ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

DEVELOPMENT OF A MODEL UNMANNED AERIAL VEHICLE WITH SIMULINK : MODELLING AND CONTROL

M.Sc. THESIS

Akçay ÇALIŞIR

Department of Mechatronics Engineering

Mechatronics Engineering Programme

MAY 2015

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

DEVELOPMENT OF A MODEL UNMANNED AERIAL VEHICLE WITH SIMULINK : MODELLING AND CONTROL

M.Sc. THESIS

Akçay ÇALIŞIR
(518101045)

Department of Mechatronics Engineering

Mechatronics Engineering Programme

Thesis Advisor : Assist.Prof. Dr. Gülay ÖKE GÜNEL

20 MAY 2015

<u>**İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ**</u>

SİMULİNK İLE MODEL İNSANSIZ HAVA ARACI GELİŞTİRİLMESİ : MODELLEMESİ VE KONTROLÜ

YÜKSEK LİSANS TEZİ

Akçay ÇALIŞIR
(518101045)

Mekatronik Mühendisliği Bölümü

Mekatronik Mühendisliği Programı

Tez Danışmanı: Yrd.Doç. Dr. Gülay ÖKE GÜNEL

20 MAYIS 2015

Student ID 518101045, a M.Sc. student of ITU Institute of Science and Technology, Akçay ÇALIŞIR, successfully defended the thesis entitled ‘‘DEVELOPMENT OF A MODEL UNMANNED AERIAL VEHICLE WITH SIMULINK : MODELLING AND CONTROL’’, which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Assist. Prof. Dr. Gülay ÖKE GÜNEL** Istanbul Technical University	..……………...
Jury Members :	**Assoc. Prof. Dr. Pınar BOYRAZ** Istanbul Technical University	………………...
	Assist. Prof. Dr. Figen ÖZEN Haliç University	………………...

Date of Submission : 24 April 2015

Date of Defense : 20 May 2015

To my beloved family,

FOREWORD

I would like to express my special thanks to my thesis advisor Assist. Prof. Dr. Gülay ÖKE GÜNEL for her valuable guidance and sharing her knowledge through every step of thesis which describing unmanned aerial vehicle.

Finally, for their valuable support in every step of my life, I am grateful to my mother, my father, my sister and the rest of my family.

May 2015 Akçay ÇALIŞIR

TABLE OF CONTENTS

ABBREVIATIONS

UAV	: Unmanned Aerial Vehicle
ANFIS	: Adaptive Neuro Fuzzy Inference System
DCM	: Direction Cosine Matrix
GA	: Genetic Algorithm
NASA	: National Aeronatics and Space Administration
PID	: Proportional Integral Derivative
MIMO	: Multi Input Multi Output
NN	: Neural Network
SISO	: Single Input Single Output
GD	: Gradient Descent
HLO	: Hybrid Learning Algorithm
FIS	: Fuzzy Inference System
FLC	: Fuzzy Logic Control
SF	: Scaling Factor
ITAE	: Integral of the Time Weighted Absolute Error
MF	: Membership Function
TS	: Takagi Sugeno

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

A	: System Matrix
A'	: Augmented system matrix
B	: Conrol Input Matrix
B'	: Augmented control input matrix
b	: Wing Span
C	: Output Matrix
c	: Wing Chord
$\bar{c}$	: Mean Aerodynamic Chord
C_r	: Wing Root Chord
C_D	: Aircraft Drag Coefficient
C_L	: Aircraft Lift Coefficient
C_l	: Aircraft rolling moment coefficient
C_m	: Aircraft pitching moment coefficient
C_n	: Aircraft yawing moment coefficient
C_T	: Thrust coefficient
C_x	: Aircraft axial force coefficient
C_y	: Aircraft vertical force coefficient
C_z	: Aircraft normal force coefficient
dm	: Aircraft mass element
ds	: Aircraft surface area element
$\frac{dv_w}{dt}$	: Body-fixed on y axes accelaration due to wind and turbulence
$\frac{dw_w}{dt}$	: Body-fixed on z axes accelaration due to wind and turbulence
D	: Relationship between control inputs and outputs
D_w	: Washout filter
e	: Error
Δe	: Changes of error
E_x	: Euler axis about x axis
E_y	: Euler axis about y axis
E_z	: Euler axis about z axis
f	: Nonlinear system function
$\vec{F}$	: Body-fixed total force
F_x	: Axial force
F_y	: Vertical force
F_z	: Normal force
$\vec{g}$	: Gravitational acceleration
h	: Nonlinear output function
H	: Angular Momentum
$\dot{h}$	: Rate of climb
$\vec{i},\vec{j},\vec{k}$	: Unit vectors along body-fixed axes
I_x	: Moment of inertia about the x axis

I_y	: Moment of inertia about the y axis
I_z	: Moment of inertia about the z axis
I_{xy}	: Moment of inertia about the xy axis
I_{xz}	: Moment of inertia about the xz axis
I_{yz}	: Moment of inertia about the yz axis
K	: Correction factor for pressure drag
K_q	: Feedback gain
K_d	: Value of Derivation
K_i	: Value of Integral
K_p	: Value of Proportional
L_A	: Rolling moment
L_P	: Rolling moment per unit roll rate
M_A	: Pitch moment
m	: Aircraft mass
M	: Mach number
M_α	: Pitch angular accelaration per unit rudder angle
M_{T_α}	: Pitch angular accelaration per unit angle of attack due to thrust
M_u	: Pitch angular accelaration per unit change in speed
M_{T_u}	: Pitch angular accelaration per unit change in speed due to thrust
$M_{\dot{\alpha}}$	: Pitch angular accelaration per unit rate of change angle of attack
M_q	: Pitch angular accelaration per unit change pitch rate
M_p	: Overshoot
M_w	: Pitch angular accelaration per unit change in speed
M_{δ_e}	: Pitch angular accelaration per unit elevator angle
N_A	: Yaw moment
N_β	: Yaw angular accelaration per unit sideslip angle
N_{T_β}	: Yaw angular accelaration per unit sideslip angle due to thrust
N_p	: Yaw angular accelaration per unit roll rate
N_r	: Yaw angular accelaration per unit yaw rate
N_{δ_a}	: Yaw angular accelaration per unit aileron deflecetion angle
N_{δ_r}	: Yaw angular accelaration per unit rudder deflecetion angle
p	: Roll rate
P_D	: Position in the down direction
P_E	: Position in the east direction
P_N	: Position in the north direction
q	: Pitch rate
$\overline{q}$	: Dynamic pressure
r	: Yaw rate
S	: Wing area
T	: Engine thrust magnitude
T_p	: Peak Time
T_r	: Rise Time
T_s	: Settling Time
u	: Forward velocity

$\mathbf{v}$	: Transverse velocity
$\mathbf{V}$	: Speed
$\mathbf{w}$	: Vertical velocity
$\boldsymbol{X_a}$	: Forward acceleration per unit angle of attack
$\boldsymbol{X_E}$	: Earth-fixed x-position
$\boldsymbol{X_u}$	: Forward acceleration per unit change in speed
$\boldsymbol{X_q}$	: Forward acceleration per unit change in pitch rate
$\boldsymbol{X_w}$	: Forward accelaration per unit change in speed
$\mathbf{X_{T_u}}$	: Forward acceleration per unit change in speed due to trust
$\mathbf{X_{\delta_e}}$	: Forward acceleration per unit elevator deflection angle
$\boldsymbol{Y_\beta}$	: Lateral accelaration per unit sideslip angle
$\boldsymbol{Y_E}$	: Earth-fixed y-position
$\boldsymbol{Y_P}$	: Lateral accelaration per unit roll rate
$\boldsymbol{Y_r}$	: Lateral acceleration per unit yaw rate
$\mathbf{Y_{\delta_a}}$	: Lateral accelaration per unit aileron deflection angle
$\mathbf{Y_{\delta_r}}$	: Lateral accelaration per unit rudder deflection angle
$\boldsymbol{Z_\alpha}$	: Vertical accelaration per unit angle of attack
$\boldsymbol{Z_E}$	: Earth-fixed z-position
$\boldsymbol{Z_u}$	: Vertical accelaration per unit change in speed
$\boldsymbol{Z_w}$	: Vertical accelaration per unit change in speed
$\boldsymbol{Z_{\dot{\alpha}}}$	: Vertical accelaration unit per unit rate of change of angle of attack
$\boldsymbol{Z_q}$	: Vertical accelaration per unit pitch rate
$\mathbf{Z_{T_u}}$	: Vertical accelaration per unit change in speed due to trust
$\mathbf{Z_{\delta_e}}$	: Vertical accelaration per unit elevator deflection angle
$\mathbf{0}$	: Trim and zero angle of attack

GREEK

$\boldsymbol{\alpha}$	: Angle of attack
$\boldsymbol{\dot{\alpha}}$	: Rate of change of angle of attack
$\boldsymbol{\beta}$	: Angle of sideslip
$\boldsymbol{\delta}$	: Relative pressure ratio at the flight altitude
$\boldsymbol{\delta_a}$	: Aileron deflection
$\boldsymbol{\delta_e}$	: Elevator deflection
$\boldsymbol{\delta_r}$	: Rudder deflection
$\boldsymbol{\delta_t}$	: Throttle deflection
$\boldsymbol{\zeta}$	: Damping ratio
$\boldsymbol{\Delta}$	: Increment of a parameter
$\boldsymbol{\Phi}$	: Roll angle
$\boldsymbol{\Theta}$	: Pitch angle
$\boldsymbol{\psi}$	: Yaw angle
$\boldsymbol{\lambda}$	: Eigenvalue
$\boldsymbol{\rho}$	: Air density
$\boldsymbol{\omega_n}$	: Natural frequency
$\boldsymbol{\Pi}$	: pi

DEVELOPMENT OF A MODEL UNMANNED AERIAL VEHICLE WITH SIMULINK : MODELLING AND CONTROL

SUMMARY

The nonlinear simulation model of an unmanned aerial vehicle (UAV) for Piper J3-Cub in MATLAB/Simulink® environment is developed by taking into consideration all the possible major system components such as actuators, gravity, engine, atmosphere, wind-turbulence models, as well as the aerodynamics components in the 6 DOF equations of motion.

Trim which is determined at the specific value and linearization is derived from the nonliner model at trim point are accomplished and various related analyses are carried out. The model is validated by comparing with a similar UAV data in terms of open loop dynamic stability characteristics. Using two main approaches; namely, fuzzy based on PID which is the obtained value using genetic algorithm and adaptive neural fuzzy inference system , linear controllers are designed.

The controllers are designed for control of roll, heading with coordinated turn, flight path, attitude command in pitch, altitude in achieving desired, i.e., for the achievement of control functions. These linear controllers are integrated into the nonlinear model, by gain scheduling with respect to altitude, controller input in terms of longitudinal and lateral linearization regarding the perturbed states and control. The responses of the nonlinear model controlled with the two controllers by various aspect are compared based on flight control requirements.

The fuzzy logic control based on PID system is applied to control three angular angles of nonlinear model.The body axis roll, pitch, yaw are determined parameter tuning process of PID due to obtaining satisfactory using genetic algorithm to elevator, aileron and rudder actuators.

The ANFIS architecture is formulized a systematic approach to generate the fuzzy rule and membership functions.The training data and the checkhing data is obtained from systems which have nonlinear and controlled system model for UAV.The ANFIS controller system architecture and design are applied in the related section in terms of longitudinal and lateral motions respectively.

Keywords: UAV, Nonlinear Modeling, Linearization, Trim, Dynamic Stability, Linear Control, Flight Control System, Root Locus, Target Zeros, Target Poles, Genetic Algorithm, Gain Secheduling, PI Control, PID Control, Fuzzy Logic Controller, ANFIS.

SİMULİNK İLE MODEL İNSANSIZ HAVA ARACI GELİŞTİRİLMESİ : MODELLEMESİ VE KONTROLÜ

ÖZET

Günümüz tarihe gelmeden önce insansız hava araçının tarihçesinden başlayarak ayrıntılı olarak gelişimi araştırılmıştır.M.Ö yıllarda birçok araştırmacı, filozof gibi düşünürlerin buluşları günümüz için esin kaynağı olmuştur.Tezimin başlarında tarihçesinden başlayarak günümüzde kullanılan insansız hava taşıtlarıyla ilgili kısa bilgiler yer almıştır.Aynı zamanda kullanılan bu araçların günümüzde ne amaçlarla kullanıldığı belirtilmiştir.

Doğrusal olmayan bir model insansız hava taşıtımız yer çekimi, atmosferik ve rüzgar türbülansına kalarak modeller oluşturularak 6 serbeslik dereceli hareket denklerindeki aerodinamik bileşenleri göz önünde bulundurularak benzetim modelleri Matlab/Simulink® programında geliştirilmiştir.Geliştirilmiş olan doğrusal olamayan modellimiz doğrusallaştırma işlemi yapılarak analizler elde edilmeye çalışılmıştır.3 yaklaşım metodu kullanılarak doğrusal kontrolcüler tasarlanmıştır.

Kontrolcüler yunuslama, yuvarlanma, sapma hareketleri kullanılarak kök yer eğrisi, PID tabanlı bulanık kontroller ve doğrusal olmayan sistem üzerinde belirtilen koşullara göre eğitilmiş verilerimiz elde edilerek adaptif sinirsel bulanık çıkarım sistemi kullanırak bağlantılı kontrolcü girdileri baz alınarak optimum sonuçlar elde edilmeye çalışılmıştır.

Uçuş kontrol sistemleri için açık ve kapalı çevrim olarak geri besleme yöntemi ile üç farklı hareket kontrolleri verilen referansa uygun şekilde sıfıra en kısa sürede ulaşacak şekilde kontrol edilmiştir.

Bulanık kontrol kullanılarak PID değerleri ayarlanması sağlanarak sistem kontrolü elde edilmeye çalışılmıştır.Aynı zamanda PID değerlerinin buluması tezin en zor kısmını teşkil etmekle birlikte popülasyon değer aralıkları belirlenerek sistemin tanıtılması sonucunda üç hareket üzerinde en optimum değerler belirlenerek sisteme gömülmüştür.Ve sonuç olarak kontrol çeşitleri karşılaştıralarak tablolar halinde en iyi oturma süresine ait kontrolcüler belirlenmiştir.

Sinirsel bulanık sistemler için yapay sinir ağlarının hesaplanması ve öğrenebilme kabileyeti için bulanık mantık uzman bilgilerine ihtiyaç duyulmuştur.Bunlar için önceki bölümde tasarlanan otamatik pilot modelimizin eğitilmiş verileri ve linear olmayan uçak modelimizin verileri ayarlanarak en küçük hata değerine kadar indirgenmiştir.Ve bu şekilde ele alınan model için oluşturulan yapı ve kurallar veriler yardımıyla uzman ataması gerçekleştirilmiştir.Anfis'de öğrenme algoritması olarak, en küçük kareler yöntemi ile geri yayılma öğrenme algoritması bir arada kullanılarak melez öğrenme algoritması kurulmuştur.Birçok katman içinde gerçekleşen üyelik fonsiyonları sistemimiz üzerinde uzunlamasına ve yanlama olarak uçağımızın kontrolü sağlamıştır.

Anahtar Kelimeler: İnsansız hava aracı, Doğrusal olmayan model, Doğrusal model, Kestirim, Doğrusallaştırma, Doğrusal Kontrol, Klasik Uçuş Kontrolü, Oransal Integral Turev Kontrol, Oransal Integral Türev tabanlı Bulanık Mantık Metodu, Bulanım Mantık, Genetik Algoritma, Adaptif Sinirsel Bulanık Çıkarım Sistemi.

1. INTRODUCTION and OVERVIEW

1.1 Introduction of Unmanned Aerial Vehicle

An unmanned aerial vehicle is a remotely piloted vehicle that has no crew or pilots on board. They can be controlled remotely by an operator, or autonomously via preprogrammed flight paths.It is flown remotely by a highly skilled and trained pilot that uses a complex program to fly the aircraft from a remote location and also can be flown by automated programs that give the UAV direction. A majority of UAV's that are in service are being used for military operations.They use them for surveillance and target destruction. There is a wide variety of UAV's, from different shapes and sizes to different characteristics. UAV's have also more recently been used in the civilian sector for surveillance of pipelines, and firefighting and are used when the missions or work is too dangerous or dirty for manned aircraft.

UAV's can be categorized into six main functions of use:

- Target and Decoy
- Reconnaissance
- Combat
- Logistics
- Research and Development
- Civil and Commercial

UAV's really started being used widely across the world in the late 90's. Previous to that the first real UAV's were hot air balloons that were used for weather research. UAV's have come along way since weather balloons, now there are fixed wing and rotary UAV's in full operation.The wide variety of uses has stretch well beyond the military. They are being used in agriculture for fertilizing and pesticide dispersing, all the way up to the use in news broadcasting. UAV's are the future of aerial combat and are playing a major role in the civilian sector. Almost every branch of the military uses some kind of UAV to accomplish missions.

When UAV's were first used in the military is played a very small role. They were first introduced during Vietnam and were typically used as spy planes to gather intelligence about where the enemies were and crucial targets that needed to be destroyed. Now not only can a UAV collect information using cameras, it can be programmed to fly on its own to deliver supplies to troops and to drop bombs on enemy targets.

As technology improves and the need for less expensive methods of achieving certain goals arises UAV's will play a major role not only in the Military but also in the commercial aspect of the world. UAV's being used on a daily basis around the world and are especially playing a major role in the war efforts overseas.

UAVs, like any other new form of technology, have benefits and drawbacks. While these aircraft seem as though they have been brought here from the future, they are not perfect. But as technology moves forward, they seem to be getting closer and closer to being flawless. While there are many different applications in which UAVs can be used, I will be focusing on UAVs being used for military operations seeing as how this is their most common application.

Listed below are the benefits and drawbacks to Unmanned Aerial Vehicles.

Benefits

- Ideal for dangerous missions
- Max G's are limited only by the strenght of the aircraft, not the human inside
- Smaller than manned aircraft
- Can remain airbone for days at a time
- Lower in service costs

Drawbacks

- Many people needed to operate/maintain aircraft
- High development costs
- Delay between operator's inputs and aircraft's reactions
- This can lead to higher risk of incidents
- Do not have the ‘‘ human factor’’ present to trouble shoot or change the game plan when something goes wrong

UAVs are successfully being used in every branch of the military and using UAVs instead of manned aircraft for certain missions keeps our pilots and crews safe. Companies are building unmanned aircraft that can fly higher and faster than any manned aircraft on the planet. Astechnology continues to grow and improvements continue to be made, it is clear to see that the UAV is the future of the military aviation industry.

While the most common use for Unmanned Aerial Vehicles is military operations, they are also being used for numerous civilian purposes such as:

- Agriculture
- News Broadcasting
- Border Patrol
- Highway Surveillance
- Atmospheric / Air Pollution Studies

UAVs are being used in the Middle East by providing troops with locations of enemies as well as delivering supplies to troops in areas where a manned vehicle does not have access to.

1.2 History and Future of UAV

First implementation of flying documented that flying mechanisms has been obtain during ancient era of Pythagoras. The first breakthrough is attributed to Archytas from the city of Tarantas in South Italy, known as Archytas the Tarantine, also referred to as Leonardo Da Vinci of the Ancient World. In 425 B.C. he created the first UAV of all times by building a mechanical bird, a pigeon that could fly by moving its wings getting energy from a mechanism in its stomach. It is alleged that it flew about 200 meters before falling to the ground, once all energy was used [1].

During the same era of the Pythagorean Mathematicians, at another part of the Ancient World, in China, at about 400 B.C., the Chinese were the first to document the idea of a vertical flight aircraft [2].

Leonardo Da Vinci, in 1483, designed an aircraft capable of hovering, called aerial screw or air gyroscope, as shown in Figure 1.1. It had a 5 meter diameter and the idea was to make the shaft turn and if enough force were applied, the machine could spun and fly. This machine is considered by some experts as the ancestor of today's helicopter [3].

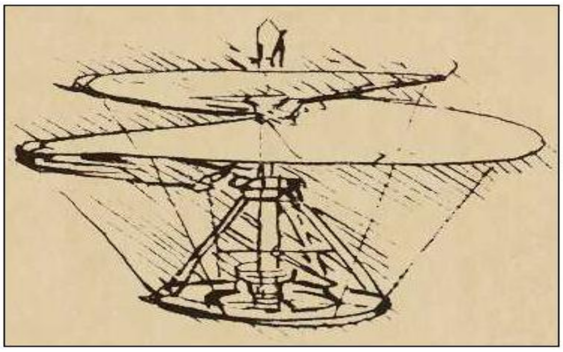


Fig.1.1 : Leonardo Da Vinci's air screw [1].

Further, Da Vinci devised a mechanical bird in 1508 that could flap its wings by means of a double crank mechanism as it descended along a cable as shown in **Figure 1.2** [4].

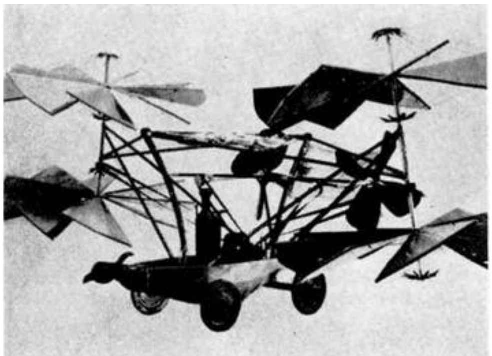


Fig.1.2 : Aerial carriage (Hiller Aviation Museum) [4].

Sir George Cayley of Great Britain drew up plans for this "aerial carriage" that used rotors on opposite sides to counteract torque.This configuration is sometimes still used. Vertical flying models were introduced eventually using an electric motor for power between 1860 and 1907 [2].

In 1907, Paul Cornu developed a two-rotor vertically flying machine that presumably carried the first human off the ground for the first time (**Figure 1.3**). Rotors rotated in opposite directions, the machine flew for about 20 seconds and was merely lifted off the ground [4].

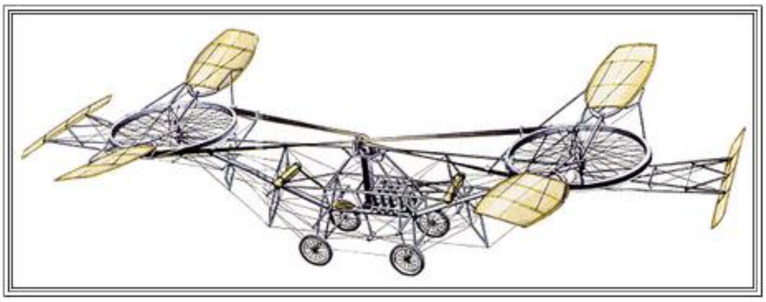


Fig.1.3 : Cornu two-rotor flying machine (Hiller Aviation Museum) [2].

With the first flight demonstrated by the Wright brothers in 1903. Focusing on unmanned fixed-wing aircraft, major breakthroughs happened over the past thirty years. Focusing on unmanned fixed-wing aircraft, major breakthroughs happened over the past thirty years.The First American Unmanned Aerial Vehicle, The Kettering Bug was an experimental, unmanned aerial torpedo. It was capable of striking ground targets up to 121 kilometres (75 mil) from its launch point, while traveling at speeds of 80 kilometres per hour (50 mph). The Charles Kettering Aerial Torpedo, also known as the Kettering Bug, was a parallel effort backed by the American Army as shown in Figure 1.4 [13].

Fig.1.4 Kettering Bug [13].

The first of the so-called Vergeltungswaffen series designed for terror bombing of London, the V-1 as shown figure 1.5 was fired from launch sites along the French (Pas-de-Calais) and Dutch coasts. The first V-1 was launched at London on 13 June 1944 [5].

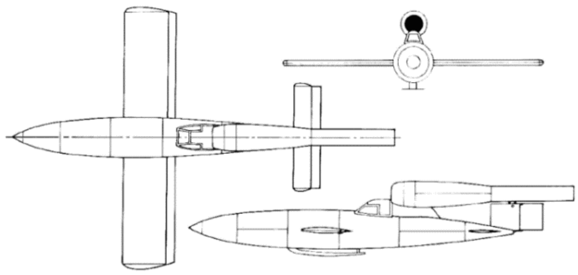


Fig. 1.5 : Fieseler FI 103 V1 [5].

The Interstate TDR was an early unmanned combat aerial vehicle in **Figure 1.6** — referred to at the time as an "assault drone" — developed by the Interstate Aircraft and Engineering Corporation during the Second World War for use by the United States Navy [14].

Fig.1.6 : Interstate BQ-4/TDR [14].

The Ryan Firebee was a series of target drones developed by the Ryan Aeronautical Company beginning in 1951.It was one of the first jet-propelled drones, and one of the most widely used target drones ever built. As the XQ-2 these 22-foot-long pilotless drones were powered by a 1,060 pound-thrust Continental J69-T-19B turbojet engine providing a top speed well in excess of 500 mph. as shown **figure 1.7** [15].

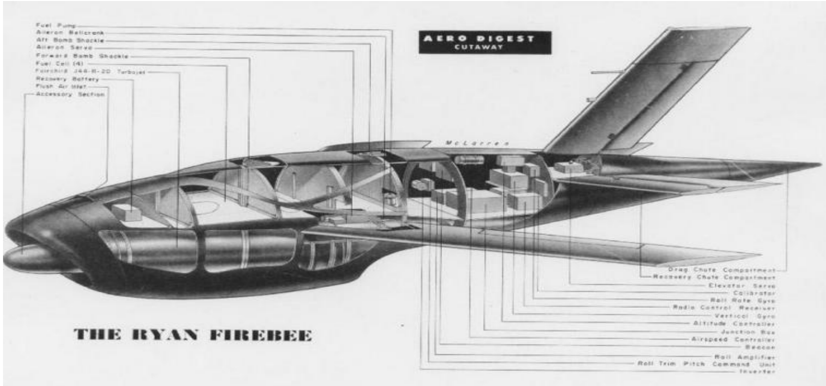


Fig.1.7 : Ryan Firebee [15].

The Northrop SM-62 Snark was an early-model intercontinental cruise missile that could carry a W39 thermonuclear warhead. The Snark was deployed by the U.S Air Force's Strategic Air Command from 1958 through 1961. The jet propelled 20.5 meter-long Snark missile had a top speed of about 650 m.p.h. (1,046 kilometer/hour) and a maximum range of about 5,500 nautical miles (10,200 kilometers).The fuselage would have performed an abrupt pitch-up maneuver in order to avoid a collision with the warhead [16].

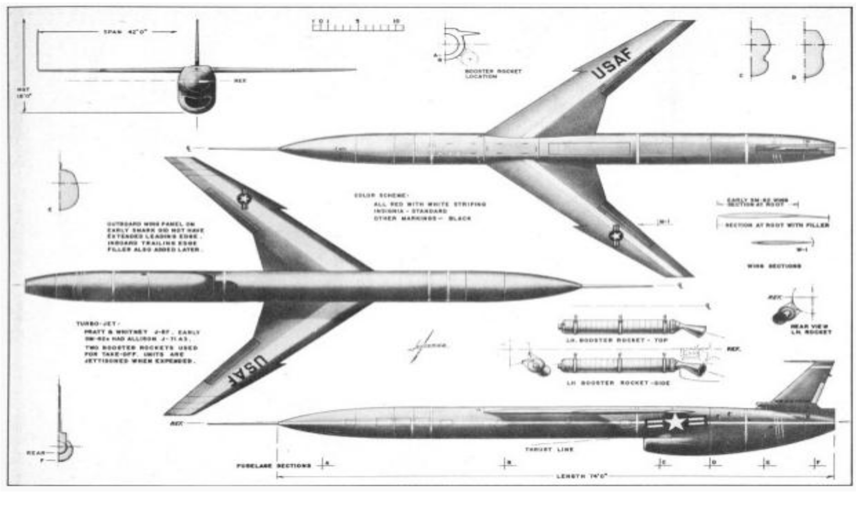


Fig.1.8 : Northrop SM-62 [16].

The GNAT 750's configuration was similar to that of the Amber, except that the GNAT 750's wing was mounted low on the fuselage, instead of being mounted on a pylon on top. The aircraft is powered by a Rotax 912 piston flat-four four-cycle engine with 64 kW (85 hp). It can fly to an operational area from 2,000 kilometers (1,240 miles) away and loiter there for 12 hours before returning home [6].

Table 1.1 : Gnat-750 and I-Gnat [6].

	Gnat-750	I-Gnat
Length	5.33 m (17 ft 6 in)	6.32 m (20 ft 9in)
Wingspan	10.76 m (35 ft 3.6 in)	12.86 m (42 ft 2.4 in)
Weight	Max: 510 kg (1126lb); empty 254 kg (560 lb)	Max: 700 kg (1550 lb);empty: 385 kg (850lb)
Speed	259 km/h (161 mph)	296 km/h (184 mph)
Ceiling	7620 m (25000 ft)	9140 m (30000 ft)

The last unmanned aerial vehicle is Global Hawk. The development of UAVs continued in fits and starts with some military commanders questioning their usefulness. Global Hawk is an example for long range/long endurance UAV which is equipped with several sensors, cameras and radars for surveillance. NASA prefers the Global Hawk for atmospheric research at high altitude [13].

Global Haws specifications are,

Lenght 44ft(13.5m), Wingspan 116.2ft(35.4m), Height 15.2ft(4.6m)

as air vehicle true airspeed 335 KTAS, Turn radius 6.5 nmi, Altitude 65000 ft,

Payload >1500 lb, Range 11000+nmi /31 hours

Take-off fuel: 14500 lb , Take off gross weight: 25600 lb

DC(engine generator) 186.3A (5.2 KW), AC 115 VAC, 3 phase 400 Hz

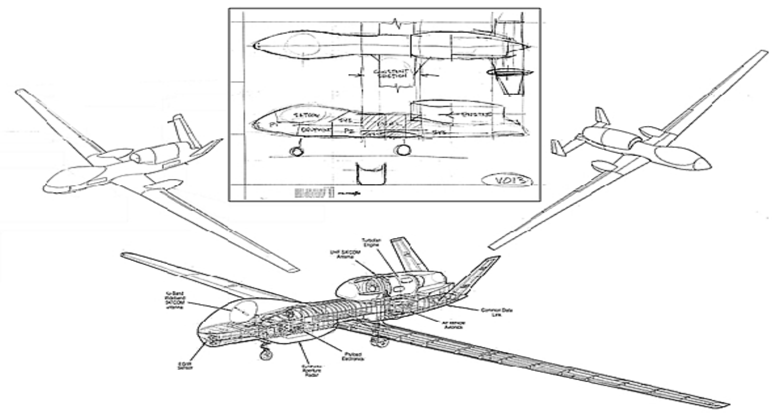


Fig.1.9 : Global Haws [13].

1.3 Outline of Thesis

The compeleted thesis structure is shown flow diagram below.

Aircraft equations → Modelling → Analysis → Control → Scope

Based on Root Locus Contr.   FLC based on PID using GA   ANFIS

The thesis consists of all aspects of the aircraft design from first principles to the final principles and solutions. In section 2, a nonlinear aircraft model is deveploed with equations and simulink model as all of motions, but the nonlinear aircraft model is not satisfied to correct solutions.In section 3, Nonlinear model is linearised using small disturbance technique, about a level flight trim condition. The linear aircraft model, specific to the aircraft used in this thesis, also finding some of values is analysed in section 4 using model decomposition tecniques. The analysis sets for the controller design settling root locus of section 5, where the controller architecture is developed and design details are presented in root locus based and fuzzy logic control based on PID using GA is presented in section 6.Adaptive neuro fuzzy inference system is developed in section 7, where obtained training data using developed nonlinear and linear aircraft model.Lastly, Conclusion In section 8.

2. DEVELOPMENT OF NONLINEAR SIMULINK MODEL

2.1 Introduction

It is well known that for the analysis of an aircraft dynamics and linear controller applications on this dynamics, linear models are needed. By the development of a nonlinear model on which a linearization procedure is carried out, linear models can be obtained. In addition, a detailed simulink environment generated by the nonlinear model itself helps visualizing and analyzing various flights including maneuvers thereby determining the limitations for the UAV throughout the operational flight envelope, But these additional items in the thesis do not have. MATLAB/Simulink® provides effective means of modeling, simulation, and controller development to the designer.

The basic geometrical data of the UAV that is referred to during the computation of aerodynamic coefficients and nonlinear model development phase is given in **Table 2.1**.

Table 2.1 : Basic Geometrical Data.

Mass/Center of Mass/Inertial Moments	Values
Mass, m [kg]	5.4795
Moment of inertia about x-axis, Ixx [kg.m2]	0.9036
Moment of inertia about y-axis, Iyy [kg.m2]	0.8196
Moment of inertia about z-axis, Izz [kg.m2]	1.4721
xz product of inertia, Ixz [kg.m2]	0.0648
xy product of inertia, Ixy [kg.m2]	0
yz product of inertia, Iyz [kg.m2]	0
Fuselage	**Values**
Length [m]	1.727
Wing	**Values**
Surface area, S [m2]	1.0217
Span, b [m]	2.68
Aspect ratio, A 22	0
Sweep angle at 25% chord, Λ	0
Tip chord, ct [m]	0.4

2.2 Dynamic Equation of Aircraft

Related Chapter begins by defining the all of axis systems used in the thesis. The equations of motion for rigid aircraft are developed from first principles using Newton's laws.The kinematic equations are developed, the motion of as a rigid body can be described by a set of six nonlinear differential equations including the translational and rotational motion of the aircraft.Lastly obtaining forces and moments that these equations are written in terms of aircraft's states.

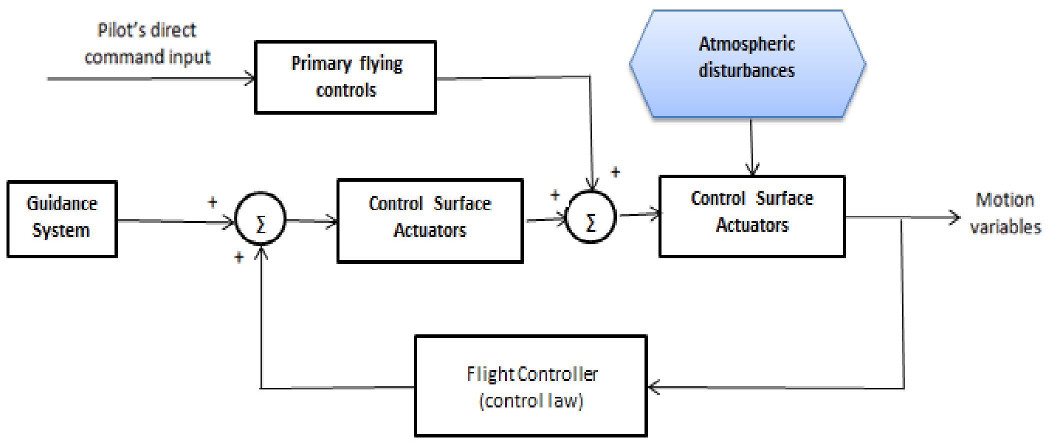


Figure 2.1 : Developed aircraft system.

2.2.1 Assumptions

The air vehicle is modeled as a standart 6 DOF system with the following assumptions:

1. The aerodynamic database composed of static and dynamic aerodynamic coefficients does not include any nonlinearities at low speeds.
2. Only the aerodynamic coefficients for the flaps-up configuration are included, i.e., takeoff and landing phases are not to be taken into account during analyses or flight control system design.
3. Rigid body assumption is done, i.e aeroelastic effects are not included.
4. Ground effect is not included only atmospheric effect is included.
5. Landing gear model is not included.

6. Hinge moments effects are not included.
7. Airframe has a fixed centre of mass position.
8. Vehicle mass and moments of inertia are fixed time invariant quantities.
9. Vehicle has a centered longitudinal plane of symmetry that passes through the c.m
10. Gravitational acceleration, g is constant over the air vehicle body.
11. Earth is flat and fixed in space, and atmosphere is fixed with respect to Earth.

2.3 Definition of Axis System for Aircraft

2.3.1 Inertial axes

In inertial axis system, all of Newton's(first,second) laws can be applied.Before Newton's laws can be used to develop a dynamic model for the aircraft.
The inertial axis system, $F_E(O_E X_E Y_E Z_E)$, is chosen as an earth fixed axis system with its origin at some appropriate point on the earth's surface.

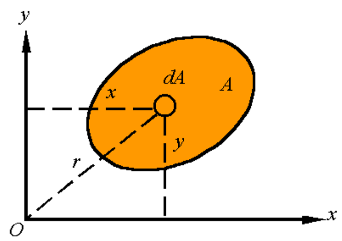


Figure : 2.2 Inertial axes on earth [17].

This axis system is right handed with the positive Z_E axis pointing toward the centre of the earth.The X_E axis is perpendicular to the Z_E-axis, positive to the North.The Y_E axis is perpendicular to the Z_E axis, positive to the North. The Y_E axis is perpendicular to the $X_E Z_E$ plane, positive to the East.

2.3.2 Body axes

As the aircraft model is obtained from essential of the body axis system.The body axis system, $F_B(O_B X_B Y_B Z_B)$, is a right handed reference frame, attached to the aircraft, with is center related with the aircraft's centre of gravity.

The x axis is directed toward the nose of the aircraft, the y axis toward the right wing, and the z axis toward the bottom of the aircraft. X_B axis is in the aircraft's plane of symmetry and is positive in the forward direction.The exact forward direction is determined by the specific body axis system used.The Z_B axis is also in the plane of symmerty, perpendicular to the X_B axis and positive pointing downward.The Y_B axis is perpendicular to the $X_B Z_B$ plane, positive to the right.The aircraft is said to roll about X_B axis, pitch about the Y_B axis and yaw about the Z_B axis with the positve directions as shown in **Figure 2.2**.

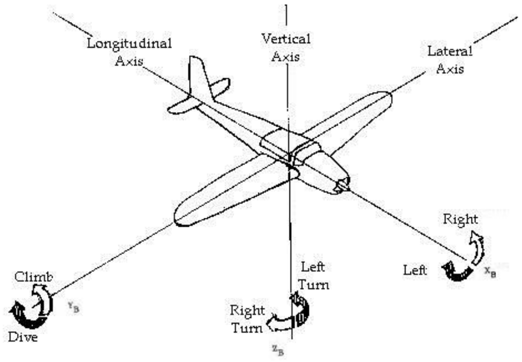


Figure 2.3 : Body axes on aircraft [6].

- **Stability axes on aircraft**

Forward direction of longitudinal axis (OX_B axis) related with aircraft velocity vector during trim flight condition so this body axes system is named to as stability axes. The aircraft linear and angular velocity vector as well as external force and moment vectors are defined below.The relationships between the body, wind and stability axes as shown in **Figure 2.3**. All three systems have their origin at the center of gravity of the aircraft.The x axis in the wind reference system is aligned with the velocity vector the aircraft.The angle of sideslip β and angle of attack α define the orientation of the wind axes with respect to the body axes.

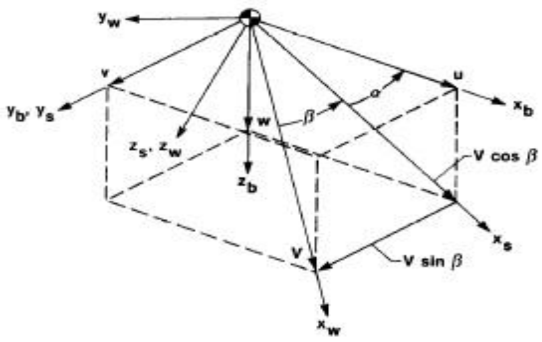


Figure 2.4 : Relation of body,stability and inertial axes [10].

Definitions of all vectors will be inspiration for modelling of nonlinear system. It is important to realise that they are all measured relative to inertial space, but are coordinated in the instantaneous body frame.

$$V_B = \mathrm{U}i_B + \mathrm{V}j_B + \mathrm{W}k_B \tag{2.1}$$

$$w_B = \mathrm{P}i_B + \mathrm{Q}j_B + \mathrm{R}k_B \tag{2.2}$$

$$F_B = \mathrm{X}i_B + \mathrm{Y}j_B + \mathrm{Z}k_B \tag{2.3}$$

$$M_B = \mathrm{L}i_B + \mathrm{M}j_B + \mathrm{N}k_B \tag{2.4}$$

The relationship between vectors express in different coordinate systems can be derived using rotation matrices.The body-fixed coordinate system is first rotated a negative sideslip angle β about the z-axis.The new coordinate system is then rotated a positive angle of attack α about the new y-axis such that the resulting x-axis points in the direction of the total V_T.The first rotation defines the wind axes while the second rotation defines the stability axes.This can be mathematically expressed as where i,j,k are unit vectors along the $X_B Y_B Z_B$ denoted body-fixed axes.In **Table 2.2**, the positive for dynamic angles α, β, and Euler angles φ,θ,ψ are defined, whereas in **Figure 2.4** the positive directions for these angles and body-fixed axes components are illustrated on representetive air vehicle.

Table 2.2 : Positive Sign Conventions for Angles

Parameter	Symbol	Positive direction
Angle of attack	α	Nose up with respect to freestream
Angle of sideslip	β	Nose left, looking forward
Roll angle	ϕ	Right wing down, looking forward
Pitch angle	θ	Nose up
Yaw angle	ψ	Nose right, looking forward

The positive sign conventions of the control surface deflections and the resultant moments should also be defined. The control surfaces of the subject UAV are composed of rudder and ailerons (flaps are thought as configuration components).

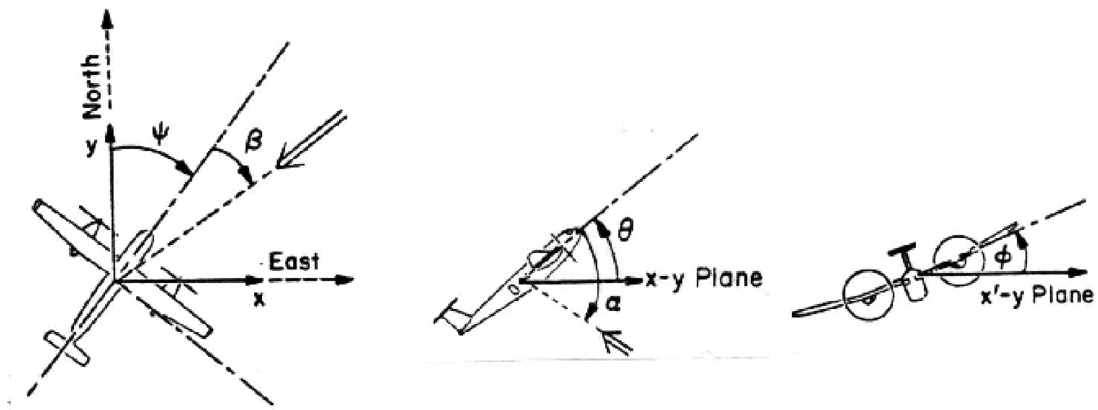


Figure 2.5 : Positive directions for body-fixed axes components and angles [19].

$$P^{wind} = R_{z,-\beta}P^{stab} = \begin{bmatrix} \cos(\beta) & sin\,(\beta) & 0 \\ -sin\,(\beta) & cos\,(\beta) & 0 \\ 0 & 0 & 1 \end{bmatrix} P^{stab} \quad \textbf{(2.5)}$$

$$P^{stab} = R_{y,-\alpha}P^{body} = \begin{bmatrix} \cos(\alpha) & 0 & sin\,(\alpha) \\ 0 & 1 & 0 \\ -sin\,(\alpha) & 0 & cos\,(\alpha) \end{bmatrix} P^{body} \quad \textbf{(2.6)}$$

The rotation matrix becomes:

$$R_{body}^{wind} = R_{z,-\beta}R_{y,\alpha} \quad \textbf{(2.7)}$$

Hence,

$$P^{wind} = R_{body}^{wind}\, P^{body} \quad \textbf{(2.8)}$$

$$P^{wind} = \begin{bmatrix} \cos(\beta) & \sin(\beta) & 0 \\ -\sin(\beta) & \cos(\beta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\alpha) & 0 & \sin(\alpha) \\ 0 & 1 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix} P^{body} \tag{2.9}$$

$$P^{wind} = \begin{bmatrix} \cos(\beta)\cos(\alpha) & \sin(\beta) & \sin(\alpha)\sin(\beta) \\ -\cos(\alpha)\sin(\beta) & \cos(\beta) & -\sin(\alpha)\sin(\beta) \\ -\sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix} P^{body} \tag{2.10}$$

This gives the following relationship between the velocities in body and wind axes [7] :

$$V^{body} = \begin{bmatrix} U \\ V \\ W \end{bmatrix} = (R_{body}^{wind})^T v^{wind} = (R_{y,\alpha})^T (R_{z,-\beta})^T \begin{bmatrix} V^T \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} V^T\cos(\beta)\cos(\alpha) \\ V^T\sin(\beta) \\ V^T\cos(\beta)\sin(\alpha) \end{bmatrix} \tag{2.11}$$

Consequently,

$$U = V^T\cos(\beta)\cos(\alpha) \tag{2.12}$$

$$V = V^T\sin(\beta) \tag{2.13}$$

$$W = V^T\cos(\beta)\sin(\alpha) \tag{2.14}$$

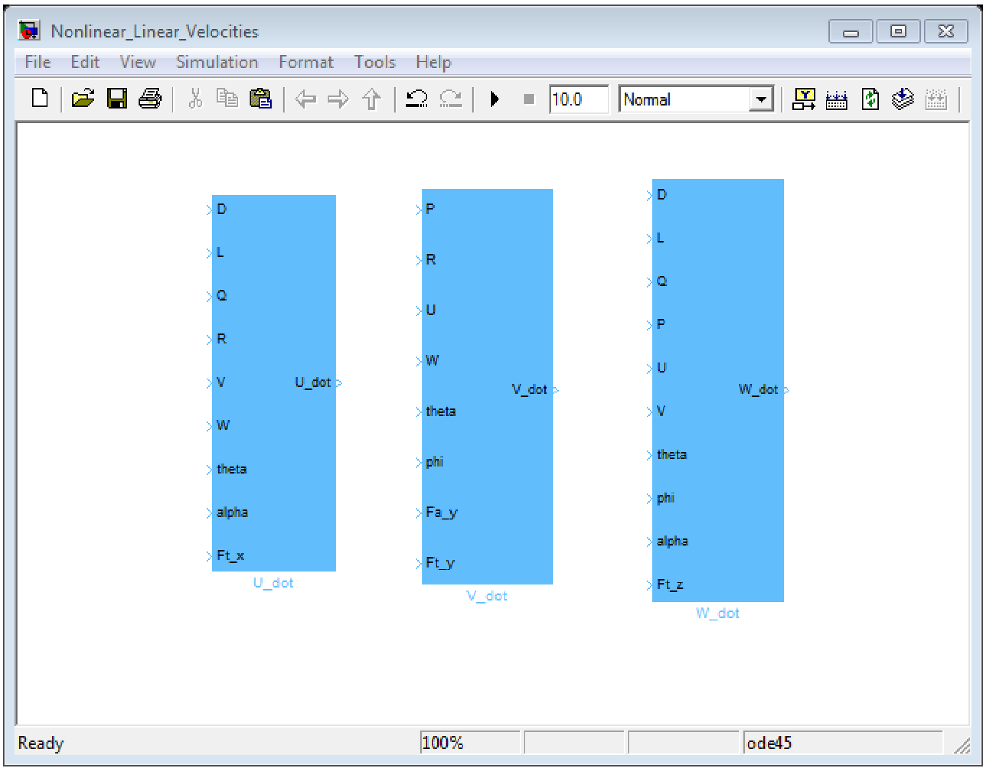


Figure 2.6 Nonlinear linear velocities model in Simulink.

where, i_B, j_B and k_B are unit vectors in the OX_B, OY_B, OZ_B axis directions respectively.The linear velocity vector V_B can be written in polar form,

$$\hat{V} = \sqrt{U^2 + V^2 + W^2} \quad \textbf{(2.15)}$$

$$\alpha = \tan^{-1}(W/U) \quad \textbf{(2.16)}$$

$$\beta = \sin^{-1}(V/\hat{V}) \quad \textbf{(2.17)}$$

where $\hat{V}$, α and β are the aircraft speed, angle of attack and angle of sideslip respectively.The speed is denoted with a bar so that it is not confused with the lateral velocity variable V. The conventional aerodynamic control surface deflection variables are defines below,

- δ_A – Aileron control surface deflection
  A posive deflection causes a negative rolling moment.
- δ_E – Elevator control surface deflection
  A posive deflection causes a negative pitching moment.
- δ_R – Rudder control surface deflection
  A posive deflection causes a negative yawing moment.

2.4 Aircraft Equation of Motion

The axis systems and notation have been defined, the equations of motion for a rigid aircraft can be derived from Newton's second law of motion. Through the application of Newton's second law, two vector relations can be used to derive the six required equations, three translational and three rotational. This states that the summation of all external forces acting on a body must be equal to the time rate of change of its momentum, and the summation of all external moments must be equal to the time rate of change of its momentum, and the summation of all external moments must be equal to the time rate of change of its angular momentum [7]. The time of change are with respect to inertial space and the laws can be expressed by the following vector equations.

$$F_B = \frac{d}{dt}\,(\mathrm{m}V_B)\Big]_E \quad \textbf{(2.18)}$$

$$M_B = \frac{d}{dt}\,(\mathrm{m}H_B)\Big]_E \quad \textbf{(2.19)}$$

where m is the aircraft's mass and H_B is the aircraft's angular momentum defined as,[15].

$$H_B = I_B w_B \tag{2.20}$$

where,

$$I_B = \begin{bmatrix} I_x^2 & I_{xy} & I_{xz} \\ I_{xy} & I_y^2 & I_{yz} \\ I_{xz} & I_{yz} & I_z^2 \end{bmatrix} \tag{2.21}$$

is the moment of inertia dynamic of the aircraft in body axes. the product of inertia, I_{xy} and I_{yz} are zero. The equation of Coriolis is used to relate the intertial time derivative to the body time derivate.This is given as,

$$\frac{d}{dt}\ (\mathrm{R})\ \Big]_E = \frac{d}{dt}\ (\mathrm{R})\ \Big]_B + \ w_B \ \mathrm{x}\ \mathrm{R} \tag{2.22}$$

where R is an arbitrary vector.
Appling equation,

$$F_B = \ \frac{d}{dt}\ (\mathrm{m}V_B)\Big]_B + w_B \ \mathrm{x}\ (\mathrm{m}V_B) \tag{2.23}$$

$$M_B = \ \frac{d}{dt}\ (I_B w_B)\Big]_B + w_B \ \mathrm{x}\ (I_B w_B) \tag{2.24}$$

Equations above are the vector equations for general six degree of freedom motion.They are commonly displayed in scalar form as a set of six, non-linear coupled differential equations. Equations above are the vector equations for general six degree of freedom motion.They are more commonly displayed in scalar form as a set of six, non-linear, coupled differential equaitons. The nonlinear flight dynamics of an airplane is represented by translational and rotational equations of motion. In this section, the translational and rotational (angular) equations of motion, kinematical relations in body-fixed axes and equations for air vehicle coordinates are given.These are shown below assuming only that the aircraft is symmetrical about the $x_B z_B$ plane and that the inertial properties do not change over time [20].
Translational equations of motion:

Drag Equation : $\mathrm{X} = \mathrm{m}\ (\dot{\mathrm{U}} + \mathrm{WQ} - \mathrm{VR})$ **(2.25)**

Side Force Equation : $\mathrm{Y} = \mathrm{m}\ (\dot{\mathrm{V}} + \mathrm{UR} - \mathrm{WR})$ **(2.26)**

Lift Equation : $\mathrm{Z} = \mathrm{m}\ (\dot{\mathrm{W}} + \mathrm{VR} - \mathrm{UQ})$ **(2.27)**

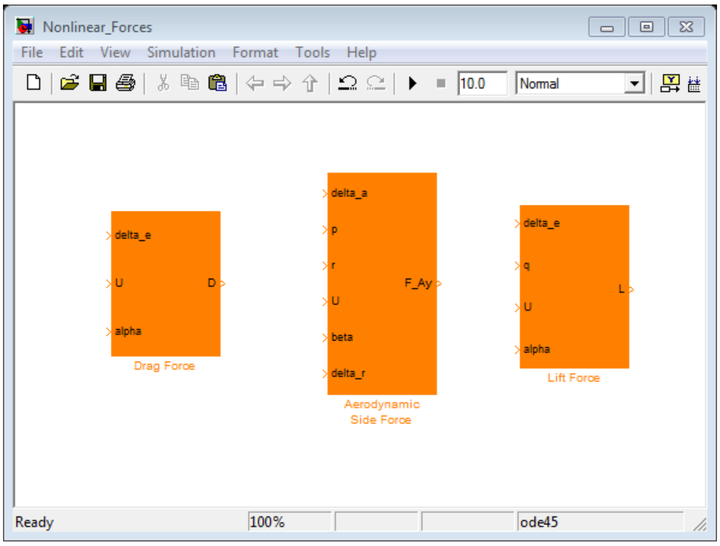


Figure 2.7 : Nonlinear forces model in Simulink.

<u>Rotational equations of motion:</u>

Roll Moment Equation: $L = \dot{P}I_x - \dot{R}I_{xz} + QR\,(I_z - I_y) - PQI_{xz}$ **(2.28)**

Pitch Moment Equation: $M = \dot{Q}I_y + PR(I_x - I_z) + (P^2 - R^2)\,I_{xz}$ **(2.29)**

Yaw Moment Equation : $N = \dot{R}I_z - \dot{P}I_{xz} + PQ\,(I_y - I_x) + QRI_{xz}$ **(2.30)**

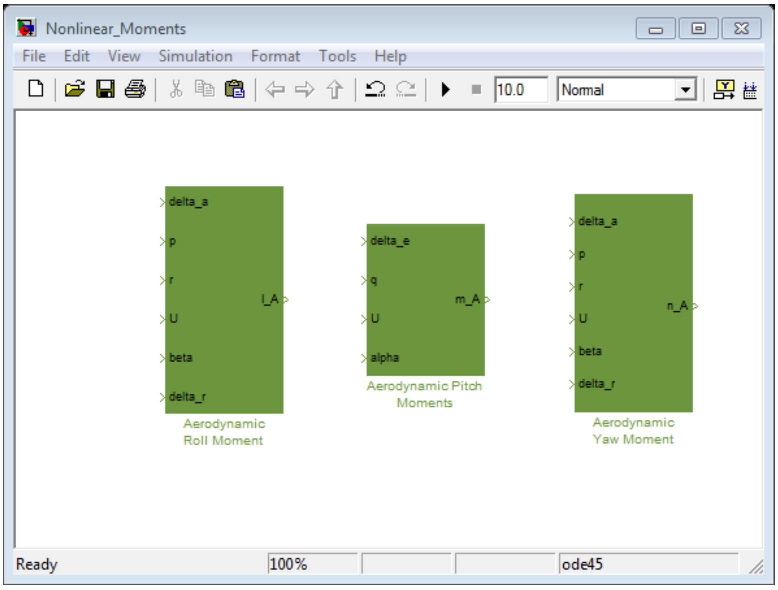


Figure 2.8 : Nonlinear moments model in Simulink.

2.5 Kinematic Equations for Translation

The kinematic equations for translation and rotation of a body-fixed coordinate system can be expressed in terms of the Euler angles.Define an axis system $F_0(O_0X_0Y_0Z_0)$ that related with the intertial axis system metions above F_E. A vector V coordinated in F_0 and described as V_0 = $[x_0, y_0, z_0]$. F_0 is now moved through three consecutive rotations to align itself with the body axis system with the goal of attaining a relationship between V coordinated in the orginal and final axis systems.

1.Rotate F_0 about the OZ_0-axis by the yaw angle ψ, to create a new coordiante system $F_1(O_1X_1Y_1Z_1)$.The vector V, coordinate in F_1, is now described as V_1=$[x_1, y_1, z_1]$.Note that the rotation leaves z_0=z_1.

2.Rotate F_1 about the OY_1-axis by the pitch angle θ, to create a new coordiante system $F_2(O_2X_2Y_2Z_2)$.The vector V, coordinate in F_2, is now described as V_2=$[x_2, y_2, z_2]$.Note that the rotation leaves y_1=y_2.

3.Rotate F_2 about the OX_2-axis by the roll angle φ, to create a new coordiante system $F_3(O_3X_3Y_3Z_3)$.The vector V, coordinate in F_3, is now described as V_3=$[x_3, y_3, z_3]$.Note that the rotation leaves x_2=x_3.

The coordiante system F_3 coincides with the body axis system F_B.With the rotation sequence defined, the transformation matrix that convertes V_0 to V_3 or equivalently V_E to V_B can easily be found [20].

$$\begin{bmatrix} \dot{X}_E \\ \dot{Y}_E \\ \dot{Z}_E \end{bmatrix} = R_{abc}^{ned} = \begin{bmatrix} U \\ V \\ W \end{bmatrix} = R_{z,\psi} R_{y,\theta} R_{x,\varphi} \begin{bmatrix} U \\ V \\ W \end{bmatrix} \qquad \textbf{(2.31)}$$

Expanding this expression gives:

$$\begin{bmatrix} \dot{X}_E \\ \dot{Y}_E \\ \dot{Z}_E \end{bmatrix} = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\varphi & -s\varphi \\ 0 & s\varphi & c\varphi \end{bmatrix} \begin{bmatrix} U \\ V \\ W \end{bmatrix} \qquad \textbf{(2.32)}$$

$$\begin{bmatrix} \dot{X}_E \\ \dot{Y}_E \\ \dot{Z}_E \end{bmatrix} = \begin{bmatrix} c\psi c\theta & -s\psi c\varphi + c\psi s\theta s\varphi & s\psi s\theta + c\psi c\theta s\theta \\ s\psi c\theta & c\psi c\varphi + s\varphi s\theta s\psi & -c\psi s\theta + s\theta s\psi c\varphi \\ -s\theta & c\theta s\varphi & c\varphi c\varphi \end{bmatrix} \begin{bmatrix} U \\ V \\ W \end{bmatrix} \qquad \textbf{(2.33)}$$

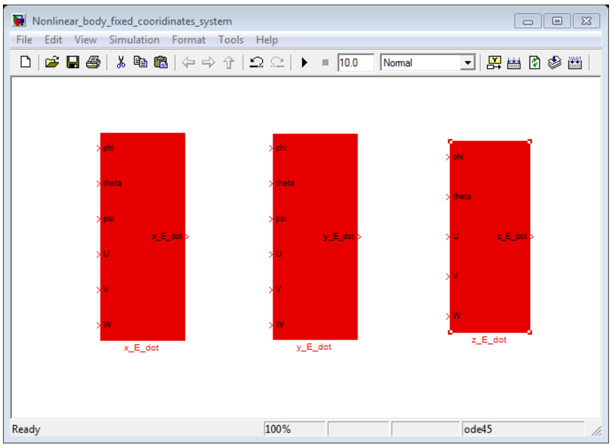


Figure 2.9 : Nonlinear body fixed coordinates model in Simulink.

2.6 Kinematic Equations for Attitute

The dynamic equations developed in above section describe how the forces the moments coordinated in the body axis frame cause rates of change in the body axis linear and angular velocities [9].

The attitude is given by:

$$\begin{bmatrix} P \\ Q \\ R \end{bmatrix} = \begin{bmatrix} \dot{\varphi} \\ 0 \\ 0 \end{bmatrix} + R_{x,\varphi}^{T} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R_{x,\varphi}^{T} R_{y,\theta}^{T} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \tag{2.34}$$

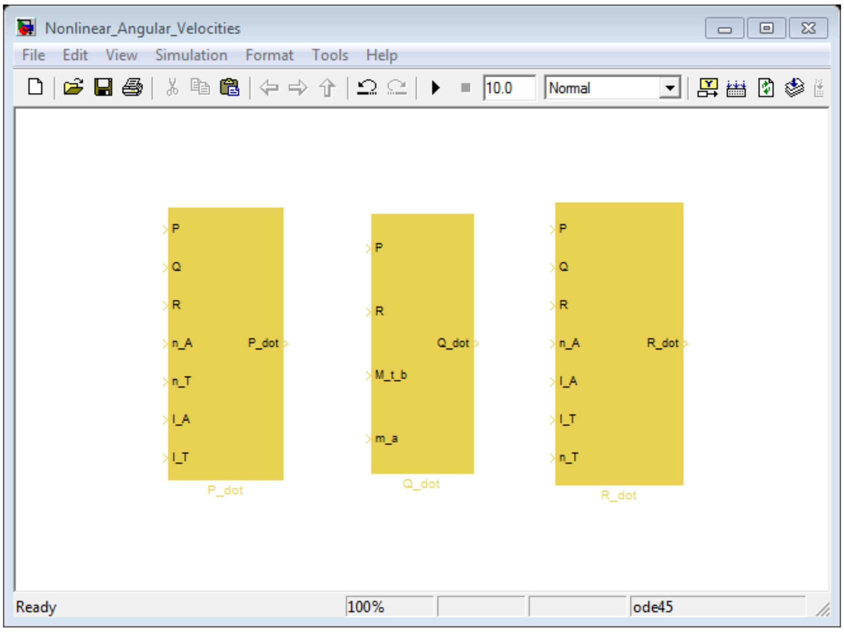


Figure 2.10 : Nonlinear angular velocities model in Simulink.

Which gives:

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & s\varphi t\theta & c\varphi t\theta \\ 0 & c\varphi & -s\varphi \\ 0 & s\varphi/c\theta & c\varphi/c\theta \end{bmatrix} \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \qquad \textbf{(2.35)}$$

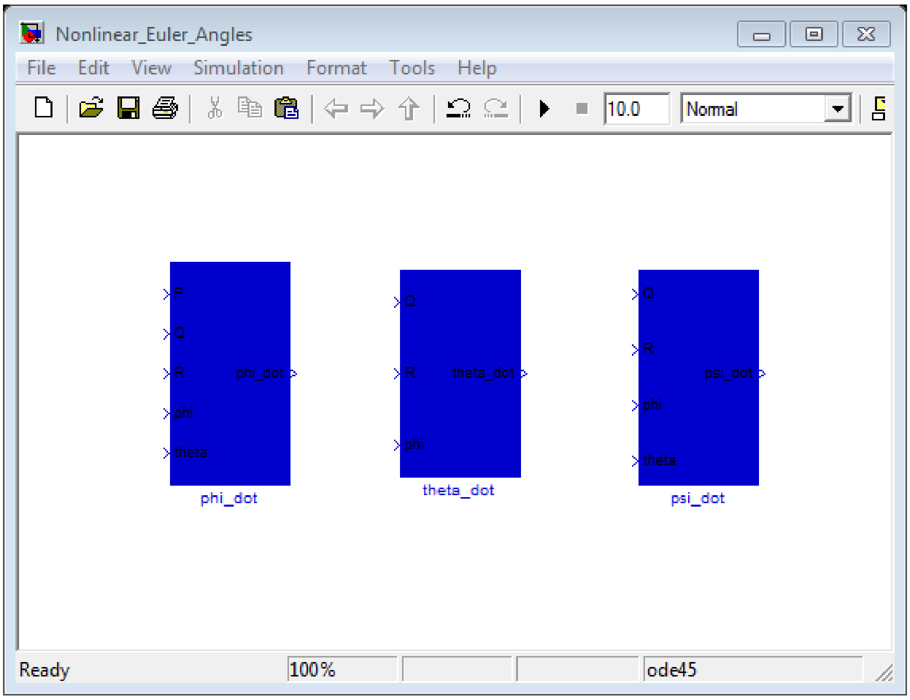


Figure 2.11 : Nonlinear Euler angles model in Simulink.

This is commonly done using a set of three Euler angles, which define an ordered set of rotations from one axis system to another.

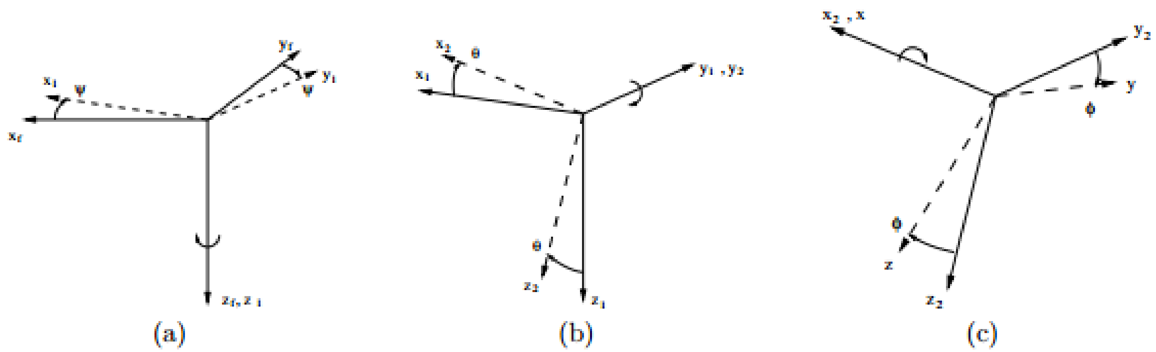


Figure 2.12 : The Euler angles ψ, θ, and φ [20].

Euler angles determines the orientation of the body axes of a flight vehicle

a) Yaw rotation about z-axis, nose right
b) Pitch rotation about y-axis, nose up
c) Roll rotation about x-axis, right wing down

The orientation of the body axis system is specified by starting with the inertial system, then, in the following order performing:

1. a positive rotation about the z_f axis through the heading angle ψ to produce the (x_1, y_1, z_1) system; then
2. a positive rotation about the y_1 axis through the pitch angle θ to produce the (x_2, y_2, z_2) system; then
3. a positive rotation about the x_2 through the bank angle φ to produce (x,y,z) system.

2.7 Forces and Moments

The contributions to forces acting on the air vehicle are from aerodynamics, thrust, and gravity, whereas contributions to moments are from aerodynamics and thrust.Before the equations of motion above can be solved, moments and forces depending on athe aircraft must be determined.All body axis system developed to one point are found the equation of motions that O$X_B Z_B$– related with the symmtery aircraft. The moments of the forces were taken was always at this point to the plane containing the forces.

The aerodynamic forces and moments acting on an aircraft are the result of multiple factors whose significance varies with flight condition as well as from vehicle to vehicle.In general , these forces and moments are nonlinear functions primarily of Mach number, angle of attack, angle of sideslip, altitude, rotational rates, and control-surface deflections.The aerodynamic forces and moments are assumed to be functions having the following form [15].

$$F = \varphi\,(\alpha, \beta, V, h, p, q, r, \dot{\alpha}, \dot{\beta}, \delta_1, \ldots, \delta_n) \qquad \textbf{(2.36)}$$

where F is an arbitrary force or moment, φ is an arbitrary function and the δ_i are the n control surface deflections.

2.7.1 Aerodynamic

The aircraft aerodynamic analysis is restriced into the two dimensional as incompressible flow and subsonic.Two dimensional means that the airflow velocity at every point on an airfoil has a direction that is paralel to some reference plane, such as the O$X_B Z_B$– axis during symetrical flow.As a result, the aerodynamic forces can merely be resolved into lift and drag components.Restricting the analysis in terms of subsonic, incompressible flow greatly simplfies the aerodynamic analysis and is a reasonable assumption for model aircraft flight, where speeds are well below Mach 0.4.

Force coefficients and their moment arms with reference to a given axis are summed up to find the global aircraft moment coefficients in body axes.

$$X_A = \frac{1}{2} p \, \bar{V}^2 S C_x \tag{2.37}$$

$$Y_A = \frac{1}{2} p \, \bar{V}^2 S C_y \tag{2.38}$$

$$Z_A = \frac{1}{2} p \, \bar{V}^2 S C_z \tag{2.39}$$

$$L_A = \frac{1}{2} p \, \bar{V}^2 S b C_l \tag{2.40}$$

$$M_A = \frac{1}{2} p \, \bar{V}^2 S \bar{c} C_m \tag{2.41}$$

$$N_A = \frac{1}{2} p \, \bar{V}^2 S b C_n \tag{2.42}$$

where S,b and $\bar{c}$ are the aircraft's wing area, wing span and wing mean aerodynamic chord respectively and are used as the reference areas and lengths for the aircraft nondimensional aerodyamic analysis.

It is now necessary to develop equations for the dimensionless force and moment coefficients of equations above.These coefficients are functions of the force and moment coefficients of the various aircraft airfoils and the aircraft geometry surfaces: the wing, tailplane, fin and fuselage.

2.8 Actuators Model

The UAV model has four control surfaces:

Two ailerons: right (starboard) and left (port) ailerons; always deflect asymmetrically in the same magnitudes, i.e. they do not move independently and a single actuator is used.

Two rudder: right and left rudder; a symmetric deflection gives the effect of an elevator whereas an asymmetric deflection gives the effect of a rudder requiring independent movements and actuators.

2.9 Atmosphere and Wind-Turbulence Model

2.9.1 Atmosphere model

With this model, the mathematical representation of the international standard atmosphere values for ambient temperature, pressure, density, and speed of sound for the input geo-potential altitude is implemented.

2.9.2 Wind-Turbulence model

For the evaluation of the performance of the air vehicle control systems, it is necessary to include wind and atmospheric turbulence to simulations of the air vehicle. In this section, the utilized wind-turbulence model components are described.

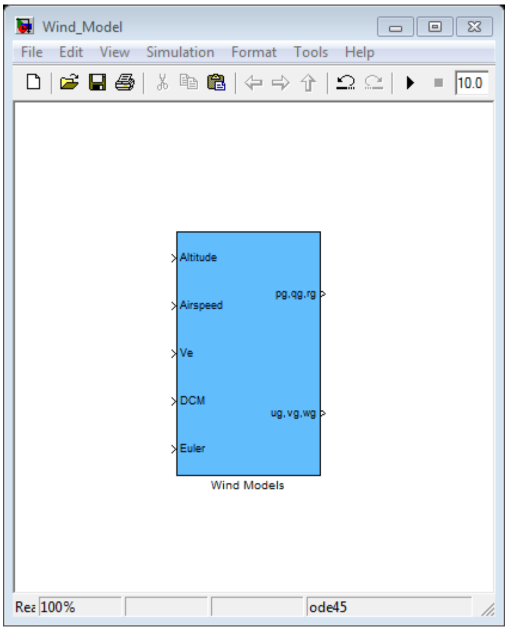


Figure 2.13 : Atmospheric wind model in Simulink.

2.9.2.1 Background wind model

This model computes the background wind velocity and acceleration components in the body-fixed axes. The wind velocity vector in the Earth-fixed frame (North-East-Down) is multiplied with the rotation matrix DCM, for its components to be transformed into wind velocities in body-fixed axes. Wind accelerations in the bodyfixed axes are also obtained, by taking the numerical time derivative of the wind velocity vector in the Earth-fixed frame and matrix multiplying with DCM in a similar manner. This captures the effect of time-varying background.

Wind which can be encountered in some weather conditions (wind shear, thermals, and cyclones).

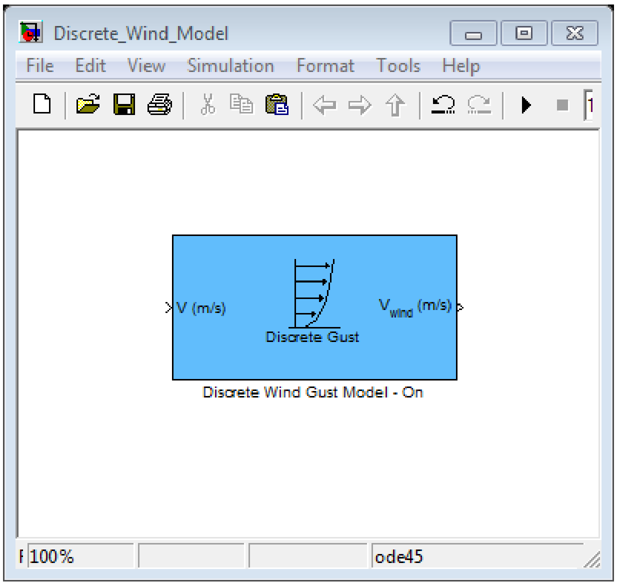


Figure 2.14 : Discrete wind gust model in Simulink.

2.9.2.2 Turbulence model

The computation is performed by implementing the Von Kármán spectral representation to add turbulence to the nonlinear model by passing a band-limited white noise through appropriate longitudinal, lateral and vertical turbulence shaping filters. The filter parameters depend on background wind magnitude and current air vehicle altitude. One can refer to [4, 17, 30] for the detailed mathematical representation of the Von Kármán turbulence model [10].

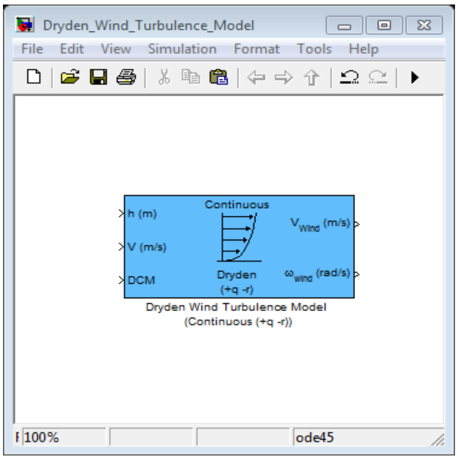


Figure 2.15 : Dryden wind turbulence model in Simulink.

2.9.2.3 Wind shear model

The wind shear model computes the body-fixed angular rate effects caused by the variation in time space of the background wind and turbulence velocities. The wind shear effects considered are the angular velocities and accelerations for pitch and yaw where roll wind shear effect is zero. The wind angular accelerations are computed by taking numerical time derivatives of the following angular rates due to wind [21].

Roll rate due to wind [rad/s] : $p_w = 0$ **(2.43)**

Pitch rate due to wind [rad/s] : $q_w = \frac{1}{u}\frac{dw_w}{dt}$ **(2.44)**

Yaw rate due to wind [rad/s] : $r_w = \frac{1}{u}\frac{dv_w}{dt}$ **(2.45)**

where $\frac{dw_w}{dt}$ and $\frac{dv_w}{dt}$ stand for the body-fixed y- and z-axes accelerations respectively due to background wind plus turbulence which is totally expressed as “wind”.

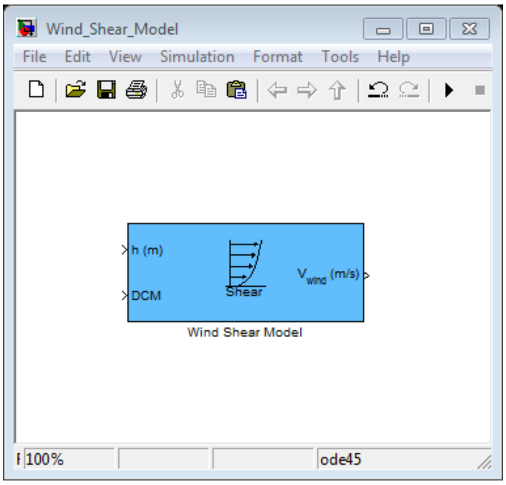


Figure 2.16 : Wind shear model in Simulink.

- **Longitudinal Model**

The following assumptions are made about the aircraft airfoils during symmetric flow,

- Only the wing and the tailplane contribute to the aircraft lift
- Only the wing and the fuselage contribute to the aircraft drag

The fuselage lift in longitudinal model is unusable because of a little smaller than of the related wing.The tailplane lift in longitudinal model also often unusable compared with the that of wing, but it is included in the model because of its significant contribution to the pitching moment coefficient.The tail drag however, is unusable because it is insignificant compared with that of wing and fusulage.The symmetical airflow always distributed symmetrically about the plane of symmetry that the fin produces no lift force and its drag force is unusable of that tailplane.

the longitudinal aerodynamic coefficients can be written as,

$$C_x = -C_D\cos\alpha + C_L\sin\alpha \qquad \textbf{(2.46)}$$

$$C_z = -C_D\sin\alpha - C_L\cos\alpha \qquad \textbf{(2.47)}$$

$$C_m = C_{m0} + (\mathrm{h}-h_0)\, C_{LW} - (\frac{S_T L_T}{S\bar{c}})\, C_{L_T} \qquad \textbf{(2.48)}$$

where,

$$C_L = C_{L_w} + (\frac{S_T}{S})\, \boldsymbol{C_{L_T}} \qquad \textbf{(2.49)}$$

$$C_D = C_{D_w} + (\frac{S_B}{S})\, C_{D_B} \qquad \textbf{(2.50)}$$

And the subscripts W,T and B indicate wing, tailplane and fuselage(body) respectively. C_L and C_D are the lift and drag coefficients of the respective airfoils and C_{m0} is the zero lift.Longitudinal aerodynamic model, expressions for the tailplane and wing and tailplane lift coefficient fusulage drag coefficients (C_{D_w} and C_{D_B}) and tailplane lift coefficients (C_{L_w} and C_{D_w}) must be provided.

$C_{L_w} = f_w\ (\alpha_0+\alpha)$ **(2.51)**

$C_{D_w} = C_{D_{w_o}} + k_w\ C_{L_w}^2$ **(2.52)**

$C_{L_T} = \frac{\delta C_{L_T}}{\delta \alpha}\ [\alpha_T + (1 - \frac{d\varepsilon}{d\alpha})(\ \alpha_T + \alpha)] + \frac{\delta C_{L_T}}{\delta \delta_E}\ \delta_E$ **(2.53)**

$C_{D_B} = C_{D_{B_o}}$ **(2.54)**

where, f_w is non-linear function that describes the wing's relation with the angle of incidence, $C_{D_{w_o}}$ is the wing drag coefficient, k_w is the wing induced drag coeffient, α_T is the tailplane setting angle, $\frac{d\varepsilon}{d\alpha}$ is the downwash angle variation with angle of attack, and $C_{D_{B_o}}$ fusuelage drag coefficient.

- **Lateral Model**

The following assumptions are made about the aircraft airfoils during symmetric flow,

- Only the lift force of fin contributes to the aircraft side force
- Only the wing and fin contribute to the aircraft roll and yaw moments

Non-symmetric flow also induces different of lift across the wing which gives rise to yaw and roll moments with unusable side force.The fuselage effects can be significant beacuse of lateral motion.Meanwhile very difficult to quantity without using complex model techniques.

the lateral aerodynamic coefficients can be written as,

$C_y = (\frac{S_F}{S})\ C_{L_F}\ \cos\beta$ **(2.55)**

$C_l = C_{l_w} - (\frac{S_F h_F}{Sb})\ C_{L_F}\ \cos\beta$ **(2.56)**

$C_n = C_{n_w} + (\frac{S_F h_F}{Sb})\ C_{L_F}\ \cos\beta$ **(2.57)**

where, C_{L_F} is the lift coefficient of the fin and C_{l_w} and C_{n_w} are the roll and yaw moment coefficients of the wing respectively. the lateral aerodynamic model is to obtain expressions for the fin lift coefficients and the wing roll and yaw moment coefficients.

Because of aircraft motion are neglected, lateral coefficients are considered, symmetric conditions.

$C_{L_F} = 0$ **(2.58)**

$C_{l_w} = 0$ **(2.59)**

$C_{n_w} = 0$ **(2.60)**

2.10 Summary

In Section 2, flight dynamics equations of aircraft all motion for a rigid body,

$X_A + X_G + X_T = m\,(\dot{U} + WQ - VR)$ **(2.61)**

$Y_A + Y_G + Y_T = m\,(\dot{V} + UR - WP)$ **(2.62)**

$Z_A + Z_G + Z_T = m\,(\dot{W} + VP - UQ)$ **(2.63)**

$L_A + L_G + L_T = \dot{P}I_x - \dot{R}I_{xz} + QR\,(I_z - I_y) - PQI_{xz}$ **(2.64)**

$M_A + M_G + M_T = \dot{Q}I_y + PR(I_x - I_z) + (P^2 - R^2)\,I_{xz}$ **(2.65)**

$N_A + N_G + N_T = \dot{R}I_z - \dot{P}I_{xz} + PQ\,(I_y - I_x) + QRI_{xz}$ **(2.66)**

$$\begin{bmatrix}\dot{\varphi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix}1 & s\varphi t\theta & c\varphi t\theta\\ 0 & c\varphi & -s\varphi\\ 0 & s\varphi/c\theta & c\varphi/c\theta\end{bmatrix}\begin{bmatrix}P\\ Q\\ R\end{bmatrix} \qquad \textbf{(2.67)}$$

$$\begin{bmatrix}\dot{P}_N\\ \dot{P}_E\\ \dot{P}_D\end{bmatrix} = \begin{bmatrix}c\psi c\theta & -s\psi c\varphi + c\psi s\theta s\varphi & s\psi s\theta + c\psi c\theta s\theta\\ s\psi c\theta & c\psi c\varphi + s\varphi s\theta s\psi & -c\psi s\theta + s\theta s\psi c\varphi\\ -s\theta & c\theta s\varphi & c\varphi c\varphi\end{bmatrix}\begin{bmatrix}U\\ V\\ W\end{bmatrix} \qquad \textbf{(2.68)}$$

DEVELOPMENT OF A LINEAR AIRCRAFT MODEL– LINERIZATION

3.1 Introduction

In Chapter 3, A linearization of aircraft model is developed by converting from non-linear dynamic model of the section 2 to linear dynamic model at a trim condition.In normal flight, the motion an air vehicle as a rigid body can be describes by a set of nonlinear state equations represented as,

$\dot{x}$ = f (x,u) **(3.1)**

where x is the n-dimensional state vector, $\dot{x}$ is derivative of x vector with respect to time, u is control input vector, and f is an nonlinear function.Outputs of the aircraft state can be represented as,

y = h (x,u) **(3.2)**

where, y is a output vector, and h is a nonlinear function expressing the relationship for the outputs in terms of aircraft states and control inputs.The total 12 state equations above mentions variables used in the representation of 6 degrees of freedom rigid body equations of the aircraft motions, and four inputs used.

The classical approach to a stability and control analysis of nonlinear dynamical system is to start with the complete equations of motion that would help to linearize these equations about a specific equilibrium point by a process called trim. The linear system theory can be applied to these linear mathematical models of air vehicle dynamics.

Table 3.1 : State equations of air vehicle dynamics.

$$x=\begin{bmatrix} X_E \\ Y_E \\ Z_E \\ u \\ v \\ w \\ \varphi \\ \theta \\ \psi \\ p \\ q \\ r \end{bmatrix} = \begin{bmatrix} position\ in\ x\ direction\ [m] \\ position\ in\ y\ direction\ [m] \\ altitude [m] \\ velocity\ in\ x\ axis\ [m/s] \\ velocity\ in\ y\ axis\ [m/s] \\ velocity\ in\ z\ axis\ [m/s] \\ bank\ angle\ [rad] \\ pitch\ angle\ [rad] \\ heading\ angle\ [rad] \\ roll\ rate\ [rad/s] \\ pitch\ rate\ [rad/s] \\ yaw\ rate\ [rad/s] \end{bmatrix},$$

$$u=\begin{bmatrix} \delta_{throttle} \\ \delta_{elevator} \\ \delta_{aileron} \\ \delta_{rudder} \end{bmatrix} = \begin{bmatrix} throttle\ deflection \\ elevator\ deflection \\ aileron\ deflection \\ rudder\ deflection \end{bmatrix} \quad \textbf{(3.3)}$$

3.2 Trim

Trim relates to a condition of static equilibrium corresponding to a set of constant controls. Under a trimmed condition, there should not be net moments or forces acting on the center of mass of air vehicle, as a result no changes in the motion variables in time; that's, they all should be zero or constant [11].

$$f(x_n, u_n) = 0 \quad \textbf{(3.4)}$$

correspondingly the nonlinear output equations,

$$y_n = h(x_u, u_u) \quad \textbf{(3.5)}$$

During level flight, u_n are determined which maintain steady state flight ($\dot{x}$) at constant altitude,airspeed, and heading.

where, $\dot{p} = \dot{q} = \dot{r} = \dot{u} = \dot{v} = \dot{w} = 0$ and all control surface inputs are constant or zero. Trim determines the equilbrıım points of the nonlinear model based on the specified altitude, airspeed, center of mass, flight path, angles, etc.

3.2.1 Trim Results

In order to show the flight trimming and linerization results, a flight condition of 29.158 knot and 492.12598 ft altitude is selected as a representative example.The values of the state, state derivatives, and control input vectors obtained from the steady flight trim,

Table 3.2 : Trimmed State

Forces/Moments	Values
Lift Forces, L	53.92 N
Drag Forces, D	2.688 N
Side Forces, FA_y	2.69 N

Trimmed control surface	Values
throttle deflection	0.0142
elevator deflection	0.0164
aileron deflection	0
rudder deflection	0

Angles/Linear - Angular Velocity	Values
bank angle, φ	0
pitch angle, θ	0
heading angle, ψ	0
velocity in x axis, u	0
velocity in y axis, v	0
velocity in z axis, w	0
roll rate, p	0
pitch rate, q	0
yaw rate, r	0

3.3 Small Disturbance Theory

Small disturbance theory is described on aircraft states by converting from non-linear system to linear model in a constant finding trim value and a small deviation from trim state in **Table 3.1** and axial velocity below,

$$U = U_0 + u \qquad \textbf{(3.6)}$$

The trim states and small perturbation variables for linerization need to be defined by equations.The OX_B – axis is relation with the aircraft and initially aligned with the trim velocity vector[11]. A trim velocity of U_0, trim angle of attack of α_0 and trim elevator and throttle actuator settings of δ_{E_0} and δ_{T_0}, the trim contidion can be written as equations below,

$$V_B = U_0 i_B \quad \textbf{(3.7)}$$

$$w_B = 0 \quad \textbf{(3.8)}$$

$$e = 0 \quad \textbf{(3.9)}$$

$$c = (0, \delta_{E_0}, 0, \delta_{T_0}) \quad \textbf{(3.10)}$$

for a small disturbances in angular velocity, linear velocity, control deflection, attitude, shown in (u,v,w), (p,q,r), (ψ, θ, φ) and (δ_a, δ_e, δ_r, δ_t) respectively, then aircraft states become,

$$V_B = (U_0+u)\, i_B + v j_B + w k_B \quad \textbf{(3.11)}$$

$$w_B = p\, i_B + q j_B + r k_B \quad \textbf{(3.12)}$$

$$e = (\psi, \theta, \varphi) \quad \textbf{(3.13)}$$

$$c = (\delta_a, \delta_e, \delta_r, \delta_t) \quad \textbf{(3.14)}$$

under small perturbation assumptions, the angle of attack and angle of sideslip in above equations, become linearly related to the lateral and normal velocity perturations as shown below,

$$\alpha = tan^{-1}\left(\frac{w}{U_0 + u}\right) \approx \frac{w}{U_0} \quad \textbf{(3.15)}$$

$$\beta = sin^{-1}\left(\frac{v}{\sqrt{(U_0 + u)^2 + v^2 + w^2}}\right) \approx \frac{w}{U_0} \quad \textbf{(3.16)}$$

Also moment and force equations can be linearised regarding the trim condition using small disturbance technique.

3.4 Linearization the Equations of Motion using Small Disturbance Technique

v_B , w_B , and e from equations (3.7), (3.8), (3.9) into equations (2.61), (2.62), (2.63), (2.64), (2.65), (2.66), (2.67), (2.68) is substitued and applied the small disturbance technique, gives the linear dynamic equations for the aircraft below.

$$X_A + X_G + X_T = m\,(\dot{u}) \quad \textbf{(3.17)}$$

$$Y_A + Y_G + Y_T = m\,(\dot{v} + cr) \quad \textbf{(3.18)}$$

$$Z_A + Z_G + Z_T = m\,(\dot{w} + U_0 q) \quad \textbf{(3.19)}$$

$L_A + L_G + L_T = \dot{p}I_x - \dot{r}I_{xz}$ **(3.20)**

$M_A + M_G + M_T = \dot{q}I_y$ **(3.21)**

$N_A + N_G + N_T = \dot{r}I_z - \dot{p}I_{xz}$ **(3.22)**

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad \textbf{(3.23)}$$

$$\begin{bmatrix} \dot{P}_N \\ \dot{P}_E \\ \dot{P}_D \end{bmatrix} = \begin{bmatrix} U_0 + u \\ U_0\psi + v \\ -U_0\theta + w \end{bmatrix} \quad \textbf{(3.24)}$$

3.5 Linear Aerodynamic Model

A linear aerodynamic model is built up by carring a Taylor-Maclaurin series expansion of aerodynamic forces about trim condition.The coefficient of linear values in the a Taylor-Maclaurin series are examined in detail in terms of aircraft geometry and airfoil properties.The developed linear dynamic model is used in static airfoil coeffient models [12].

3.5.1 Linearising of aerodynamic forces and moments

Aerodynamic forces opposite the gravitational and engine forces and equilibrium of aircraft maintains in the trim condition. Small disturbances are being considered the Taylar – Maclaurin series expansions can be used to first order in each motion or control variable as shown below for aircraft axial force.

$$X_A = X_{A_0} + (\frac{\partial X_A}{\partial u}u + \frac{\partial X_A}{\partial \dot{u}}\dot{u}) + (\frac{\partial X_A}{\partial v}v + \frac{\partial X_A}{\partial \dot{v}}\dot{v}) + (\frac{\partial X_A}{\partial w}w + \frac{\partial X_A}{\partial \dot{w}}\dot{w})$$

$$+ (\frac{\partial X_A}{\partial p}p + \frac{\partial X_A}{\partial \dot{p}}\dot{p}) + (\frac{\partial X_A}{\partial q}q + \frac{\partial X_A}{\partial \dot{q}}\dot{q}) + (\frac{\partial X_A}{\partial r}r + \frac{\partial X_A}{\partial \dot{r}}\dot{r})$$

$$+ (\frac{\partial X_A}{\partial \delta_a}\delta_a + \frac{\partial X_A}{\partial \dot{\delta_a}}\dot{\delta_a}) + (\frac{\partial X_A}{\partial \delta_e}\delta_e + \frac{\partial X_A}{\partial \dot{\delta_e}}\dot{\delta_e}) + (\frac{\partial X_A}{\partial \delta_r}\delta_r + \frac{\partial X_A}{\partial \dot{\delta_r}}\dot{\delta_r}) \quad \textbf{(3.25)}$$

where X_{A_0} is the axial aerodynamic trim force and trim states $(U_0, \alpha_0, \delta_{E_0}, \delta_{T_0})$ the only state derivate term of any significance is the $\dot{w}$ term and thus equation can be further simplied to,

$$X_A = X_{A_0} + \frac{\partial X_A}{\partial u}u + \frac{\partial X_A}{\partial v}v + \frac{\partial X_A}{\partial w}w + \frac{\partial X_A}{\partial p}p + \frac{\partial X_A}{\partial q}q +$$

$$\frac{\partial X_A}{\partial r}r + \frac{\partial X_A}{\partial \dot{w}}\dot{w} + \frac{\partial X_A}{\partial \delta_e}\delta_e + \frac{\partial X_A}{\partial \delta_a}\delta_a + \frac{\partial X_A}{\partial \delta_r}\delta_r \qquad \textbf{(3.26)}$$

The partial derivatives on the top of line equation 3.25 are described as the aerodynamic stability derivatives.They are called derivatives due to relating a change in motion variable to a change in moment or force.There are five equations which can be found by replacing X with Y, Z, L, M and N.

With the decoupling simplifications, the aerodynamic moments and forces as a result of motion and control variables reduce to,

$$X_A = X_{A_0} + X_{A_0}\mathrm{u} + X_{A_w}\mathrm{w} + X_{A_q}\mathrm{q} + X_{A_{\dot{w}}}\dot{w} + X_{A_{\delta_e}}\delta_e \qquad \textbf{(3.27)}$$

$$Y_A = Y_{A_0} + Y_{A_v}\mathrm{v} + Y_{A_p}\mathrm{p} + Y_{A_r}\mathrm{r} + Y_{A_{\delta_a}}\delta_a + Y_{A_{\delta_r}}\delta_r \qquad \textbf{(3.28)}$$

$$Z_A = Z_{A_0} + Z_{A_u}\mathrm{u} + Z_{A_w}\mathrm{w} + Z_{A_q}\mathrm{q} + Z_{A_{\dot{w}}}\dot{w} + Z_{A_{\delta_e}}\delta_e \qquad \textbf{(3.29)}$$

$$L_A = L_{A_0} + L_{A_v}\mathrm{v} + L_{A_p}\mathrm{p} + L_{A_r}\mathrm{r} + L_{A_{\delta_a}}\delta_a + L_{A_{\delta_r}}\delta_r \qquad \textbf{(3.30)}$$

$$M_A = M_{A_0} + M_{A_u}\mathrm{u} + M_{A_w}\mathrm{w} + M_{A_q}\mathrm{q} + M_{A_{\dot{w}}}\dot{w} + M_{A_{\delta_e}}\delta_e \qquad \textbf{(3.31)}$$

$$N_A = N_{A_0} + N_{A_v}\mathrm{v} + N_{A_p}\mathrm{p} + N_{A_r}\mathrm{r} + N_{A_{\delta_a}}\delta_e + M_{A_{\delta_r}}\delta_r \qquad \textbf{(3.32)}$$

The aerodynamic model of the aircraft has now been reduced to a set of control derivatives and stability.

The dependencies in the equations describing the longitudinal state variables can be written,

$$\Delta \mathrm{X} = \frac{\partial X}{\partial u}u + \frac{\partial X}{\partial w}w + \frac{\partial X}{\partial \delta_e}\delta_e + \frac{\partial X}{\partial \delta_T}\delta_T \qquad \textbf{(3.33)}$$

$$\Delta \mathrm{Z} = \frac{\partial Z}{\partial u}u + \frac{\partial Z}{\partial w}w + \frac{\partial Z}{\partial \dot{w}}\dot{w} + \frac{\partial Z}{\partial q}q + \frac{\partial X}{\partial \delta_e}\delta_e + \frac{\partial X}{\partial \delta_T}\delta_T \qquad \textbf{(3.34)}$$

$$\Delta \mathrm{M} = \frac{\partial M}{\partial u}u + \frac{\partial M}{\partial w}w + \frac{\partial M}{\partial \dot{w}}\dot{w} + \frac{\partial M}{\partial q}q + \frac{\partial M}{\partial \delta_e}\delta_e + \frac{\partial M}{\partial \delta_T}\delta_T \qquad \textbf{(3.35)}$$

the dependencies in the equations describing the lateral/directional state variables can be written,

$$\Delta Y = \frac{\partial Y}{\partial v} v + \frac{\partial Y}{\partial p} p + \frac{\partial Y}{\partial r} r + \frac{\partial Y}{\partial \delta_r} \delta_r \quad \textbf{(3.36)}$$

$$\Delta L = \frac{\partial L}{\partial v} v + \frac{\partial L}{\partial p} p + \frac{\partial L}{\partial r} r + \frac{\partial L}{\partial \delta_r} \delta_r + \frac{\partial L}{\partial \delta_a} \delta_a \quad \textbf{(3.37)}$$

$$\Delta N = \frac{\partial N}{\partial v} v + \frac{\partial N}{\partial p} p + \frac{\partial N}{\partial r} r + \frac{\partial N}{\partial \delta_e} \delta_e + \frac{\partial N}{\partial \delta_a} \delta_a \quad \textbf{(3.38)}$$

3.5.2 Longitudinal stability and control derivatives

The longitudinal control derivatives and stability is provided by perturbation of control or motion variable.

- **Axial velocity derivatives**

A change in axial velocity changes drag and lift due to the airflow speed change over the aerodynamic surfaces. The angle of attack remains almost constant during this perturbation.

Mach effects are ignores which means that $\frac{\partial C_L}{\partial u} = \frac{\partial C_D}{\partial u} = \frac{\partial C_m}{\partial u} = 0$.

$$X_u = \frac{1}{2} \rho U_0{}^2 S\, [-2C_D] \frac{1}{U_0} \quad \textbf{(3.39)}$$

$$Z_u = \frac{1}{2} \rho U_0{}^2 S\, [-2C_L] \frac{1}{U_0} \quad \textbf{(3.40)}$$

$$M_u = 0 \quad \textbf{(3.41)}$$

the dependencies in the equations describing the lateral/directional state variables can be written,

- **Normal velocity derivatives**

A small perturbation in normal velocity is essentially a small perturbation in angle of attack.This changes the drag, lift and pitching moment coefficient of aircraft and uses the following derivatives.

$$X_w = \frac{1}{2} \rho U_0{}^2 S\, [C_L - \frac{\partial C_D}{\partial \alpha}] \frac{1}{U_0} \quad \textbf{(3.42)}$$

$$Z_w = \frac{1}{2} \rho U_0{}^2 S\, [-(\frac{\partial C_L}{\partial \alpha} + C_D)] \frac{1}{U_0} \quad \textbf{(3.43)}$$

$$M_w = \frac{1}{2}\rho {U_0}^2 S\bar{c}\, [\frac{\partial C_m}{\partial \alpha}]\, \frac{1}{U_0} \quad \textbf{(3.44)}$$

- **Pitch rate derivatives**

The pitch rate derivatives are often expressed in terms of a normalized pitch rate and caused by the induced angle of incidence on the tailplane during a pitch rate perturbation.

the induced tailplane incidence is given by $\Delta\alpha_T \approx \frac{qI_T}{U_0}$.

$$X_q = 0 \quad \textbf{(3.45)}$$

$$Z_q = \frac{1}{2}\rho {U_0}^2 S\, [-\frac{S_T l_T}{S\bar{c}} \frac{\partial C_{L_T}}{\partial \alpha}]\, \frac{\bar{c}}{U_0} \quad \textbf{(3.46)}$$

$$M_q = \frac{1}{2}\rho {U_0}^2 S\bar{c}\, [-\frac{S_T l_T}{S\bar{c}} \frac{\partial C_{L_T}}{\partial \alpha}]\, \frac{\bar{c}}{U_0} \quad \textbf{(3.47)}$$

- **Normal acceleration derivatives**

The derivative due to the rate of change of normal velocity perturbation can be quite significant. , can be quantified by $\Delta\alpha_T \approx \frac{d\varepsilon}{d\alpha} \frac{\dot{w}I_T}{{U_0}^2}$. Again, because the tailplane is dragless, only a normal force and pitching moment are produced.

$$X_{\dot{w}} = 0 \quad \textbf{(3.48)}$$

$$Z_{\dot{w}} = \frac{1}{2}\rho {U_0}^2 S\, [-\frac{S_T l_T}{S\bar{c}} \frac{\partial C_{L_T}}{\partial \alpha} \frac{d\varepsilon}{d\alpha}]\, \frac{\bar{c}}{{U_0}^2} \quad \textbf{(3.49)}$$

$$M_{\dot{w}} = \frac{1}{2}\rho {U_0}^2 S\, [-\frac{S_T {l_T}^2}{S\bar{c}^2} \frac{\partial C_{L_T}}{\partial \alpha} \frac{d\varepsilon}{d\alpha}]\, \frac{\bar{c}}{{U_0}^2} \quad \textbf{(3.50)}$$

- **Elevator derivatives**

An elevator disturbance essentially changes the tailplane which produces an aircraft normal force and pitching moment also changes the tailplane angle of incidence.

$$X_{\delta_e} = 0 \quad \textbf{(3.51)}$$

$$Z_{\delta_e} = \frac{1}{2}\rho {U_0}^2 S\, [-\frac{S_T}{S} \frac{\partial C_{L_T}}{\partial \alpha} \frac{d\varepsilon}{d\alpha}] \quad \textbf{(3.52)}$$

$$M_{\delta_e} = \frac{1}{2}\rho {U_0}^2 S\bar{c}\, [-\frac{S_T {l_T}^2}{S\bar{c}^2} \frac{\partial C_{L_T}}{\partial \alpha}] \quad \textbf{(3.53)}$$

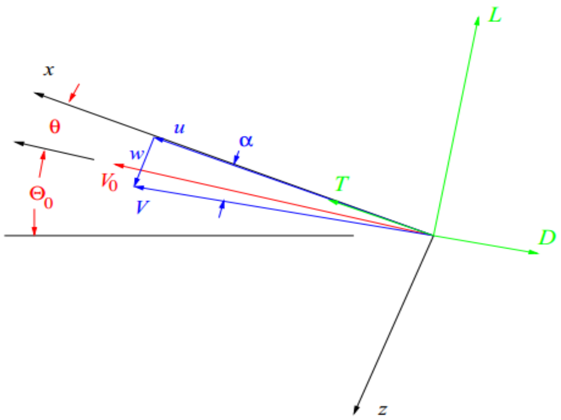


Figure 3.1 : Relation between force components in body longitudinal [21].

3.5.3 Lateral stability and control derivatives

The calculation of each lateral stability and control derivatives starts with the partial differentiation of equations. In order to solve the equations describing lateral/directional vehicle motions, we need to be able to evaluate all the coefficients appearing in Eqs.
For a complete discussion of the lateral derivatives.

- **Sideslip derivatives**

The side force due to sideslip is due primarily to the side force (or "lift") produced by the vertical tail.During a sideslip, the airflow induces an angle of incidence on the fin which creates a lateral force and a yawing moment. The fin can also cause a considerable roll moment because of its aerodynamic centre from the OX_B-axis.

$$Y_v = \frac{1}{2}\rho {U_0}^2 \mathrm{S}\,[-\frac{S_T}{S}\frac{\partial C_{L_F}}{\partial \beta}]\,\frac{1}{U_0} \quad \textbf{(3.54)}$$

$$L_v = \frac{1}{2}\rho {U_0}^2 \mathrm{Sb}[-\frac{2\int_0^S 2C_y y^2 dy}{Sb}.\left(2\partial C_{L_w} \tan \wedge_{1/4} + \Gamma \frac{\partial C_{L_W}}{\partial \alpha}\right) - \frac{S_F h_F}{Sb}\frac{\partial C_{L_F}}{\partial \beta}]\frac{1}{U_0} \quad \textbf{(3.55)}$$

$$N_v = \frac{1}{2}\rho {U_0}^2 \mathrm{Sb}\,[-\frac{S_F l_F}{Sb}\frac{\partial C_{L_F}}{\partial \beta}]\,\frac{1}{U_0} \quad \textbf{(3.56)}$$

- **Roll rate derivatives**

The only part having a more or less significant contribution is the vertical tailplane. Due to this rolling, the vertical tailplane is moving horizontally. A roll rate disturbance alerts an increase in angle of incidence on the down going wing and a decrease in angle of incidence on the up going wing.

$Y_p = 0$ **(3.57)**

$$L_p = \frac{1}{2}\rho U_o^2 \mathrm{Sb}\left[-\frac{2\int_0^S 2C_y y^2 dy}{Sb^2}.\left(\frac{\partial C_{L_W}}{\partial \alpha} + C_{D_w}\right)\right]\frac{b}{U_0} \quad \textbf{(3.58)}$$

$$N_p = \frac{1}{2}\rho U_o^2 \mathrm{Sb}\left[-\frac{2\int_0^S 2C_y y^2 dy}{Sb^2}.\left(C_{L_w} - \frac{\partial C_{D_W}}{\partial \alpha}\right)\right]\frac{b}{U_0} \quad \textbf{(3.59)}$$

- **Yaw rate derivatives**

A yaw rate disturbance increases the airflow velocity over the forward moving wing and decreases the velocity over the aft moving wing. The disturbance also gives rise to an alert angle of incidence on the fin which provides the aircraft's natural yaw damping, roll moment and sideforce.

$Y_r = 0$ **(3.60)**

$$L_r = \frac{1}{2}\rho U_o^2 \mathrm{Sb}\left[\frac{4\int_0^S 2C_y y^2 dy}{Sb^2} C_{L_w} + \frac{S_F l_F h_F}{Sb^2}\frac{\partial C_{L_F}}{\partial \beta}\right]\frac{b}{U_0} \quad \textbf{(3.61)}$$

$$N_r = \frac{1}{2}\rho U_o^2 \mathrm{Sb}\left[-\frac{4\int_0^S 2C_y y^2 dy}{Sb^2} C_{D_w} + \frac{S_F {l_F}^2}{Sb^2}\frac{\partial C_{L_F}}{\partial \beta}\right]\frac{b}{U_0} \quad \textbf{(3.62)}$$

- **Aileron derivatives**

Aileron works accross the wing by differentially angle of indicidence and camber.This effect causes differential drag and lift over the wing span after that result of roll and yaw moments. The side force is considered negligible.on the lateral control derivatives as shown below.

$Y_{\delta_a} = 0$ **(3.63)**

$$L_{\delta_a} = \frac{1}{2}\rho {U_0}^2 \mathrm{Sb}\left[\frac{\partial C_{L_W}}{\partial \delta_a}\right] \quad \textbf{(3.64)}$$

$$N_{\delta_a} = \frac{1}{2}\rho {U_0}^2 \mathrm{Sb}\left[\frac{\partial C_{n_W}}{\partial \delta_a}\right] \quad \textbf{(3.65)}$$

- **Rudder derivatives**

Rudder deflections change the camber of the fin.The main changing result is in the lift force.This causes an aircraft side force and regarding yaw moment and small roll moment.

$$Y_{\delta_r} = \frac{1}{2}\rho U_0{}^2 S\,[\frac{S_F}{S}\frac{\partial C_{L_F}}{\partial \delta_r}] \qquad \textbf{(3.66)}$$

$$L_{\delta_r} = \frac{1}{2}\rho U_0{}^2 Sb\,[\,\frac{S_F h_F}{Sb}\frac{\partial C_{L_F}}{\partial \delta_r}] \qquad \textbf{(3.67)}$$

$$N_{\delta_r} = \frac{1}{2}\rho U_0{}^2 Sb\,[\,-\frac{S_F h_F}{Sb}\frac{\partial C_{L_F}}{\partial \delta_r}] \qquad \textbf{(3.68)}$$

3.5.4 Airfoil aerodynamic coefficients

The analysis of stability derivative has provided into aerodynamic moments and forces provided aircraft motion variables.The extended longitudinal and lateral coefficient models are listed below.

- **Longitudinal Airfoil Coefficients**

$$C_{l_W} = f_W(\alpha_0+\alpha) \qquad \textbf{(3.69)}$$

$$C_{D_W} = C_{D_{W_o}} + k_W c_{L_W}^2 \qquad \textbf{(3.70)}$$

$$C_{L_T} = \frac{\partial C_{L_T}}{\partial \alpha}\,[\alpha_T + (1+\frac{d\varepsilon}{d\alpha})\,(\alpha_0+\alpha) + \frac{QI_T}{\bar{V}} + \frac{\dot{\alpha} I_T}{\bar{V}}\frac{d\varepsilon}{d\alpha}\,] + \frac{\partial C_{L_T}}{\partial \delta_E}\,\delta_E \qquad \textbf{(3.71)}$$

$$C_{D_B} = C_{D_{B_o}} \qquad \textbf{(3.72)}$$

where, linear terms that describe the induced angle of attack due to pitch rate have been added to the tailplane coefficient model.

- **Lateral Airfoil Coefficients**

$$C_{L_F} = \frac{\partial C_{L_F}}{\partial \beta}\,(\beta + \frac{RI_T}{\bar{V}}) + \frac{\partial C_{L_F}}{\partial \delta_R}\,\partial_R \qquad \textbf{(3.73)}$$

$$C_{l_W} = -2\int_0^S [(2\,C_{l_W}\tan\Delta_{1/4} + \Gamma\frac{\partial C_{L_W}}{\partial \alpha})\beta + (\frac{\partial C_{L_W}}{\partial \alpha} + C_{D_W})\,\frac{P_T}{\bar{V}}) - 2C_{L_W}\frac{R_y}{\bar{V}}\,]\,\frac{c_y y d_y}{Sb} + \frac{\partial C_{l_W}}{\partial \delta_A} \qquad \textbf{(3.74)}$$

$$C_{n_W} = 2\int_0^S [(2\,C_{l_W} - \frac{\partial C_{D_W}}{\partial \alpha})\,\frac{P_y}{\bar{V}} + 2C_{D_W}\frac{R_y}{\bar{V}}]\,\frac{c_y y d_y}{Sb} + \frac{\partial C_{n_W}}{\partial \delta_A}\,\delta_A \qquad \textbf{(3.75)}$$

where, linear terms that describe the induced angle of incidence on the fin during sideslip and yaw rate motions changing in fin lift coefficient due to the rudder deflection.

where, linear terms that describe the wing’s yaw and roll moments due to sidedlip, roll rate and yaw rate motions as well as the aileron deflection.

- **Summary**

The longitudinal and lateral airfoil coefficient models of equations above, complete the aerodynamic model of section 3. The model is stable with the stability and control derivatives displayed in Section 3.5.2 and 3.5.3.

- The tailplane, wing and fin lift curve slopes as a function of their incidence angles.
- The wing yaw and roll moment change with aileron deflection
- The wing and fuselage drag coefficients
- The wing induced downwash variation with angle of attack and drag coefficient.

3.6 Linear Gravitational Model

In the trim condition and applying small perturbation assumptions, the linearised gravitational model becomes,

$$\begin{bmatrix} X_G \\ Y_G \\ Z_G \end{bmatrix} = \begin{bmatrix} -mg\theta \\ mg\varphi \\ mg \end{bmatrix} \quad \textbf{(3.76)}$$

$$L_G = M_G = N_G = 0 \quad \textbf{(3.77)}$$

3.7 Engine Model

The formulation of the linear aerodynamic model, also needed to the engine thrust formula can be written as the Taylor Maclaurin series, in the aircraft motion and control variables, about the trim condition.

The linear engine thrust model reduces to,

$$\mathrm{T} = T_0 + \frac{\partial T}{\partial u}\mathrm{u} + \frac{\partial T}{\partial \delta_t}\delta_t \quad \textbf{(3.78)}$$

where, T_0 engine trim trust that's a function of trim states (U_0, α_0, δ_{E_0}, δ_{T_0}), $\frac{\partial T}{\partial u}$ is the engine velocity derivative and $\frac{\partial T}{\partial \delta_t}$ is the engine control derivative.The engine control derivative can be reduced.

$$X_T = (T_O + T_U\mathrm{u} + \delta_t)\cos\varepsilon_T \quad \textbf{(3.79)}$$

$$Z_T = -(T_O + T_U\mathrm{u} + \delta_t)\sin\varepsilon_T \quad \textbf{(3.80)}$$

$$M_T = -(T_O + T_U\mathrm{u} + \delta_t)\, m_T \quad \textbf{(3.81)}$$

$$Y_T = L_T = N_T = 0 \quad \textbf{(3.82)}$$

Also, cos ε_T, sin ε_T and m_T terms are combined with the thrust variables as below,

$$X_T = X_{T_O} + X_{T_U}\text{u} + X_{T_{\delta_t}}\delta_t \tag{3.83}$$

$$Z_T = Z_{T_O} + Z_{T_U}\text{u} + Z_{T_{\delta_t}}\delta_t \tag{3.84}$$

$$M_T = M_{T_O} + M_{T_U}\text{u} + M_{T_{\delta_t}}\delta_t \tag{3.85}$$

$$Y_T = L_T = N_T = 0 \tag{3.86}$$

where, as an example,

$$X_{T_O} = T_O \cos \varepsilon_T \tag{3.87}$$

$$\dot{\delta}_t = \frac{T_0}{\tau_T}\delta_t + \frac{T_0}{\tau_T}\delta_{t_c} \tag{3.88}$$

where, τ_T is the engine time constant and δ_{t_c} is the commanded throttle perturbation which can be determined.

3.8 Linear Aircraft Model

The short term local behavior of the air vehicle at a given flight condition can be approximated by the linearization of its nonlinear model about the equilibrium points obtained by trimming. The standard linear state equations or the state space representation for a linear differential system has the form, substituting the respective force and moment terms into equations gives,

$$X_{A_u}u + X_{A_w}w + X_{A_q}q + X_{A_{\dot{w}}}\dot{w} + X_{A_{\delta_e}}\delta_e - \text{mg}\theta + X_{T_u}u + X_{T_{\delta_t}}\delta_t = \text{m}\dot{u} \tag{3.89}$$

$$Y_{A_v}v + Y_{A_p}p + Y_{A_r}r + Y_{A_{\delta_a}}\delta_a + Y_{A_{\delta_r}}\delta_r - \text{mg}\varphi = \text{m}(\dot{v} + U_0 r) \tag{3.90}$$

$$Z_{A_u}u + Z_{A_w}w + Z_{A_q}q + Z_{A_{\dot{w}}}\dot{w} + Z_{A_{\delta_e}}\delta_e + Z_{T_u}u + Z_{T_{\delta_t}}\delta_t = \text{m}(\dot{w} - U_0 q) \tag{3.91}$$

$$L_{A_v}v + L_{A_p}p + L_{A_r}r + L_{A_{\delta_a}}\delta_a + L_{A_{\delta_r}}\delta_r = \dot{p}I_x - \dot{r}I_{xz} \tag{3.92}$$

$$M_{A_u}u + M_{A_w}w + M_{A_q}q + M_{A_{\dot{w}}}\dot{w} + M_{A_{\delta_a}}\delta_a + M_{T_u}u + M_{A_{\delta_t}}\delta_t = \dot{q}I_y \tag{3.93}$$

$$N_{A_v}v + N_{A_p}p + N_{A_r}r + N_{A_{\delta_a}}\delta_a + N_{A_{\delta_r}}\delta_r = \dot{r}I_z - \dot{p}I_{xz} \tag{3.94}$$

where, giving the trim condition equations are related with aerodynamic, gravitational and engine forces and moments cancel in the steady state.

X_{A_O} = - X_{T_O} **(3.95)**

Z_{A_O} = - (X_{T_O}+ mg) **(3.96)**

M_{A_O} = - M_{T_O} **(3.97)**

$Y_{A_O} = L_{T_O} = N_{T_O} = 0$ **(3.98)**

Above equations form a set of three equations in the four trim variables (U_O , α_O , δ_{E_0}, δ_{T_0})which, for a completely linear model, means that only a single trim variable need be defined to fully define a level flight trim state.

Equations of aircraft motions are classified with equations and divided into a longitudinal and lateral set.

3.8.1 Longitudinal dynamic equations

The dynamic of the aircraft can be represented by this state space equation:

$\dot{x} = A.x + B.u$ **(3.99)**

Where x is called the state vector and u the control vector

$M_l \dot{x}_l = A'_l x_l + B'_l u_l$ **(3.100)**

$\dot{\delta}_t = \frac{T_0}{\tau_T}\delta_t + \frac{T_0}{\tau_T}\delta_{t_c}$ **(3.101)**

$\dot{h}$ = - $\dot{P}_D = U_O\,\theta - w$ **(3.102)**

$\dot{p_N} = \dot{P}_N$ - $U_O = u$ **(3.103)**

where, and h is the height state and p_N is the longitudinal position error state.The longitudinal state space vector (x_l) is only made up of the core longitudinal states.

$x_l = [\,u \;\; w \;\; q \;\; \theta\,]^T$ **(3.104)**

$u_l = [\,\delta_e \quad \delta_t\,]^T$ **(3.105)**

$$M_l = \begin{bmatrix} m & -X_{\dot{w}} & 0 & 0 \\ 0 & (m - Z_{\dot{w}}) & 0 & 0 \\ 0 & M_{\dot{w}} & I_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$ **(3.106)**

$$A_l' = \begin{bmatrix} (X_u + X_{T_u}) & -X_w & X_q & -mg \\ (Z_u + Z_{T_u}) & Z_w & Z_q + m\,U_O & 0 \\ (M_u + M_{T_u}) & M_w & M_q & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \textbf{(3.107)}$$

$$B_l' = \begin{bmatrix} \mathrm{X}_{\mathrm{A}_{\delta_e}} & \mathrm{X}_{\mathrm{T}_{\delta_T}} \\ \mathrm{Z}_{\mathrm{A}_{\delta_e}} & \mathrm{Z}_{\mathrm{T}_{\delta_T}} \\ \mathrm{M}_{\mathrm{A}_{\delta_e}} & \mathrm{M}_{\mathrm{T}_{\delta_T}} \\ 0 & 0 \end{bmatrix} \quad \textbf{(3.108)}$$

3.8.2 Lateral Dynamic Equations

The dynamic of the aircraft can be represented by this state space equation:

$$M_t\,\dot{x}_t = A_t' x_t + B_t' u_t \quad \textbf{(3.109)}$$

$$\dot{\psi} = \mathrm{r} \quad \textbf{(3.110)}$$

$$y = \dot{P}_E = U_O \psi + v \quad \textbf{(3.111)}$$

$$x_t = [\ v \ \ p \ \ r \ \ \varphi\ ]^T \quad \textbf{(3.112)}$$

$$u_t = [\ \delta_a \quad \delta_r\ ]^T \quad \textbf{(3.113)}$$

$$M_l = \begin{bmatrix} m & 0 & 0 & 0 \\ 0 & I_x & -I_{xz} & 0 \\ 0 & -I_{xz} & I_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \textbf{(3.114)}$$

$$A_t' = \begin{bmatrix} Y_v & Y_p & (Y_p - mU_O) & mg \\ L_v & L_p & L_r & 0 \\ N_v & N_p & N_r & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \textbf{(3.115)}$$

$$B_t' = \begin{bmatrix} \mathrm{Y}_{\mathrm{A}_{\delta_a}} & \mathrm{Y}_{\mathrm{A}_{\delta_r}} \\ \mathrm{L}_{\mathrm{A}_{\delta_a}} & \mathrm{L}_{\mathrm{A}_{\delta_r}} \\ \mathrm{N}_{\mathrm{A}_{\delta_a}} & \mathrm{N}_{\mathrm{A}_{\delta_r}} \\ 0 & 0 \end{bmatrix} \quad \textbf{(3.116)}$$

And y is the cross track error state.The lateral state space vector (x_t) is only made up of the core lateral states.

3.9 Summary

In Section 3 aircraft linear model about a level flight trim condition was developed.The force, moment equations and aircraft dynamic equations was found using small disturbance theory. Finally, the model was decoupled into longitudinal and lateral systems and presented in a form one matrix inversion away from state space form.

4. AIRCRAFT NATURAL MOTIONS

Aircraft natural motion is now described to the anlaysis of the linear model to the aircraft used in this thesis. The Piper J-3 ¼ scale trainer as shown in **Figure 4.1**.Calculating of the linear model for an aircraft process was written and developed Matlab/Simulink Program.Also parameters accepts an aircraft's pysical parameters and condition of the trim. For a trim condition of 29.158 knot is mentioned in the thesis.

Figure 4.1: Piper J-3 ¼ Cub.

In the Chapter 4, the longitudinal and lateral aircraft model are presented and analysed in details all aspects area using model decomposition method which represented briefly in Appendix A.Modal decomposition involves using an eigenvector transformation matrix to convert the current system to modal form.

4.1 Longitudinal Aircraft Model

As shown in **Figure 4.2** a block diagram represents model of longitudinal aircraft with equations.

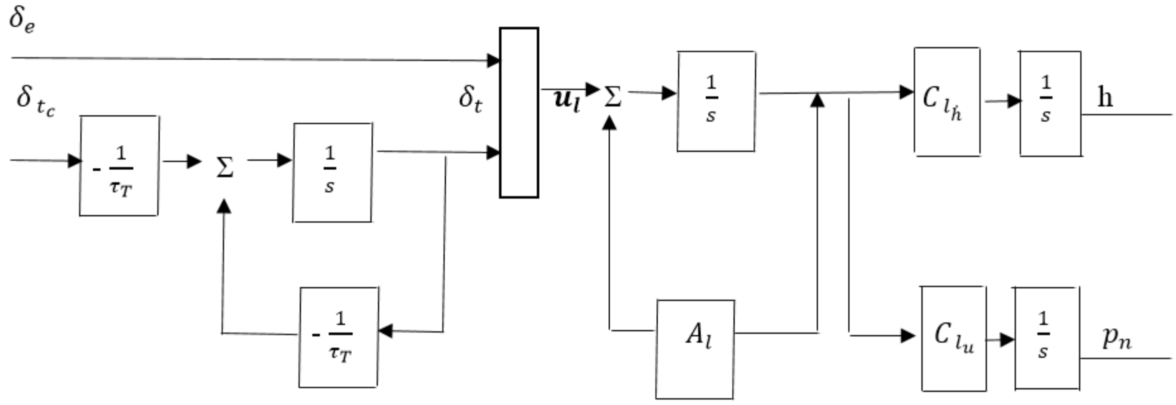


Figure 4.2 Longitudinal aircraft model.

The longitudinal model can be written as,

$$\dot{x}_l = A_l x_l + B_l u_l \tag{4.1}$$

$$\dot{h}_n = C_{l_{\dot{h}}} x_l \tag{4.2}$$

$$\dot{p} = C_{l_u} x_l \tag{4.3}$$

$$\dot{\delta}_t = -\frac{1}{\tau_T}\delta_t + \frac{1}{\tau_T}\delta_{t_c} \tag{4.4}$$

The state vector (x_l) and control input vector (u_l) are defined in above equations . The units of u and w are in metres per second (m/s), h and p_N in metres (m), q in tens of degrees per second (10deg/s), θ and δ_e in degrees (deg) and δ_t and δ_{t_c} in specific force (m/s^2).

The maximum deviations of the variables in the longitudinal state vector are discussed in Table 4.1 and the maximum allowed control deflections are listed in **Table 4.2**.

Table 4.1 : Longitudinal state vector maximum deviations.

Variation	Description	Max	Unit
u	With a trim velocity of 20 m/s, it is estimated that the maximum deviation from this trim condition will be approximately 15 m/s.	15	m/s
w	At a trim velocity of 20 m/s, a normal velocity disturbance of 5 m/s will produce an angle of incidence of approximately 15 deg and stall the aircraft.	5	m/s
q	The maximum pitch rate disturbance during conventional flight is estimated as 60 deg/s	6	10 deg/s
θ	The maximum pitch angle deviation during conventional flight is estimated to be 15 deg.	15	deg

Table 4.2 : Longitudinal control state vector maximum deviations.

Variation	Description	Max	Unit
δ_e	The maximum elevator deflection is limited to 10 deg	10	m/s
δ_t	From engine test date, the maximum specific force deviation is set at 15m/s^2	15	m/s^2

This is done using model decomposition techniques method.The modal transformation matrix that converts the system of equation to modal form is as shown below.

$$\begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} = \begin{bmatrix} 0.59026 & 10.76107 & 0.03533 & -9.60432 \\ -0.67759 & -8.97706 & 1.48736 & 0.43201 \\ -4.89487 & -88.04203 & -21.23939 & -1.63890 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} z_{l_1} \\ z_{l_2} \\ z_{l_3} \\ z_{l_4} \end{bmatrix} \tag{4.5}$$

where the state vector which called x is shown on the left and the corresponding modal state vector which called u on the right. The modal transformation matrix has the real and imaginary parts of each eigenvector as its columns.Transforming the longitudinal state space model of equation (**4.1**) reveals,

$$\dot{\boldsymbol{z}}_{\boldsymbol{l}} = F_l z_l + G u_l \qquad \textbf{(4.6)}$$

where,

$$\boldsymbol{z_l} = [\, z_{l_1} \;\; z_{l_2} \;\; z_{l_3} \;\; z_{l_4} \,]^T \qquad \textbf{(4.7)}$$

$$\boldsymbol{F_l} = \begin{bmatrix} -14.76 & 9.56 & 0 & 0 \\ -9.56 & -14.76 & 0 & 0 \\ 0 & 0 & -0.0531 & 0.7048 \\ 0 & 0 & -0.7048 & -0.0531 \end{bmatrix} \qquad \textbf{(4.8)}$$

$$\boldsymbol{G_l} = \begin{bmatrix} -152.24309 & 3.13927 \\ -9.56 & 1.66828 \\ -10.4463 & 3.72794 \\ -4.67735 & 0.59839 \end{bmatrix} \qquad \textbf{(4.9)}$$

$\boldsymbol{F_l}$ for modal form in system matrix shows the complex poles that define the natural modes of the primary longitudinal system.The higher frequency pole set is called as the short period mode, the lower frequency set is called is as the phugoid(long period) mode.

4.1.1 Phugoid mode

The phugoid mode poles are summarised below,

$\lambda = -0.0531 \pm 0.7048i$

$w_n = 0.0297$ rad/s

$\zeta = 0.113$

These poles are a stable, low frequency, lightly damped oscillatory mode.

The eigenvector corresponding to the phugoid is displayes in phasor form below,

$$\begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} = \begin{bmatrix} 1.0000 \angle 0.00 \\ 0.0673 \angle 178.74 \\ 0.0507 \angle -2.62 \\ 0.0720 \angle 186.9 \end{bmatrix} \qquad \textbf{(4.10)}$$

All of phasor angles are described in degrees.The eigenvector are described the magnitude and phase contributions of each element of state vector (x_l).Also it is summarized that phugoid mode is dominated by pitch angle and axial velocity perturbations with small changes in pitch rate and normal velocity.Under small changes, thr low frequency of the normal velocity is equal to observing little angle of attack(α) change. Typically, the lift-to-drag ratio is far more critical to an airplane's performance than the phugoid damping. However, in the case that the damping must be increased, Equations clearly show that the only way to accomplish is to increase the air vehicle drag and therefore decrease the lift to drag ratio.

The results of the dynamic stability analyses for phugoid mode damping ratio values with respect to airspeed and altitude in the operational flight envelope are displayed in **Figure 4.3**.

Response of pure long term mode motion, obtained by initialising the system state vector (x) with finding the real parts of phugoid mode eigenvector, as shown in **Figure 4.2**, The long term mode motion is well observed in axial velocity and pitch angle.

An extarct of the longitudinal modal input matrix ($\boldsymbol{G_l}$) of equation, pertinent to the phugoid mode, is shown below,

$$\boldsymbol{G_{l_{ph}}} = \begin{bmatrix} -10.4463 & 3.72794 \\ -4.67735 & 0.59839 \end{bmatrix} \qquad \textbf{(4.11)}$$

The magnitude of the first column vector of $G_{l_{ph}}$ describes the control gain form the elevator (δ_e) to the long term mode states. Similarly, the magnitude of the second column vector of $G_{l_{ph}}$ describes the control gain from the throttle (δ_{th}) to the long term mode states.

Calculating the relative magnitudes of the scaled columns of $G_{l_{ph}}$ gives,

$$\frac{Elv}{Thr} = \frac{\sqrt{(-10.4463)^2 + (-4.67735)^2}}{\sqrt{(3.72794)^2 + (0.59839)^2}} = 9.19$$

Finding result indicates elevator is about 9 times more effective at controlling long term mode than throttle.

Variables are small deviations from operating point or trim conditions state:

Disturbance inputs :

u_w : velocity of wind along axis

v_w : velocity of wind perpendicular to body axis

Control inputs :

δ_e : elevator angle (δ_e>0 is down)

δ_t : thrust

$y(t) = Ce^{tA}\ (Rx_{phugoid})$ – Phugoid mode motion

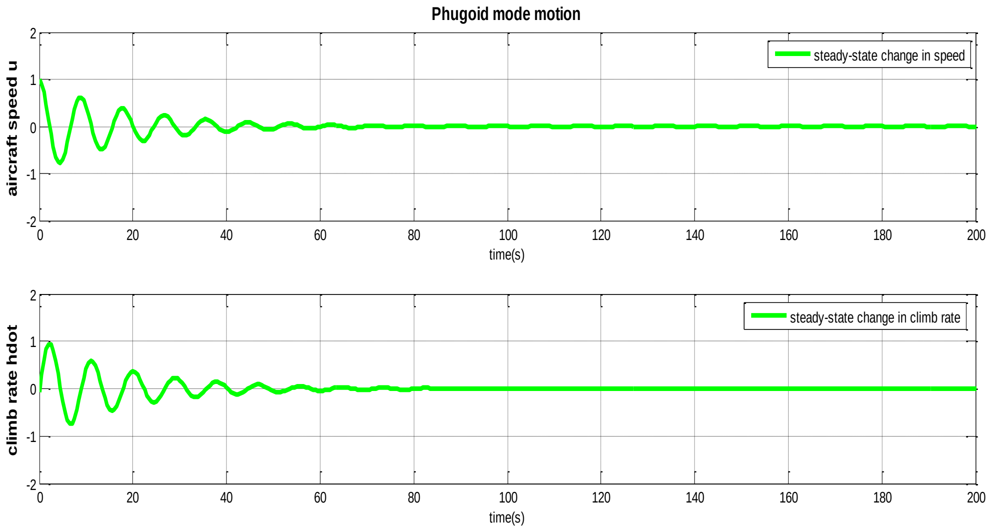


Figure 4.3 : Phugoid mode.

- Affects both speed and climb rate
- Period ≈ 10 sec ; decays in ≈ 80 sec

4.1.2 Short period mode

The short mode poles are summarised below,

$\lambda = -14.76 \pm 9.56i$

w_n = 17,6 rad/s

$\zeta = 0.0084$

These poles form a stable, high frequency, well damped oscillatory mode. The period of oscillation is about 0.2s with an exponential decay time constant of just under 0.4s. Eigenvector of the short period mode is shown in phasor form below.

$$\begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} = \begin{bmatrix} 0.0312 \measuredangle -132.49 \\ 0.7965 \measuredangle 0.0000 \\ 0.6028 \measuredangle 122.90 \\ 0.0342 \measuredangle -24.067 \end{bmatrix} \qquad \textbf{(4.12)}$$

The eigenvector describes the mode is observed in pitch rate (q) with smaller regarding changes in pitch angle (θ) and angle of attack (α).

By the pitching moment due to normal velocity (u) disturbance derivative (M_w) effected short mode period.This effect is linked with by pitch rate (q) damping stability derivative (M_q).The natural damping of the short period mode is so perfect.

Input matrix of the longitudinal modal (G_l) of equation, about to the short period mode, is shown below,

$$G_{l_{sh}} = \begin{bmatrix} -152.24309 & 3.13927 \\ -9.56 & 1.66828 \end{bmatrix} \tag{4.13}$$

Calculating the relative magnitudes of the scaled columns of $G_{l_{sh}}$ gives,

$$\frac{Elv}{Thr} = \frac{\sqrt{(-152.24309)^2 + (-9.56)^2}}{\sqrt{(3.13927)^2 + (1.66828)^2}} = 42.09$$

Finding result indicates elevator is about 42 times more effective at controlling short term mode than throttle.

y(t) = Ce^{tA} (Rx_{short}) – Short period mode motion

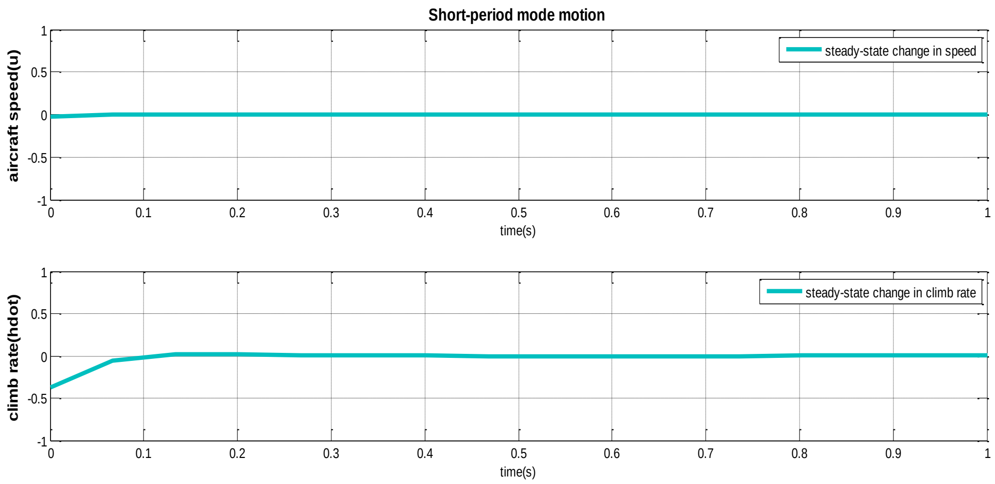


Figure 4.4 : Short period mode.

- Affects only small effect on speed u
- Period ≈ 0.1 sec ; decays in ≈ under 0.4 sec

4.1.3 Dynamic response to wind

Impulse response matrix from (u_n, v_n) to $(u, \dot{h})$, over time period [0,20]

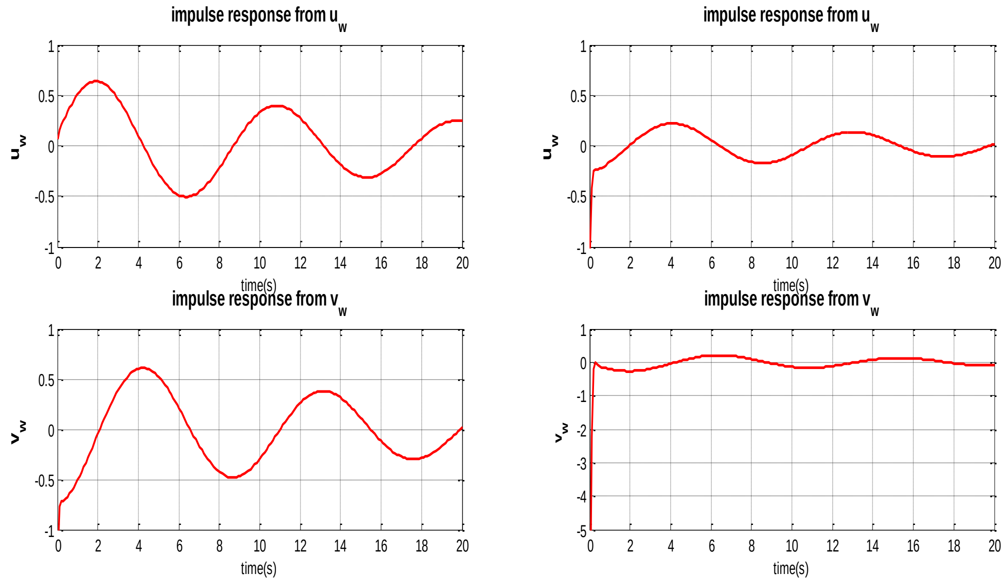


Figure 4.5: Response to wind.

4.1.4 Dynamic response to actuators

Impulse response matrix from (δ_e, δ_t) to $(u, \dot{h})$, over time period [0,20].

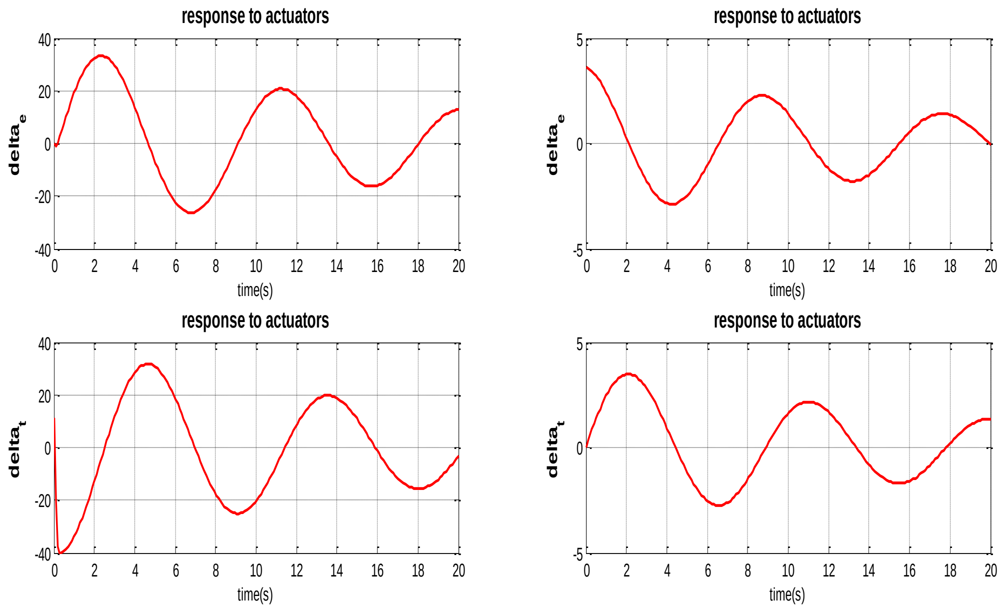


Figure 4.6 : Response to actuator.

4.2 Lateral Aircraft Model

Figure below shows a block diagram representation of the lateral model described by equations.

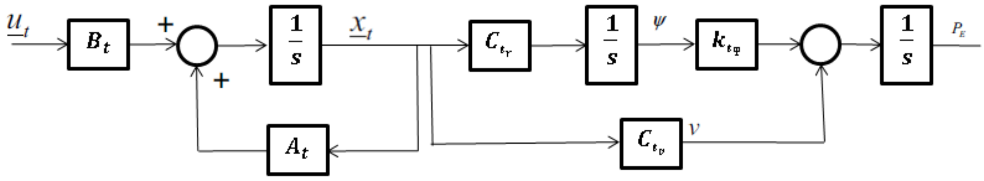


Figure 4.7 : Lateral aircraft model.

$$\dot{x}_t = A_t x_t + B_t u_t \quad \textbf{(4.14)}$$

$$\dot{\psi} = C_{t_r}\, x_t \quad \textbf{(4.15)}$$

$$\dot{y} = k_t\, \psi + C_{t_v} x_t \quad \textbf{(4.16)}$$

The state vector (x_t) and control input vector (u_t) are defined in equations above. The units of *v* are in metres per second (m/s), y in metres (m), p and r in tens of degrees per second (10deg/s) and φ, ψ, δ_a and δ_r in degrees (deg). . The maximum deviations of the variables in the lateral state vector are discussed in **Table 4.3** and the maximum allowed control deflections are listed below.

The modal transformation matrix that converts the system of equation to modal form is shown below,

$$\begin{bmatrix} v \\ p \\ r \\ \varphi \end{bmatrix} = \begin{bmatrix} -0.34527 & 0.00603 & -0.96795 & -0.65662 \\ -4.36674 & -18.85181 & 1.73613 & -0.00197 \\ 27.09295 & -0.61368 & -2.75322 & -0.02964 \\ 0.00313 & 1.00237 & 0.00042 & 0.04561 \end{bmatrix} \begin{bmatrix} z_{l_1} \\ z_{l_2} \\ z_{l_3} \\ z_{l_4} \end{bmatrix} \quad \textbf{(4.17)}$$

Table 4.3 : Lateral state vector maximum deviations.

Variation	Description	Max	Unit
v	At trim velocity of 20m/s, a lateral velocity of 5m/s causes approximately 15deg of sideslip. This is considered the maximum angle of sideslip	5	m/s
p	The maximum roll rate disturbance during conventional flight is estimated as 90deg/s.	9	10 deg/s
r	The maximum yaw rate disturbance during conventional flight is estimated as 60 deg/s	6	10 deg/s
φ	The maximum roll angle deviation during conventional flight is estimated to be 15 deg.	15	deg

Table 4.4 : Lateral control vector maximum deflections.

Variation	Description	Max	Unit
δ_a	The maximum aileron deflection is limited to 1 deg	1	deg
δ_r	The maximum rudder deflection is limited to 1 deg	0.5	m/s^2

where the original state vector is shown on the left and the corresponding modal state vector on the right. Transforming the lateral state space model of equation reveals,

$$\dot{\boldsymbol{z}}_{\boldsymbol{t}} = F_t z_t + G_t u_t \quad \textbf{(4.18)}$$

where,

$$\boldsymbol{z}_{\boldsymbol{l}} = [\ z_{l_1}\ \ z_{l_2}\ \ z_{l_3}\ \ z_{l_4}\ ]^T \quad \textbf{(4.19)}$$

$$F_l = \begin{bmatrix} -18.784 & 0 & 0 & 0 \\ 0 & -1.56039 & 4.9626 & 0 \\ 0 & -4.9626 & -1.5604 & 0 \\ 0 & 0 & 0 & -0.04503 \end{bmatrix} \quad \textbf{(4.20)}$$

$$G_l = \begin{bmatrix} -137.79875 & -3.24447 \\ -4.59723 & -23.07731 \\ 4.03361 & 7.21558 \\ 8.72497 & -0.15338 \end{bmatrix} \quad \textbf{(4.21)}$$

The lateral system matrix for modal form (F_t), shows that the two real poles and the set of complex poles that define lateral system.

Dutch roll mode is known as complex polex poles.While spiral mode is known as real pole in low frequency, roll mode is known as real pole in high frequency.

4.2.1 Roll mode

λ = -18.784

This corresponds to a very fast, first order exponential response with a time constant of 0.

$$\begin{bmatrix} v \\ p \\ r \\ \varphi \end{bmatrix} = \begin{bmatrix} -0.34527 \angle 0.000 \\ -4.36674 \angle 0.000 \\ 27.09295 \angle 0.000 \\ 0.00313 \angle 180 \end{bmatrix} \quad \textbf{(4.22)}$$

The roll made can be made by an atmospheric disturbance or an aileron input.The motion can be approximated by a single degree of freedom rolling motion.The magnitude of the roll damping L_p is dependent on the size of the wing and tail surfaces. The lateral velocity and yaw rate perturbations are negligible during this motion.

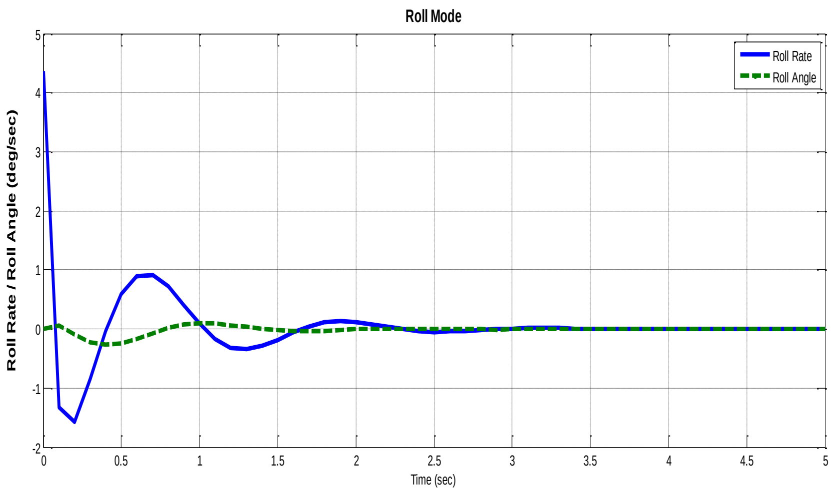


Figure 4.8 : Roll mode.

The eigenvector indicates that the roll mode is strongly observed in roll rate with a smaller corresponding amount of roll angle. The lateral velocity and yaw rate perturbations are negligible during this motion.

An extract of the lateral modal input matrix (G_t) of equation (**4.26**), about to the roll mode, is shown below,

$$G_{t_r} = [-137.79875 \ -3.24447\ ] \qquad \textbf{(4.23)}$$

With reference to Table above, the maximum deflections of both the aileron and rudder are set at 1^0. This means that the relative magnitudes of the elements of G_{t_r} can be directly compared without the need for scaling. Comparing the column magnitudes of G_{t_r} reveals,

$$\frac{\text{Ail}}{\text{Rud}} = \frac{|-137.79875|}{|-3.24447|} = 42.47$$

Finding result indicates aileron is about 42 times more effective at controlling roll mode than rudder.

4.2.2 Spiral mode

The spiral mode pole is at,

$\lambda = -0.04503$

This corresponds to an extremely slow, first order exponential response with a time constant of 450s.

$$\begin{bmatrix} v \\ p \\ r \\ \varphi \end{bmatrix} = \begin{bmatrix} -0.65662 \angle 0.000 \\ -0.00197 \angle 180 \\ -0.02964 \angle 0.000 \\ 0.04561 \angle 0.000 \end{bmatrix} \qquad \textbf{(4.24)}$$

The spiral mode consists of a slow rolling/yawing motion for which the sideslip is relatively small.The roll rate is quite small compared to the yaw rate

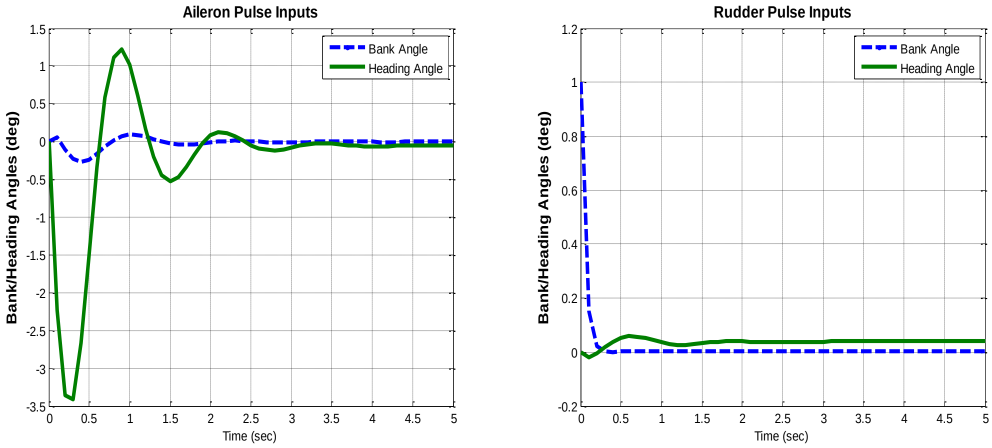


Figure 4.9 : Spiral Mode.

The eigenvector indicates that the spiral mode is primarily observed in bank angle. With such a long time constant, the bank angle will translate into a slow turn which will make the spiral mode very well observed in heading angle.

An extract of the lateral modal input matrix (G_t) of equation, pertinent to the spiral mode, is shown below,

$$G_{t_r} = [8.72497 - 0.15338] \quad \textbf{(4.25)}$$

$$\frac{\text{Ail}}{\text{Rud}} = \frac{|8.72497|}{|-0.15338|} = 56.8847$$

Finding result indicates aileron is about 56 times more effective at controlling spiral mode than rudder.

4.2.3 Dutch roll mode

The spiral mode pole is at,

$\lambda = -1.56039 \pm 4.96256i$

$w_{n_{dr}}$: 5.21 rad/s

ζ_{DR} : 0.302

These poles form a stable, lightly damped oscillatory mode. The period of oscillation is 0.6 s with an exponential decay time constant of 0.45s. The dutch roll mode eigenvector is displayed in phasor form below,

$$\begin{bmatrix} v \\ p \\ r \\ \varphi \end{bmatrix} = \begin{bmatrix} 0.968 \measuredangle -0.4 \\ 18.9315 \measuredangle 95.26 \\ 2.8208 \measuredangle -102.57 \\ 1.0023 \measuredangle 0.000 \end{bmatrix} \quad \textbf{(4.26)}$$

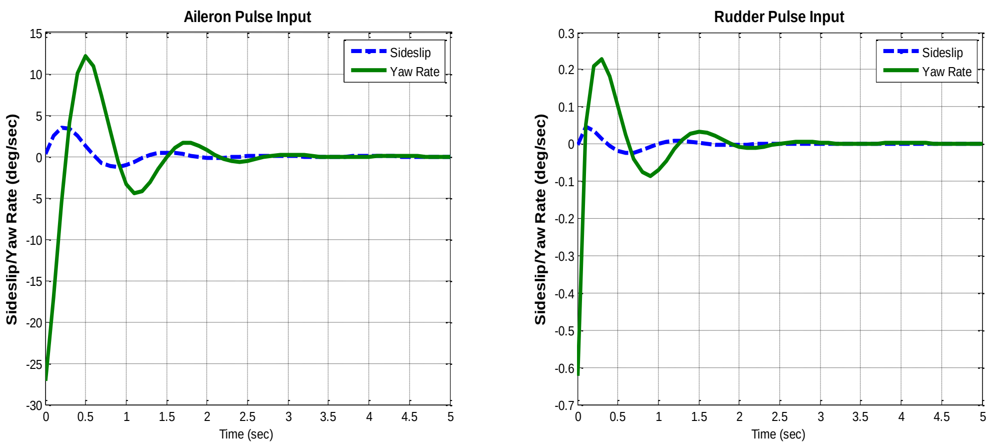


Figure 4.10 : Dutch roll mode.

The eigenvector indicates that the dutch roll mode is observed well in all of the lateral states but predominantly in yaw rate. Note that the yaw rate motion leads the roll rate motion by approximately 90^0 which is a classic feature of this mode.

An extract of the lateral modal input matrix G_t of equation, pertinent to the dutch roll mode, is shown below,

$$G_{t_r} = \begin{bmatrix} -4.59723 & -23.07731 \\ 4.03361 & 7.21558 \end{bmatrix} \qquad \textbf{(4.27)}$$

The relative magnitudes of the columns of $G_{l_{dr}}$ reveal,

$$\frac{\text{Ail}}{\text{Rud}} = \frac{\sqrt{(-4.59723)^2 + (4.03361)^2}}{\sqrt{(-23.07731)^2 + (7.21558)^2}} = 0.25294$$

Finding result indicates rudder is about 3.95 times more effective at controlling dutch roll mode than aileron.

4.3 Response due to variation in Cruise Speed for Non-Linear and Linear

4.3.1 Response for nonlinear and linear system in longitudinal motion

The nonlinear responses are simulated at 3 different nominal degree doublet excitation signal, (±1 degree, ±5 degree and ±10 degree) at constant altitude. After the aircraft is trimmed for straight and level flight, the simulation is started at t = 0 with a doublet input along the axes.

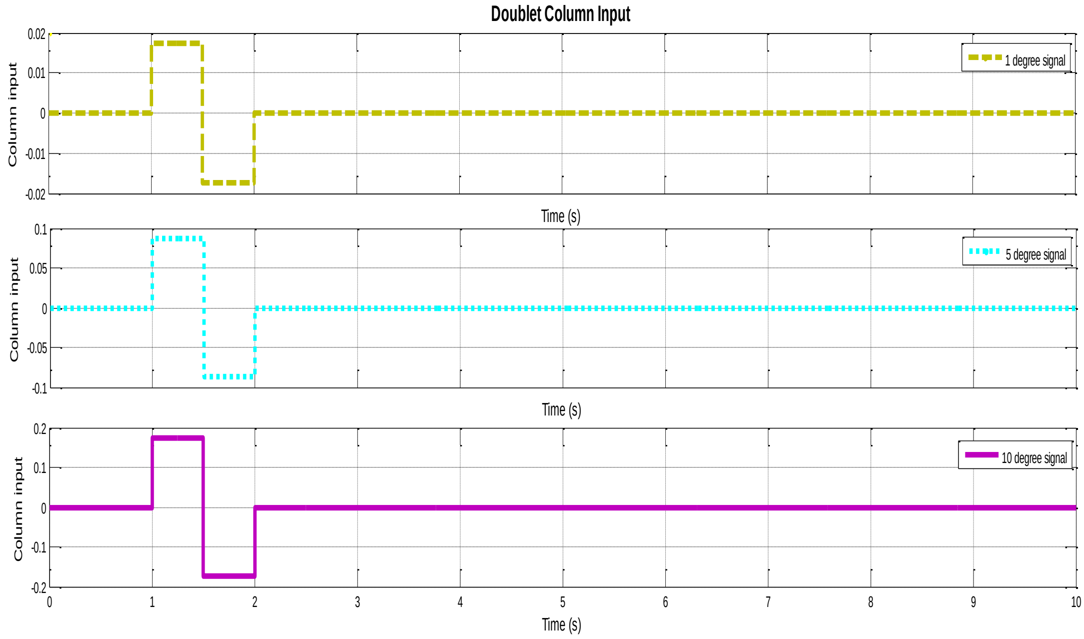


Figure 4.11 : Doublet column input.

1. Response for nonlinear system in longutudinal motion (u)

The nonlinear responses after the 3 different elevator input are simulated at 3 different nominal result.

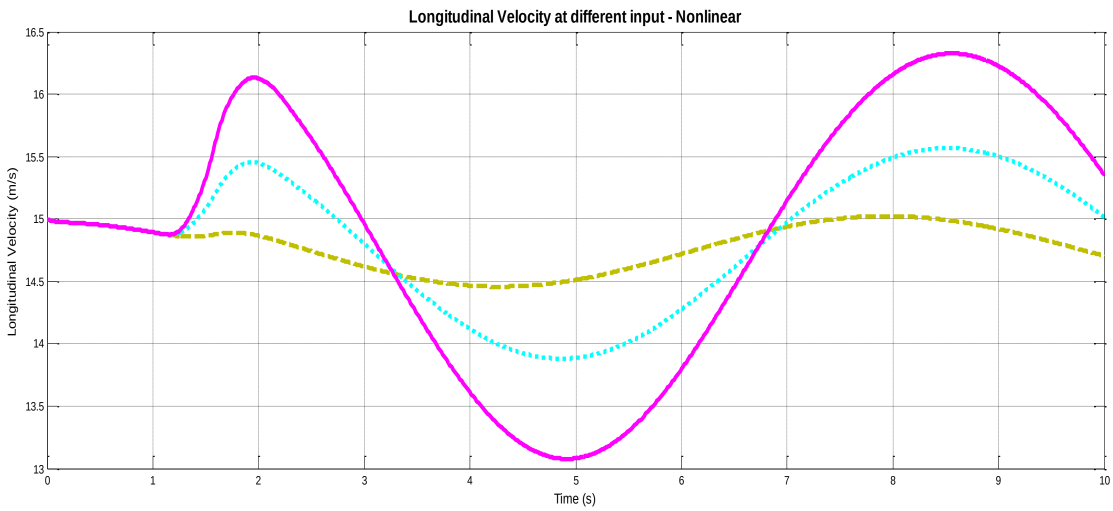


Figure 4.12 : Longitudinal velocity (nonlinear) in longitudinal motion.

2. Response for Linear System in Longutudinal Motion (u)

The linear responses after the 3 different elevator input are simulated at 3 different nominal result.

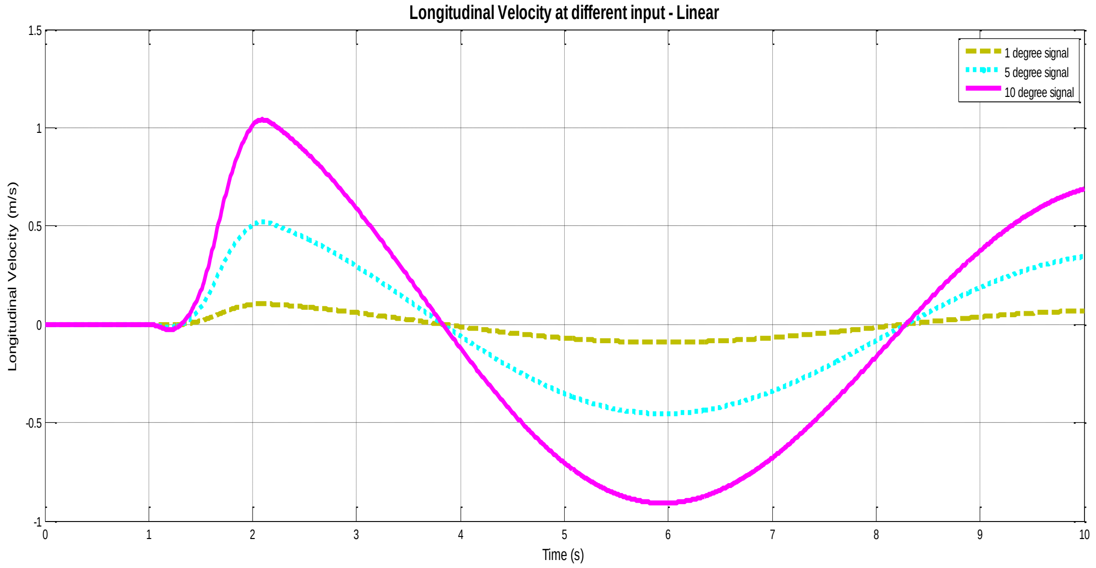


Figure 4.13 Longitudinal velocity (linear) in longutudinal motion.

3. Response for Nonlinear System in Longutudinal Motion (w)

The nonlinear responses after the 3 different elevator input are simulated at 3 different nominal result.

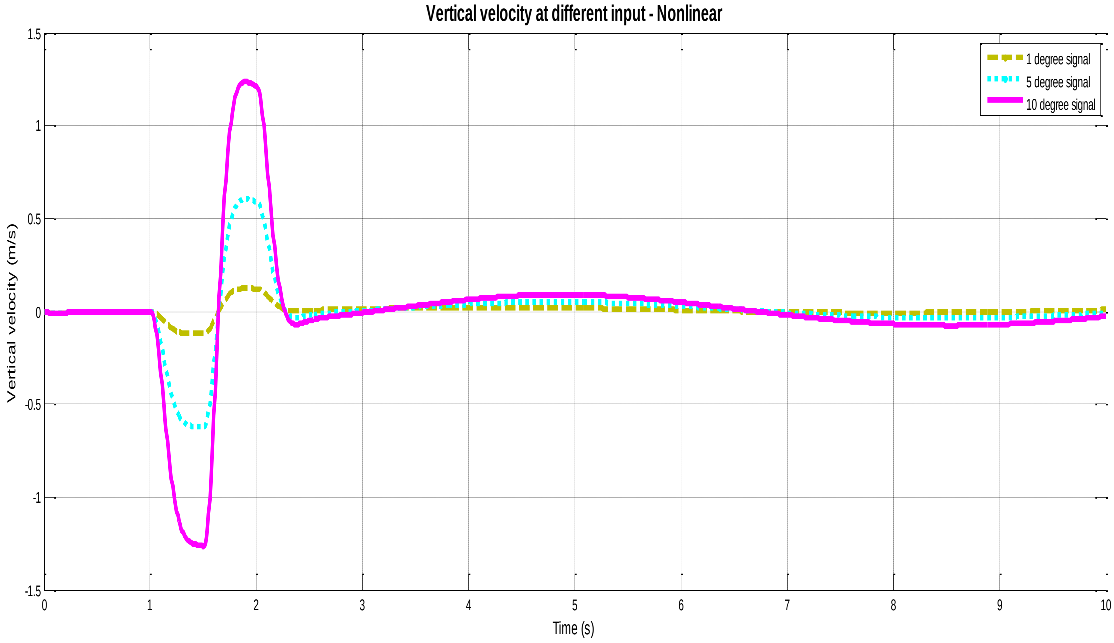


Figure 4.14 : Vertical velocity (nonlinear) in longutudinal motion.

4. Response for Linear System in Longutudinal Motion (w)

The linear responses after the 3 different elevator input are simulated at 3 different nominal result.

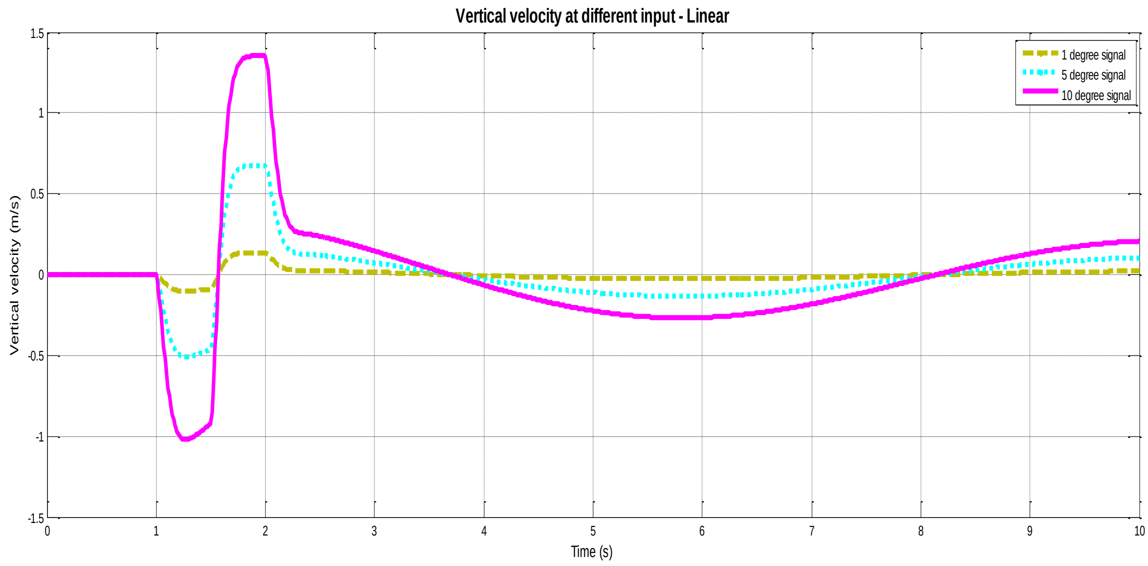


Figure 4.15 : Vertical velocity (linear) in longutudinal motion.

5. Response for Nonlinear System in Longutudinal Motion (q)

The nonlinear responses after the 3 different elevator input are simulated at 3 different nominal result.

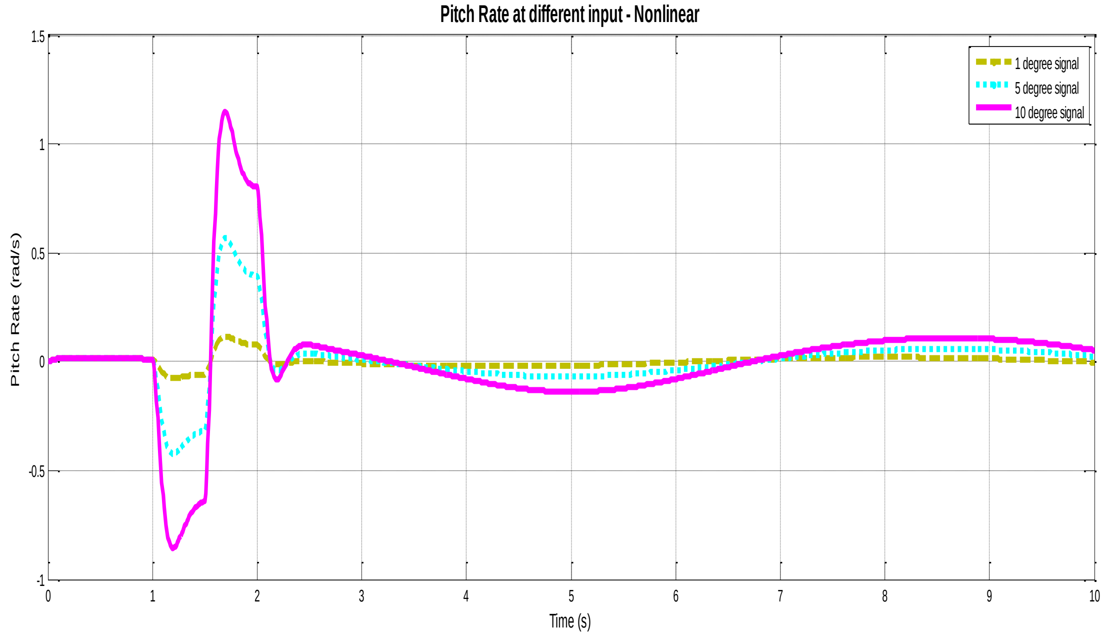


Figure 4.16 : Pitch rate (nonlinear) in longutudinal motion.

6. Response for Linear System in Longutudinal Motion (q)

The linear responses after the 3 different elevator input are simulated at 3 different nominal result.

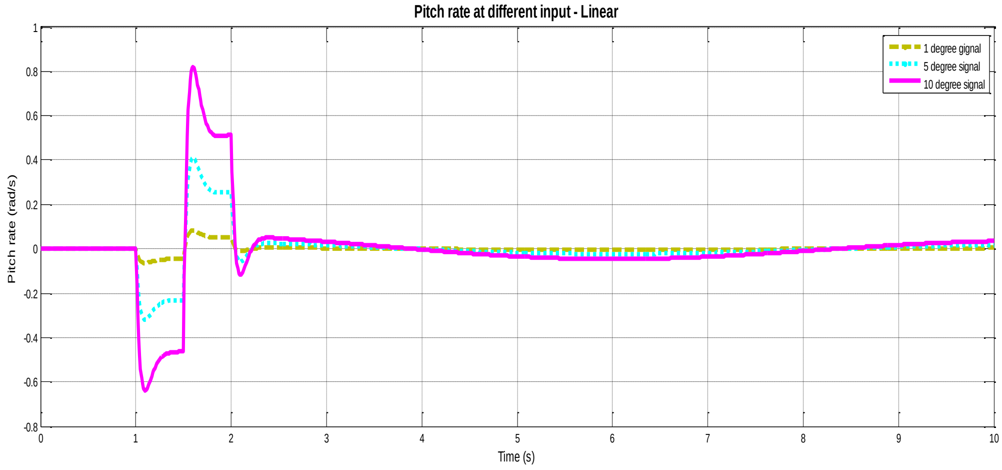


Figure 4.17 : Pitch rate (linear) in longutudinal motion.

7. Response for Nonlinear System in Longutudinal Motion (θ)

The linear responses after the 3 different elevator input are simulated at 3 different nominal result

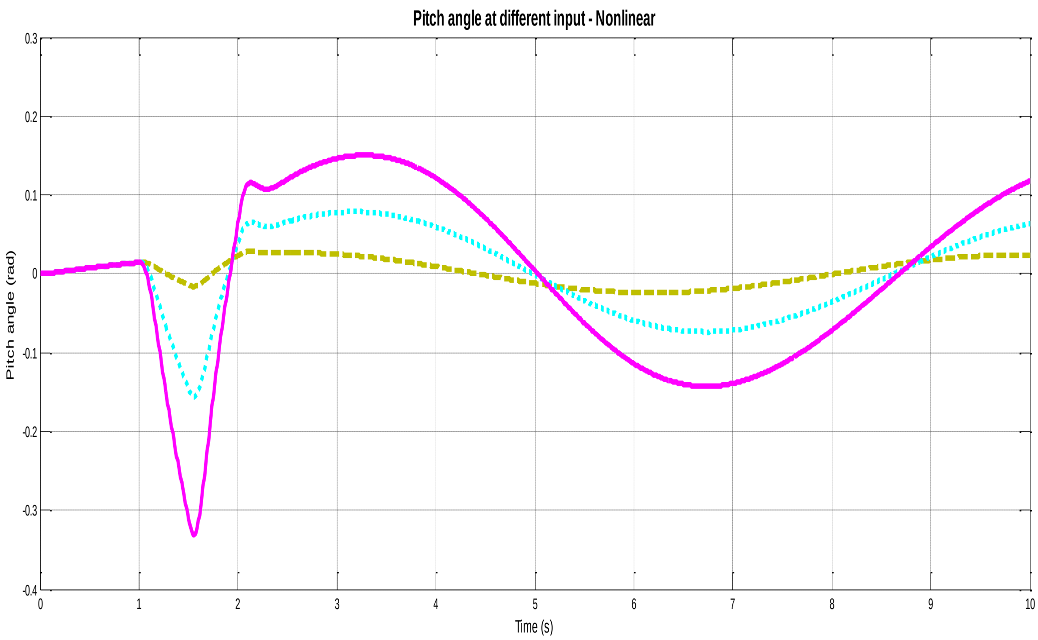


Figure 4.18 : Pitch angle (nonlinear) in longutudinal motion.

8. Response for Linear System in Longutudinal Motion (θ)

The linear responses after the 3 different aileron input are simulated at 3 different nominal result.

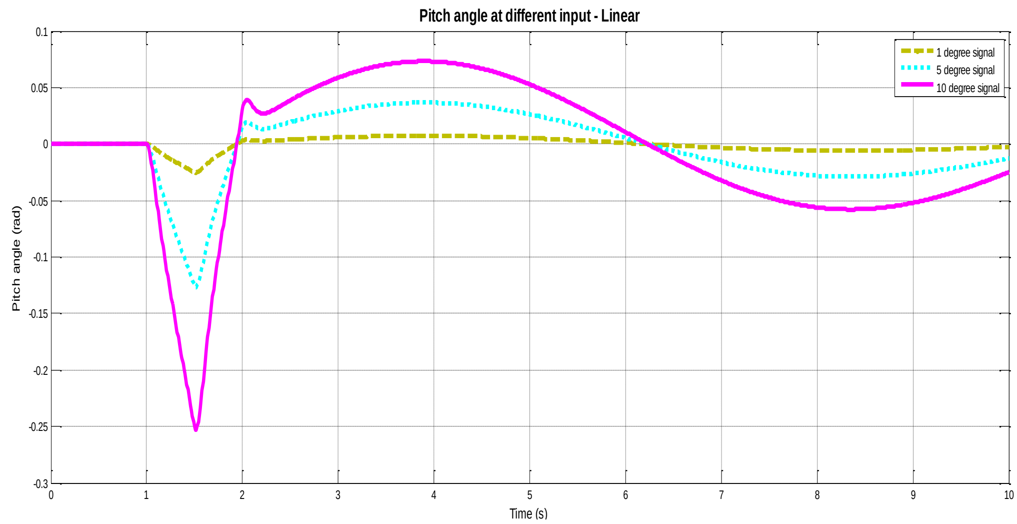


Figure 4.19 : Pitch angle (linear) in longutudinal motion.

4.3.2 Response for Nonlinear and Linear System in Lateral Motion

1. Response for Nonlinear System in Lateral Motion (β)

The nonlinear responses after the 3 different aileron input are simulated at 3 different nominal result.

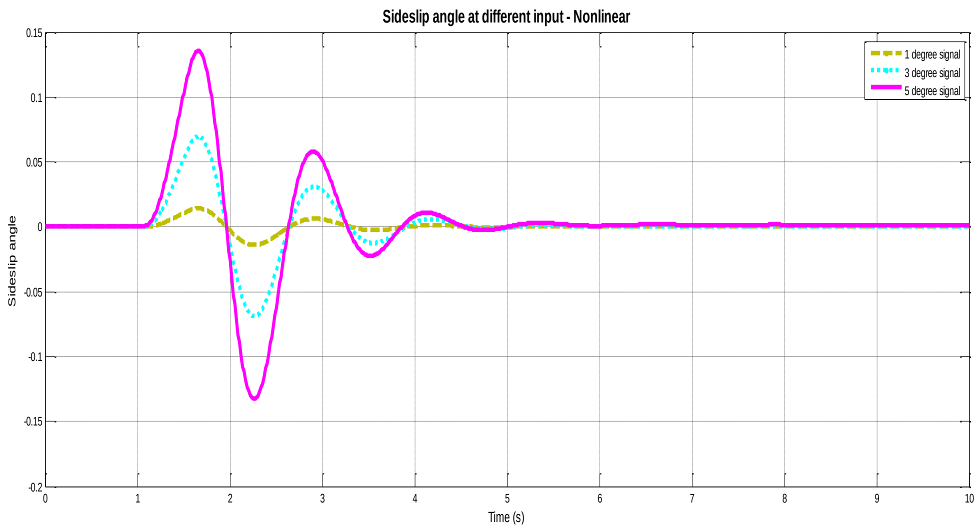


Figure 4.20 : Sideslip angle (nonlinear) in lateral motion for aileron.

2. Response for linear system in lateral motion (β)

The linear responses after the 3 different aileron input are simulated at 3 different nominal result.

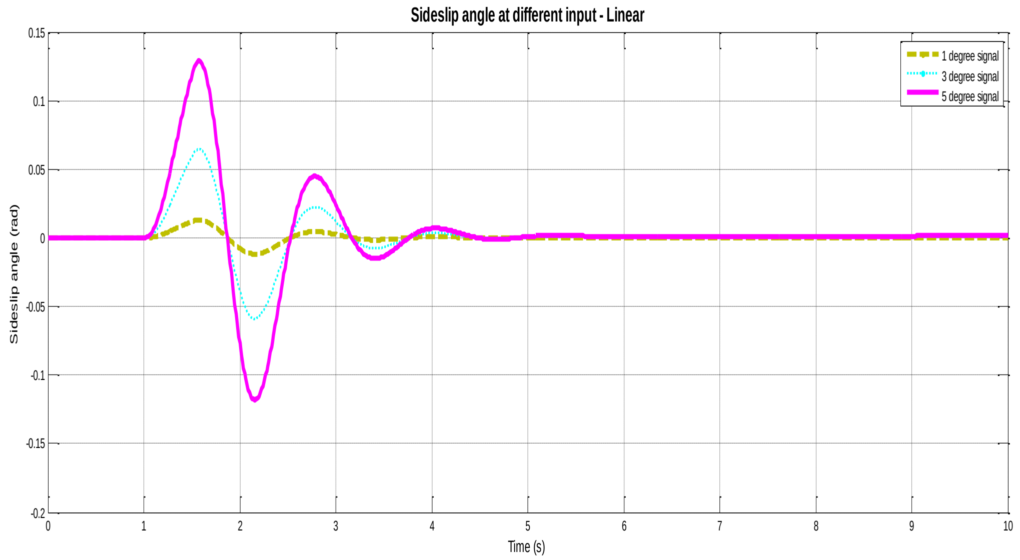


Figure 4.21 : Sideslip angle (linear) in lateral motion for aileron.

3. Response for nonlinear system in lateral motion (p)

The nonlinear responses after the 3 different aileron input are simulated at 3 different nominal result.

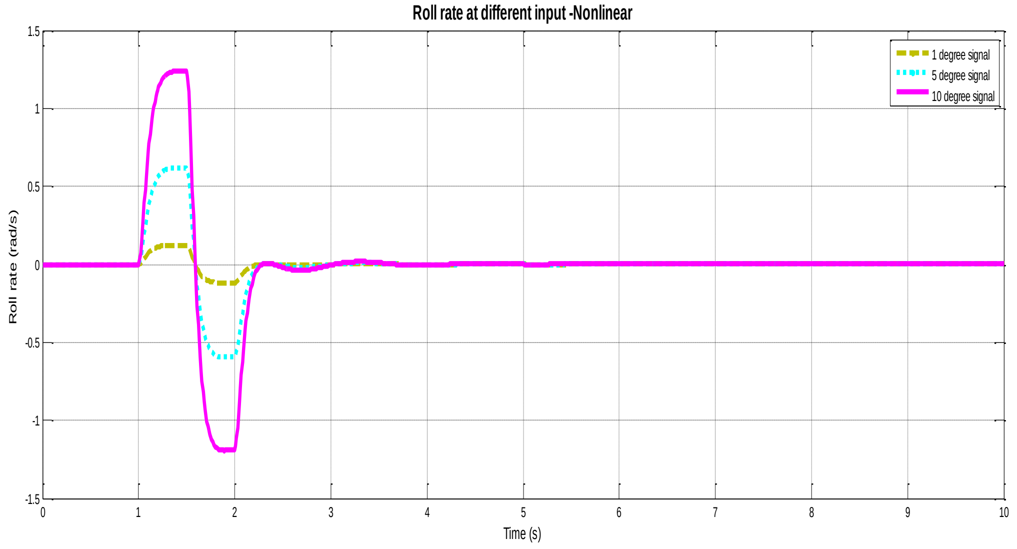


Figure 4.22 : Roll rate (nonlinear) in lateral motion for aileron.

4. Response for linear system in lateral motion (p)

The linear responses after the 3 different aileron input are simulated at 3 different nominal result

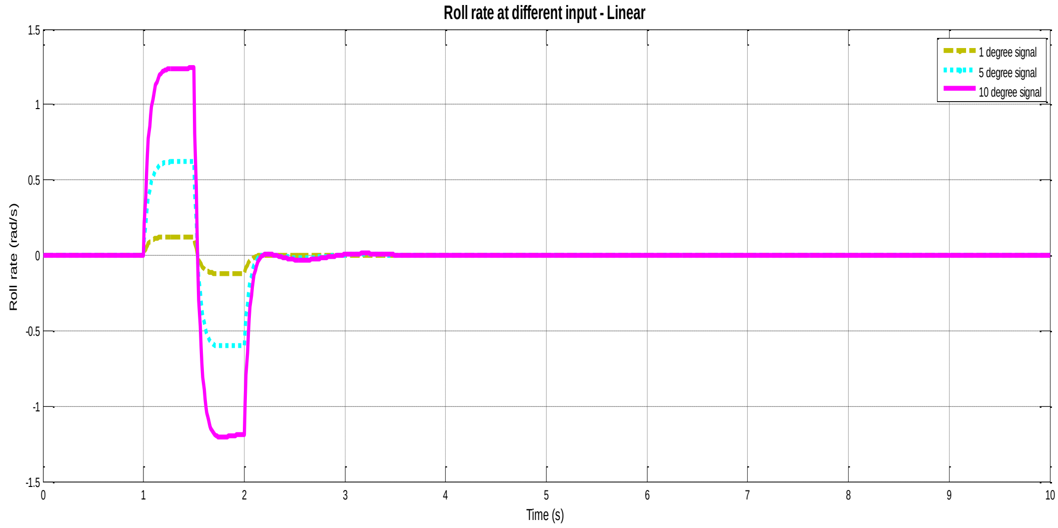


Figure 4.23 : Roll rate (linear) in Lateral Motion for aileron.

5. Response for nonlinear system in lateral motion (r)

The nonlinear responses after the 3 different aileron input are simulated at 3 different nominal result.

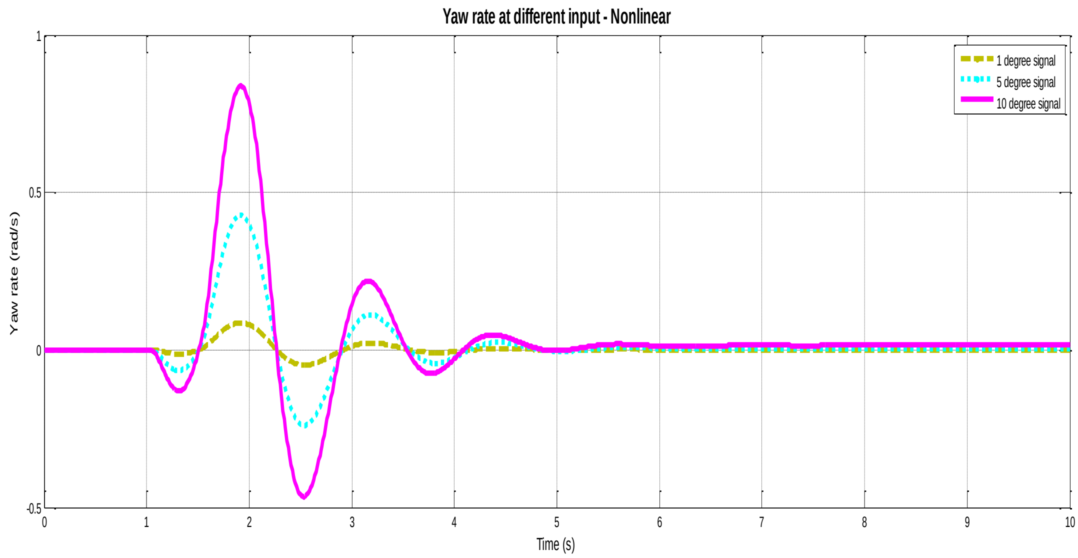


Figure 4.24 : Yaw rate (nonlinear) in lateral motion for aileron.

6. Response for linear system in lateral motion (r)

The linear responses after the 3 different aileron input are simulated at 3 different nominal result.

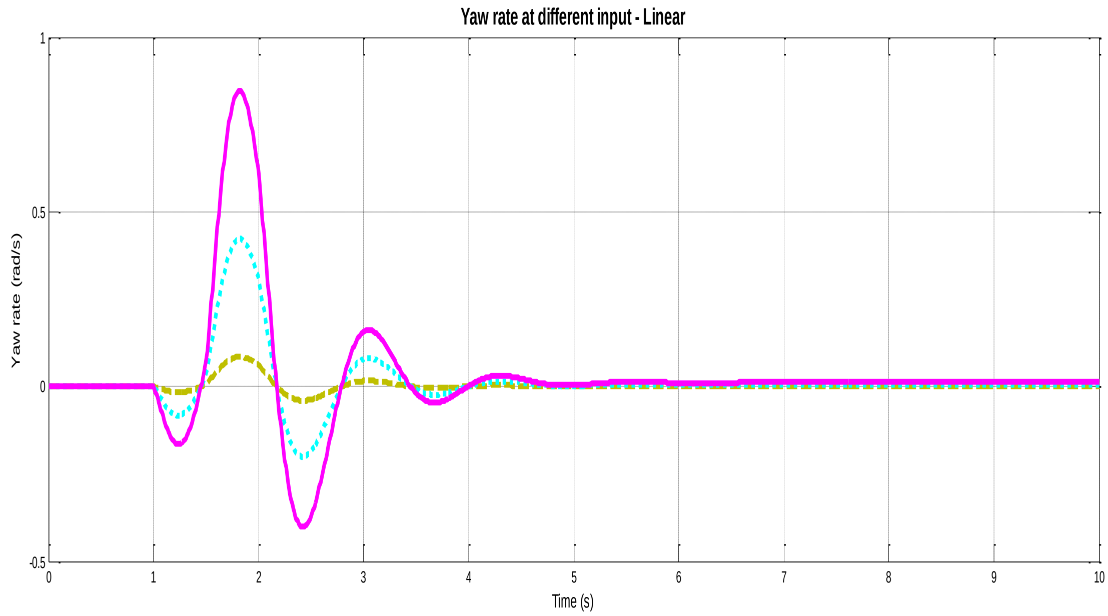


Figure 4.25 : Yaw rate (linear) in lateral motion for aileron.

7. Response for nonlinear system in lateral motion (φ)

The nonlinear responses after the 3 different aileron input are simulated at 3 different nominal result.

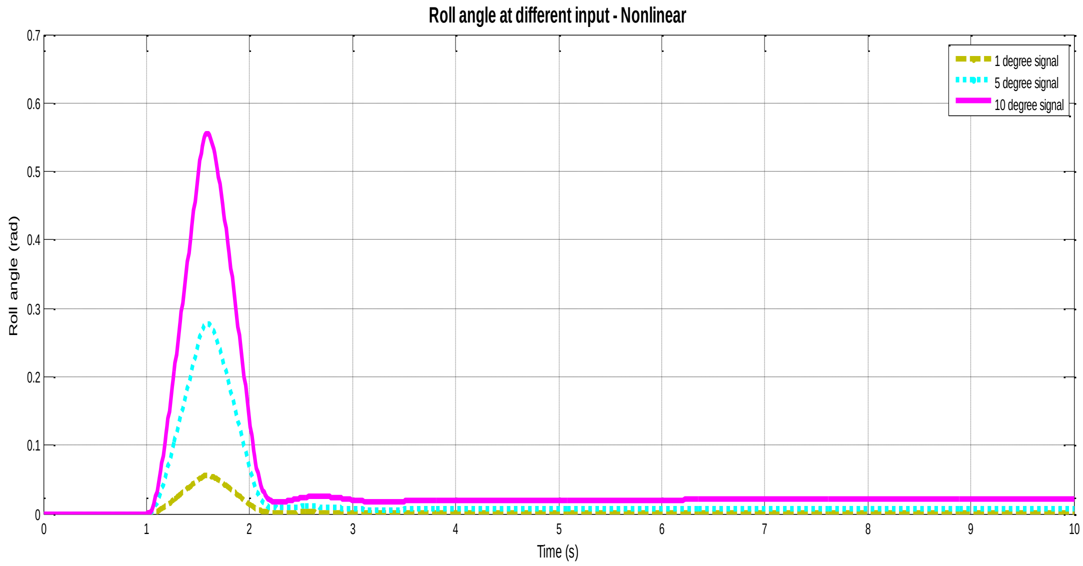


Figure 4.26 : Roll angle (nonlinear) in lateral motion for aileron.

8. Response for linear system in lateral motion (φ)

The linear responses after the 3 different aileron input are simulated at 3 different nominal result.

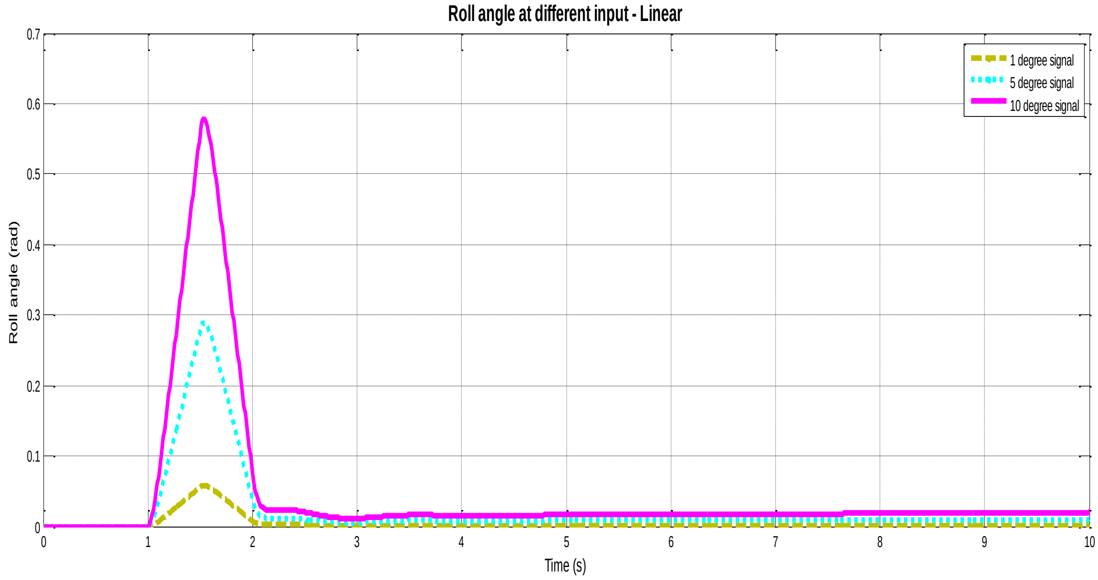


Figure 4.27 : Roll angle (linear) in lateral motion for aileron.

9. Response for Nonlinear System in Lateral Motion (β)

The nonlinear responses after the 3 different rudder input are simulated at 3 different nominal result.

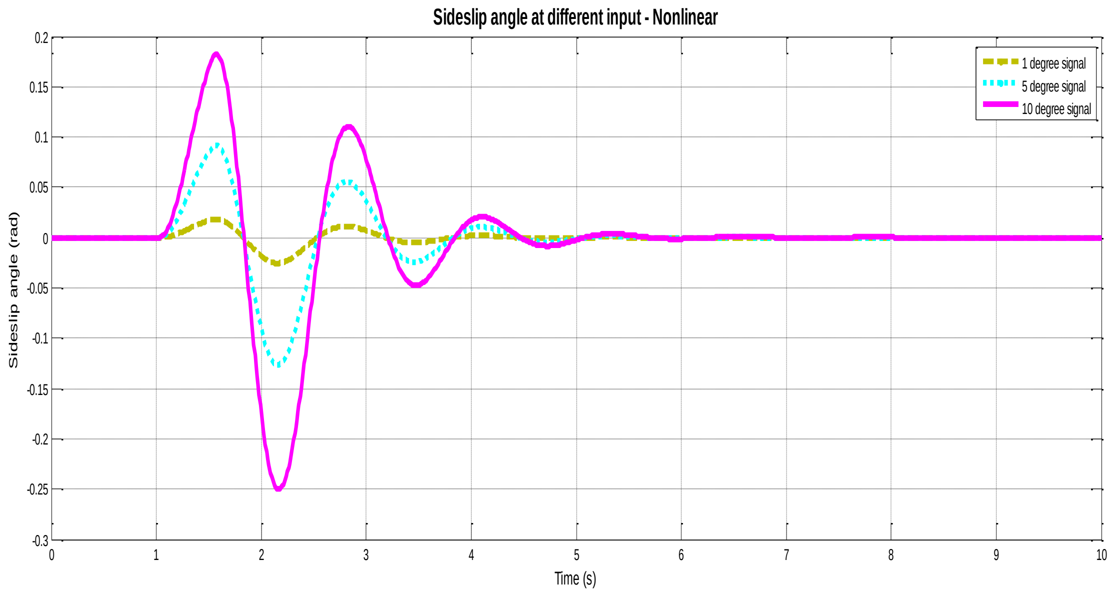


Figure 4.28 : Sideslip angle (nonlinear) in Lateral Motion for rudder.

10. Response for linear system in lateral motion (β)

The linear responses after the 3 different rudder input are simulated at 3 different nominal result.

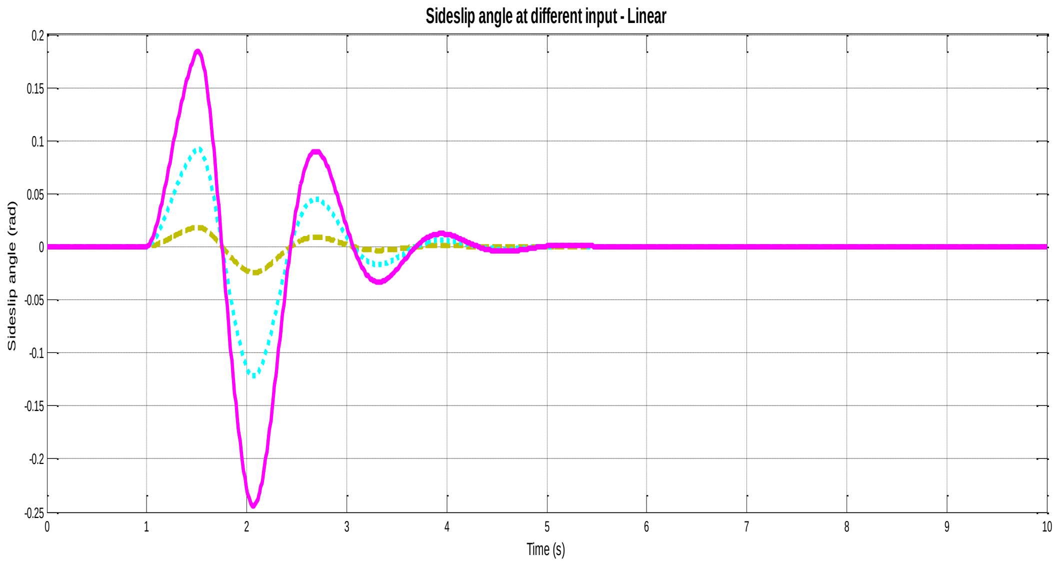


Figure 4.29 : Sideslip angle (linear) in Lateral Motion for rudder.

11. Response for nonlinear system in lateral motion (p)

The nonlinear responses after the 3 different rudder input are simulated at 3 different nominal result.

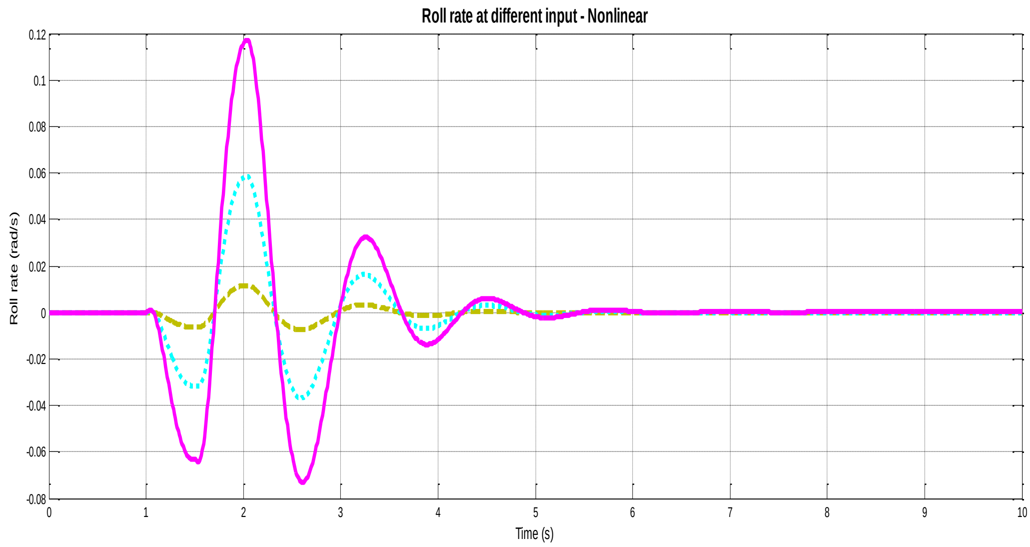


Figure 4.30 : Roll rate (nonlinear) in lateral motion for rudder.

12. Response for linear system in lateral motion (p)

The nonlinear responses after the 3 different rudder input are simulated at 3 different nominal result.

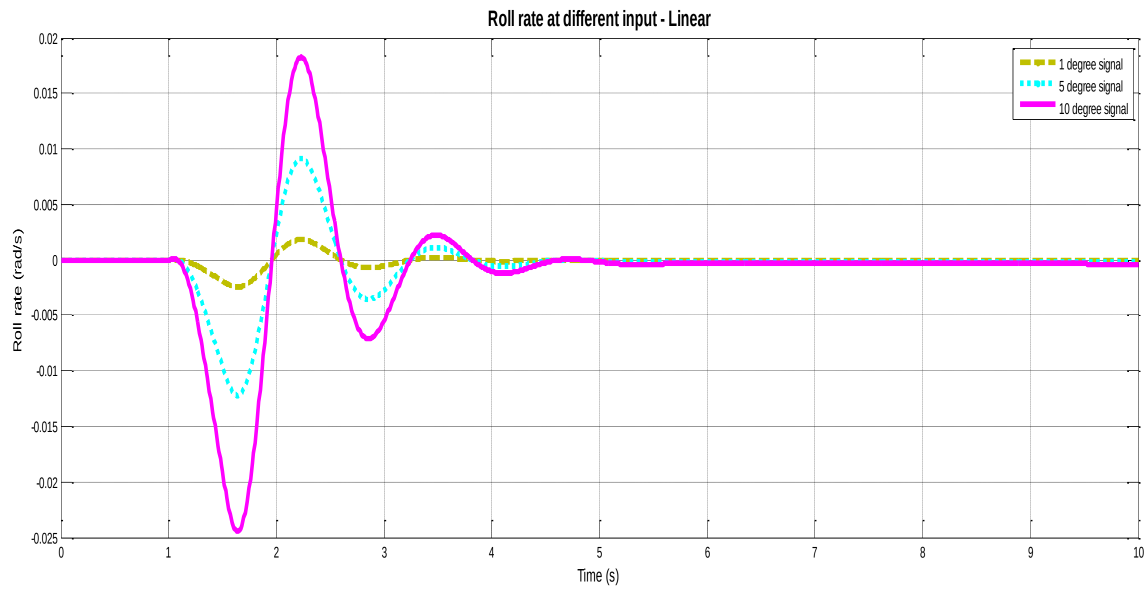


Figure 4.31 : Roll rate (linear) in lateral motion for rudder.

13. Response for nonlinear system in lateral motion (r)

The nonlinear responses after the 3 different rudder input are simulated at 3 different nominal result.

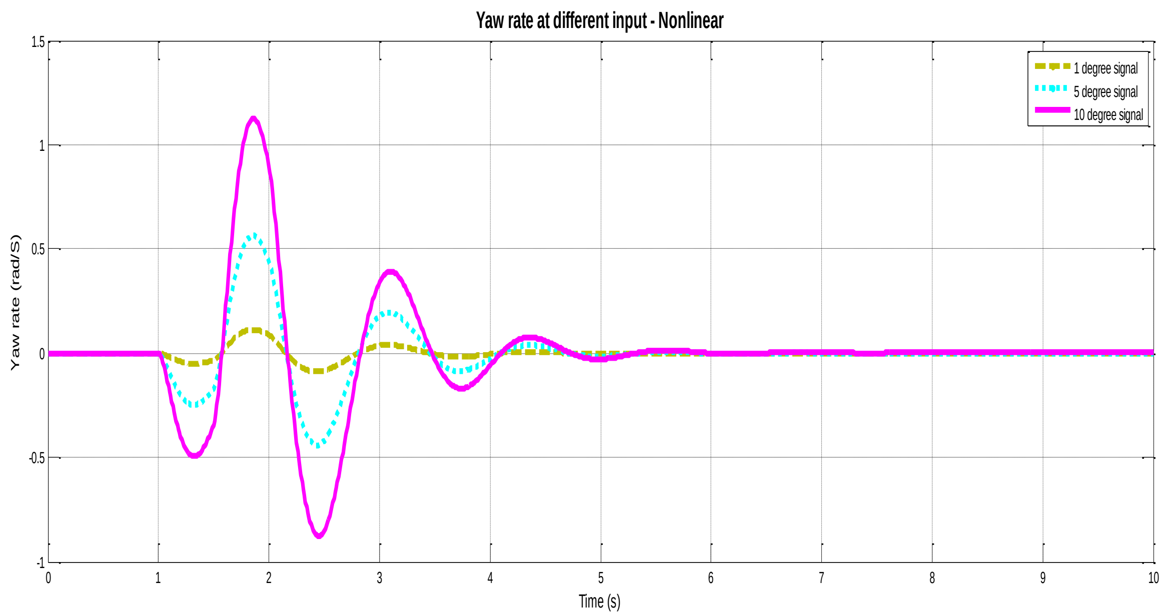


Figure 4.32 : Yaw rate (nonlinear) in lateral motion for rudder.

14. Response for linear system in lateral motion (r)

The linear responses after the 3 different rudder input are simulated at 3 different nominal result.

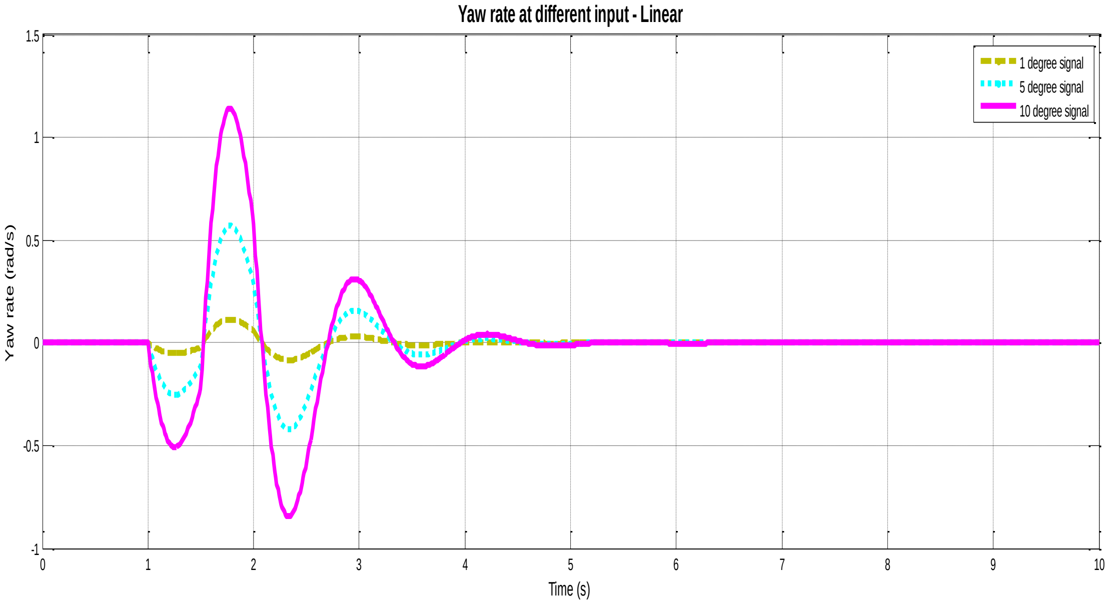


Figure 4.33 : Yaw rate (linear) in lateral motion for rudder.

15. Response for nonlinear system in lateral motion (φ)

The nonlinear responses after the 3 different rudder input are simulated at 3 different nominal result.

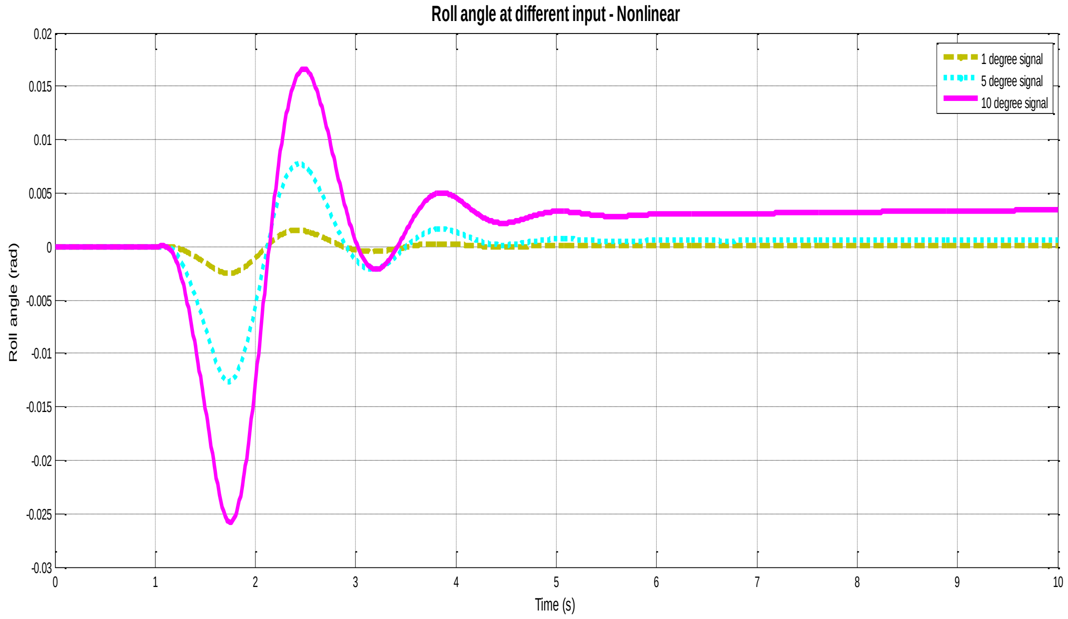


Figure 4.34 : Roll angle (nonlinear) in lateral motion for rudder.

16. Response for linear system in lateral motion (φ)

The linear responses after the 3 different rudder input are simulated at 3 different nominal result.

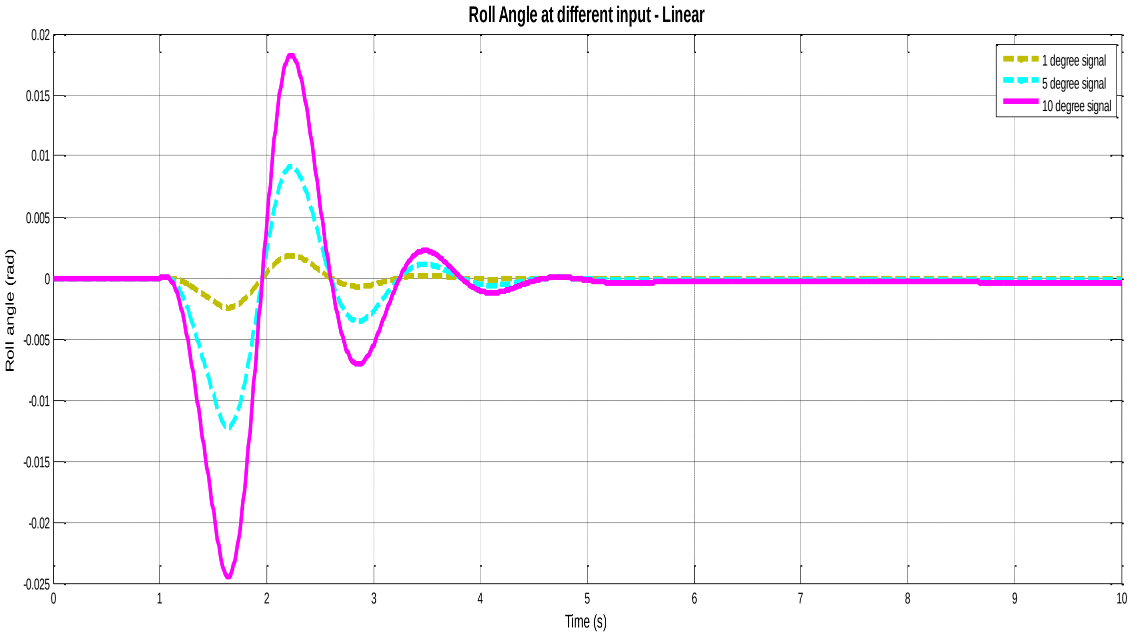


Figure 4.35 : Roll angle (linear) in lateral motion for rudder.

5. FLIGHT CONTROL SYSTEM

5.1 Introduction

The functions of an autopilot or “flight control system” can be divided in two areas: “guidance” and “control”, which can be defined as :

- Guidance: Determining the some of the action altitude, speed, relative to some reference system, to followed by the vehicle.
- Control: The development and application of appropriate forces and moments to the vehicle, which, establish some equilibrium state of vehicle motion and restore a disturbed vehicle to its equilibrium state, by point conditions into the desired limits [22].

The control method is used to determine the gains of the controllers.Using the root locus method, the nonlinear model with controllers is indicated.Then the nonlinear model with small disturbances is linearizes about specific point, as shown in section 3 with details.The block diagram of the root locus control design with desiring controller gains is founded.The finding transfer function for longitudinal and lateral motion one by one every outputs are obtained automatic control principle.The open loop transfer function of root locus is drawn and gain controller are found according to desiring design criteria.Lastly the closed loop system for step response with system controller autopilot are drawn.

The controller role metioned below is to act the aircraft commonded motions which the where the aircraft poles are merely cancelled by compensator zeros and the compensator poles used to define the desired aircraft dynamics.

In light of above this, the controller design is broken into three sections

1. Stability augmentation controllers
2. Attitude regulation controllers
3. Trajectory controllers

The stability augmentation controllers are responsible for augmenting the natural stability of the higher frequency aircraft modes, by providing artificial damping.The attitude controllers then form the lowest level of regulation, and are responsible for regulating the aircraft's pitch and roll angles, effectively controlling its acceleration. Finally, the trajectory controllers(just ramps up and down) form the outer loops that regulate trajectory motion variables such as heading and altitude.
The reason that damping is used to reduce the high frequency dynamics.Modal analysis showed that the natural high frequency aircraft modes are the short period mode, the dutch roll mode and the roll mode which obtained all of the values for every modes in section 4. The natural damping of these modes is caused by induced angles of incidence on the aircraft's airfoils during angular rate perturbations. This means that the damping is dependant on the unmanned aerial vehicle velocity and centre of gravity location, both of which change during flight.
In this thesis , only a short period damper (pitch rate damper) and a dutch roll damper are implemented. No artificial roll damping is included because the roll mode is heavily damped and of a high frequency.

$$X_u u + X_w w - g\cos\gamma_0\theta = \dot{u} \quad \textbf{(5.1)}$$

$$Z_u u + Z_w w + U_0 q - g\sin\gamma_0\theta + Z_{\delta_e}\delta_e = \dot{w} \quad \textbf{(5.2)}$$

$$M_u u + M_w w + M_{w}\dot{w} + M_q q + M_{\delta_e}\delta_e = \dot{q} \quad \textbf{(5.3)}$$

$$q = \dot{\theta} \quad \textbf{(5.4)}$$

5.2 Pitch Rate Damper

The natural damping of the short period mode arises from pitch rate causing the change in tail angle of attack α_T induced angles of incidence on the tailplane during pitch rate perturbations.Thus, the damping is a function of the airspeed and the tailplane moment arm length l_T. The aircraft's airspeed changes regularly during flight.But desired airspeed is constant.Artificial stabilisation in pitch rate damping is thus desired to desensitise the aircraft pitch rate damping to variations in these flight parameters located section 2, as well as to provide damping with respect to inertial space for wind gust disturbance rejection. Modal analysis showed that the short period mode is orders of magnitude better controlled by the elevator than by the throttle located section 4. This suggests that simple feedback of the aircraft's pitch rate to the elevator will create an effective pitch rate damper.

- **Controller Design**

The control system is summarised in the block diagram of figure 5.1 where the plant dynamic equations are obtained from equation (**3.99**). The output matrix (C) is set to extract the pitch rate state in degrees per second (deg/s).

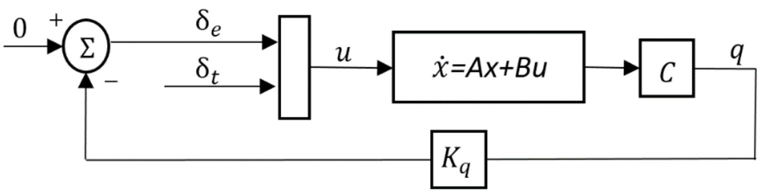


Figure 5.1: Pitch rate damper block diagram.

The root locus plot for variation of the feedback gain from pitch rate to elevator is shown **figure 5.3**. Short period mode poles which defined section 4.1.2 are shown in **figure 5.2** the two poles with opposite to the axis large movement while the poles of phugoid mode are much less visible near the origin as shown **in figure 5.3**.

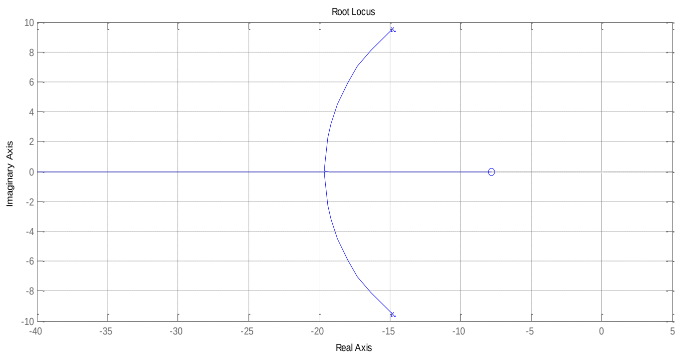


Figure 5.2 : Short period- pitch rate controller.

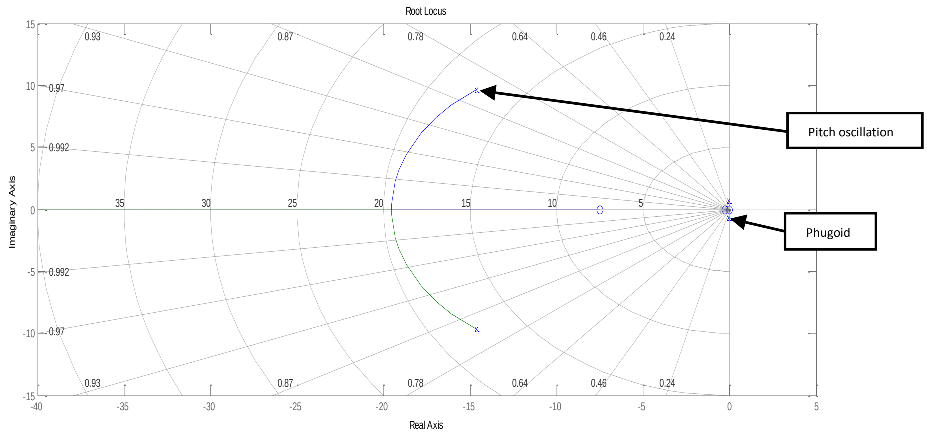


Figure 5.3 : Pitch rate damper controller.

It is desired to choose the pitch rate damper feedback gain over founded poles as high as possible, so as to provide maximum artificial damping to stability to improve the flying. But, In case of the feedback gain which is defined in figure 5.1 is made too high above average then the bandwidth of the system begins to increase and unmodeled dynamics can be unstable. In view of structural resonant also nonlinear effects can be include dynamics A feedback gain of,

$K_{q\_short} = 0.9$

$K_{q\_pitch\_rate} = 0.08$

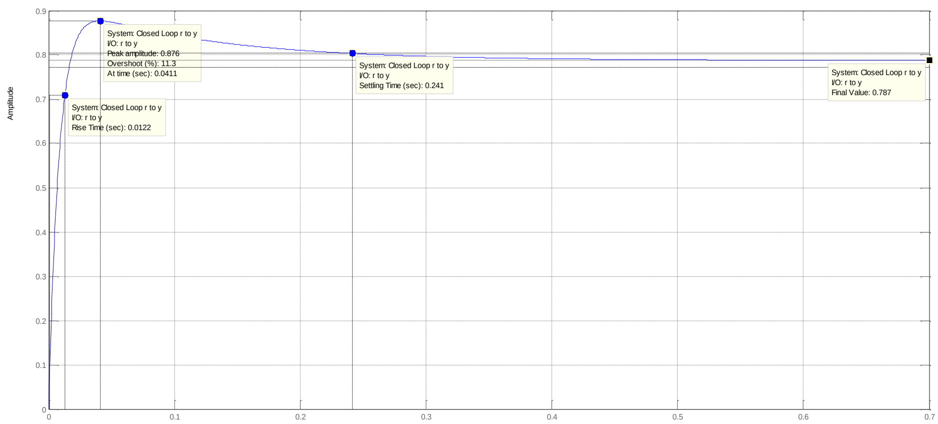


Figure 5.4 : Short time period pitch rate response.

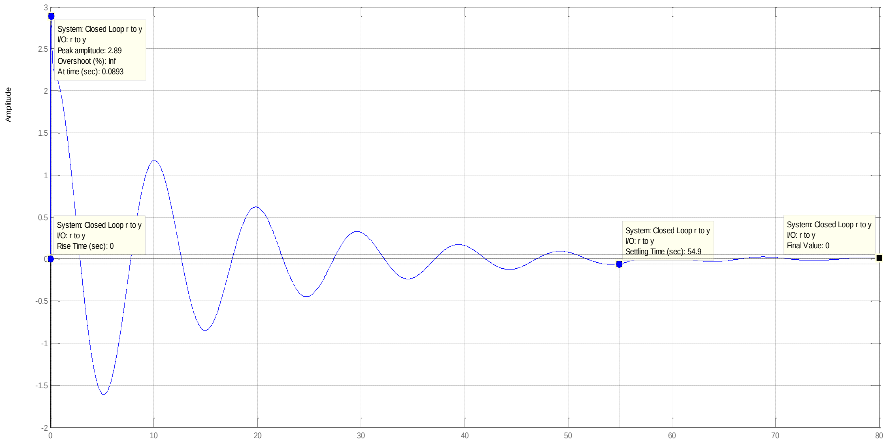


Figure 5.5: Phogoid time period pitch rate response.

increases the short period mode damping to 0.9, which is deemed sufficient to adequately desensitise the pitch damping loop without exciting unmodeled dynamics. Of course, this feedback gain can easily be fine tuned during practical flight tests as desired. The closed loop system is described by,

where the input is elevator deflection angle δ_e and the output is the aircraft pitch angle θ.For a step reference of 0.087 radians, the design criteria are the following.

For systems,

Table 5.1 : Eigenvalues,Damping and Frequencies

EigenValues	Damping	Freq.(rad/s)
-5.31e-002 + 7.05e-001i	7.51e-002	7.07e-001
-5.31e-002 - 7.05e-001i	7.51e-002	7.07e-001
-1.48e+001 + 9.56e+000i	8.39e-001	1.76e+001
-1.48e+001 - 9.56e+000i	8.39e-001	1.76e+001

Table 5.2 : Poles and Zeros

Zeros	Poles
0	-14.7620 + 9.5575i
0	-14.7620 - 9.5575i
-7,5005	-0.0531 + 0.7048i
-0,265	-0.0531 - 0.7048i

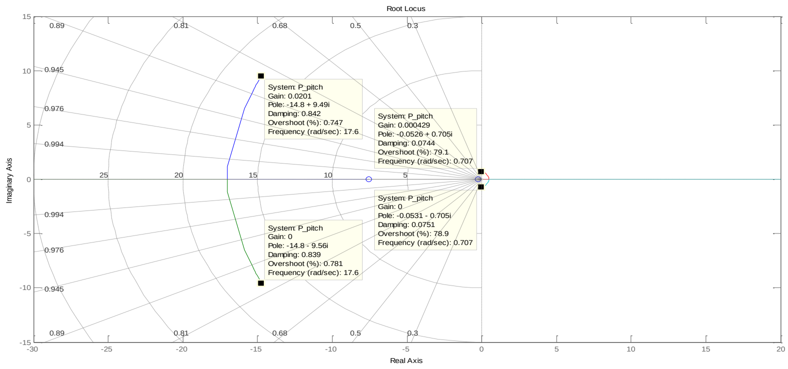


Figure 5.6 : Root locus for the aircraft pitch dynamics.

Using the root locus to design for the requirements on transient response, assuming the following unity-feedback architecture.

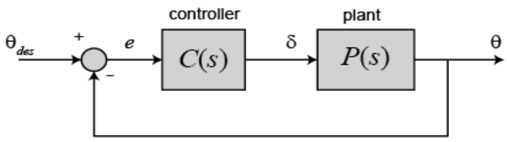
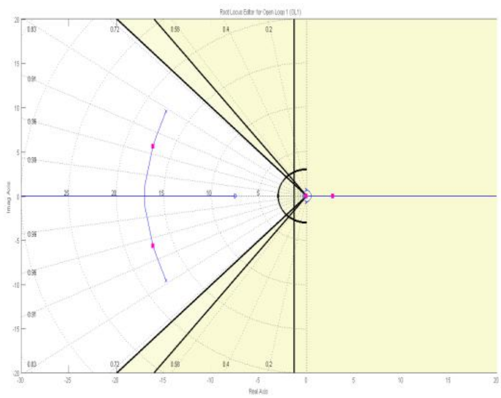


Figure 5.7 : Closed loop system.

For a step reference of elevator deflection angle δ_e ± 5 degree, the design criteria are the following.



ζ (Damping rate)	Settling Time
0.707	3 sn
Risen Time	**Overshoot %**
0.6 sn	8%
State Space Error	
2%	

Figure 5.8 : Design criteria.

5.2.1 PI controller design

PI controller produces as its output a weighted sum of the input signal and its integral. Its transfer function is,

$$G_c(s) = K_p + K_i \frac{1}{s} \qquad \textbf{(5.5)}$$

The system design with the PI controller zero is placed very close to its pole located at the origin so that the root locus is almost zero, the location of pole can be determined using equations. A PI controller is used to improve the system response steady state errors since it increases the control system type by one.

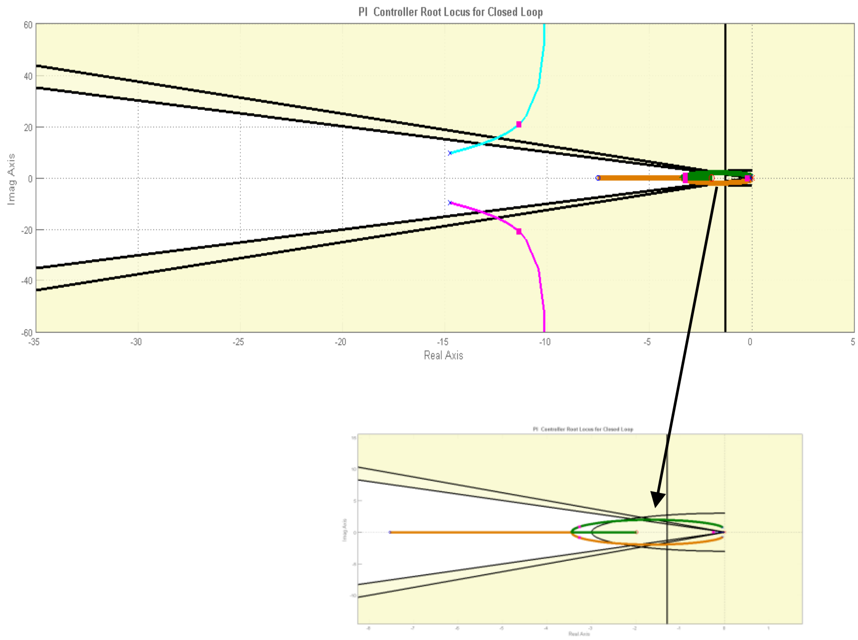


Figure 5.9 : PI controller root locus of pitch angle.

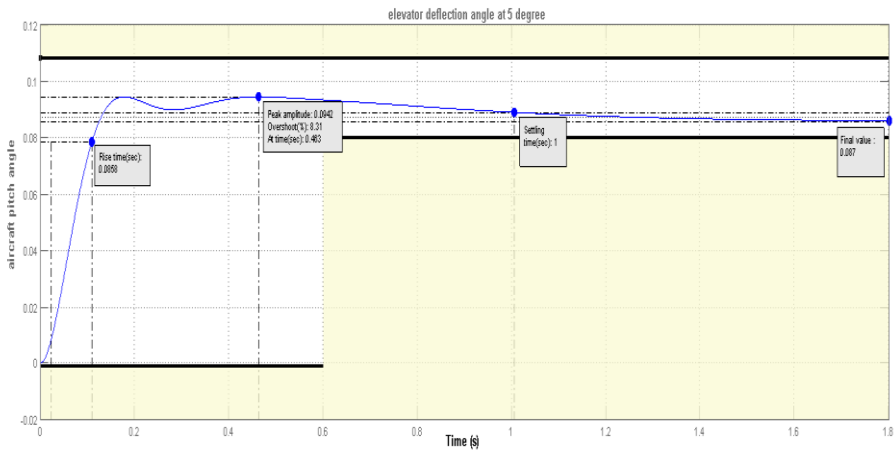


Figure 5.10 : Response PI controller of pitch angle.

K_p = -3.850176 K_I = -8.7504

5.2.2 PID controller design

The PID controller is a combination of PD and PI controllers; hence its transfer function is given by,

$$G_c(s) = K_p + K_i \frac{1}{s} + K_D s \tag{5.6}$$

The PID controller can be used to improve both the system transient response and steady state errors.

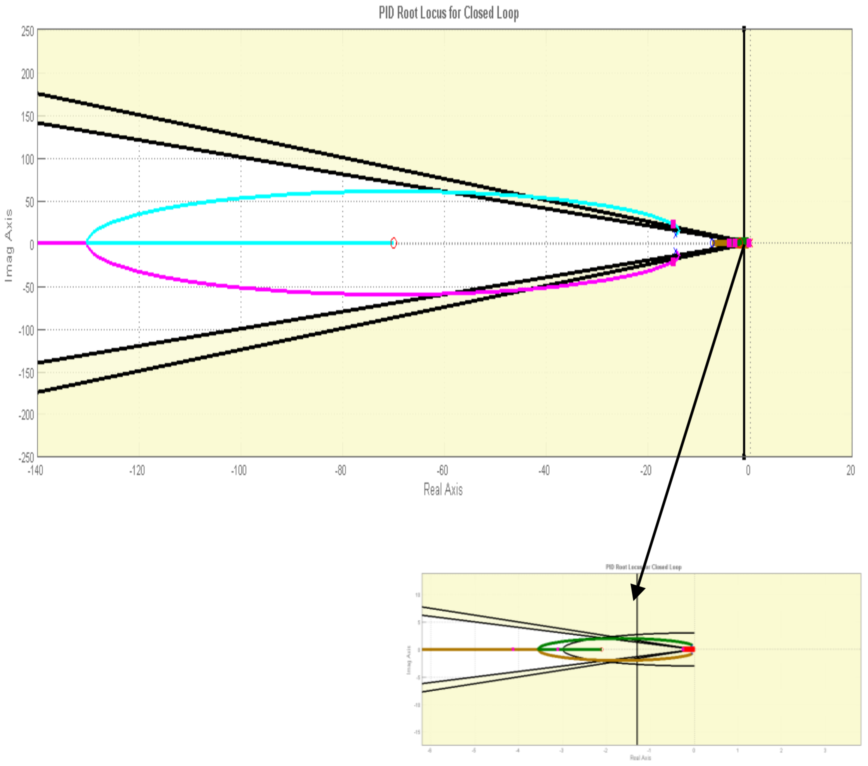


Figure 5.11 : PID Controller root locus of pitch angle.

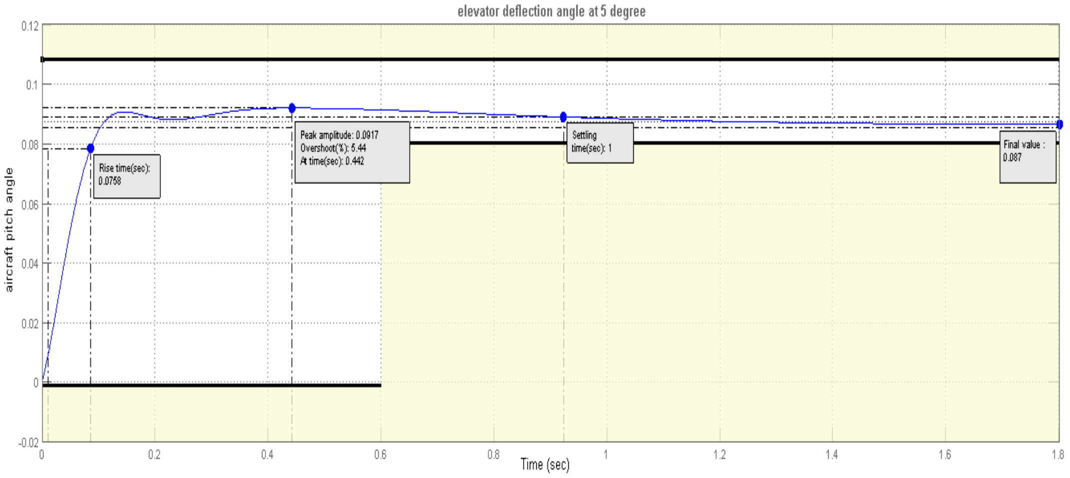


Figure 5.12 : Response PID Controller of Pitch angle.

K_p = -5.51131 K_i = -11.387 K_D = -0.074265

5.3 Altitude Controller

The altitude controller is the only longitudinal trajectory controller. Automatic control of the longitudinal trajectory requires simultaneous control of trust and pitch attitude may result in a loss of speed and an eventual stall.The design altitude controller approach founded as shown in **figure 5.13** is to generate a climb rate command from the altitude error signal, which means that just the extra integrator factor because of the height state need be stabilised by the feedback. The approach eases the design of the altitude controller because all of the dynamics specific to the aircraft, are already encapsulated by the airspeed and climb rate controller. The commanded speed and altitude are the reference values u_0, and h_0, so that the two corresponding error signals are -*Δu* and -*Δh*..

If there is an initial error in h, say the altitude is too low, then in order to correct it, the unmanned aerial vehicle's flight path must be deflected upward. This requires an increase in angle of attack to produce an increase in lift. The end result of the modeling to follow is a system of differential equations that is then integrated to get time solutions. Since the limiter is inherently a nonlinear element, it is in any case not possible to include it in a transfer function based analysis. J_e modified to account for the elevator servo actuator.

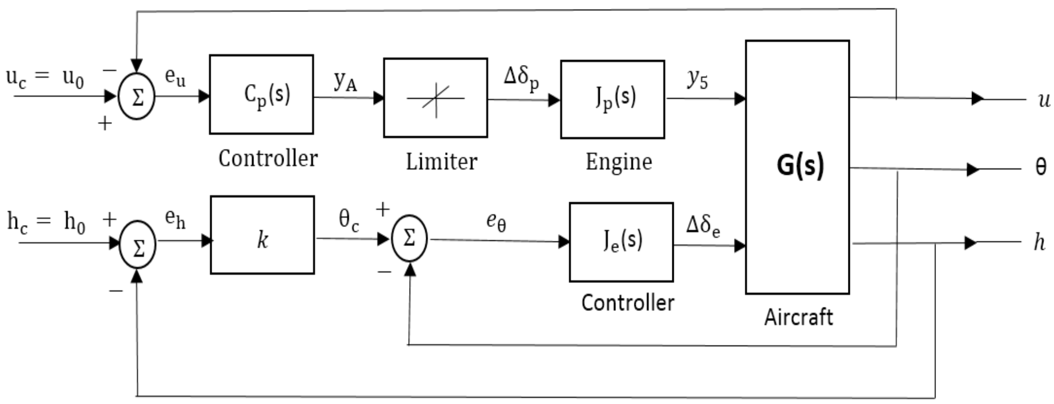


Figure 5.13 : Block diagram of altitude controller.

$\dot{x}$=*Ax*+*Bc** **(5.7)**

x=[*Δu w q θ Δh*] **(5.8)**

c^*= $[\delta_e \quad y_5]^T$ **(5.9)**

A=$[a_{ij}]$, B=$[b_{ij}]$ **(5.10)**

The state vector now consists of the original five variables from (3.104) plus the two control variables $\Delta\delta_e$, and $\Delta\delta_t$, plus the five y_i defined above, plus the five y_i defined above, making 12 in all.

Change of altitude h can be written as the flight path angle times the velocity, so that

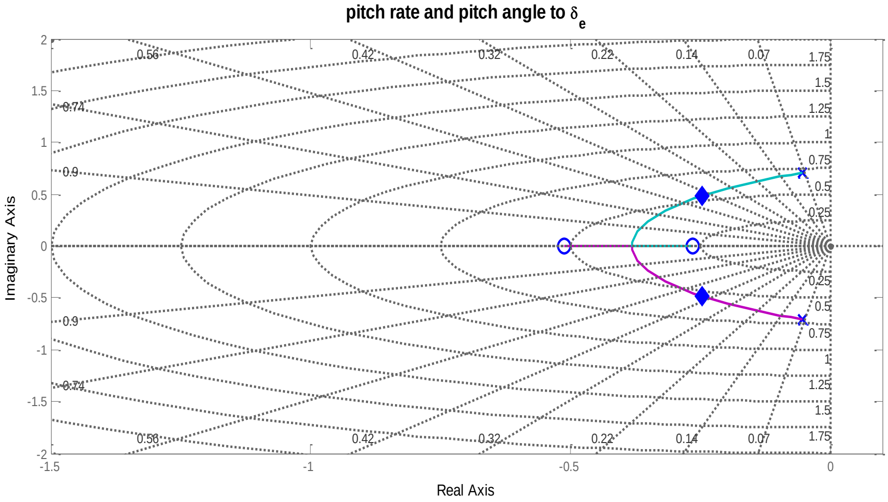


Figure 5.14 : Root Locus related with simulink (q and θ, δ_e).

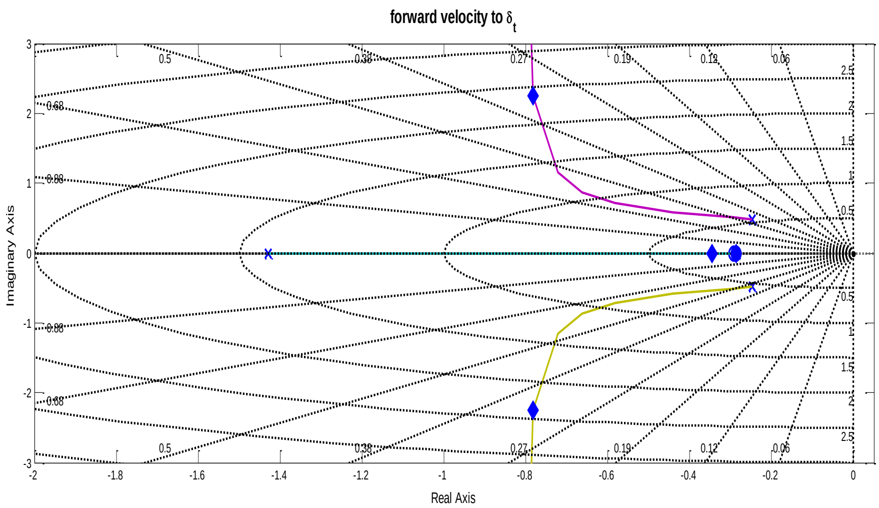


Figure 5.15: Root Locus related with simulink (u, δ_t).

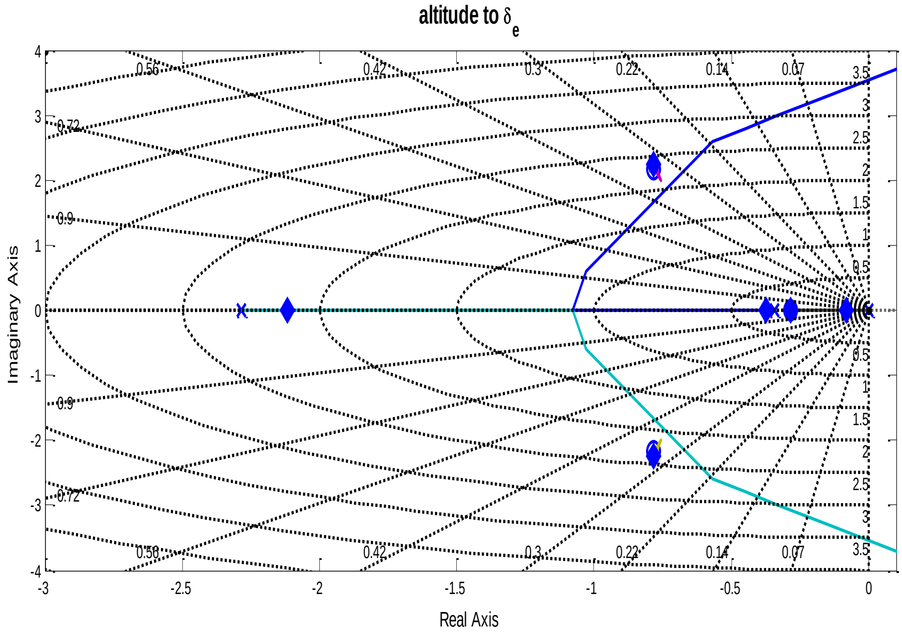


Figure 5.16: Root locus related with simulink (h, δ_e), Lead controller.

is found to yield acceptable closed loop pole locations.

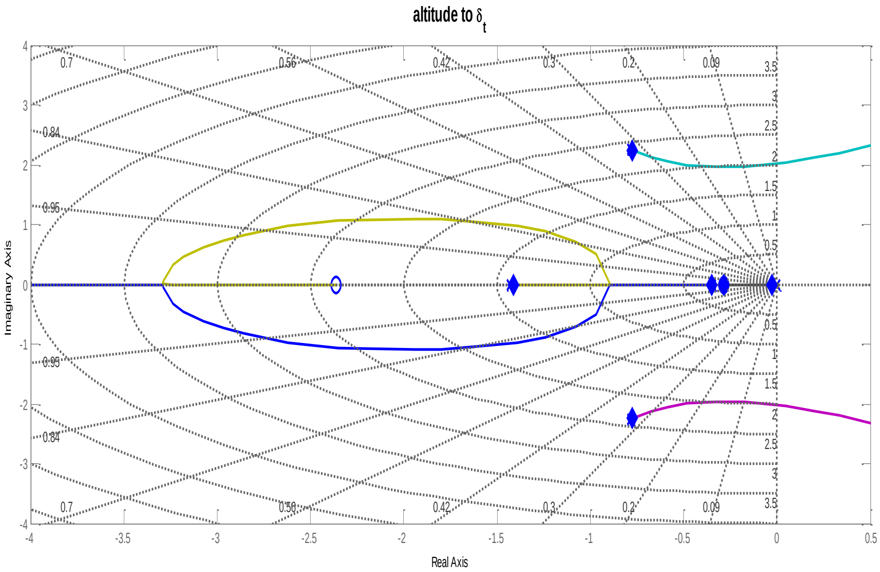


Figure 5.17: Root Locus related with simulink (h, δ_t), Lag Controller.

For constant velocity,

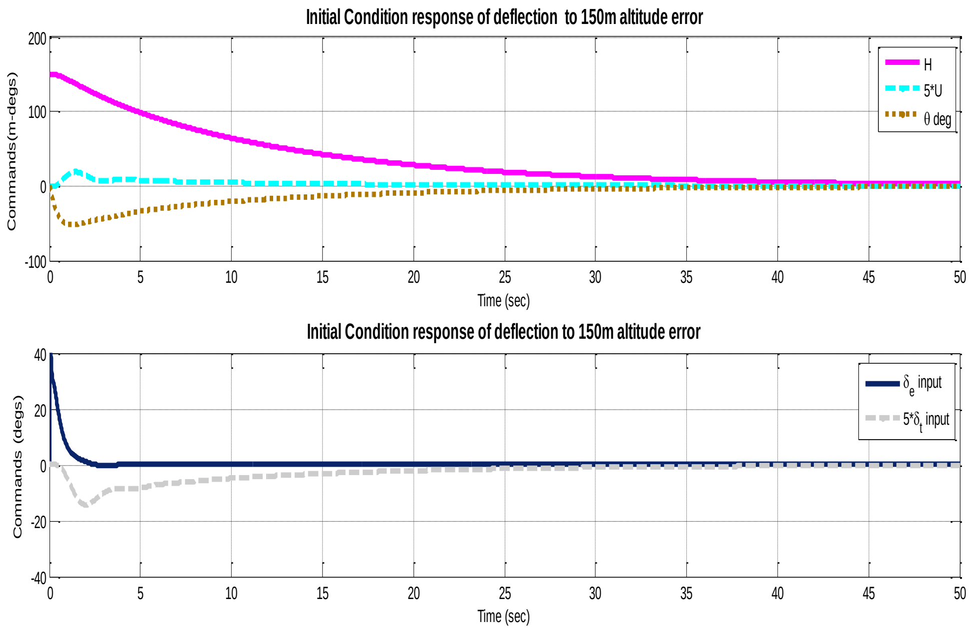


Figure 5.18: Initial condition response for 150m altitude using elevator controller.

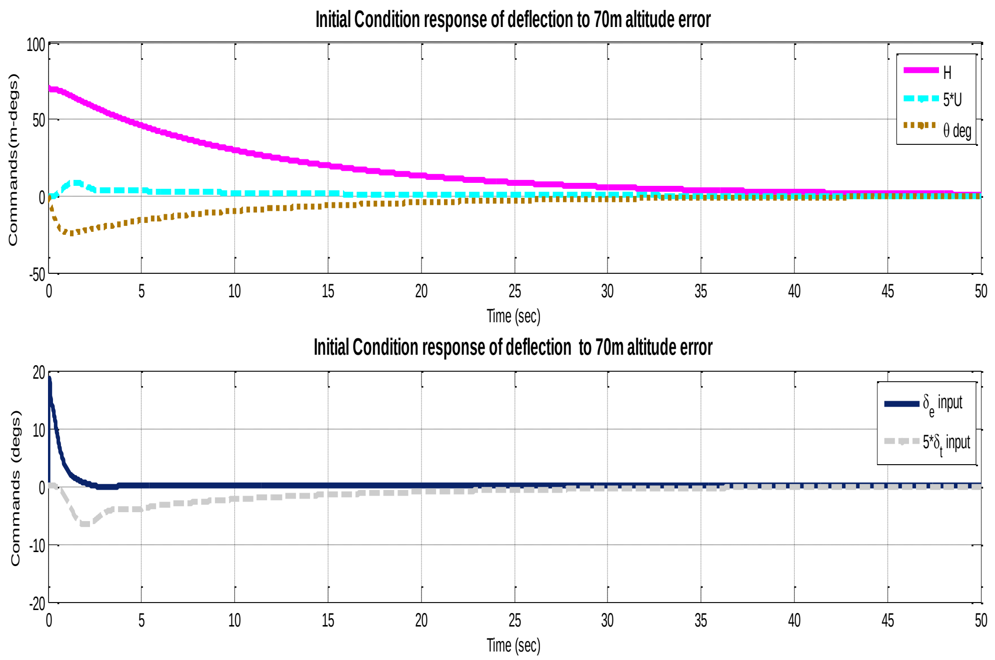


Figure 5.19: Initial condition response for 70m altitude using elevator controller.

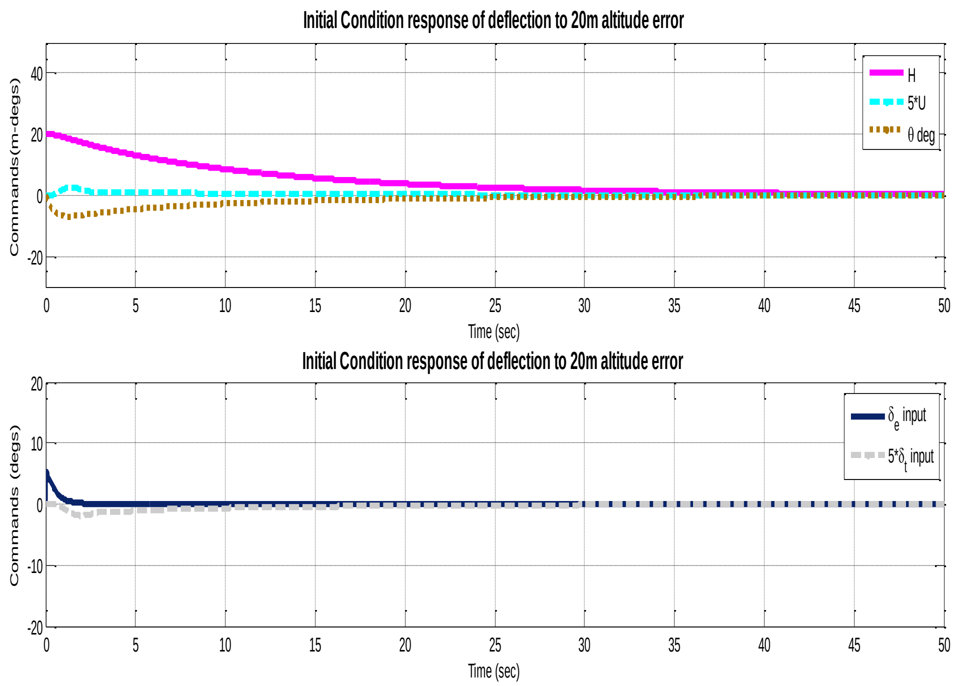


Figure 5.20: Initial condition response for 20m altitude using elevator controller.

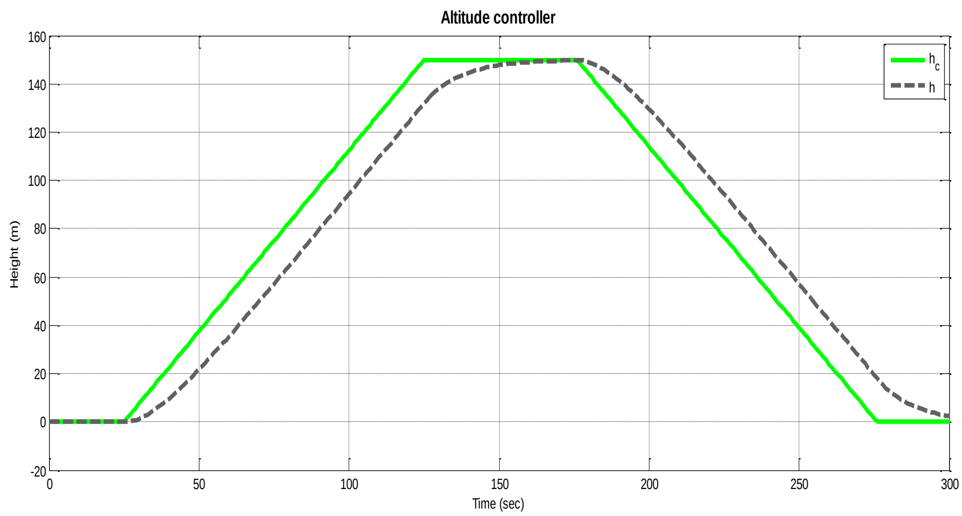


Figure 5.21: Command following ramps up and down.

For altitude control mode is moved to the altitude position which defined input above. Adjustable ramp up and ramp down times, independently set depends on elevator deflection and throttle action.

5.4 Dutch Roll Damper

The dutch roll mode is a mode that includes a roll oscillation with an out-of-phase yaw oscillation. The mode is excited by the use of rudder or ailerons creating a sideslip.Modal analysis showed that the natural damping of the dutch roll mode is low. This natural damping is dependent on the fin moment arm length (l_f) and the aircraft velocity, both of which vary during flight.

The dutch roll damper is implemented by feeding a yaw rate signal back to the rudder as shown **figure 5.22**. The rudder is used to provide the damping because it was shown by modal analysis to be 3.95 times more effective at controlling the dutch roll mode than the ailerons are.

The yaw rate signal must however, be high pass filtered before it is fed to the rudder so that the damper does not counter constant turn rate motions.

- **Controller Design**

The control system is summarised in the block diagram of **Figure 5.22** where the plant dynamics are taken from equation (**3.109**). The output matrix (C) is set to extract the yaw rate state in degrees per second (deg/s).

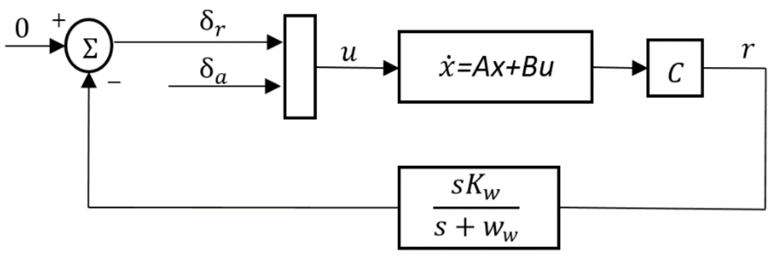


Figure 5.22 : Dutch roll damper block diagram.

the yaw rate command signal moves variable through a washout filter which is a filter that makes a transition caused by changes in the input source more smooth so the output of the limiter is fed through a filter to rudder before being applied to the servo. The choice of the washout filter type cutoff frequency is an iterative process. But , it can be chosen negative or positive differential equations which have low enough to ensure the frequency of the dutch roll mode.With the natural frequency of the dutch roll mode at 5.21 rad/s, a washout filter cutoff frequency of 3.27 rad/s is found to work well. With the filter dynamics included, the root locus plot for the variation of the filter feedback gain (KW) is shown in **Figure 5.23** [23].

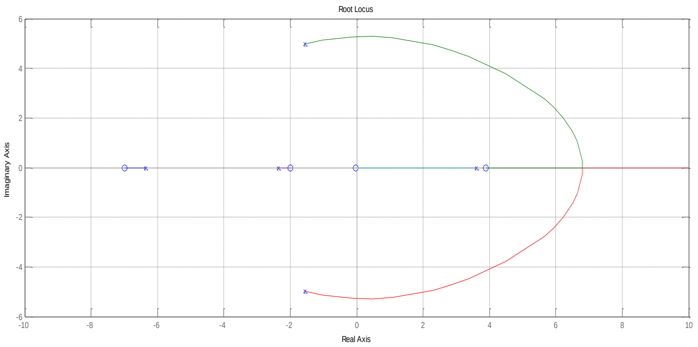


Figure 5.23 : Dutch roll damper root Locus.

The filter gain is increased to provide maximum damping of the dutch roll mode for the cutoff frequency chosen. The complete washout filter with the feedback gain included is stated below,

$$D_w\,(s) = \frac{-0.2s}{s+2} \qquad \textbf{(5.11)}$$

The closed loop system dynamics with the augmented washout state (x_w) are,

$$\begin{bmatrix} \dot{x} \\ \dot{x}_w \end{bmatrix} = \begin{bmatrix} A - BD_wC & -BC_w \\ B_wC & A_w \end{bmatrix} \begin{bmatrix} x \\ x_w \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u \qquad \textbf{(5.12)}$$

Linear Simulation

The effect of the dutch roll damper can be verified by linear simulation. The yaw rate response of the aircraft with the dutch roll damper both armed and disarmed, and the lateral state vector initialize with the real part of the dutch roll mode eigenvector.It is clear that the dutch roll mode dynamics are well suppressed by the damper.

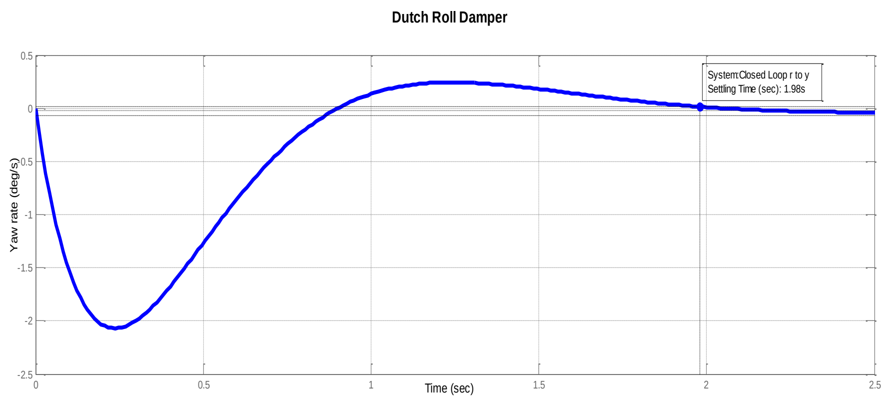


Figure 5.24 : Dutch roll damper result with washout filter.

5.5 Yaw Rate Controller

When an aircraft has a low speed at a high altitude (max 150 m), the Dutch roll properties of the aircraft deteriorate. To prevent to this, a yaw damper is used.

The yaw rate controller indirectly regulates the roll angle of the aircraft. The ailerons provide the best control over pure roll motions, as indicated by modal analysis of the roll mode. This suggests that a simple, effective approach to controlling the yaw rate, would be to feed a yaw rate signal back to the ailerons. However, it is desired to be able to command a certain yaw rate with zero steady state error. in that situation, the yaw damper commands a rudder angle because it senses an r, the angle would no doubt not be the right one needed for a coordinated turn. In fact during a right turn the yaw damper would always produce left rudder, whereas right rudder would usually be required.

5.6 Yaw Damper Design

The yaw damper first to control the rudder, and then is designed the autopilot to control the ailerons. The yaw damper system provides aircraft stabilization about the yaw axis and it minimize the Dutch roll during flight the aircraft by providing rudder displacement proportional to, and opposing, the yaw rate of aircraft. The yaw damper for the purpose of improving aircraft Dutch-roll model damping will be designed using state variable feedback concept.

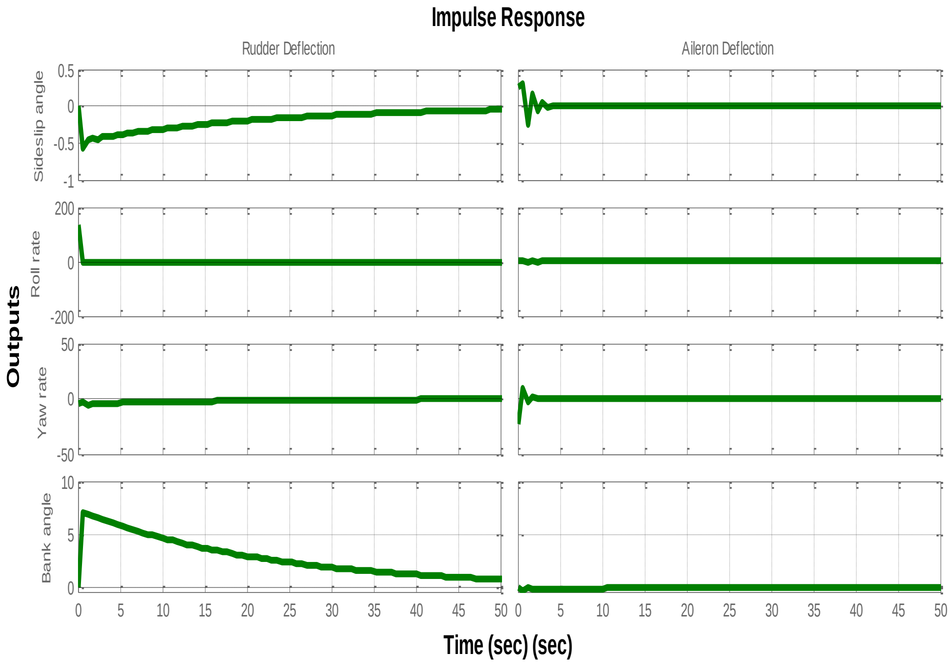


Figure 5.25 : Deflection of control surface result.

Typically yaw dampers are designed using yaw rate as the sensed output and rudder as the input.

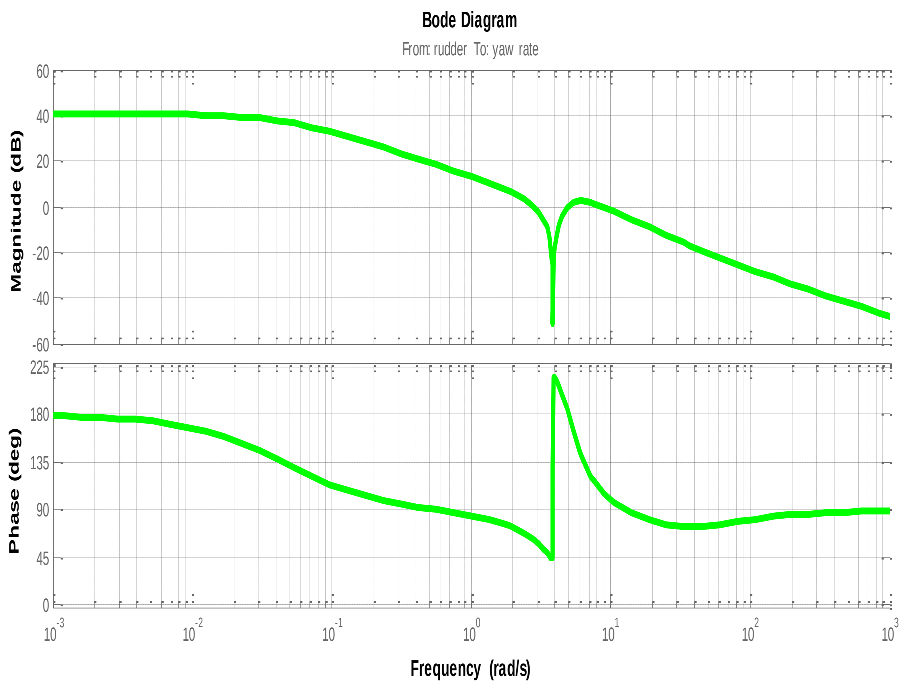


Figure 5.26: Bode diagram of yaw damper system.

From this Bode diagram, it can see that the rudder has significant effect around the lightly damped Dutch roll mode (that is, near ω = 5 rad/sec).

- **Root Locus Design**

A reasonable design objective is to provide a damping ration $\zeta > 0.35$ with a natural frequency $\omega_n < 3.0$ rad/sec. Since the simplest compensator is a static gain, first try to determine appropriate gain values using the root locus technique.

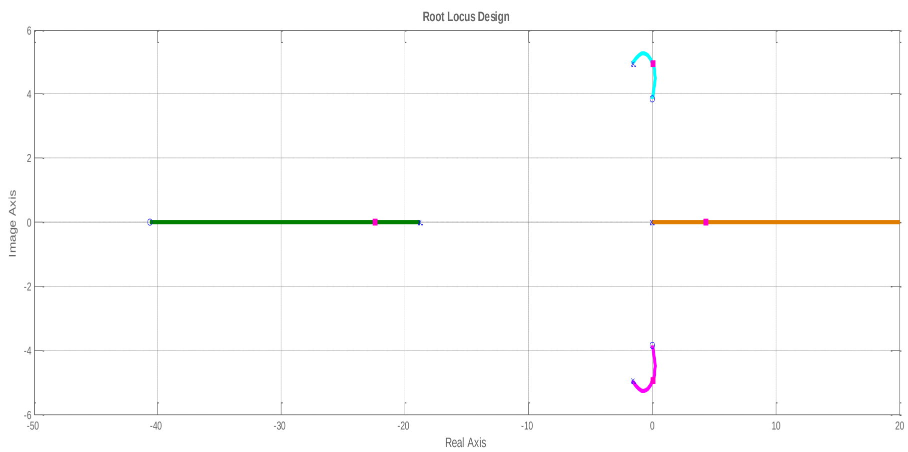


Figure 5.27: Root Locus yaw rate for rudder.

Root locus for negative feedback is shown as **figure 5.30** and shows that the system do not go stable almost immediately. If, instead, using positive feedback, may be able to keep the system stable.

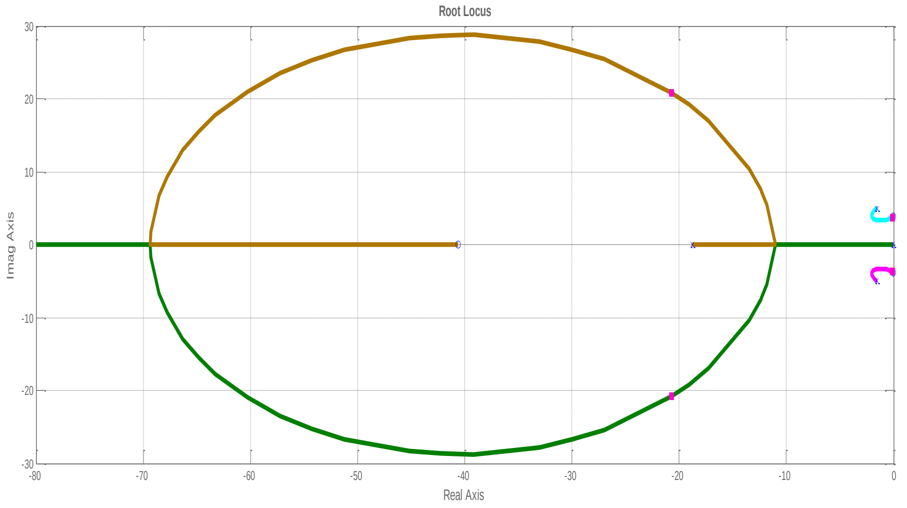


Figure 5.28: Root locus for negative feedback.

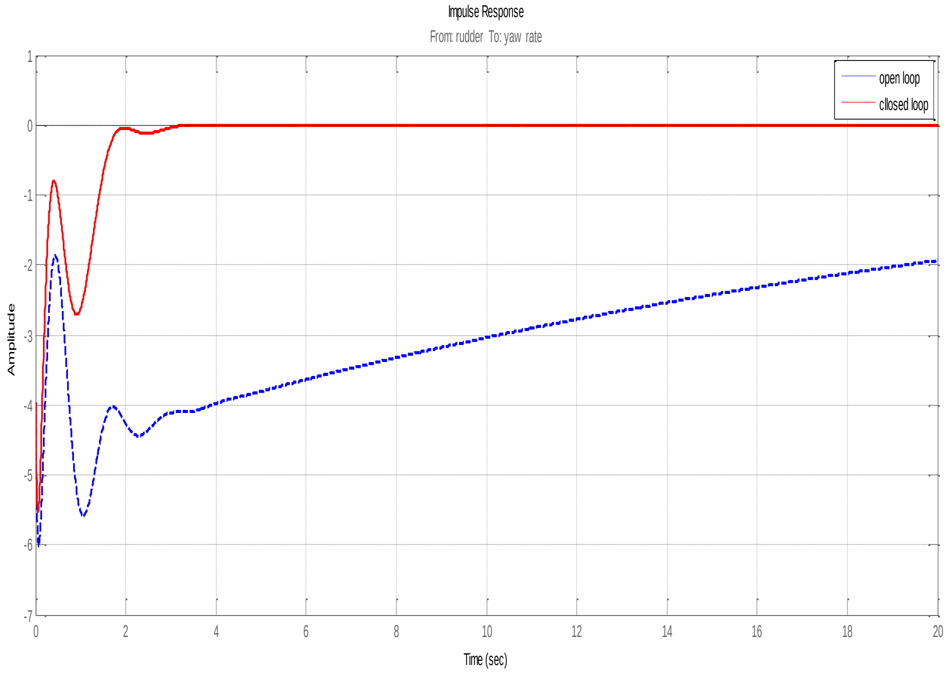


Figure 5.29 : Impulse response for system.

The above response in **figure 5.29** looks pretty good.

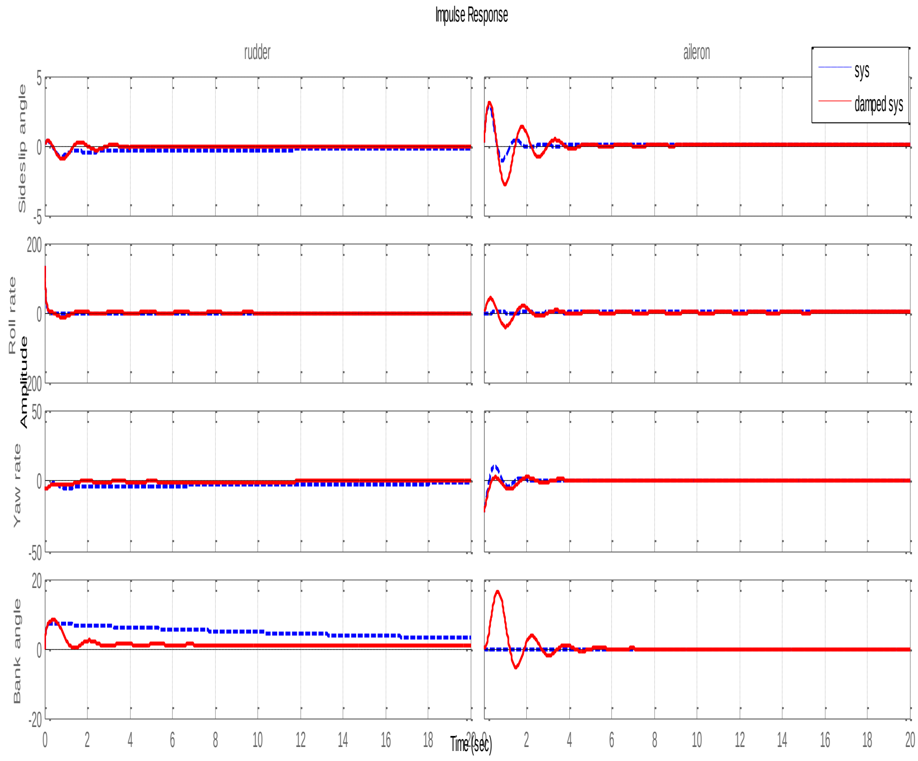


Figure 5.30 : Impulse response for damped system.

The yaw rate response is now well damped.

5.7 Roll Controller

The block diagram of the system is shown in **Fig. 5.31**. It incorporates the yaw damper described in the previous section and adds two additional loops. The outer loop commands φ. The error is converted to a roll rate command by J_p, and it is the roll rate error that is then used to drive the aileron servo actuator.

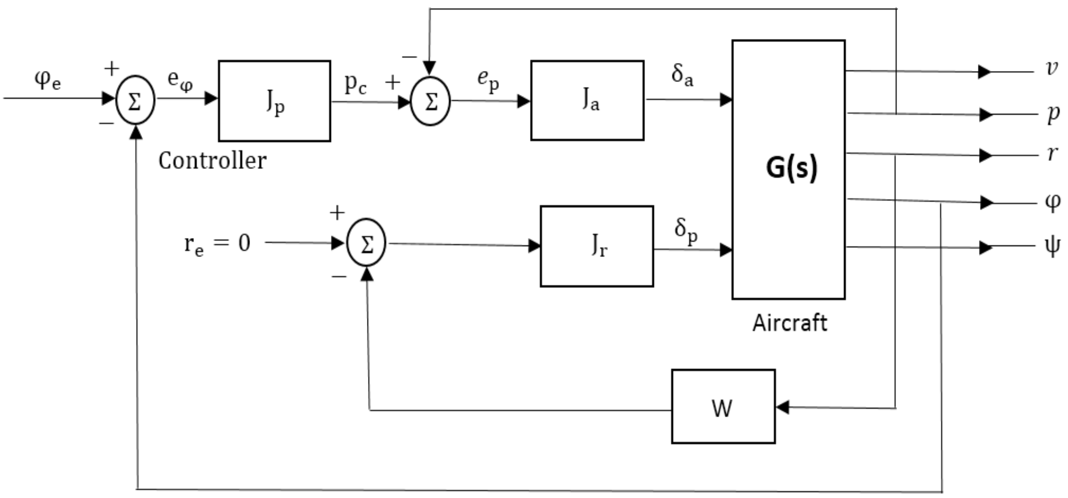


Figure 5.31 : Block diagram of Roll Controller.

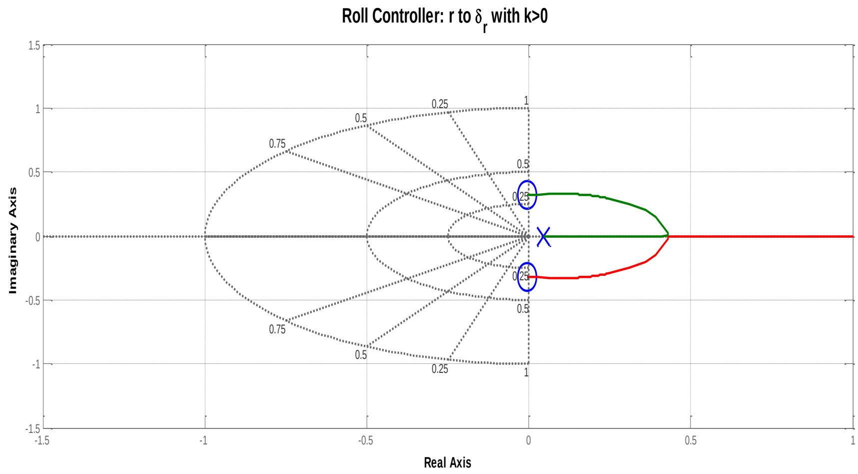


Figure 5.32 : Root locus for loop system_1.

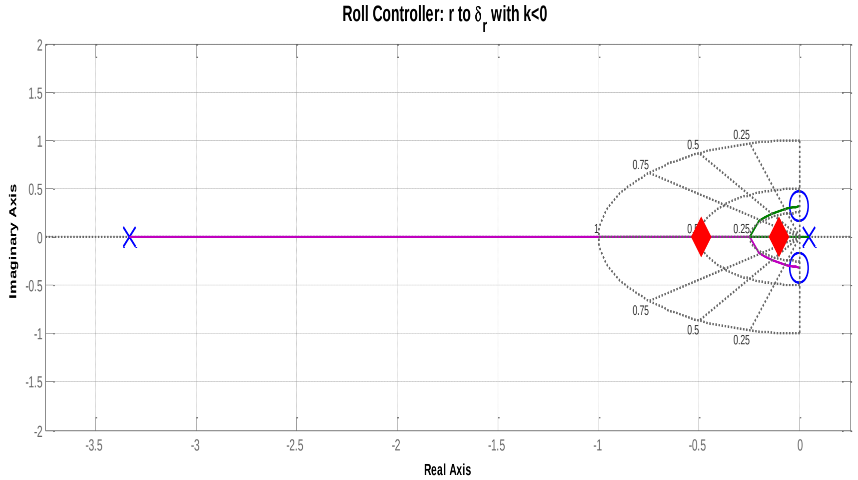


Figure 5.33 : Root locus for loop system_2.

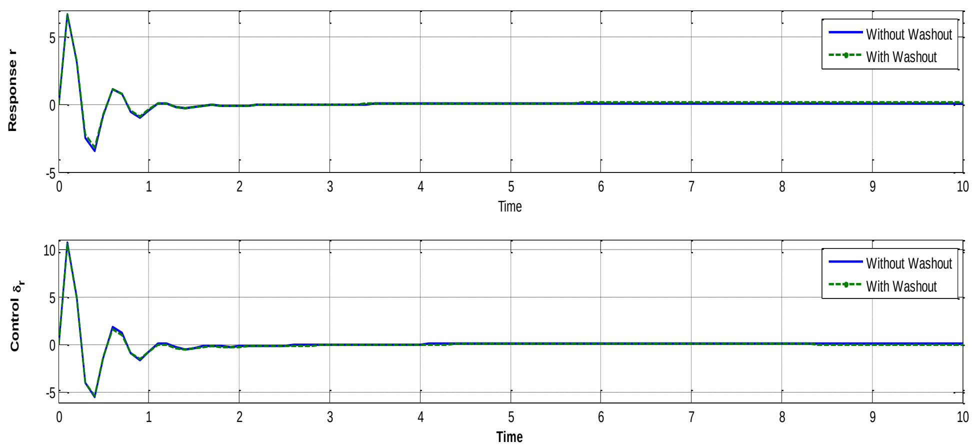


Figure 5.34 : Impulse response of closed loop system for roll controller.

To apply system with washout and without washout the filter, above system as shown **figure 5.31** response converges to 0 in both issues, extra control do not need to correct motion.

6. FLC SYSTEM BASED PID CONTROLLER USING GA

In this chapter, The controller system architecture and design are included in sections in terms of longitudinal and lateral motions respectively.

6.1 Fuzzy Controller Overview

One of the latest methodologies FLC is based on the so-called intelligent control which is defined as a combination of Control Theory and Artificial Intelligence. The basic idea of a fuzzy logic controller. which is to imitate the control actions of a human operator- can generally be represented as a collection of "if-then" rules.

Fuzzy logic's benefits are fault tolerance and the ability to provide accurate response to ambiguous data. A fuzzy logic controller provides an algorithm which control the linguistic control strategies, based on expert information. into an automatic control strategy.

The methodology of the FLC appears to be very useful and required results when the flight processes includes many differential equations are too complex for systems analysis.Generally. fùzzy logic control provides a clear application of the maximum input and maximum output (MIMO) systems.

Fuzzy logic controllers are basically non-linear, and general enough to provide the desired non-linear control actions through careful adjustment[24]. With the application of fuzzy sets and fuzzy operations, it is possible to implement a design for a fùzzy logic reasoning system as a controller. Scaling is performed to retain the signals between the input parameter and the output limits for the fuzzy process (rules).

The fuzzy controller is consist of four elements. All of the four elements are fuzzification, defuzzification, inference mechanism and rule base.A fuzzy control schematic block diagram is shown in **Figure 6.1**.

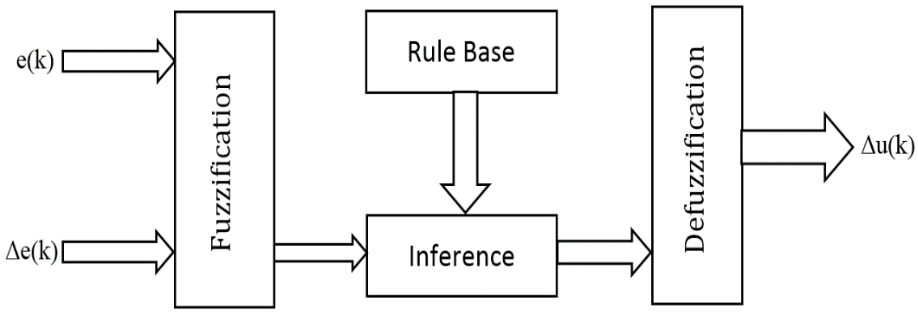


Figure 6.1 : Fuzzy Controller Schematic.

6.2 Fuzzy Logic Control Design

Fuzzy logic is the process of solving problems with uncertain (ambiguous) data to represent a crisp-logic system. In a fuzzy logic process, the three logic operations - and, or, and not- retum a degree of membership that is a number valued between 0 and 1 [25].

The determination of inputs and outputs, the rule base, the membership functions, scaling factors. the inference options. stability. etc., characterize the design of the fùzzy logic controller.Scaling is performed to keep the signals between the inputs and output limits for which the fuzzy rules are defined. The membership functions provide a smooth interface from the linguistic knowledge to the numerical process variables

- Fuzzy rules

Combine two or more input fuzzy sets, called antecedent sets. And associate with them an output, or consequent.

- Operation on Fuzzy Sets

a set U = { x_1, x_2, x_3, …} where x_i could be discrete or continuous. U is called the universe of discourse, x_i represents a generic element of U.A and B be two fuzzy sets in U with membership functions μ_A and μ_B respectively.

- Crisp Set

The crisp set can be descibed according to the traditional definition of a set in classical logic. Crisp sets have strict membership criteria according to which an object is either completely included or excluded from the set.

- Fuzzification

Fuzzification is the process for converting crisp numerical input to fuzzy logic value by combining actual input values with stored membership functions.

- Degree of Fulfillment

The degree of fullfilment for the antecedent of each rule is computed using fuzzy logic operators.

- Implication

The degree of filfilment is used to modify the consequent of the corresponding rule. This operation represents the if-then implication defined as a t-norm.

- Aggregation

The consequents of al1 rules are combined into a single fuzzy set. The aggregation operator depends on the implication function used.

- Defuzzification

Defuzzification is a process in which fuzzy output is converted into crisp,numerical results as indicated in Figure above. Defuzzification can be considered as an operator that replaces a fuzzy set a representative value.

- Fuzzy Set

An ordinary set divides the universe into those items that are completely in the set and those items that are completely outside of the set. Describing this phenomenon by assigning the value 1 to al1 those items which are members of the set and the value 0 to al1 items which are not mernbers of the set.

- Membership

Fuzzy sets have values between 0 and 1 that indicate the degree to which an element has membership in the set. At 0, the element has no membership; while at 1, it has full membership.

For analyzing the parameters of aircraft motion, the linguistic terms are presented by membership functions. For the longitudinal motion includes a pitch angle controller with pitch rate damping; for the lateral motion, a roll angle holds system and yaw damper.

6.3 Genetic Algorithm Overview

Genetic Algorithms (GAs) are a population-based search heuristic method that mimics the natural evolution process.GAs operate on a population of potential solutions by appliying the principle of survival of the fittest to produce better and better approximations to a solution [26].The GA starts with no knowledge which obtained from any system of the right solution and depends completely on responses from its factor of environment and operators of evolution as mutation, selection, crossover and reproduction to reach at the best solution. By starting at several independent points anywhere and searching in parallel layout, the algorithm process avoids local minima and converging to sub optimal solutions [27].

6.4 Genetic Algorithm Design

The GA is a multi-point one of the search method which implements the biological principle of the survival of the fittest ones. It combined the simulated mechanisms of biological crossover and mutation, to drive a population of potential solutions toward the most promising regions of the search space. A GA is typically started with a random population composed of between 0-100 individuals. Selected between any variable population or also known as mating pool is usually represented by a binary string or a real-valued number called a chromosome. A corresponding number is assigned individual the objective function called its fitness. The fitness of each chromosome being evaluated for the system is assessed and a survival of the fittest achievement of a solution used in different fields is applied. In this thesis, the magnitude of the error is used to assess the evolution of mating selection for fitness functions of each chromosome being evaluated for the system every time.

Main stages of a genetic algorithm are divided into three stages, these are called as *crossover, mutation and reproduction.*

6.4.1 Population size

There are many important step in GA and one of them is determining the number of population. it is suggested that the safe population size is scaled from 10 to 300. In this thesis an initial population of 80 were used for fuzzy pid also initial population of 250 were used for finding value of pid as stability and the result observed.

6.4.2 Calculation of the fitness function

The fitness of each chromosome is evaluated by converting the binary to real numbers. The GA uses the scaled fitness values to select the parents of the next generation. In this thesis, the gains of the PID controller are optimally tuned by using the GA. all the parameters of the solution, each chromosome is compared with the other to obtain the fitness rating.

- **Reproduction**

The fitness value for selected each chromosome being evaluated for the system during the phase of reproduction is set. Fitness value for each selected fitter chromosomes used in the selection process to ensure that bias towards fitter individuals. Section is proportional to the size criteria for the appropriateness of the individual. Ensure the possibility of an individual is more likely to leave a fitter offspring individual is selected so related fitness [28].

- **Crossover**

Once the selection process is completed, the crossover which is a genetic operator used to vary the programming of a chromosome is initiated. Depending on how the chromosome represents the solution, the crossover operations swaps certain parts of the two selected strings which is not possible a direct swap in a bid to capture the good parts of old chromosomes and create better new ones from applied for ordered chromosomes. Probability chromosomes are divided actually into two means.And one of them 0% means that the offspring. will be exact replicas of their parents and the other one probability of 100% means that each generation will be composed of entirely new offspring [28].

- **Mutation**

Using selection in which individual genomes and crossover will generate a large amount of different strings. But by this light it can occur two main problems with this:

- It can be chosen different population both Fuzzy PID and PID using GA depending on the selected initial population, if there is not enough diversity because of initial population in the initial strings it can occured entire space of problem to ensure the Genetic Algorithm searches.
- The initial population due to genetic algorithms can converge on a bad choice as sub-optimal strings.

Above mentioned items may be overcome by this mutation in genetic algorithm.

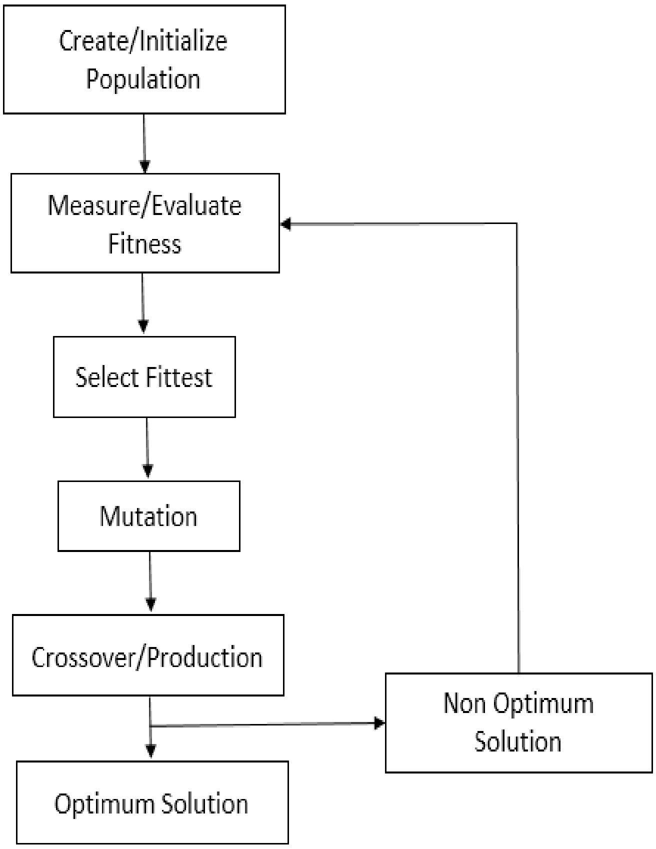


Figure 6.2 : Genetic algorithm process flowchart

6.4.3 Elitism

During the crossover and mutation operation there is a chance that the optimal solutions may be lost, as there is no guarantee that these operators will save the fittest values. Once the new population is created, it is examined to verify whether the optimal solution has been saved.

6.5 Genetic Algorithm Based Optimal Tuning Of the Gains of the PID Controller

The best model parameters are then passed directly to the GA tuning loop, which uses them to simulate the tuning process. Once the GA has gone through its evolutionary process to optimize the performance of the PID controller, the PID controller gains are implemented on the real process [29].

Table 6.1 : Minimizing performance criterion.

Name of criterion	Formula
Integral of the absolute value of of the error IAE	IAE = $\int_0^{\infty}[e(t)]dt$
Integral of the squared error ISE	ISE = $\int_0^{\infty}[e(t)]^2$dt
Integral of the time weighted absolute error ITAE	ITAE = $\int_0^{\infty} t[e(t)]dt$

In the experiment, the fitness is evaluated using ITAE in this thesis. This criteria balances error size and duration and avoids positive and negative errors canceling. Controller parameters are picked to minimize the integral. The best tuning results in the smallest ITAE.

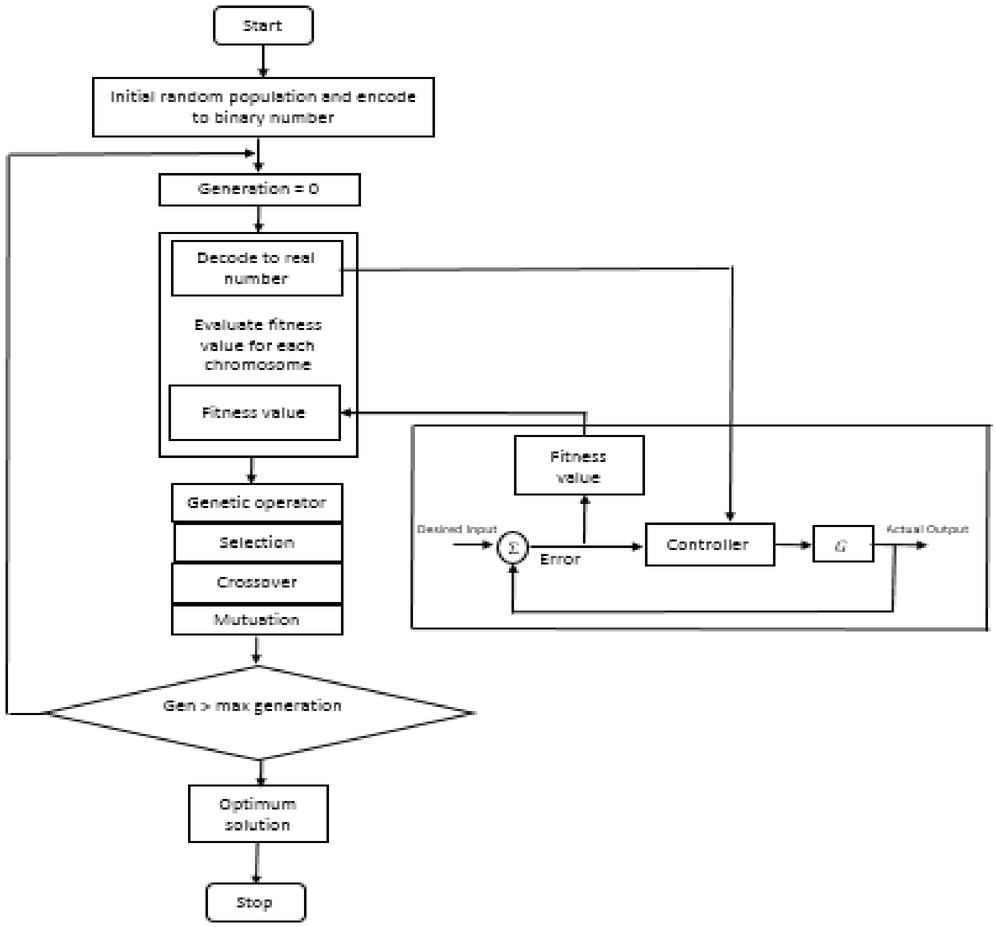


Figure 6.3: Objective function PID control system flowchart of Genetic Algorithm.

6.5.1 Optimal tuning of the gains of the pid controller using genetic algorithm

The following steps are involved in the optimal tuning of the gains of the PID controller, using the GA:

- Step 1 : Initialise the population size. Each variable in the minimization problem is called a gene. In the thesis, there are three genes, i.e. kp, ki and kd the population size is 80.
- Step 2 : Evaluate the objective/function of all the chromosomes.
- Step 3 : Evaluate the fitness function. Since the GA can be used only for maximization problems, the fitness function is evaluated as given below Fitness function = 1/ (1+F (φ))
- Step 4 : Generation of off spring.
- Step 5 : Selection of the parent chromosome for mutation and crossover.
- Step 6 : Perform the cross over and mutation operations.
- Step 7 : StwEvaluate the new population with the selected members of the old population and generate a new one.
- Step 8 : Terminate the process if the optimal gains are achieved; else go to step 2.

6.5.2 Initialization of the GA

The population size, variable bounds and the evaluation functions are initialized for performing the genetic algorithm. utilizing for the initialization of the parameters, as given codes below.After several iterations, the size of the population is chosen as 80. Since the GA has been used to optimize the gains of the PID controller, three strings have been assigned to each member of the population. The objective function is also defined for the optimization of the gains of the PID controller, and it is evaluated using the code ‘evalFn’. The term ‘option’ is used to encode the real number and the floating point value.

```
%%Initialising the genetic algorithm
populationSize=80;
variableBounds=[0 40;0 40;0 40];
evalFN='Fuzzy_PID_objfun_ITAE';
evalOps=[];
options=[1e-6 1];
initPop=initializega(populationSize,variableBounds,evalFN,...
evalOps,options);
```

6.5.3 Setting the parameters of the GA

The parameters of the GA are defined as given codes below. The entire search space for the operation of GA is defined on the bounds by changing.The starting population and the termination criteria are assigned as ‘startpop’ and ‘termFn’ respectively, to start and to terminate the operation of the GA. In this thesis, the GA gets terminated when the number of generations reaches the maximum value.

```
%%Setting the parameters for the genetic algorithm
bounds=[-100 100;-100 100;-100 100];
evalFN='Fuzzy_PID_objfun_ITAE';
evalOps=[];
startPop=initPop;
opts=[1e-6 1 0];
termFN='maxGenTerm';
termOps=220;
selectFN='normGeomSelect';
selectOps=0.08;
xOverFNs='arithXover';
xOverOps=4;
mutFNs='unifMutation';
mutOps=8;
```

6.5.4 Operation of the GA

The GA starts compiling a source code step by step, using the command given as below. The GA evaluates the parameters of system which defined to command window already assigned, and undergoes a number of iterations till it fulfills the criteria described in the termination function [30].

```
%Iterating the genetic algorithm
[x,endPop,bPop,traceInfo]=ga(bounds,evalFN,
evalOps,startPop,opts,...
termFN,termOps,selectFN,selectOps,xOverFNs,
xOverOps,mutFNs,mutOps);
```

6.5.5 Defining the objective function of the ga

The objective function can be to minimize the error of system which obtained from nonlinear system’s output. Each chromosome is passed through the objective function to be optimised for processing.The chromosomes needed to processing are evaluated and assigned with a number to represent their fitness value. The GA uses this fitness value of converting the binary to real numbers to create the new population contains the fittest members in process [30].

```
%Placing Fuzzy Controller in unity feedback system with 'sysrl'
%sys1=series(rl_pid,sysrl);
%rl_sys=feedback(sys1,[1]);
%Creating the Fuzzy PID controller from GA results

ga_pid=tf([x(1) x(2) x(3)],[1 0]);
ga_sys=feedback(series(ga_pid,sysrl),1);


function [x_pop, fx_val]=Fuzzy_PID_objfun_ITAE(x_pop,options)
global sys_controlled
global time
global sysrl
%%%
Kp=x_pop(2);
Ki=x_pop(3);
Kd=x_pop(1);
%Creating the Fuzzy_PID controller from current values
pid_den=[1 0];
pid_num=[Kd Kp Ki];
pid_sys=tf(pid_num,pid_den); %systems of controller
%Creating Fuzzy_PID feedback loop
sys_series=series(pid_sys,sysrl);
sys_controlled=feedback(sys_series,1);
```

The values of error (e[k]) and its changes of error (Δe[k]) which are acceptable as inputs of system during the system operation form the crisp inputs of the system. These two inputs in the system defined shown (**6.1**) and (**6.2**)

$$e(k) = r(k) - y(k) \tag{6.1}$$

$$\Delta e(k) = e(k) - e(k-1) \tag{6.2}$$

Above expression r(k), y(k) and k are described as reference input, These crisp inputs e(k) and Δe(k) , which the actual system output are converted to fuzzy membership value as output of system .

6.6 Longitudinal Motion

An investigation of the performance of a fuzzy PID controller using GA , Fuzzy PID, PID controller using GA and Ziegler Nichols was carried out. The effects of these controllers were observed with particular attention given to longitudinal-lateral stability control. The pitch of the fuzzy PID controller serves as an inner loop for the longitudinal motion of the aircraft. The simulation results of the theta and elevator deflection parameters are almost the same in both controllers, though the Fuzzy PID controller seems a little faster than others controllers.

6.6.1 Building the Fuzzy inference system

There are two major types of fuzzy controllers namely Mamdani and Takagi-Sugeno.Both of them is used in this thesis and Mamdani fuzzy inference system with Fuzzy PID Controller system is used to be obtained best solutions. The classification mainly depends on the output form; Mamdani type produces output in the form of fuzzy sets while Takagi-Sugeno produces output in the form of functions or LTI systems.

- **Mamdani Fuzzy Controllers**

In Mamdani type model, the inputs and outputs are defined in fuzzy sets through membership functions which also define the range of the inputs and outputs beyond which the controller will be futile. The basic process involves different stages such as 1) fuzzification of crisp values of the input fuzzy sets, 2) fuzzy inference where the fuzzy sets are mapped according to the fuzzy rules, and 3) defuzzification. The controller process has been shown in Figure.

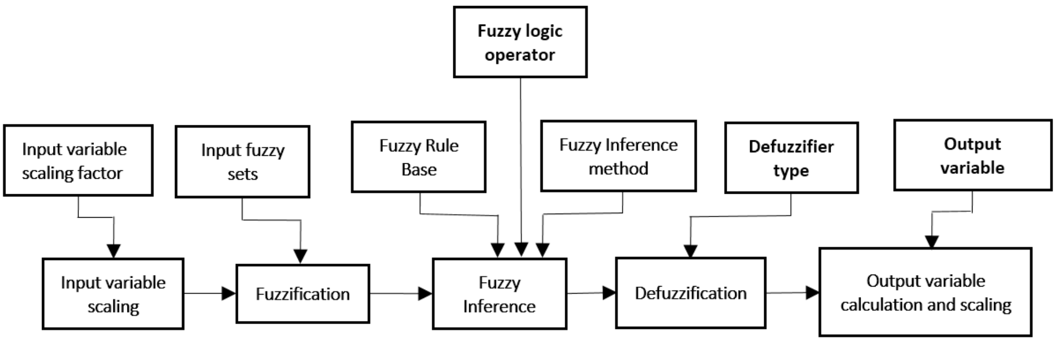


Figure 6.4 : Structure of a typical MISO Mamdani fuzzy controller.

The rules are defined in a linguistic manner which can be quantified mathematically later

IF $\lambda 1$ is $\Lambda 11$ AND $\lambda 2$ is $\Lambda 12$ …. AND λv is $\Lambda 1v$ THEN ω is $\Omega 1$

IF $\lambda 1$ is $\Lambda 11$ AND $\lambda 2$ is $\Lambda 12$ …. AND λv is $\Lambda 2v$ THEN ω is $\Omega 2$

…

IF $\lambda 1$ is $\Lambda n1$ AND $\lambda 2$ is $\Lambda n2$ …. AND λv is Λnv THEN ω is Ωn

where the λj ($j = 1,2,\ldots v$) is input to the fuzzy system, Λji ($i = 1,2\ldots,n$) are input fuzzy sets, v is the number of inputs, n is the number of rules, ω is the output of the fuzzy system and Ωj is the output fuzzy set. The fuzzy sets are represented through membership functions. There are number of different membership functions expressed in various shapes such as Triangular, Gaussian and Trapezoidal etc. In this thesis the membership functions are denoted by μ. The function of the fuzzy inference is to produce an output fuzzy set from the defined rules. Final stage involves with defuzzification of the output fuzzy sets computed in the fuzzy inference [31].

- **Fuzzy Logic Controller**

One of system controller is fuzzy logic controller which can be achieved best performance like faster response, steady space response in terms of PID controller using genetic algorithm.It is considered that a better control system to provide advantage of FLC and PID control system can be achieved adequately by integrating these two techniques [32]. The PID parameters in **figure 6.10** are tuned by using fuzzy inference as shown in **figure 6.11** which provides a nonlinear mapping from the error (e(k)) and derivation of error (Δe(k)) to PID parameters.

➢ **Membership function**

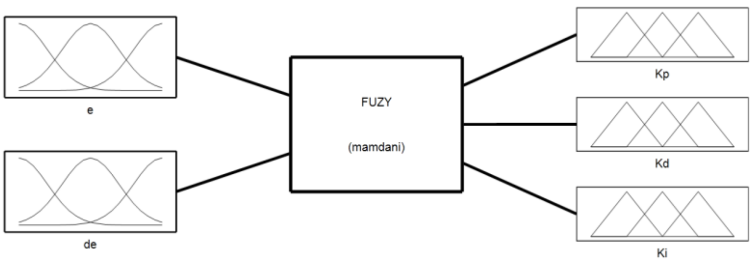


Figure 6.5 : Fuzzy inference system of Fuzzy PID.

The membership functions used were simple triangular {y = trimf(x,[a b c])} and Z-shaped {y = zmf(x,[a b])} functions with different range of angles for inputs and output as shown in **Figure 6.5**.

Each input variables which have error and deviation of error for the fuzzy sets consist of seven linguistic variables:

e = {Negative Big, Negative Middle, Negative Small, Zero, Positive Small, Positive Middle, Positive Big}

de = {Negative Big, Negative Middle, Negative Small, Zero, Positive Small, Positive Middle, Positive Big}

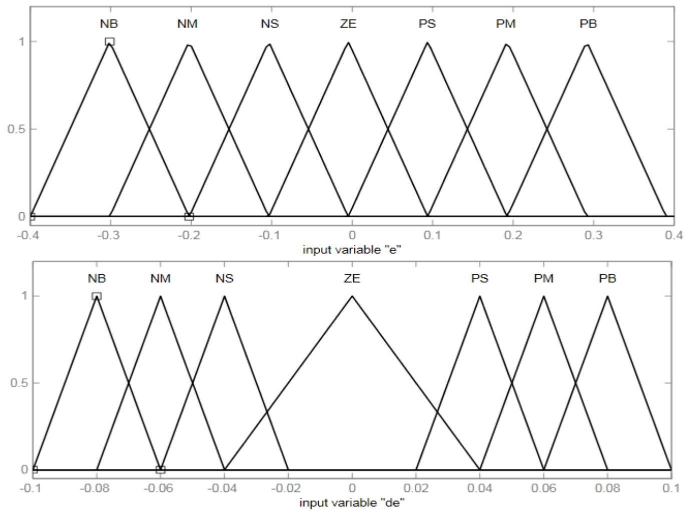


Figure 6.6 : Membership function of e(t) and de(t).

The methods used for fuzzification and defuzzification are shown in **Table 6.2** which is the default setup in fuzzy interface system in Matlab.

Table 6.2 : FIS Method.

Fuzzy Inference System	
AND method	Min
OR method	Max
Implication	Min
Aggregation	Max
Defuzzification	Centroid

The membership functions of outputs PID parameters (K_p, K_i, K_d)are shown in **Figure 6.8**, **Figure 6.9**, **Figure 6.10**.The linguistic variables of outputs below are assigned as:

K_p = {N, Z, P}, K_i = {N, Z, P}, K_d = {N, Z, P}

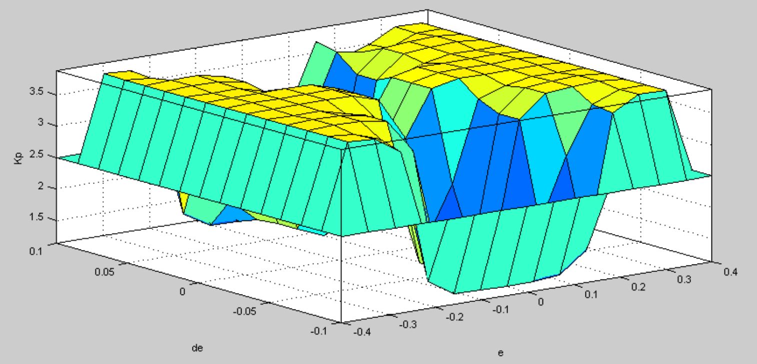


Figure 6.7 : Surface of membership function of K_p.

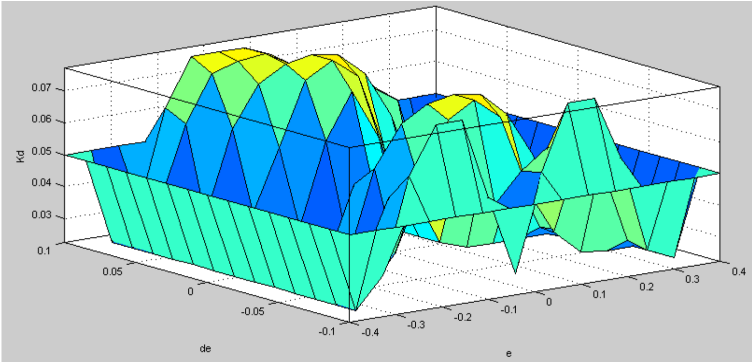


Figure 6.8: Surface of membership function of K_i.

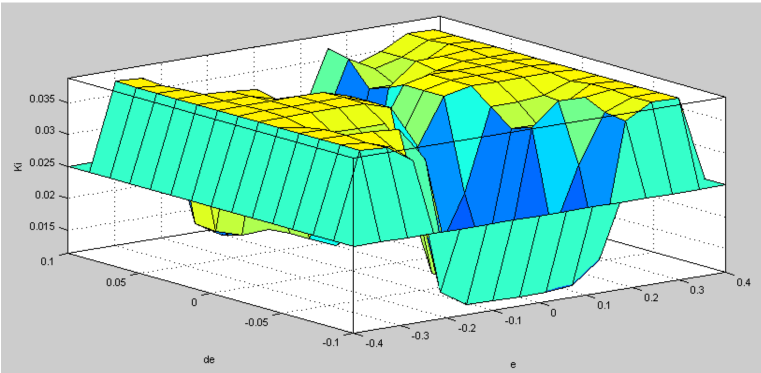


Figure 6.9 : Surface of membership function of K_d.

The fuzzy rules are shown in **Table 6.3**, here the abbreviations of the membership functions denoting NB – negative big, NM – negative medium, NS – negative small, AZ – around zero, PS – positive big, PM – positive medium and PB – positive big.

- **Rules**

Table 6.3: Rules of FIS.

	e	NB	NM	NS	ZE	PS	PM	PB
de	K_p K_i K_d							
NB		PNP	ZPZ	NPN	NNN	NPN	ZPZ	PNP
NS		PNP	PZP	ZPZ	NPN	ZPZ	PZZ	PNP
NM		PNP	PNP	ZZZ	NPN	ZZZ	PNP	PNP
ZE		PNP	PNP	PNP	ZZZ	PNP	PNP	PNP
PS		PNP	PNP	ZZZ	NPN	ZZZ	PNP	PNP
PM		PNP	PZZ	ZPZ	NPN	ZPZ	PZP	PNP
PB		PNP	ZPZ	NPN	NPN	PNP	ZPZ	PNP

- **PID Controller**

K_p, K_i, K_d gain of nonlinear is nonlinear function of controlling error regarding function includes integrated absolute value of error of signal which is the difference which is the difference between wanted signal and actual signal as small as possible. A PID controller can be described as:

$$u(n) = K_p e(t) + K_i \int_0^{\tau} e_{\tau} d\tau + K_d \frac{d}{dt} e(t)$$

The selection of gains for the PID controllers can be determined by a method developed by Ziegler and Nichols and Genetic Algorithm analysis , who studied the performance of PID controllers by examining the integral of the absolute error. Control of higher order process, large uncertainties systems, lightly damped oscillatory systems, highly coupled multi-input, time delays system, multi-output systems like lateral controller. To overcome these highly systems requires the use of more complicated control methods like adaptive knowledge based controllers. These sophisticated limitations with FLC can be minimized and overcome.

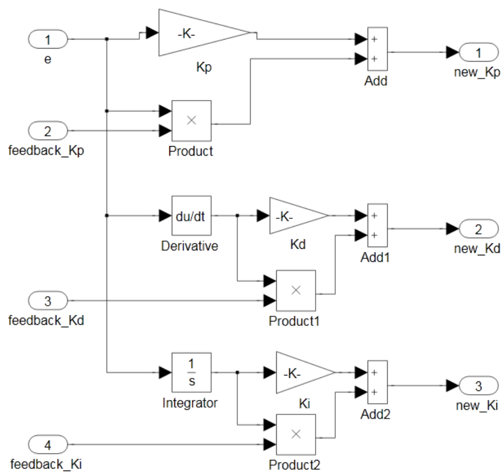


Figure 6.10 : PID controller for Fuzzy PID system.

- **Fuzzy – PID Controller**

Fuzzy logic controller is considered as outcome method to deal with higly disturbances includes wind effect and uncertainties that includes the Fuzzy Inference System (FIS) to tune the parameters of K_p, K_i, and K_d, using genetic algorithm code mentioned above and other methods according to error (e) and derivative of error (de). FLC and PID controller expressed earlier sections in a detail are combined to the system in this chapter. The FLC output is fed as input of PID controller.It is necessary to parameter tuning process of PID due to obtaining satisfactory response against not sometimes tuned with unpredictable parameter. Also the scaling factors, K_1 and K_2 are chosen to achieve both satisfactory set of time domain parameters and for speed up the system. Having been assigned a scaling factor of 10000.

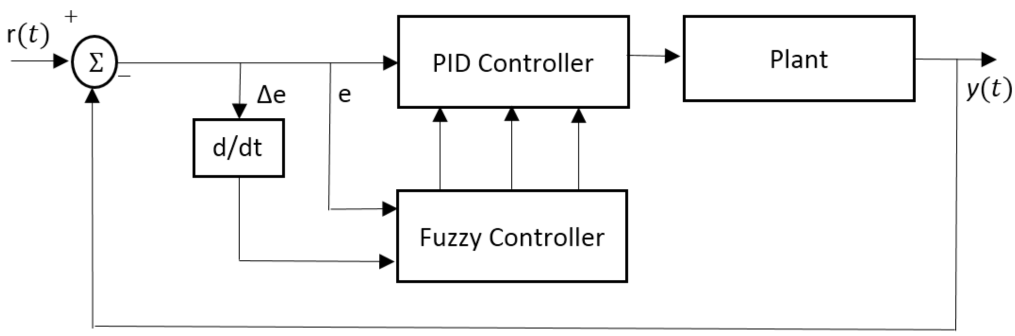


Figure 6.11 : Structure of Fuzzy-PID Controller.

6.6.2 Modelling of pitch control system

Pitch control is a longitudinal problem in UAV and this issue controls the pitch of an aircraft. Pitch is controlled by the rear part of the tail plane's horizontal stabilizer being hinged to create an elevator. By moving the elevator control backwards the controlled system move the elevator up (a position of negative camber) and the downwards force on the horizontal tail is increased.The pitch angle of an aircraft is controlled by adjusting the angle and therefore the lift force of the rear elevator.The fuzzy-PID controller is used for best state space response performance and stability of the pitch control system.System response with various inputs namely pitch angle are observed in this thesis.The system output for the closed loop system response changes under certain operating [33].

For all responses, the reference value is selected as 0.087 radians which is equal to 5 degrees. First of all, the many various FLC membership functions are examined and the best membership function for pitch control system is determined.

In **Figure 6.11**, two inputs have been applied to fuzzy- PID controller which is the error (e) as one of inputs that computed by comparing based on the reference point, desired pitch angle with the output of system and the change of error, δ_e last of inputs which generated by the derivation of the error (Δe). As shown from **Table 6.4**, it can be observed that the pitch angle follows the reference value respectively. The fuzzy PID controller is able to give a good response little produce any overshoot.

Table 6.4 : Values of PID for Pitch Controller.

Controller	K_p	K_i	K_d
Fuzzy PID using GA	5.43403	0.5189	0.75810
GA PID	38.553050	0.609689	-21.387
Classic PID	2.28	7.6	0.171

Fuzzy PID control was expressed by Mamdani type which uses fuzzy sets to define the input variables but the output is defined by means of functions, also defining others control is shown as **figure 6.12** below.

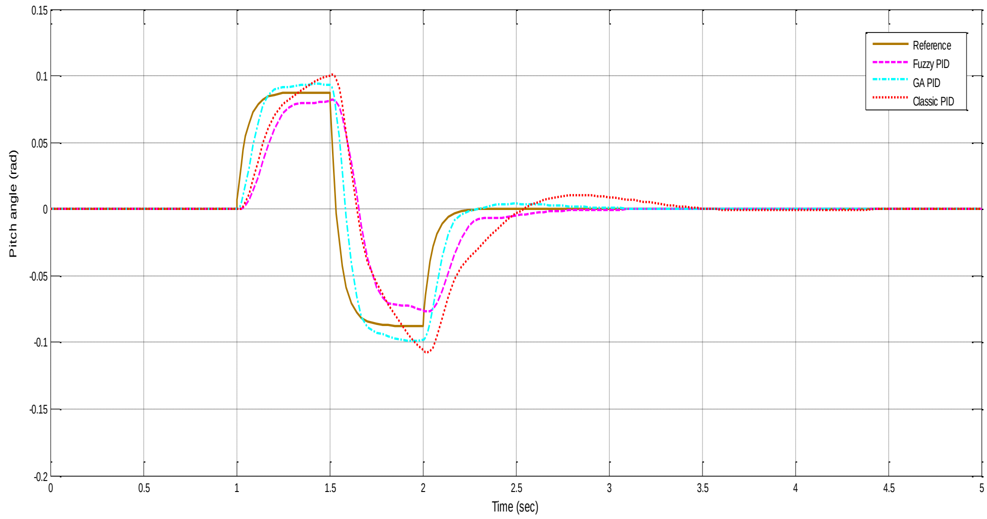


Figure 6.12 : Pitch controller result_1.

The response is faster than other controller compared to Fuzzy PID controller which offers a better response at nominal degree doublet excitation signal ±5 degree at constant altitude.

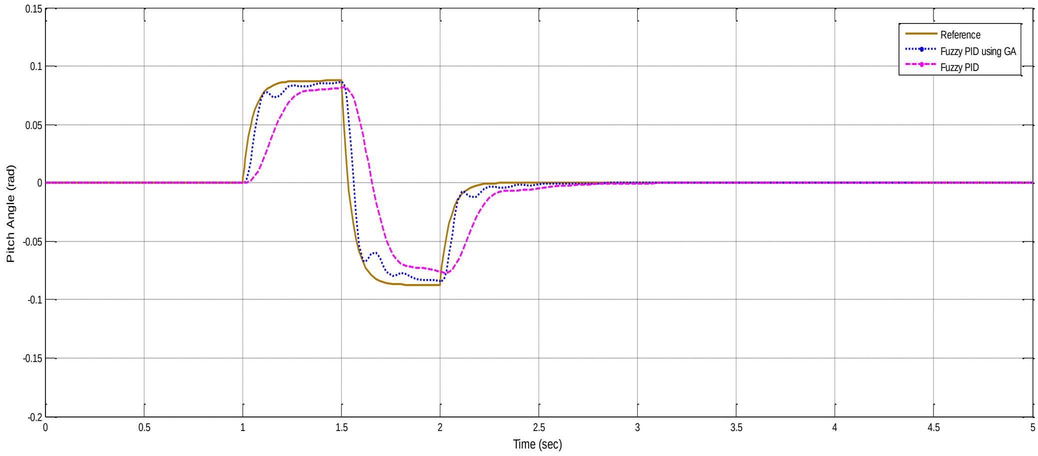


Figure 6.13 : Pitch controller result_2.

The response is faster than another compared to Fuzzy PID using GA controller at nominal degree doublet excitation signal ±5 degree at constant altitude.

Table 6.5 : Value of response characteristics for pitch controller_1

Response characteristics	**Fuzzy PID using GA**	**Fuzzy PID**	**GA PID**	**Classic PID**
Rise Time, T_r in sec	3.3202e-006	3.6270e-004	7.7746e-006	0.0065
Settling Time T_s in sec	2.4874	2.6913	2.8040	3.3680
Settling Min	-0.0845	-0.0770	-0.0988	-0.1076
Settling Max	-9.2455e-006	0.0819	0.0937	0.1008
Overshoot M_p in %	9.1406e+005	5.1549e+008	3.0531e+008	1.0345e+005
Peak Time, T_p in sec	1.5039	1.5117	1.9402	2.0205

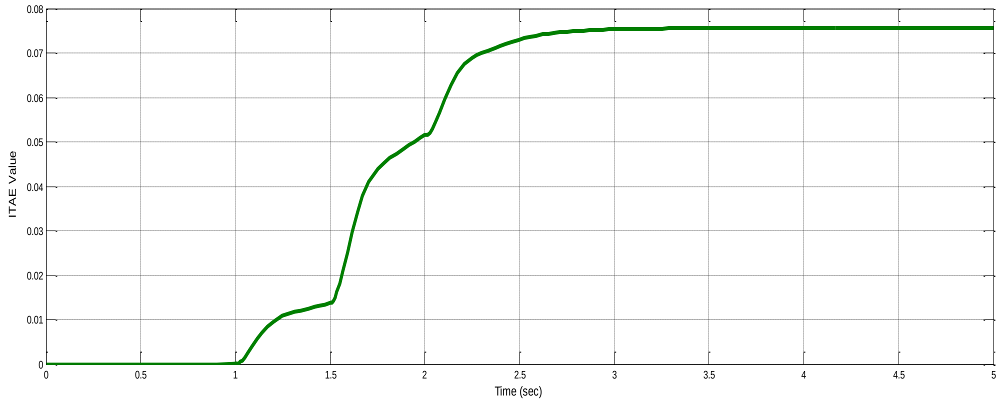


Figure 6.14 : ITAE value of pitch controller.

The step response for the approach in flight condition from dynamic equilibrium is illustrated below in **Figure 6.15**.

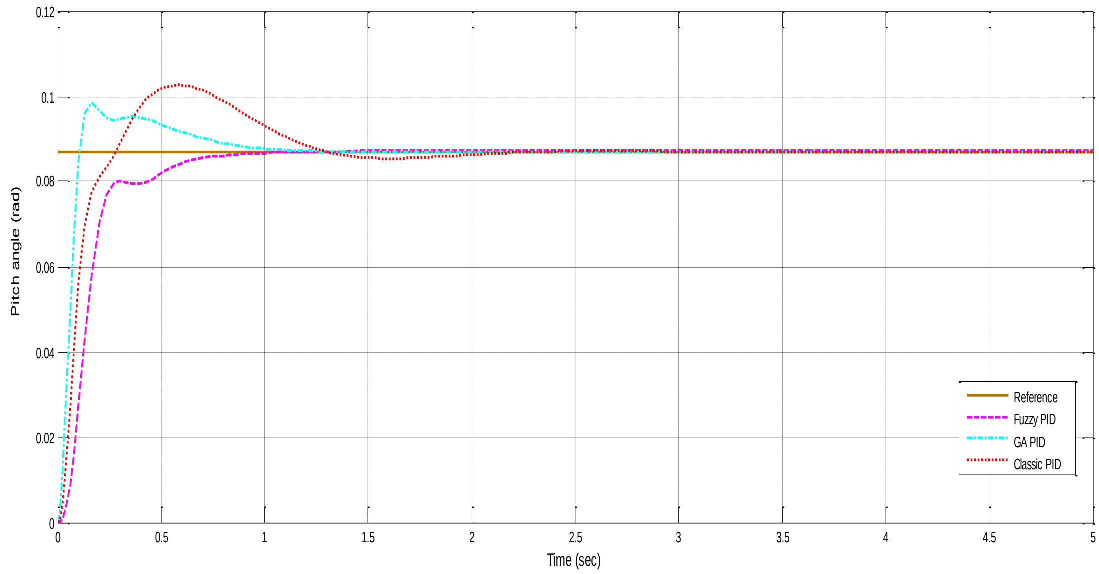


Figure 6.15 : Pitch controller result_3.

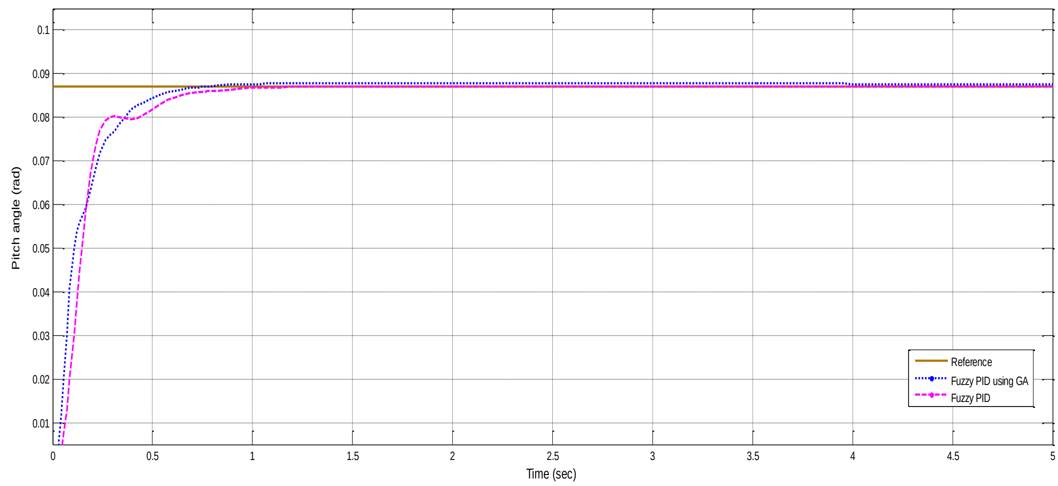


Figure 6.16 : Pitch controller result_4.

Notice the quick rise time is shown **Table 6.6** below in the fuzzy PID system for this condition, slight slower than the classic PID system, but settling time accurately is rather faster than the classic PID system . While the overshoot is higher and settling time is longer for classic PID system, they are rather great for the fuzzy PID system.

Table 6.6 : Value of response characteristics for Pitch Controller_2.

Response characteristics	Fuzzy PID using GA	Fuzzy PID	GA PID	Classic PID
Rise Time, T_r in sec	0.3041	0.1947	0.0722	0.1384
Settling Time T_s in sec	0.5992	0.6770	0.8345	1.1843
Settling Min	0.0802	0.0794	0.0870	0.0804
Settling Max	0.0878	0.0871	0.0989	0.1027
Overshoot M_p in %	0.1902	0.0041	13.6990	18.0291
Peak Time, T_p in sec	1.6248	2.4480	0.1663	0.5759

Fuzzy logic based intelligent controller design is introduced for controlling longitudinal pitch control of an aircraft. The proposed Fuzzy logic controller can effectively eliminate these dangerous oscillations and provides smooth operation in transient period.

6.7 Lateral Motion

The roll and yaw of the fuzzy PID controller serves as an inner loop for the lateral motion of the aircraft. The simulation results of the phi and aileron and rudder deflection parameters are almost the same in both controllers, though the Fuzzy PID controller seems a little faster than others controllers.

6.7.1 Modelling of roll control sytem

Roll control is a lateral problem in UAV and this issue controls the roll angle of an UAV in order to stabilize the system best responses when an aircraft performs the rolling motion which is described combination of translational and rotational. The input to the PID controller from output to fuzzy system block is the error, which is the difference between the input and the feedback from the plant. The PID controller block values can be genetic algorithm codes in Appendix C tuned and last one values manually (with method) tuned to get the best response for the system as shown **Table 6.7** below.

Table 6.7 : Values of PID for Roll Controller.

Controller	K_p	K_i	K_d
Fuzzy PID using GA	44.795164	99.69615	0.696616
GA PID	73.78419	99.7862	1.700567
Classic PID	3.24	7.44	0.35235

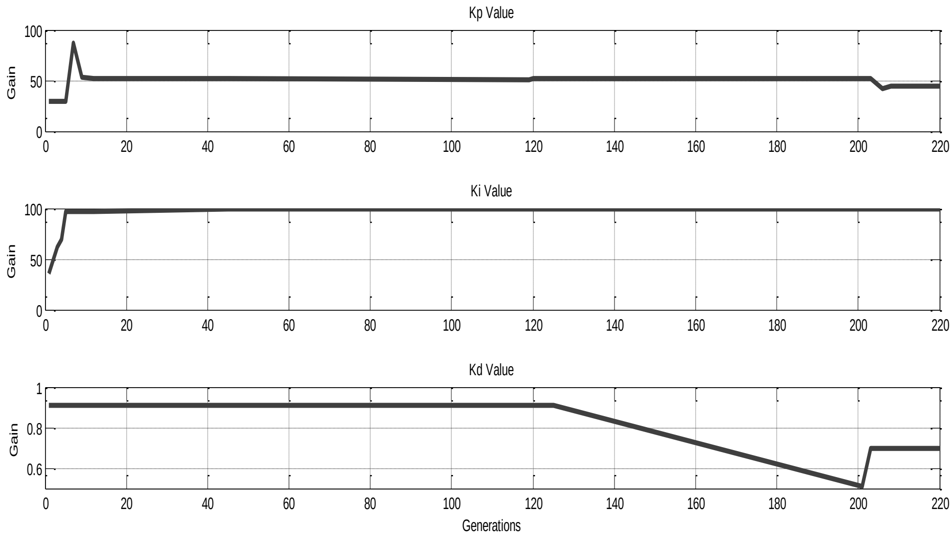


Figure 6.17 : Values of PID of Fuzzy PID for roll controller.

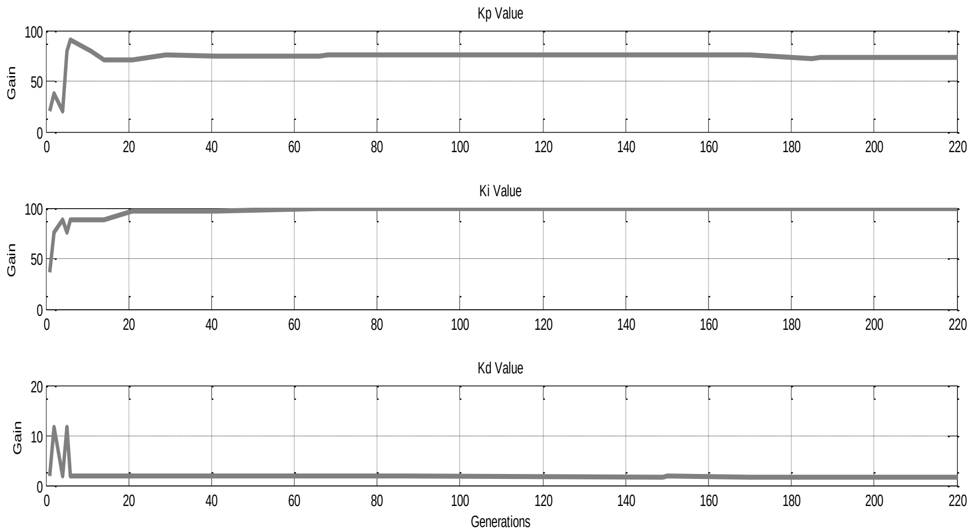


Figure 6.18 : Values of PID for roll controller.

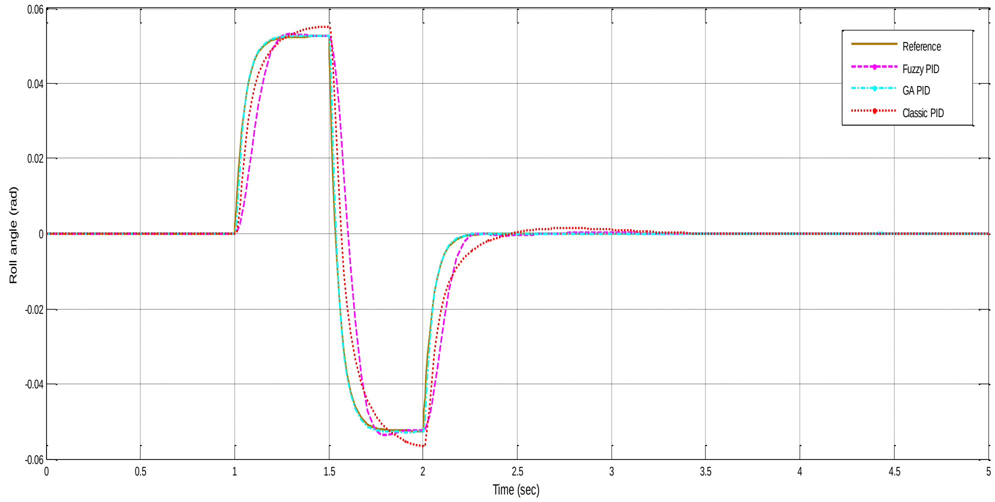


Figure 6.19 : Roll controller result_1.

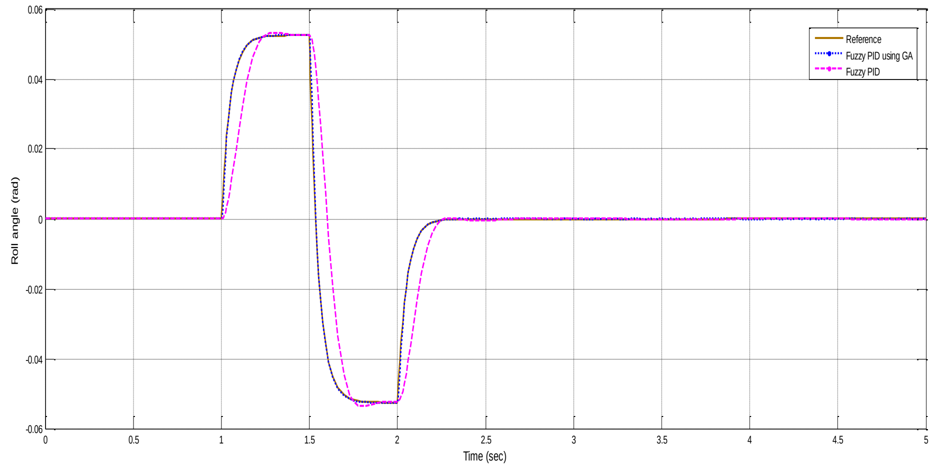


Figure 6.20 : Roll controller result_2.

The response is comparatively faster compared to PID controller at nominal degree doublet excitation signal ±5 degree at constant altitude.

Table 6.8: Value of response characteristics for roll controller_1.

Response characteristics	Fuzzy PID using GA	Fuzzy PID	GA PID	Classic PID
Rise Time, Tr in sec	2.2752e-005	1.1523e-005	1.1640e-005	1.2019e-005
Settling Time Ts in sec	2.1960	2.2356	2.1954	2.9779
Settling Min	-0.0526	-0.0536	-0.0526	-0.0566
Settling Max	0.0526	2.8147e-004	6.7288e-005	0.0014
Overshoot Mp in %	9.6825e+006	5.3826e+005	3.1782e+005	5.0794e+005
Peak Time, Tp in sec	2	1.7984	1.9852	2.0014

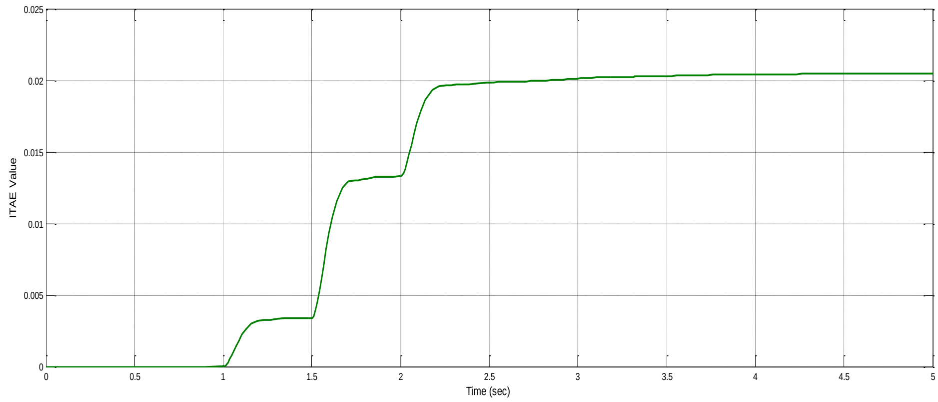


Figure 6.21: ITAE value of roll controller.

The step response for the approach in the flight condition is shown below in Figure **Figure 6.22** and **Figure 6.23**. Notice the quick rise time is shown in Table 6.9 the Fuzzy PID system for this condition, slight slower than the classic PID system, but settling time accurately is rather faster than the classic PID system . While the overshoot is higher and settling time is longer for classic PID system, they are rather great for the fuzzy PID system.

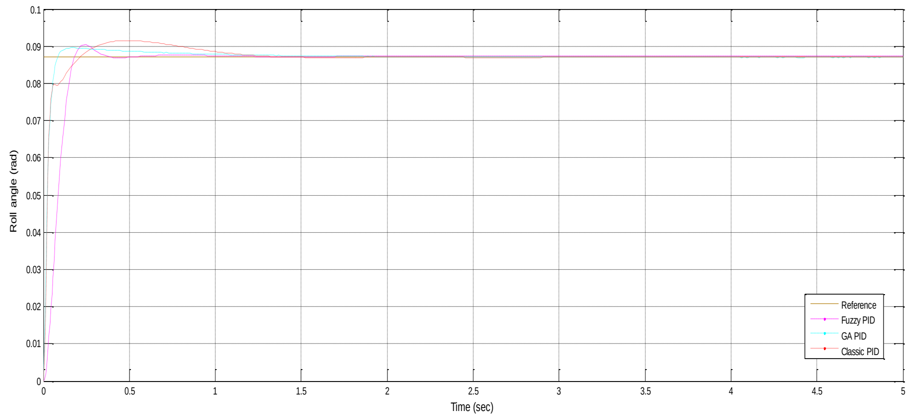


Figure 6.22 : Roll controller result_3.

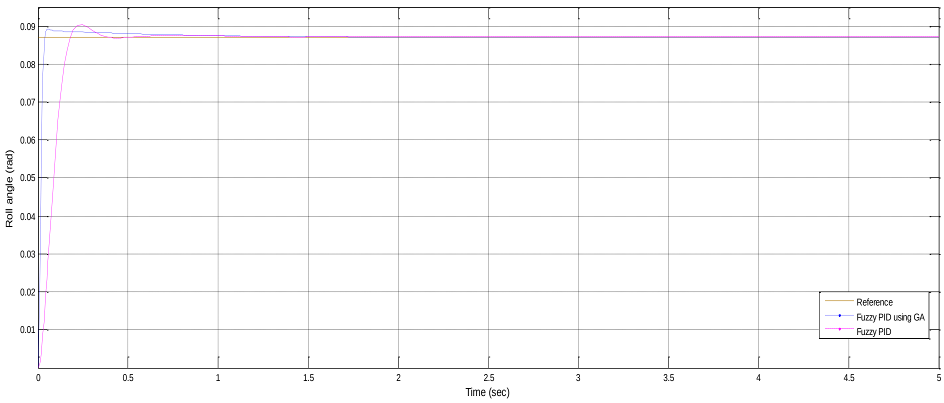


Figure 6.23 : Roll controller result_4.

Table 6.9 : Value of response characteristics for roll controller_1.

Response characteristics	**Fuzzy PID using GA**	**Fuzzy PID**	**GA PID**	**Classic PID**
Rise Time, Tr in sec	0.0235	0.1161	0.0452	0.0429
Settling Time Ts in sec	0.0923	0.2970	0.4480	0.9695
Settling Min	0.0832	0.0799	0.0807	0.0794
Settling Max	0.0892	0.0904	0.0895	0.0915
Overshoot Mp in %	2.5476	3.5329	2.8606	5.210
Peak Time, Tp in sec	0.0557	0.2463	0.1617	0.0915

Fuzzy logic based intelligent controller design is introduced for controlling lateral roll control of an aircraft. The proposed Fuzzy logic controller can effectively eliminate these dangerous oscillations and provides smooth operation in transient period.

6.7.2 Modelling of yaw control sytem

The yaw control loop can be configured either as a simple yaw damper which mentions in section 5 or as a combined yaw damper and side-slip controller. Because control of side-slip uses measured lateral acceleration, it will only work for those models that have enough fuselage side area to produce a measurable lateral acceleration when they side-slip.

Table 6.10 : Values of PID for Yaw Controller.

Controller	K_p	K_i	K_d
Fuzzy PID using GA	-24.26654	-99.6419	-5.08964
GA PID	-18.50308	-98.6151	-5.43244
Classic PID	0.506616	1.5352	1.5352

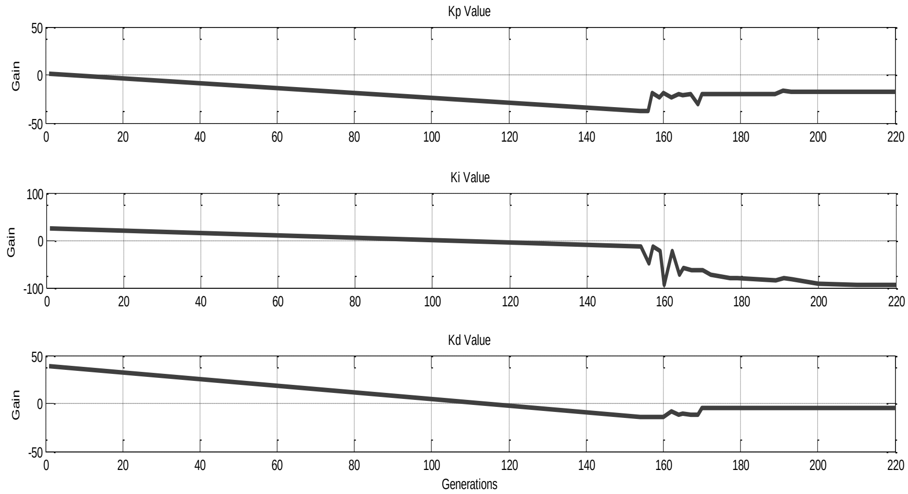


Figure 6.24: Values of PID of Fuzzy PID for yaw controller.

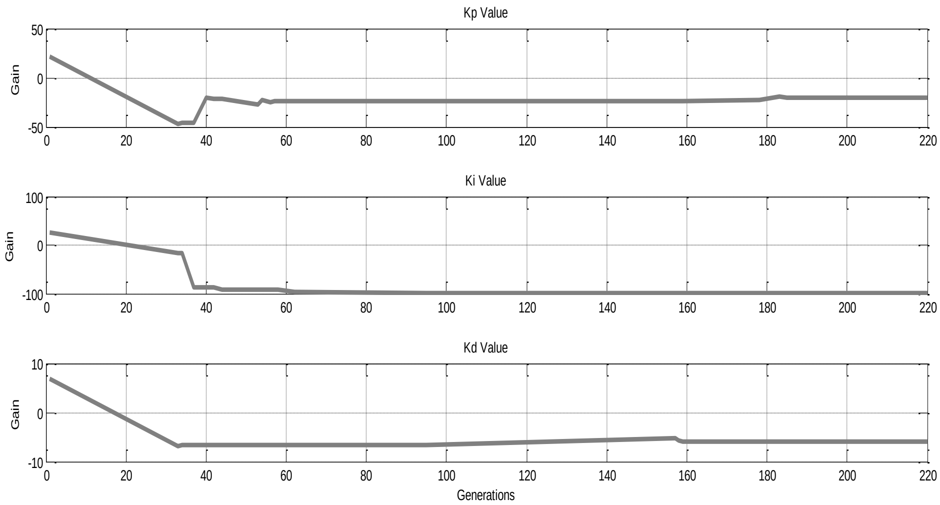


Figure 6.25: Values of PID for roll controller.

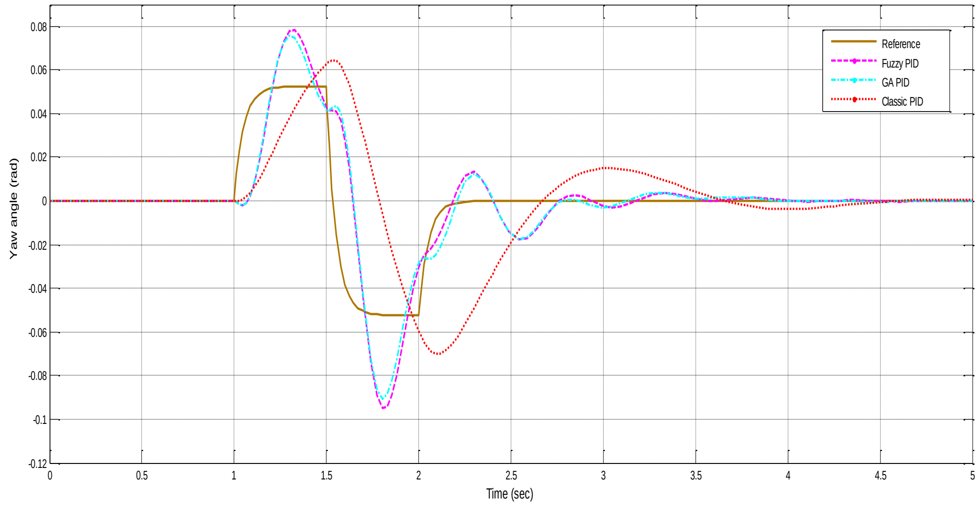


Figure 6.26: Yaw controller result_1.

Table 6.11: Value of response characteristics for yaw controller_1.

Response characteristics	Fuzzy PID using GA	GA PID	Classic PID
Rise Time, Tr in sec	4.1952e-004	0.0037	0.0247
Settling Time Ts in sec	3.4451	3.7589	4.4674
Settling Min	-0.0949	-0.0908	-0.0699
Settling Max	0.0782	0.0755	0.0645
Overshoot Mp in %	9.7277e+004	1.2961e+005	8.7528e+003
Peak Time, Tp in sec	1.8026	1.8026	2.0910

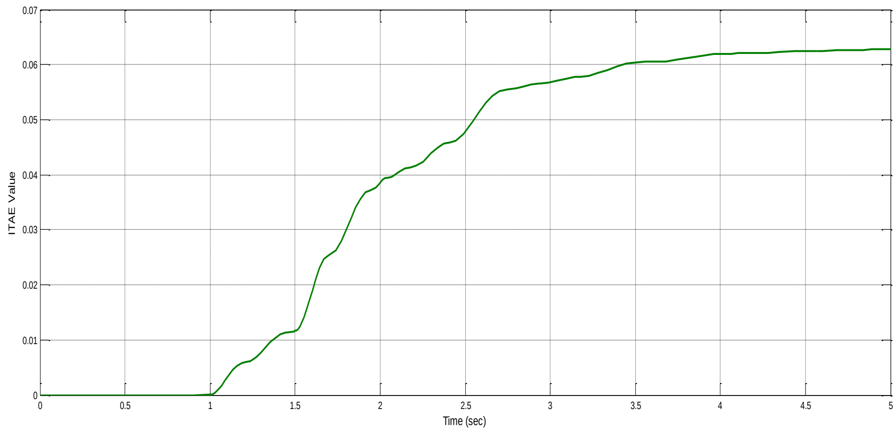


Figure 6.27: ITAE value of roll controller.

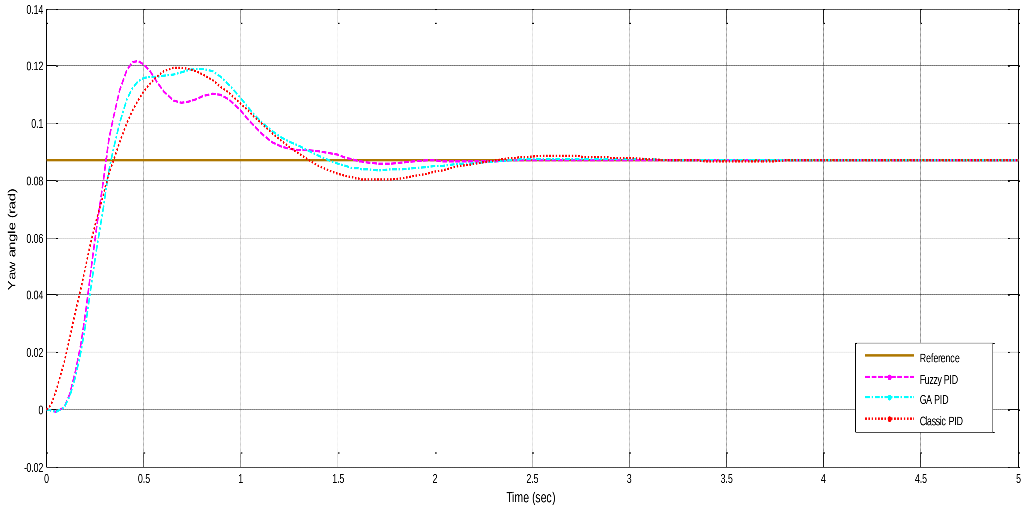


Figure 6.28: Yaw controller result_2.

Table 6.12: Value of response characteristics for yaw controller_2.

Response characteristics	**Fuzzy PID using GA**	**GA PID**	**Classic PID**
Rise Time, Tr in sec	0.1555	0.1724	0.2449
Settling Time Ts in sec	1.5021	2.0592	2.1548
Settling Min	0.0801	0.0787	0.0786
Settling Max	0.1217	0.1190	0.1194
Overshoot Mp in %	39.9051	36.8314	37.2218
Peak Time, Tp in sec	0.4734	0.8089	0.6942

fuzzy logic based intelligent controller design is introduced for controlling lateral yaw control of an aircraft. The proposed Fuzzy logic controller can effectively eliminate these dangerous oscillations and provides smooth operation in transient period.

7. Adaptive Neuro Fuzzy Inference System

In this chapter, The controller system architecture and design are included in the section in terms of longitudinal and lateral motions respectively.

7.1 Introduction

One of the system controller which have an artificial neural network (ANN) is defined as an information-processing system inspired by the structure of the human brain in biological nervous system. Being the source of natural intelligence, the human brain which tends to perceive complete shapes can process incomplete information obtained by perception at a very rapid rate. Inspired by the structure of this biological property of the nervous systems and the brain, researchers attempted to model the human brain resulting in the evolution of the neural network. In the thesis the brain has been modeled as a continuous time nonlinear dynamic of unmanned aerial vehicle system includes one more differential equations with a connection architecture. This gives the neural network the capability for learning and adaptation by adjusting the interconnection dynamics between artificial neurons of the layers [34].

The network of the system will be considered to be an artificial neural network if the same basis function is used throughout the network. The layers of a neural network architecture may be subdivided. Depending upon the number of inputs all the input neurons together constitute the input layer. Similarly all the output neurons together constitute the output layer [35].

Fuzzy sets can be considered as an extension of crisp set theory.At the same time the fuzzy set theory provides a systematic calculus to deal with linguistic information, and that improves the numerical computation by using linguistic labels stipulated by membership functions. The Fuzzy inference system (FIS) is based on the concepts of fuzzy set theory, fuzzy if-then rules and fuzzy reasoning. The framing of the fuzzy if-then rules forms the key component in FIS.The fuzzy is located in a detail in Section 6 [36]..

A artifical neural network -fuzzy technique called Adaptive network based fuzzy inference system (ANFIS) has been used as a primary tool in the thesis. Adaptive network based fuzzy inference system (ANFIS) is a neuro fuzzy modeling is an approach where the fusion is made between the neural network and the fuzzy inference system [37]. In the thesis's ANFIS the parameters can be estimated in such a way that both TS (Takagi-Sugeno)fuzzy models are represented by the ANFIS architecture. This ANFIS methodology consist of a hybrid system of fuzzy logic and neural network technique. The fuzzy logic takes into account the imprecision and uncertainty of the system that is being modeled while the neural network gives it a sense of adaptability. A hybrid intelligent system which provides an efficient combining the FLC approach and ANN capabilities towards better performance. The ANFIS approach was applied both longitudinal control system and lateral control system in the modelling .

Using this hybrid method, firstly an initial fuzzy model along with its input variables depending on system's conditions are derived with the help of the rules extracted from the input output data of the system that is being modeled. After that, the neural network is used to fine tune the rules selected generate FIS consist of grid partition, clustering method of the initial fuzzy model to produce the final ANFIS model of the system. In this thesis work ANFIS is used as the models (linear-nonlinear system) for the identification of real world systems [38]..

7.2 Types of Fuzzy System

Takagi–Sugeno (TS), method of fuzzy inference systems. The main advantage of the TS model is that is a simple model and can be used in many practical applications for modeling and control of complex systems [39]. But the premise of a Takagi-Sugeno fuzzy system is linguistically interpretable. For a dynamic process modelling the Takagi-Sugeno models possess an excellent interpretation.Consequents are real numbers, which can be either linear or constant. The final output (known as a singleton output membership function), is the weighted average of each rule's output.

The fuzzy rule of the first-order Sugeno-style has the form:

IF x is A AND y is B so $z = f(x,y) = ax + by + c$ **(7.1)**

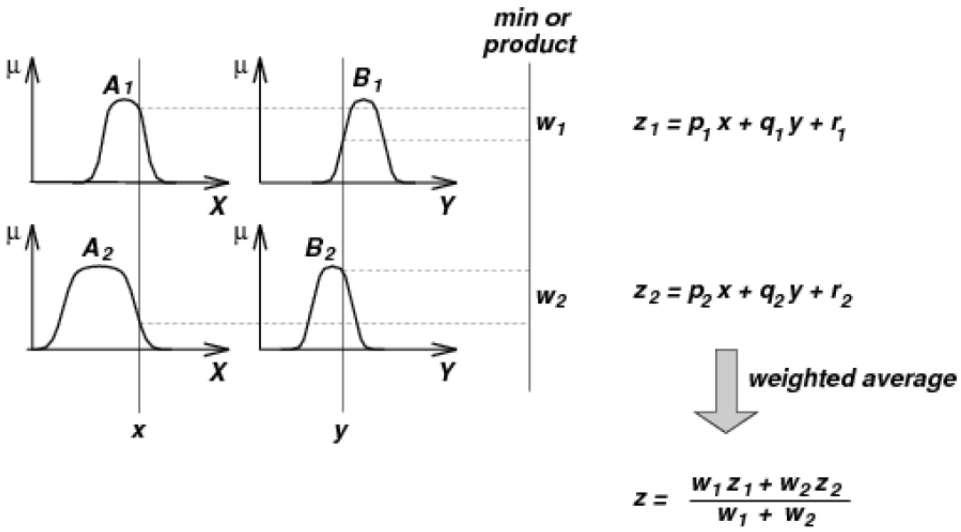


Figure 7.1 : First-order Sugeno Fuzzy model.

where A and B are fuzzy sets in the antecedent, while z=f(x,y) is a crisp function in the consequent [40].

7.3 ANFIS Method

ANFIS is the implementation of FIS to adaptive networks for developing fuzzy rules which determined as depending on numbers of input with convenient membership functions which obtained as many types like manually or from systems to have required inputs and outputs. FIS is main computing tool to which fuzzy if-then rules and fuzzy reasoning create bases that performs mapping from a given input knowledge (parameters) to desired output using fuzzy theory [41].

7.3.1 Overview of ANFIS

This ANFIS system is designed to allow IF-THEN rules and membership functions (fuzzy logic) to be expressed based on the historical data which obtained from nonlinear system and controlled system in this thesis and also includes the adaptive nature for automatic tuning of the membership functions.The advantage of the FIS is that it can deal with linguistic expressions and the advantage of a neural network is that it can be trained. The ANFIS is a data driven procedure representing a neural network approach for the solution of function approximation problems [42].

Adaptive neuro-fuzzy inference system is composed of five functional blocks as shown in **Figure 7.2**.

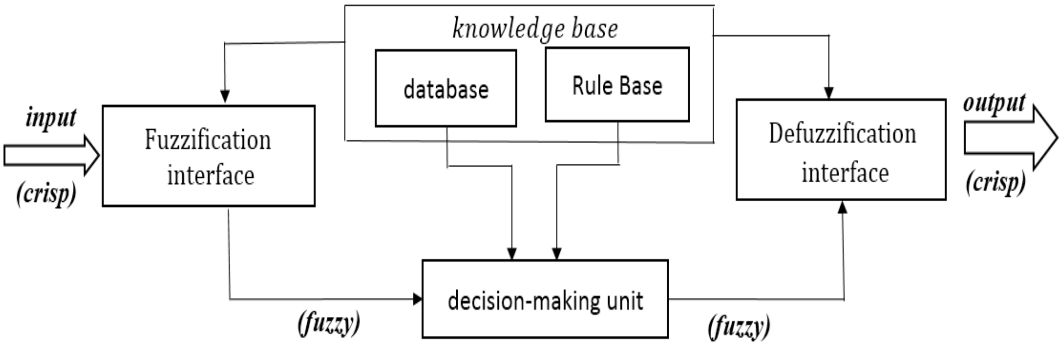


Figure 7.2: Block representation of ANFIS.

- a rule base containing a number of fuzzy IF - THEN rules
- a database which defines the membership functions include items one more than of the fuzzy sets used in the fuzzy rules
- a decision-making unit which performs the inference operations on the rules
- a fuzzification interface which transforms the crisp inputs into degrees of match with linguistic values
- a defuzzification interface which transform the fuzzy results of the inference into a crisp output

7.3.2 Structure of ANFIS

In fuzzy reasoning, fuzzy rules obtained from experts in special field, but it suffers from lack of an effective learning mechanism. Neuro-fuzzy control system is introduced in above, which is used neural network to implement fuzzy inference control system. For simplicity, it is assumed that the fuzzy inference system have three inputs and one output also they are considered different inputs and output for all of controller.

The rule base contains the fuzzy if-then rules of TS's type :

If x is A and y is B then z is f(x,y) **(7.2)**

where A and B are the fuzzy sets in the antecedents and z = f(x, y) is a crisp function in the consequent. Usually f(x, y) is a polynomial for the input variables x and y. But it can also be any other function that can approximately describe the output of the system within the fuzzy region as specified by the antecedent.If f(x,y) is taken to be a first order polynomial a first order Sugeno fuzzy model is formed [43]. For a first order two rule Sugeno fuzzy inference system, the two rules may be stated as:

Rule 1: If x is A_1 and y is B_1 then $f_1 = p_1 x + q_1 y + r_1$ **(7.3)**

Rule 2: If x is A_2 and y is B_2 then $f_2 = p_2 x + q_2 y + r_2$ **(7.4)**

…

The final output is the weighted average of each rule's output. The corresponding equivalent ANFIS structure is shown **figure 7.3**.

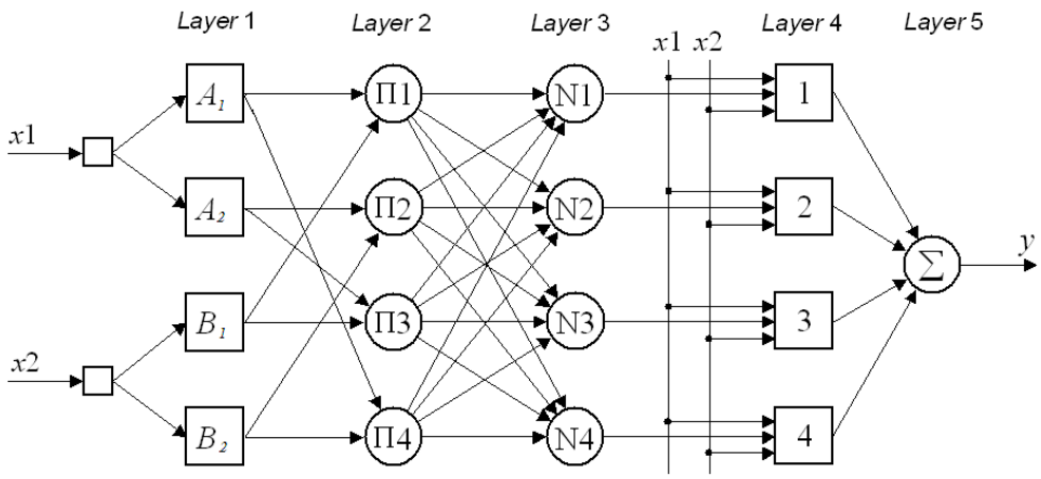


Figure 7.3: Type-3 ANFIS Structure [44].

The individual layers of this ANFIS structure are briefly described below :

Layer 1: Every node i in this layer is adaptive with a node function

$O_i^1 = \mu A_1(\mathrm{x})$ **(7.5)**

where, x is the input to node i, A_i is the linguistic variable associated with this node function and μA_1 is the membership function of A_i. μA_i (x) is chosen as

$\mu A_i(\mathrm{x}) = \frac{1}{1+[(\frac{x-c_i}{a_i})^2]^{b_i}}$ **(7.6)**

$\mu A_i(\mathrm{x}) = \exp\{-(\frac{x-c_i}{a_i})^2\}$ **(7.7)**

Gaussian MF as shown in **Figure 7.4**

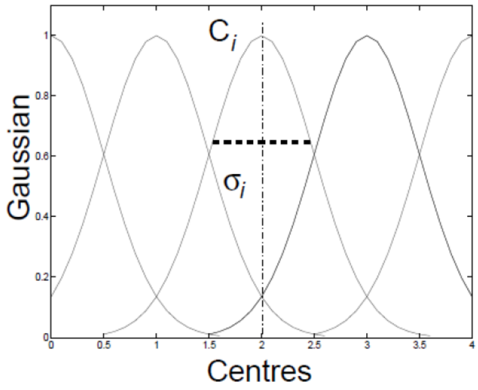


Figure 7.4: Gaussian membership function with centre (c_i) and width (σ_i).

where x is the input and $\{a_i, b_i, c_i\}$ is the premise parameter set.

Layer 2: Each node in this layer is a fixed node which calculates the firing strength w_i of a rule. The output of each node is the product of all the incoming signals to it and is given by,

$O_i^2 = w_i = \mu A_1(\mathrm{x}) \times \mu B_1(\mathrm{y}),\ i = 1,2$ **(7.8)**

Layer 3: Every node in this layer is a fixed node. Each i^{th} node calculates the ratio of the i^{th} rule's firing strength to the sum of firing strengths of all the rules. The output from the i^{th} node is the normalized firing strength given by,

$O_i^3 = \overline{w_i} = \frac{w_i}{w_1 + w_2},\ i = 1,2$ **(7.9)**

Layer 4: Every node in this layer is an adaptive node with a node function given by

$O_i^4 = \overline{w_i} f_i = \overline{w_i}\,(p_i x + q_i y + r_i),\ i=1,2$ **(7.10)**

where $\overline{w_i}$ is the output of Layer 3 and $\{p_i, q_i, r_i\}$ is the consequent parameter set.

where $\overline{w_i}$ is the output of Layer 3 and $\{p_i, q_i, r_i\}$ is the consequent parameter set.

Layer 5: This layer comprises of only one fixed node that calculates the overall output as the summation of all incoming signals, [45]

O_i^5 = overall output = $\sum_i \overline{w_i} f_i = \frac{\sum_i \overline{w_i} f_i}{\sum_i w_i}$ [10] **(7.11)**

7.3.3 Learning algorithm

In the ANFIS structure, it is observed that given values described system of the premise parameters, the final output can be expressed as a linear combination of the consequent parameters.The shape of membership funtions determined ‘‘ if ’’ are called premise parameters while the membership functions determined ''then'' are called consequent parameters.

$$f = \frac{w_1}{w_1+w_2} f_1 + \frac{w_2}{w_1+w_2} f_2 \quad \textbf{(7.12)}$$

$$= \overline{w_1} f_1 + \overline{w_2} f_2 \quad \textbf{(7.13)}$$

$$=(\overline{w_1}\, x)\, p_1 + (\overline{w_1}\, y)\, q_1 + (\overline{w_1})\, r_1 + (\overline{w_2}\, x)\, p_2 + (\overline{w_2}\, y)\, q_2 + (\overline{w_2})\, r_2 \quad \textbf{(7.14)}$$

Where f is linear in the consequent parameters $(p_1, q_1, r_1, p_2, q_2, r_2)$

In the forward pass of the learning algorithm node outputs go forward, and the consequent parameters {pi, qi, ri} are estimated by least squares method, while the premise parameters remain fixed for this. In the backward pass of the learning algorithm, the error signals, which are the derivatives of the squared error with respect to each node output, propagate backward from the output layer to the input layer the parameters of antecedent are updated according to the chain rule. the premise parameters {ai, bi, ci} are updated by the gradient descent algorithm [44].

7.3.4 Derivation of the initial Fuzzy model

As described earlier section, in ANFIS based system modeling for a set of rules with fixed premise parameters (p,q,r…as shown 7.3), identification of an optimal fuzzy model with respect to the training data reduces to a linear least squares estimation problems since it easy to find premise parameters (p,q,r..)

7.3.4.1 Extracting the initial Fuzzy model

In order to start the modeling process, an initial fuzzy model has to be derived. There are many datas set up modeling process but it can not be obtain right result into the anfis model so this model is required to find the number of inputs, number of linguistic variables and hence the number of rules in the final fuzzy model. The initial model is also required to select the input variables for the final model and also the model selection criteria, before the final optimal model can be derived.

7.3.4.2 Selection of input variables and final Fuzzy model

Any good criteria which is used for model selection based on some inputs trained result of obtained from nonlinear system model for checking datas cannot guarantee that the model will be producing the optimal result unless the results from all possible combinations of the input variables in the model are compared.

7.4 ANFIS Model for UAV

Performance measures are used to indicate how well a method performs its tasks.The performance of the model can be measured in two controller: Longitudinal motion and lateral motion controller. In longitudinal controller, linear or nonlinear of the aircraft and actuator behavior are performed that uses by a desired pitch command signal as the set point for the aircraft's pitch attitude to make fine adjustments and uses aircraft pitch angle and pitch rate to determine commands.In lateral motion controller, there are two control inputs, aileron deflection δ_a and rudder deflection δ_r in order to determine the lateral control angles required for sideslip and trim under steady roll turns.

Proposed model, the ANFIS approach implements the hybrid learning algorithm (HLA), a combination of the gradient descent (GD) algorithm for backward pass and a recursive least squares for forward pass algorithm to update the antecedent and consequent parameters. The combination of FIS to represent knowledge in an interpretable manner and the learning ability of neural network that can adjust the membership functions parameters and linguistic rules directly from data enhances the system performance.

The ANFIS architecture contains a five-layer feedforward neural network and implements a Sugeno fuzzy inference system as shown in **figure 7.1** for a systematic approach to generating fuzzy rules from a given input output dataset.

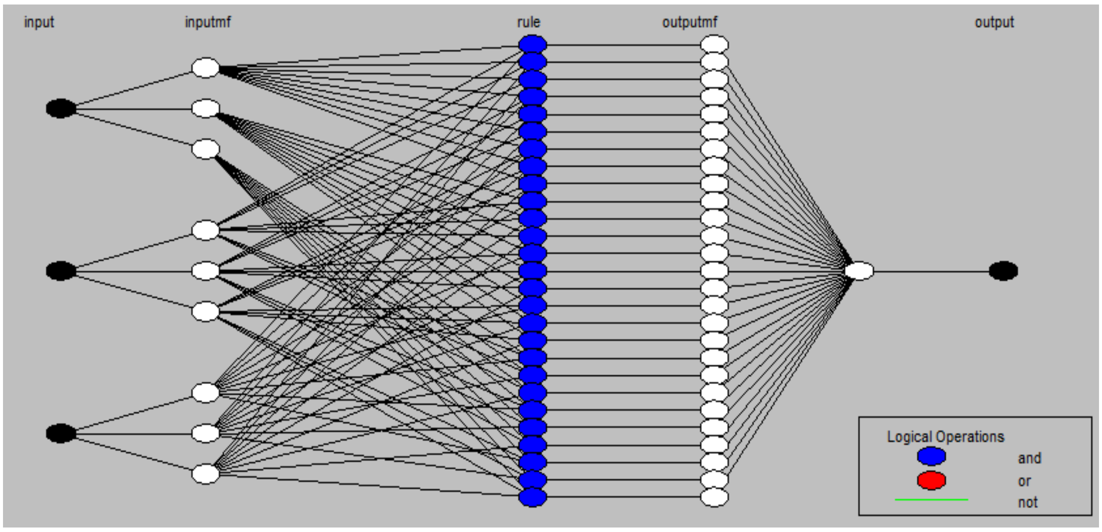


Figure 7.5: ANFIS model structure.

Initial parameters of the FIS have to be made before the training process. The number of membership functions assigned to each input was chosen by trial and error, and the performance measure (RMSE) was used to ensure the model did not cause over fitting. Gaussian MF was used as the input membership function type, while constants were used for the rule outputs (a thirth-order Sugeno model). In thirth-order Sugeno fuzzy inference.

ANFIS models were configured with different numbers of membership function. The number of training process are 30 epochs. The parameters for training options have been selected empirically as follows; 30 epochs were used to stop the training process, and the step size was 0. The step size, (the length of each gradient transition in the parameter space) can be changed by the attitude checking data was compared with trained data which obtained from nonlinear system.

7.4.1 ANFIS model for longitudinal controller

The controller of the UAV was developed using fuzzy logic.Two control methods were used. The first approach collected information about the inputs of the autopilot using a learning algorithm. The unmanned aeriel vehicle used a PID controller and two fuzzy logic combinations to allow the its to maintain its altitude. The PID selected the most appropriate control surface displacements. The inputs to the proposed adaptive neuro fuzzy controller are the error (e) and the rate of change of error (de/dt) while the outputs are the proportional gain K_p , the integral gain K_i and the derivative gain K_D.The two separate fuzzy logics adjusted the proportional and derivative gains by considering the pitch and change angle errors.

To determine UAV model base on identification model system is needed dynamic pitch angle response of UAV . There are three variabels: time (t), actual speed (y(t)), setting speed (u(t)). Four data is needed to train ANFIS, such as three input data and an output data. The input data are actual speed and delay actual speed. The output data is training target that represented by step and elevator degree function. The data for training are acquired from autopilot controller.

ANFIS Training, The two anfis parameter optimization method options available for FIS training. Hybrid method is selected because of best solutions.The experiment was done by the gauss function with 3 membership function for each input and output variables. The ANFIS used here contains 27 (3x3x3=27) rules, 45 total number of fitting parameters, including 18 (3x3+3x3) premise (non linear) parameters and 27

(3x9=27) consequent (nonlinear) parameters. Tolerance of Error which is used for stopping of train criterion, which is involved the error size.The training and root mean square (RMS) errors obtained from the ANFIS are 0,000040796 for 30 epochs.

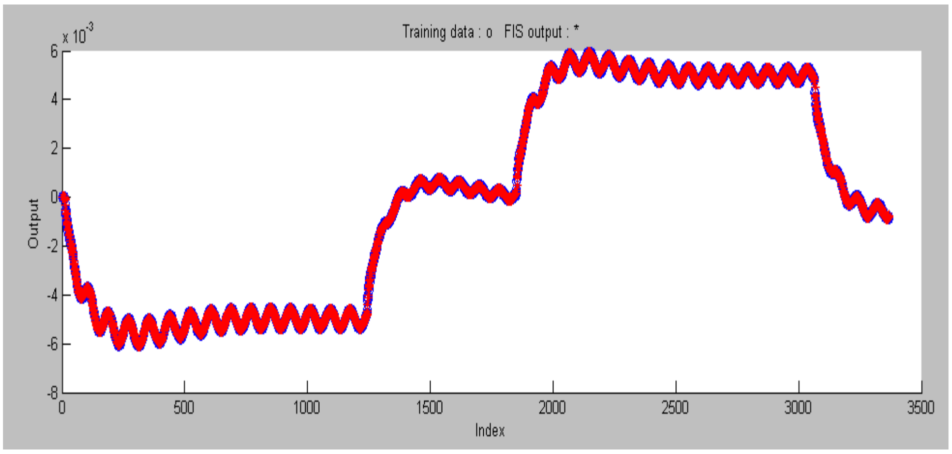


Figure 7.6: ANFIS model training for longitudinal.

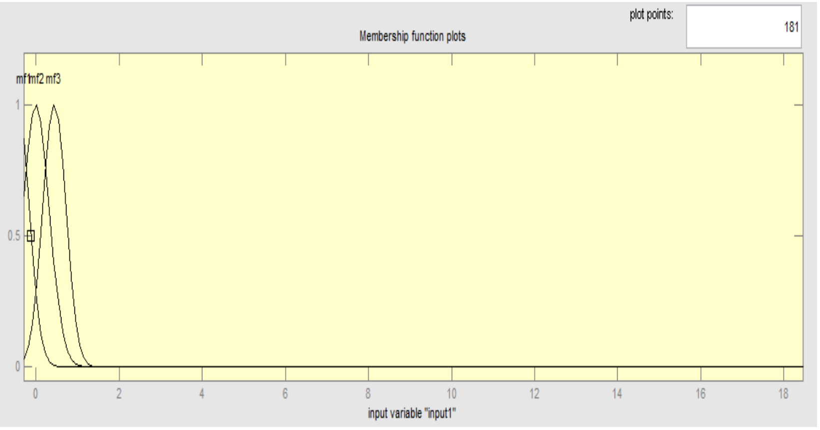


Figure 7.7: The membership function for input_1 after trained.

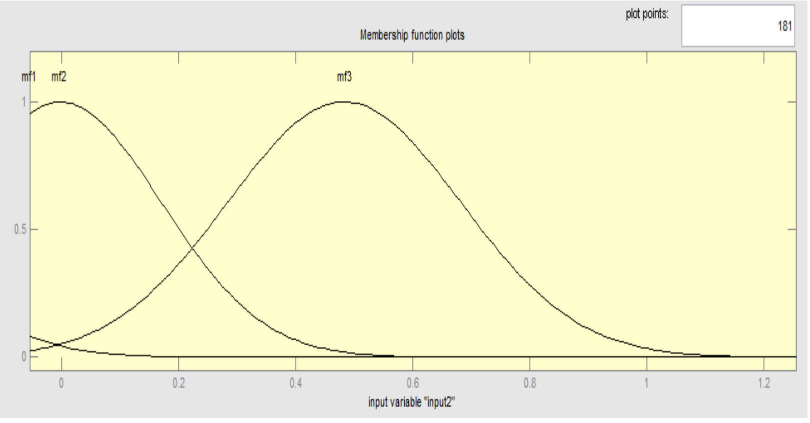


Figure 7.8: The membership function for input_2 after trained.

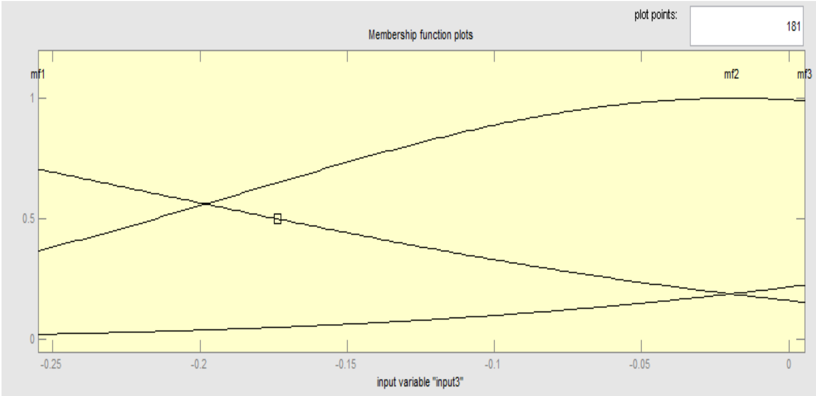


Figure 7.9 : The membership function for input_3 after trained.

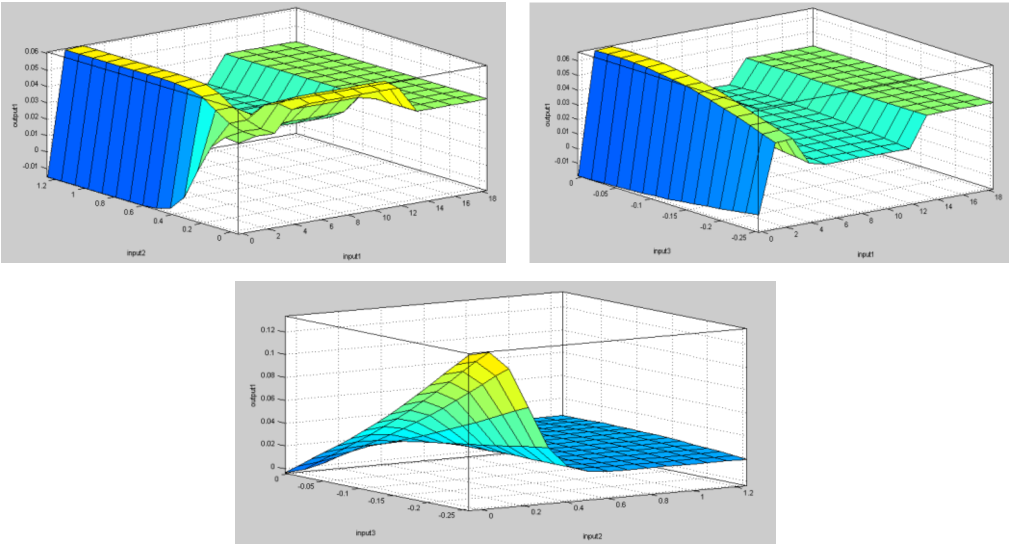


Figure 7.10 : Inputs-output surface elevator.

Figure 7.11 : Longitudinal ANFIS Simulink Model.

The anfis model responses after implemention of longitudinal motion system are simulated at 3 different nominal result.

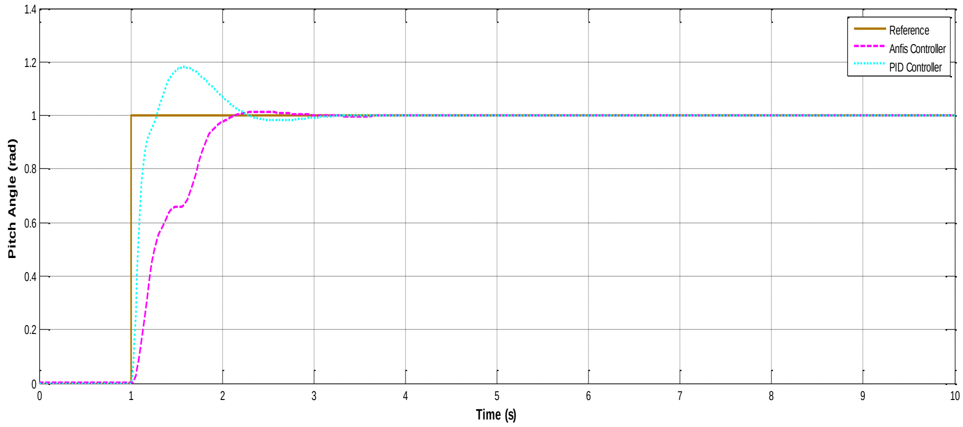


Figure 7.12: Pitch step command of ANFIS.

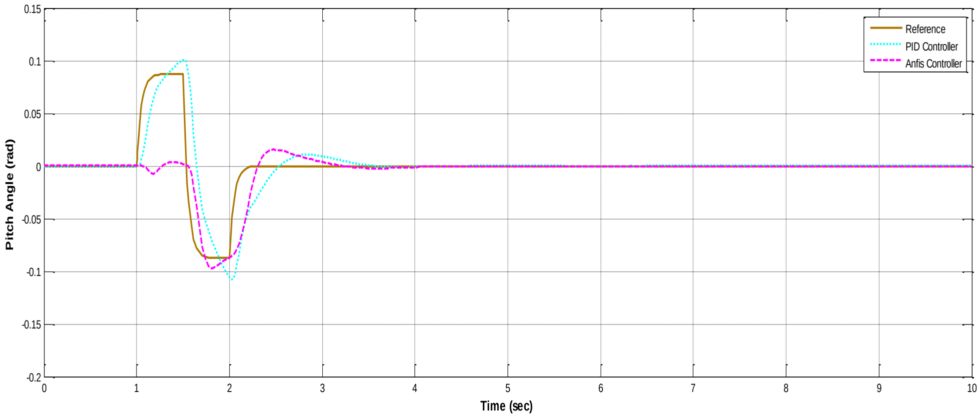


Figure 7.13: Pitch degree command of ANFIS.

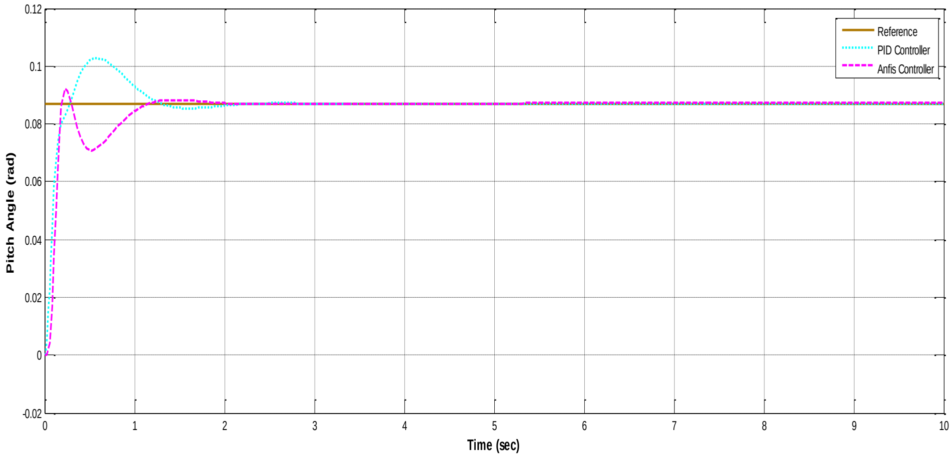


Figure 7.14: Pitch command of ANFIS.

7.4.2 ANFIS Model for Lateral Controller

A detailed coverage of ANFIS can be found in **Figure 7.2**. The TS fuzzy controller with ANFIS architecture can formalize a systematic approach to generate the fuzzy rule and MFs. The training data set of lateral error, roll rate, yaw rate, roll angle is obtained from system which have nonlinear model for UAV. The rule base includes the fuzzy-if-then rules of TS's type. Lateral controller has an important role in maneuvers, so ANFIS controller is applied to lateral controller sorely.In this perspective, the aim of this section in the thesis is to apply ANFIS to UAVs lateral controller, controls sideslip and roll angle to enhancing the lateral controller's performance. It will guaranty the best possible controlling in hard situations such as winds, gusts, instead of returning PID coefficients.

For this purpose, fuzzy modules are designed, a module is used for adjusting the bank angle value to control the latitude coordinates and to obtain the desired angular values, lateral controller produces a proper value for rudder deflection (δ_r) and ailerons deflection (δ_a).

The experiment was done by the gauss function with 3 membership function for each input and output variables. The ANFIS used here contains 27 (3x3x3=27) rules, 45 total number of fitting parameters, including 18 (3x3+3x3) premise (nonlinear) parameters and 27 (3x9=27) consequent (nonlinear) parameters. Tolerance of Error which is used for stopping of train criterion, which is involved the error size.The training and root mean square (RMS) errors obtained from the ANFIS are 0,00072742 for 30 epochs.

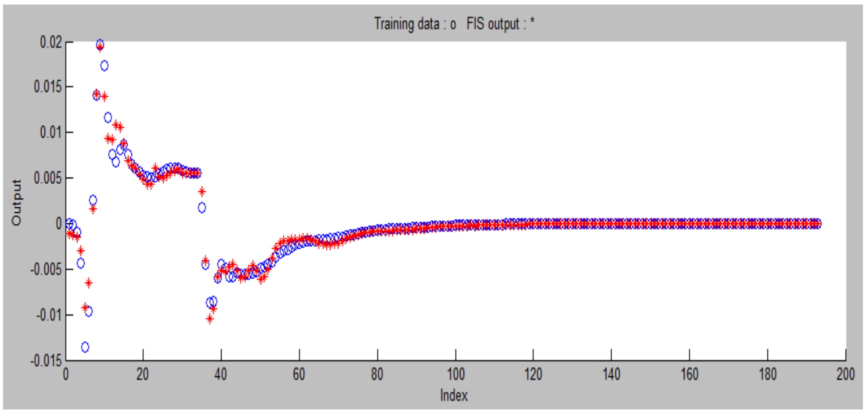


Figure 7.15: ANFIS model training for lateral.

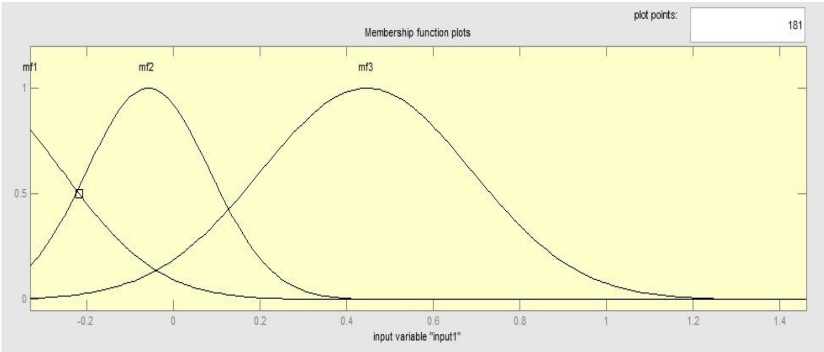


Figure 7.16: The membership function for input_1 after trained.

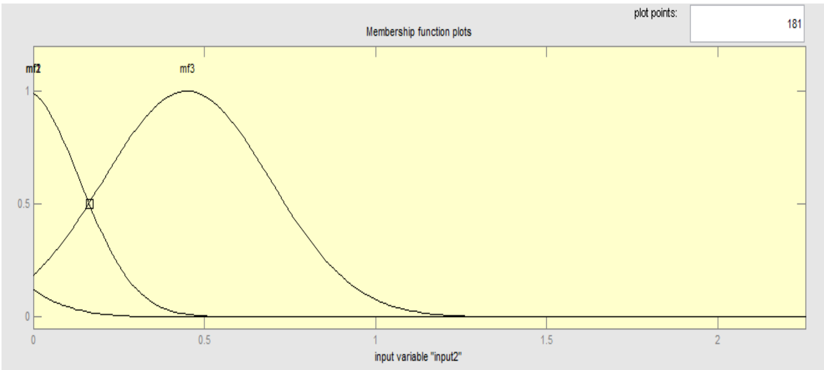


Figure 7.17: The membership function for input_2 after trained.

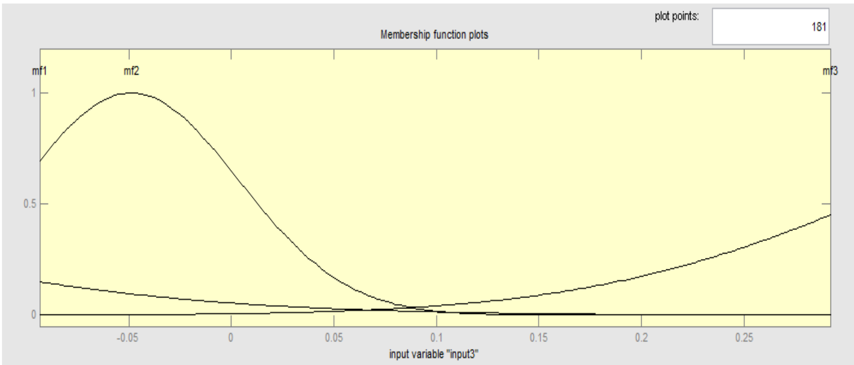


Figure 7.18: The membership function for input_3 after trained.

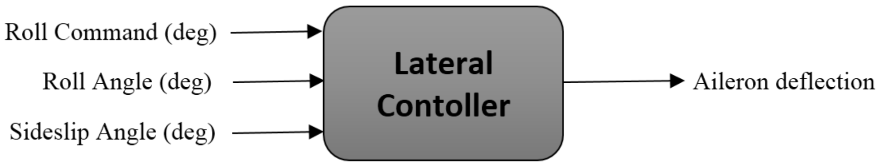


Figure 7.19: Lateral ANFIS Simulink Model_1.

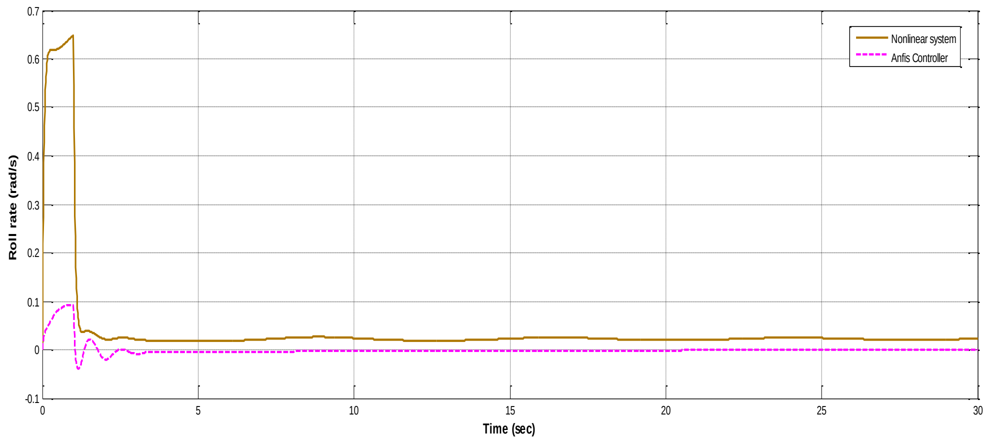


Figure 7.20: Roll rate of Anfis for Lateral ANFIS Simulink Model_1.

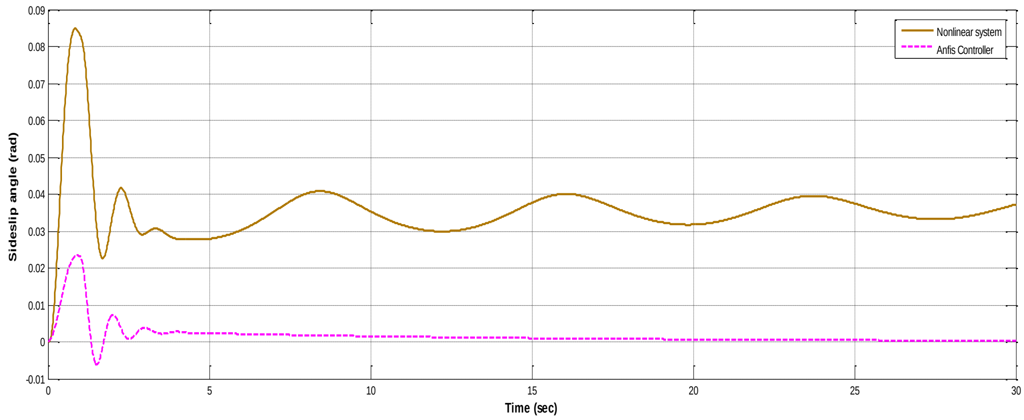


Figure 7.21: Sideslip angle of Anfis for Lateral ANFIS Simulink Model_1.

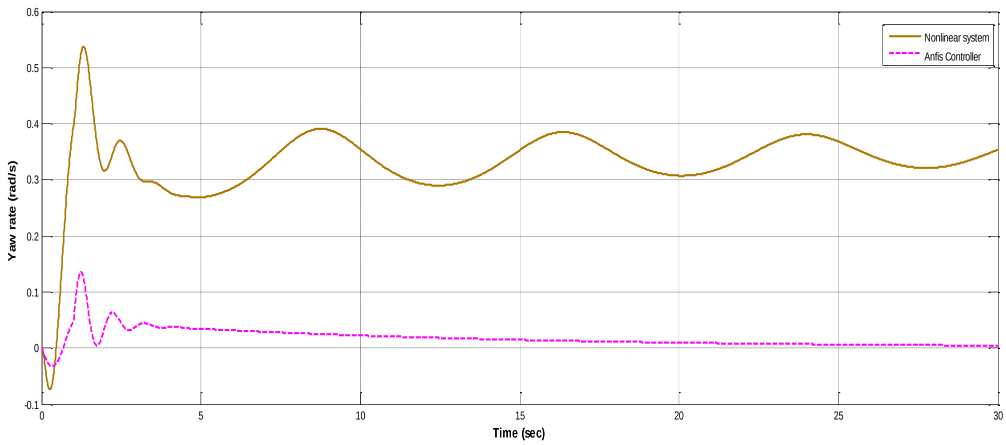


Figure 7.22: Yaw rate of Anfis for Lateral ANFIS Simulink Model_1.

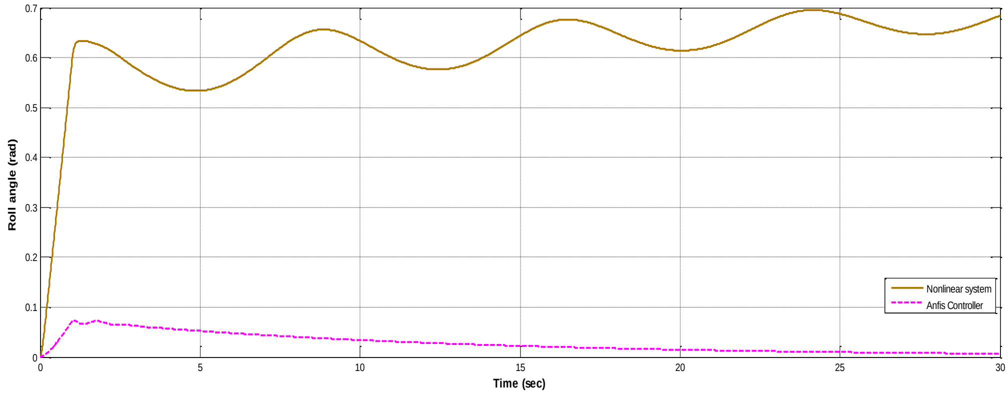


Figure 7.23: Roll angle of Anfis for Lateral ANFIS Simulink Model_1.

what is described above; the output of the one ANFIS controllers enable the banking angle, sideslip angle be controlled means that roll controller. That is to say, the attributes of the UAV is kept under control so the system guides the platform to the desired position in four dimensional space.

The inputs to this subsystem are shown figure, which is the yaw command, roll angle the last input is the yaw rate.The function of the Lateral Controller subsystem is to reach the desired sideslip angle and the desired yaw rate and therefore it controls the rudder position as outputs.

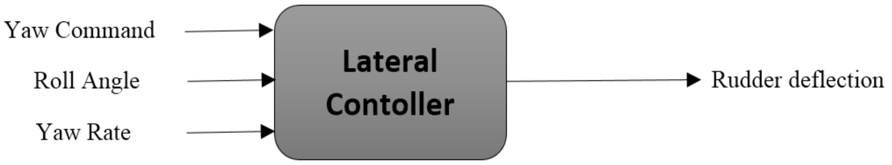


Figure 7.24: Lateral ANFIS Simulink Model_2.

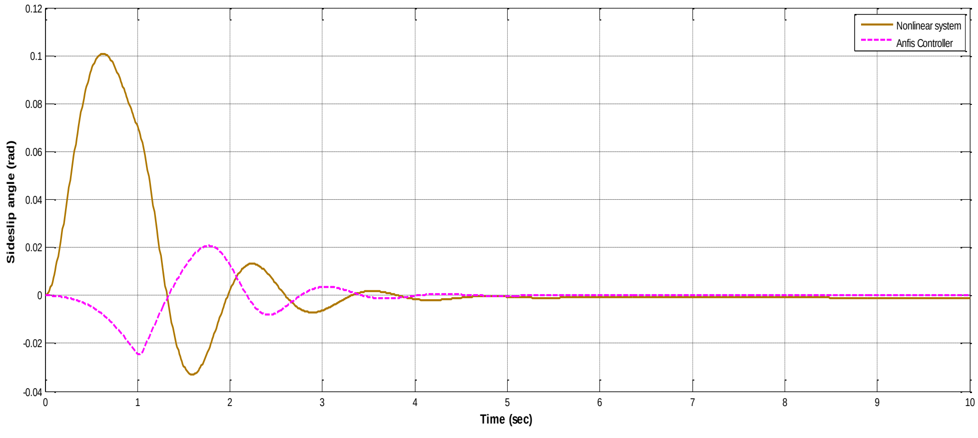


Figure 7.25: Sideslip angle of Anfis for Lateral ANFIS Simulink Model_2.

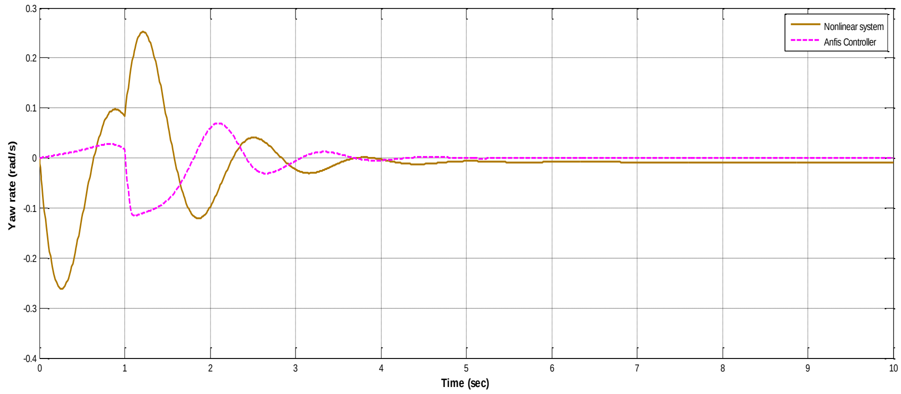


Figure 7.26: Yaw rate of Anfis for Lateral ANFIS Simulink Model_1.

what is described above; the output of the one ANFIS controllers enable the banking angle, sideslip angle be controlled means that roll controller. That is to say, the attributes of the UAV is kept under control so the system guides the platform to the desired position in four dimensional space.

The inputs to this subsystem are shown **figure 7.27**, which is the roll angle, roll error which is the different between the desired roll angle and the current roll angle, the last input is the sideslip angle. The function of the Lateral Controller subsystem is to reach the desired sideslip angle and the desired roll angle and therefore it controls both rudder and aileron position as outputs.

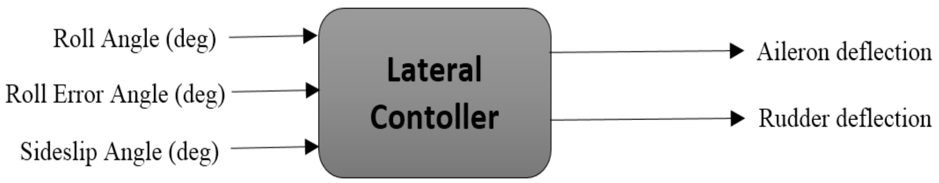


Figure 7.27 : Lateral ANFIS Simulink Model_3.

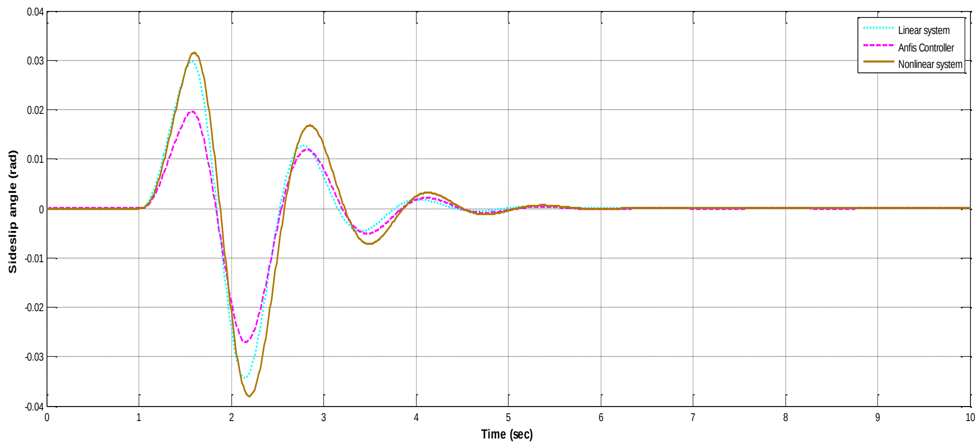


Figure 7.28: Sideslip angle of Anfis for Lateral ANFIS Simulink Model_3.

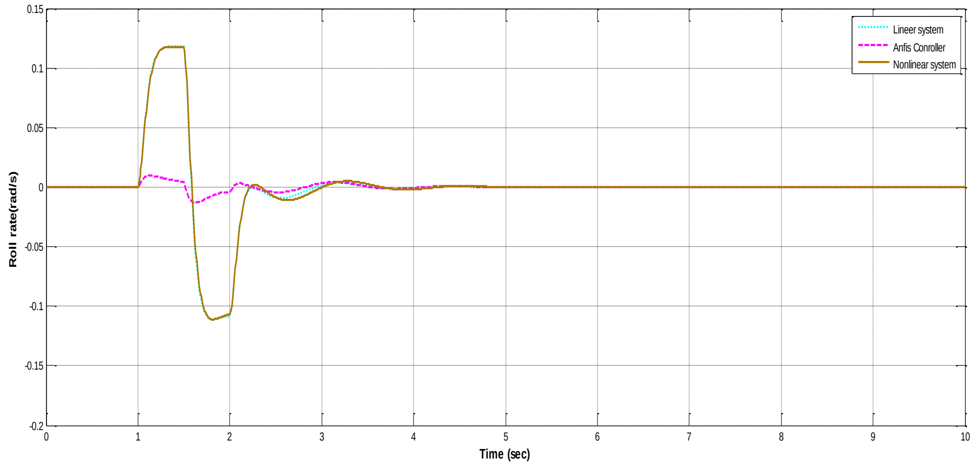


Figure 7.29: Roll rate of ANFIS for Lateral ANFIS Simulink Model_3.

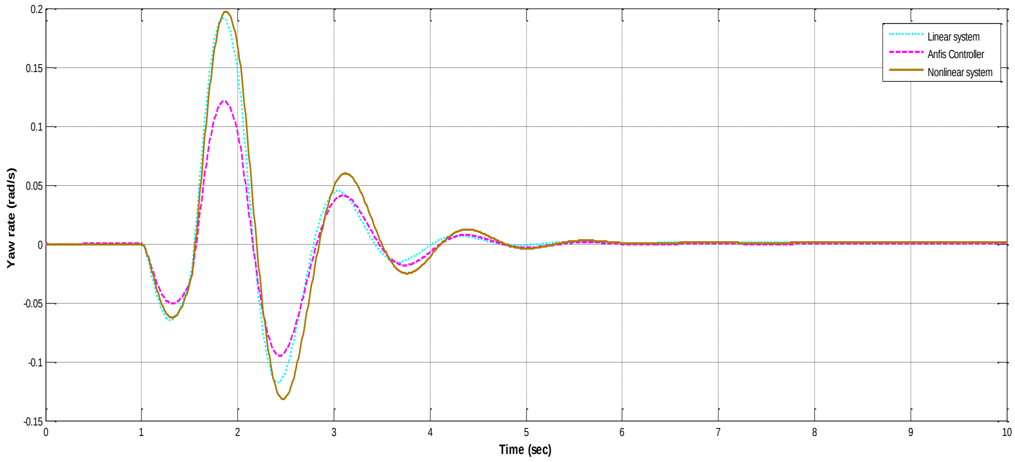


Figure 7.30: Yaw rate of ANFIS for Lateral ANFIS Simulink Model_3.

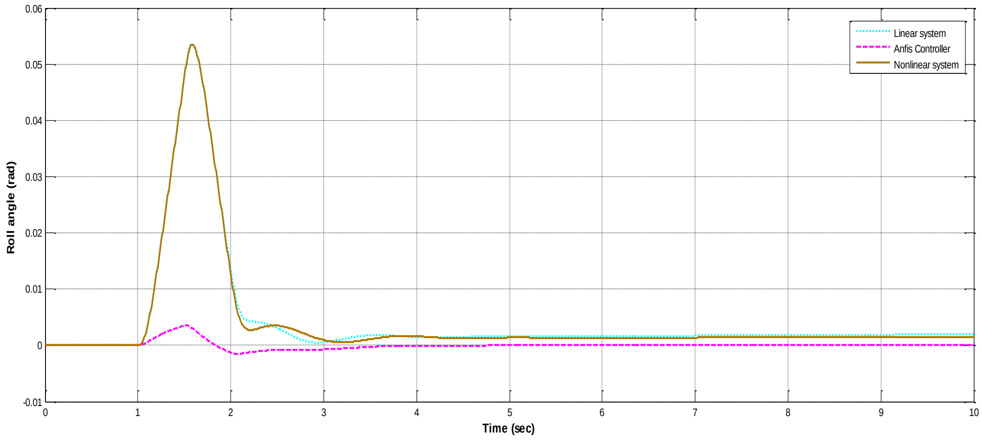


Figure 7.31: Roll angle of ANFIS for Lateral ANFIS Simulink Model_3

what is described above; the output of the one ANFIS controllers enable the banking angle, sideslip angle be controlled means that roll controller. That is to say, the attributes of the UAV is kept under control so the system guides the platform to the desired position in four dimensional space.

7.5 ANFIS Conclusion

A totally different approach to the flight control is to use the ANFIS technology.ANFIS have powerful control capacity, learning properties and parallel implementation paradigm.Since ANFIS have the ability to approximate uncertain nonlinear model to high degree of accuracy.ANFIS based control laws have been designed a longitudinal control system, a lateral – directional control system.

In this chapter, a more comprehensive approach is proposed.ANFIS dynamic modeling and intelligent controller schemes for the unmanned aerial vehicle are developed without simplifications or linerizations in the UAV.All of the control surfaces of the aircraft model are controlled by the developed controllers.

An ANFIS based adaptive PID control for longitudinal scheme is developed.In this scheme, the learning and adaptation abilities of ANFIS are combined with performance of PID controller.It can adapt to the changes in flight conditions and system parameter. Also , an ANFIS model for lateral scheme is developed.It is used to control three angular rates of the UAV.The performances of the developed ANFIS controller are applied by performing 6 DOF nonlinear model.

Also the neuro-fuzzy modeling of systems using ANFIS has been demonstrated using the input-output data pairs collected from nonlinear and autopilot controller which have controlled by system. The modeling results obtained by using the various controllers are compared with the existing results for anfis.It can be observed that the mean square error (MSE) of the model obtained by using the trained method clustering based ANFIS is the two input Takagi-Sugeno model. These results show that the 2-input ANFIS model has good prediction capability.

In the ANFIS model, hybrid learning method is used for training the neural network for optimization of the fuzzy rules. A sufficient number of data is used to guarantee good training.There is no formula to find the data of model required to train a neural network. Once trained , the anfis is applied to through the new fuzzy set.

8. CONCLUSION

Including the nonlinear modeling of Piper J3-Cub ¼ scale unmanned aerial vehicle was accomplished throughout the whole thesis as well as trim and linearization processes of the developed nonlinear model in the thesis, the fuzzy logic based on PID contoller using genetic algorithm, and adaptive neuro fuzzy inference system approaches respectively.

The nonlinear model is very detailed in regarding section with both simulink model in Matlab/Simulink® and formulation, and is well-integrated with detailed atmospheric,gravity models and wind-turbulence models.In addition, the methods and respective implementations carried out to trim and linearize the nonlinear model are well defined and the results are highly satisfying which can also be observed from the linear and nonlinear models comparison graphs **from 4.12 to 4.35**.

The major accomplishments throughout the whole thesis are to implement unmanned aerial vehicle Piper J3-Cub ¼ scale modeling and mainly two different controller designs around this model with transient analyses between those, also expressed such as the linear model open loop dynamic stability analyses, investigation of modal matrix, eigenvalues, eigenvector & pole-zero maps, etc.

The aircraft's dynamic model is normally constant formulation which was developed as a function of its physical parameters, which allows it to be readily applied to various conventional model aircraft.

As unmanned aerial vehicle have become more complex, tradional methods have not yielded acceptable performance.To overcome these problems, some of techniques such as fuzzy logic based on pid using genetic algorithm and adaptive neuro inference system controller were studied as alternatives to gain scheduling. These techniques are difficult to use because they rely heavily on accurate knowledge of UAV dynamics. However, some aerodynamic effects are very diffucult to model resulting in uncertainties in the aircraft dynamics. The total 12 state equations regarding section mentions variables used in the representation of 6 degrees of freedom rigid body equations of the aircraft motions, and four inputs used. The modal matrices are obtained for the purpose of validating the decoupling condition between the two major axes; the longitudinal and lateral-directional respectively.With these modal matrices, the dominant states on the main air vehicle flight modes are also demonstrated. The linearization methods are verified, by comparing the linear model simulation responses with the nonlinear model simulation responses to the same control inputs and investigating the matching degree.Since the accuracy of the nonlinear model is important, the analyses for the open loop dynamic stability are carried out for examining the flight modes in terms of flying quality levels.

In the thesis, we proposed various controller for UAV. In the scope of classical controller design, the controllers designed using this approach that are responsible for the control functions are the roll attitude, turn coordination, heading, pitch attitude, altitude controllers.

The second controller using fuzzy PID controller using genetic algorithm.Fuzzy Inference System (FIS) to tune the parameters of K_p, K_i, and K_d, using genetic algorithm code according to error (e) and derivative of error (de) for control functions are the roll attitude, turn coordination, heading, pitch attitude.

The third controller is using adaptive neuro fuzzy inference system. The decoupled longitudinal and lateral-directional axes are based on in the design procedure. In longitudinal axis, the controller which is designed was a successful attempt to design a fuzzy control based PID system for longitudinal motion as well as lateral motion is that the proposed control system can compensate the uncertainties of the system and adopt to the changes in flight given inputs for dynamic behaviour. And ANFIS model is developed for all angular rates of identification accuracy.

As possible future works, the analyses for robustness can be implemented additionally, in order to compare the obtained controllers regarding sections in this scope, which is also very important.Also by developing fuzzy control system which can perform a given specific flight path with accuracy navigational systems into the flight management system.

REFERENCES

[1] **Angelov, Plamen (2011).** *Sense and Avoid in UAS, Research and Applications.*

[2] **Dalamagkidis Konstantinos, Valavanis Kimon P. , Les A. Piegl (2012).** *On Integrating Unmanned Aircraft Systems into the National Airspace System* (2nd edition).

[3] **Valavanis Kimon P. (2007).** *Advances in Unmanned Aerial Vehicles: State of the Art and the Road to Autonomy* (1st edition)

[4] **Arnal Eric, Escusa Carlos. (2009).** *Study, design and construction of a Bi-Rotor and Tri-Rotor MAV platforms.*

[5] **Ryan Dan. (2012).** *Dorneywood, The Caldwell Series.*

[6] **Victorino, Terrence James. (2012).** *General Atomics Gnat*

[7] **Duke Eugene L., Antoniewicz Robert F., Krambeer Keith D. (1988).** *Derivation and Definition of a Linear Aircraft Model (Publicition 1207).*

[8] **Nelson, C.Nelson. (1997).***Flight Stability Automatic Control* (2nd edition).

[9] **Roskam, Jan. (2001).** *Airplane Flight Dynamics and Automatic Flight Controls (1st edition).*

[10] **Liden, Sam P. (1996).** Apparatus and method for computing wind-sensitive optimum altitude steps in a flight management system

[11] **Schmidt, Louis V. (1998).** *Introduction to Aircraft Flight Dynamics***.**

[12] **Ting Eric, Nguyen Nhan, Trinh Khanh.(2014).** *Static Aeroelastic and Longitudinal Trim Model of Flexible Wing Aicraft Using Finite-Element Vortex- Lattice Coupled Solution*, American Institute of Aeronautics and Astronautics.

[13] A Short History of Unmaned Aerial Vechicles http://www.draganfly.com/

[14] US Navy Aircraft Designations of the Second World War http://www.historyofwar.org/

[15] Grandfather to the Modern UAV, http://gizmodo.com/

[16] Cruise Missile, https://helloreverb.com/

[17] Area Moment of Inertia, http://emweb.unl.edu/

[18] Basic Avivation, http://smoknjoe.tripod.com/aviation.html

[19] Measurement Techniques: air motion sensing, https://www.eol.ucar.edu/

[20] Euler Angles, http://www.globalspec.com/

[21] Flight Dyamics (fixed-wing aircraft), http://en.wikipedia.org/

[22] **McDonnell, John D., (1968).** Pilot Rating Techniques for the Estimation and Evaluation of Handling Qualities.

[23] http://archive.org/stream/nasa_techdoc_19720004293/19720004293

[24] **Agustín Jiménez, Basil M.Al-Hadithi, Fernando Matía. (2011).** *Extended Kalman Filter for the Estimation and Fuzzy Optimal Control of Takagi- Sugeno Model.*

[25] **Farah Hassan, (1999).***The Fuzzy Logic Control of Aircraft.*

[26] **Melin P., Castillo O., Ramírez E.G., Pedrycz W., (2007).** *Analysis and Design of Intelligent Systems Using Soft Computing Techniques.*

[27] **Hassanzadeh I., Mobayen S. and Sedaghat G., (2008).** *Design and Implementation of a Controller for Magnetic Levitation System Using Genetic Algorithms* **''**Journal of Applied Sciences'' 8: 4644-4649 .

[28] **Melanie M., (1996).** *An Introduction to Genetic Algorithms* (2nd edition)

[29] **Shichao Zhang, Chengqi Zhang, Xindong Wu, (2004).** *Knowledge Discovery in Multiple Databases.*

[30] Putting fuzzy logic into focus, http://www.strano.net/

[31] **Mamdani, E.H., (1976).** *Advances in the linguistic synthesis of fuzzy controllers.*''*IEEE Transactions on Computers*, Vol. 26''.

[32] **Zili Zhang, (2004).** *Agent-Based Hybrid Intelligent Systems.*

[33] http://en.wikipedia.org/wiki/Aircraft_flight_mechanics

[34] **Dr.Hesham A. Hefny,** *Fuzzy Logic Lecture Note.*

[35] **Nagaraj B., Murugananth N., (2010).** Soft Compuring-Based Optimum Design of PID Controller for a Position Control of DC Motor. ''Mediamira Science Publisher, Vol. 51''.

[36] **Patrica M., Oscar Castillo, Eduardo G. Ramirez, Witold Pedrycz, (2007).** Analysis and Design of Intelligent Systems Using Soft Computing Techniques (1st edition).

[37] **Rahul Malhotra, Narinder Singh, Yaduvir Singh, (2011).** Genetic Algotrihms : Concepts, Design for Optimization of Process Controllers.''Computer and Information Science'' , Vol.4''.

[38] **Vinu V Das, R.Vijaykumar et al., (2010).** Information and Communication Technologies.''ICT 2010''.

[39] **Tushar Patel, Professor V.P.Patel, (2014).** Optimal FOPDT Modeling Of Liquid Level Control System Using Genetic Algorithm. "International Journal for Scientific Research & Development, Vol.2''.

[40] **S.Srinivasan, S.M.Girirajkumar,(2012).**Controler Tuning Of A Biological Proces Using Optimization Techniques, ‘‘International Journal of ChemTech Research, Vol.4’’ .

[41] **Noradin Ghadimi , Ardabil Branch, (2012).** Genetically tuning of lead-lag controller in order to control of fuel cell voltage.‘‘ Scientific Research and Essays Vol. 7’’ .

[42] **Oscar Cordón, (2011).** A historical review of evolutionary learning methods for Mamdani-type fuzzy rule-based systems: Designing interpretable genetic fuzzy systems, ‘‘International Journal of Approximate Reasoning 52’’.

[43] **Pandhiti, S., (2011).**Blackboard Architecture for an Unmanned Aerial Vehicle Controller using Fuzzy Inference Systems, Unpublished thesis, University of Georgia.

[44] **Duane T. McRuer, (2014). Dunstan Graham, Irving Ashkenas,** Aircraft Dynamics and Automatic Control.

[45] http://neuron.csie.ntust.edu.tw

Appendix A

A.1 MODEL DECOMPOSITION

Consider the general state space system below,

$$\dot{x} = Ax + Bu \tag{A.1}$$

$$Y = Cx + Du \tag{A.2}$$

where x is an arbitrary state vector of lenght f, u is the control vector of lenght g according to describing motion in aircraft and y is the output vector of lenght h .A is the $f \; x \; f$ system matrix. B is the $f \; x \; g$ input matrix, C is the $h \; x \; g$ output matrix and D is a $h \; x \; g$ matrix that is zero for the all systems defining accuretely.

The response of the system as linear derivated from nonlinear system can be analysed by setting the input vector of equation above to zero (u=0) (A.1) and definition the well know solution for a set of linear, first order differential equations as shown below,

$$x(t) = x_0 e^{\lambda t} \tag{A.3}$$

To solve for λ equation (A.3) is substituted into (A.1) to give,

$$\lambda x_0 = A x_0 \tag{A.4}$$

all equations is collected to left side with I matrix as defining lenght

$$(\lambda I - A)\, x_0 = 0$$

(A.5)

Equation (D.5) has non-trivial solutions when,

$$det(\lambda I - A) = 0$$

(A.6)

Characteristic equation of the system is defined equation above (A.6). It provides for λ called eigenvalues also finding for eigenvectors, which either real or occur in complex conjugate pairs. As before, a solution is assumed and substituted back into the equations of motion. The assumed solution is, Each eigenvalue is interested in a corresponding eigenvector, which can be visualised as a trajectory *space* to be translation invariant in the system space

Equation above (A.4) can be written in matrix form for all *f* eigenvalue-eigenvector solutions,

$$[e_1 \;\; e_2 \;\; \cdots \;\; e_f]\begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_f \end{bmatrix} = [\,A\,]\,[e_1 \;\; e_2 \;\; \cdots \;\; e_f] \tag{A.7}$$

where e_k represents the eigenvector regarding to the eigenvalue λ_k where k runs from 1 to *k* .Eqution (A.7) can be summarised as,

$$\boldsymbol{PR} = \boldsymbol{AP} \tag{A.8}$$

Let λ_1, λ_2, . . . , λ_n be the eigenvalues of a matrix A, let x_1, x_2, . . . , x_n be a set of corresponding eigenvectors, where ***P*** is a matrix of eigenvectors and ***R*** is the corresponding matrix of eigenvalues.The element of ***P*** and ***R*** are complex for each complex conjugate eigenvalue-eigenvector pair. With desired to write the eigenvalue and eigenvector matrices with real elements only.This can be done using a linear transformation to transform each complex conjugate eigenvalue-eigenvector pair to a real form as shown below,

$$\boldsymbol{P'} = \boldsymbol{PT} \tag{A.9}$$

$$\boldsymbol{R'} = [\boldsymbol{T^{-1}RT}]$$

(A.10)

where ***T*** is the transformation matrix, $\boldsymbol{P'}$ and $\boldsymbol{R'}$ are matrices with only real elements, and the relationship of equation,

$$\boldsymbol{P'R'} = \mathbf{A}\boldsymbol{R'} \tag{A.11}$$

still holds.If the 2X2 diagaonal blocks of *T* are set to,

$$\boldsymbol{P'} = [\,Re(e_1) \quad Im(e_1) \;\; \ldots\,] \tag{A.12}$$

$$\boldsymbol{R'} = \begin{bmatrix} Re(\lambda_1) & Im(\lambda_1) & 0 \\ -Im(\lambda_1) & Re(\lambda_1) & 0 \\ 0 & 0 & \ddots \end{bmatrix} \tag{A.13}$$

the diagonal elements of the transformation matrix are only unity for each real eigenvalue-eigenvector.Note that after the transformation, the matrices $\boldsymbol{P'}$ and $\boldsymbol{R'}$ have real elements only and that $\boldsymbol{R'}$ 2 x 2 blocks on the diagonal for eaxh complex conjugate eigenvalue-eigenvector pair.

Making $\boldsymbol{A}$ the subject of equation (A.11) and subsituting it into equation (A.1)

$$\dot{\boldsymbol{x}} = \boldsymbol{P'}\boldsymbol{R'}\boldsymbol{P'}^{-1}\mathbf{x} + \mathbf{Bu} \tag{A.14}$$

Now, let

$$\mathbf{z} = \boldsymbol{P'}^{-1}\mathbf{x} \qquad \mathbf{x} = \boldsymbol{P'}^{-1}\mathbf{z}$$
(A.15)

then, the modal form of equations is

$$\dot{\boldsymbol{z}} = \boldsymbol{R'}\mathbf{z} + \boldsymbol{P'}^{-1}\mathbf{Bu} \tag{A.16}$$

$$\mathbf{y} = \mathbf{C}\boldsymbol{P'}\mathbf{z} + \mathbf{Du}$$
(A.17)

Appendix B

B.1 MATLAB CODE

```
%NOTE : only small effect on speed u - period 0.2 sec, decays in 0.4
sec
%NOTE : - period 10 sec, decays in 70 sec

% u: velocity of aircraft along body axis
% v: velocity of aircraft perpendicular to body axis(down is
positive)
% theta: angle between body axis and horizontal(up is positive)
% q =theta_dot: angular velocity of aircraft (pitch rate)

%INPUTS
%disturbance inputs
%u_w: velocity of wind along body axis
%v_w: velocity of wind perpendicular to body axis
%control or actuator inputs
%_e: elevator angle
%_t: thrust
```

```
% Piper J-3 Cub longitudinal axis
% 150 m steady, level flight, 15 m/s
% x = u, w, q, theta

% modal analysis
[V,Gam]=eig(A); xshort = real(V(:,1); xphug = real(V(:,3);
xshort= xshort/normxshort; xphug = xphug/normxphug;

Nsamp = 300; %number of time samples
yshort=zeros((2,Nsamp)); t=linspace((0,20,Nsamp));
for i=1:Nsamp,
yshort(:,i)=C*exp(t(i)*A)*xshort;
end

figure %figure(1)
subplot(2,1,1),plot(t,yshort(1,:)');
axis([0 1 -1 1]),xlabel('time(s)'),ylabel('aircraft speed
u'),title('pure short-period mode motion'),legend('steady-state
change in speed');
subplot(2,1,2),plot(t,yshort(2,:)');
axis([0 1 -1 1]),ylabel('hdot'),xlabel('time(s)'),ylabel('climb rate
hdot'),title('pure short-period mode motion'),legend('steady-state
change in climb rate-rad');


Nsamp = 4000; %number of time samples
yshort=zeros(2,Nsamp); t=linspace(0,2000,Nsamp);
for i=1:Nsamp,
yphug(:,i)=C*exp(t(i)*A)*xphug;
end

figure %figure(2)
subplot(2,1,1),plot(t,yphug(1,:)');
axis([0 300 -2 2]),xlabel('time(s)'),ylabel('aircraft speed
u'),title('Phugoid mode motion'),legend('steady-state change in
speed');

subplot(2,1,2),plot(t,yphug(2,:)');
axis([0 300 -2 2]),xlabel('time(s)'),ylabel('climb rate
hdot'),title('Phugoid mode motion'),legend('steady-state change in
speed');
% now do responses to various impulses
% impulse response matrix from (uw; vw)

figure % figure(3)
Nsamp = 200; %number of time samples
h1=zeros((2,Nsamp)); h2=zeros(2,Nsamp); t=linspace(0,20,Nsamp);
for i=1:Nsamp,
h1(:,i)=C*exp(t(i)*A)**B(:,1); %%% impulse response from u_w
h2(:,i)=C*exp(t(i)*A)**B(:,2); %%% imp resp from v_w
end
subplot(2,2,1),plot(t,h1(1,:)'); axis([0 20 -.1
0.1]),xlabel('time(s)'),ylabel('from velocity of wind along body
axis'),title('wind')
subplot(2,2,2),plot(t,h2(1,:)'); axis([0 20 -.1
0.1]),xlabel('time(s)'),ylabel('from velocity of wind perpendicular
to body axis'),title('wind')
subplot(2,2,3),plot(t,h1(2,:)'); axis([0 20 -.5
0.5]),xlabel('time(s)'),ylabel('from velocity of wind along body
axis'),title('wind')
```

```
subplot(2,2,4),plot(t,h2(2,:)'); axis([0 20 -.5
0.5]),xlabel('time(s)'),ylabel('from velocity of wind perpendicular
to body axis'),title('wind')


% now do same plots over longer time scale
figure % figure(4);
t=linspace(0,600,Nsamp);
for i=1:Nsamp,
h1(:,i)=C*expm(t(i)*A)*B(:,1); % impulse response from u_w
h2(:,i)=C*expm(t(i)*A)*B(:,2); % imp resp from v_w
end
subplot(2,2,1),plot(t,h1(1,:)'); axis([0 600 -.1
0.1]),xlabel('time(s)'),ylabel('velocity of wind along body
axis'),title('wind')
subplot(2,2,2),plot(t,h2(1,:)'); axis([0 600 -.1
0.1]),xlabel('time(s)'),ylabel('velocity of wind perpendicular to
body axis'),title('wind')
subplot(2,2,3),plot(t,h1(2,:)'); axis([0 600 -.5
0.5]),xlabel('time(s)'),ylabel('velocity of wind along body
axis'),title('wind')
subplot(2,2,4),plot(t,h2(2,:)'); axis([0 600 -.5
0.5]),xlabel('time(s)'),ylabel('velocity of wind perpendicular to
body axis'),title('wind')


% now do same things, but for actuator inputs

% now do same plots over longer time scale
figure % figure(5);
Nsamp = 200; %number of time samples
h1=zeros(2,Nsamp); h2=zeros(2,Nsamp); t=linspace(0,20,Nsamp);
for i=1:Nsamp,
h1(:,i)=C*expm(t(i)*A)*B(:,3); % impulse response from delta_e
h2(:,i)=C*expm(t(i)*A)*B(:,4); % imp resp from delta_t
end
subplot(2,2,1),plot(t,h1(1,:)'); axis([0 20 -2
2]),xlabel('time(s)'),ylabel('impulse response from
delta_e'),title('response to actuators')
subplot(2,2,2),plot(t,h2(1,:)'); axis([0 20 -2
2]),xlabel('time(s)'),ylabel('impulse response from
delta_t'),title('response to actuators')
subplot(2,2,3),plot(t,h1(2,:)'); axis([0 20 -5
5]),xlabel('time(s)'),ylabel('impulse response from
delta_e'),title('response to actuators')
subplot(2,2,4),plot(t,h2(2,:)'); axis([0 20 0
3]),xlabel('time(s)'),ylabel('impulse response from
delta_r'),title('response to actuators')

figure %figure(6)
t=linspace(0,600,Nsamp);
for i=1:Nsamp,
h1(:,i)=C*expm(t(i)*A)*B(:,3); % impulse response from delta_e
h2(:,i)=C*expm(t(i)*A)*B(:,4); % imp resp from delta_t
end

subplot(2,2,1),plot(t,h1(1,:)); axis([0 600 -2 2]),ylabel('h11')
subplot(2,2,2),plot(t,h2(1,:)); axis([0 600 -2 2]),ylabel('h12')
subplot(2,2,3),plot(t,h1(2,:)); axis([0 600 -3 3]),ylabel('h21')
subplot(2,2,4),plot(t,h2(2,:)); axis([0 600 -3 3]),ylabel('h22')
```

```
%%%Roll Mode responses%%%
%Trim angle of attack
alphao=0.0142*pi/180;
thetao=alphao;

%Add the heading equation
Ax=[[Alat,zeros(4,1)];[0,0,1/cos(thetao),0,0]];
Bx=[Blat;[0,0]];

%Dutch roll, spiral and roll modes
eig(Alat);
eig(Ax);

%Aileron pulse inputs
xo=-Ax(:,1)*pi/180; %Switch sign for + right aileron down
t1=[0:.1:5];
u=zeros(size(t1));
y1=lsim(Ax,Bx(:,1),eye(5),zeros(5,1),u,t1,xo);

figure
subplot();
plot(t1,180*y1(:,2)/pi,t1,180*y1(:,4)/pi);
grid;
xlabel('Time (sec)')
ylabel('Roll Rate / Angle (deg/sec)')
title('Roll Mode')
legend('Roll Rate','Roll Angle')


%%%Spiral Mode responses%%%
%Trim angle of attack
alphao=0.0142*pi/180;
thetao=alphao;

%Add the heading equation
Ax=[[Alat,zeros(4,1)];[0,0,1/cos(thetao),0,0]];
Bx=[Blat;[0,0]];

%Dutch roll, spiral and roll modes
eig(Alat);
eig(Ax);

%Aileron pulse inputs
xo=-Ax(:,1)*pi/180; %Switch sign for + right aileron down
t1=[0:.1:5];
u=zeros(size(t1));
y1=l_sim(Ax,Bx(:,1),eye(5),zeros(5,1),u,t1,xo);

subplot(121);
plot(t1,180*y1(:,4)/pi,t1,180*y1(:,5)/pi);
grid;
xlabel('Time (sec)')
ylabel('Bank/Heading Angles (deg)')
title('Aileron Pulse Inputs')
legend('Bank Angle','Heading Angle')

%Rudder pulse inputs
xo=Ax(:,2)*pi/180;
t1=[0:.1:5];
u=zeros(size(t1));
```

```
y1=lsim(Ax,Bx(:,2),eye(5),zeros(5,1),u,t1,xo);

subplot(122);
plot(t1,180*y1(:,4)/pi,t1,180*y1(:,5)/pi);
grid;
xlabel('Time (sec)')
ylabel('Bank/Heading Angles (deg)')
title('Rudder Pulse Inputs')
legend('Bank Angle','Heading Angle')


%%%Dutch Roll Mode responses%%%
%Trim angle of attack
alphao=0.0142*pi/180;
thetao=alphao;

%Add the heading equation
Ax=[[Alat,zeros(4,1)];[0,0,1/cos(thetao),0,0]];
Bx=[Blat;[0,0]];

%Dutch roll, spiral and roll modes
eig(Alat);
eig(Ax);

%Aileron pulse inputs
xo=-Ax(:,1)*pi/180; %Switch sign for + right aileron down
t1=[0:.1:5];
u=zeros(size(t1));
y1=lsim(Ax,Bx(:,1),eye(5),zeros(5,1),u,t1,xo);
figure
subplot(121);
plot(t1,180*y1(:,1)/pi,t1,180*y1(:,3)/pi);
grid;
xlabel('Time (sec)')
ylabel('Sideslip/Yaw Rate (deg/sec)')
title('Aileron Pulse Input')
legend('Sideslip','Yaw Rate')

%Rudder pulse inputs
xo=Ax(:,2)*pi/180;
t1=[0:.1:5];
u=zeros(size(t1));
y1=l_sim(Ax,Bx(:,2),eye(5),zeros(5,1),u,t1,xo);

subplot(122);
plot(t1,180*y1(:,1)/pi,t1,180*y1(:,3)/pi);
grid;
xlabel('Time (sec)')
ylabel('Sideslip/Yaw Rate (deg/sec)')
title('Rudder Pulse Input')
legend('Sideslip','Yaw Rate')
```

```
%%% altitude hold controller %%%

for ii=1:9
  figure(ii);clf;
  set(gcf,'DefaultLineLineWidth',2);
  set(gcf,'DefaultlineMarkerSize',10)
end

% use the full longitudinal model
% x=[u w q theta h];
% u=[de;dt];

sen_u=1;sen_w=2;sen_q=3;sen_t=4;sen_h=5;sen_de=6;sen_dt=7;
act_e=1;act_t=2;
%%%%%%%%%%%%%%%%%%%
% dot h = U0 (theta - alpha) = U0 theta - w
%
% add actuator dynamics
% elevator
%tau_e=.1;%sec
tau_e=1;%sec
JNe=10;JDe=[tau_e 10];
syse=tf(JNe,JDe);
%%%%%%%%%%%%%%%%%%%
% thrust
%
tau_t=3.5;%sec
JNt=1;JDt=[tau_t 1];
syst=tf(JNt,JDt);
sysd=append(syse,syst);
syslong=ss(A,B,C,D);
syslong2=series(sysd,syslong);
[A,B,C,D]=ssdata(syslong2);
na=size(A);
%%%%%%%%%%%%%%%%%%%%
% q and theta loops
%
% inner loop on theta/q
% u=kq*q+Kth*theta + de_c
K_th=0.25;K_q=1.95*K_th;
if 1
   figure(1);clf;%orient tall
   sgrid([.5 .707]',[1]);
   rlocus(tf(-[1.95 1],1)*tf(syslong2(sen_t,act_e)))
   r_th=rlocus(tf(-[1.95 1],1)*tf(syslong2(sen_t,act_e)),K_th)
   hold on
   plot(r_th+eps*j,'bd','MarkerFace','b');axis([-3,1,-3,3]);hold off
   axis([-3 .1 -3 3])
   title('with q and theta FB to \delta_e');
end
syscl=feedback(syslong2,[K_q K_th],act_e,[sen_q sen_t],1);
[eig(syscl) [nan;r_th;nan]];
[Acl,Bcl,Ccl,Dcl]=ssdata(syscl);
%%%%%%%%%%%%%%%%%%%%
% engine loops
% mostly the phugoid ==> design on top of the short period
%
% system with q/th loop feedback
figure(9);clf
% interested in vel output and engine input
sys_u=syscl(sen_u,act_t);
```

```
Kun=[1/0.2857 1]; % comp zero on plant pole
Kud=[1/(0.2857*5) 1];
[Au,Bu,Cu,Du]=tf2ss(Kun,Kud);
%K_u=.075;
%K_u=.35;
K_u=1;
rlocus(sys_u*tf(Kun,Kud));
rr_u=rlocus(sys_u*tf(Kun,Kud),K_u);
hold on;plot(rr_u+eps*j,'bd','MarkerFace','b');hold off;
if 1
   na=size(Acl,1);
   Aclt=[Acl Bcl(:,act_t)*Cu;zeros(1,na)  Au];
   Bclt=[[Bcl(:,act_e);0] [Bcl(:,act_t)*Du;Bu]];
   Cclt=[Ccl zeros(size(Ccl,1),1)];
   Dclt=[zeros(size(Ccl,1),2)];
   figure(2);clf;axis([-.5 .05,-.4,.4]);sgrid([0.707 3]',[.05]);hold
on;
   rlocus(Aclt,Bclt(:,act_t),Cclt(sen_u,:),Dclt(sen_u,act_t));
   axis([-2 .05,-2,2])

r_u=rlocus(Aclt,Bclt(:,act_t),Cclt(sen_u,:),Dclt(sen_u,act_t),K_u)';

   hold on;plot(r_u+eps*j,'bd','MarkerFace','b'),hold off
   title('with u FB to \delta_t')
end
Acl2=Aclt-Bclt(:,act_t)*K_u*Cclt(sen_u,:);
Bcl2=Bclt;
Ccl2=Cclt;
Dcl2=Dclt;
sysclt=ss(Acl2,Bcl2,Ccl2,Dcl2);
%%%%%%%%%%%%%%%%%%%%%
% now close loop on altitude
%
% de_c = kh*(h_c-h)
if 1
    % design lead by canceling troubling plant pole
    % zero located p*8
   tt=eig(sysclt);[ee,ii]=min(abs(tt+.165));
   Khn=[1/abs(tt(ii)) 1];Khd=[1/(8*abs(tt(ii))) 1];
   K_h=-1*.0017;
   Gc_eng=tf(Khn,Khd);
   Loopt=series(append(Gc_eng,tf(1,1)),sysclt);
   figure(3);clf;
   sgrid([0.707 3]',[.1:.1:1]);
   hold on;
   rlocus(sign(K_h)*Loopt(sen_h,act_e));
   axis([-3 .1, -4 4])
   r_h=rlocus(Loopt(sen_h,act_e),K_h)
   hold on;plot(r_h+eps*j,'bd','MarkerFace','b'),hold off
   title('with h FB to \delta_e command')
end
syscl3=feedback(series(append(tf(K_h,1),tf(1,1)),Loopt),[1],act_e,se
n_h,-1);
[r_h eig(syscl3)]
%%%%%%%%%%%%%%%%%%%%%%
% try using the thrust act to control altitude
%
if 1
     %K_hu=.154;
   K_hu=0.025;
   tt=eig(sysclt);[ee,ii]=min(abs(tt+.165));
```

```
   Khnu=[1/abs(tt(ii)) 1];Khdu=[1/(5*abs(tt(ii))) 1];
   Gc_engu=tf(Khnu,Khdu);
   Loopt=series(append(tf(1,1),Gc_engu),sysclt);
   figure(4);clf;axis([-.6 .1,-.5,.5]);sgrid([.5 .707]',[.05]);hold
on;
   rlocus(sign(K_hu)*Loopt(sen_h,act_t));
   axis([-1 .1,-2,2])
   r_hu=rlocus(Loopt(sen_h,act_t),K_hu);
   hold on;plot(r_hu+eps*j,'bd','MarkerFace','b'),hold off
   title('with h FB to \delta_t command')
end
syscl4=feedback(series(append(tf(1,1),tf(K_hu,1)),Loopt),[1],act_t,s
en_h,-1);
[r_hu eig(syscl4)]
t=[0:.01:50]';
%%%%%%%%%%%% altitude 150
[ystep,t]=initial(syscl3,[0 0 0 0 150 0 0 0 0],t);
figure(5);clf;orient tall;
subplot(211)
ll=1:length(t);
U=ystep(:,sen_u);W=ystep(:,sen_w);q=ystep(:,sen_q);
TH=ystep(:,sen_t);H=ystep(:,sen_h);
DELe=ystep(:,sen_de);DELt=ystep(:,sen_dt);
plot(t(ll),H(ll),t(ll),5*U(ll),t(ll),TH(ll)*180/pi);
title('Initial Condition response to 150m altitude error. E-FB')
legend('H','5*U','\theta deg');grid
subplot(212)
plot(t(ll),DELe(ll)*180/pi,t(ll),5*DELt(ll));

legend('\delta_e input','5*\delta_t input');grid
axis([0 50 -40 40])
ylabel('Commands (degs)')
title('Initial Condition response to 150m altitude error')
%%%%%%%%%%%% altitude 70
t=[0:.01:50]';
[ystep,t]=initial(syscl3,[0 0 0 0 70 0 0 0 0],t);
figure(6);clf;orient tall;
subplot(211)
ll=1:length(t);
U=ystep(:,sen_u);W=ystep(:,sen_w);q=ystep(:,sen_q);
TH=ystep(:,sen_t);H=ystep(:,sen_h);
DELe=ystep(:,sen_de);DELt=ystep(:,sen_dt);
plot(t(ll),H(ll),t(ll),5*U(ll),t(ll),TH(ll)*180/pi);
title('Initial Condition response to 70m altitude error. E-FB')
legend('H','5*U','\theta deg');grid
subplot(212)
plot(t(ll),DELe(ll)*180/pi,t(ll),5*DELt(ll));

legend('\delta_e input','5*\delta_t input');grid
axis([0 50 -40 40])
ylabel('Commands (degs)')
title('Initial Condition response to 70m altitude error')
%%%%%%%%%%%%%% altitude 20
t=[0:.01:50]';
[ystep,t]=initial(syscl3,[0 0 0 0 20 0 0 0 0],t);
figure(7);clf;orient tall;
subplot(211)
ll=1:length(t);
U=ystep(:,sen_u);W=ystep(:,sen_w);q=ystep(:,sen_q);
TH=ystep(:,sen_t);H=ystep(:,sen_h);
DELe=ystep(:,sen_de);DELt=ystep(:,sen_dt);
```

```
plot(t(ll),H(ll),t(ll),5*U(ll),t(ll),TH(ll)*180/pi);
title('Initial Condition response to 20m altitude error. E-FB')
legend('H','5*U','\theta deg');grid
subplot(212)
plot(t(ll),DELe(ll)*180/pi,t(ll),5*DELt(ll));

legend('\delta_e input','5*\delta_t input');grid
axis([0 50 -40 40])
ylabel('Commands (degs)')
title('Initial Condition response to 20m altitude error')

%%%%%%%%%%%%%%%%
figure(8);clf
%Path to follow
Hmax=150
Hc=[zeros(25,1);Hmax/100*[0:1:100]';Hmax*ones(50,1);Hmax/100*[100:-
1:0]';zeros(50,1)];
T=0:length(Hc)-1;
[Yh,T]=lsim(syscl3(:,1),Hc,T);
U=Yh(:,sen_u);W=Yh(:,sen_w);q=Yh(:,sen_q);TH=Yh(:,sen_t);H=Yh(:,sen_
h);
DELe=Yh(:,sen_de);DELt=Yh(:,sen_dt);
plot(T,[Hc],T,H,'--');
axis([0 300 -20 160])
legend('h_c','h')
title('Altitude controller: elevator FB')
ylabel('height')
xlabel('time')

figure(15);clf
%Path to follow
Hmax=150;
Hc=[zeros(25,1);Hmax/100*[0:1:100]';Hmax*ones(50,1);Hmax/100*[100:-
1:0]';zeros(50,1)];
T=0:length(Hc)-1;
[Yh,T]=lsim(syscl4(:,2),Hc,T);
U=Yh(:,sen_u);W=Yh(:,sen_w);q=Yh(:,sen_q);TH=Yh(:,sen_t);H=Yh(:,sen_
h);
DELe=Yh(:,sen_de);DELt=Yh(:,sen_dt);
plot(T,[Hc],T,H,'--');
axis([0 300 -20 160])
legend('h_c','h')
title('Altitude controller: thrust FB')
ylabel('height')
xlabel('time')


%%% model of the aircraft for yaw %%%
% has four states= roll rate, yaw rate, sideslip angle, bank angle

states = {'beta' 'yaw' 'roll' 'phi'};
inputs = {'rudder' 'aileron'};
outputs = {'yaw' 'bank angle'};

figure(1)
axis(gca,'normal')
h = pzplot(sys1);
setoptions(h,'FreqUnits','rad/s','Grid','off');
figure(15)
impulseplot(sys1,10)
```

```
figure(2)
axis(gca,'normal')
h2 = pzplot(sys2);
setoptions(h2,'FreqUnits','rad/s','Grid','off');
figure(16)
impulseplot(sys2,10)

%% yaw dampers are performed using yaw rate as matlab code
%% rudder as the input
sys13 = sys1(3,1);% select I/O pair
figure(3)
h = bodeplot(sys13);
setoptions(h,
'FreqUnits','rad/s','MagUnits','dB','PhaseUnits','deg');

figure(4)
h = rlocusplot(sys13);
setoptions(h,'FreqUnits','rad/s')

figure(5)
h = rlocusplot(-sys13);
setoptions(h,'FreqUnits','rad/s')

% of K=0.5.
k = 0.5;
cl11 = feedback(sys13,-k);
%feedback assumes negative feedback by default

figure(6)
impulseplot(sys13,'b--',cl11,'r')
legend('open loop','closed loop',4)
%%Now the full MIMO model and
%%see from the aileron looks.
%%The feedback loop involves input 1 and output 1 of the plant:

figure(7)
cloop = feedback(sys1,-k,1,1);
impulseplot(sys1,'b--',cloop,'r',20)           % MIMO impulse
response

figure(8)
cloop = feedback(sys2,-k,1,1);
impulseplot(sys2,'b--',cloop,'r',20)           % MIMO impulse
response

%%flight control by using a washout filter.

H = tf([-0.15],[1 0.4]);
figure(9)
oloop = H * (-sys13);          % open loop'
h = rlocusplot(oloop);
setoptions(h, 'FreqUnits','rad/s')
sgrid

figure(10)
k = 234;
wof = -k * H;              % washout compensator
cloop = feedback(sys13,wof,1,1);
```

```
impulseplot(sys13,'b--',cloop,'r',20)

figure(11)
impulseplot(sys13,'b--',cloop(3,1),'r',20)
legend('open loop','closed loop',4)


%%% Model from rudder to r %%%
G_dr_r=ss(Alat,Blat(:,2),[0 0 1 0],0);
%
% rudder servo 3.33/(s+3.33)
Hr=tf(10,[1 10]);
%
% expected control gain
Kr=-1.6;
tf(G_dr_r*Hr)

figure(1);clf
set(gcf,'DefaultLineLineWidth',2)
set(gcf,'DefaultlineMarkerSize',18)
pzmap(sign(Kr)*G_dr_r*Hr);
hh=get(gcf,'children');hhh=get(hh(1),'children')

axis([-3.75 .25 -2 2])
title('Lateral autopilot: r to rudder','FontSize',12)
figure(2);clf
set(gcf,'DefaultLineLineWidth',2)
set(gcf,'DefaultlineMarkerSize',18)
rlocus(G_dr_r*Hr); % note positive sign used here
sgrid([.25 .5 .75],[.25 .5 1])
axis([-2 2 -2 2])
title('Lateral autopilot: r to \delta_r with k>0 ','FontSize',16)


figure(3);clf
set(gcf,'DefaultLineLineWidth',2)
set(gcf,'DefaultlineMarkerSize',18)
rlocus(sign(Kr)*G_dr_r*Hr); % note negative sign used here
rr=rlocus(G_dr_r*Hr,Kr);
hold on; plot(rr+eps*j,'rd','MarkerFace','r');hold off
hh=get(1,'children');hhh=get(hh(1),'children')
sgrid([.25 .5 .75],[.25 .5 1])
axis([-3.75 .25 -2 2])
title('Lateral autopilot: r to \delta_r with k<0 ','FontSize',16)

% clp response to r_comm
%sys=tf(Nrdr*Kpsi,Drdr+Kpsi*Nrdr);
sys_cl=feedback(Kr*G_dr_r*Hr,1,[1],[1],-1)
t=[0:.1:500]';
rc=ones(size(t));
[r1,t]=impulse(sys_cl,t);

% with washout Kwash=[tau 0]/[tau 1]
figure(4);clf
tau=0.15; %sec
Nwash=[tau];Dwash=[1 0.4];
w=logspace(-2,2,250);
Hw=tf(Nwash,Dwash)
Gw=freqresp(Hw,w);
loglog(w,abs(squeeze(Gw)),'LineWidth',2)
```

```
axis([.01 20 .01 2])
title('Washout filter with \tau=4.2')
ylabel('|H_w(s)|')
xlabel('Freq (rad/sec)')

%sysw));

sys_cl_w=feedback(Kr*G_dr_r*Hr,Hw,[1],[1],-1);
[r2,t]=impulse(sys_cl_w,t);
[u1,t]=lsim(Hw,r2,t);

figure(4);clf;
subplot(211)
plot(t,[r1 r2 ]);axis([0 30 -5 20 ])
ylabel('Response r');xlabel('Time')
legend('Without Washout','With Washout')
subplot(212)
plot(t,[Kr*(0-r1) Kr*(0-u1)]);axis([0 30 -5 20 ])
ylabel('Control \delta_r');xlabel('Time')
legend('Without Washout','With Washout')
```

Appendix C

GENETIC ALGORITHM CODE

C.1 Initialization of GA

- *VariableBounds* – Since this thesis is using genetic algorithms to optimise the gains of a PID controller there are going to be three strings assigned to each member of the population, these members will be comprised of a P, I and a D string that will be evaluated throughout the course of the GA.
- *EvalFN* - The evaluation function is the declaration of the file name containing the objective function.
- *Options* - the fact that they use less memory, real (floating point) numbers will be used to encode the population. This is signified in the options command, which the '1e-6' term is the floating point precision 18 and the '1' term indicates that real numbers are being used (0 indicates binary encoding is being used).
- *Initialisega* - This command combines all the previously described terms and creates an initial population of 80 real valued members between –100 and 100 with 6 decimal place precision.

C.2 Setting the Parameters of the GA

- *Bounds* - The bounds for the genetic algorithm to search within are set using this command.
- *startPop* - The starting population of the GA, '*startPop*', is defined as the population described
- *opts* - The options for the Genetic Algorithm consist of the precision of the string values i.e. 1e-6, the declaration of real coded values.
- *TermFN* - This is the declaration of the termination function for the genetic algorithm. This is used to terminate the genetic algorithm once certain criterion has been met.
- *TermOps* - This command defines the options, if any, for the termination function. In this example the termination options are set to 100, which means that the GA will reproduce one hundred generations before terminating.
- *SelectFN* - Normalised geometric selection ('normGeomSelect') is the primary selection process to be used in this project. The GAOT toolbox provides two other selection functions.
- *SelectOps* - When using the 'normGeomSelect' option, the only parameter thathas to be declared is the probability of selecting the fittest chromosome of each generation,
- *XOverFN* - Arithmetic crossover was chosen as the crossover procedure. Single point crossover is too simplistic to work effectively on a chromosome with three alleles, a more uniform crossover procedure throughout the chromosome is required. Heuristic crossover was discarded because it performs the crossover procedure a number of times and then picks the best one.
- *XOverOptions* -This is where the number of crossover points is specified.
- *mutFNs* - The 'multiNonUnifMutation', or multi non-uniformly distributed mutation operator, was chosen as the mutation operator as it is considered to function well with multiple variables.
- *MutOps* - The mutation operator takes in three options when using the 'multiNonUnifMutation' function. The first is the total number of mutations, normally set with a probability of around 0.1%

```
[T,state,outputs]=sim('PID_objfun_MSE',10);

err = outputs(:,1); %use simout block with array config in simulink
J = err'*err; %minimization index
for i=1:length(err)
    if err(i)<(-0.05*setpoint) %this reduces overshoot to below 5%,
add/replace with your own criteria
    MSE=100e300;
    end
end


%Initialising the genetic algorithm
populationSize=90;
variableBounds=[-20 20;-20 20;-20 20];
evalFN='PID_objfun_MSE';

%Change this to relevant object function
evalOps=[];
options=[1e-6 1];
initPop=initializega(populationSize,variableBounds,evalFN,evalOps,op
tions);

%Setting the parameters for the genetic algorithm
bounds=[-20 20;-20 20;-20 20];
evalFN='PID_objfun_MSE'; %change this to relevant object function
evalOps=[];
startPop=initPop;
opts=[1e-6 1 0];
termFN='maxGenTerm';
termOps=70;
selectFN='normGeomSelect';
selectOps=0.08;
xOverFNs='arithXover';
xOverOps=4;
mutFNs='unifMutation';
mutOps=8;

%Iterating the genetic algorithm
[x,endPop,bPop,traceInfo]=ga(bounds,evalFN,evalOps,startPop,opts,...
termFN,termOps,selectFN,selectOps,xOverFNs,xOverOps,mutFNs,...
mutOps);

function[x_pop,
fx_val]=Fuzzy_PID_Controller_for_pitch_Control_aircraft(x_pop,option
s);
global sys_controlled
global time
global sysrl

%Splitting the chromosones into 3 separate strings
Kp=x_pop(2);
Ki=x_pop(3);
Kd=x_pop(1);
%creating the PID controller from current values
pid_den=[1 0];
pid_num=[Kd Kp Ki];
pid_sys=tf(pid_num,pid_den); %overall PID controller
%Calculating the error
for i=1:301
```

```
error(i) = 1-y(i);
end
%Calculating the MSE
error_sq = error*error';
MSE=error_sq/max(size(error));


function [pop] = initializega(num, bounds, evalFN,evalOps,options)
% function [pop]=initializega(populationSize, variableBounds,evalFN,
%                           evalOps,options)
%    initializega creates a matrix of random numbers with
%    a number of rows equal to the populationSize and a number
%    columns equal to the number of rows in bounds plus 1 for
%    the f(x) value which is found by applying the evalFN.
%    This is used by the ga to create the population if it
%    is not supplied.
%
% pop            - the initial, evaluated, random population
% populatoinSize - the size of the population, i.e. the number to
create
% variableBounds - a matrix which contains the bounds of each
variable, i.e.
%                  [var1_high var1_low; var2_high var2_low; ....]
% evalFN         - the evaluation fn, usually the name of the .m
file for
%                  evaluation
% evalOps        - any options to be passed to the eval function
defaults []
% options        - options to the initialize function, ie.
%                  [type prec] where eps is the epsilon value
%                  and the second option is 1 for float and 0 for
binary,
%                  prec is the precision of the variables defaults
[1e-6 1]


if nargin<5
  options=[1e-6 1];
end
if nargin<4
  evalOps=[];
end

if any(evalFN<48) %Not a .m file
  if options(2)==1 %Float GA
    estr=['x=pop(i,1); pop(i,xZomeLength)=', evalFN ';'];
  else %Binary GA
    estr=['x=b2f(pop(i,:),bounds,bits); pop(i,xZomeLength)=', evalFN
';'];
  end
else %A .m file
  if options(2)==1 %Float GA
    estr=['[ pop(i,:) pop(i,xZomeLength)]=' evalFN '(pop(i,:),[0
evalOps]);'];
  else %Binary GA
    estr=['x=b2f(pop(i,:),bounds,bits);[x v]=' evalFN ...
    '(x,[0 evalOps]); pop(i,:)=[f2b(x,bounds,bits) v];'];
    end
end
```

```
numVars     = size(bounds,1);        %Number of variables
rng         = (bounds(:,2)-bounds(:,1))'; %The variable ranges'

if options(2)==1 %Float GA
  xZomeLength = numVars+1;       %Length of string is numVar + fit
  pop         = zeros(num,xZomeLength);     %Allocate the new
population
  pop(:,1:numVars)=(ones(num,1)*rng).*(rand(num,numVars))+...
    (ones(num,1)*bounds(:,1)');
else %Binary GA
  bits=calcbits(bounds,options(1));
  xZomeLength = sum(bits)+1;        %Length of string is numVar +
fit
  pop = round(rand(num,sum(bits)+1));
end

for i=1:num
  eval(estr);
end
```

CURRICULUM VITAE

Name Surname: Akçay ÇALIŞIR

Place and Date Birth: İZMİR – 28.12.1987

Address: İSTANBUL

E-Mail: akcaycalisir@hotmail.com

Bachelor Degree : Yıldız Technical University, Faculty of Mechanical Engineering, Mechanical Engineering Department

Master Degree : Istanbul Technical University, Institute of Science and Technology, Mechatronics Engineering Department

Professional Experience:

THY Technic Inc. (Full Time)
Engineering Directore, 01.02.2012 – continue

Dogus Automotive (Intern)
Department of After Sales Marketing, 23.08.2010 – 06.10.2010

Mercedes-Benz Turk Inc. (Intern)
Department of Production Assembly, 26.07.2010 – 20.08.2010

THY Technic Inc. (Intern)
Department of Production Planning and Control, 14.06.2010 – 16.07.2010

Izmir Demir Celik Inc. (Intern)
Department of Mechanical Atelier, 25.01.2010 – 05.02.2010

Izmir Demir Celik Inc. (Intern)
Department of Mechanical Atelier, 15.06.2009 – 10.07.2009
Department of Rolling Mill, 13.07.2009 – 24.07.2009
Department of Meltshop, 27.07.2009 – 07.08.2009