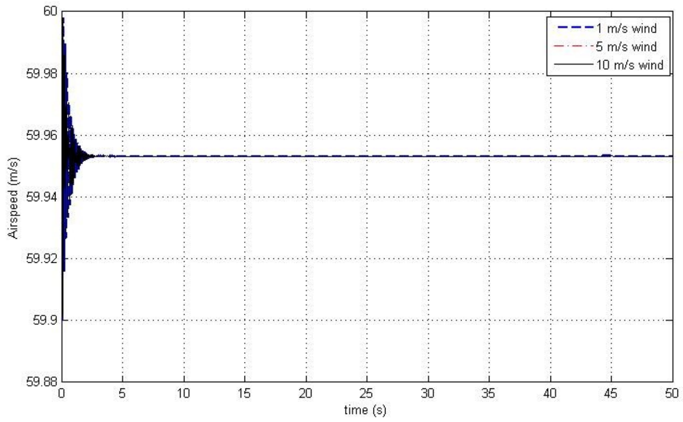

Figure 115 - Scn 1.2 Airspeed Plot for Controller 4

No effect is seen on Controller 4 airspeed block performance under tailwind.

- ❖ **Scenario 1.3: Windspeed = 1, 5, 10 m/s, Wind Direction = 0° (From North)**

In this scenario, wind direction is changed such that it is coming from north (head-wind), and the controllers' response is analyzed. Again, here the y-axis position is not expected to change, for this reason only altitude and airspeed is plotted.

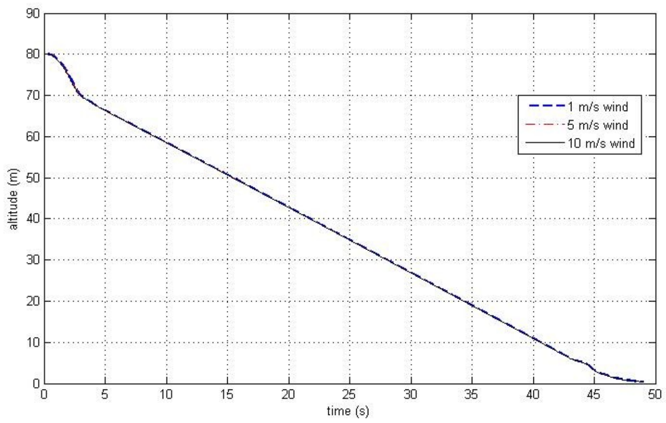

Figure 116 - Scn 1.3 Altitude Plot for Controller 1

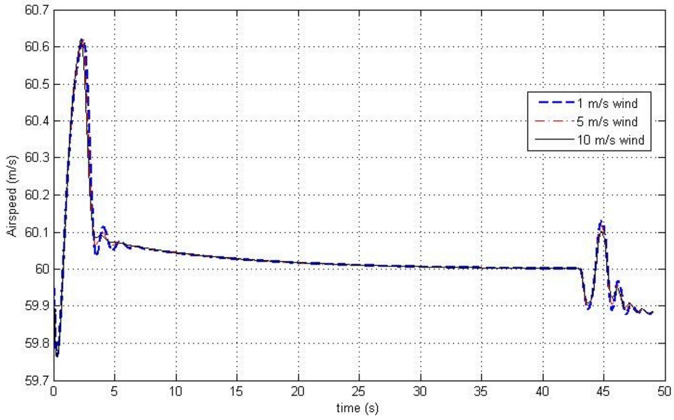

Figure 117 - Scn 1.3 Airspeed Plot for Controller 1

The altitude response of the aircraft is not affected by the head-wind, and there is a slight affect on the airspeed. There is a slight increase in the aircraft airspeed.

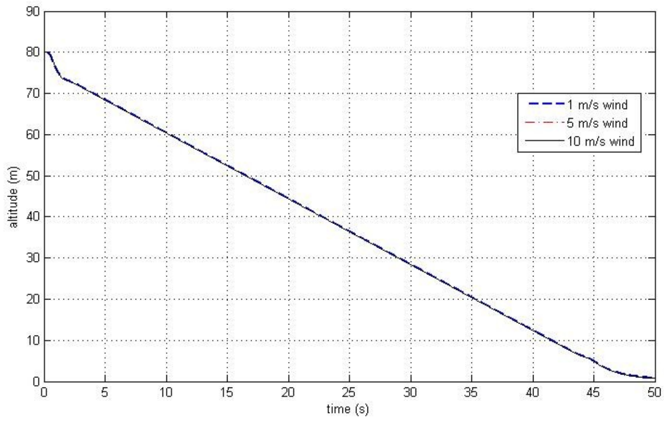

Figure 118- Scn 1.3 Altitude Plot for Controller 2

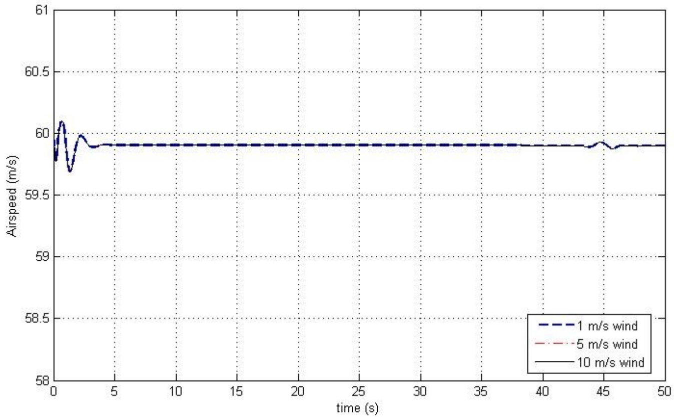

Figure 119 - Scn 1.3 Airspeed Plot for Controller 2

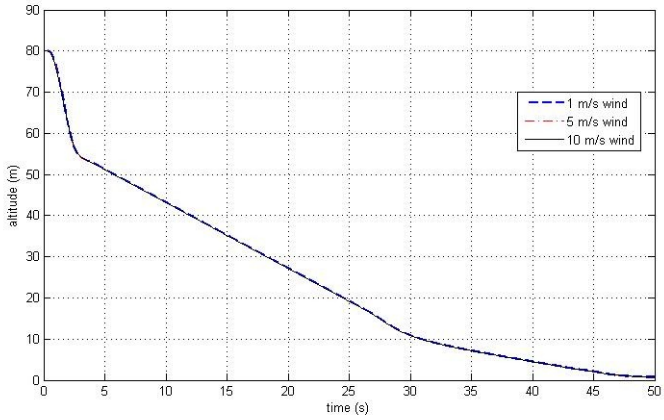

Figure 120 - Scn 1.3 Altitude Plot for Controller 3

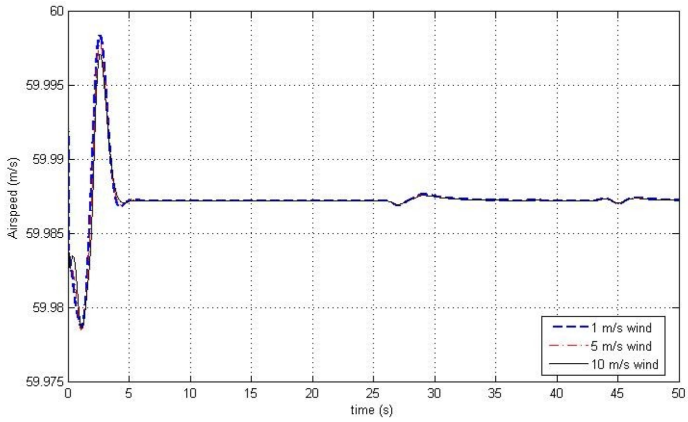

Figure 121 - Scn 1.3 Airspeed Plot for Controller 3

These results prove us that the Controller 2 and Controller 3 are capable of keeping the aircraft on track under head-winds up to 10 m/s. There is a very little effect of wind on the airspeed, and it is only visible during the longitudinal maneuvers. It can be stated that under continuous tail and head winds even starting at the flare maneuver, the aircraft can hold its track successfully.

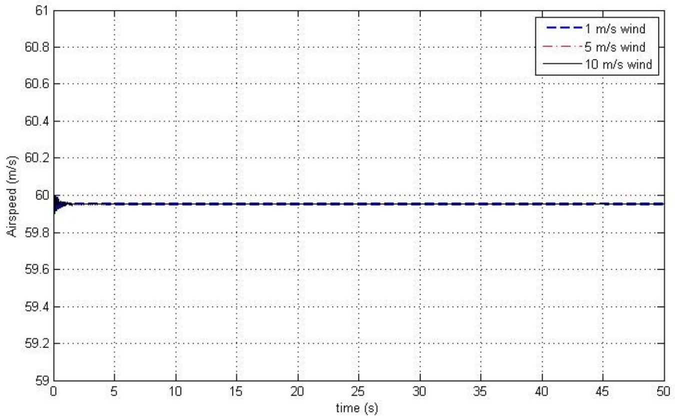

Figure 122 - Scn 1.3 Airspeed Plot for Controller 4

5.1.2 Scenario 2: Constant Wind Speed, Same Direction, Different Duration

In this scenario, wind speed, and direction are kept the same during one simulation. However, wind durations will be kept small (about 1-2 seconds) and since it was proven that the winds other than cross-wind do not have serious impact on landing maneuver, the simulations will be focused on cross-wind scenarios. With this scenario, it is aimed to test the controllers' performance for instantaneous winds (gust like winds).

❖ **Scenario 2.1: Windspeed = 5, 10, Wind Direction = 90°, Wind Duration = 1 sec.**

In this scenario, wind duration is changed such that it is not a continuous wind, and applied at a random time of the simulation, which is close to touchdown point. The criteria for success is to land the aircraft at a point which is max ±2 m of the center of the runway, since the runway is assumed to be 8-10 meters. Simulations are stopped at the touchdown moment to see the aircrafts y-axis position while touching the landing gears down.

Following is the simulation result of y-axis position for the case that wind is applied at the simulation time of 40 seconds. That is the time when the aircraft altitude is about 11 meters.

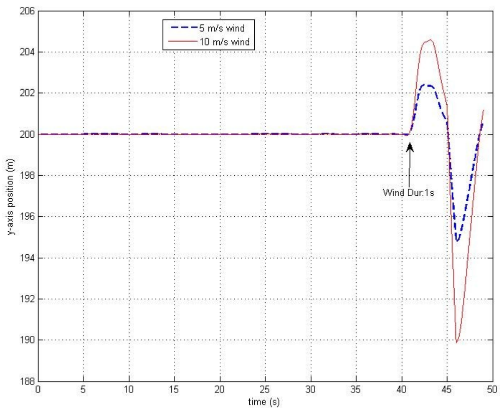

Figure 123 - Scn 2.1 y-Axis Position Plot for Controller 1 with CTG

The Controller 1 with CTG hardly managed to keep the aircraft within the desired range under 5 and 10 m/s winds. The heading of the aircraft at that moment is very close 2 degrees, in both cases.

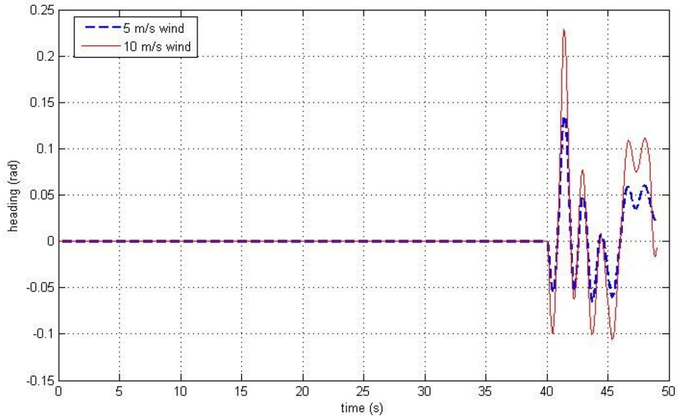

Figure 124 - Scn 2.1 Heading (in radians) Plot for Controller 1 with CTG

This shows us that the Controller 1 with CTG can land the aircraft under 5 and 10 m/s cross-winds that are encountered at the last 10-11 meters of the landing phase; however, beyond that altitudes safety of the landing is put into danger.

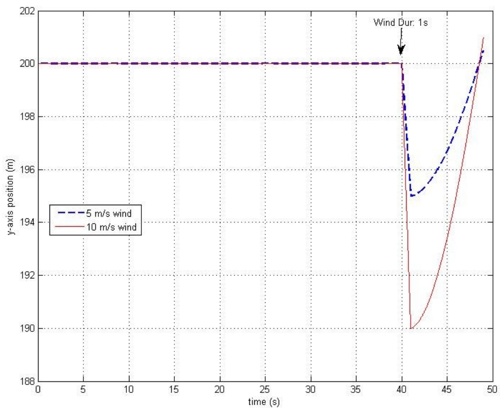

Figure 125 - Scn 2.1 y-Axis Position Plot for Controller 1 with LTG

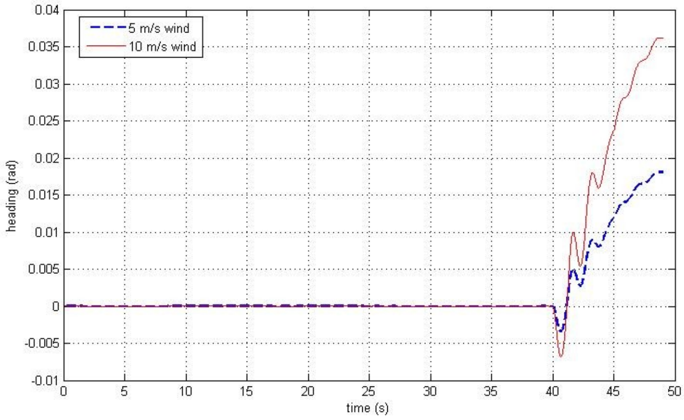

Figure 126 - Scn 2.1 Heading (in radians) Plot for Controller 1 with LTG

Controller 1 with LTG response is slower with respect to the one with CTG; however, heading angle of the aircraft at the touchdown point is smaller.

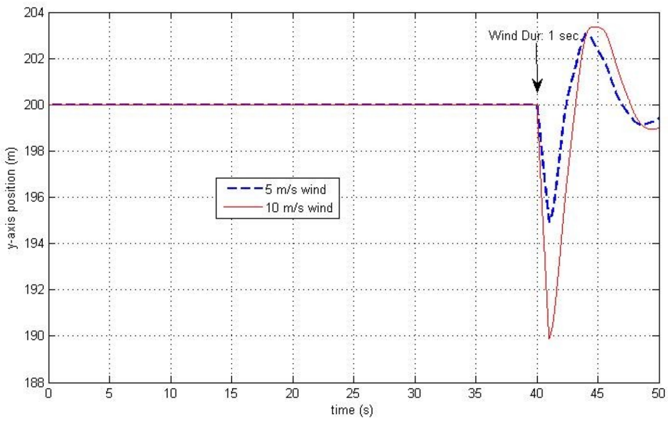

Figure 127 - Scn 2.1 y-Axis Position Plot for Controller 2

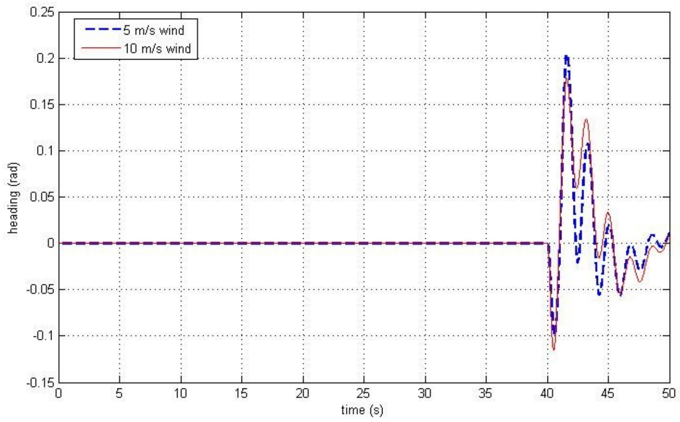

Figure 128 - Scn 2.1 Heading (in radians) Plot for Controller 2

Y-Axis position of the aircraft is in the desired limits for both wind conditions, and heading of the aircraft is almost zero. The results of Controller 2 are appropriate for safe landing.

The following figures represent the simulation results for the Controller 3. The controller is very fast and robust, for this reason, the results are the best among all others. Controller 3 is tested also under 15 m/s crosswind and managed to keep the aircraft on track.

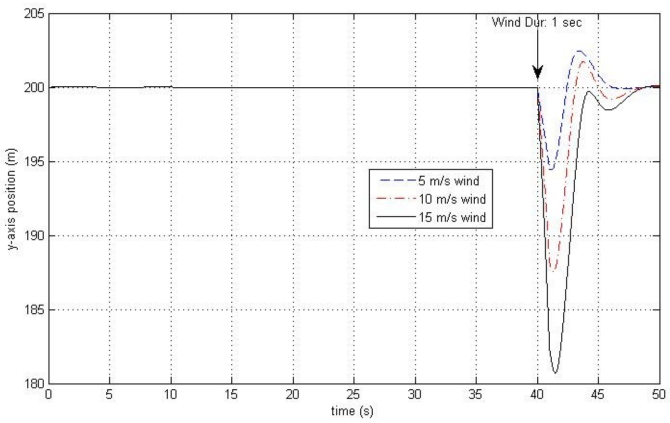

Figure 129 - Scn 2.1 y-Axis Position Plot for Controller 3

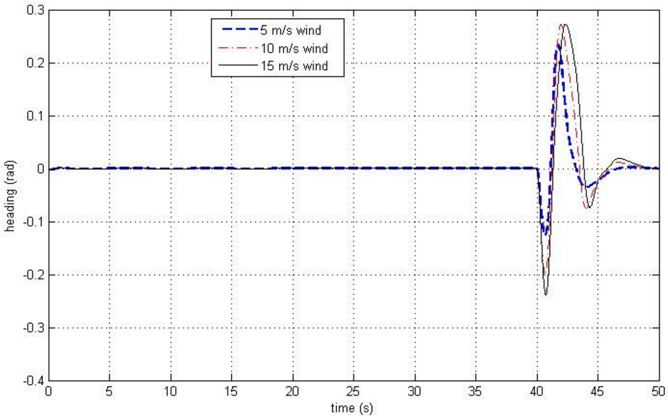

Figure 130 - Scn 2.1 Heading (in radians) Plot for Controller 3

Performance of Controller 4 under wind gust started at the last 11 meters of landing phase is given in Figure 131:

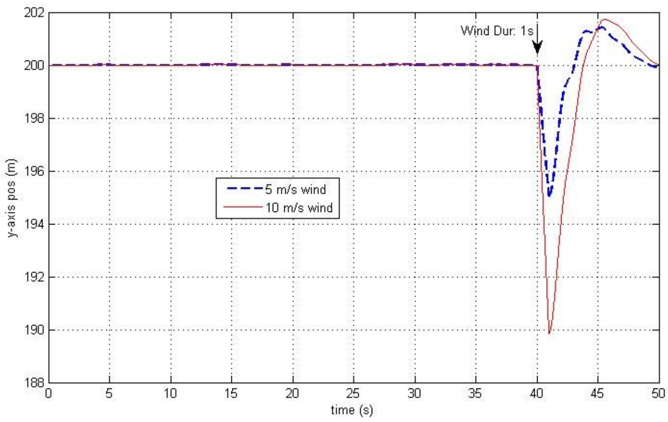

Figure 131 - Scn 2.1 y-Axis Position Plot for Controller 4

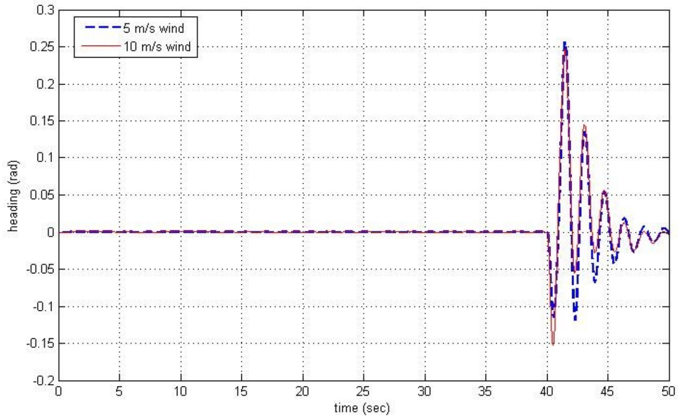

Figure 132 - Scn 2.1 Heading (in radians) Plot for Controller 4

Controller 4 performance for scenario 2.1 is acceptable for landing.

- **Scenario 2.2: Windspeed = 10, 15, Wind Direction = 90°, Wind Duration = 1 sec.**

In this scenario, conditions that are more adverse are generated and the performances of the Controller 3 and Controller 4 are analyzed, since these have the best performance among all under crosswind.

For this case, 2 wind gusts are applied one after another with 2 seconds of interval.

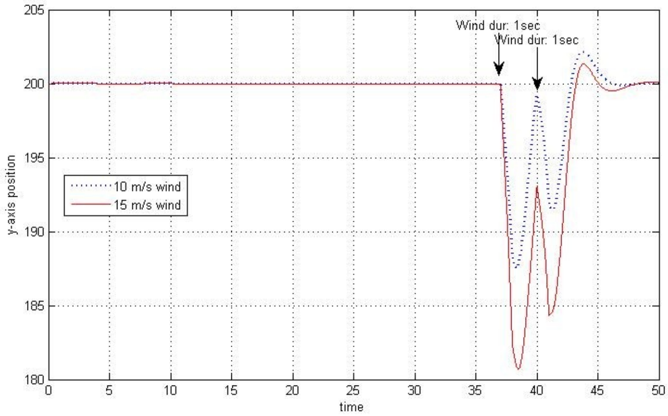

Figure 133 - Scn 2.2 y-Axis Position Plot for Controller 3

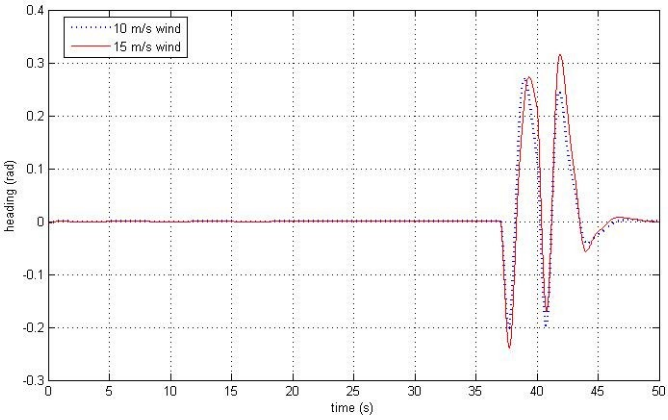

Figure 134 - Scn 2.2 Heading (in radians) Plot for Controller 3

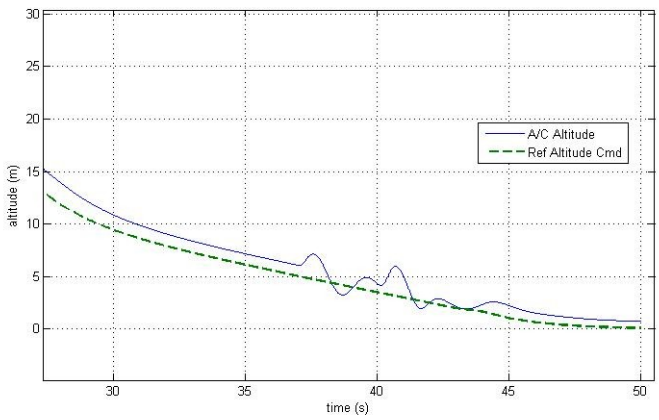

Figure 135 - Scn 2.2 Altitude Plot for Controller 3 (15 m/s wind)

Some oscillations are observed in aircraft altitude during landing under 15 m/s wind. At some points, aircraft loses 2-4 meters of altitude, which can be dangerous at lower altitudes of landing.

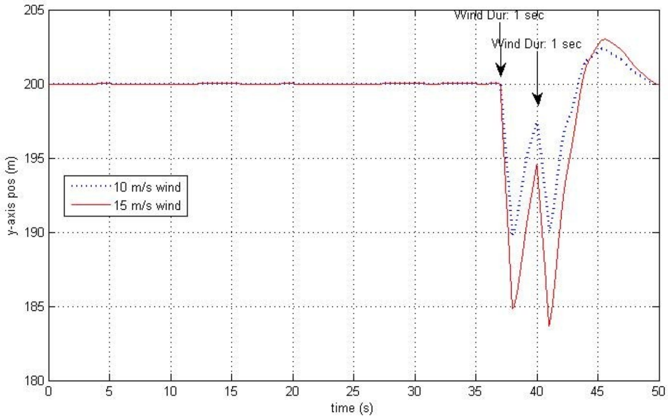

Figure 136 - Scn 2.2 y-Axis Position Plot for Controller 4

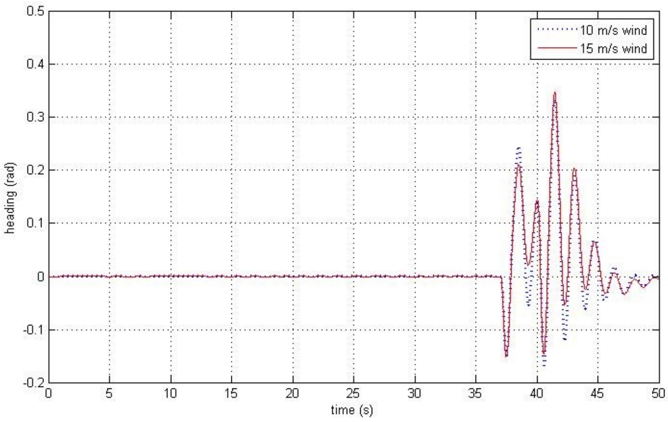

Figure 137 - Scn 2.2 Heading (in radians) Plot for Controller 4

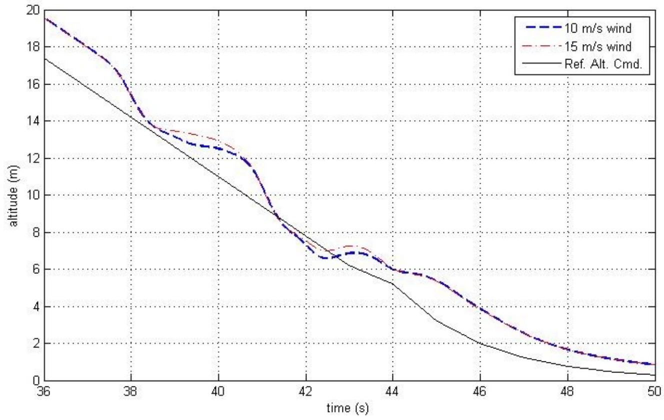

Figure 138 - Scn 2.2 Altitude Plot for Controller 4

Lateral axis performance of Controller 4 is adequate for landing; however, like in Controller 3 oscillations are observed in altitude of the aircraft during roll maneuvers. The amplitude of altitude loss is not high as much as Controller 3, and so Controller 4 is better at high-speed gust like crosswinds.

- **Scenario 2.3: Windspeed = 10, 15, Wind Direction = 90°, -90°, Wind Duration = 1 sec.**

In this scenario, even harder case will be analyzed such that wind direction, airspeed changes during landing.

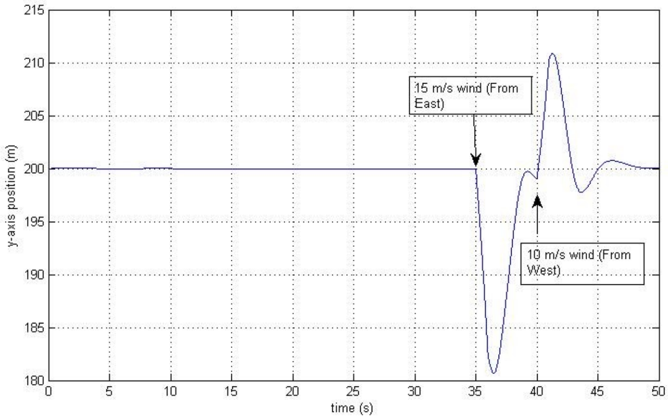

Figure 139 - Scn 2.3 y-Axis Position Plot for Controller 3

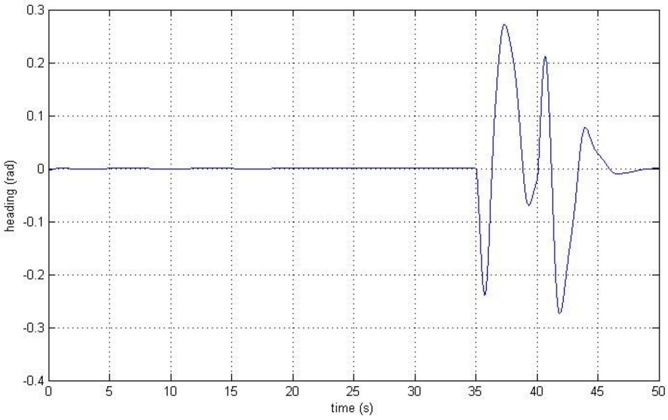

Figure 140 - Scn 2.3 Heading (in radians) Plot for Controller 3

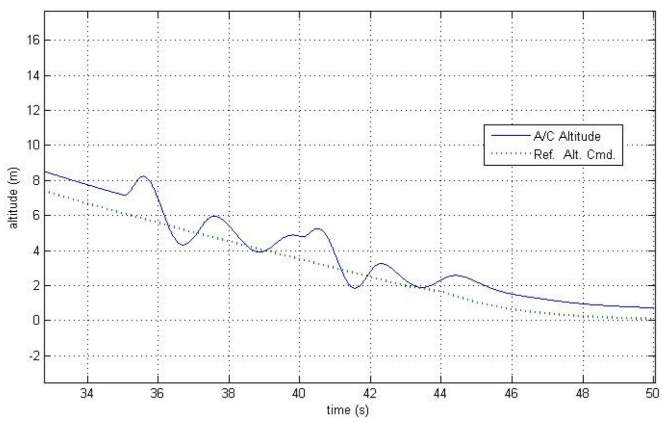

Figure 141 - Scn 2.3 Altitude Plot for Controller 3

The effects of change in wind direction can be seen directly on y-axis position of the aircraft. The controller maneuvers the aircraft in order to keep it on track, and hence some oscillations in aircraft altitude are observed. Like in previous scenario, the aircraft altitude was enough to tolerate those oscillations.

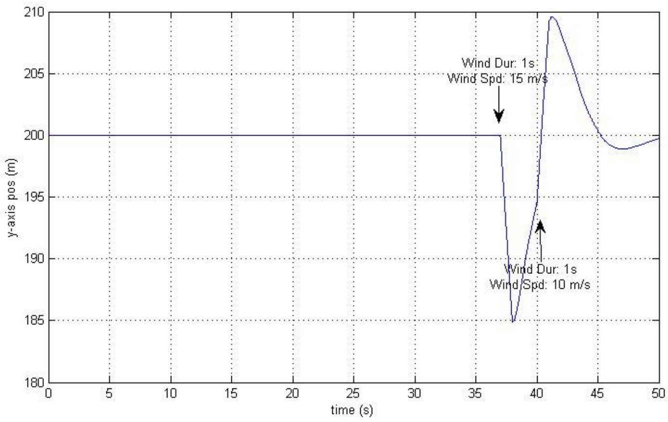

Figure 142 - Scn 2.3 y-Axis Position Plot for Controller 4

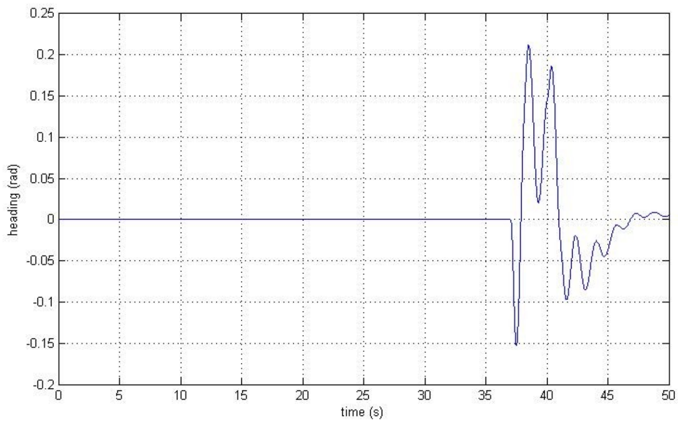

Figure 143 - Scn 2.3 Heading (in radians) Plot for Controller 4

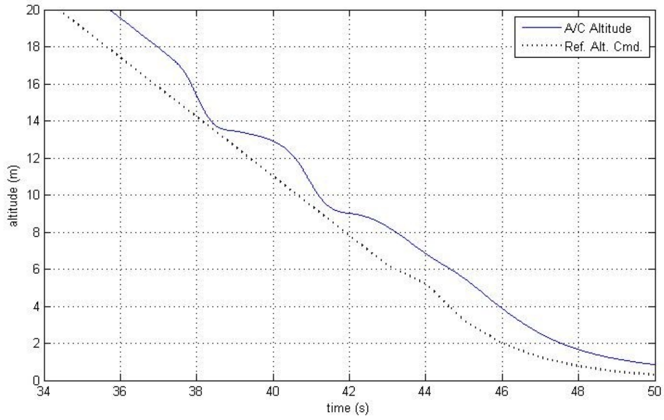

Figure 144 - Scn 2.3 Altitude Plot for Controller 4

1-2 meters of change in altitude is observed; however, the movements of aircraft in longitudinal axis are very smooth compared to Controller 3. Both controllers managed to front the runway in desired ranges.

5.1.3 Scenario 3: Different Wind Speed, Different Direction

In this scenario, continuous wind is assumed to be coming from 2 axes at the same time. The amplitudes of wind vectors are also different from one another.

- **Scenario 3.1: Windspeed = 5, 10, Wind Direction = 45°**

Continuous wind is applied to aircraft from northeast, and the performance of Controller 1, 2 and 3 are analyzed.

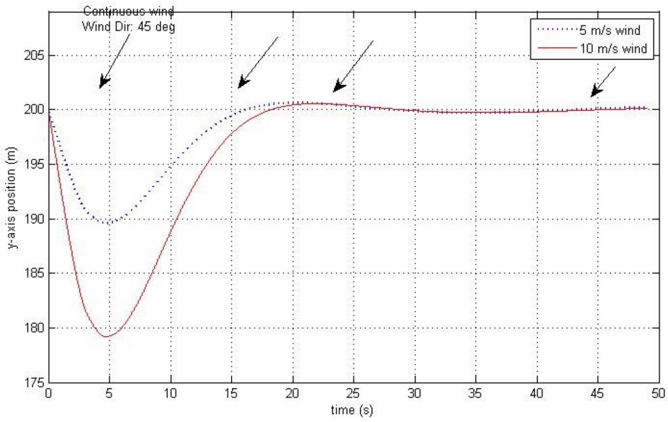

Figure 145 - Scn 3.1 y-Axis Position Plot for Controller 1 with LTG

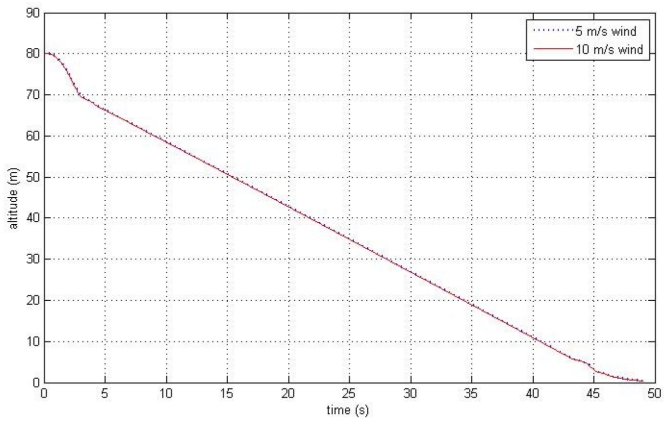

Figure 146 - Scn 3.1 Altitude Plot for Controller 1 with LTG

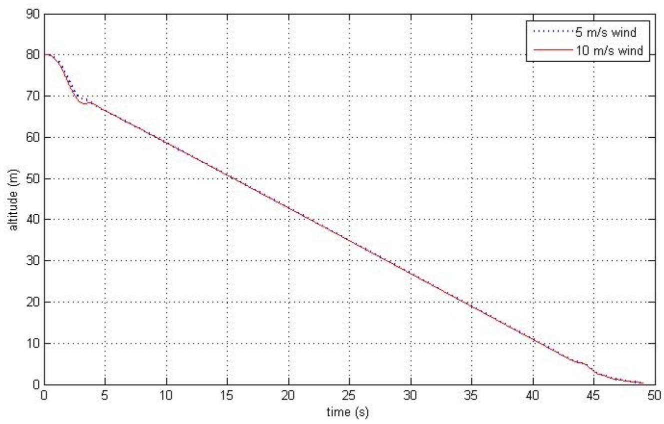

Figure 147 - Scn 3.1 Altitude Plot for Controller 1 with CTG

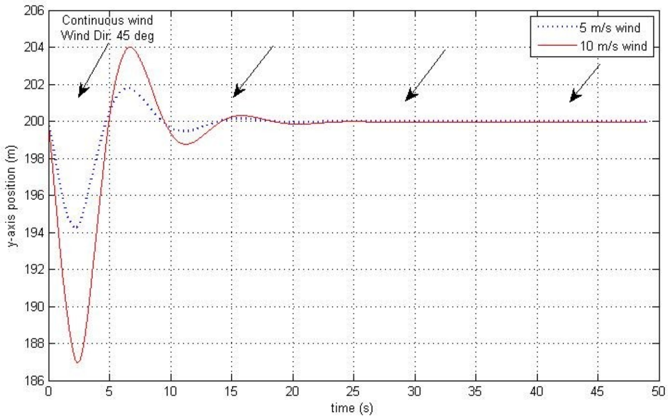

Figure 148 - Scn 3.1 y-Axis Position Plot for Controller 1 with CTG

Controller 1 with CTG is faster with respect to Controller 1 LTG; however, it has some amount of overshoot. Still, both of them manage to keep the aircraft on track. It can also be observed that altitude controller is not affected from long-term winds. The same response is expected from all other controllers.

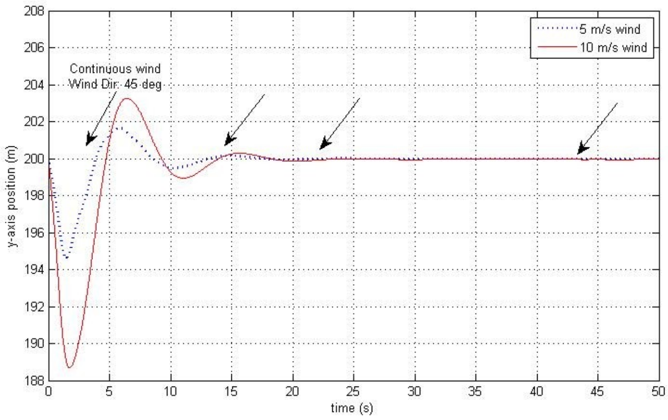

Figure 149 - Scn 3.1 y-Axis Position Plot for Controller 2

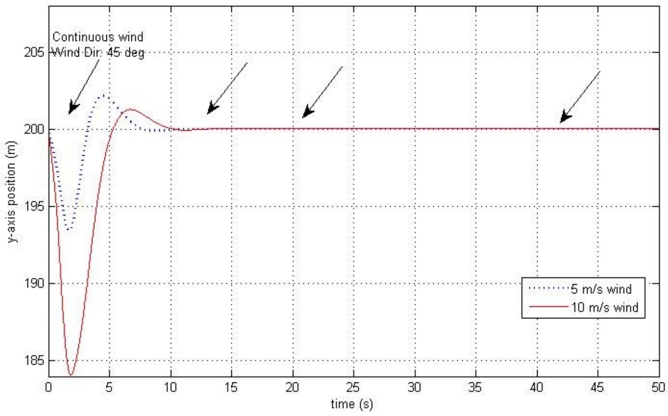

Figure 150 - Scn 3.1 y-Axis Position Plot for Controller 3

There is an overshoot in Controller 2, but Controller 3 can track the trajectory with little overshoot.

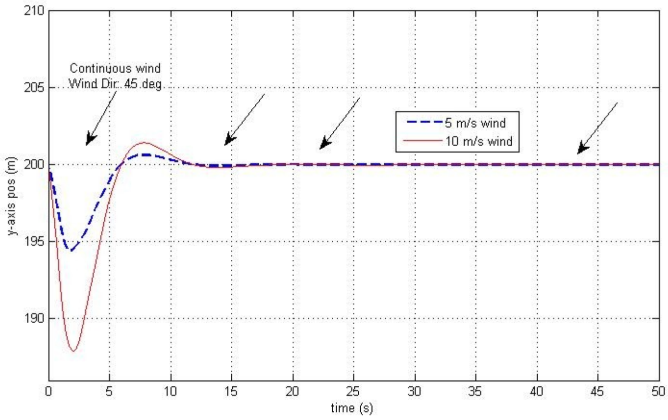

Figure 151 - Scn 3.1 y-Axis Position Plot for Controller 4

Controller 3 is the fastest among other controllers in settling time, but Controller 4's initial response is very good.

- **Scenario 3.2: Windspeed = 5, 10, Wind Direction = 135°**

This time, continuous wind is applied from south-east and the performances of the controllers are measured.

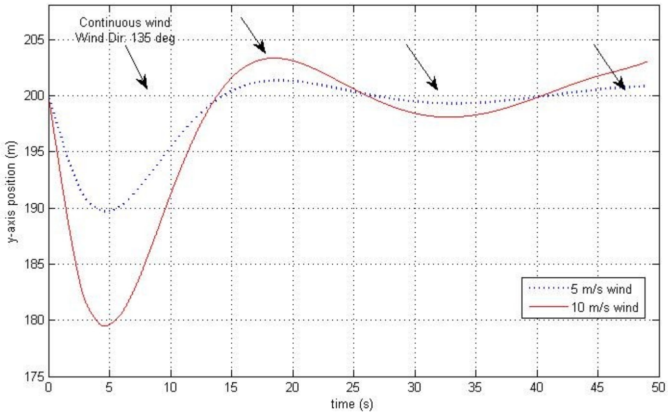

Figure 152 - Scn 3.2 y-Axis Position Plot for Controller 1 with LTG

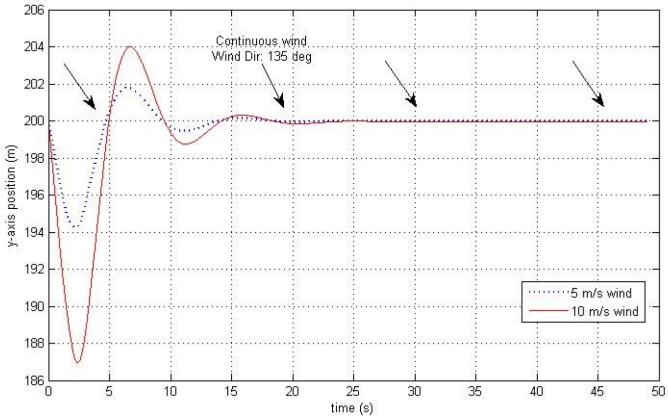

Figure 153 - Scn 3.2 y-Axis Position Plot for Controller 1 with CTG

It can be observed from the figures that Controller 1 with LTG is not adequate for winds coming from southeast. The result would be the same if the wind was coming from south-west. It is seen that the speed of the controller/guidance is slow regarding Controller 1 with CTG, hence not suitable for landing under such winds.

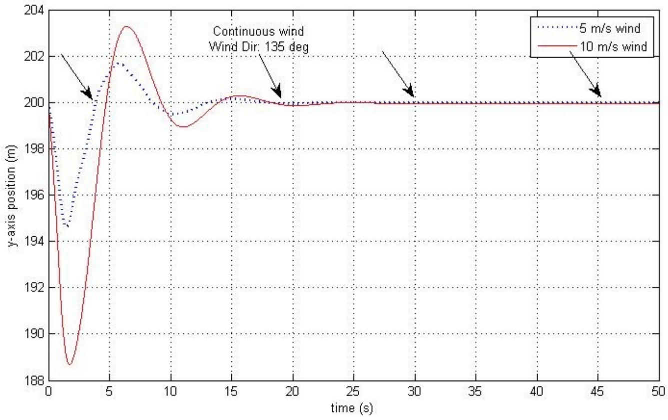

Figure 154 - Scn 3.2 y-Axis Position Plot for Controller 2

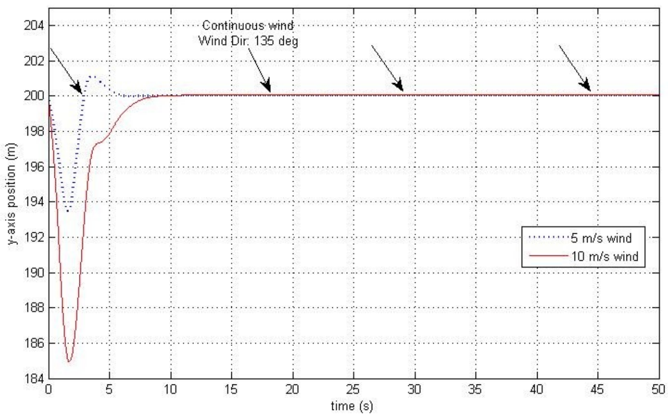

Figure 155 - Scn 3.2 y-Axis Position Plot for Controller 3

Controller 2 is reacting faster than Controller 3 against the disturbance initially; however, overshoot is observed in Controller 2 results.

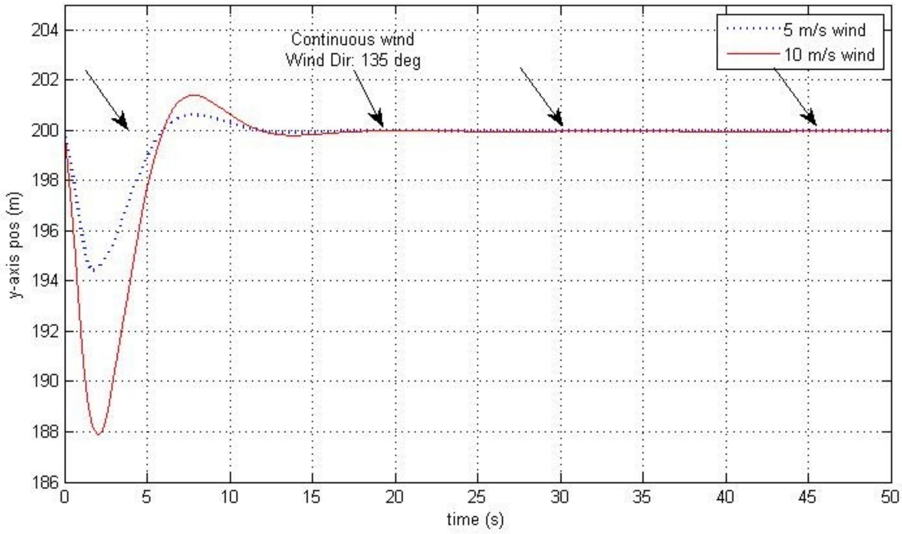

Figure 156 - Scn 3.2 y-Axis Position Plot for Controller 4

Controller 4 results are very similar to Controller 3 (linear case) results, but with lower overshoot.

5.1.4 Scenario 4: Continuous Wind From Above After Flare Point

In this scenario, wind coming from above at flare point is simulated, and controllers' performances are analyzed. 2 simulations are run with 5 m/s and 10 m/s wind amplitudes, respectively. Altitude and airspeed are important parameters and will be observed for this simulation.

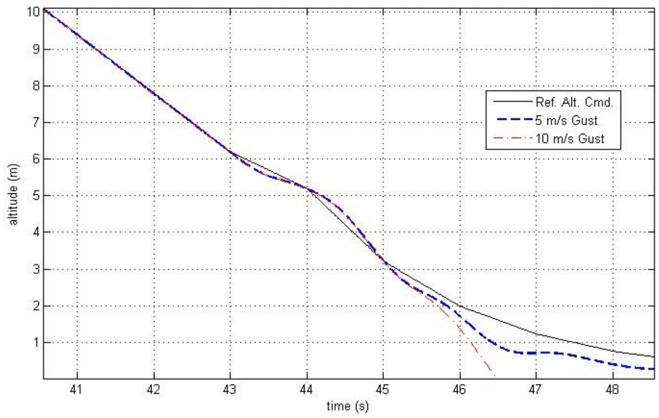

Figure 157 - Scn 4 Altitude Plot for Controller 1

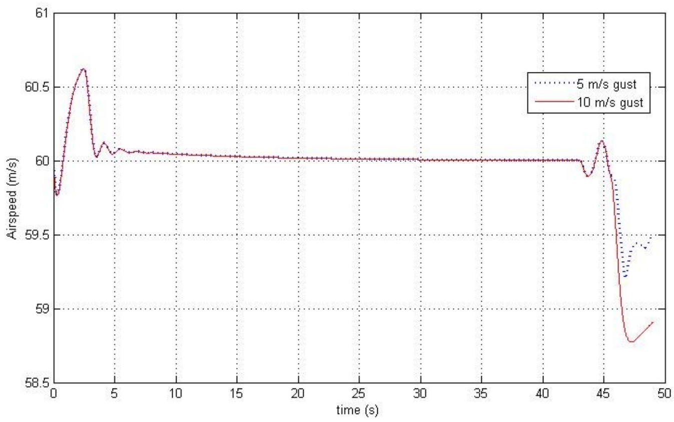

Figure 158 - Scn 4 Airspeed Plot for Controller 1

A significant drop in altitude is observed under 10 m/s wind, and it yields a harsh impact on ground. The result of such an impact will be a big jump on ground, which is undesired in landing. Some decrease on airspeed is also observed, but it is very small, hence not affecting the landing.

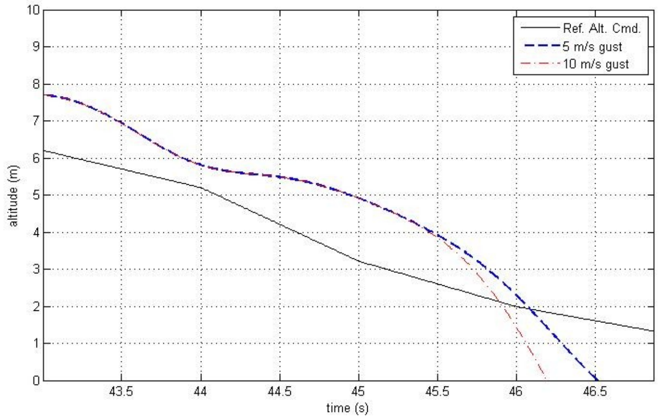

Figure 159 - Scn 4 Altitude Plot for Controller 2

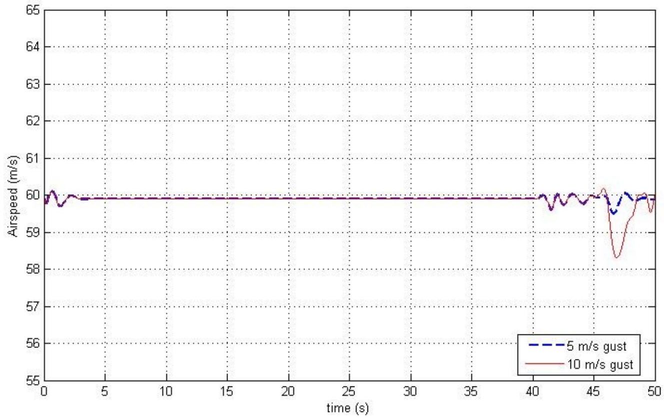

Figure 160 - Scn 4 Airspeed Plot for Controller 2

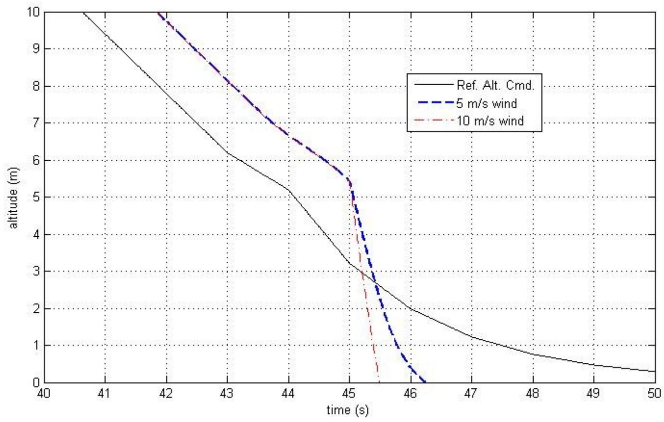

Figure 161 - Scn 4 Altitude Plot for Controller 3

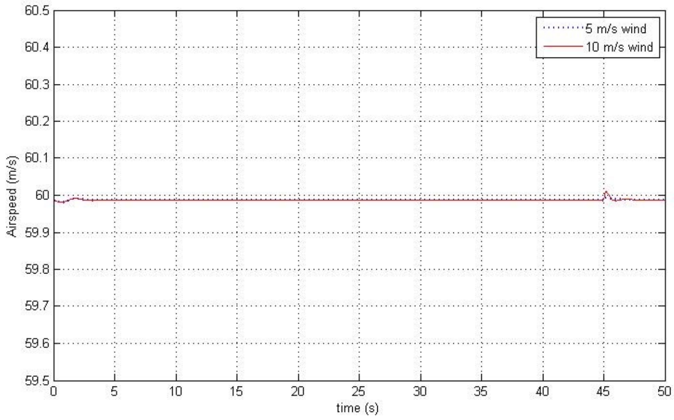

Figure 162 - Scn 4 Airspeed Plot for Controller 3

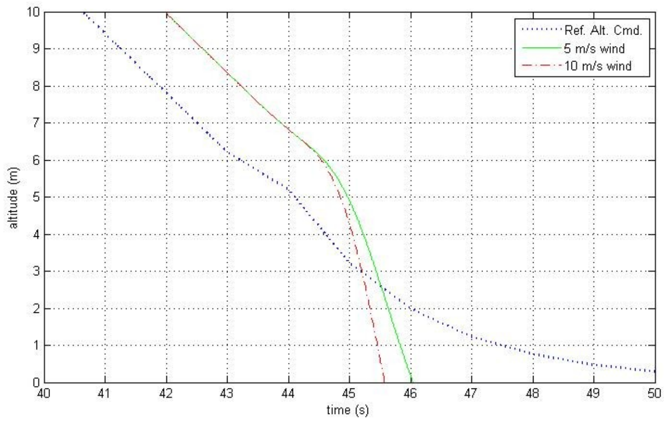

Figure 163 - Scn 4 Altitude Plot for Controller 4

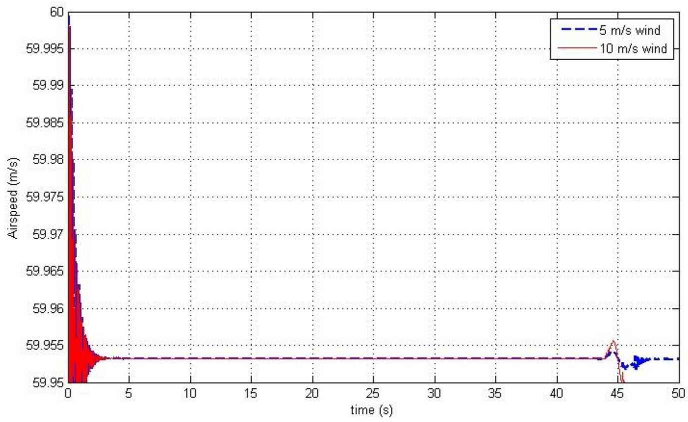

Figure 164 - Scn 4 Airspeed Plot for Controller 4

Since the wind starts just at the beginning of the flare maneuver, the aircraft's alpha and gamma angles are negative. When the wind contacts the aircraft, a nose-down behavior is observed, and this results in a collision of the aircraft's nose with the ground. These results show that the Controller 2, 3 and 4 are not adequate for this scenario. However, as the wind starting time is moved backward (away from flare point) these controllers are expected to follow the trajectory after a loss of altitude.

5.1.5 Scenario 5: Wind Gust From Above at Flare Point

In this scenario, controllers' performance will be analyzed under a wind gust with duration of 1 second just at the flare point. Again, airspeed and altitude are the parameters to be observed.

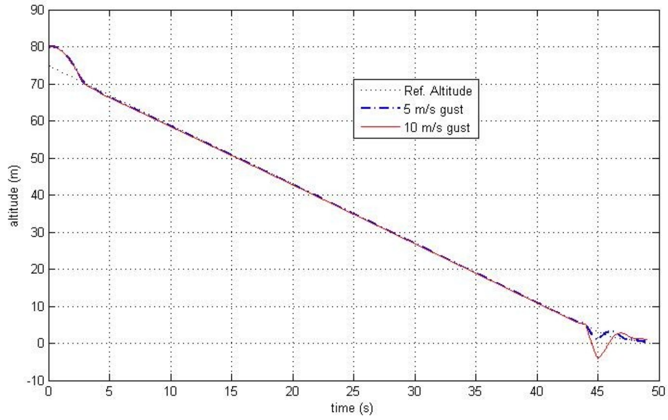

Figure 165 - Scn 5 Altitude Plot for Controller 1

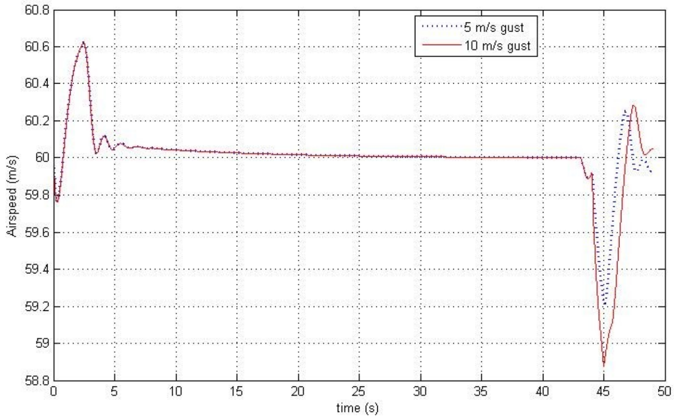

Figure 166 - Scn 5 Airspeed Plot for Controller 1

At 10 m/s gust the aircraft collides with the ground, because the controller's response time is not fast enough to tolerate the gust.

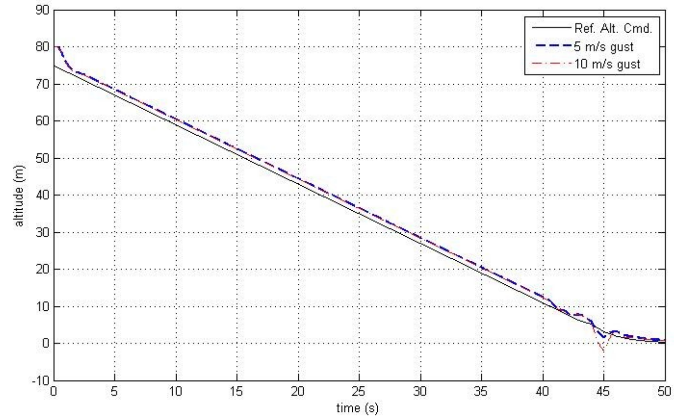

Figure 167 - Scn 5 Altitude Plot for Controller 2

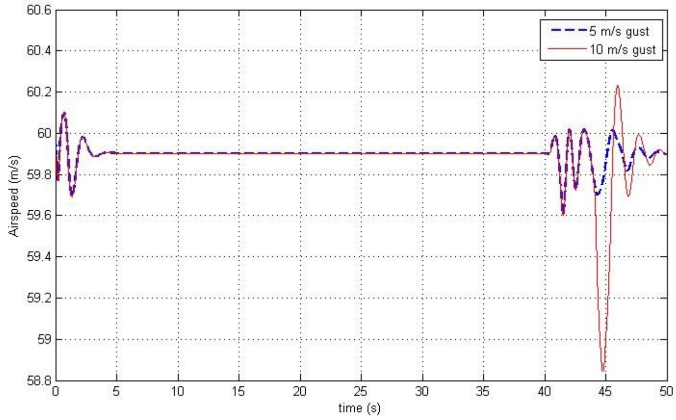

Figure 168 - Scn 5 Airspeed Plot for Controller 2

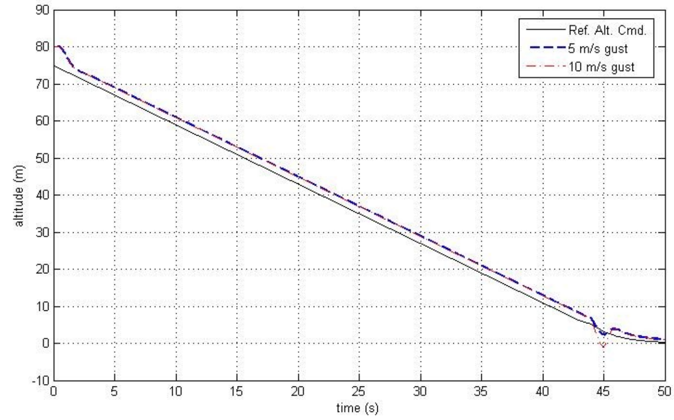

Figure 169 - Scn 5 Altitude Plot for Controller 3

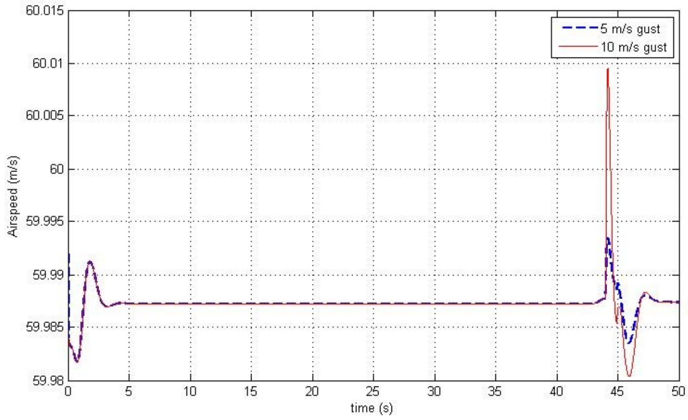

Figure 170 - Scn 5 Airspeed Plot for Controller 3

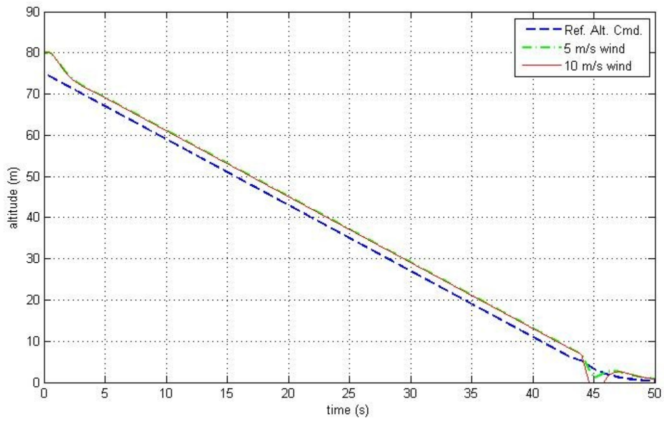

Figure 171 - Scn 5 Altitude Plot for Controller 4

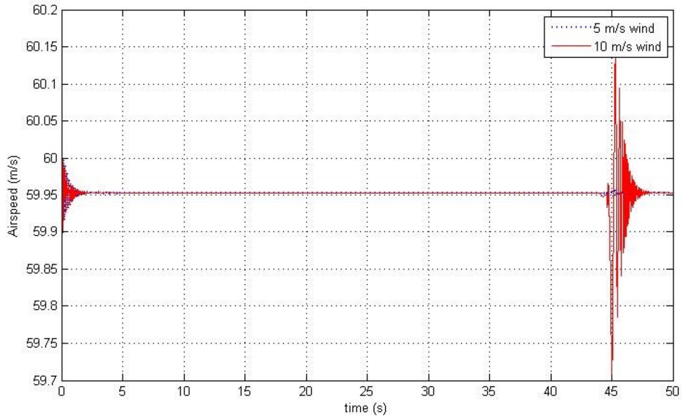

Figure 172 - Scn 5 Airspeed Plot for Controller 4

It can be easily stated that 10 m/s wind from above is higher than the controllers can tolerate. Under 5 m/s wind gust all of the controllers managed to keep track after a little disturbance.

5.1.6 Scenario 6: 10 m/s Cross-wind + Turbulence

A light turbulence is applied together with a cross-wind. Von Karman Wind Turbulence Model in Matlab/Simulink Aerospace Blockset is used for this purpose, which implements the mathematical representation in the Military Specification MIL-F-8785C. It generates continuous wind turbulence with Von Kármán velocity spectra. Wind profile for this scenario is given in Figure 173.

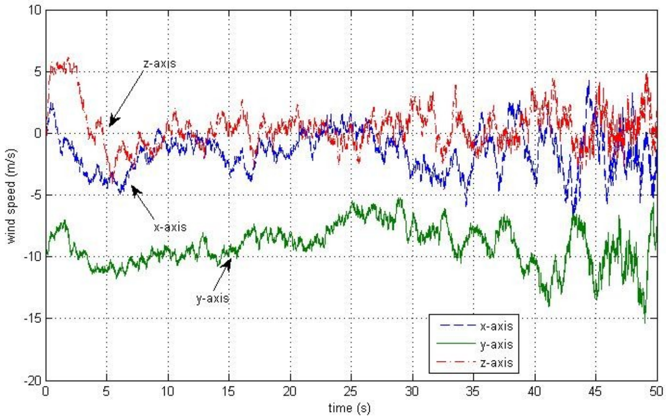

Figure 173 - Wind Profile for Scn.6

As the results are investigated for Controller 4, a small increase in airspeed is noticed resulting in an early touchdown than expected. Despite the early touchdown aircraft follows the altitude trajectory and maintains an acceptable airspeed.

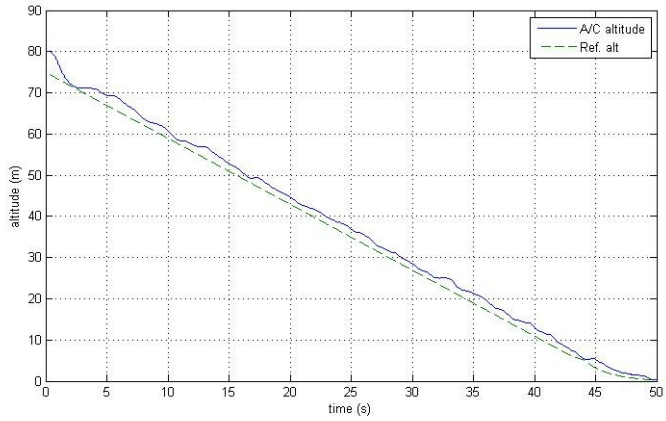

Figure 174 - Scn 6 Altitude Plot for Controller 4

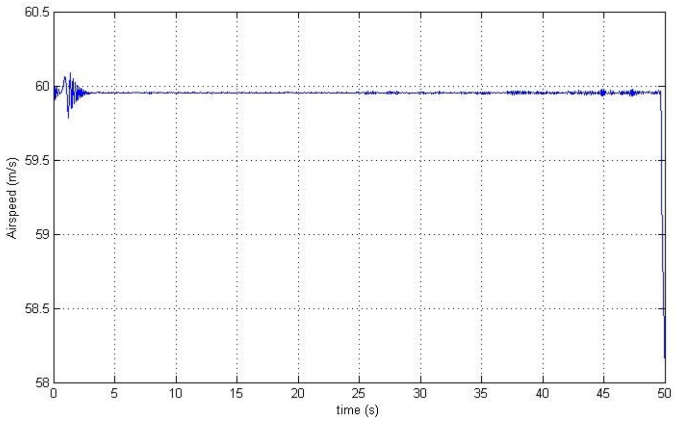

Figure 175 - Scn 6 Airspeed Plot for Controller 4

Oscillations are observed on airspeed of the aircraft, this is expected, as there exists turbulence. There is a dramatic decrease in aircraft airspeed at the end of the simulation, the reason for that is the contact of the aircraft with the ground. This is due to the increase in airspeed.

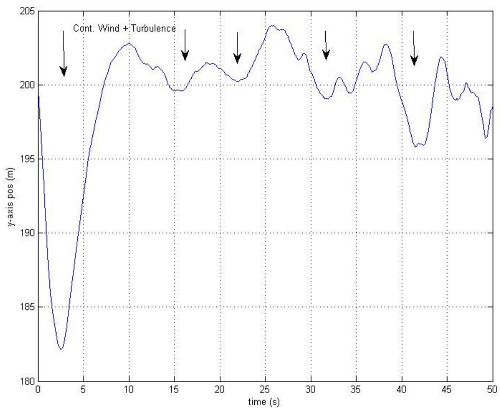

Figure 176 - Scn 6 y-Axis Position Plot for Controller 4

Control surface deflections and Euler angles of the aircraft during landing is as follows:

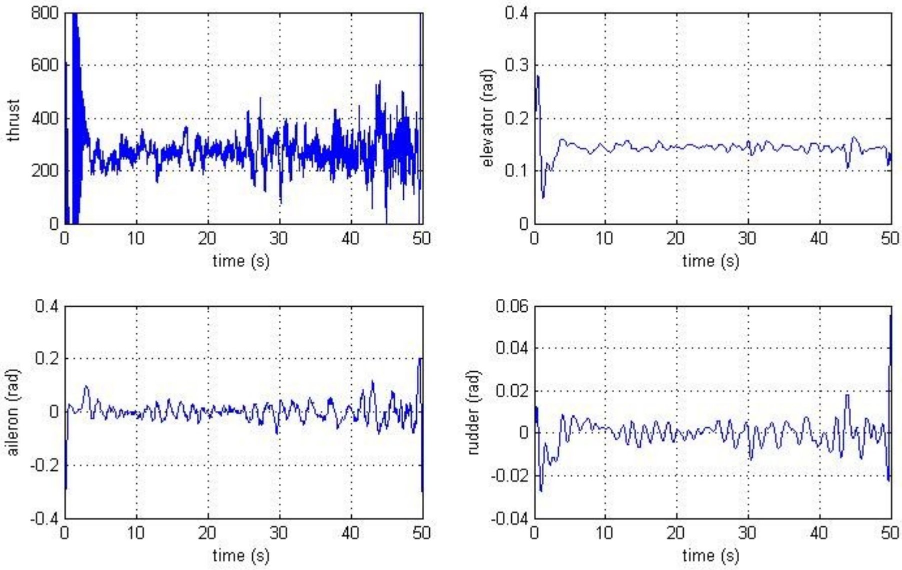

Figure 177 - Scn 6 Control Surface Deflections for Controller 4

Important thing to be noticed here is that control surfaces are not saturated under turbulence. Thrust and aileron are the most frequently used controls during landing under turbulence. This is an expected outcome, as the most effected parameters are the airspeed and the y-axis position of the aircraft. Oscillations are also observed on control surfaces; however, the frequency of the oscillations are acceptable.

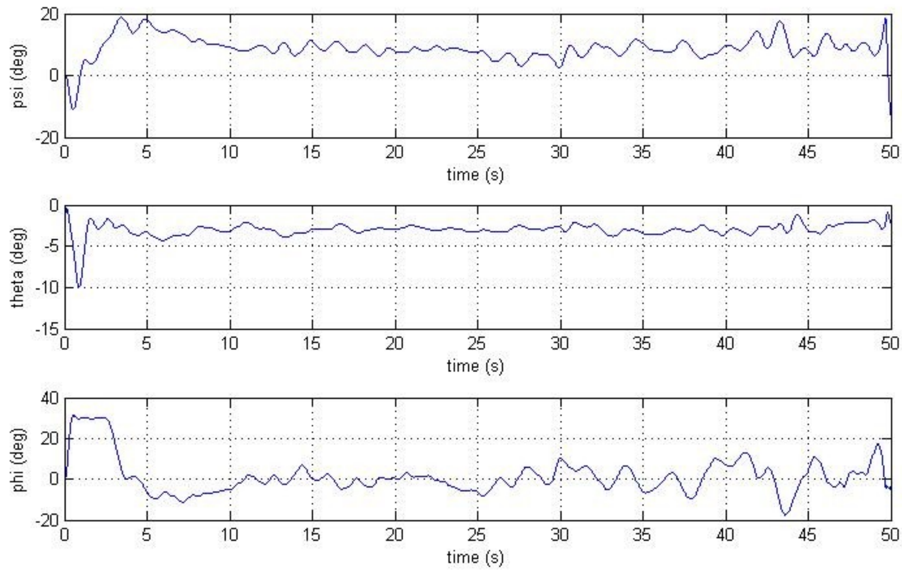

Figure 178 - Scn 6 Euler Angles for Controller 4

The most effected response of the aircraft is the y-axis position since the wind mostly comes from the side. However, the aircraft manages to be in the predefined range at the instance of the touchdown. The descent rate of the aircraft is about -0.2 deg/s, which is also an important parameter for a safe landing. Pitch angle of the aircraft is very close to zero, but the yaw angle is not zero, which means a decrab maneuver is needed before touchdown. A small roll angle exists just before touchdown, that can be a problem for the landing. For this reason, landing gears’ ground contact times are analyzed to assure safe landing.

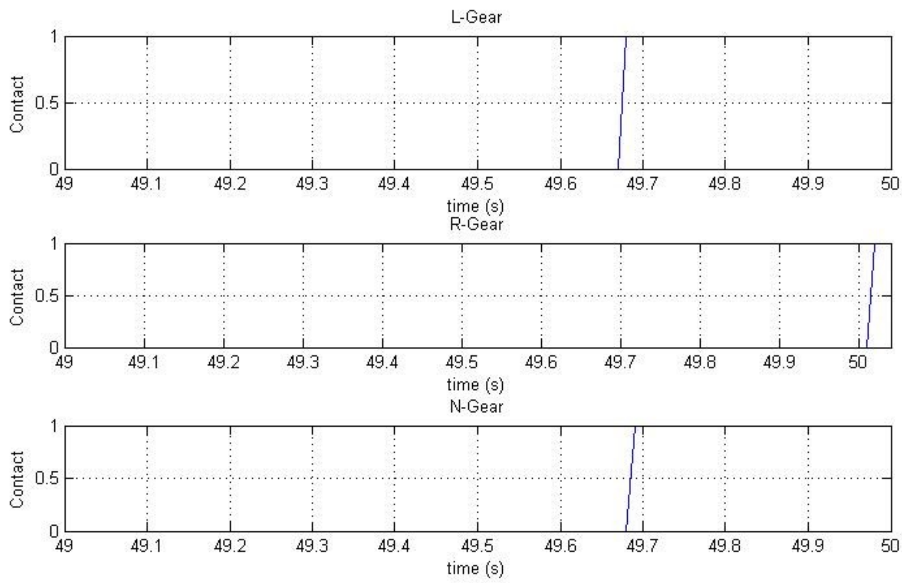

Figure 179 - Scn 6 Landing Gears' Contact Times for Controller 4

In Figure 179, '0' means no contact, and '1' means ground contact. It is seen that the right landing gear contacts the ground 0.3 seconds later than left. This difference is very small and can be neglected.

Same scenario is applied on Controller 2. Again, oscillations are observed in airspeed response as expected. However, early landing as in Controller 4 is not encountered for Controller 2. This means that the Controller 2 keeps the airspeed closer to the desired values. Y-Axis response is very similar, although Controller 2 could not keep the aircraft from crossing the y-axis position of 205. This does not affect the behavior of the aircraft at touchdown. Controller 2 altitude tracking performance is superior to Controller 4. Very small oscillations are observed and the aircraft is landed safely.

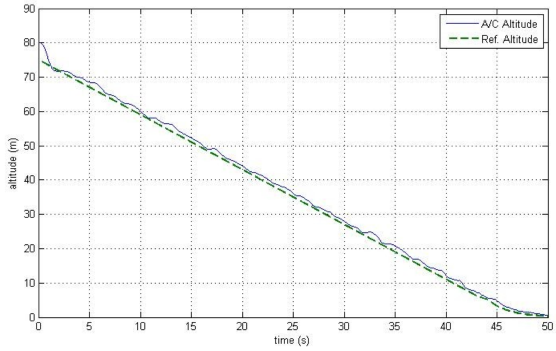

Figure 180 - Scn 6 Altitude Plot for Controller 2

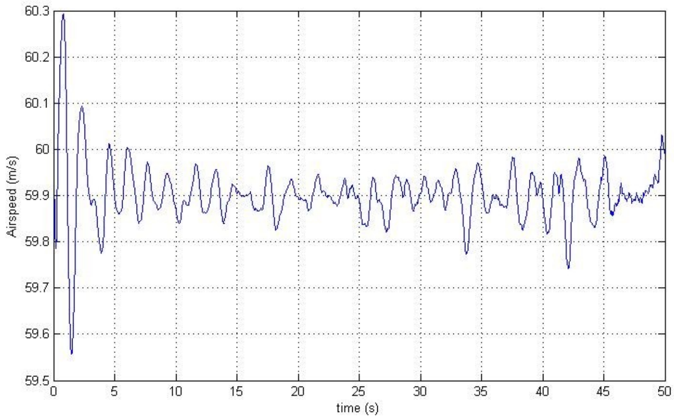

Figure 181 - Scn 6 Airspeed Plot for Controller 2

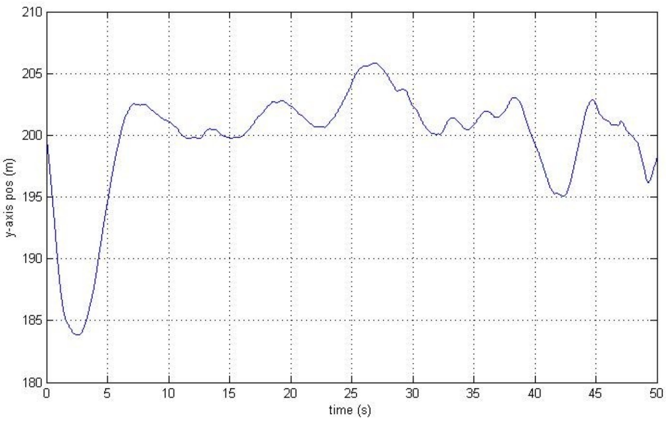

Figure 182 - Scn 6 y-Axis Position Plot for Controller 2

Control surface deflections and Euler angles of the aircraft during landing is as follows:

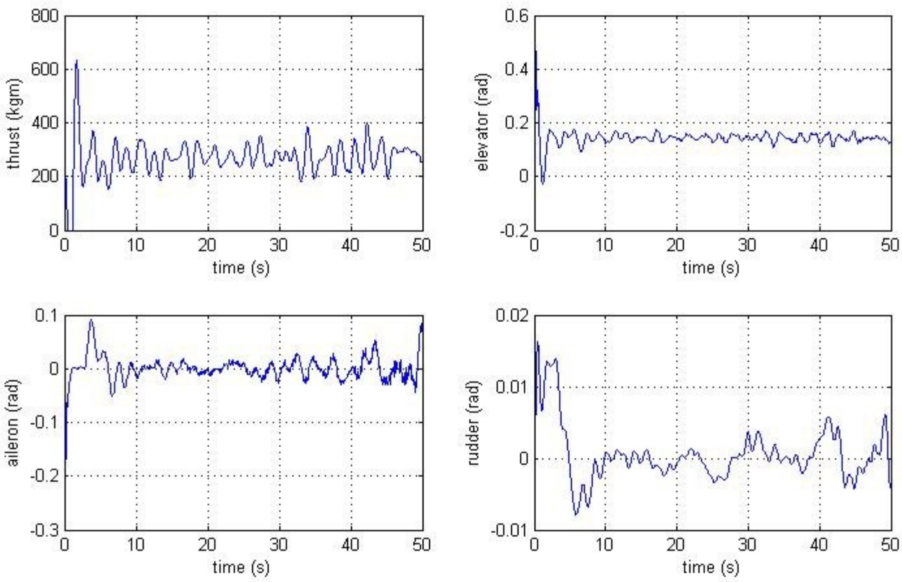

Figure 183 - Scn 6 Control Surface Deflections for Controller 2

Behavior of the control surfaces when using Controller 2 is very similar in terms of usage frequency of controls. However, there is also a performance improvement in deflection amounts. It is observed that Controller 2 uses less control input to keep the aircraft in track. This is a preferable situation in terms of control.

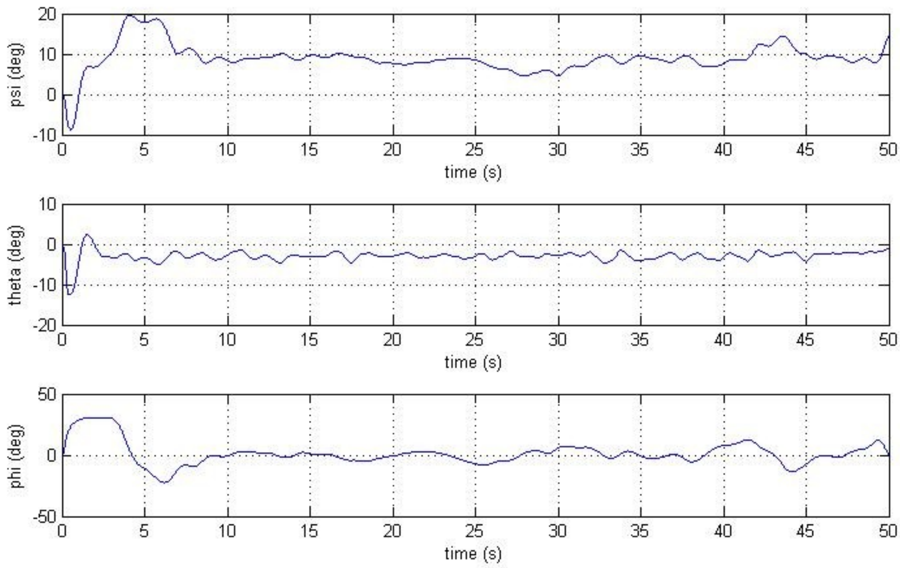

Figure 184 - Scn 6 Euler Angles for Controller 2

Euler angle states are almost the same with the results of Controller 4. Figure 185 shows that there is a 0.3 seconds difference between contact times of left and right gears, indicating a safe landing.

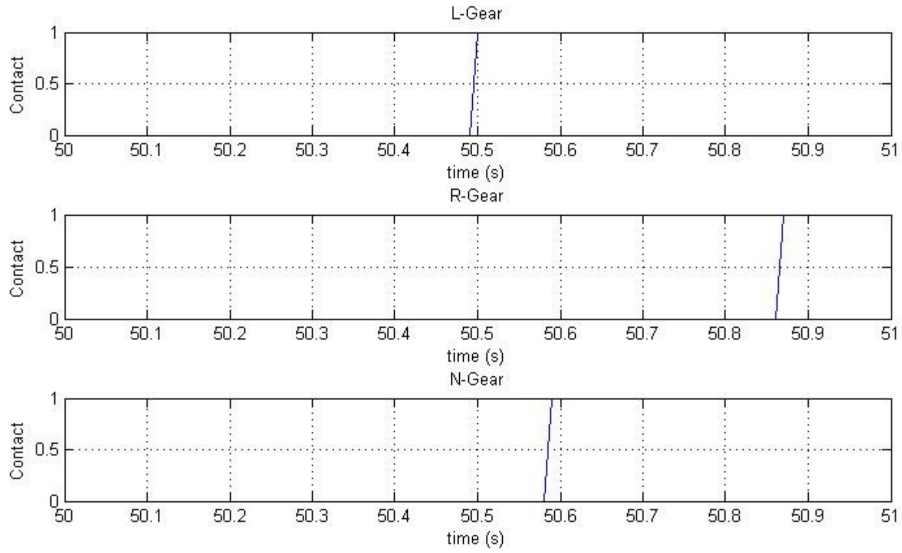

Figure 185 - Scn 6 Landing Gears' Contact Times for Controller 2

Figure 186 is the comparison for y-axis position of the aircraft with different controllers under turbulence. It is hard to say that one of them is superior to the other.

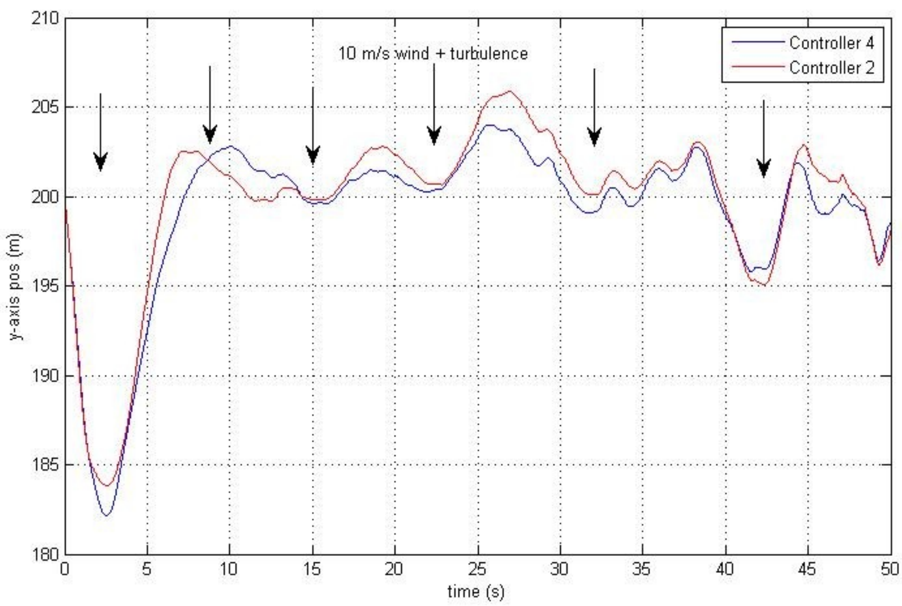

Figure 186 - Scn 6 y-Axis Position Plot for All Controllers

As stated before, the emergency coordinator is not active during the final simulations to observe the complete behavior of the aircraft. The final portion of the corridor is the narrowest of all with 13 m. width and 8 m. height. Simulation results show that the aircraft does not leave the landing cone from above or below. Even under wind coming from above the aircraft's altitude does not change more than two or three meters. This proves that the safe landing can be achieved in terms of longitudinal axes. The situation is different in lateral case. During the crosswind scenarios, it is observed that the aircraft slips in y-axis up to a distance of 30 meters (for PID controller) under 10 m/s crosswinds. This is the highest distance, and the value is lower for other controllers. 30 m. of slip is an acceptable value for most parts of the landing cone; however, as the distance to ground is smaller the cone gets narrower. Especially it is seen that in the areas closer to the final window of the corridor, the aircraft can get out of the cone under emerging strong crosswinds such as 10 m/s with any controller. Under 5 m/s crosswinds, the LQT and SMC performances are almost enough to keep the aircraft in cone near the final window. The landing corridor is designed to define a zone, which is safe for landing in all conditions. Simulation results show that, even under wind gust after final window, LQT and SMC manages to keep the aircraft on track before touchdown. However, landing an aircraft under these circumstances is a risk in real life applications.

CHAPTER 6

CONCLUSION

6.1 Summary and Conclusions

In this study, several autopilot and guidance algorithms are developed for a nonlinear, unmanned aerial vehicle Pioneer. First, a nonlinear aircraft model is obtained. Classical and modern control theory based methods are selected in designing the autopilot. Linearization procedures are applied in order to obtain a linear model, where most of the autopilot design process takes place. To be baseline or reference to other methods, a classical PID controller is designed for different trim points of the aircraft. Then, linear controllers are combined using the Gain Scheduling technique. Finally, the controller is tuned again to run with a nonlinear model. Among several robust control based methods, sliding mode control and linear quadratic tracker are selected for designing a nonlinear aircraft autopilot. The reasons behind this selection are their robust and invariant nature in control of nonlinear systems. One of the most important properties of LQ regulator is that provided certain conditions are met, it guarantees nominally stable closed loop systems. The sliding mode controller can be applied directly to nonlinear model or can be designed over a linear model then transferred to nonlinear model. Both controller types are designed and performances are compared to each other. For the nonlinear case, in order to obtain the form given in (3.31), a nonlinear Pioneer dynamics is generated using Maple Software. The code generated by Maple is then exported to Simulink. As expected sliding mode controller's performance is superior against PID controller, especially under disturbance. Slight difference is observed between linear and nonlinear case of sliding mode controllers and linear quadratic tracker. One of the most important advantages of SMC and LQT against the other classical controllers is speed, which is also one of the most wanted abilities from a controller during landing. However, it is hard to say whether SMC and LQT are superior to each other for the simulations run in this thesis. There are some differences observed in initial response to a disturbance and settling time of controllers, and somewhat two parameters are related to each other.

A guidance system is added to the model in order to guide the aircraft using waypoints, under disturbance, or in an emergency. Using an optimal guidance method, an optimal altitude trajectory for landing is generated for the aircraft. A cost function using altitude error is defined and it is minimized yielding the optimal input. Borders for the landing trajectory is defined and optimization algorithm is run for different initial points in that border. Then, for lateral guidance under disturbance, a lateral track control law and a cross track controller are designed. They are simulated separately and their performances are compared to each other. Lateral track control law is also used in waypoint navigation simulations. Cross track guidance is fast compared to lateral track control law; however, some overshoots are observed in cross track guidance in some simulations. This means that cross track guidance is more preferable at close proximity to touchdown point under crosswind and lateral track control law can be preferred for waypoint navigation for its smooth guidance.

An emergency flight plan is generated in order to cancel landing and return to flight in an emergency such as the aircraft being out of the predefined landing cone. This means that aircraft cannot accomplish a safe landing. In such a situation, according to emergency plan, the aircraft autonomously flies to a predefined altitude and completes a tour around the runway to enter the cone again in a safe way. Rules for mode transition are strictly defined such that no undefined areas left. Undefined areas may result in wrong mode decisions and undesirable results. Several emergency scenarios are generated and mode transitions observed. The results indicate no error in mode transition, and aircraft is safely landed in all emergency scenarios.

Different scenarios are generated in order to test and compare the controllers' performance under wind disturbance. Different wind amplitudes from different directions are applied to the aircraft. It is observed that in all scenarios before the response of the controller the aircraft is pulled out of track by disturbance, then the controller dominates and keeps the aircraft on track. All of the controllers manage to keep the aircraft on track successfully under 10 m/s crosswind after some "settle time". The "settle time" is different for all controllers, making some of the controllers unsuitable for certain phases of landing. Sliding mode controller (nonlinear case) has the best performance at the final phases of landing because of its speed. Guidance method also affects the speed of response. Cross track controller method has a fast response with respect to lateral track controller.

In summary, most important aim of this thesis is to control an aircraft's motion during landing phase of flight under disturbance. In order to realize this objective, using a suitable nonlinear aircraft model, several control and guidance methods are developed. Simulations demonstrated that all controllers and guidance systems are able to land and navigate the aircraft under adverse weather conditions.

6.2 Future Work

In this study, sensor noise and delay, maneuvers before landing (decrab, deroll, etc.), and control after landing and are not taken into consideration. Future studies will surely include these methods to increase the fidelity of the system. Additional future studies may also include:

- Higher-order Sliding Mode Control,
- MIMO approach for LQ and Sliding Mode Controllers
- Adaptive Sliding Mode Control in order to increase the system tolerance,
- Autonomous takeoff,
- Stability and robustness analysis of the system.

REFERENCES

[1] Kargın V., "Design of Autonomous Landing Control Algorithm for a Fixed Wing UAV," M.S. thesis, Aerospace Eng. Dept., Middle East Technical University, 2007.

[2] Cho A. et al., "Fully Automatic Taxiing, Takeoff and Landing of a VAV Using a Single-Antenna GPS Receiver Only," International Conference on Control, Automation and Systems, pp. 821-825, 2007.

[3] Attar, M., Wahnon, E., Chaimovitz, D., "Advanced Flight Control Technologies for UAVs," 2nd AIAA "Unmanned Unlimited" Systems, Technologies and Operations, 2003.

[4] Juang, J.G., Chiou, H.K., Chien L.H., "Analysis and Comparison of Aircraft Landing Control Using Recurrent Neural Networks and Genetic Algorithms Approaches," Elsevier B.V, 2008.

[5] Juang, J.G., Lin, B.S., Chin, K.C., "Intelligent Fuzzy Systems for Aircraft Landing Control," Springer-Verlag, Berlin, 2005.

[6] Liguni, Y., Aknoshi, H., Adachi, N., "An Intelligent Landing System Based on 21 Human Skill Model," IEEE Transactions on Aerospace and Electronic Systems, vol. 34, no. 3, pp. 877-882, 1998.

[7] Rong, H.J., Sundararajan, N., Saratchandran, P., Huang, G.B., "Adaptive Fuzzy Fault-Tolerant Controller for Aircraft Autolanding Under Failures," IEEE Transactions on Aerospace and Electronic Systems, vol. 43, no. 4, pp. 1586-1603 2007.

[8] "AAI_RQ-2_Pioneer." Internet: http://en.wikipedia.org/wiki/AAI_RQ-2_Pioneer, [March 9, 2011].

[9] Christiansen, R.S., "Design of an Autopilot for Small Unmanned Aerial Vehicles," M.S. thesis, Electrical and Computer Engineering, Brigham Young University, 2004.

[10] "Aircraft-uiuc." Internet: http://www.ae.illinois.edu/m-selig/apasim/Aircraft-uiuc.html, [March 9, 2011].

[11] Rauw, M., "FDC 1.2 - A Simulink Toolbox for Flight Dynamics and Control Analysis," Delft University of Technology, 2nd Edition, 2001.

[12] Evcimen, Ç., “Development and Comparison of Autopilot and Guidance Algorithms for Missiles,” M.S. thesis, Electrical and Electronics Eng. Dept., Middle East Technical University, 2007.
[13] Visioli, A., “Practical PID Control,” IEEE Transactions on Automatic Control, vol. 53, no. 9, pp. 2217-2218, 2006.
[14] Aström, K., Hagglund, T., “PID Controllers: Theory, Design, and Tuning,” 2nd Edition, 1995.
[15] McLean, D., “Automatic Flight Control Systems”, Prentice Hall Inc., 1990.
[16] Becerra, V.M., “Optimal Control Lecture Notes: Lecture 4: LQ Regulators and Tracking Optimal Control,” School of Systems Engineering, 2009.
[17] Wells, S.R., “Application of Sliding Mode Methods to the Design of Reconfigurable Flight Control Systems,” Ph.D. dissertation, Mechanical and Aeronautical Eng. Dept., University of California, 2002.
[18] Utkin, V.I., “Sliding Modes in Control and Optimization,” Springer-Verlag, 1992.
[19] DeCarlo, R.A., Zak, S.H. and Matthews, G.P., “Variable Structure Control of Nonlinear Multivariable Systems: A Tutorial,” Proc. IEEE, vol. 76, no. 3, pp. 212-232, 1988.
[20] Yavrucuk, I., Kargin, V., “Autolanding Controller Strategies for Fixed Wing UAV in Adverse Athmospheric Conditions,” American Institute of Aeronautics and Astronautics, Guidance, Navigation and Control Conference, Honolulu, HI, USA, August 2008.
[21] Zak, S.H., “Systems and Control,” Oxford University Press, USA, 2003.
[22] Sabanovic, A., Fridman, L.M. and Spurgeon, S., “Variable Structure Systems from Principles to Implementation,” IET Control Engineering Series 66, 2004.
[23] Luenberger, D.G., “Linear and Nonlinear Programming,” Addison Wesley, Second Edition, 1984.
[24] Meza, J.C., “Steepest Descent”, Wiley Interdisciplinary Reviews: Computational Statistics, vol. 2, no 6, pp. 719-722, 2010.
[25] Niculescu, M., “Lateral Track Control Law for Aerosonde UAV”, AIAA 2001-0016, 2001.
[26] “WindTriangle.”
Internet:http://www.delphiforfun.org/programs/math_topics/WindTriangle.htm,
[April 12, 2011].

[27] Edwards, C., Spurgeon, S.K., “Sliding Mode Control: Theory and Applications”, Copyright Taylor & Francis Ltd, 1998.
[28] Gao, W.B. and Hung, J.C., “Variable Structure Control of Nonlinear Systems: A New Approach”, IEEE Trans. Industrial Electronics, vol. 40, no. 1, pp.45-55, 1993.
[29] “Maple_(software).” Internet: http://en.wikipedia.org/wiki/Maple_(software), [June 8, 2011].
[30] Hanköylü, M., “Landing Autopilot Design for an UAV”, M.S. thesis, Electrical and Electronics Eng. Dept., Middle East Technical University, 2011.
[31] Utkin, V.I., “Variable Structure Systems with Sliding Modes”, IEEE Trans. Automat. Cont., vol. AC-22, no. 2, pp. 212-222, 1977.
[32] Utkin, V.I., “Sliding Modes and Their Application in Variable Structure Systems”, Moscow, Soviet Union: MIR Publishers, 1978.