Expendable Launch Vehicle Flight Control-Design & Simulation with Matlab/Simulink.

Bhar Kisabo Aliyu
National Space Research and Development Agency
44 PUBLICATIONS **91** CITATIONS

Ministry of Education and Science
of the Russian Federation

Saint-Petersburg State University of Aerospace Instrumentation

Chair #11

MODERN APPROACH TO AEROSPACE VEHICLE NAVIGATION AND MOTION CONTROL SYSTEMS DESIGN.

Master Thesis on the direction “Instrumentation”

by

Aliyu Bhar Kisabo

Saint-Petersburg

2010

Abstract

This thesis explores the application of modern control technique-Linear Quadratic Gaussian control to a launch vehicle during ascent or boost phase, specifically the pitch-plane control of a generic based Expendable Launch Vehicle for a wind gust disturbance rejection scheme. Alongside the LQG autopilot, an Artificial Intelligent control approach using Fuzzy Logic was also synthesised for possible comparison between conventional and fuzzy logic control. For this reason, a classical single-input, single-output design is also presented to serve as a benchmark for the comparison.

The objective of the modern control synthesis approach presented here is to design compensation for pitch tracking and disturbance rejection. This was achieved by an effective combination of a solution to the optimal regulation problem by designing a Linear Quadratic Regulator and optimal estimation problem by designing a Linear Kalman Filter that provides a solution to a *differential Riccati equation*. A successful combination of the Optimal Regulator and Estimator was achieved in Matlab/Simulink, forming a closed-loop dynamic system that tracks the set point command reference.

Being one of the most active research and development areas in Artificial Intelligence at the present time and with vast advantages over conventional means of automatic control, the fuzzy control is based on fuzzy logic- a logical system which is much closer in spirit to human thinking and natural language than traditional logic systems. The fuzzy logic controller based on fuzzy logic provides a means of converting a linguistic control strategy based on expert knowledge into automatic control strategy. In this thesis, a PD-FLC was implemented; a general methodology for constructing a FLC and assessing its performance is described; in particular, the exposition includes the implementation of *Mamdani-style inference*. The two-input-one-output FLC with seven *membership function* resulting into a 49 rules was effectively implemented in Matlab/Simulink *FIS editor*. Hence, a Matlab/Simulink algorithm for the PD-FL autopilot was created and the *FIS editor* was integrated to it to realise the FL controller.

A comparison of the synthesised controllers reveals that fuzzy logic design approach is capable of creating controllers of satisfactory nominal and robust performance. Also it was proved in this thesis that fuzzy logic controllers are capable of achieving a shorter *rise-time* and smaller overshoot, over conventional controllers, such as linear PID controller. The major advantages of the fuzzy logic approach are realized in the design process itself. The hierarchical methodology and intuitive nature of the conventional control approach does not help to manage the selection of design parameters. Thus, fuzzy logic (FL) seems to provide a method of reducing system complexity while increasing control performance.

Современные методы проектирования систем навигации и управления движением аэрокосмических аппаратов

Реферат магистерской диссертации

Алийу Бхар Кисабо

Этот тезис исследует заявление современного контроля Линейный техникой Квадратный Гауссовский метод контроля к ракета-носителю в течение подъема или фазы повышения, в частности контроль самолета подачи родового базируемого Потребляемого Ракета-носителя. Рядом с автопилотом LQG, Искусственный Интеллектуальный подход контроля, использующий Нечеткую Логику также синтезировался для возможного сравнения между обычным и нечетким логическим контролем. Поэтому классический единственный вход, проект единственной продукции также представлен, чтобы служить в качестве точки отсчета для сравнения.

Цель современного подхода синтеза контроля, представленного здесь состоит в том, чтобы проектировать компенсацию за прослеживающую подачу и отклонение волнения. Это было достигнуто эффективной комбинацией решения оптимальной установленной проблемы, проектируя Линейный Квадратный Регулятор и оптимальную проблему оценки, проектируя Линейный Фильтр Kalman, который обеспечивает решение отличительного уравнения Riccati. Успешная комбинация Оптимального Регулятора и Оценщика была достигнута в Matlab/Simulink, формируя динамическую систему с обратной связью, которая прослеживает ссылку команды пункта набора.

Будучи одной из самых активных научно-исследовательских областей в Искусственном интеллекте в настоящее время и с обширными преимуществами перед обычными средствами автоматического контроля, нечеткий контроль основан по нечеткой логике - логической системе, которая намного более близка в духе к человеческому мышлению и естественному языку чем традиционные логические системы. Нечеткий логический диспетчер, основанный по нечеткой логике обеспечивает средства преобразования лингвистической стратегии контроля, основанной на экспертном знании в автоматическую стратегию контроля. В этом тезисе, был осуществлен ФУНТ-FLC; общая методология чтобы строить FLC и оценивать его показатель описана; в частности выставка включает внедрение вывода Mamdani-стиля. "Два входа одна продукция" FLC с семью функциями членства, заканчивающимися в 49 правил были эффективно осуществлены в Matlab/Simulink FIS редактор . Следовательно, алгоритм Matlab/Simulink для автопилота ФУНТА-FL был создан, и редактор FIS был объединен к этому, чтобы понять диспетчера FL.

Сравнение синтезируемых диспетчеров показывает, что нечеткий логический подход проекта способен к созданию диспетчеров удовлетворительной номинальной и здравой работы. Также это было доказано в этом тезисе, что нечеткие логические диспетчеры способны к достижению более короткого разового повышением и меньшего проскакивания, по обычным диспетчерам, таковы как линейный ИЗОДРОМНЫЙ С ПРЕДВАРЕНИЕМ диспетчер. Главные преимущества нечеткого логического подхода поняты в процессе проектирования непосредственно. Иерархическая методология и интуитивная природа обычного подхода контроля не помогают управлять выбором параметров проекта. Таким образом, нечеткая логика (FL), кажется, обеспечивает метод сокращения сложности системы, увеличивая работу контроля.

Acknowledgement

Exactly seven years ago (2003), I prayed to God Almighty to make me an Aerospace/Aeronautical engineer after falling in love with the said field of engineering during my final year of Bachelors' degree. This year (2010), that feat is a reality. My utmost gratitude is to God for making my dreams come through despite my unfaithfulness. I pray that exactly seven years from this year God will fully establish what He has put in my hands beyond my wildest dream of being an *International Consultant* in the field of aerospace/aeronautical engineering, in Jesus name, Amen!

I am greatly indebted to the Federal Government of Nigerian, the National Space Research and Development Agency (NARSDA), for giving me this life changing privilege of pursuing a postgraduate degree abroad. In particular, my biggest thanks goes to Dr. Femi Agboola, Engr. Kunle Fashade and the entire staff of Centre for Space Transportation and Propulsion (CSTP) Epe, for their strong support and sacrifices throughout the period.
I will also like to single out Engr. Lanre Adetoro, who has served as a mentor and source of great encouragement in the pursuit of this feat from the very beginning to its very end. His great leadership skills saw us through difficult times. Particularly, his invaluable academic ideas and source of academic materials (*emule*) were part of the great piece that made this feat a grand reality. Never will I forget to appreciate the entire team of 10, from CSTP Epe who underwent this programme with me, God will reward us all in Jesus name.

Without my family, I would not have gone anywhere this far. I will like to appreciate my lovely wife, Mrs. Funmilayo Aliyu whose companionship was everything to me. Also in the same light are, my father and mother, Mr. and Mrs. S.K Aliyu, my brothers, Turaki Aliyu and Babajoe Aliyu, my sisters, Mrs. Asabe Jeferry, Miss Talatu ALiyu, Miss Tima Aliyu. Whose persistent love, encouragement and prayers saw me through the good times and hard times of life. Never will I forget to mention my in-laws, specifically Mr. and Mrs Peter Ogun, Mr. and Mrs. Tosin Ogun, Mrs. Tope Adegoke and Her husband Mr. Kayode Adegoke, Ayodeji Ogun, Dapo Ogun, and Miss Yemi Ogun. All, made life appreciable to me with their unwavering love and consistent supplication to God on our behalf.

My biggest thanks are due to my Head-of-Department Professor Alexander Nebylov (DSc, *Honoured Scientist of the Russia Federation*), he always supported my scientific endeavours and was, and will be, the source of inspiration for my research. Coming to St. Petersburg State University of Aerospace Instrumentation (SUAI), was accompanied by mixed emotions of hype for the challenge and new experience as well as fear of the unknown-moving from a third world educational system to that of a first world. But the genius in Prof. Alexandra Nebylov made it all a life changing experience.
I consider myself extremely lucky in having found Dr. Sergey Brodsky as my thesis advisor, not only has he been an excellent scientific advisor, but he is also a great man of outstanding

scientific intellect. His continuous faith in my work and his valuable and “out-of-the-box” perspectives were very useful in the course of these studies.

I will also like to distinguish Dr. Roman Malakhanov for his interesting scientific discussions and suggestions which unravelled the scientific simplicity associated with *fuzzy logic*. My greatest thanks go to Emeritus professor, Alexandra Panferov for introducing and guiding me through the wonderful world of *Modern Control Theory* with the preciseness and clarity so typical of him.

These years as a graduate student in SUAI were also interesting in experience in a multidisciplinary research environment and being exposed to several researchers and ideas has given me a very rich and different view of engineering as a whole. On a final note, I will like to thank the brilliant and highly resourceful team of engineering experts in St. Petersburg that help in one way or the order in making this feat a grand success. Particularly, my utmost thanks go to Prof. Andrey Barabanov of the Faculty of Mathematics and Mechanics, Saint-Petersburg State University, who first introduced me to *Kalman Filter*. And never will I forget to appreciate Prof. Sergey Konstantinovich Savelyev, an outstanding man with great wealth of knowledge and experience whose approach to the solution of engineering problems is most lucid. Last but not the least is the Rector of SUAI- Prof. A. Ovodenko, a man of great integrity and sound scientific mind. Words cannot express our appreciation but I strongly believe that the impact the wonderful research environment of SUAI has on my life will be a living testimony.

May God Almighty reward you all in His own special way forever because you have all shaped my life in a very positive and unique way beyond my expectations.

Contents

List of Figures

List of Tables

Notations

List of Symbols

D	drag
I_{yy}	moment of inertia of the vehicle about the pitch axis
l_C	distance from mass centre of the vehicle to engine swivel point
l_α	distance from mass centre of vehicle to centre of pressure
L_α	aerodynamic load per unit angle of attack
M	mass of vehicle
v	forward velocity of the vehicle
v_W	wind velocity
z	normal displacement of the vehicle relative to inertial frame
α	angle of attack
θ	perturbation attitude error
θ_0	nominal attitude angle relative to the earth-fixed reference
$\dot{\theta}$	attitude rate
$\ddot{\theta}$	attitude
α_w	gust angle-of-attack $\left[-\left[\frac{v_w}{v}\right]\right]$
δ	thrust Vector deflection angle
μ_α	aerodynamic moment coefficient $\left[\frac{L_\alpha l_\alpha}{I_{yy}}\right]$
μ_C	control moment coefficient $\left[\frac{T_C l_C}{I_{yy}}\right]$
T_C	effective control thrust
T_T	total thrust
Q	maximum dynamic pressure

g	gravity acceleration
C_d	aerodynamic drag constant
ΔF_z	perturbation force in z-direction
ΔM_y	momentum about y-direction
U_0	steady-state value of linear velocity
$\dot{w}_{vech}$	perturbation angular velocity
K	closed-loop feedback gain matrix
L	Kalman filter gain matrix
x	state vector of system
$\hat{x}$	estimated state vector of the system
$x(0)$	state vector defining equilibrium of the system
u	input vector of the system
y	output vector of the system
$\hat{y}$	estimated output vector of the system
$w(t)$	state noise
$v(t)$	sensor noise
r	referential signal
e	error signal
A	state matrix
B	input matrix
C	output matrix
G	disturbance input matrix
I	identity matrix
$c0$	computed controllability matrix
$0b$	computed Observability matrix

Abbreviations

ELV	Expendable Launch vehicle
EOM	Equation of motion
LQR	Linear Quadratic Regulator
KF	Kalman Filter
FLC	Fuzzy Logic Control
FIS	Fuzzy Inference System
LQG	Linear Quadratic Gaussian
AI	Artificial Intelligence
GNC	Guidance Navigation and Control
PID	Proportional, Integral and Derivative
PD	Proportional and Derivative
$G_{C-PD}(s)$	PD controller transfer function
$G_{PD}(s)$	PD open-loop transfer function
$H_{PD}(s)$	PD closed-loop transfer function
$G_{C-PID}(s)$	PID controller transfer function
$G_{PID}(s)$	PID open-loop transfer function
$H_{PID}(s)$	PID closed-loop transfer function

Chapter 1

INTRODUCTION

1.1 Motivation

The ultimate goal of the Control engineer is to design feedback systems (also called controllers) that force the Plant to conform to a desired behaviour. Moreover, feedback can achieve the specification even with disturbances and imprecise knowledge of the Plant. One class of models for which both analysis and design are well understood is the class of linear time-invariant Plants. For these systems, there are many **Modern Control Approaches** available in literature. These include *Miu-infinity*, *H-infinity*, **Linear Quadratic Gaussian** approach, etc. Thus, allowing the design of a compensator to be systematic and relatively straightforward. Unfortunately, the systems encountered in nature, and in particularly in aerospace engineering are rarely linear time-invariant thus requiring alternative method for design of the controllers.

One popular method which has been extensively used is the 'time slice' or 'frozen' approach. The idea is to select several operating points which cover the range of the Plant's dynamics. Then, at each of them, the designer makes a linear time-invariant approximation to the Plant and designs a linear compensator for each linearized Plant. In between operating points, the parameters (gains) of the compensator are then interpolated, or scheduled (by time or any other exogenous parameter such as dynamic pressure, velocity...), thus resulting in global Control System.

Intelligent Control Systems uses various Artificial Intelligence computing approaches like Neural Networks (NN), Bayesian probability, **Fuzzy Logic**, machine learning evolutionary computation and genetic algorithms. During recent years considerable efforts has been devoted to guarantee the stability of **Fuzzy Logic Control.** Several stability analysis methods have been established, and stable designs have been introduced. This is important to pave the way for the uses of FLC in applications were no risks should be run. Aerospace is a clear example of a field where control must be sure [1].

Up to the time of this research, Controllers designed from any established means come with no guarantee on nominal stability, stability robustness or command-following. Rather any such properties are inferred from extensive computer simulations. In order words, a complete and systematic design methodology is yet to emerge.

1.2 Objectives of Thesis

Most of the work in this research has been carried out with a linear time varying system of the order two, representative of the motion of a generic launch vehicle about a nominal trajectory.

Starting with this system, the main objective is to design pitch-plane autopilots for the atmospheric ascent flight of a generic based ELV using firstly, the LQG approach; this involves the realisation of a compensator that is simple, relatively easy to implement and comes with satisfactory stability properties and performance. Moreover, an important requirement is that the choice of the feedback will be sure the deviation from nominal trajectory due to wind will be minimised. Secondly, FLC approach with *Mamdani* dynamic block. Though, a classical controller in the frequency domain is also designed to pave the way for the third objective. Thirdly, a comparative analysis of control laws, synthesis approach [2] and finally, Fuzzy Logic assessment.

1.3 Scope

A central goal of this Thesis is the successful application of the LQG control approach to a highly unstable ELV and also the demonstration of an alternative design of an autopilot for the generic based ELV using Fuzzy logic with the classical PID controller as a benchmark. To this end, great effort was taken to limit the simplifications and assumptions used in both design frameworks. However, several notable simplifications were deemed necessary and appropriate to aid in the completion of this work. These simplifications are discussed in the following sections.

1.3.1 Continuous Time Solution

Although the physical system is best described with continuous non-linear, time varying deferential equations, the math model utilized in the design framework are simplified to continuous linear time invariant (LTI) systems that characterise only dynamics associated with small perturbations from equilibrium. This linearization process is fundamental to the gain-scheduled linear control synthesis technique, in which many point designs are made along the trajectory and blended together. However, the physical implementation of the final ascent control law is digital. For this research, the design process has been restricted to continuous analysis. Studies surrounding the continuous-to-digital conversion of the controllers and its impact on system performance are left for future work.

1.3.2 Boost Phase Pitch Plane Control

The development of control synthesis methodology for the generic based ELV ascent is limited to pitch plane only. This is defined in Chapter 2. This limitation assumes that the symmetry of the ELV permits the extension of any pitch plane analysis into the lateral plane. Therefore, the lateral and the longitudinal motions will be uncoupled using reasonable assumptions consistent with past experience. Roll and yaw control are viewed as separate, decoupled problem and are not considered in this research.

1.3.3 Design point Selection

Traditionally, load created by the non-zero angle-of-attack is the most extreme. The size of the load with the most difficult design points during a boost phase trajectory are the transonic region and maximum pressure regions. The transonic region is characterized by large fluctuations in the aerodynamic derivative coefficients. The dramatic changes create elevated levels of uncertainty in the linearized equations, and emphasise the importance of quality controller during this region. The maximum dynamic pressure on the vehicle also places increased importance on controller quality. During this section of the trajectory, the structural is characterized by dynamic pressure and angle-of-attack ($Q\alpha$), serves as a physical limitation that must be accommodated by the control law.

These historic points; $Mach$ 1.0, occurs 72 seconds into the boost phase and the time at which maximum dynamic pressure, Q, occurs is at exactly 82 seconds. For the ELV, these two points occur within seconds of each other, approximately 72 seconds after launch [3]. Therefore, only one point ($Mach$ $1.0, Q = 480psf$) during the mission trajectory will be used as a base for comparison of the LQG and FL synthesis methods. In developing such modern control synthesis method for this dynamic severe point in the trajectory, it is assumed that the design method can be extended with less challenge for further design points in order to develop a control law for the entire boost phase. Obviously there will be issues surrounding the blending of controllers for the various points designs, but such issues are common to all gain-scheduled linear control designs and are not investigated here.

1.3.4 Design Method Portability

Dynamic effects associated with forces caused by flexure, slosh, aerodynamics, engine motion, and actuators are topics central to a majority of ELVs control law designs. These are not considered here. The research presented here is intended to provide a preliminary engineering insight from which its concept can be used and applied to a much general consideration

1.4 Organisation of Thesis

Following the introductory chapter (i.e., chapter 1), this Thesis is organised as follows.

Chapter 2 presents Expendable Launch Vehicles (ELVs) historical background, physical description, development of the vehicles Pitch Plane linearized EOMs with the necessary assumptions made. Mathematical and State-space model of the Launcher are presented in details. Also, the flight parameters of a Launch vehicle were given from a generic standpoint for system evaluation.

Chapter 3 brings to light the role of an Autopilot in ELVs, attitude control with regards to atmospheric ascent phase of ELV. Also, how the Autopilot influences the GNC of the vehicle. General design specifications for controllers are spelt out and those that suite this research work are put forward.

Chapter 4 summarizes the LQG design approach used to develop a pitch plane controller for the $Mach$ 1.0 point. Also includes the design methodology of the technique, Matlab/Simulink based model of the realised controller and a brief summary of the design of the State Estimator using Kalman Filter.

Chapter 5 gives a simplified design approach to the realisation of a PID controllers in the in frequency domain. Matlab/Simulink model for the controlled ELV, with simulated disturbance was realised in the process.

Chapter 6 gives a brief introduction of concept of Fuzzy Logic Control from a general standpoint, describes the Main modules in a FIS. Its implementation in a *FIS Editor* and finally, Matlab/Simulink model of the FL autopilot.

Chapter 7 provides a comparison of the control laws synthesized between the LQG, PID and FL controllers developed in chapter 4, 5 and 6 respectively. Nominal and robust performance are analysed, and the simulated pitch responses subject to wind disturbances are generated for each control design. Additionally, the capabilities and limitations of the two design processes are examined.

Chapter 8 summarises the findings of this Thesis and concludes with recommendations on areas for further investigation.

Chapter 2

Expendable Launch Vehicles

2.1 Introduction

An expendable launch vehicle (ELV) is used to carry a payload into space. The vehicles are designed to be used only once (i.e. they are “expended” during a single flight), and their components are not recovered after launch. The vehicle typically consists of several rocket stages, discarded one by one as the vehicle gains altitude and speed. The ELV design differs from that of reusable launch systems, where the vehicle is launched and recovered more than once. Reuse might seem to make systems like the Space Shuttle more cost effective than ELVs, but in practice launches using ELVs have been less expensive than Shuttle launches. Most satellites are currently launched using expendable launchers; they are perceived as having a low risk of mission failure, a short time to launch and a relatively low cost. A Shuttle orbiter is a major national asset, and its high cost (far more than a single expendable launch vehicle) and presence of a crew require stringent "man rated" flight safety precautions that increase launch and payload costs. Only five or biters were built, and the unexpected loss of two (Challenger and Columbia) significantly impacted the capacity and viability of the Shuttle program. Each loss also resulted in an extended hiatus in Shuttle flights compared to that following most expendable launch failure, each of which impacted only that model of launcher.For these reasons it is generally agreed that the Space Shuttle has not delivered on its original promise to reduce the costs of constructing and launching payloads into orbit. The Shuttle was originally intended to replace expendable launchers in the launching of satellites, but after the loss of Challenger the Shuttle was reserved for previously planned missions and those requiring a crew [4].

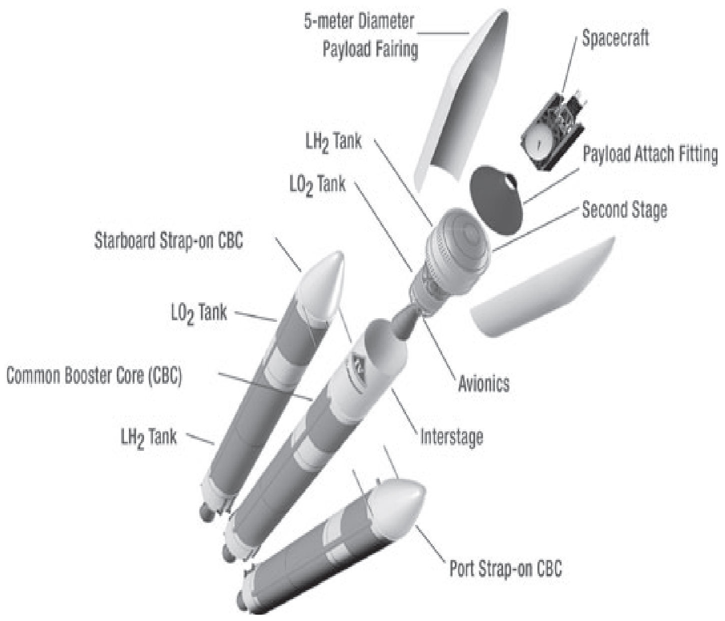


Figure 2.1: Major features of a typical Expendable Launch Vehicle

2.2 Mathematical description

The equations of motion of an ELV are complicated by the fact that the vehicle has time-varying mass and inertia. There can also be relative motion between various masses within the vehicle and origin of the body axes, such as fuel sloshing, engine gimbal rotation, and vehicle flexibility.

Expendable Launch vehicle dynamics are generally described by 'short-period' equations of motion during the atmospheric flight. Although, the 'short-period' equations are the results of a linearization process, the lateral dynamics on the pitch and yaw axes can be coupled by the vehicle roll rate. In fact, when the launch vehicle has a negligible roll rate, an action on the pitch plane produces an effect in the yaw plane.

There are many possible frames of reference to present the forces and moments that influence the motion of the vehicle in its pitch plane. The three set of axes used in this study are shown in Figure 2.1.

- The earth-fixed coordinate frame ($X_E - Z_E$) which coincide with the trajectory plane with $X_E -$axis and $Z_E -$axis coincide with the local horizontal and vertical lines of the earth at the launch pad.

- The non-rotational vehicle reference system (X_b, Z_b) where $X_b -$axis lies along the longitudinal axis of the vehicle, positive in the nominal direction of positive thrust acceleration. The $Z_b -$axis completes for a standard right-handed system.

- The trajectory inertial system (X_I, Z_I) that is tangent to the nominal trajectory of the vehicle. It is obtained from the earth-fixed coordinate frame by rotation of θ_0 (nominal attitude) about the $Z_E \times X_E -$axis.

Neglecting the effects of bending and sloshing and lags due to actuator and instrumentation, the linearized equation of motion of the vehicle have been derived in details in [5] and take the form:

Force equation:

$$\ddot{z} = -\left[\frac{T_T - D}{M} + g\cos(\theta_0)\right]\theta - \frac{L_\alpha \alpha}{M} + \frac{T_C \delta}{M} = -\frac{(T_T - D)\theta}{M} - \frac{L_\alpha \alpha}{M} + \frac{T_C \delta}{M} \tag{2.1}$$

Moment equation:

$$\ddot{\theta} = \mu_\alpha \alpha + \mu_C \delta \tag{2.2}$$

Angle of attack:

$$\alpha = \theta + \frac{\dot{z}}{v} + \alpha_W \tag{2.3}$$

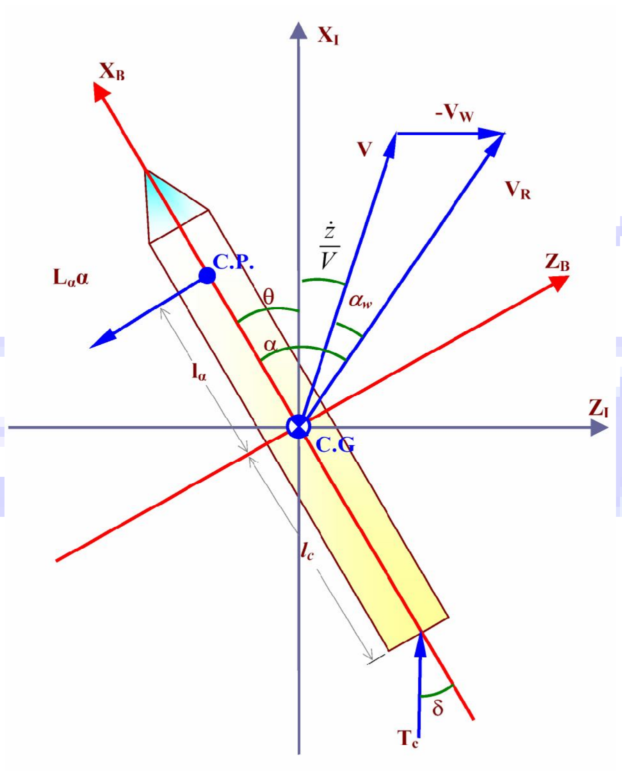


Figure 2.2: Launcher Model in the pitch plane

2.2.1 Assumptions and Simplifications

The derivations and models assumption that lead to the linearized equations of motion for the LV stated in the above section follow those stated by Greensite. The major assumptions are outlined as:

- The inertial origin is at the launch pad on a flat-Earth coordinate frame.

- The body axes origin is at C.G of the vehicle.

➢ Only rigid-body motions are considered in the pitch plane (3-DOF).

➢ The cross products of inertia are negligible as the vehicle is assumed symmetric.

➢ Aerodynamic forces are found via look-up tables.

➢ Acceleration due to gravity is constant.

➢ Engine deflections are small enough $(\delta << 1)$ such that $\sin\delta = \delta$, and $\cos\delta = 1$

➢ Inertial positions do not affect attitude dynamics directly.

The launch azimuth is assumed to be directly east.

2.2.2 State-Space Model of Launcher

Considering the equations in section 2.2 and selecting the state variables as, attitude angle, body rate and lateral drift. The State-Space model of the launch vehicle is given in [6] as,

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \mu_\alpha & 0 & \mu_\alpha \\ -L_\alpha/Mv & 0 & -L_\alpha/Mv \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ \dot{z}/v \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \mu_C & \mu_\alpha \\ T_C/Mv & -L_\alpha/Mv \end{bmatrix} \begin{bmatrix} \delta \\ \alpha_W \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ \dot{z}/v \end{bmatrix} \qquad (2.4)$$

The actuator deflection δ is a known input to the plant, whereas the wind-induced angle- of-attack α_W is an unknown input. This has to be estimated and thus, it becomes an input estimate problem, unfortunately outside the scope of this research. The attitude angle (θ) and body rate ($\dot{\theta}$) are available on-line during flight. The lateral drift $(\dot{z})$ is estimated and used thereby improving the cost and weight significantly, since there is no need for a separate sensor to measure the lateral drift. The estimated lateral drift is used to limit the vehicle angle-of-attack in high dynamic pressure region (Q) regions of the ascent. Vehicle lateral loads are proportional to the product of Q and total angle-of-attack.

The State-space model of a typical highly aerodynamically unstable Launcher in the highly dynamic pressure region is given in [7] as,

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 14.7805 & 0 & 0.01958 \\ -100.858 & 1 & -0.1256 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ \dot{z} \end{bmatrix} + \begin{bmatrix} 0 \\ 3.4858 \\ 20.42 \end{bmatrix} \delta + \begin{bmatrix} 0 \\ 14.7805 \\ -94.8557 \end{bmatrix} \alpha_w$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ \dot{z} \end{bmatrix} \tag{2.5}$$

All launch vehicles are inherently unstable and require an attitude stabilization system for successful operation. At lift-off, a launch vehicle is moving too slowly for aerodynamic stabilization through fins. A similar situation prevails once the vehicle leaves the sensible atmosphere above an altitude of about $100\,km$. The use of multi-axis rate gyroscopes to sense and feedback back the vehicles departure from the desired trajectory is a traditional method of attitude stabilization, which was operationally incorporated in the German $V-2$ rocket during World War II [8].

2.3 Parameters of a Generic Launch Vehicle

The following sets of parameters are for a generic launch vehicle based on the work of Ramnath [9].

Table 2.1: Parameter of ELV EOM simulation

S/N	Parameter	Expressional value	Numeric value ($t = 72\,\text{sec}$)
1.	$q(t)$	$t^2 \exp(-1.72 + 0.00975 \times t - 0.000276 \times t^2)$	447.89
2.	D	$684 \times q, Ibs$	$1.363 \times 10^6\,N$
3.	T_C	592000 *Ibs*	$2.634 \times 10^6\,N$
4.	$M(t)$	$171200 - 815 \times t, slugs$	$1.642 \times 10^6\,Kg$
5.	T_T	$(893 - 1.43 \times t) \times 10^4\,Ibs$	$3.516 \times 10^7\,N$
6.	μ_C	-	3.4858m/s^2
7.	μ_α	-	14.7805deg/s^2
8.	v	$\mu_\alpha / v = 0.01958$	$754.877\,m/s$
9.	v_w	-	$20\,m/s$

Note that the parameters 6, 7 and 8 of Table 2.1 were not from the work of Ramnath but from that of **Amol A**. (see ref. 7). This is necessary because Ramnath in his work considered a

Space Shuttle and ours here is an ELV hence with different *distance from mass centre of vehicle to engine swivel point* and *distance from mass centre of vehicle to centre of pressure.*

2.4 Matlab/Simulink model of the hypothetical ELV

The model described in this section is based on the equations of motion described by (2.1), (2.2) and (2.3). The time varying parameters of the Launcher were evaluated from those presented in section 2.3 at a frozen time of 72 seconds.

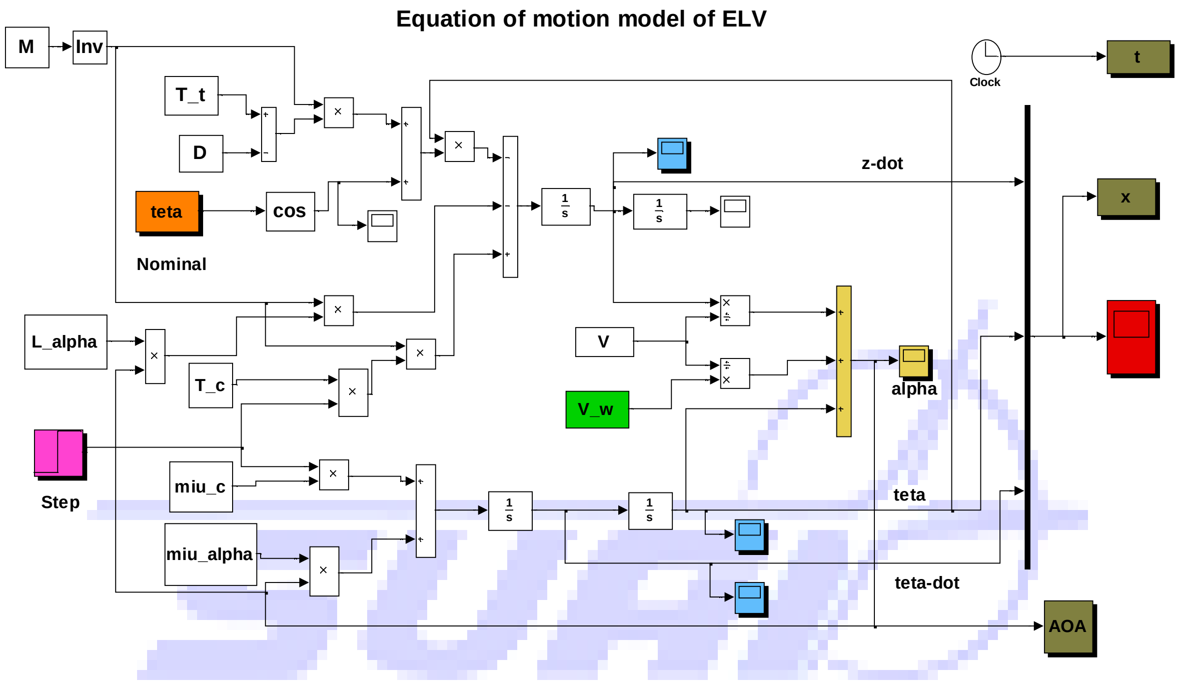


Figure 2.3: Simulink model of EOM of ELV

Figure 2.4 is the result of the simulation depicting the trend of the state vectors θ , $\dot{\theta}$ and $\dot{z}$ for the linearized equation of the ELV as given in (2.1),(2.2) and (2.3). The Matlab m-file that loads the above Simulink model are given in Appendix A.

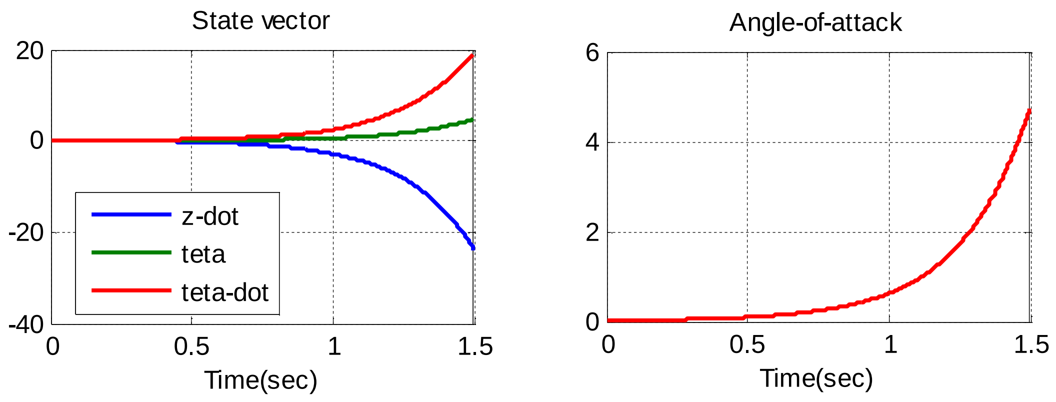


Figure 2.4: Step response of ELV EOM simulation

Chapter 3

Autopilot

3.1 Introduction

An autopilot is a mechanical, electrical or hydraulic system used to guide a vehicle without assistance from human being. Most people understand an autopilot to refer specifically to aircraft, but self-steering gear for ships, boats, spacecraft and missile is also called an autopilot.
Modern autopilot use computer software to control space vehicles. The software reads the vehicle’s current position, velocity, and attitude in space-navigation. It then proceeds to decide how to steer to the desired location-guidance. The Guidance System provides the control input necessary to fly a prescribed trajectory. These inputs are generally in the form of attitude commands whose purpose is to orient the vehicle properly. Thus, an autopilot is a close-loop system and is the inner-loop inside the main guidance loop. It primarily controls the motion of the vehicle in pitch, yaw and roll planes and its design is essentially aimed at the stability of the vehicle in the presence of disturbance forces and moments caused by various sources. In conventional launch vehicle autopilots, the function of the Steering Loop is to control the attitude rate and the function of the Flight Control System is to control the attitude in response to an attitude command supplied by the Steering Loop. With these two principles in mind, the schematic in figure 3.0 corresponds closely enough to the actual hardware implementation of a conventional single-channel Autopilot.

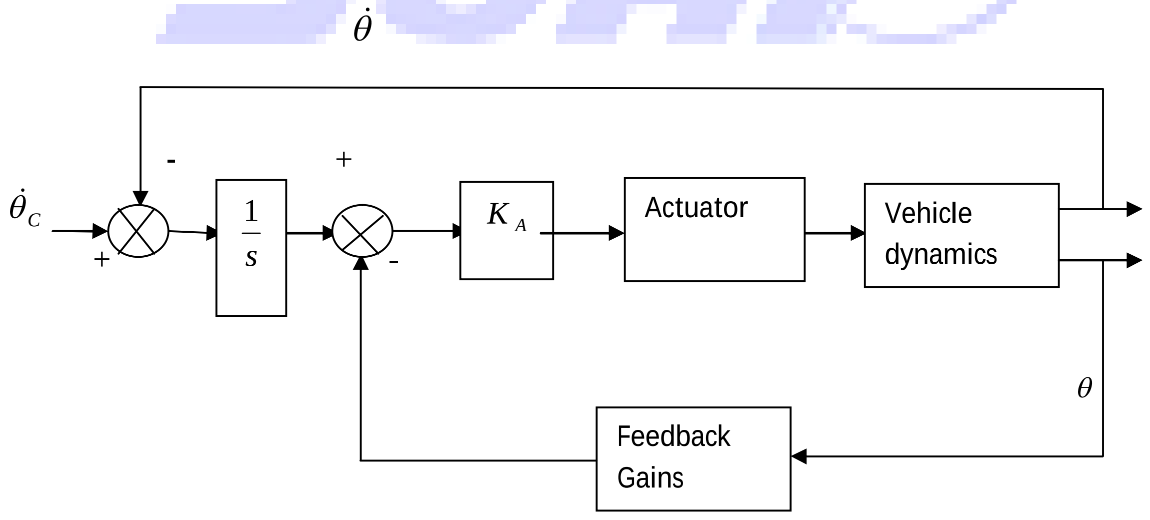


Figure 3.1: Schematic of a Pitch-Plane Autopilot

3.2 Attitude Control

The attitude of a vehicle is its orientation with respect to a defined frame of reference. Attitude control is the purposeful manipulation of controllable external forces (using vehicle actuators) to establish a desired attitude.
The attitude control problem is concerned with the short period dynamics of the vehicle where the fundamental aim is to achieve adequate stability and reasonable rapid (well-damped) response to input commands, with moderate insensitivity to external disturbances (wind and other sensor noise). The distinguishing features of attitude control problem of launch vehicle are:

- The use of swivelled engines for attitude control.
- Aerodynamic instability of the airframe.
- The extreme flexibility of the vehicle.
- The influence of propellant sloshing.
- The varying mass and inertial properties of the vehicle.

The first requirement of the attitude control system is that there exists sufficient control capability to counteract anticipated aerodynamic loads and following Greensite, this requirement can be formulated quantitatively by

$$T_C \delta_{\max} l_C > L_\alpha l_\alpha \alpha_{\max} \tag{3.1}$$

In short, the thrust control moment must be larger than the maximum anticipated aerodynamic force. Based on a maximum available thrust deflection angle $\delta_{\max}$, a maximum permissible angle-of-attack $\alpha_{\max}$ can thus be determined which in turn limits the weather conditions that the vehicle can safely fly through.

Major design problems arise because a launch vehicle is aerodynamically unstable and hugely flexible. The problem that requires most care is probably that of vehicle flexibility, which manifests itself in the sensing of undesirable local elastic deflections by the gyroscopes. When the bending mode and control frequency are of the same order of magnitude, potential stability problem exist. The use of passive filter is needed but other problems may arise by the fact that these filters compromise the gain and phase margins of the control loop and that the bending-mode frequency vary in flight. Further complications are due to sloshing of the liquid propellants and the inertia effect of the swivelling engines. Last but not the less important, the flight parameters and the inertial properties vary during flight. In conventional designs, this makes use of the so-called 'time slice' approach, in which all the parameters are ‘frozen’ over a short period of time, about a specific operating point. With this artefact, classical linear control tools are used to derive a sequence of flight control system each of them being characterised by a set of gains. In between operating points, the flight control system switches from one set of gains to another using time or dynamic pressure as gain-scheduling parameters. Of course, though these kinds of flight control systems do work in practise, they come with no guaranteed performances and extensive time-varying simulations on computers are necessary to make some refinements and finally validate them [10]. As it has been pointed out in chapter 1, the purpose of this Thesis is to apply LQG and FL techniques as means to derive a time-varying flight control system hence, compare both synthesis using the LQG synthesis as he base of comparison in order to validate the claim for stability viability of FLCs in the Aerospace industry. The first stage in designing an attitude control system is to make a rigid body analysis to determine characteristic (drift, loading, response time) and to come along with order of magnitude for the gains of the control system. The second phase is concerned with flexible body analysis to determine filters and gyro locations as well as refine the values of the gains.

This Thesis deals with phase 1 only. This work is just a preliminary design to get some crude estimates of gains variation in the control system that is necessary to begin work on phase 2. Consequently, the simplest mathematical model that takes into account for all significant phenomena such as drift and load without taking into account other phenomena such as bending, actuator dynamics of fuel sloshing are presented in section 2.2.

3.2.1 Interaction of control with Guidance law

The usual method of studying the short period motion of a launch vehicle is to analyse perturbation from a reference condition via linear method and to design an autopilot configuration that meets the specifications. Though this problem can be analysed to a large extent independently of the guidance considerations, the two cannot be completely divorced. As a matter of fact, the characteristics of the autopilot will significantly influence on trajectory dispersions and on the vehicle loading, especially during the atmospheric phase where the vehicle is subjected to high dynamic pressure and high velocity wind disturbances. Indeed if the vehicle attempt to follow the pre-specified trajectory in the presence of large wind disturbances, the angle-of-attack may be significantly increase, resulting in a large aerodynamic normal force as well as some lateral drift in the wind direction. This large aerodynamic force, in combination with the thrust component normal to the vehicle to balance the torque produced by the aerodynamic force can cause large bending moment which the vehicle may not be able to withstand. These normal forces to the vehicle can be reduced almost completely through ‘load relief’ scheme, [11] that cause the vehicle to rotate into the wind so as to reduce the angle-of-attack and corresponding loads. But these schemes lead to large deviation from the nominal trajectory.

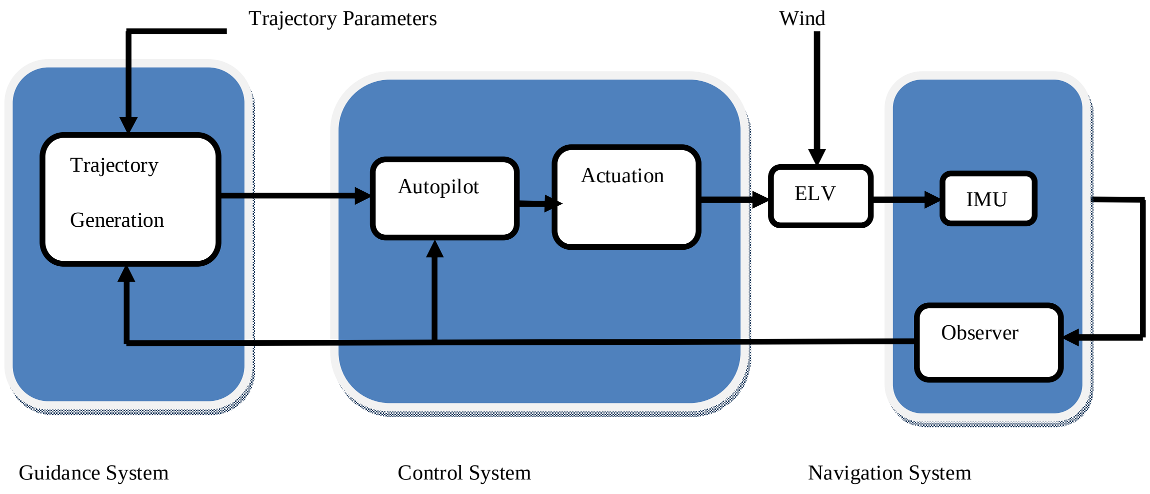


Figure 3.2: Interaction between the Guidance, Navigation and Control blocks

3.2.2 Design Specification

The primary role of an autopilot is to stabilize the vehicle and to control the attitude angle. It is also of prime importance to minimize the deviation from nominal trajectory and to limit the induced angle-of-attack in response to wind. These requirements can be presented in the form of design specifications numbered 1 to 7.

Specification 1: The compensated system must exhibit good stability behaviour in response to perturbation that deviate the state from 0. The settling time (time after which the error in θ is less than, say $0.5\deg$) must be acceptable.

Specification 2: The gains of the compensator must be such that the deviation from the nominal trajectory is minimized.

Specification 3: For a typical non-zero initial states and a typical wind profile, the thrust angle deflection (control) must not exceed a maximum allowable value $\delta_{\max}$.

Specification 4: The gains of the feedback system must not be too large to avoid magnification of the sensor noise and excitation of the un-modelled dynamics.

Specification 5: The angle of attack excursion must be minimized to avoid loss of control and excitation of the bending modes.

Specification 6: The flight control system must have good command following in the range of low frequency for compatibility with guidance law.

Specification 7: In order to be reliable and not expensive, the control system must be easy to implement.

A trade-off must be made between specification 2 and 7 because to achieve adequate short-period stability, the vehicle flight control system will be required to meet two conflicting requirements:

- To minimise trajectory deviations since excessive guidance correction results in payload penalty.

- To minimise the angle-of-attack excursion in order to reduce the resulting bending moment or in other words, ensure structural integrity of the vehicle.

Depending on which requirement the emphasis is made on, the control structure will be different. The final decision must be made by considering the specific mission (what is the worst tolerable deviation from the nominal trajectory at the end of the flight?) and the structural configuration of the airframe (what is the maximum lateral force that it can withstand?). Solving such trade-off problem of course is out of the scope of this Thesis. Rather, the emphasis has been put on the problem of minimizing the deviation from trajectory (Specification 2), [12]

3.3 Dynamic system behaviour

It is imperative to investigate the state-space model given in (2.4) and it is expected to have a trend similar to that obtained from the simulation in Fig. 2.4. To do this, *unit step* (Dirac's impulse function) is very useful for investigation of the response of dynamic system. It is given as an impulse with rectangular shape with width t_1 and its area equals one

$$\delta(t) = \begin{cases} 0, pro \cdots t \geq t_1 \\ \dfrac{1}{t_1}, pro \cdots 0 \leq t < t_1 \end{cases}$$

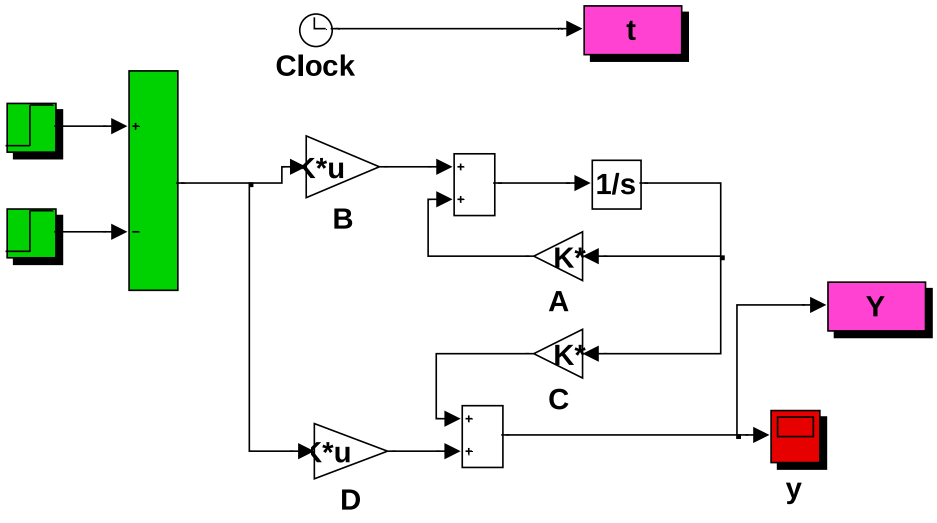


Figure 3.3: Simulink model for Dirac's investigation

In Simulink we used two blocks for step function *Step* as shown in Fig 3.1. The impulse starts in time 1 second and ending in time 3 seconds with amplitude 0.5, this satisfy's the definition of Dirac's impulse function. Its area is $0.5(3-1)=1$. [13]. The time response of the Pitch-Plane model of the autopilot is given in Fig 3.2, to a zero initial condition.

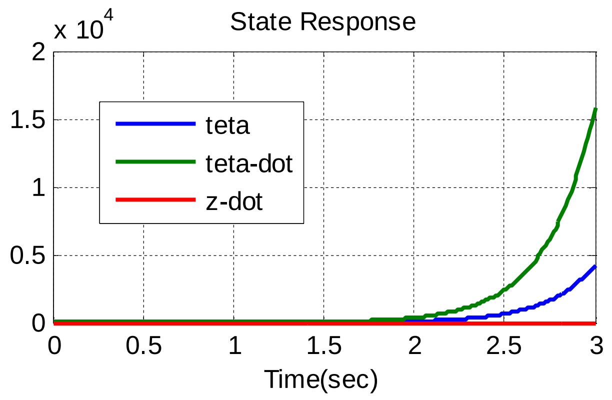


Figure 3.4: Impulse response of state vectors

It should be noted at this point that the state vectors θ and $\dot{\theta}$ have an unstable trend from what is observed in Fig 3.4 though a further investigation (stability analysis) needs to be carried out to ascertain this claim. Also it is important to note that the trend of the two state vectors in Fig 2.4 match that in Fig 3.4.

3.3.1 Stability Analysis

For the linear time-invariant state equation

$$\begin{aligned}\dot{x}(t) &= Ax(t)+Bu(t)\\ y(t) &= Cx(t)+Du(t)\end{aligned} \qquad x(0)=x_0 \tag{3.2}$$

Firstly, the notion of internal stability involves qualitative behaviour of the zero-input state response, i.e., the response of a related homogeneous state equation that depends solely on the initial state. For the linear case, there exist explicit stability criteria available that involve the eigenvalues of the system dynamics matrix A.

The second type of stability focuses on external, or input-output, behaviour. In particular, we characterize state equations for which the zero-state output response is a bounded signal for every bounded input signal. This is referred to, not surprisingly, as *bounded input, bounded-output stability*. Although these internal and external stability properties are fundamentally different in character, as they pertain to different response components, they are related, though this fact will not be exploited in this work, because only the notion of internal stability will be applied for the reason of the limitation(s) of the control strategy that will be applied to the ELV.

Internal Stability: The stability-related notions we consider are illustrated in Figure 3.3. Point a represents an unstable equilibrium. A ball perfectly balanced atop the curved surface on the left will remain at rest if undisturbed. However, the slightest perturbation will cause the ball to roll away from that rest position. Point b illustrates a stable equilibrium in the sense that the ball will move only a small distance from a rest position on the flat surface when

experiencing a perturbation that is suitably small. Finally, point c depicts an asymptotically stable equilibrium. Suppose that the ball is initially at rest nestled in the curved surface on the right. For a reasonably
small perturbation, the ball will not stray too far from the rest position and, in addition, eventually will return to the rest position.

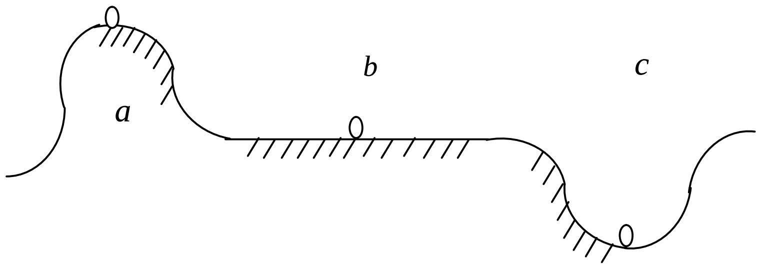


Figure 3.5: Equilibrium state

Eigenvalue Analysis If all real parts of all system eigenvalues are strictly negative, the system is stable. If just one real part of an eigenvalue is zero (and the real parts of the remaining system eigenvalues are zero or strictly negative), the system is marginally stable. If just one real part of an eigenvalue is positive (regardless of the real parts of the remaining System eigenvalues), the system is unstable [14].
The open-loop system eigenvalues for our system, found from the eigenvalues of A, are $S_{1,2,3} = -3.9143, 3.7807, 0.008$. Thus, this open-loop system is unstable, because only one real root is negative.

3.4 Controllability and Observability

The concept of controllability was introduced by Kalman (1960) and plays an important role in the control of multivariable systems.

Considering a system as described by (3.2), the system is said to be controllable if a control vector $u(t)$ exist that will transfer the system from any initial state $x(t_0)$ to some final state $x(t)$ in a finite time interval. Sufficient condition for complete state controllability is that the $n \times n$ matrix

$$M = [B : AB : \ldots : A^{n-1}B] \tag{3.3}$$

Contains n linearly independent row or column vectors, i.e. is of the rank n (that is, the matrix is non-singular, i.e. the determinant is non-zero). Thus, (3.3) is called the controllability matrix.

The system described by (3.2) is completely observable if the $n \times n$ matrix

$$N = [C^T : A^T C^T : \ldots : (A^T)^{n-1} C^T] \tag{3.4}$$

Is of the rank n, i.e. is non-singular having a non-zero determinant. Thus, (3.4) is called the Observability matrix.

For the state-space equation of the ELV given in (2.5) MATLAB code with Matlab in-built functions were used to compute controllability and Observability matrices (see Appendix);

$$c0 = \begin{bmatrix} 0 & 3.4858 & 0.3998 \\ 3.4858 & 0.3998 & 51.5399 \\ 20.420 & 0.9210 & -351.2867 \end{bmatrix} \tag{3.4}$$

$$ob = \begin{bmatrix} 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & 1.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 1.0000 & 0.0000 \\ 14.7805 & 0.0000 & 0.0196 \\ 0.0000 & 0.0000 & 0.0000 \\ 14.7805 & 0.0000 & 0.0196 \\ -1.9748 & 14.8001 & -0.0025 \\ 0.0000 & 0.0000 & 0.0000 \end{bmatrix} \tag{3.5}$$

The above results proves that the system is both completely state controllable and state observable, hence makes the system suitable for LQG controller design.

Chapter 4

Linear Quadratic Gaussian Synthesis

4.1 Introduction

The solution of the LQG control problem is one of the most celebrated results in the control and systems field. There are actually several LQG control problems depending on the information available for computing the control law.

A control system that contains a Linear Quadratic Regulator controller together with a Kalman filter State estimator as shown in figure 4.1 is called a Linear Quadratic Gaussian control system.

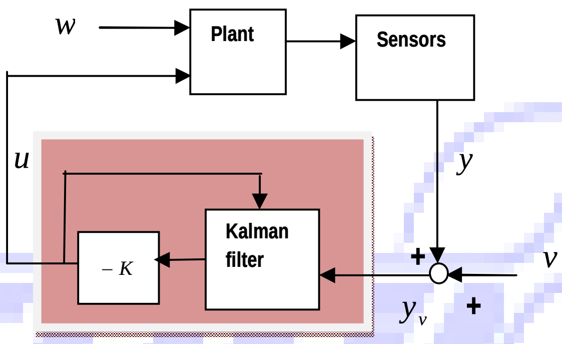


Figure 4.1: Schematic block for LQG controller

The Linear Quadratic Regulator is an optimal control system which seeks to minimize the return from a system for the minimum cost. In general terms, the optimal control problem is to find u which causes the system

$$\dot{x} = g(x(t), u(t), t) \tag{4.1}$$

To follow an optimal trajectory $x(t)$ that minimizes the performance criterion, or cost function

$$J = \int_{t_0}^{t_1} h(x(t), u(t), t)\,dt \tag{4.2}$$

The problem is one of constrained functional minimization, and has several approaches. The Hamilton-Jacobi equation is usually solved for the important and special case of linear time-invariant plant with quadratic performance criterion (called the performance), which takes the form of the matrix Riccati equation. This produces an optimal control law as a linear function of the state vector components which is always stable provided the system is controllable.

4.2 Linear Quadratic Regulator (LQR)

The linear Quadratic Regulator provides an optimal control law for a linear system with a quadratic performance index.

Define a function equation of the form

$$f(x,t) = \min_u \int_{t_0}^{t_1} h(x,u)dt \tag{4.3}$$

Where over the time interval t_0 to t_1,

$$f(x,t_0) = f(x(0))$$
$$f(x,t_1) = 0$$

A Hamilton-Jacobi equation may be expressed as

$$\frac{\partial f}{\partial t} = -\min_u \left[h(x,u) + \left(\frac{\partial f}{\partial x} \right)^T g(x,u) \right] \tag{4.4}$$

For a linear, time invariant plant, (4.3) becomes

$$\dot{x} = Ax + Bu \tag{4.5}$$

And if (4.1) is a quadratic performance index

$$J = \int_{t_0}^{t_1} (x^T Qx + u^T Ru)dt \tag{4.6}$$

Where Q is an $l \times l$ symmetric positive-definite matrix, and R an $m \times m$ symmetric positive-definite matrix. A first selection for the matrices $Q$and $R$is given by *Bryson's rule*: select $Q$and Rdiagonal with

$$Q_{ii} = \frac{1}{\max imum.acceptable.value.of..{z_i}^2} \qquad i \in \{1,2,...,l\}$$

$$R_{jj} = \frac{1}{\max imum..acceptable..value..of..u_j^{\,2}} \qquad j \in \{1,2...,m\} \tag{4.7}$$

In essence the Bryson's rule scales the variables that appear in (4.6) so that the *maximum acceptable value for each term is one*. This is especially important when the unit used for the different components of u and x makes the values for these variables numerically different from each other.

Although Bryson's rule sometimes gives good results, often it is just the starting point to a trail-and-error iterative design procedure aimed at obtaining desirable properties for the closed-loop system [15].

To set matrices of the weighing factor for this research the values adopted as the maximum acceptable values were given in [16] and the process of realising Q and R weighing coefficient are explicitly presented in the Appendix.

Thus,

$$Q = \begin{bmatrix} 2.5 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 173.6 \end{bmatrix},$$

$$R = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}. \tag{4.8}$$

Substituting (4.5) and (4.6) in (4.4) gives

$$\frac{\partial f}{\partial t} = -\min_u \left[x^T Qx + u^T Ru + \left(\frac{\partial f}{\partial x} \right)^T (Ax + Bu) \right]. \tag{4.9}$$

Introducing a relationship of the form

$$f(x,t) = x^T Px. \tag{4.10}$$

Where P is a square, symmetric matrix, then

$$\frac{\partial f}{\partial t} = x^T \frac{\partial}{\partial t} Px, \tag{4.11}$$

and

$$\frac{\partial f}{\partial t} = x^T \frac{\partial}{\partial t} Px,$$

$$\left[\frac{\partial f}{\partial x}\right]^T = 2x^T P. \tag{4.12}$$

Inserting (4.11) and (4.12) into (4.9) gives

$$x^T \frac{\partial P}{\partial t} x = -\min_u \left[x^T Qx + u^T Ru + 2x^T P(Ax + Bu)\right]. \tag{4.13}$$

To minimize u , from equation (4.10)

$$\frac{\partial[\partial f / \partial t]}{\partial u} = 2u^T R + 2x^T PB = 0. \tag{4.14}$$

Now (4.14) can be re-arranged to give the optimal control law

$$u_{opt} = -R^{-1} B^T Px, \tag{4.15}$$

or

$$u_{opt} = -Kx, \tag{4.16}$$

where,

$$K = R^{-1} B^T P. \tag{4.17}$$

Substituting (4.15) back into (4.13) gives

$$x^T \dot{P} x = -x^T (Q + 2PA - PBR^{-1} B^T P)x, \tag{4.18}$$

since

$$2x^T PAx = x^T (A^T P + PA)x,$$

then

$$\dot{P} = -PA - A^T P - Q + PBR^{-1} B^T P. \tag{4.19}$$

Note that (4.17) belongs to the class of nonlinear differential equations known as the matrix of Riccati equations.

The coefficients of $P(t)$ are found by integration in reverse time starting with the boundary condition

$$x^T(t_1)P(t_1)x(t_1) = 0. \tag{4.20}$$

Kalman demonstrated that as integration the reverse time proceeds, the solution of $P(t)$ converges to constant values. Should t_1 be infinite, or far removed from t_0, the matrix of Riccati equations reduce to a set of simultaneous equations

$$PA + A^T P + Q - PBR^{-1}B^T P = 0. \tag{4.21}$$

Therefore, (4.17) and (4.19) are the continuous solution of the matrix of Riccati equation . The Matlab in-built function $[K, P, E] = lqr(sys, Q, R)$ was used to obtain the solution to the following; P, of the associated algebraic Riccati equation as stated in (4.21), the state feedback gain matrix K,as given in (4.17), and $E = eig(A - BK)$, the closed-loop eigenvalues for the associated ELV system. These values are;

$$P = \begin{bmatrix} 3.2948 & 0.0796 & 0.0101 \\ 0.0796 & 0.0992 & 0.0139 \\ 0.0101 & 0.0139 & 0.0449 \end{bmatrix}, \tag{4.22}$$

$$K = \begin{bmatrix} 4.8283 & 6.2977 & 9.6519 \\ 2.2162 & 1.4861 & -40.5306 \end{bmatrix},$$

(4.23)

and

$$E = -40437, -0.8 \text{ and } -41.2\,. \tag{4.24}$$

4.2.1 State variable feedback design

Considering a system described by the state and output equations

$$\begin{aligned} \dot{x} &= Ax + Bu, \\ y &= Cx. \end{aligned} \tag{4.25}$$

Selecting a control law of the form

$$u = (r - Kx), \tag{4.26}$$

where, $r = 0$.

Equations, (4.25) and (4.26) are represented in block diagram form in Fig 4.2. Substituting (4.25) into (4.26) gives

$$\dot{x} = Ax + B(r - Kx),$$

or

$$\dot{x} = (A - BK)x + Br. \tag{4.27}$$

In (4.27), the matrix $(A - BK)$ is the closed-loop system matrix. For the system described by (4.25), the characteristic equation is given by

$$|(sI - A)| = 0. \tag{4.28}$$

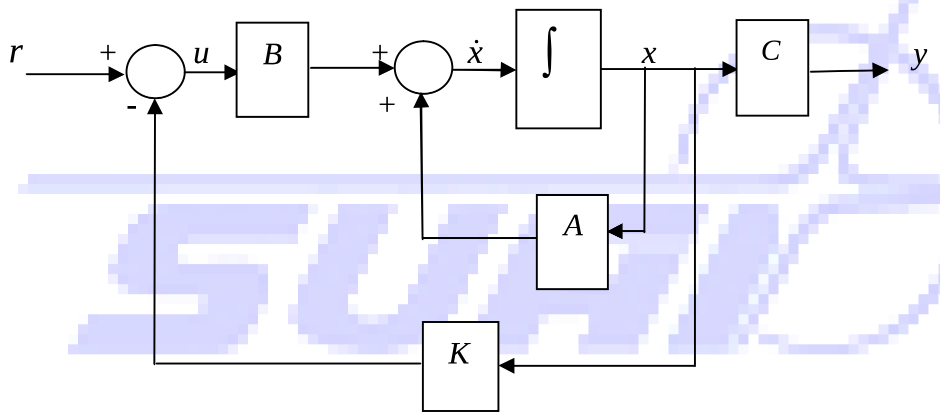


Figure 4.2: Optimal regulator block diagram

The roots of (4.28) are the open-loop poles or eigenvalues. For the closed-loop system described by (4.27), the characteristic equation is

$$|(sI - A + BK)| = 0. \tag{4.29}$$

The roots of (4.29) are the closed-loop poles or eigenvalues [17].

4.2.2 Matlab/Simulink synthesis of the Optimal Regulator

The optimal control law was implemented by using

$$u_1 = Ke, \tag{4.30}$$

where

$$e = (r - x). \tag{4.31}$$

The desired state vectors are

$$r = [0.05 \quad 0 \quad 0]^T. \tag{4.32}$$

The initial conditions are

$$x_0 = [0.012 \quad 0 \quad 0]^T. \tag{4.33}$$

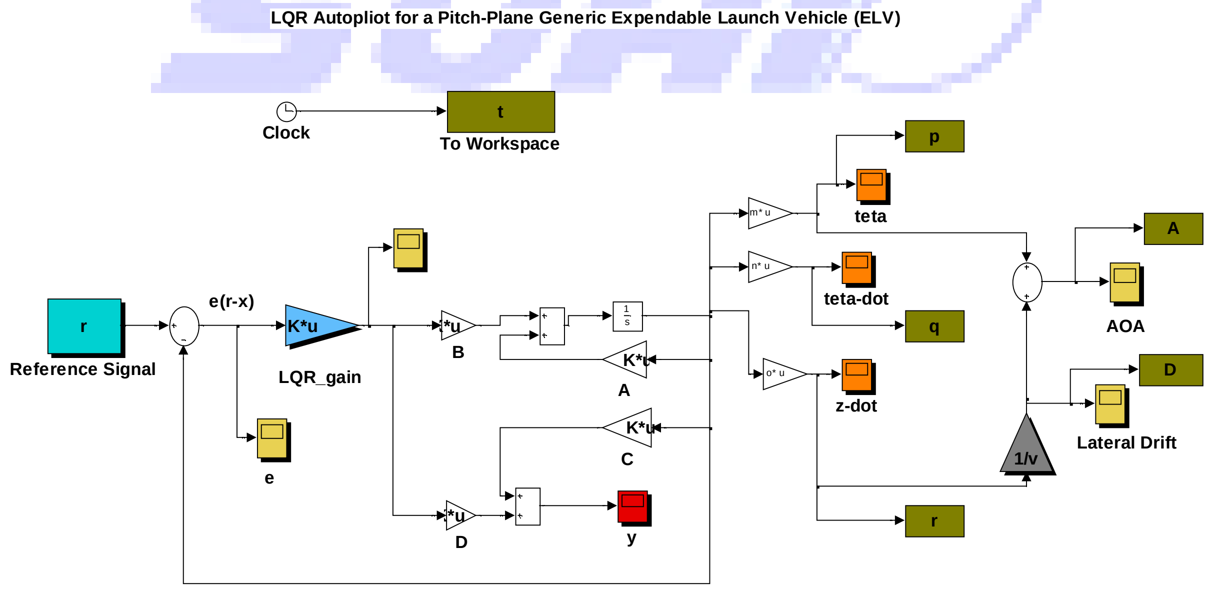


Figure 4.3: Simulink model of the LQR autopilot

Note that Fig 4.3 contains some blocks which are not part of the design of an optimal controller (the far right hand side blocks connected by a summation block). They were added to aid the investigation of lateral drift and angle-of-attack during the stabilization of the plant.

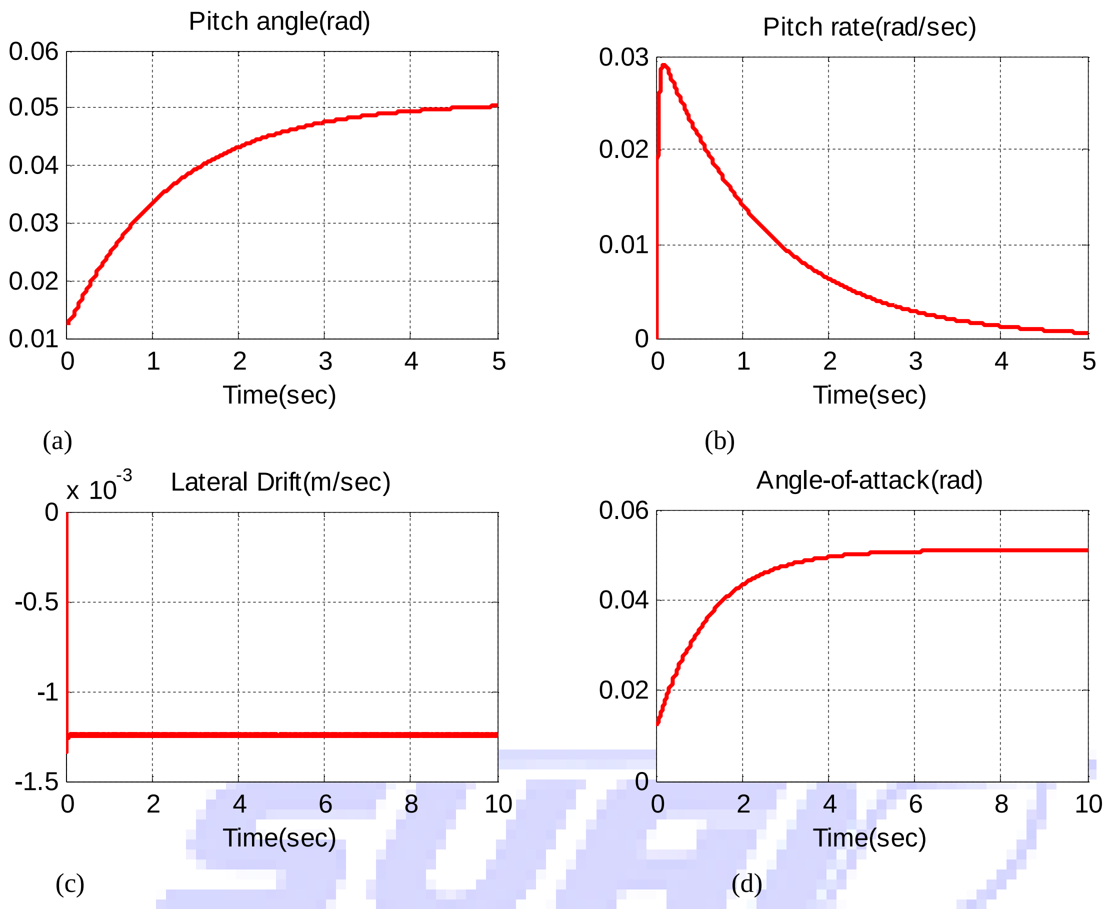


Figure 4.4: Plots of LQR autopilot set point command tracking.

The results of the LQR controller in Fig 4.4 shows good stabilization of the plant, with respect to pitch angle, but it should be noted that wind gust was not rejected from the angle-of-attack-Fig. 4.4 (d), hence it is imperative to implement LQG.

4.3 Observers and Observer-based Compensators

Considering the linear time-invariant state equation as presented in (4.25). We know that if the pair (A, B) is controllable then, a state feedback control law can be constructed to arbitrarily locate the closed-loop eigenvalues. While this is an extremely powerful result, there is one serious drawback. It is unreasonable to expect in real-world applications that every state variable is measurable, which jeopardizes the ability to implement a state feedback control law. This fact motivated our pursuit of an estimate of the state vector derived from measurements of the input and output and led us to the concept of Observability. In particular, we established the fundamental result that the initial state can be uniquely determined (and therefore, the state trajectory can be reconstructed) from input and output measurements when and only when the state equation is observable.
Note that for the particular system in this research, it was established in section 3.4, that the ELV is Observable. A full-order state observer estimates all of the system state variables. If,

however, some of the state variables are measured, it may only be necessary to estimate a few of them. This is referred to as a reduced-order state observer. Observer is itself a dynamic system that for an observable state equation can be designed to provide an asymptotically convergent estimate of the state.

The system property of *detectability* is related to Observability in the same way that stabilizability is related to controllability. It is a proven fact that Observability implies detectability but the converse those not hold. A state equation can be detectable but not observable. Thus, detectability is regarded as a weaker condition than Observability, just as stabilizability is a weaker condition than controllability. It can then be generalised as follows: if a pair (A, C) is observable, then it is detectable. Note that state feedback control laws can be implemented using the Observer-generated state estimate in place of the actual state. The result is an observer-based compensator that has very special properties.

For an $n-$dimensional linear state equation we define a linear state observer to be an $n-$dimensional linear state equation that accepts u and y as input and whose state represents the estimate of x.

Suppose we construct the estimate $\hat{x}$ by replacing the process dynamic as in

$$\dot{\hat{x}} = A\hat{x} + Bu. \tag{4.34}$$

To see of this would generate a good estimate for x, we can define the state estimation error in (4.34) and study its dynamics

$$e = x - \hat{x}. \tag{4.35}$$

From (4.25) and (4.34), we can conclude that

$$\dot{e} = Ax - A\hat{x} = Ae. \tag{4.36}$$

This shows that when the matrix A is asymptotically stable the error e converges to zero for any input u, which is good news because it means that $\hat{x}$ eventually converges to x as $t \to \infty$. However, when A is not stable e is unbounded and $\hat{x}$ grow further and further apart from x as $t \to \infty$. To avoid this, one includes a correction term in (4.34):

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - \hat{y}), \qquad \hat{y} = C\hat{x}. \tag{4.37}$$

Where $\hat{y}$ should be viewed as an estimate of y and L a given $n \times k$ matrix. When $\hat{x}$ is equal (or very close) to x, then $\hat{y}$ will be equal (or very close) to y and the correction term $L(y - \hat{y})$ plays no role. However, when $\hat{x}$ grows away from x, this term will (hopefully!) correct the error. To see how this can be done, we re-write the estimation error dynamics for (4.25) and (4.34) as:

$$\dot{e} = Ax - A\hat{x} - L(Cx - C\hat{x}) = (A - LC)e. \tag{4.38}$$

Now e converges to zero as long as $A - LC$ is asymptotically stable. It turns out that, even when A is unstable, in general we will be able to select L so that $A - LC$ is asymptotically stable. The system in (4.37) can be re-written as

$$\dot{\hat{x}} = (A - LC)\hat{x} + Bu + Ly. \tag{4.39}$$

Now (4.39) is called a *full-order observer* for the process. Full-order observers have two inputs- the process' control input u and its measured output y (via sensors) - and a single output-the state estimate $\hat{x}$.

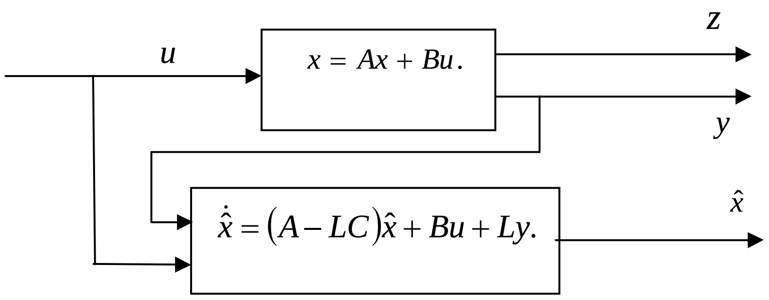


Figure 4.5: Full-order observer block diagram

A diagram showing how a full-order observer is connected to the process is shown in Figure 4.5. The observer state equation is shown graphically in detail block diagram in Fig 4.6 [18].

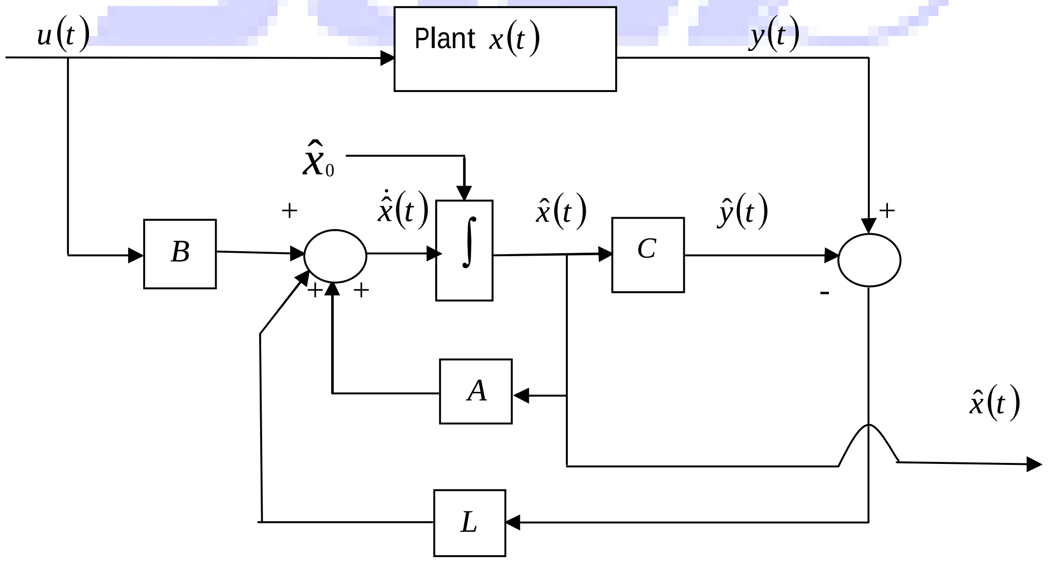


Figure 4.6: Detailed Observer block diagram

In the design of linear state observer it is assumed that the plant and measurement are noise free. In practice, this is not usually the case and therefore the observer state vector $\hat{x}$ may also be contaminated with noise.

4.3.1 Kalman Filter

Theoretically the Kalman Filter is an estimator for the linear-quadratic problem, which is the problem of estimating the instantaneous ‘state’ of a linear dynamic system perturbed by white noise-by using measurements linearly related to the state but corrupted by white noise. The resulting estimator is statistically optimal with respect to any quadratic function of the estimation error. Practically, it is certainly one of the greater discoveries in the history of statistical estimation theory and possibly the greatest discovery of the twentieth century. It has enable humankind to do many things that could not have been done without it, and it has become as indispensable as silicon in the makeup of electronic systems. It most immediate application has been in the control of complex dynamic systems such as continuous manufacturing processes, aircraft, ships, or spacecraft. To control a dynamic system, you must first know what it is doing. For these applications, it is not always possible or desirable to measure every variable that you want to control, and Kalman filter provides a means of inferring the missing information from indirect (and noisy) measurements. The Kalman Filter is also used for predicting the likely future course of dynamic systems that people are not likely to control, such as the flow of rivers during flood, the trajectory of celestial bodies, or the prices of traded commodities. Kalman Filter has been called ‘ideally suited to digital computer implementation’, in part because it uses a finite representation of the estimated problem-by a *finite* number of variables. It does, however, assume that these variables are *real* numbers-with *infinite* precision. Some of the problems encountered in its use arise from its distinction between ‘finite’ and ‘manageable’ problem sizes. These are all issues on the practical side of Kalman filtering that must be considered along with the theory. Also it is a complete statistical characterization of an estimated problem. It is much more than an estimator, because it propagates the entire probability distribution of the variables in its task to estimate. This is a complete characterization of the current state of knowledge of the dynamic system, including influence of all past measurements. These probability distributions are also useful for statistical analysis and predictive design of sensor systems. The applications of Kalman filtering encompass many fields, but its use as a tool is almost exclusively for two purposes: *estimation* and *performance* analysis of estimators. Figure 4.7 depicts the essential the essential subject forming the foundation for Kalman filtering theory. Although this shows Kalman filtering as the apex of the pyramid, it is itself but part of the foundation of another discipline-‘modern’ control theory- and a proper subset of statistical decision theory [19].

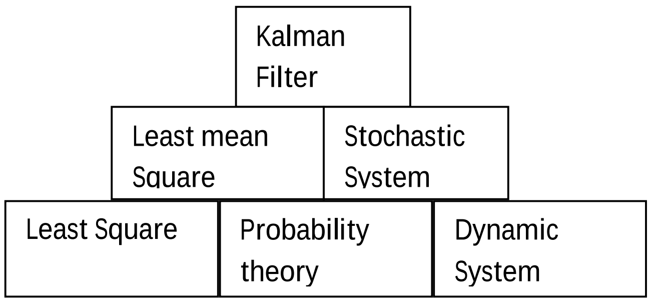


Figure 4.7: Foundation concept in Kalman filtering.

4.3.2 The Kalman Filter based-Observer

Any choice of L in (4.39) for which $A - LC$ is asymptotically stable will make $\hat{x}$ converge to x, as long as the process dynamics is given by (4.25). They all assume that the states and outputs can be measured without errors at all. In real world system there would be noise on these measurements, and in order to analyse this problem realistically noise need to be added to the model. In general the output y is affected by measurement noise and the process dynamics are also affected by disturbance, specifically turbulence of atmosphere. In light of this, a more reasonable model for the process is

$$\begin{aligned} \dot{x} &= Ax + Bu + Gw(t), \\ y &= Cx + v(t), \end{aligned} \tag{4.40}$$

where $w(t)$ and $v(t)$ are zero-mean Gaussian noise processes (uncorrelated from each other) with power spectrum,

$$S_w(\omega) = Q_N, \qquad S_v(\omega) = R_N, \qquad \forall \omega .$$

Taking,

$$Q_N = 0.0326, \qquad R_N = \begin{bmatrix} 0.002 & 0 & 0 \\ 0 & 0.002 & 0 \\ 0 & 0 & 0.002 \end{bmatrix}. \tag{4.41}$$

It is important to note that the value for Q_N was evaluated considering severe storm mean square velocity of wind [20], detail formula and evaluation are presented in appendix. Also, the initial covariance matrix of estimation was taken as

$$P_0 = \begin{bmatrix} 6.52\times10^{-5} & 0 & 0 \\ 0 & 6.52\times10^{-5} & 0 \\ 0 & 0 & 6.52\times10^{-5} \end{bmatrix}. \tag{4.42}$$

In light of (4.40) we need to re-write the estimation error dynamics presented in (4.38), this leads to

$$\dot{e} = (A - LC)e + Gw(t) - v(t). \tag{4.43}$$

Because of $Gw(t)$ and $v(t)$, the estimated error will generally not converge to zero, but we would still like it to remain small by appropriate choice of the matrix L. This motivates the so called *Linear-quadratic Gaussian (LQG) estimation problem*:
Find the matrix L that minimizes the asymptotic expected value of the estimated error:

$$J_{LQG} = \lim_{t\to\infty} \mathrm{E}\left[\left\|e(t)\right\|^2\right]. \tag{4.44}$$

The solution to the *optimal LQG Problem* formulated in (4.44) is the LQG estimator gain matrix L (an $n \times k$ matrix) given by

$$L = PC' R_N^{-1}. \tag{4.45}$$

Note that L is the Kalman gain matrix as expressed in equation and P is the unique positive-definite solution to the differential Riccati equation,

$$AP + PA' + BQB - PC' R^{-1} CP = \dot{P}. \tag{4.46}$$

Hence, a detail LQG control block diagram [21] is depicted in Fig. 4.8

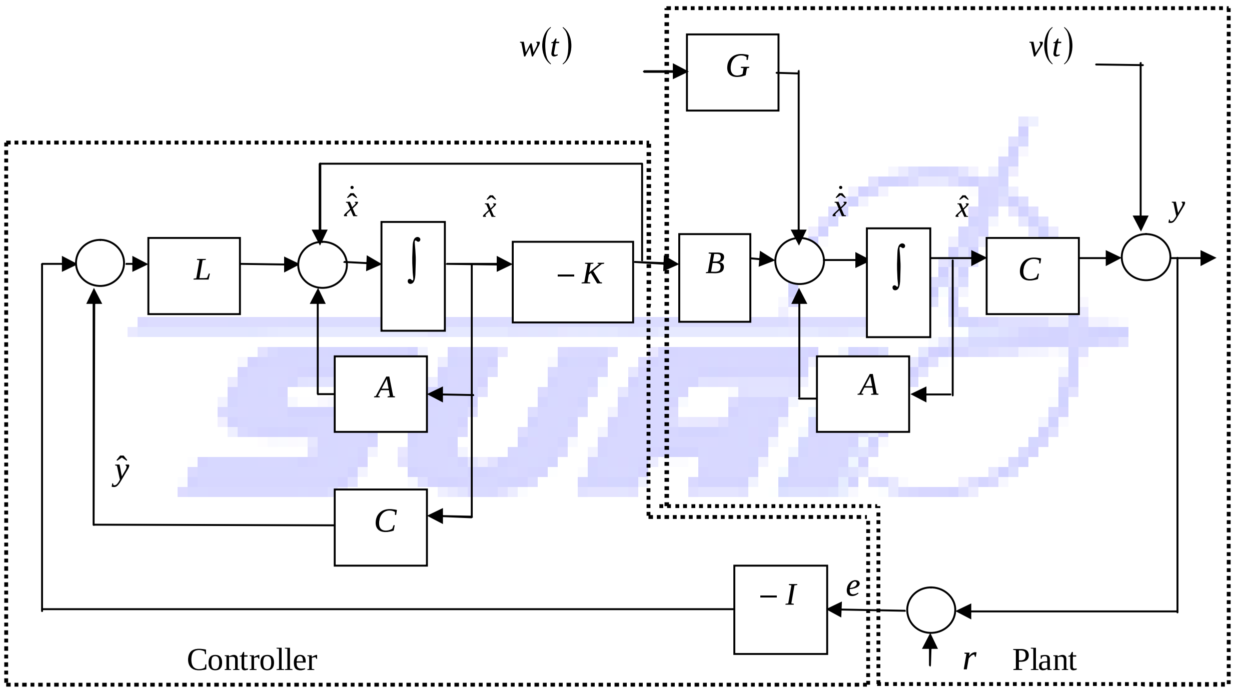
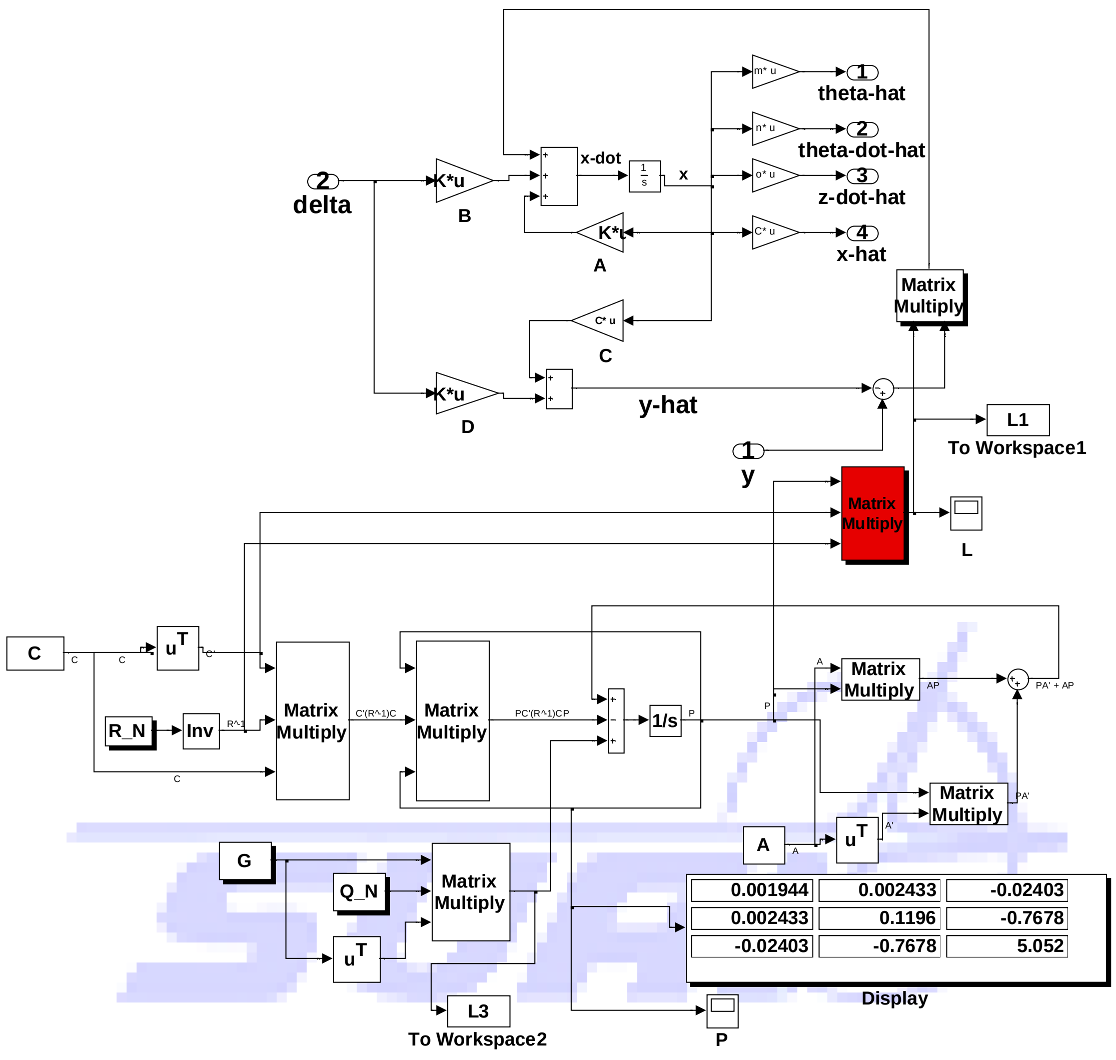


Figure 4.8: Detail block diagram of LQG controlled plant

4.3.3 Matlab/Simulink synthesis of the Linear Kalman filter

The choice for obtaining the solution to L using a differential Riccati equation as expressed in (4.46) was prompted by the lack of optimality observed in the simulated result presented in [22] at about $0.75\,\text{sec}$ of the simulation time. The Simulink model for the Kalman filter implemented in this research is presented in Fig 4.9.



0.001944	0.002433	-0.02403
0.002433	0.1196	-0.7678
-0.02403	-0.7678	5.052

Figure 4.9: Simulink model of the implemented Kalman-Bucy filter

Hence, for the specific choice of Q and R as presented in (4.41) and P_0 as given in (4.42), the following were realised after running the simulation in Fig 4.10

$$L = \begin{bmatrix} 1.1614 & 1.2600 & 0 \\ 1.2600 & 59.8196 & 0 \\ -12.5508 & -383.9599 & 0 \end{bmatrix}, \qquad (4.47)$$

$$A - LC = \begin{bmatrix} -1.1614 & -0.2600 & 0 \\ 13.5205 & -59.8196 & 0.0196 \\ -88.3072 & 384.9599 & -0.1256 \end{bmatrix}, \qquad (4.48)$$

$$eig(A - LC) = -59.8857\ , -1.2214, \text{ and } 0.0005 \tag{4.49}$$

while,

$$P = \begin{bmatrix} 0.001944 & 0.002433 & -0.02403 \\ 0.002433 & 0.1196 & -0.7678 \\ -0.02403 & -0.7678 & 5.052 \end{bmatrix}. \tag{4.50}$$

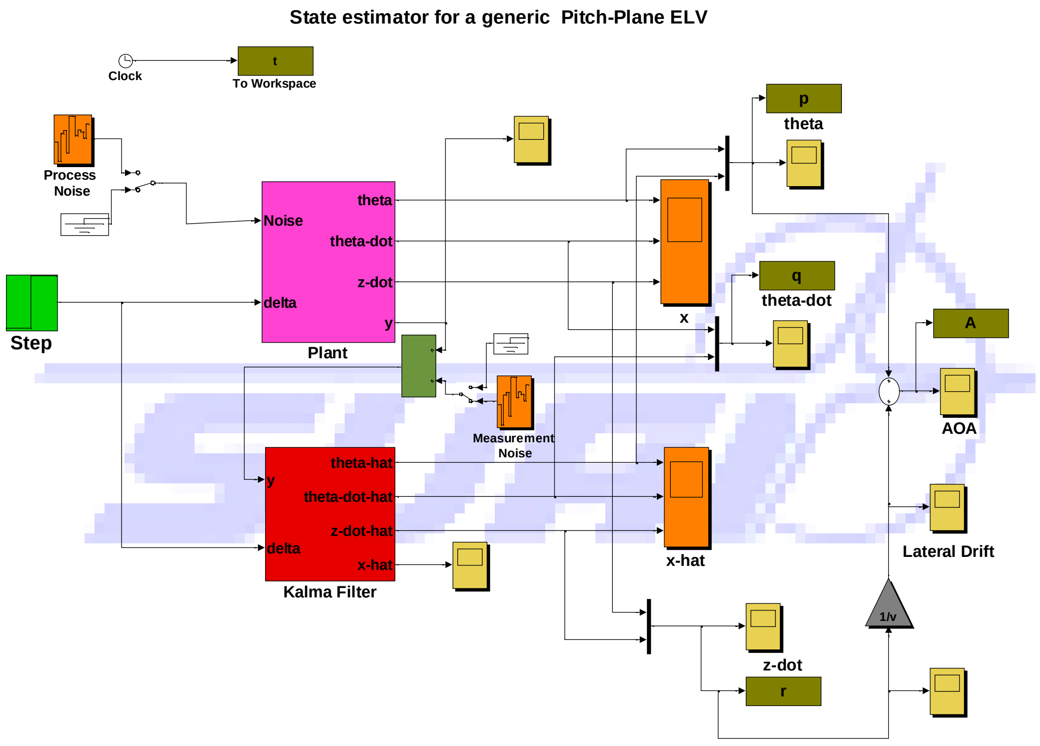


Figure 4.10: Simulink model of the KF based observer

The results of the synthesized KF based observer are presented below in Fig 4.11; depicting the plot of all state vectors and angle-of-attack. It could be seen that Kalman filter gives a very good estimate of these variables as expected. It is also necessary to note that the initial observation used for this KF estimation of the state variables is

$$\hat{x}_0 = [0 \quad 0 \quad 0]^T \tag{4.51}$$

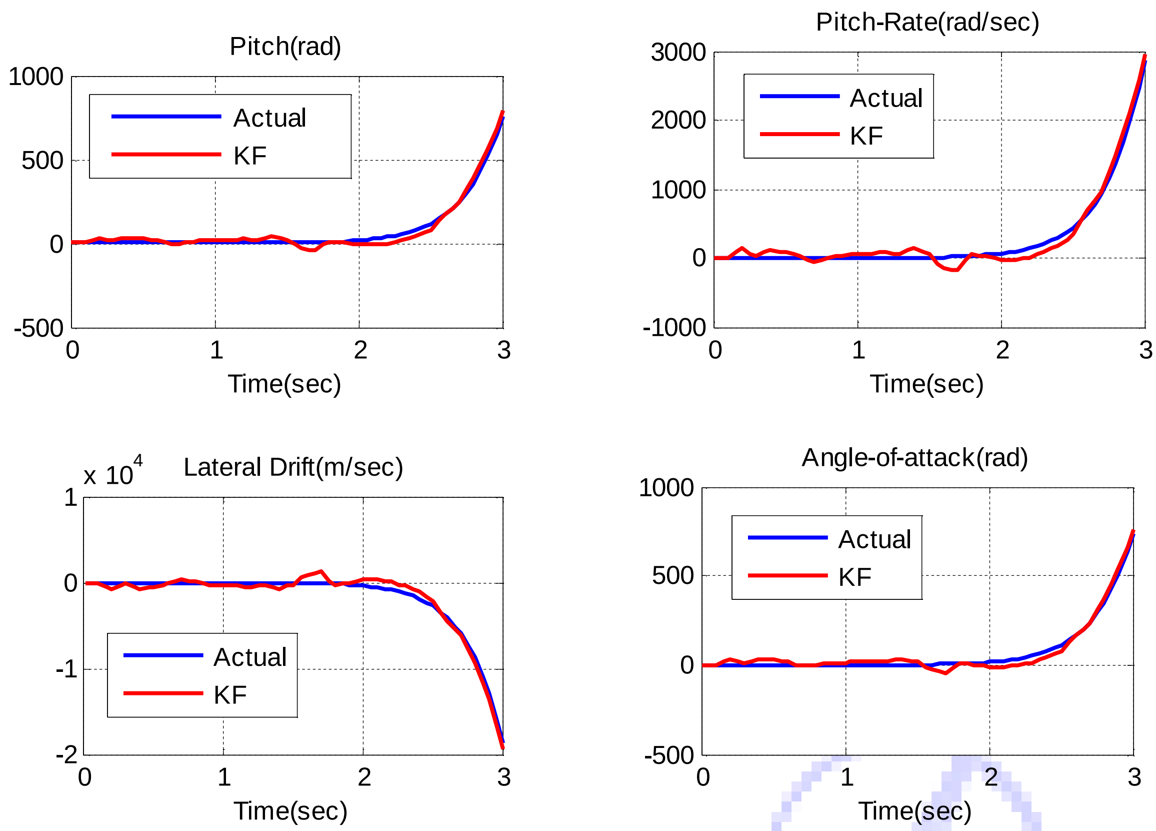


Figure 4.11: Step response of the Kalman filter based estimator .

4.4 Matlab/Simulink synthesis of LQG autopilot

The synthesized LQG autopilot for our hypothetical ELV in Simulink presented in Fig 4.12 is based on the *separation principle*, by which control system is designed in two independent phases, which are

- Design a full state feedback for the subsystem $(A-BK)$ and
- Design an observer for the subsystem $(A-LC)$

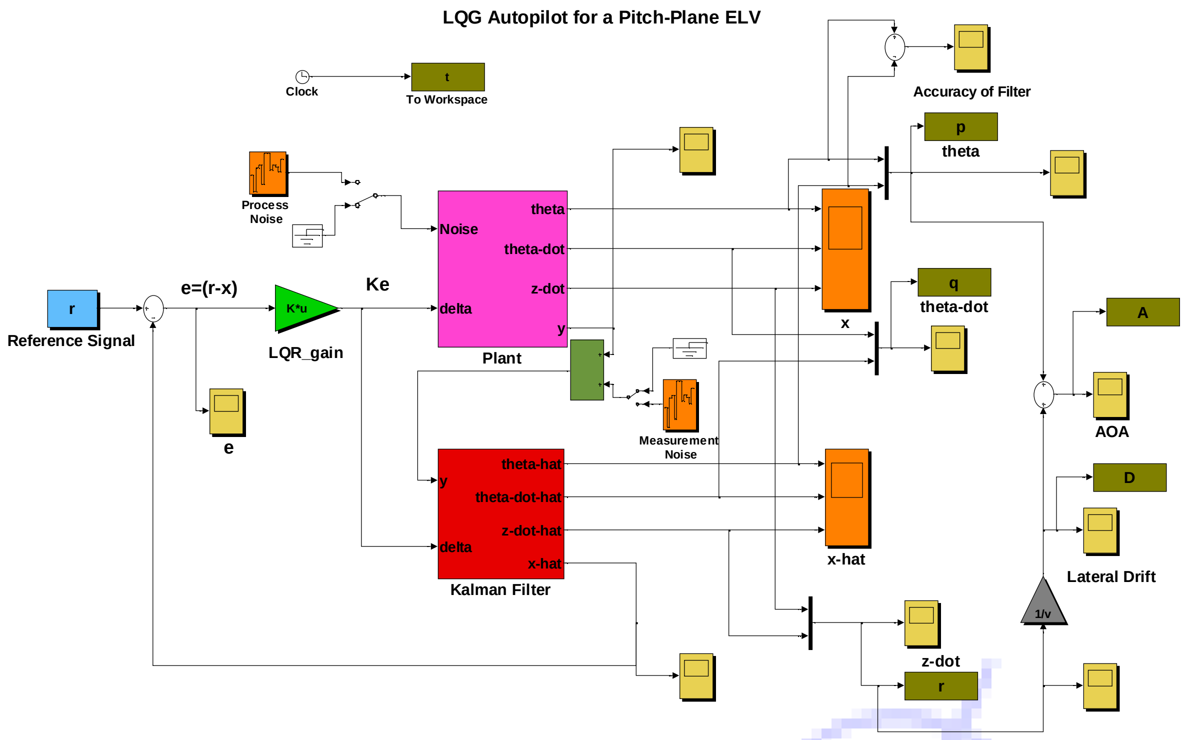
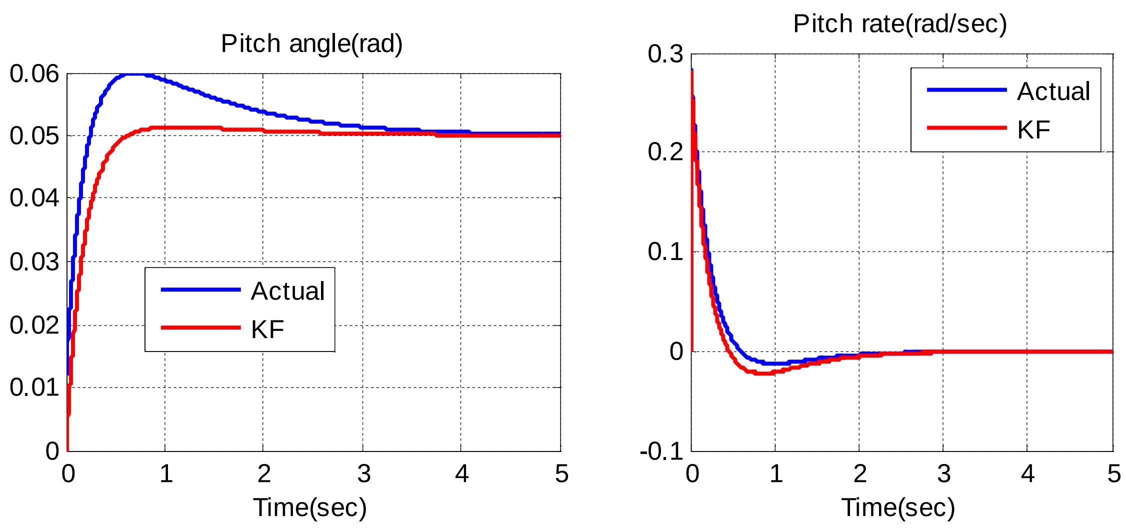


Figure 4.12: Simulink model of the LQG autopilot

This is to design two separate subsystems and then put them together such that the poles (eigenvalues) of the close-loop system comprise the poles of the observer and the poles that would be present if full state feedback were implemented. In other words, the stability of the resulting closed-loop system will be guaranteed by designing stable state feedback and observer dynamics. However, this separation property holds only when the model of the plant used in implementing the observer is a faithful model of the physical plant [23].
The results of the synthesized LQG autopilot in Fig 4.12 are presented in Fig 4.13, with the angle-of-attack during the stabilization of ELV.



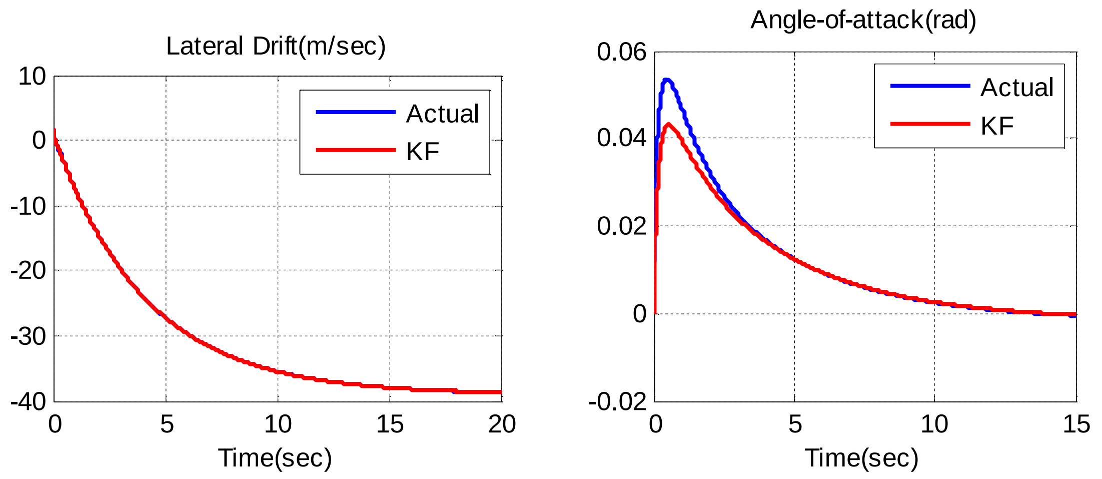
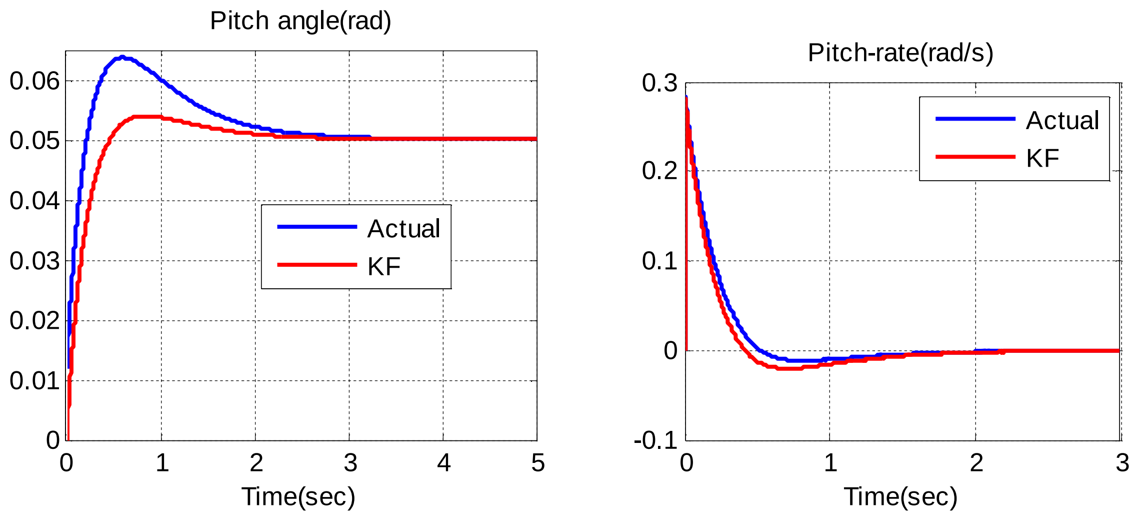


Figure 4.13: Plots of LQG autopilot set point command tracking.

It is interesting to note the time response characteristics of the pitch angle only, from Fig 4.13 which is the main objective of the synthesized autopilot is as expected and the lateral drift has a negative sign meaning that the vehicle drifts in the direction of the wind. The magnitude of the sensor noise was changed to

$$R_N = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}. \qquad (4.52)$$

With (4.52) in perspective, then a re-simulation of the LQG autopilot was carried out and the following results were realised;



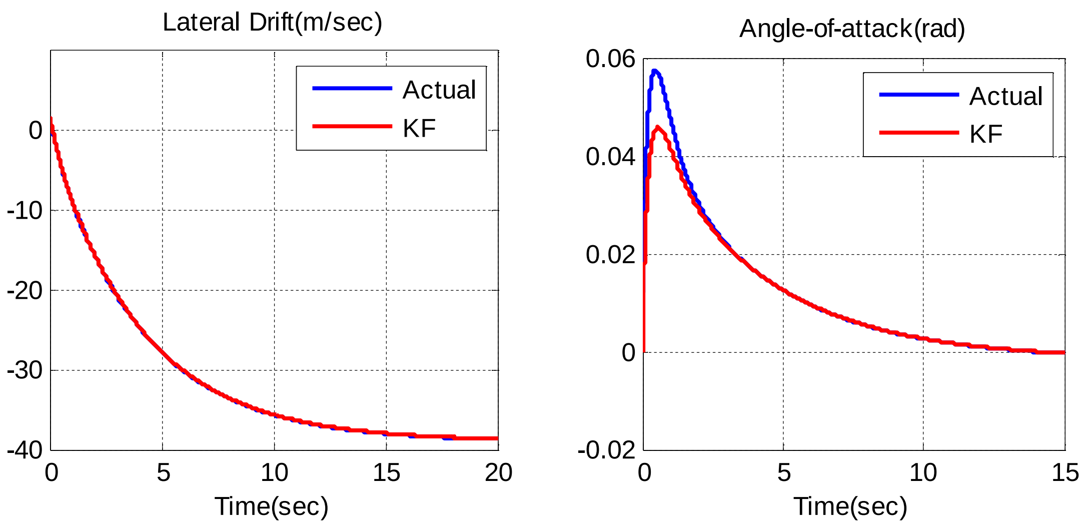


Figure 4.14: Plots of LQG autopilot set point command tracking with increased sensor noise.

The first question to ask about an LQG controller is whether or not the closed-loop system will be stable. To answer this question we collect all the equations that defines the closed-loop system:

$$\dot{x} = Ax + Bu, \qquad y = Cx, \tag{4.53}$$

$$\dot{\hat{x}} = (A - LC)\hat{x} + Bu + Ly, \qquad u = -K\hat{x}. \tag{4.54}$$

To check the stability of this system it is more convenient to consider the dynamics of the estimation error $e = x - \hat{x}$ instead of the state estimate $\hat{x}$. To this effect we replace in (4.53) and (4.54) $\hat{x}$ by $x - e,$ which yields:

$$\dot{x} = Ax + Bu = (A - BK)x + BKe, \qquad y = Cx, \tag{4.55}$$

$$\dot{e} = (A - LC)e, \qquad u = -K(x - e). \tag{4.56}$$

This can be written in matrix notation as

$$\begin{bmatrix} \dot{x} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} A - BK & BK \\ 0 & A - LC \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix}, \qquad y = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix}. \tag{4.57}$$

Thus, for this work, (4.57) can be expressed as

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \ddot{z} \\ \dot{\hat{\theta}} \\ \ddot{\hat{\theta}} \\ \ddot{\hat{z}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -9398 & -1642 & 133 & 9413 & 1642 & -133 \\ -55240 & -9620 & 780 & 55139 & 9621 & -780 \\ 0 & 0 & 0 & -1.1614 & -0.26 & 0 \\ 0 & 0 & 0 & 13.5205 & -59.8196 & 0.0196 \\ 0 & 0 & 0 & -88.3072 & 384.9599 & -0.1256 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ \dot{z} \\ \hat{\theta} \\ \dot{\hat{\theta}} \\ \dot{\hat{z}} \end{bmatrix},$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ \dot{z} \\ \hat{\theta} \\ \dot{\hat{\theta}} \\ \dot{\hat{z}} \end{bmatrix}. \tag{4.58}$$

The eigenvalues associated with (4.58) are

$$\begin{array}{l} -852.5238 \\ -6.5041 \\ -2.9721 \\ -59.8859 \\ -1.2214 \\ +0.0006 \end{array} \tag{4.59}$$

Separation Principle: the eigenvalues of the closed-loop system (4.53) are given by those of the state-feedback regulator dynamics $A-BK$ together with those of the state-estimator dynamics $A-LC$. In case these both matrices are asymptotically stable, then so is the closed-loop (4.53) [24].
But from the six eigenvalues in (4.59) it could be clearly seen that the last eigenvalue is positive, this is not a problem because the last three eigenvalues are for the state-estimator dynamics $A-LC$, and it being stable or unstable- its status does not make the closed-loop system unstable. Most important is the closed-loop of the process $A-BK$ and the effect of the state-estimator on it. The simulated results in Fig. 4.13 and 4.14 and (4.59) prove the fact of this claim; the designed state-estimator does asymptotically stabilise the closed-loop process.

Chapter 5

Proportional, Integral, and Derivative Control

5.1 Introduction

Proportional, Integral and Derivative controllers fall in the class of *classical* controller design. Even though control theory has been developed significantly, the proportional-integral-derivative (PID) controllers are used for a wide range of process control, motor drives, magnetic and optic memories, automotive, flight control, instrumentation, etc. The PID controller was first placed on the market in 1939 and has remained the most widely used controller in process control until today. An investigation performed in 1989 in Japan indicated that more than 90% of the controllers used in process industries are PID controllers and advanced versions of the PID controller. "PID control" is the method of feedback control that uses the PID controller as the main tool. The basic structure of a conventional PID feedback control systems is shown in Figure 5.1, using a block diagram representation. In this figure, the process or plant is the object to be controlled. The purpose of control is to make the process variable *y* follow the set-point value *r*. To achieve this purpose, the manipulated variable *u* is changed at the command of the controller. The error *e* is defined by *e* = *r* – *y*. The compensator *C(s)* is the computational rule that determines the manipulated variable *u* based on its input data, which is the error *e* in the case of Figure 5.1.

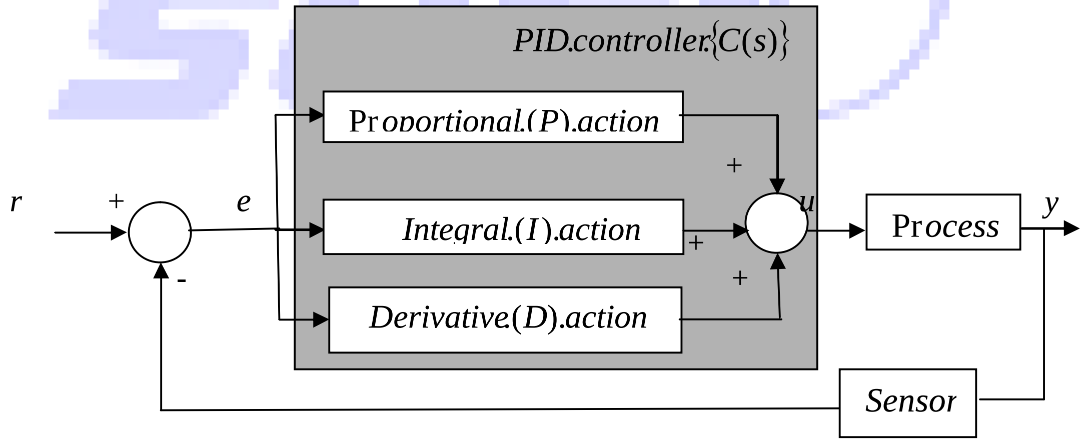


Figure 5.1: Block diagram of a PID controller in a closed-loop system.

Early PID control system had exactly the structure shown in figure 5.1 where the PID controller is used as the compensator $C(s)$. When used in this way the three elements of the PID controller produce outputs with the following nature;

- P element: proportional to the error at the instant t, which is the 'present' error.

- I element: proportional to the integral of the error up to the instant t, which can be interpreted as the accumulation of the 'past' error.

- D element: proportional to the derivative of the error at the instant t, which can be interpreted as the prediction of the 'future' error.

Thus, the PID controller can be understood as a controller that takes the present, the past, and the future of the error into consideration [25].

5.2 Transfer function of PID Controller

This *three-term* controller has a transfer function of the form

$$G_c(s) = K_P + \frac{K_I}{s} + K_D s. \tag{5.1}$$

The controller provides a proportional term, an integral term, and a derivative term. The equation for the output in the time domain is

$$u(t) = K_P e(t) + K_I \int e(t)dt + K_D \frac{de(t)}{dt}. \tag{5.2}$$

The transfer function of the derivative term is actually

$$G_d(s) = \frac{K_D s}{\tau_d s + 1}, \tag{5.3}$$

but usually τ_d is much smaller than the time constants of the process itself and so may be neglected. If we set $K_D = 0$, in (5.1) then we have the *proportional* plus *integral (PI)controller*

$$G_c(s) = K_p + \frac{K_I}{s}. \tag{5.4}$$

Considering (5.1),when $K_I = 0$, we have

$$G_c(s) = K_p + K_D s, \tag{5.5}$$

which is called a *proportional* plus *derivative (PD) controller*. The popularity of PID controllers can be attributed partly to their good performance in a wide range of operating conditions and partly to their functional simplicity, which allows engineers to operate them in simple, straightforward manner. To implement such a controller, three parameters must be determined for the given process: proportional gain, integral gain, and derivative gain [26].

5.2.1 Characteristics of a *P*, *I*, and *D* controller

A proportional controller (K_p) will have the effect of reducing the *rise time* and will reduce, but never eliminate the *steady-state error*. An integral control (K_I) will have the effect of eliminating the steady-state error, but it may make the transient response worse. A derivative control (K_D) will have the effect of increasing the stability of the system, reducing the overshoot, and improving the transient response. Effects of each of the controllers K_p, K_D, and K_I on a closed-loop system are summarised in Table 5.1 below.

Table 5.1: Closed-loop effects of K_p, K_D, and K_I Controllers

Controller(s)	Rise Time	Over Shoot	Settling Time	Steady-State Error
K_p	Decrease	Increase	Small change	Decrease
K_I	Decrease	Increase	Increase	Eliminate
K_D	Small Change	Decrease	Decrease	Small change

Note that these correlations may not be exactly accurate, because K_p, K_I, and K_D are dependent of each other. In fact, changing one of these variables can change the effect of the other two. For this reason, Table 5.1 should only be used as a reference when determining the values for K_p, K_I, and K_D.

5.2.2 Action modes of PID controller

In application, engineers have freedom of using the three functional elements (P, I and D) of the PID controller in whatever combination they consider most appropriate for their problems. The combination of element(s) used is called the 'action mode' of the PID controller. Theoretically, there exist seven action modes. Among them, the five listed in Table 5.2 are important in practice.

Table 5.2: PID action modes

Action mode	Element(s) used	Transfer function
Proportional (P)	P element only	K_p
Integral (I) with $T_i = 1$	I element only	$\frac{K_p}{T_i s}$
Proportional-Integral (PI)	P and I elements	$K_P\left\{1+\frac{1}{T_i s}\right\}$
Proportional-Derivative (PD)	P and D elements	$K_p\{1+T_d D(s)\}$
Proportional-Integral-Derivative (PID)	All 3 elements	$K_P\left\{1+\frac{1}{T_i s}+T_d D(s)\right\}$

Note that, $K_p = K_p$, $K_I = K_p / T_i$, and $K_D = K_p T_d$.

5.3 Model of ELV in transfer function

The rigid model for a Launcher is well known and is described in details in (Greensite A.L., 1970). Considering the ELV in section 2.2 with all assumptions still valid for basic analysis and perturbation, force and moment equation can be written as;

$$\Delta F_z = T\delta - L_\alpha \alpha - Mg\cos\theta_0, \tag{5.6}$$

$$\Delta M_y = Tl_c\delta + L_\alpha l_\alpha \alpha, \tag{5.7}$$

and the combined equations as:

$$M(\dot{w}_{vech} - \dot{\theta}U_0) = T\delta - L_\alpha\alpha - Mg\theta\cos\theta_0, \tag{5.8}$$

$$I_{yy}\ddot{\theta} = Tl_c\delta + L_\alpha l_\alpha \alpha. \tag{5.9}$$

Realising that

$$\dot{w}_{veh} = \dot{\alpha}U_0, \tag{5.10}$$

the equations transform into

$$(\dot{\alpha} - \dot{\theta}) = \frac{T\delta - L_\alpha\alpha - Mg\theta\cos\theta_0}{MU_0}, \tag{5.11}$$

$$I_{yy}\ddot{\theta} = Tl_c\delta + L_\alpha l_\alpha \alpha. \tag{5.12}$$

Simultaneously solving (5.11) and (5.12) to eliminate angle-of-attack and its derivative, we get

$$\ddot{\theta} + \frac{L_\alpha}{MU_0}\ddot{\theta} - \mu_\alpha\dot{\theta} + \frac{\mu_\alpha g\cos\theta_0}{U_0}\theta = \mu_c\dot{\delta} + \frac{L_\alpha\mu_c}{MU_0}\left\{1 + \frac{l_\alpha}{l_c}\right\}\delta, \tag{5.13}$$

which after taking Laplace transform with zero initial state takes the form as:

$$\frac{\theta(s)}{\delta(s)} = \frac{\mu_c\left[s + \frac{L_\alpha}{MU_0}\left(1 + \frac{l_\alpha}{l_c}\right)\right]}{\left[s^3 + \frac{L_\alpha}{MU_0}s^2 - \mu_\alpha s + \frac{\mu_\alpha g\cos\theta_0}{U_0}\right]}. \tag{5.14}$$

Taking the assumption that the perturbation flight path angle is zero (perturbation angle-of-attack becomes equal to the perturbation pitch attitude angle), we get:

$$\frac{\theta(s)}{\delta(s)} = \frac{\mu_C}{s^2 - \mu_\alpha}. \tag{5.15}$$

From Table 2.1, with respect to our design selection point and considering the frozen time or 'Time slice' approach, as earlier mentioned in chapter 1, the following parameters are associated our hypothetical ELV, $\mu_C = 3.4858\, m/s^2$ and $\mu_\alpha = 14.7805\, \text{deg}/s^2$. Hence, for (5.15), it is imperative to investigate it behaviour to a step signal before any form of design is considered or undertaken, as such Figure 5.2 unfolds the nature of the system we are dealing with; unstable.

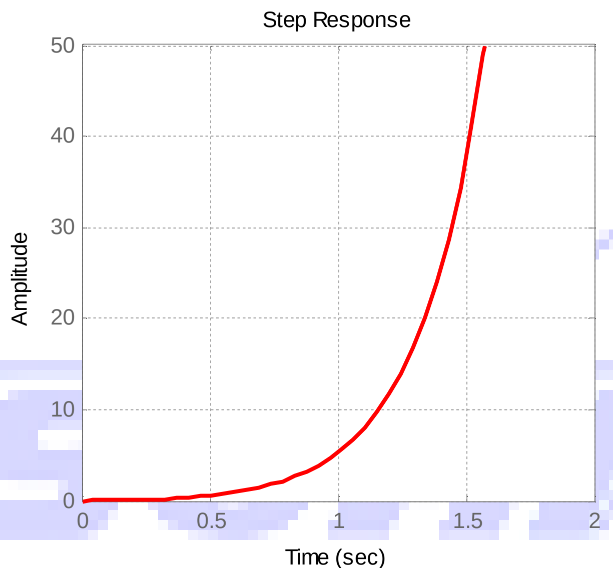


Figure 5.2: Plot of open-loop step response of the ELV.

5.4 Classical Design in Frequency Domain

Control system design in the frequency domain can be undertaken using a purely theoretical approach, or alternatively, using measurements taken from the components in the control loop. The technique allows transfer functions of both the system elements and complete system to be estimated, and suitable controller/compensator to be designed. The frequency response method may be less intuitive than other methods however it has certain advantages, especially in real-life situations such as modelling transfer functions from physical data. The frequency response is a representation of the system's response to sinusoidal input at varying frequencies. The output of a linear system to a sinusoidal input is a sinusoid of the same frequency but with a different magnitude and phase. The frequency response is defined as the magnitude and phase difference between the input and output sinusoids. This method of design and analysis uses the open-loop frequency response to predict its behaviour in closed-loop. The frequency response of a system can be viewed in two different ways: via the **Bode** plot or via the Nyquist diagram. Both methods display the same information; the difference

lies in the way the information is presented. A Bode diagram consists of two graphs; a magnitude plot and a phase plot. The first is a plot of logarithmic magnitude of the transfer function in dBs against frequency (which is also plotted logarithmically). The phase plot is also a plot against frequency, this time representing the phase change introduced by the system to a sinusoidal signal. The standard representation of logarithmic magnitude of *G(j*ω*)* is 20 log |*G(j*ω*)*|, in base 10. The curves obtained are drawn on semilog paper, using the log scale for frequency and the linear scale for either magnitude (in dBs) or phase (in degrees).
In the design of linear control systems using frequency-domain methods, it is necessary to define a set of specifications so that the performance of the system can be identified. Specifications such as maximum overshoot, damping ratio, and so, used in the time-domain, can no longer be used directly in the frequency domain. The following frequency-domain specifications are often used in practice.

5.4.1 Gain and Phase Margin

Making the ELV stable is one thing. But giving it a satisfactory behaviour and being able to control it is another story. In this section, we're going to look at some parameters which a system can have. The *gain margin* is defined as the open-loop gain required to make the system unstable. Systems with greater *gain margins* can withstand greater changes in the system parameters before becoming unstable in the closed-loop.
The *phase margin* is defined as the change in open-loop phase shift required to make a closed-loop system unstable. The phase margin also measures the system's tolerance to time delay. If there is a time delay greater than π/ω_ϕ in the loop (where ω_ϕ is the frequency where the phase shift is -180^0), the system will become unstable in closed-loop. The time delay can be thought of as an extra block in the forward path of the block diagram that adds phase to the system but has no effect on the gain. The *phase margin* is the difference in phase between the phase curve and -180^0 at the point corresponding to the frequency that gives us a gain of $0\,db$ (the gain cross over frequency, ω_K). Likewise, the *gain margin* is the difference between the magnitude curve and $0\,db$ at the point corresponding to the frequency that gives us a phase of -180^0 the phase crossover frequency, ω_ϕ).
From a mathematical standpoint, let's consider the system with transfer function in (5.16), this function is in frequency domain,

$$F(s) = \frac{G(s)}{1+G(s)H(s)}, \tag{5.16}$$

thus, substituting s by $j\omega$. If the term $G(j\omega)H(j\omega)$ ever becomes -1, then the system becomes unstable. We are thus interested in the point where $|G(j\omega)H(j\omega)=1|$ and $\arg(G(j\omega)H(j\omega)) = -180^0$. The frequency at which $\phi = \arg(G(j\omega)H(j\omega)) = -180^0$, is called the phase crossover frequency $\omega_\phi = -180^0$. Similarly, the frequency at which $K = |G(j\omega)H(j\omega)| = 1$, (or $K_{db} = 0$) is called the gain crossover frequency $\omega_K = 1$.

Traditionally, requirements placed on controller performance laws are based on system frequency response characteristics. The basic rigid body frequency domain requirements are:

- Gain margin (GM) > 6 db

- Phase margin (PM) $30^0 <$ PM $< 60^0$

The phase and gain margins can, however, be misleading. It may happen that the phase and gain margin appear safe, but there still is a value of ω for which the open-loop transfer function comes close to -1. Therefore, instead of looking at phase and gain margins, it is often wise to simply look at the Nyquist plot of the open-loop transfer function and see if it comes close to -1.

5.4.2 Stability Analysis with Bode plots

Bode plots are extensively used to obtain a measure of the systems stability. Based on the definitions above the following conclusion can be drawn about the stability of closed-loop systems based on Bode plots of the open-loop transfer function.

- The gain margin is positive and the system is stable if the magnitude of the open-loop transfer function at phase cross over is negative in dB. That is, the gain margin is measured below the 0-dB line. If the gain margin is measured above the 0 dB line, the gain margin is negative and the system unstable.

- The phase margin is positive and the system is stable if the phase of the open-loop transfer function is greater than -180^0 at gain crossover. That is, the phase margin is measured above the -180^0 line. If the phase margin is measure below the -180^0 line, the phase margin is negative, and the system is unstable.

5.4.3 Other frequency domain parameters

There are more parameters that are related to the frequency domain. Most of these parameters can easily be derived from the Bode plot. In a Bode diagram, there exists a frequency region in which the system performs satisfactory. This region is usually a region with a more or less constant gain K_0. The point(s) where the gain drops below 3 dB less than this constant gain K_0 is called the *cut-off frequency*. The slope of the Bode plot at this point is called the *cut-off rate*. Also, the frequency range in which the system performs satisfactory (being the frequency range between the cut-off frequencies) is called the *bandwidth* ω_b. In a Bode diagram, you can often find a peak at which the gain K is at a maximum. This phenomenon is called *resonance*. The corresponding maximum value of the gain K is denoted by the *resonance peak* M_p. The frequency at which this resonance occurs is called the resonance frequency ω_p. The last important parameter for the frequency domain is the delay time. The delay time $t_d(\omega)$ for a given frequency ω is given by

$$t_d(\omega) = -\frac{d\phi}{d\omega} = -\frac{d\arg(G(j\omega))}{d\omega}. \qquad (5.1)$$

5.5 Matlab/Simulink synthesis of the classical autopilot

It can shown that for (5.15) and considering the values of μ_C and μ_α as earlier stated, the roots of the characteristic equation are ± 3.8445, hence the ELV is unstable even as depicted in Fig 5.2. This unstable nature can be further affirmed from the open-loop Bode plot in Fig. 5.3.

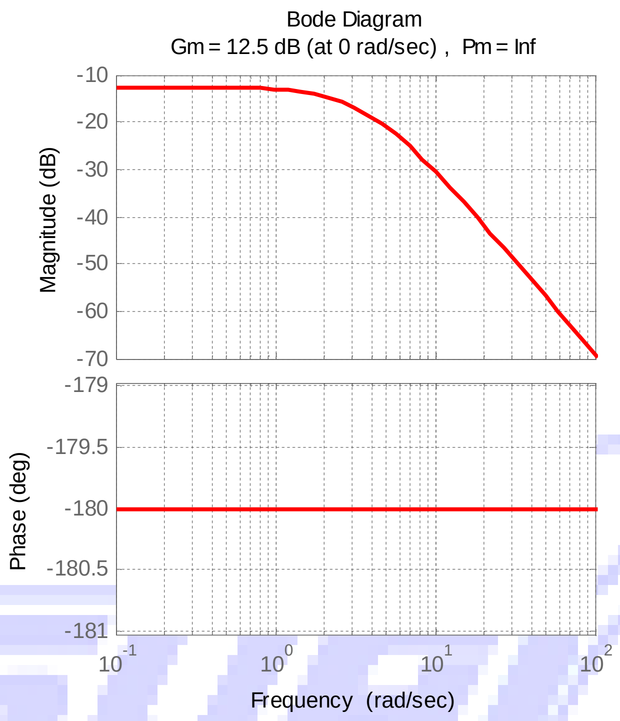


Figure 5.3: Bode plot of open-loop transfer function of the ELV.

We therefore seek to design a classical controller which is simple and meets the time-domain design requirement as given below.

- Less than 2% steady-state error.
- Maximum overshoot must be more than 20 per cent.
- Settling time must be less than 3 seconds.
- Rise time must be less than 1 second.

5.5.1 The PD autopilot

A P-controller will be investigated firstly and from, Fig 5.2 it could be inferred that it will not stabilize the system; in the sense that its open-loop transfer function has just a gain added and to the unstable plant adding a gain only shifts the magnitude plot up which is equivalent of changing the $y-axis$ on the magnitude plot. Hence designing a Proportional controller will not stabilize the ELV. By itself, a proportional controller can raise or lower the system gain, which shifts the Bode magnitude up or down. In a PI controller, the integral mode (often referred to as reset) corrects for any offset (error) that may occur between the desired value (set-point) and the process output automatically over time hence. The transfer function of an integrator, which is a pole at the origin in the zero-pole plot is $1/s$. It is sometimes taken as with a gain K, i.e. K/s. Here K will be replaced by $1/T$ to give the transfer function as given in Table 5.2. On a Bode diagram the magnitude is a constant slope of -6 db/octave passing through 0 db at frequency $1/T$. The phase is -90^0 at all frequencies. Thus, a PI controller will not stabilize the ELV, but the fact is that the presence of an integral component will decrease the phase margin from it current state as shown in Fig. 5.3 further downwrds thus, increase its instability. Certainly a PD-controller will have a promising effect on the ELV. A combined PD controller introduces a zero into the system at the value $s = K_P/K_D$. Thus, PD control is a type of phase-lead compensation, which adds phase and gain to loop transfer function. A differentiator has a transfer function of sT which gives a gain characteristic with a slope of 6 db/octave passing through 0 db at a frequency of $1/T$. It has a phase of +90^0 at all frequencies [27].Considering the control schematic in Fig 5.4, where $G_c(s)$ is the PD controller transfer function as given in (5.18).

The design of the PD controller is to determine the two constant control gains, K_P and K_d, such that the system output can track the set-point r while the entire feedback control system is stable even if the given transfer function of the plant or process is unstable as in the case of our ELV.

$$G_{C-PD}(s) = K_P + K_D s, \tag{5.18}$$

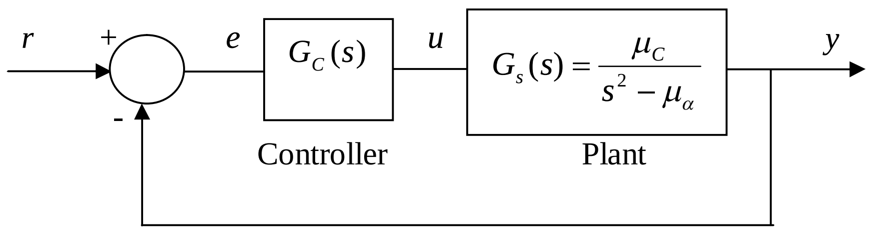


Figure 5.4: A typical closed-loop set-point tracking.

Here and throughout the chapter, we use capital letters for the Laplace transform $L\{.\}$ of a continuous-time signal. It follows from Fig. 5.4 that

$$U(s) = K_P E(s) + K_D s E(s), \tag{5.19}$$

$$E(s) = R(s) - Y(s), \tag{5.20}$$

$$Y(s) = \frac{\mu_C}{s^2 - \mu_\alpha} U(s), \tag{5.21}$$

So that

$$Y(s) = H_{PD}(s)R(s) = \frac{\mu_C K_D s + \mu_C K_P}{s^2 + \mu_C K_D s + \mu_C K_P - \mu_\alpha} R(s). \tag{5.22}$$

The transfer function $H_{PD}(s)$ of the overall feedback control system has two poles:

$$s_{1,2} = \frac{-\mu_C K_D \pm \sqrt{(\mu_C K_D)^2 - 4a(\mu_C K_P - \mu_\alpha)}}{2}, \tag{5.23}$$

and so the selection

$$K_D > 0 \tag{5.24}$$

and

$$K_P = \frac{(\mu_C K_D)^2}{4a} + \mu_\alpha, \tag{5.25}$$

can guarantee the controlled system be stable and have no oscillation on the output trajectory during the set-point tracking process. However, the asymptotic tracking error for this PD controlled system is

$$\begin{aligned} \lim_{t\to\infty} e(t) &= \lim_{|s|\to 0} sE(s) \\ &= \lim_{|s|\to 0} s[I - H_{PD}(s)]R(s) \\ &= \lim_{|s|\to 0} s.\frac{s^2 - \mu_\alpha}{s^2 + \mu_C K_D s + (\mu_C K_P - \mu_\alpha)}.\frac{r}{s} \\ &= \frac{-\mu_\alpha r}{\mu_C K_P - \mu_\alpha}, \end{aligned} \tag{5.26}$$

which will not be zero if $\mu_\alpha \neq 0$ and $r \neq 0$. This implies that the PD controller generally cannot eliminate the steady-state error in the set-point tracking. Thus, tuning for the PD parameters using *signal constraint* block [28] in a Simulink Model of the purported PD-controller the following values were realized $K_P = 94.2988$, and $K_D = 5.691$.

$$G_{PD}(s) = (K_P + K_D s)\left\{\frac{\mu_C}{s^2 - \mu_\alpha}\right\}, \tag{5.27}$$

and will yield

$$G_{PD}(s) = \frac{\mu_C(K_D s + K_P)}{s^2 - \mu_\alpha}. \tag{5.28}$$

Thus, from (5.28) it can be shown that the open-loop Bode plot of the PD controller will be like Fig. 5.5

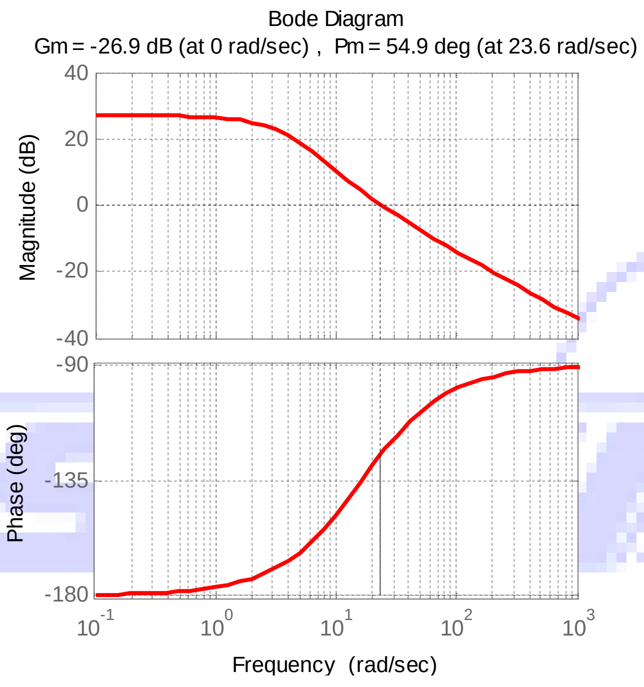


Figure 5.5: Bode plot of open-loop PD transfer function.

Though, the closed-loop system if a PD-controller is design for our hypothetical ELV is predicted to be stable by Fig. 5.5. It is also imperative to see how the step response of a purported PD-controller for our ELV will behave with the above selected parameters.

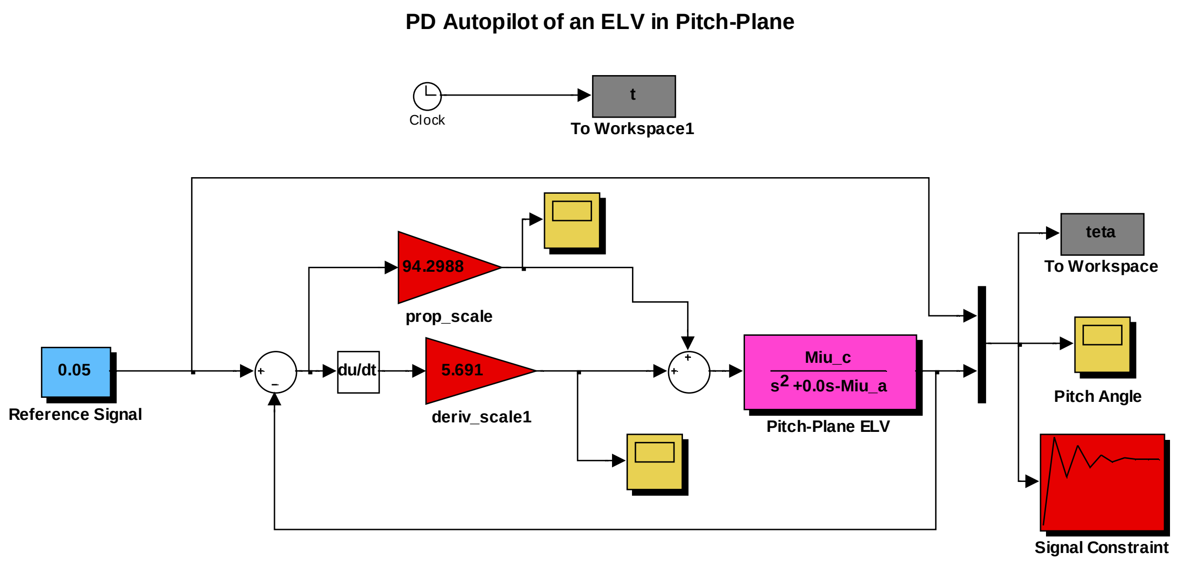


Figure 5.6: Simulink model of the PD autopilot.

Fig. 5.5 depicts the Simulink model for the PD controlled ELV while Fig. 5.6 gives step response of the closed-loop PD-controller and it can be seen that a sustained steady-state error plagues the system.

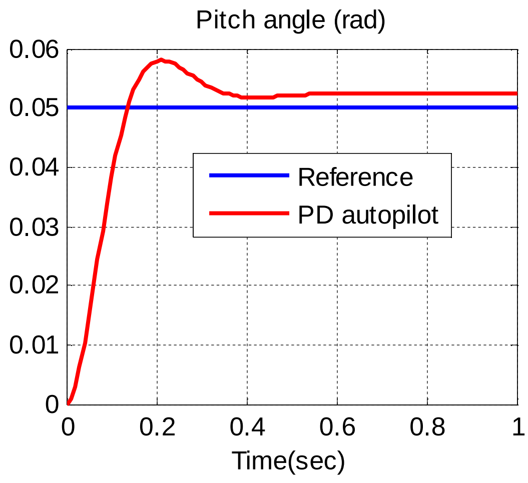


Figure 5.7: plot of set point command tracking of PD autopilot.

With the designed parameters of the PD autopilot, its closed-loop transfer function will have the following *poles* $-9.92 \pm 14.6797i$, and *zero* -16.5675. Since the synthesized PD controller characteristic in Fig. 5.7 have the following unsatisfactory performance characteristics of 16% *over shoot* and 0.0024 *steady-state* error, we felt obliged to seek a PID controller that might perhaps satisfy at least three out of four of the specified requirement.

5.5.2 The PID autopilot

Hence, without doubt an integral component needs to be added to the PD controller to eliminate the steady state error. If an integral component is to be added to the PD-controller above, then our PID-controller having a transfer function as in (5.29) will yield an open-loop transfer function as described in (5.30) considering Fig. 5.4.

$$G_{C-PID}(s) = \frac{K_D s^2 + K_P s + K_I}{s}. \tag{5.29}$$

$$G_{PID}(s) = \left\{\frac{K_D s^2 + K_P s + K_I}{s}\right\}\left\{\frac{\mu_C}{s^2 - \mu_\alpha}\right\}, \tag{5.30}$$

hence,

$$G_{PID}(s) = \frac{\mu_C(K_D s^2 + K_P s + K_I)}{s^3 - \mu_\alpha s}. \tag{5.31}$$

Hence we can write the following for the closed-loop transfer function for the PID controller

$$H_{PID}(s) = \frac{\mu_C(K_D s^2 + K_P s + K_I)}{s^3 + \mu_C K_D s^2 + (\mu_C K_P - \mu_\alpha)s + \mu_C K_I}. \tag{5.32}$$

Also, with the aid of Matlab/Simulink the *signal constraint* block, the following PID parameters were obtained $K_P = 111.5$, $K_I = 52.2$ and $K_D = 19.8$ with theses parameters substituted in (5.31), we were able to obtain Fig. 5.8.

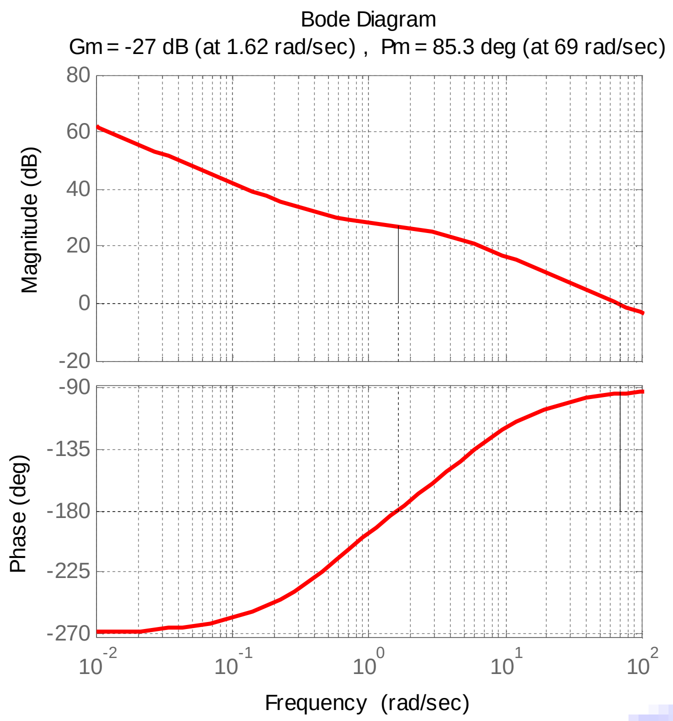


Figure 5.8: Bode plot of open-loop PID transfer function.

Although the *gain margin* test for stability (GM>0) is not reliable, the test for *phase margin* (PM>0) does apply to all systems and should be considered the best test for system stability when designing in the frequency domain although the relationships between gain and phase margins and system stability apply to some non-minimal-phase systems, they do not apply to all. It is best to regard the use of gain and phase margins to determine stability as applicable only to minimum phase systems. It is n not possible to say what the 'best' value for gain and phase margins to obtain optimum response, but in some published work a general rule of thumb is to attempt to obtain a gain margin of about 2 and a phase margin of between 45^0 and 60^0. These are obviously rough guidelines, but they do produce a response with adequate margins of stability [29]. The implementation of the PID-controller in Simulink for the investigation of its step response is shown in Fig. 5.11.

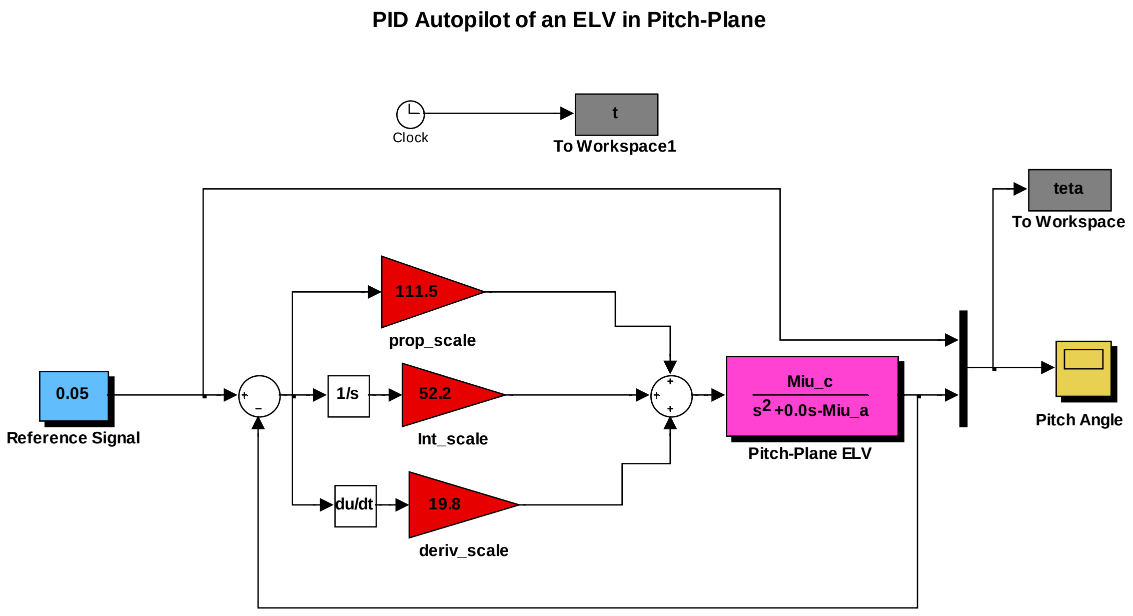


Figure 5.9: Simulink model of PID autopilot.

The result of the Simulation in Fig. 5.9 is shown in Fig.5.10, and could be seen that the *Integral* component did eliminate the steady-state error as expected.

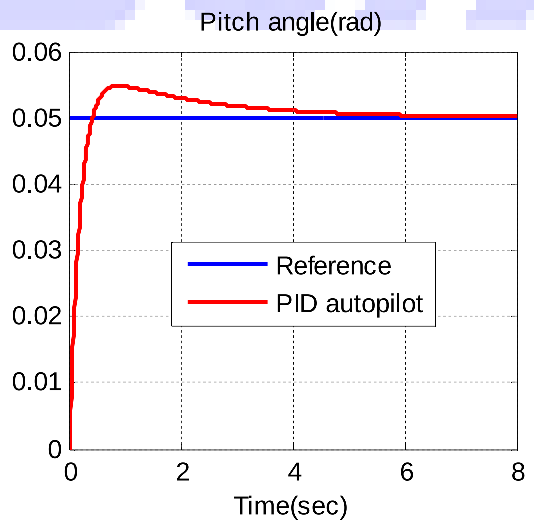


Figure 5.10: Plot of set point command tracking of PID autopilot.

The closed-loop transfer function of the PID-controller designed is given as in (5.32) and its can be shown that for the selected parameters of PID controller, the poles are -63.1443, -5.3355 and -0.5402, and the zeros are -5.1163, -0.5154. And its bandwidth frequency was obtained to be $74.25 rad/\sec$ using the Matlab in-built command; *FB=bandwidth (Sys, -3)*. See appendix for details.

5.5.3 Disturbance Rejection

Apart from improving the dynamic response beyond that available with an open-loop system, one of the other benefits of utilizing feedback control is to make the system less responsive to unwanted inputs, known as disturbances. The ability of a system to ignore unwanted inputs is often called *disturbance rejection* capability. The ability of a closed-loop system to reject disturbance better than open-loop system is quite general and is not restricted to systems of a particular order. The Band-Limited White Noise block generates normally distributed random numbers that are suitable for use in continuous or hybrid systems.

Theoretically, continuous white noise has a correlation time of 0, a flat power spectral density (PSD), and a total energy of infinity. In practice, physical systems are never disturbed by white noise, although white noise is a useful theoretical approximation when the noise disturbance has a correlation time that is very small relative to the natural bandwidth of the system. In Simulink software, you can simulate the effect of white noise by using a random sequence with a correlation time much smaller than the shortest time constant of the system. The Band-Limited White Noise block produces such a sequence. The correlation time of the noise is the sample rate of the block. For accurate simulations, we used a correlation time much smaller than the fastest dynamics of the system. To obtain a good result, we specify

$$t_C = \frac{1}{100}\frac{2\pi}{f_{\max}}, \tag{5.33}$$

where $f_{\max}$ is the bandwidth of the system in $rad/\sec$[30], and for our designed PID autopilot this value was obtained as $74.25 rad/\sec$. Hence

$$t_c = 0.001s. \tag{5.34}$$

Taking a noise power of 0.0326 spectral density, with (5.34), Fig. 5.12 shows the Simulink model for the PID control of the ELV with the modelled white noise. The result of the simulation is shown in Fig 5.13.

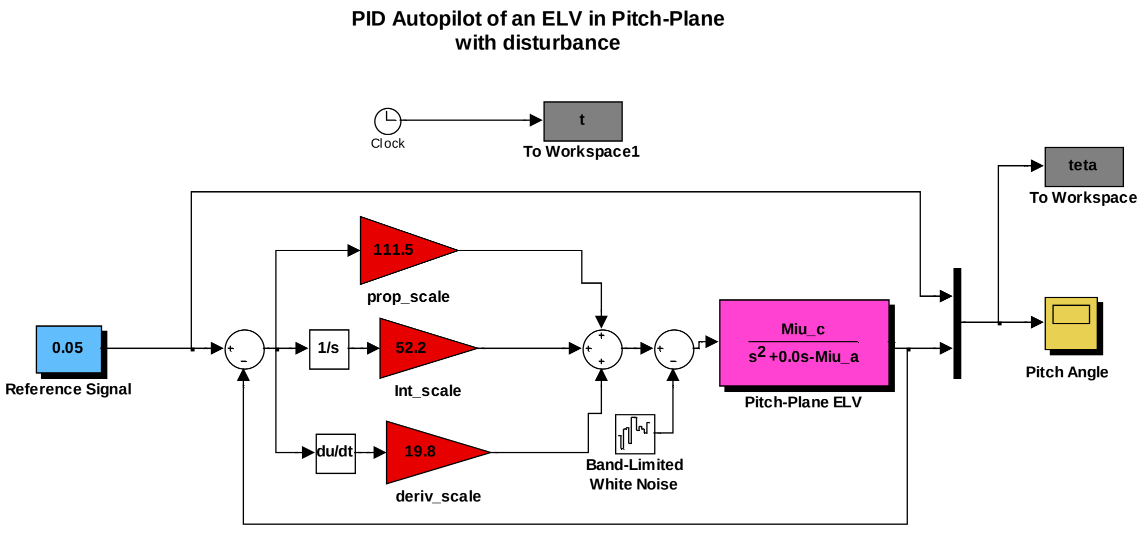


Figure 5.11: Simulink model of PID control of ELV with simulated disturbance.

The result of the simulation in Fig. 5.13 could be seen to have been as expected, that is, to the effect that the designed PID-controller is in fact capable of maintaining its stable state after being perturbed by the sustained noise.

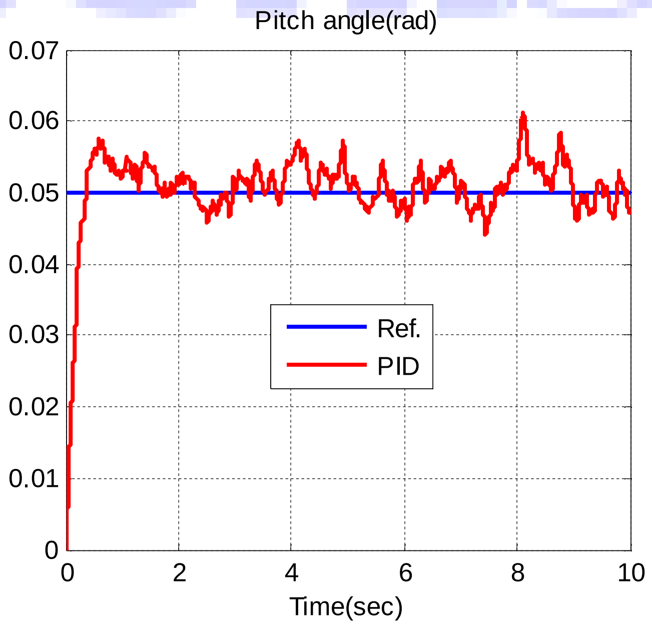


Figure 5.12: Set point tracking of PID autopilot with disturbance.

We investigated further with a noise power of 0.1 spectral density still retaining t_c as defined in (5.34). The result is depicted in Fig. 5.14

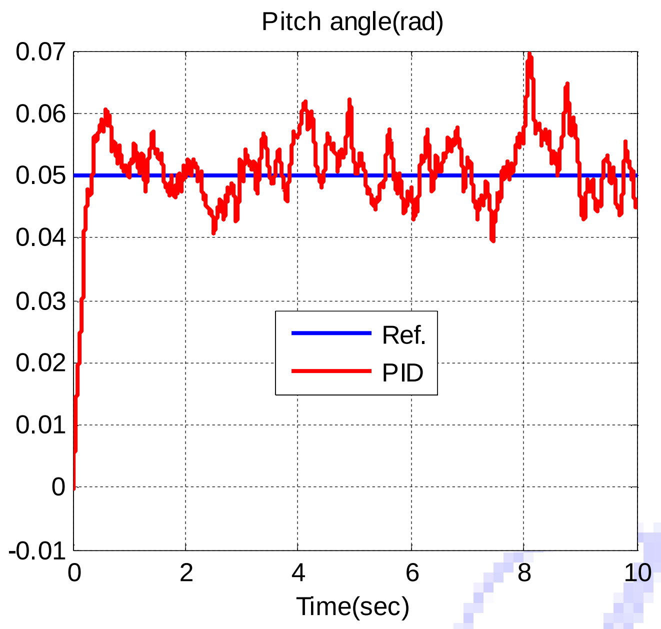


Figure 5.13: Set point command tracking plot of PID autopilot with increased disturbance.

Thus, from Fig. 5.14,it can be seen that the PID controller exhibits execution from the desired trajectory beyond 20% overshoot margin within 1s.

Chapter 6

Fuzzy Logic Control Systems

Everything is vague to a degree you do not realize till you have tried to make it precise. —Bertrand Russell

6.1 Introduction

Fuzzy logic controllers fall into the class of *Intelligent Control Systems*. According to the Oxford dictionary, the word intelligence is derived from intellect which is the faculty of knowing, reasoning and understanding. Intelligent behaviour is therefore the ability to reason, plan and learn, which in turn requires access to knowledge. Artificial Intelligence (AI) is a by-product of the Information Technology (IT) revolution, and is an attempt to replace human intelligence with machine intelligence. An intelligent control system combines the techniques from the fields of AI with those of control engineering to design autonomous systems that can sense, reason, plan, learn and act in an intelligent manner. Such a system should be able to achieve sustained desired behaviour under conditions of uncertainty, which include:

- uncertainty in plant models
- unpredictable environmental changes
- incomplete, inconsistent or unreliable sensor information
- actuator malfunction.

An intelligent control system, as considered by Johnson and Picton (1995), comprises of a number of subsystems; *the perception subsystem*-this collects information from the plant and the environment, and processes it into a form suitable for the cognition subsystem. *The cognition subsystem*- cognition in an intelligent control system is concerned with the decision making process under conditions of uncertainty. *The actuation subsystem*-operates using signals from the cognition subsystem in order to drive the plant to some desired state.

The Fuzzy logic tool was introduced in 1965, by Lofti zadeh, and is a mathematical tool for dealing with uncertainty. It offers to a soft computing partnership the important concept of computing with words. It provides a technique to deal with imprecision and information granularity. The fuzzy theory provides a mechanism for representing linguistic constructs such as ‘many’, ’low’, ‘medium’, ‘often’, ’few’. In general the fuzzy logic provides an inference structure that enables appropriate human reasoning capabilities. On the contrary, the traditional binary set theory describes crisp events, events that either do or do not occur. It uses probability theory to explain of an event will occur, measuring the chance with which a given event is expected to occur, the theory of fuzzy logic is based upon the notion of relative graded membership and so are the functions of mentation and cognitive processes. The utility of fuzzy set lies in their ability to model uncertain or ambiguous data, so often encountered in real life. It is important to observe that there is an intimate connection between fuzziness and complexity. As the complexity of a task (problem) or of a system for performing that task exceeds a certain threshold, the system must necessarily become fuzzy in nature. With the increasing of complexity of problems, our ability to make precise and yet significant statements about its behavior diminishes. Real world problems (situations) are too complex, and the complexity involves the degree of uncertainty-as uncertainty increases, so does the complexity of the problem. Traditional system modeling and analysis are too precise for such problems (systems), and in order to make the complexity less daunting we introduce appropriate simplifications, assumptions, etc. (i.e., degree of uncertainty or fuzziness) to achieve a satisfactory compromise between the information we have and the amount of uncertainty we are willing to accept. In this aspect, fuzzy system theory is similar to other engineering theories, because almost all of them characterize the real world in an approximate manner. Fuzzy logic is a paradigm for an alternative design methodology which can be applied in developing both linear and non-linear systems for embedded control. Fuzzy logic is a better solution to non-linear control because most real life systems are non-linear. Other approaches use different approaches to handle non-linearity. A linear approximation technique is relatively simple; however it tends to limit control performance and may be costly to implement in certain applications. Fuzzy logic provides a better alternative solution to non-linear control because it is closer to the real world. Non-linearity is handled by rules, membership functions, and the inference process which results in improved performance, simpler implementation, and reduced design cost [31].

6.1.1 Fuzzy set theory

Fuzzy logic is based on the concept of fuzzy sets. Fuzzy set theory provides a means for representing uncertainty. In general, probability theory is the primary tool for analyzing uncertainty, and assumes that the uncertainty is a random process. However, not all uncertainty is random, and fuzzy set theory is used to model the kind of uncertainty associated with imprecision, vagueness and lack of information.

Conventional set theory distinguishes between those elements that are members of a set and those that are not, there being very clear, or crisp boundaries. The central concept of fuzzy set theory is that the membership function μ, like the probability theory, can have a value of between 0 and 1, the membership function μ has a linear relationship with the x-axis, called the universe of discourse U. This produces a triangular shaped fuzzy set. Fuzzy sets represented by symmetrical triangles are commonly used because they give good results and computation is simple. Other arrangement includes non-symmetrical triangles, trapezoids, Gaussian and bell shaped curves.

6.1.2 Basic fuzzy set operations

Let A and B be two fuzzy sets within a universe of discourse with membership functions μ_A and μ_B respectively. The following fuzzy set operations can be defined as

Equality: Two fuzzy sets A and B are equal if they have the same membership function within a universe of discourse.

$$\mu_A(u) = \mu_B(u), \qquad \text{for all } u \in U. \tag{6.1}$$

Union: The union of two fuzzy sets A and B corresponds to the Boolean OR function and is given by

$$\mu_{A\cup B}(u) = \mu_{A+B}(u) = \max\{\mu_A(u), \mu_B(u)\}, \qquad \text{for all } u \in U. \tag{6.2}$$

Intersection: The intersection of two fuzzy sets A and B corresponds to the Boolean AND function and is given by

$$\mu_{A\cap B}(u) = \min\{\mu_A(u), \mu_B(u)\}, \quad \text{for all } u \in U. \tag{6.3}$$

Complement: The complement of fuzzy set A corresponds to the Boolean NOT function and is given by

$$\mu_{-A(u)} = 1 - \mu_A(u), \qquad \text{for all } u \in U. \tag{6.4}$$

An important aspect of fuzzy logic is the ability to relate sets with different universes of discourse. Consider the relationship

$$\text{IF } L \quad \text{THEN } M. \tag{6.5}$$

In equation (6.5) L is known as the *antecedent* and M as the *consequent*. The relationship is denoted by.

$$A = L \times M, \tag{6.6}$$

or

$$L \times M = \begin{bmatrix} \min\{\mu_L(u_1), \mu_M(v_1)\}\ldots\min\{\mu_L(u_1), \mu_M(v_k)\} \\ \min\{\mu_L(u_j), \mu_M(v_1)\}\ldots\min\{\mu_L(u_j), \mu_M(v_k)\} \end{bmatrix}, \tag{6.7}$$

where $u_1 \to u_j$ and $v_1 \to v_k$ are the discretized universe of discourse.

6.2 The physical system to be controlled

For a physical system operator, one of the most important knowledge is the time response of the system. An operator knows very well how the system responds for a change in reference input or system parameters in order to make necessary adjustment to keep the system going on under normal operating conditions. The time response of a physical system is important since it reflects the effects of the changes either in reference input or system's interior parameters. Therefore, the time response of the operating error and its derivation are usually used as two input parameters to the fuzzy logic controller. Since the error response includes the information about system output, it is used as a bridge connecting system's input to the output over a set of linguistic fuzzy rules.
A linear time invariant (LTI) system can be represented in different ways such as state-space model and transfer functions, which are mostly in first, second, or higher order. Depending on the order of model, the system output for a step input may vary as shown in Fig. 6.1, resulting in a similar response in error and error change as given in Fig. 6.2.

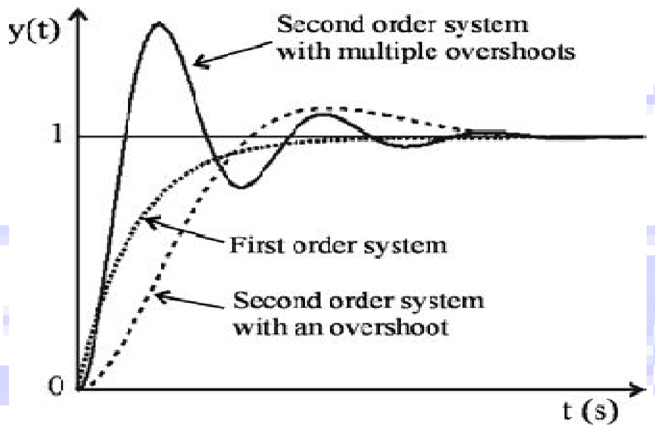


Figure 6.1: Step response of the outputs of first and second order systems

The time responses of error signals can be used to represent information related to the system output responses. As the error signals approach zero the output signals move towards reference. Depending on the performances of controllers used, the error signals may or may not become zero. The error signal of a controlled system will be sufficient to derive the controller rules since it contains the necessary information about the outputs. Therefore the error signals shown in Fig. 6.2 are used as the source of information for constructing the rule base systems for fuzzy logic controller. These signals may represent any types of control error with a step type reference input. In fact, the error signal of a control system with a ramp input will not be much different than that of given in Fig. 6.3 in spite of the differences between the outputs of the systems with step and ramp inputs.

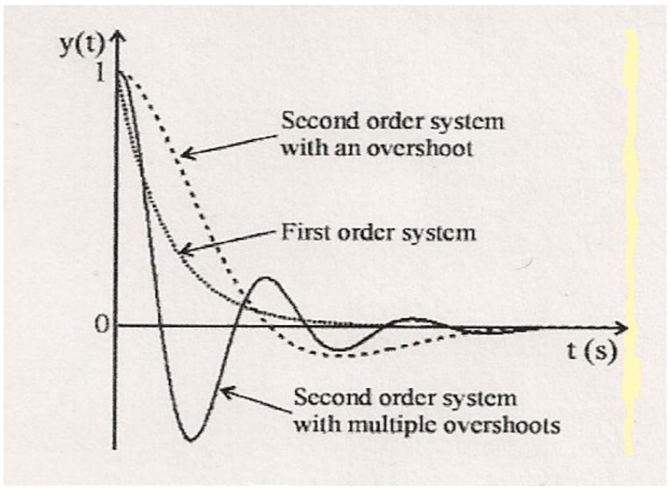


Figure 6.2: Step responses for the error in first and second order systems

In a control system, the human operator makes necessary adjustments fast or slow by looking at the system output. A fast action is required if the output is away from the target, i.e. the error is large, while a slower action may be enough for the output closer to the reference target. Therefore, the information about the amount of change in error signal over a sampling period is also required. A plot of the time variation of error versus the time variation of its change, as given in Fig.6.3, can be used to obtain this information.

The values of error, e, and its change, Δe, start with larger values and terminate at the origin or near the origin in a controlled system. As given in Fig. 6.3, the values of e and Δe from the first order system lie only in second quarter of $e - \Delta e$ space, while the values of e and Δe from a second order oscillatory system travel all four quarters. Since higher order TI systems will also have an oscillating response similar to that of a second order TI system, the plot of $e = f(\Delta e)$ on $e - \Delta e$ space of the second order system given with the solid line in Fig. 6.3 can be used as a general case that is valid for both second and higher order systems [32].

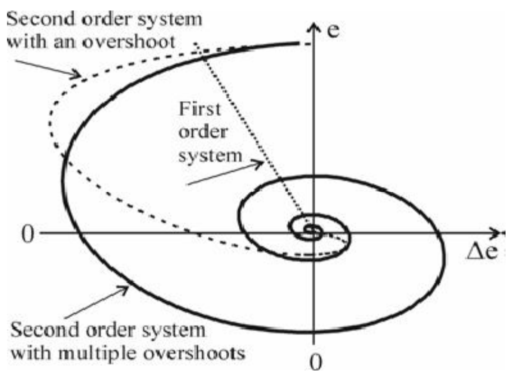


Figure 6.3: $e - \Delta e$ space representing the time response of error, e, versus the time response of the change in error, Δe.

6.3 Fuzzy logic control

Fuzzy Logic Control system is one of the main developments and successes of fuzzy sets and fuzzy logic. A FLC is a rule-base system that implements a nonlinear mapping between its inputs and outputs. A FLC is characterized by four modules:

- fuzzifier;
- defuzzifier;
- inference engine;
- rule base.

6.3.1 Fuzzification

Fuzzification is the process of mapping inputs to the FLC into fuzzy set membership values in the various input universe of discourse. This can be defined also as the operation that maps a crisp object to fuzzy set, i.e., to membership function. Decisions need to be made regarding

- number of inputs
- size of universe of discourse
- number and shape of fuzzy sets.

The size of the universes of discourse will depend upon the expected range (usually up to the saturation level) of the input variables. The number and shape of fuzzy sets in a particular universe of discourse is a trade-off between precision of control action and real-time computational complexity.

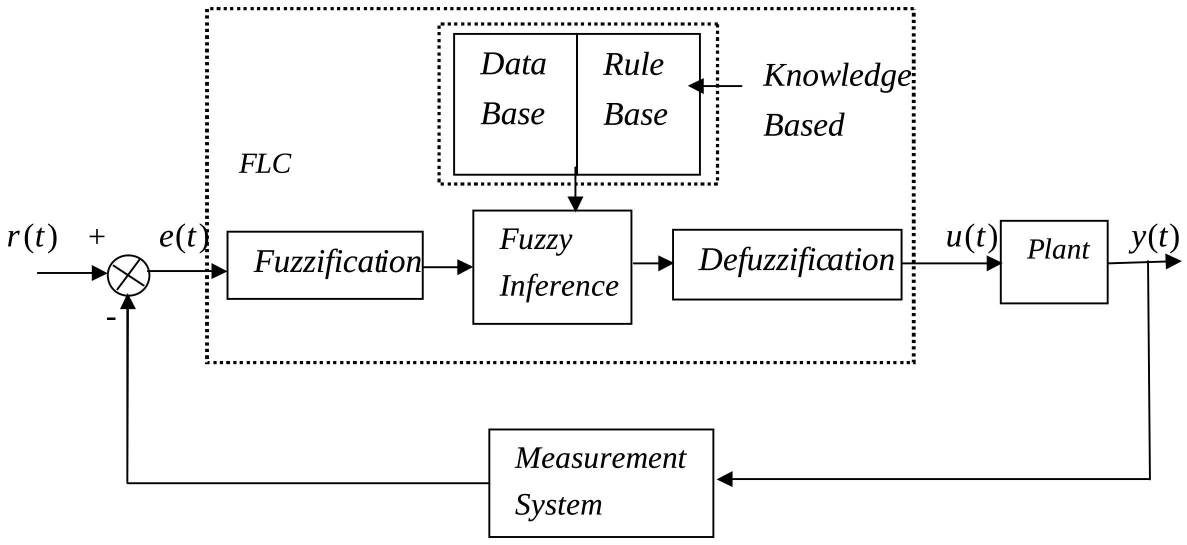


Figure 6.4: General schematic diagram of Fuzzy Logic Control System.

The basic structure of a Fuzzy Logic Control system is shown in Fig 6.4. Fuzzifiers are generally divided in *singleton* and non-*singleton* ones. A singleton fuzzifier maps an object to the singleton fuzzy set centred at the object itself (i.e., with support and core being the set containing only the given object). A non-*singleton* fuzzifier, maps an object to a fuzzy set generally centred at the object itself (i.e., the core of the fuzzy set contains the object) and with support containing the object but being a set bigger than only the object itself. A non-singleton fuzzifier maps an object into a non-singleton fuzzy set generally centred at the object itself. Typically, the use of a singleton fuzzifier is very common. Non-*singleton* fuzzifiers are also used, especially in the presence of e.g., noisy measurements. Indeed, in this case the input crisp value is affected by some uncertainty, thus, the corresponding input fuzzy set can reflect this uncertainty by allowing non-zero membership values around the (noisy) measurement. Therefore, when a non-*singleton* fuzzifier is used, the width of the corresponding fuzzy set is generally proportional to the amount of noise affecting the measurement. Figure 6.5 shows an example of singleton and non-singleton fuzzification. [33].

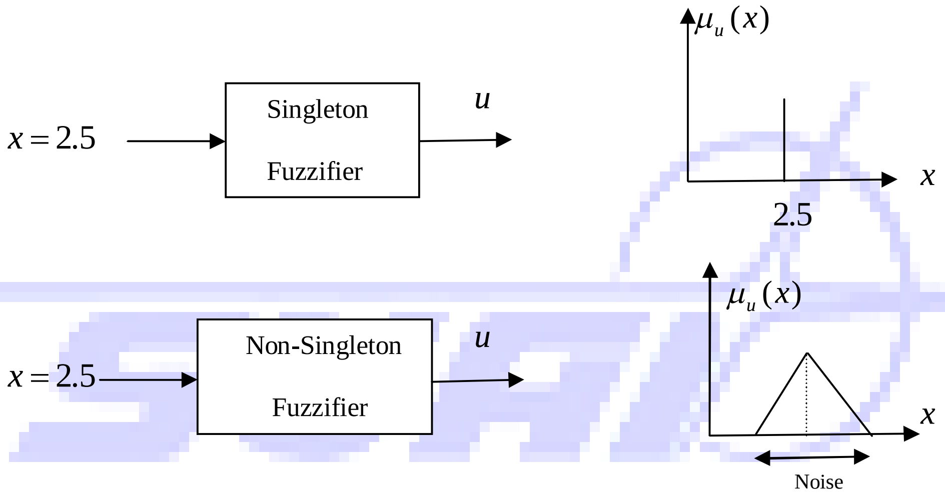


Figure 6.5: Example of singleton and non-singleton fuzzifiers.

The operation principle of a FL controller is similar to a human operator. It performs the same actions as a human operator does by adjusting the input signal looking at only the system output. Two input signals, the main signal and its change for each sampling, to the FL controller are converted to fuzzy numbers first in fuzzifier. Then they are used in the rule table to determine the fuzzy number of the compensated output signal. Finally, the resultant united fuzzy subsets representing the controller output are converted to the crisp values.

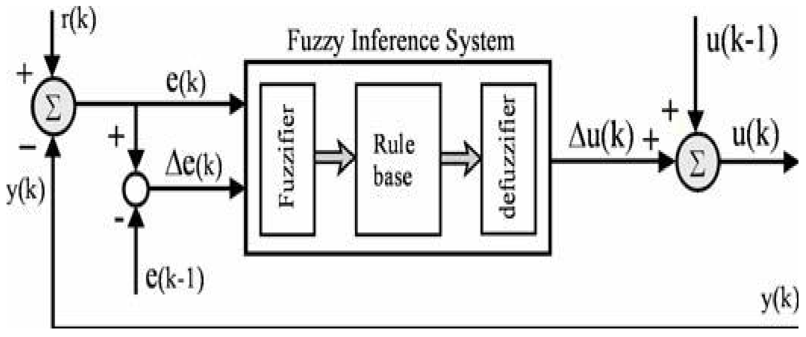


Figure 6.6: Operational principle of FLC.

The FL based controller is designed to act as an integrator controller, such that the resultant incremental output $\Delta u(k)$ in Fig. 6.6 is added to the previous value $u(k-1)$ to yield the current output $u(k)$. Recalling the digital solution of an integrator using Euler's integration as,

$$u(k) = u(k-1) + \Delta u(k). \tag{6.8}$$

In a digital integration, the term $\Delta u(k)$ is expressed as

$$\Delta u(k) = K_1 T_S e(k), \tag{6.9}$$

where K_1 is integral constant, Ts is sampling period, and $e(k)$ is the integrated signal. The change $\Delta u(k)$ on the output of an integrator becomes zero when the input $e(k)$ is zero. Therefore output of an integrator retains the previous value. Hence, (6.8) can be used for both an integrator and FL controller. The difference between an integrator and FL controller is the method that is used to obtain $\Delta u(k)$, which is obtained using (6.9) for an integrator, and using *fuzzy inference system* shown in Fig. 6.4 for FL controller. The latter is the main subject here and explained next.

6.3.2 Fuzzy rule-base

The fuzzy rule-base consist of a set of antecedent consequent linguistic rules of the form in (6.5), this style of fuzzy conditional statement is often called a 'Mamdani'- type rule, after Mamdani (1976) who first used it in a fuzzy rule-base to control steam plant. The rule-base is constructed using *a prior* knowledge from either one or all of the following sources:

- Physical laws that govern the plant dynamics
- Data from existing controllers
- Imprecise heuristic knowledge obtained from experience expert.

If the third item above is used, then knowledge of the plant mathematical model is not required.

Once the inputs are fuzzified, the corresponding input fuzzy sets are passed to the inference engine that processes current inputs using the rules retrieved from the rule base. Fuzzy inference is therefore the process of mapping membership values from the input windows, through the rule-base, to output window(s). In other words, fuzzy inference is a method that interprets the values in the input vector and, based on some set of rules, assigns values to the output vector. As shown in Fig. 6.6, there are two inputs to the *fuzzy inference system*. One is the control error $e(k)$, which is the difference between the reference signal $r(k)$ and the output signal $y(k)$, the other one is the change in this error $\Delta e(k)$. These two inputs, defined as in (6.10) and (6.11), are first fuzzified and converted to fuzzy membership values that are used in the rule base in order to execute the related rules so that an output can be generated.

$$e(k) = r(k) - y(k). \tag{6.10}$$

$$\Delta e(k) = e(k) - e(k-1). \tag{6.11}$$

The fuzzy rule base, which may also be called as the fuzzy decision table, is the unit mapping two crisp inputs, $e(k)$ and $\Delta e(k)$ to the fuzzy output space defined on the universe of $\Delta u(k)$. For simplification iteration counter (k) will be omitted from now on as long as it is not required to be referred. The time response of the control error e for a step input can be represented by the generalized step response error of a second order system as shown in Fig. 6.7 where the crisp e and Δe axis are partitioned into seven fuzzy subsets as negative big (NB), negative medium (NM), negative small (NS), zero (ZZ), positive small (PS), positive medium (PM), and positive big (PB). This error signal may have a damped or an oscillatory response with a decaying exponential component. The latter one is considered for constructing the rule table since it includes overshoot effects leading the rule base to represent more generalized cases.
The fuzzy rules represent the knowledge and abilities of a human operator who makes necessary adjustments to operate the system with minimum error and fast response. In order to model the actions that a human operator would decide whether the change, Δu , in the controller output is to be increased or decreased according to the error e and its change Δe, it is necessary to observe the behaviours of the error signal e and its change Δe on different operating regions, as shown by the Roman numbers in Fig. 6.7. The output Δu from the FL controller is the change that is required to increase or decrease the overall control action to the controlled system. Therefore, the signs of e and Δe are used to determine the signs of Δu , which determines whether the overall control signal is to be increased. The sign of Δu should be positive if u is required to be increased and it should be negative otherwise.

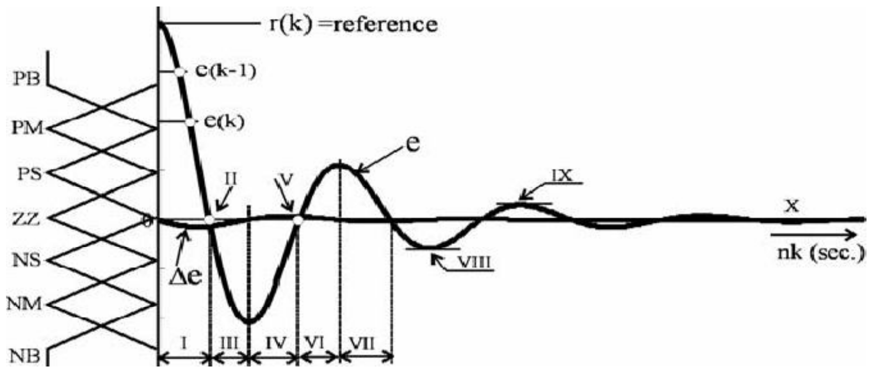


Figure 6.7: Operating regions and fuzzy partitioning of the time responses of error and error change for a generalized second order system

This simple rule is applied as follows to determine the sign of Δu .

<u>At region I:</u> $e = '+'$ and $\Delta e = '-'$. The error is positive and decreasing toward zero. Therefore, Δu is set to positive to reduce the error.
<u>At region II:</u> $e = '0'$ and $\Delta e = '-'$. The error is zero, but is getting away and increasing in negative direction. Therefore, a negative Δu is assigned to reduce the error.

<u>At region III:</u> $e = '-'$ and $\Delta e = '-'$. The error is negative and increasing. Therefore, Δu is still required to be negative to reduce the error toward zero.

<u>At region IV:</u> $e = '-'$ and $\Delta e = '+'$. The error is still negative, but decreasing. Therefore, negative Δu is kept on to reduce the negative error.

<u>At region V:</u> $e = '0'$ and $\Delta e = '+'$. The error is zero, but increasing in positive direction. Therefore, a positive Δu is applied to increase the controlled output so that the voltage error is reduced.

<u>At region VI:</u> $e = '+'$ and $\Delta e = '+'$. The error is positive and increasing. A positive value for Δu will be suitable to reduce the error.

<u>At region VII:</u> A repeat of region I.

<u>At region VIII:</u> $e = '-'$ and $\Delta e = '0'$. The error is negative and constant since there is no change. Therefore a negative value for Δu should be assigned to decrease the error.

<u>At region IX:</u> $e = '+'$ and $\Delta e = '0'$. The error is positive and constant. Therefore a positive Δu is required to reduce the error.

<u>At region X:</u> $e = '0'$ and $\Delta e = '0'$. The error is both zero and constant. Therefore Δu is set to zero since no change is required for the control signal.

The signs of Δu in the regions described above as I to X are listed in Table 6.1, which can be summarized as follows:

IF e is zero **THEN** Δu takes the sign of Δe, **ELSE** Δu takes the sign of e (6.12)

Table 6.1 The signs of basic control action

	Operating regions									
	I	II	III	IV	V	VI	VII	VIII	IX	X
e	+	0	-	-	0	+	+	-	+	0
Δe	-	-	-	+	+	+	-	0	0	0
Δu	+	-	-	-	+	+	+	-	+	0

Both Fig. 6.7 and Table 6.1 show that each one of e, Δe, and Δu has three different options for the signs to be assigned. They are either positive or negative if not zero. Keeping in mind these three options, which are represented by three fuzzy sets namely positive (P), negative (N), and zero (Z), an initial rule decision table with nine rules can be formed as shown in Table 6.2, where the main part without shading represents the rules in terms of the signs of Δu . Representing the input crisp variables e and Δe by three fuzzy sets, P, N, and Z means that these input spaces are partitioned into three fuzzy regions each yielding a fuzzy output space with nine rules maximum as given in Table 6.2. A nine-rule fuzzy decision table may be sufficient for some applications. However, many applications require more rules than nine. In order to construct a fuzzy rule decision table with more than nine rules, the input spaces must be partitioned into more than three regions each.

Table 6.2: Initial rule table

e \ Δe	N	Z	P
P	P	P	P
Z	N	Z	P
N	N	N	N

A plot of $e(t)$ vs $\Delta e(t)$ can be used to determine the regions for partitioning as shown in Fig. 6.8 where $e(t)$ and $\Delta e(t)$ axes are partitioned into three and five regions in Figs. 6.8(a) and 6.8(b), respectively. Triangular type membership functions are used for partitioning the crisp universes into fuzzy subsets. Different membership functions such as Gaussian, trapezoidal, and bell could have also been used. Each one of these membership functions has its own effects on the FLC output. However, triangular membership functions are more convenient for expressing the concept because it is easier to intercept membership degrees from a triangle. Therefore the following function is used to represent the fuzzy triangular membership functions.

$$\mu(x) = \max\left[\min\left(\frac{x - x_1}{x_2 - x_1}, \frac{x_3 - x}{x_3 - x_2}\right), 0\right], \qquad (6.13)$$

where, x is the crisp values from one of the three universes e, Δe, and Δu . x_1 is the left end point on the corresponding crisp universe as the x_2 and x_3 are the crisp points corresponding peak and right end points, respectively. The plot of e vs Δe as in Fig. 6.8 is useful in partitioning since the upper and lower limits of the values of e and Δe are shown separately enabling one to set different boundary limits for e and Δe. In this case the scaling of the fuzzy sets representing the partitioning will also be different for e and Δe. This is important because of the differences in minimum and maximum values of e and Δe as shown in Figs 6.8 and 6.9. If the same scaling of the fuzzy sets is used for both e and Δe, the fuzzy values of Δe do not change much compared to those of e as shown in Fig. 6.9 where Δe swings around zero with membership degrees higher in the fuzzy set ZZ and very lower in NS and PS, and without any membership degrees in the other fuzzy sets. Therefore Fig. 6.9 is more convenient to be used for fuzzy partitioning of crisp axes.
Besides, Figs. 6.9(a) and 6.9(b) both have the similar appearance with Table 6.2. If Fig. 6.9(a) and Table 6.2 are compared, it can readily be seen that horizontal and vertical axes in

Fig. 6.9(a) corresponds to the first definition row and column in Table 6.2, respectively. Since inner part of Table 6.2 contains the fuzzy rules that are represented by fuzzy sets defined on the universe of discourse Δu . The $e-\Delta e$ space in Fig. 6.8(a) can be used to represent the universe of discourse Δu , which is the output space of the FL controller. Therefore, using the expression (6.12), the space of $e-\Delta e$ representing Δu can be partitioned into the regions positive (P), negative (N), and zero (Z) as given in Figs. 6.9(a) and 6.9(b).

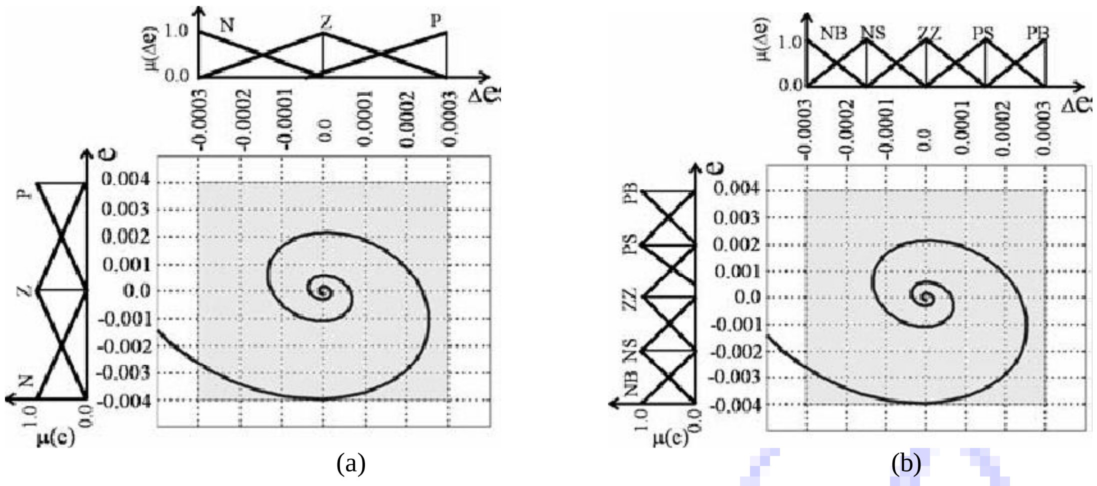


Figure 6.8: Partitioning input spaces into (a) three and (b) five regions.

The fuzzy rule decision table used in this study consists of twenty-five rules obtained by portioning e and Δe axes into five regions each as shown in Figs. 6.8(b) and 6.9(b), which are going to be used to explain the construction of the rule decision table. As it can be seen in Fig. 6.9(b), $e-\Delta e$ space representing the Δu space has been partitioned into three regions namely P, N, and Z, while each one of e and Δe axes are partitioned into five regions as NB, NS, ZZ, PS, and PB. Similar to e and Δe axes, Δu space can also be partitioned into five or more regions.

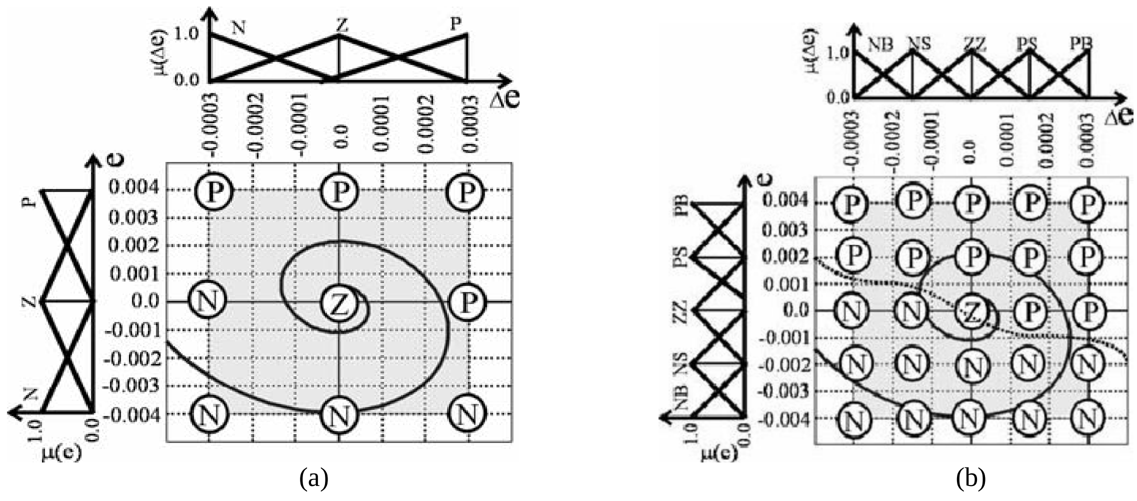


Figure 6.9: Initial rule assignment in the output space, Δu . The input space partitioned into three regions (a). The input space partitioned into five regions (b).

A close look at the partitioned e and Δe axes show that there is a transition region between the neighbouring fuzzy sets. It can also be seen that a fuzzy set zero is used to represent the transition between positive and negative fuzzy values. This transition region is the nature of fuzzy sets, and is called as shaded region. A similar transition should also be established between the positive (P) and negative (N) regions in Δu space where the positive and negative values are mostly located toward upper right and lower left corners, respectively, as separated by a dotted line in Fig. 6.9(b). Therefore a shaded region represented by the fuzzy set zero (ZZ) can be placed on the main diagonal separating the positive and negative parts as given in Table 6.3.

Table 6.3: Initial fuzzy rule decision table

		Δe				
		NB	NS	ZZ	PS	PB
e	PB	Z	P	P	P	P
	PS	N	Z	P	P	P
	ZZ	N	N	Z	P	P
	NS	N	N	N	Z	P
	NB	N	N	N	N	Z

If the output space, Δu , is to be partitioned into more than three regions as it has been done for e and Δe axes, then the positive and negative regions should be divided into sub-regions such as NB, NS, PS, and PB. The inner partitioning of positive and negative regions should be done in such a way that the order from NB to PB becomes similar to that of e and Δe axes. Therefore, if Δu surface is partitioned into five regions consisting of the fuzzy sets NB, NS, ZZ, PS, and PB, Table 6.3 can be converted to Table 6.4 where the regions from NB to PB have transitions as shown in Figs. 6.8 and 6.9, which e and Δe axes are partitioned into 5 subsets. A number from 1 to 25 has been assigned to each rule in Table 6.4

Table 6.4: Modified fuzzy rule decision table

		Δe				
		NB	NS	ZZ	PS	PB
e	PB	ZZ 1	PS 2	PS 3	PB 4	PB 5
	PS	NS 6	ZZ 7	PS 8	PS 9	PB 10
	ZZ	NS 11	NS 12	ZZ 13	PS 14	PS 15
	NS	NB 16	NS 17	NS 18	ZZ 19	PS 20
	NB	NB 21	NB 22	NS 23	NS 24	ZZ 25

6.3.3 The inference and fuzzy reasoning

The final control action is the crisp output that is defuzzified from the resultant fuzzy values of the fuzzy rule base. The fuzzy output of the rule base is obtained by triggering the active rules for the k^{th} sampling instant corresponding to the values $e(k)$ and $\Delta e(k)$ as shown in Fig. 6.10. For any point $(e(k), \Delta e(k))$ on the trajectory plot of $e(k)$ vs $\Delta e(k)$, there are maximum two intercepting fuzzy sets on each one of the universes $e(k)$ and $\Delta e(k)$. Thus, for any sampling instant, the value of $e(k)$ activates only one or two fuzzy sets in the universe of $e(k)$. Similarly, the value of $\Delta e(k)$ for the k^{th} sampling instant also activates only one or two fuzzy sets in the universe of $\Delta e(k)$.

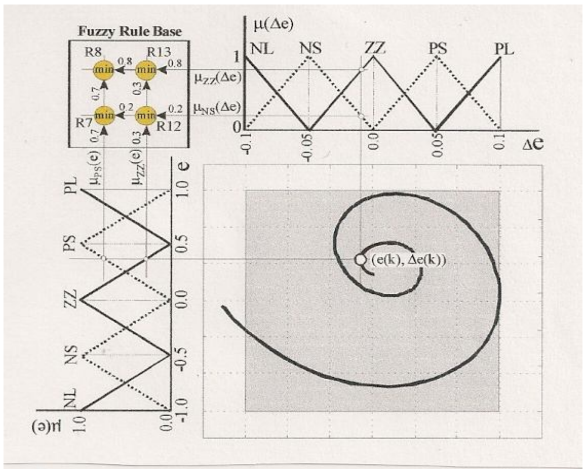


Figure 6.10: An illustration of fuzzy reasoning.

The point $(e(k), \Delta e(k)) = (0.3, -0.01)$ on the trajectory plot shown in Fig. 6.10 intercepts with the fuzzy sets ZZ and PS in the universe of e and with the fuzzy sets NS and ZZ in the universe of Δe. Therefore, the following rules are activated for the given data point $(e(k), \Delta e(k))$.

Rule 7 (R7): if e is PS and Δe is NS then Δu is ZZ. (6.14)

Rule 8 (R8): if e is PS and Δe is ZZ then Δu is PS. (6.15)

Rule 12 (R12): if e is ZZ and Δe is NS then Δu is NS. (6.17)

Rule 13 (R13): if e is ZZ and Δe is ZZ then Δu is ZZ. (6.18)

The membership degrees of e in the fuzzy sets PS and ZZ are obtained as $\mu_{PS}(e) = 0.7$ and $\mu_{ZZ}(e) = 0.3$, respectively, and the membership degrees of Δe in the fuzzy sets NS and ZZ are $\mu_{NS}(\Delta e) = 0.2$ and $\mu_{ZZ}(\Delta e) = 0.8$, respectively. The triangular membership function (6.13) can be used to obtain these memberships. Then the application of the min operator results in the following membership values from each active rule to be used in the output space Δu .

$$\mu_{R7}(\Delta u) = \min(\mu_{PS}(e), \mu_{NS}(\Delta e)) = \min(0.7, 0.2) = 0.2. \tag{6.19}$$

$$\mu_{R8}(\Delta u) = \min(\mu_{PS}(e), \mu_{zz}(\Delta e)) = \min(0.7, 0.8) = 0.7. \tag{6.20}$$

$$\mu_{R12}(\Delta u) = \min(\mu_{ZZ}(e), \mu_{NS}(\Delta e)) = \min(0.3, 0.2) = 0.2. \tag{6.21}$$

$$\mu_{R13}(\Delta u) = \min(\mu_{ZZ}(e), \mu_{zz}(\Delta e)) = \min(0.3, 0.8) = 0.3. \tag{6.22}$$

The resultant fuzzy outputs in the universe of Δu for these four active rules are depicted in Fig. 6.11 with the membership degrees shown as $\mu_{r7}, \mu_{r8}, \mu_{r12}$, and μ_{r13}.

6.3.4 Defuzzification

The resultant membership values of the active rules determine the weights of the fuzzy sets in the universe of Δu as shown in Fig. 6.10 with the shaded parts of the fuzzy sets. Then the averaged value of the union of these shaded fuzzy sets is used to obtain final crisp output as $\Delta u(k)$. This final process is called defuzzification of the fuzzy output. Several defuzzification methods have been applied in literature. However, the method called *the centre of area* is widely used in fuzzy logic control applications. The use of *the centre of area* method yields the following.

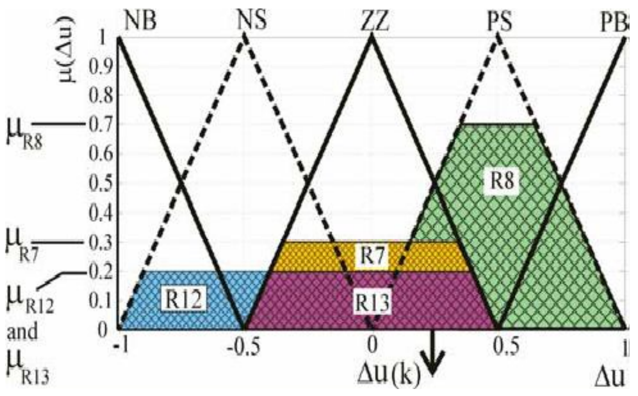


Figure 6.11: Membership function used to represent fuzzy partitioning in universe of Δu .

$$\Delta U_R(k) = \frac{\sum_{i=7,8,12,13} \mu_{Ri}(\Delta u_R)\Delta u_R(Ri)}{\sum_{i=1}^{4} \mu_i(\mu V_R)} \tag{6.23}$$

$$\Delta U_R(k) = \frac{0.2(0.0)+0.7(0.5)+0.2(-0.5)+0.3(0.0)}{0.2+0.7+0.2+0.3} = \frac{0.25}{1.4} = 0.17857, \tag{6.24}$$

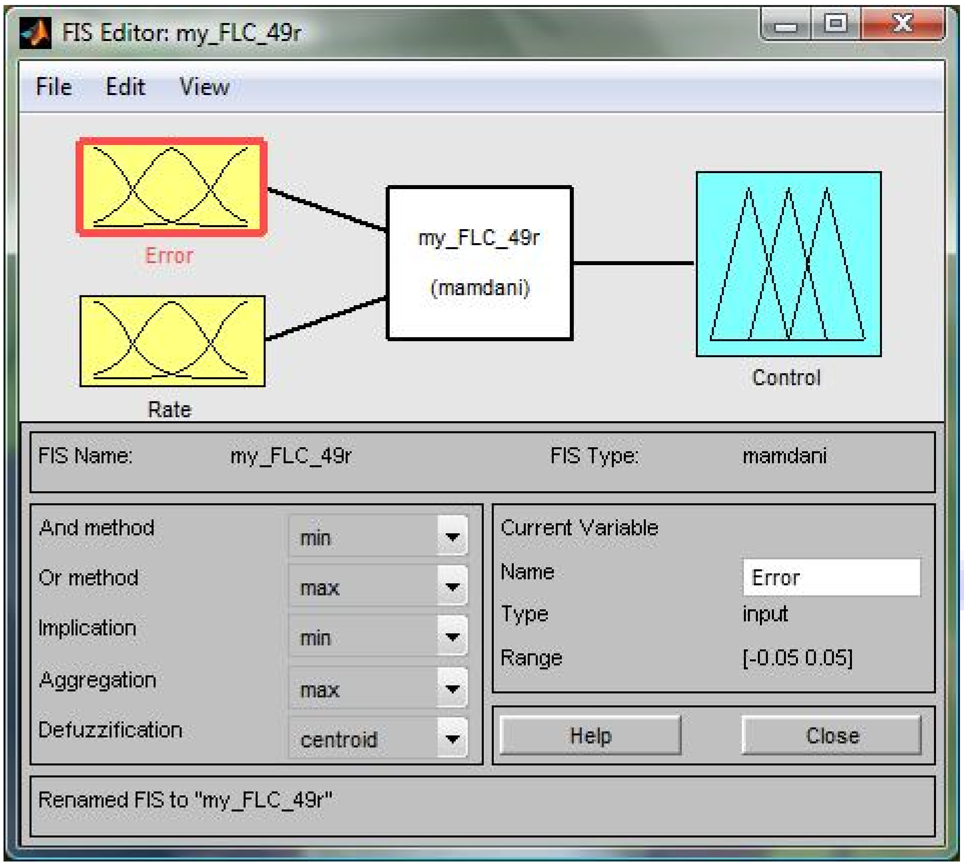
where, $\Delta\mu(Ri)$ is the crisp Δu value corresponding to the maximum membership degree of the fuzzy set that is an output from the rule decision table for the rule Ri. For example, for the rule 12, the output fuzzy set is NS and the crisp value of $\Delta u(R12)$ is (-0.5), because the membership degree of (-0.5) in the fuzzy set NS is 1.0 as shown in Fig. 6.11.
Thus, defuzzification is the procedure for mapping from a set of inferred fuzzy control signals contained within a fuzzy output window to a non-fuzzy (crisp) control signals.

6.4 Matlab/Simulink Synthesis of the FL autopilot

In order to successfully design and simulate the FL autopilot, the *FIS* has to be designed first based on all the information generated earlier about FLC and then integrate the controller into a Simulink program that has all the features of the type of FLC we intend to design for our ELV. For this research a PD fuzzy logic controller will be simulated.

6.4.1 Building the FIS

Using the *FIS* editor of Matlab, the two inputs to the fuzzy controller are the error (e) which measures the system performance and the rate at which the error changes (Δe), whereas the output of the control signal (Δu), were created as shown in Fig 6.12.



FIS Name: my_FLC_49r FIS Type: mamdani

And method: min
Or method: max
Implication: min
Aggregation: max
Defuzzification: centroid

Current Variable
Name: Error
Type: input
Range: [-0.05 0.05]

Renamed FIS to "my_FLC_49r"

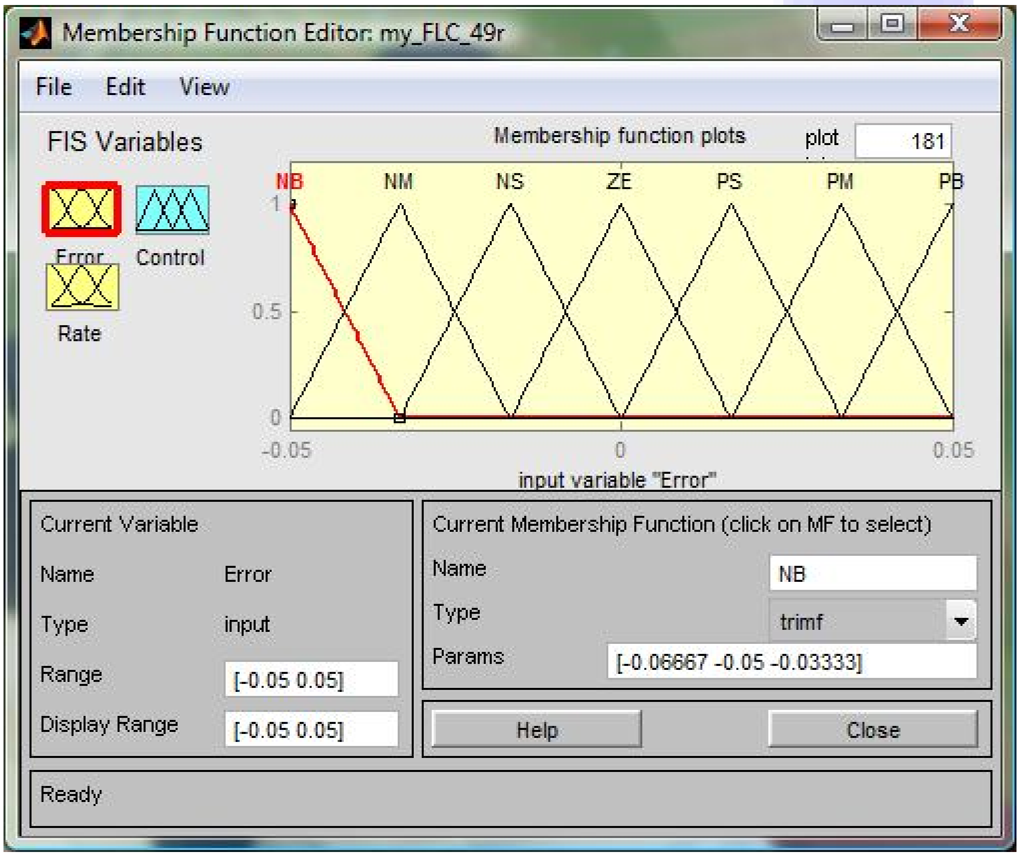
Figure 6.12: FIS Editor depicting the inputs and output of the FLC

The membership function of a variable is expressed into seven fuzzy sets defined in Table 6.5, and each membership function is constrained to be triangular so each membership function has three parameters (a modal and two half-width).The membership function maps the crisp values into fuzzy variables. The choice of membership function has an important bearing on the performance of the fuzzy logic controller. Fig. 6.13 depicts the seven membership function as implemented for the error signal; the same was applied for both the error rate and control signal.

Table 6.5: Fuzzy set of the implemented variable

NB	NEGATIVE BIG
NM	NEGATIVE MEDIUM
NS	NEGATIVE SMALL
ZE	ZERO
PS	POSITIVE SMALL
PM	POSITIVE MEDIUM
PB	POSITIVE BIG

The Membership Function Editor is the tool that lets you display and edit all of the membership functions for the entire fuzzy inference system, including both input and output variables. The membership functions from the current variable are displayed in the main graph. Below the Variable Palette is some information about the type and name of the current variable. There is one text field that lets you change the limits of the current variable's range (universe of discourse) and another that lets you set the limits of the current plot (which has no real effect on the system). In the lower right of the window are the controls that let you change the name, position, and shape of the currently selected membership function.



Current Variable

Name	Error
Type	input
Range	[-0.05 0.05]
Display Range	[-0.05 0.05]

Current Membership Function (click on MF to select)

Name	NB
Type	trimf
Params	[-0.06667 -0.05 -0.03333]

Ready

Figure 6.13: Membership Function Editor depicting the properties of the variable e .

The two inputs, error and error rate, result in 49 rules as shown in Table 6.6. The knowledge required to generate the fuzzy rules can be derived from an offline simulation. However, it has been noticed that, for monotonic systems, a symmetrical rule table is very appropriate, although sometimes it may need slight adjustments based on the behaviour of the specific system. If the system dynamics are not known or are highly nonlinear, trial-and-error procedures and experience play an important role in defining the rules [34]. The universe of discourse is set between -0.05 to 0.05, which implies the range of pitch angle, $\pm 0.05 rad$ [35].

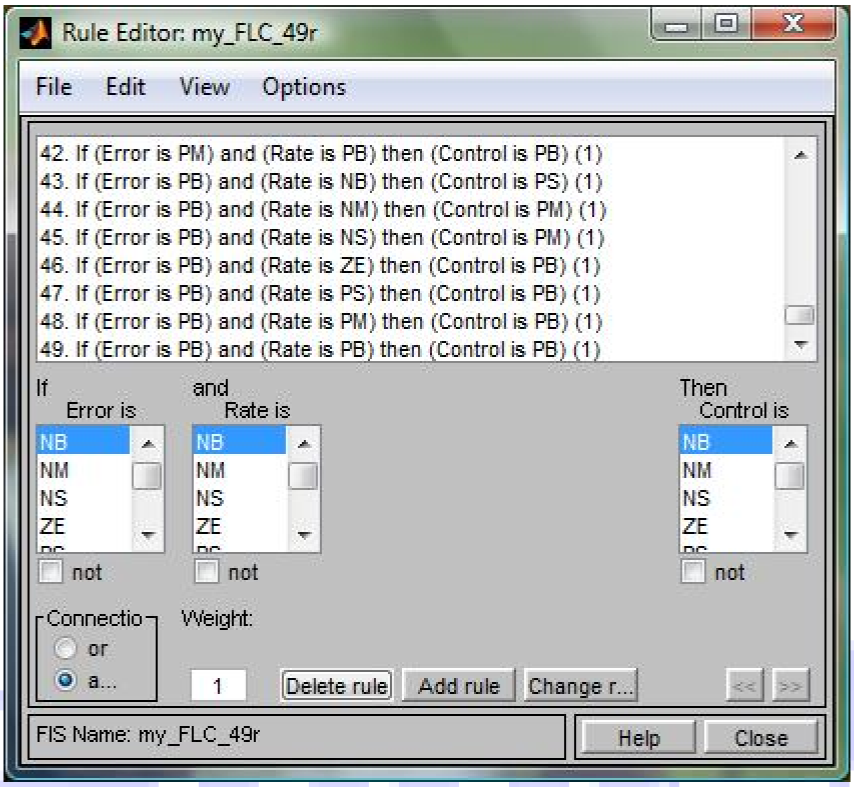
Table 6.6: The implemented fuzzy rule decision table

		Δe						
		NB	NM	NS	ZE	PS	PM	PB
	NB	NB	NB	NB	NB	NM	NM	NS
	NM	NB	NM	NM	NM	NS	NS	ZE
	NS	NM	NM	NS	NS	ZE	ZE	PS
e	ZE	NM	NS	NS	ZE	PS	PS	PM
	PS	NS	ZE	ZE	PS	PS	PM	PM
	PM	ZE	PS	PS	PM	PM	PM	PB
	PB	PS	PM	PM	PB	PB	PB	PB

The Rule editor is used to input the 49 rules given in Table 6.6. It contains a large editable text field for displaying and editing rules as shown in Fig. 6.14. The Boolean operator '*min*' is used for the verbal connector *'and'* to simulate the *input space of the rules* that have the structure as in expression (6.25).

$$\textit{If} \quad e \ \text{ is A } \textit{and} \ \ \Delta e \ \text{ is B } \ \textit{then} \ \ \Delta u \ \text{ is C.} \qquad (6.25)$$

Where A, B, and C in (6.25) represent any one of these fuzzy subsets NB, NM, NS, ZE, PS, PM, and PB as defined in Table 6.5. The input space in (6.25) is the part that represented by the expression (e is A *and* Δe is B). Therefore the '*min*' operator in the GUI of Fig. 6.12 is used to model the input space of 49 rules used by the FLC. The outputs of the *'min'* operator indicate the strength (membership degree) of the rules in the output space Δu. The implementation of the rule input space by the expression (e is A *and* Δe is B) as shown in Fig. 6.13 is nothing but the Fuzzification of the two crisp inputs e and Δe for all the rules.



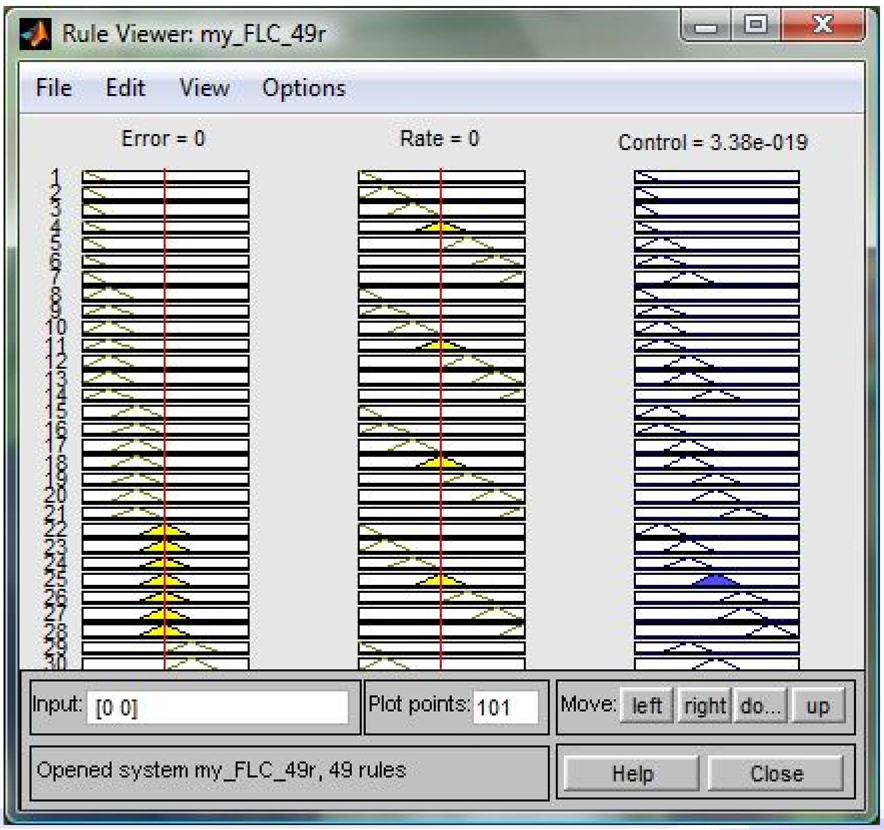
Rule Editor: my_FLC_49r

42. If (Error is PM) and (Rate is PB) then (Control is PB) (1)
43. If (Error is PB) and (Rate is NB) then (Control is PS) (1)
44. If (Error is PB) and (Rate is NM) then (Control is PM) (1)
45. If (Error is PB) and (Rate is NS) then (Control is PM) (1)
46. If (Error is PB) and (Rate is ZE) then (Control is PB) (1)
47. If (Error is PB) and (Rate is PS) then (Control is PB) (1)
48. If (Error is PB) and (Rate is PM) then (Control is PB) (1)
49. If (Error is PB) and (Rate is PB) then (Control is PB) (1)

FIS Name: my_FLC_49r

Figure 6.14: written rules in the Rule Editor.

Now the system has been completely defined: we've got the variables, membership functions, and rules necessary to control action. It would be nice, at this point, to look at a fuzzy inference diagram like and verify that everything is behaving the way we think it should. This is exactly the purpose of the Rule Viewer depicted in Fig. 6.15. The column (yellow) of plots shows how the input variable is used in the rules, while the column (blue) of the plots shows how the output variable is used in the rules. The bottom-right plot shows hoe the output of each rule is combined to make an aggregate output and then defuzzified. The Rule Viewer displays a roadmap of the whole fuzzy inference process. It's based on the fuzzy inference diagram described in section 6.3.3. The two small plots across the top of the figure represent the antecedent and consequent of the first rule. Each rule is a row of plots, and each column is a variable. So the first two columns of plots (yellow plots) show the membership functions referenced by the antecedent, or if-part, of each rule. The third column of plots (blue plots) shows the membership functions referenced by the consequent, or then-part of each rule. There is an index line across the input variables plots that you can move left and right by clicking and dragging with the mouse. This changes the input value. When you release the line, a new calculation is performed, and you can see the whole fuzzy inference process take place before your eyes. Finally the aggregation occurs down the third column, and the resultant aggregate plot is shown in the single plot to be found in the lower right corner of the plot field. The defuzzified output value is shown.



Opened system my_FLC_49r, 49 rules

Figure 6.15: The fuzzy inference in the Rule Viewer GUI.

The Rule Viewer shows one calculation at a time and in great detail. In this sense, it presents a sort of micro view of the fuzzy inference system. If you want to see the entire output surface of your system, that is the entire span of the output set based on the entire span of the input set, you need to open up the Surface Viewer as shown in Fig. 6.16. This is the last of the five basic GUI tools in the Fuzzy Logic Toolbox. This three-dimensional plot shows the output surface for any output of the system versus any the two inputs to the system The Surface Viewer has a special capability that is very helpful in cases with two (or more) inputs and one output: you can actually grab the axes and reposition them to get a different three-dimensional view on the data. [36].

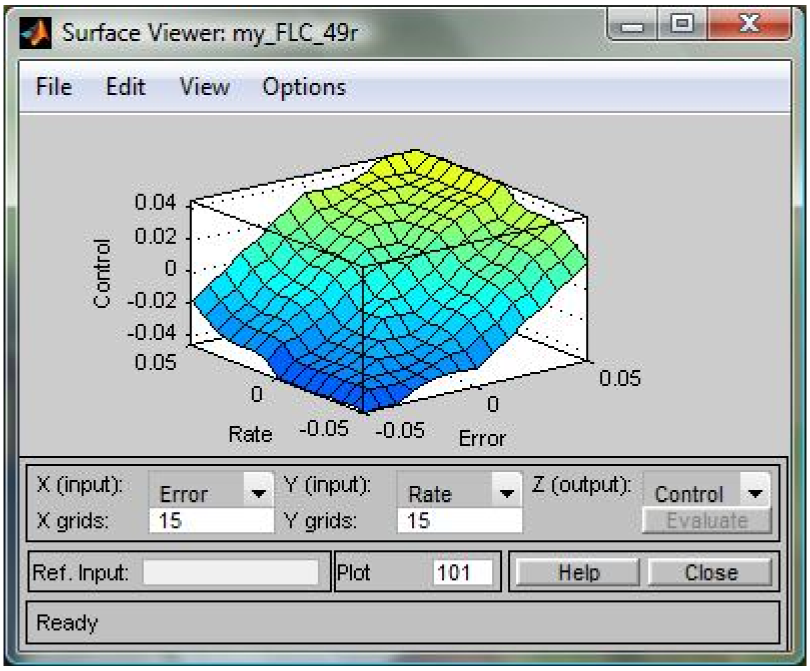


Figure 6.16: Three-Dimensional plot of the output surface.

6.4.2 The PD-FL autopilot

The Simulink model for a PD-like fuzzy logic control [37] of the ELV was implemented by linking the FIS designed above with the Simulink program in Fig. 6.17. Note the ELV transfer function used here is as earlier defined in (5.16). Manual tuning means via trial-and-error was used to obtain the proportional, derivative and output gains associated with the controller. The integral of absolute error is used as a measure of the system performance since it is known to give a better all round performance indicator of a control system response where overshoot, settling time and rise time are the are the main consideration[38].

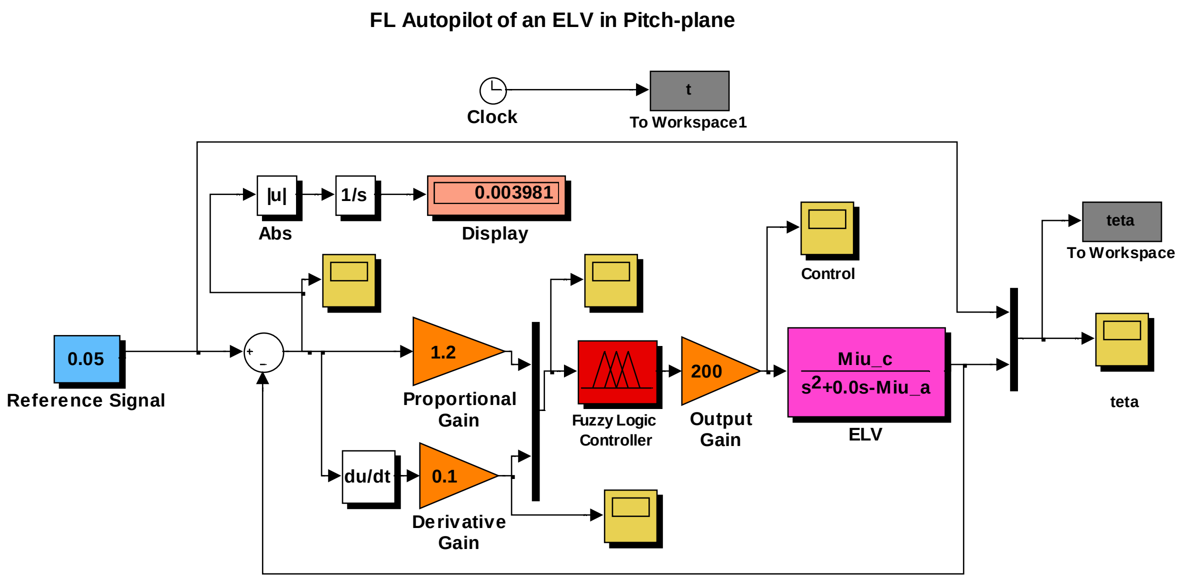


Figure 6.17: Simulink model of the PD-FL autopilot.

The result of the simulated FL autopilot is shown in Fig. 6.18, and the followings are the results of the scaling gains;

- Proportional Error input scale: 1.2
- Error Deviation input scale: 0.1
- Output Scale: 200

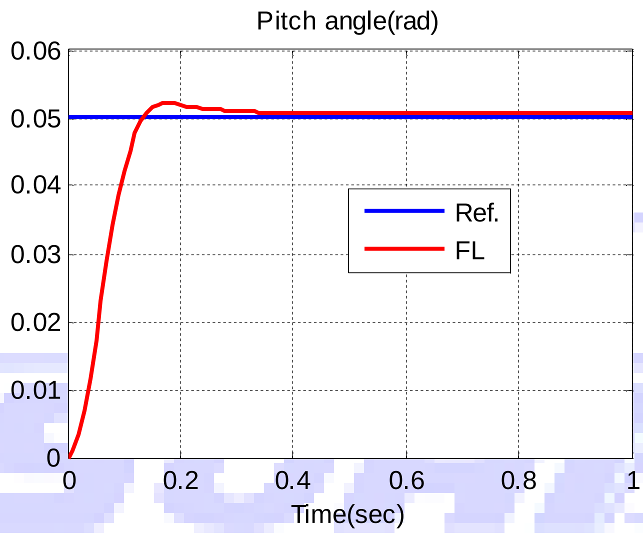


Figure 6.18: Set point command tracking of the PD-FL autopilot.

At this point, it will be prudent to look under the mask of the *FIS Wizard* subsystem of the fuzzy logic controller to see the implementation of our designed FIS since an acceptable result has been obtained. Fig. 6.19 shows part of the implementation (the entire model is too large to show in this document). As the figure shows, the Fuzzy Logic Controller block uses built-in Simulink blocks to implement the FIS. Although the models can grow complex, this representation is better suited than the *s-function sffis* for efficient code generation.

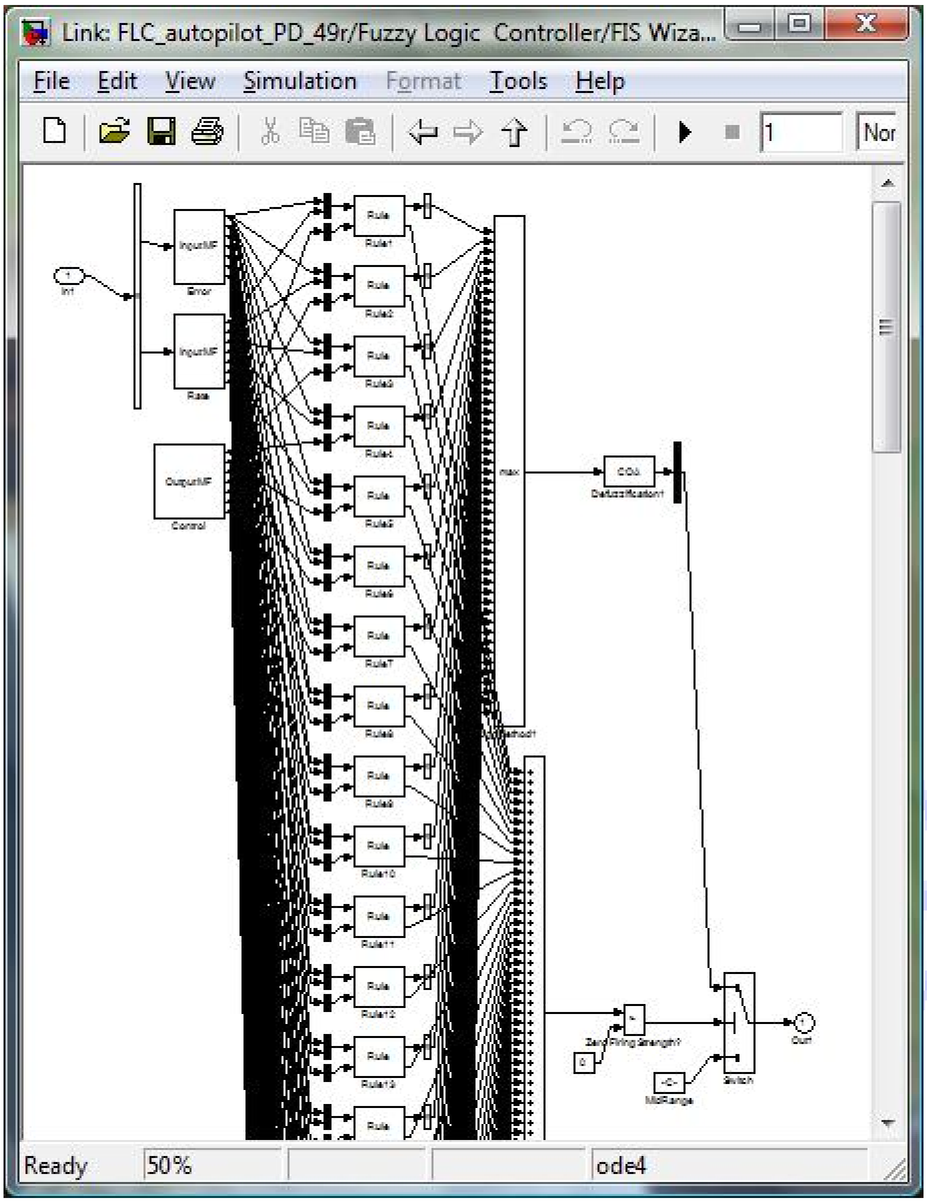


Figure 6.19: Implementation of the designed FIS by the *FIS Wizard.*

6.4.3 Disturbance rejection

Now that we satisfied with the performance of our FL autopilot, it will be fare for this research to investigate the effect of disturbance to the Fl autopilot as was done with the modern and classical design. The Simulink model in Fig. 6.20 incorporates white noise power 0.002 and the result of the simulated disturbance to the FL autopilot is shown in Fig. 6.21

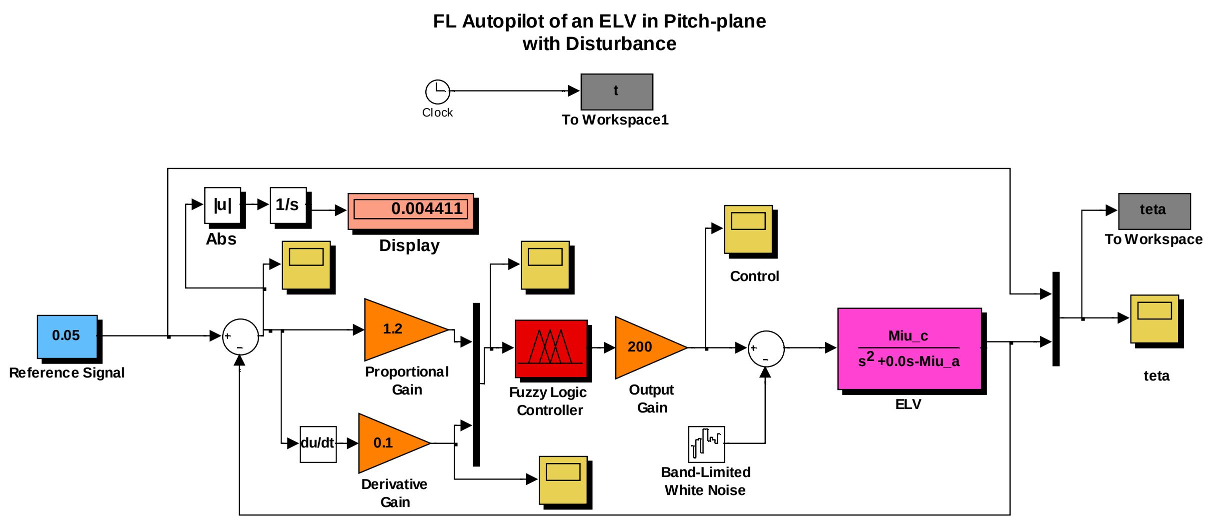


Figure 6.20: Simulink model of PD-FL autopilot with simulated disturbance.

It could be seen in Fig. 6.21 that the FL autopilot handles the specified amount of disturbance well and without losing it trajectory tracking of the reference signal, though the *percent overshoot* was observed to have increased to 14.4%.

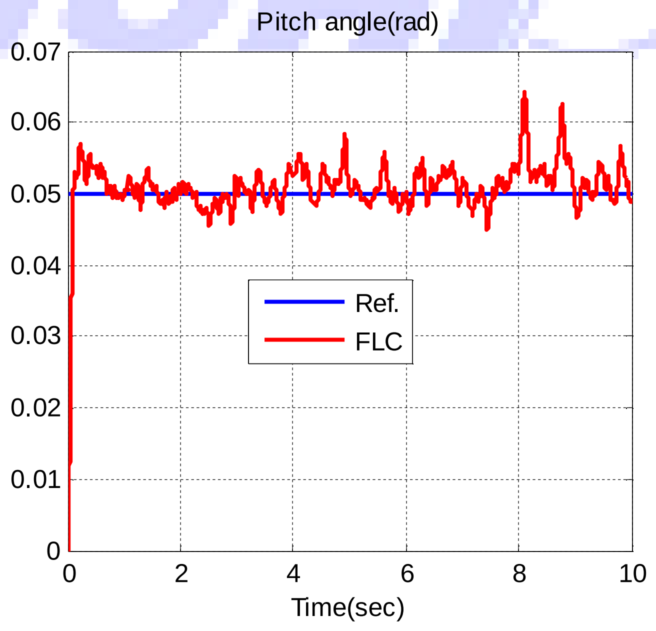


Figure 6.21: Set point command tracking of FL autopilot with disturbance.

For the final investigation, the disturbance will be increased just as was done with the PID autopilot, to a value of 0.1 PSD. The simulated result is depicted in Fig. 6.22. It is interesting to note that excursion from tracking the desired or reference trajectory beyond the 20% *overshoot* margin specified in the design specification occurred after about 5s. This is compared to that of the PID autopilot in Fig. 5.14, and was observed to have drifted outside the specified margin just within the first 2s.

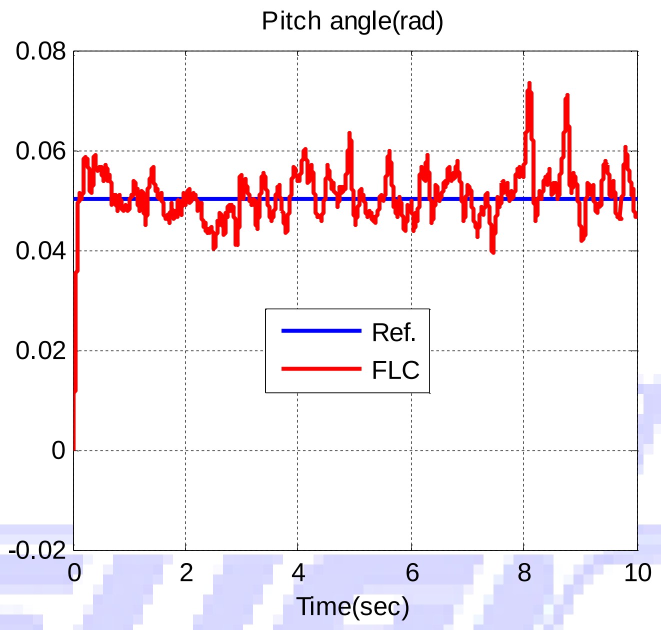


Figure 6.22: Set point command tracking of FL autopilot with increased disturbance.

Chapter 7

Comparative Analysis of Modern and Intelligent Control Design Methods

7.1 Introduction

To explore the viability of applying FLC to launch vehicle ascent and, specifically, the pitch control problem, two different controllers have also been presented. Chapter 4 shows the used of the renown modern control technique while chapter 5 the classical PID methods were applied. Using the classical method as benchmark of existing performance, the performance of the FLC autopilot in the pitch control problem was assessed. This chapter presents the evaluation in two ways. Firstly, the classical, modern and intelligent controllers developed within this thesis are comparatively analyzed to determine if FLC approach is capable of generating performance at least equivalent to those produced by the classical autopilot. Secondly, the two autopilots (LQG and FL) are compared to reveal the advantages and disadvantages of using the FL approach. In this manner, both the product and the process are evaluated to discover the true viability of the FL framework considered here in launch vehicle applications.

7.1.2 Nominal Performance Comparison

To understand how the synthesized controllers contrast in nominal performance, the time domain responses of the FL and PID controllers are presented as comparison plots in Fig. 7.1. and Fig. 7.2 present that of FL and LQG. Additionally, Table 7.1 provides a summary of the performance matrices of the three controllers.

Table 7.1: Summary of performance characteristics of Modern, Classical and Intelligent autopilots.

Time response specifications	**LQG**	**FLC**	**PID**
Settling Time (T_s)	3.0s	0.3s	4.0s
Rise Time (T_r)	0.25s	0.12s	0.45s
Percent Overshoot ($\%PO$)	20%	4%	10%
Steady State Error (e_{ss})	0.00	0.0007	0.00

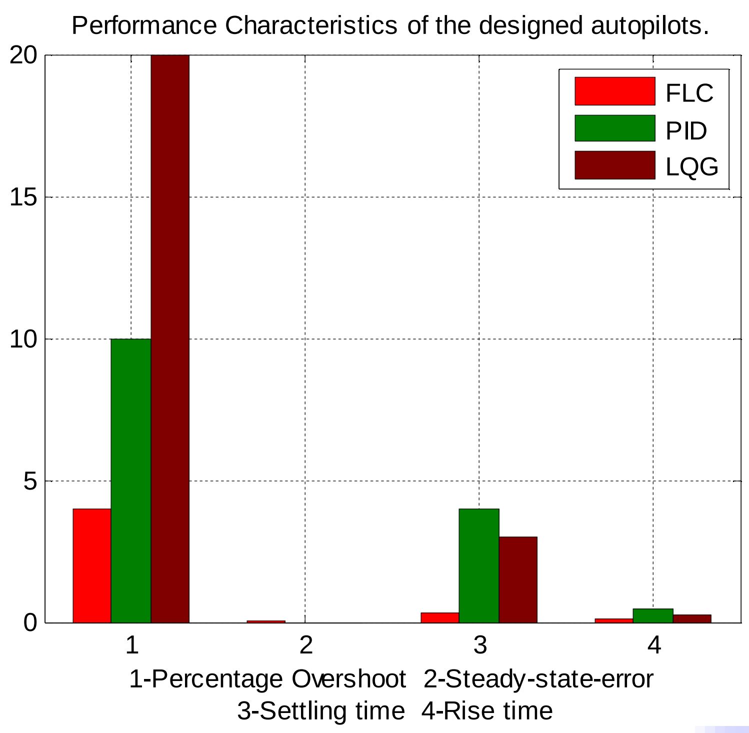


Figure 7.1: Performance characteristic of LQG, PID and FL autopilots.

The nominal performance evaluation provides several observations, which collectively indicate that the FL design does exhibit performance consistent with the expectations established.

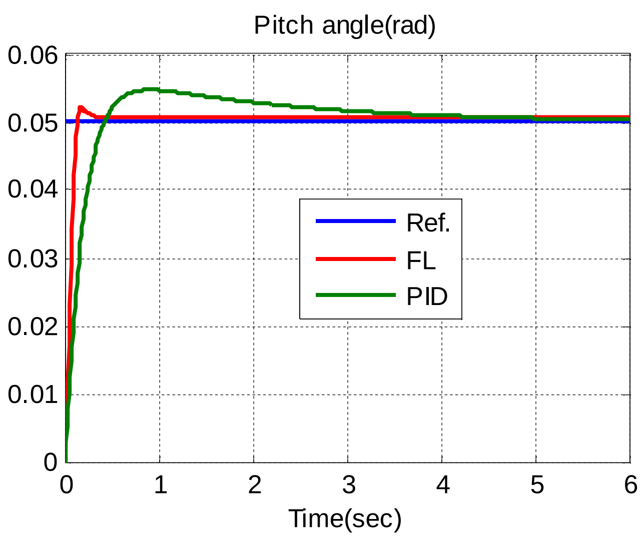


Figure 7.2: Comparison plot of PID and FL autopilot set point command tracking.

It could be clearly noted that the FLC exhibits a performance which not only could match the benchmark classical (PID) design but outperform it, in the area of *settling* and *rise* time. As a matter of fact, it did better than even the modern approach. The only notable drawback with respect to the time response requirement of the FLC synthesised in this thesis is the presence of a sustained steady state error of 1.4%.

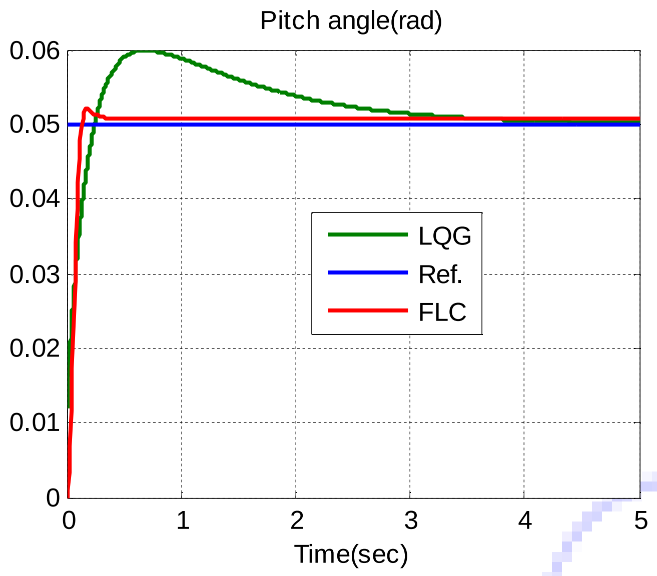


Figure 7.3: Comparison plot of LQG and FL autopilot set point command tracking.

7.1.2 Robust Performance comparison

The LQG autopilot design has a proven stand of stability when uncertainty plagues the system, as could be seen in Fig. 4.13, when uncertainties of process noise of 0.0326 PSD and sensor noise of 0.1 PSD was applied. This resulted in the following performance characteristics, $T_s = 3s$, $T_r = 0.25s$ and Percent Overshoot of 27.4% and the trajectory is smooth-most preferred. Although the PID autopilot presented in this work was proven to tolerate uncertainty too. With *PSD* of 0.0326 applied, as presented in Fig. 5.16, stability of the system was maintained, though the trajectory is not smooth. But when *PSD* was increased to 0.1, massive excursion from the desire pitch angle was seen along the trajectory as shown in Fig. 5.17 just within 2s. This is not a desirable phenomenon of any control system especially in the aerospace field as it could be seen that the perturbed trajectory is no longer smooth. On the other hand, FL autopilot was also simulated for the same values of disturbance. First with 0.0326 *PSD*, which is shown in Fig. 6.21 and with 0.1 *PSD* as shown in Fig.6.22, it could be seen that excursion of the pitch angle beyond the 20% over shoot margin occurred at about 5s. With this, not only did the FL autopilot match the robustness of the PID autopilot but outperformed it. It should also be clearly noted that the perturbed trajectory of pitch angle is also not smooth. To measure up to what is attainable with the LQG autopilot, even from the very beginning of fuzzy sets, criticism was made about the fact that the membership function of a *type-1 fuzzy set* (as implemented in this thesis) has no uncertainty associated with it, something that seems to contradict the word *fuzzy*, since that word has the connotation of lots of uncertainty. So, what does one do when there is uncertainty about the value of the membership function? The answer to this question was provided in 1975 by the inventor of fuzzy sets, Prof. Lotfi A. Zadeh, when he proposed more sophisticated kinds of fuzzy sets, the first of which he called a *type-2 fuzzy set*. A type-2 fuzzy set lets us incorporate uncertainty about the membership function into fuzzy set theory, and is a way to address the above criticism of type-1 fuzzy sets head-on. And, if there is no uncertainty, then a type-2 fuzzy set reduces to a type-1 fuzzy set, which is analogous to probability reducing to determinism when unpredictability vanishes [39]. The concept of the type-2 fuzzy set is an extension of the concept of the type-1 fuzzy set [40]. Thus, implementing the type-2 fuzzy set in this thesis will mean going outside its scope.

7.2 Synthesis approach Comparison

However, it is not enough to compare only the products of the design process. To fully assess FLC as an alternative synthesis process that should be fully embraced in the field of aerospace, the capabilities and limitations of the architectures, classical and modern should be addressed.

FLC autopilot designed incorporates a simple, rule-based IF X AND Y THEN Z approach to solving control problem rather than attempting to model a system mathematically. The modern and classical approach are computationally demanding and requires physical intuition in most cases, whereas the FL model is empirically-based, relying on an operator's experience rather than their technical understanding of the system and sometimes trial-and-error means even pay-off. Though, FL requires some numerical parameters in order to operate such as what is considered significant error and significant rate-of-change-of-error, but exact values of these numbers are usually not critical unless very responsive performance is required in which case empirical tuning would determine them. These values don't have to be symmetrical and can be "tweaked" once the system is operating in order to optimize performance. Generally, FL is so forgiving that the system will probably work the first time without any tweaking.

The design methodology with FL is simpler and faster. Considering the sequence of design steps required to develop the autopilot using the conventional and the fuzzy approach. The particular attribute of the FL approach is described in details in (Kisabo Aliyu., 2009). In designing both the LQG and PID autopilots (conventional approaches) the first step is to understand the physical system and its control requirements. Based on this understanding, the second step is to develop a model which includes the plant, sensors, and actuators. The third step is to use linear-control theory in order to determine a simplified version of the controller, such as the parameters of a proportional-integral-derivative (PID) controller. The fourth step is to develop an algorithm for the simplified controller. The last step is to simulate the design including the effect of noise, nonlinearity and parameters variations. If the performance is not satisfactory we need to modify our system modelling, re-design the controller, re-write the algorithm and retry. Fuzzy logic approach reduces the design process to three steps. Starting with understanding and characterize the system behaviour by using our knowledge and experience. The second step is to directly design the control algorithm using fuzzy rules, which describe the principles of the controller's regulation in terms of the relationship between its inputs and outputs. The last step is to simulate and debug the design. If the performance is not satisfactory we only need to modify some rules and retry. For better appreciation of this fact, Fig. 7.4 illustrates the sequence of design steps required to develop a controller using a conventional and fuzzy logic approach.

Fuzzy Logic reduces the design development cycle. With a fuzzy logic design methodology as enumerated earlier, some time consuming steps are eliminated. Moreover, during the debugging and tuning cycle you can change your system by simply modifying rules, instead of redesigning the controller. In addition, since fuzzy is rule based, you do not need to be an expert in a high or low level language which helps you focus more on your application instead of programming. As a result, Fuzzy Logic substantially reduces the overall development cycle.

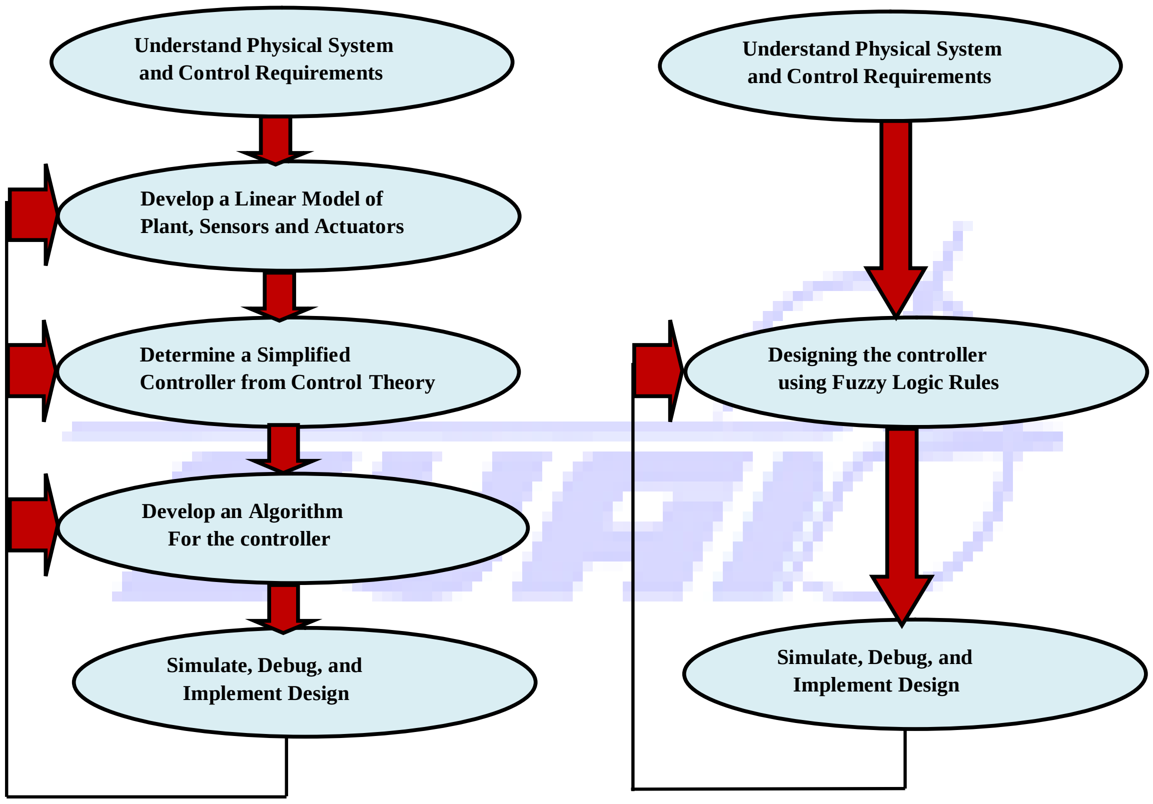

Understand Physical System and Control Requirements
Develop a Linear Model of Plant, Sensors and Actuators
Determine a Simplified Controller from Control Theory
Develop an Algorithm For the controller
Simulate, Debug, and Implement Design
Understand Physical System and Control Requirements
Designing the controller using Fuzzy Logic Rules
Simulate, Debug, and Implement Design

Figure 7.4: Conventional and fuzzy Logic design flow chart.

Fuzzy Logic simplifies design complexity. Fuzzy logic lets us describe the complex systems of the ELV using our knowledge and experience in simple English-like rules. It does not require any system modelling or complex math equations governing the relationship between inputs and outputs. Fuzzy rules are very easy to learn and use, even by non-experts. It typically takes only a few rules to describe systems that may require several of lines of conventional software. As a result, Fuzzy Logic significantly simplifies design complexity. With fuzzy logic we can use rules and membership functions to approximate any continuous function to any degree of precision. We can also add more rules to increase the accuracy of the approximation (similar to a Fourier transform), which yields an improved control performance. Rules are much simpler to implement and much easier to debug

A better alternative to non-linear control. Most real life physical systems are actually non-linear systems. LQG and PID design approaches use different approximation methods to handle non-linearity. A linear approximation technique is relatively simple, however it tends to limit control performance and may be costly to implement in certain applications. Typically, is the case with the microprocessor implementation of Kalman filter associated with LQG control approach.

7.3 FLC Assessment

Both the product and the process substantiate the viability of the FL approach in designing launch vehicle autopilot in pitch plane, specifically for the problem of atmospheric ascent. A comparison of the PID and FL autopilot synthesized in this work reveals that the FLC design algorithm is capable of creating controllers with remarkable nominal performance. This fact is supported by the simulation result in Table 7.1, and also buttresses the fact that 'it has been widely acclaimed and verified that a fuzzy logic controller can achieve a better performance, e.g. a shorter rise-time and a smaller settling-time, over conventional controllers. However, the benefit of FL approach largely arises from the design process itself. During the application of the pitch control problem, the classical method appear to be more manageable in terms of it design parameters and require less iteration than the modern approach. It must also be pointed out that the unnecessary mathematical rigour, preciseness and accuracy involved with the design of both the modern and classical controllers have been a major drawback. This has made it difficult if not impossible for designers, engineers and technology experts to design intelligent complex systems, nonlinear systems with higher order and time-delayed linear systems that can satisfactorily behave as expected while operating in the human-machine interface. Finally, it is also worth noting that fuzzy logic can be blended with conventional control techniques. This means that fuzzy systems do not necessarily replace conventional control methods. It is for these reasons that the FL approach should be viewed as a viable alternative in the development of launch vehicle ascent controls.

As the foundation of *neural networks* and artificial intelligence, the use of fuzzy logic can only expand and our understandings of it develop. It could facilitate human-to-machine speech, accounting for differing accents, tone, pronunciations, and, eventually, even different languages. Fuzzy logic could also be used in systems that are able to analyze literary works and discuss important concepts of those works. Fuzzy systems will play role in a digital and connected world, automating decisions, intelligently analyzing large amounts of data, and learning from their mistakes. Though fuzzy logic has huge potential, we will only get the best of it when it is integrated with computing, artificial intelligence, neural networks, and related areas. In the field of artificial intelligence, neuro-fuzzy refers to combination of artificial neural network and fuzzy logic. Neuro-fuzzy hybridization results in a *hybrid intelligent system* that synergizes these two techniques. The ability of neural networks and fuzzy logic to represent nonlinear systems is most exploited in the synthesis of nonlinear controllers. One of the main advantages of these techniques is that they do not require the exact determination of a system. This enables these techniques to be used for the design of robust controllers. Further, their abilities to adapt make them suitable to be used for adaptive controllers. This special issue focuses on the promise of artificial neural networks and fuzzy logic in the realm of modelling, identification and control of mechatronic systems.

Chapter 8

Conclusions and Recommendations

8.1 Conclusion

A successful application of Modern Control Synthesis approach of LQG was developed to address a generic ELV atmospheric ascent pitch control problem for a wind gust disturbance rejection scheme. Also, Fuzzy Logic Control synthesis to the same expendable launch vehicle has been presented in this thesis to assess its viability as an alternative approach to classical control synthesis methods. Through this investigation, the successful extension of like techniques to a more realistic model of expendable launch vehicles is seen to be plausible.

In this research a Linear Quadratic Gaussian controller was designed for a generic based rigid Expendable Launch Vehicle (ELV). The controller responded to an initial condition disturbance and met the basic control scheme requirement of wind gust rejection. For the optimal estimator (Linear Kalman Filter) implemented, a *Differential Riccati Equation* was solve to realise the Kalman gain, contrary to the conventional *Algebraic Riccati Equation* solution to this problem. Its performance was most desirable, with optimality throughout the regime of control and the Kalman Filter gave an accuracy of 98.8%.

To design the fuzzy logic controller, the pitch control of the highly aerodynamic launch vehicle configuration is designed based on 2 inputs and 1 output. Inputs for this controller are error and rate of change of error, and the launch vehicle output. Fuzzification is where the quantization and membership functions for input variable, error and rate and output variable, launcher output in universe of discourse are defined. It involves the conversion of the input and output signals into a number of fuzzy represented values (fuzzy sets). The knowledge based of a fuzzy logic controller consists of a data based and a rule based. The basic function of the rule based is to represent the expert knowledge in the form of if-then rule structure. The fuzzy logic input variables were defined with a 7 *membership function* which resulted into a 7×7 rule consisting of 49 rules. For this system, max-min composition is used for the Inferencing to obtain the fuzzy set describing the fuzzy value of the overall control output. Defuzzification is a mapping from a space of fuzzy control actions defined over an output universe of discourse into a space of non-fuzzy (crisp) control action. For this system, centroid method is used for defuzzification. Implementation of the FLC was successfully carried out in a Matlab *FIS Editor* and integrating it with a Matlab/Simulink algorithm to realise the FL autopilot.

The comparison of control designs produced by each method revealed that Fuzzy Logic Control synthesis provides an adequate means for developing compensation that meets nominal and robust performance criteria. It has been widely acclaimed and verified that fuzzy logic controller can achieve a better performance e.g. a shorter *rise-time* and smaller *overshoot*, over conventional linear controllers such as the a linear PID controller-this research buttresses this fact. Additionally, a comparison of the two design architectures showed that the physically intuitive nature of the FL problem formulation was an easier task

than trying to manually create the classical controller. It is evident from this research that Fuzzy logic control seems to provide a method of reducing system design complexity while increasing control performance.
These conclusions confirm that the FL approach is an alternative which offers benefits over classical methods.

8.2 Recommendation for Future Work

The research presented in this thesis has several areas which were not pursued in order to narrow its focus. We neglected the effect of actuator, assuming that the actuator acted 'infinitely fast.' This need to be included, as it is an important implementation issue Also, the results presented here were based on a 'point' design for Mach 1.0 point along the trajectory, and additional control designs can be carefully blended together to form a continuous pitch control design for the entire boost phase. Issues surrounding the complexity involved in integrating the various control laws will also be of interest in the final boost phase implementation.

Furthermore, the issue of load relief was also not addressed in the design framework to narrow the focus of the thesis. For the design presented here, the response to wind gust input was minimized. Thus, the controller essentially rejected wind disturbance. However, this is not always advantageous, because large angle-of-attack induced by wind gust can create large structural loads on the launch vehicle. A not uncommon approach is to design a control system which steers into the wind to provide a measure of load relief.

The synthesized LQG control law should be applied to the nonlinear model of the ELV and the result compared with that of the linear one. Hence an *Unscented Kalman Filter (UKF*) which has the advantages of higher accuracy, robustness and less difficult to implement [41] for nonlinear models over the *Extended Kalman Filter (EKF)* based-observer needs to be designed.

To continue a strong argument for a place for the FL approach and its application in aerospace field, it will be necessary to extend the *Type-1 fuzzy logic controller* as synthesized in this research to that of a *type-2 fuzzy logic control*. Also, the PD-type fuzzy logic controller is the most commonly seen format in the area of fuzzy logic control. However, it has been shown to have similarities to the linear PD case and will have a large steady-state error for reference points other than the one which is designed for. Following a conventional solution to such a problem, i.e. the PID controller, it will be beneficial to include the *summation of errors* in the premise of the rule-base to form the so called PID-type fuzzy controller.

Genetic algorithms (GA), which are adopted from the principle of biology evolution, are efficient search technique that manipulates the coding representing a parameter set to reach a near optimal solution. Hence by strengthening fuzzy logic controllers with genetic algorithm the search attainment of fuzzy logic rules and high-performance membership functions will be easier and faster. Also, GA is used to optimize the membership functions of the input and output fuzzy subsets in the FLC, in order to ensure satisfactory closed-loop transient responses. Genetic Algorithms, because of their robustness and ability to provide global solutions, have been used as a tool by a number of researchers to identify parameters of fuzzy logic controller. Since GAs work on coding of the parameter set, and not on the derivative of a function, they are capable of solving a vast range of optimization problems including optimization of the rule set of a fuzzy logic controller.

Fuzzy logic and neural network are two of the most promising research areas in the modern control theory of AI. Recently great efforts have been made towards integrating these two techniques into a unified hybrid or cooperative framework. Fuzzy logic provides a powerful tool to capture the uncertainties associated with human cognitive processes. Neural network, on the other hand, offer great advantages in learning, adaptation, fault tolerance, parallelism and generalization. Independent implementation of fuzzy controllers and neural networks for control systems is not effective enough. A promising approach is to combine them into a hybrid structure in order to obtain better performance [42].

This research work meets the entire research task as stated in section 1.2 of chapter 1.

References

1. Jose M. Giron-Sierra, Guillermo Ortega . 'A survey of stability of Fuzzy logic control with aerospace application.' IFAC 15th Triennial World Congress, Bercelona,2002.
2. Kisabo Aliyu. ' Autopilot for Intelligent Autonomous Aerospace Vehicles .' IFAC Workshop AGNFCS June 30- July 2, 2009, Samara-Russia.
3. Philippe Le Fur. 'An asymptotic approach in the preliminary design of Launch vehicle control system. ' Masters of Science Thesis, Department Aeronautics and Astronautics, MIT, January 1989.
4. http://en.wikipedia.org/wiki/Expendable_launch_system.
5. Greensite A.L.., 'Analysis and design of Space Vehicle Flight Control Systems.' Volume II Spartan Books, New York, 1970.
6. S. Sreeja, Dr. A Dinesh Pai, V. R Lalithambika, K Sivan. 'Launch Vehicle Trajectory Reconstruction Considering Wind Estimation from Flight Data Using Kalman Filter.'
7. Amol A. Khalate, Abdul Naseer, Sheelu Jose and M. V. Dhekane. 'Multi-Objective Control of Aerodynamically Unstable Launcher.' Proceedings of the 15th Mediterranean Conference on Control & Automation, July 27-29, 2007, Athens-Greece
8. Ashish Tewari. 'Atmospheric and Space Flight Dynamics. Modelling and Simulation with MATLAB and Simulink.'
9. Ramnath, R.V. 'A New Approach in the Design of Time Varying Control System with application to space shuttle Boost.' Proc. IFAC congress, Geneva, Italy 1973.
10. Philippe Le Fur. 'An asymptotic approach in the preliminary design of Launch vehicle control system. ' Masters of Science Thesis, Department Aeronautics and Astronautics, MIT, January 1989.
11. Hoelker, R.H. 'Theory of artificial stabilisation of missiles and Space vehicles with exposition of Four Control Principles.' NASA TN D-555,1961.
12. D. Arzelier, D. Peaucelle. 'Robust impulse-to-peak synthesis: Application to the control of an aerospace Launcher'. December 15, 2003
13. Steven T. Karris. 'Introduction to Simulink with Engineering Applications.' Orchard Publications 2006.
14. Robert L. Williams II, Douglas A. Lawrence. 'Linear State-space Control Systems'.2007.
15. Gene F. Franklin, J. David Powell, Abbas Emami- Naeini. 'Feedback Control of Dynamic Systems.' Fifth Edition, NJ, Prentice Hall 2006.
16. Prof. Alexandra Panferov. Lecture Note;'Synthesis of Optimal Systems Stabilization'. St. Petersburg State University of Aerospace Instrumentation, St. Petersburg-Russia, Spring 2010.
17. Roland S. Burns. 'Advanced Control Engineering.' Professor of control engineering Department of Mechanical & Marine Engineering University of Plymouth, UK. 2001.
18. Robert L. Williams II, Douglas A. Lawrence. 'Linear State-space Control Systems'.2007

19. Mohinder S. Grewal, Angus P. Andrews. ‘Kalman Filtering: Theory and Practise Using MATLAB.’ Second Edition. A Wiley- Interscience Publication.
20. Donald Mclean. ‘Automatic Flight Control Systems.’ Professor of Aeronautics University of Southampton, UK. Prentice Hall 1990.
21. Balázs Kulcsár. ‘LQG/LTR Controller design for an Aircraft model’. Department of Control and Transport Automation Budapest University of Technology and Economics, Hungary. November 10, 2000.
22. Aliyu B. Kisabo. ‘Optimal Approach to Aerospace Vehicle Navigation and Motion Control.’ The Scientific Conference of SUAI, Proceedings of the session April 13,2010.
23. Friedland, B. ‘Advance Control System Design.’ NJ,USA: Prentice Hall,1996.
24. João Pedro Hespanha. ‘LQG/LQR Controller Design.’ University of California, Santa Barabara (UCSB), April, 2007.
25. Araki M. ‘Control systems, Robotics, and Automation-Vol. II –PID Control.’ Kyoto University Japan, 2006.
26. Richard C. Dorf, Robeth H. Bishop. ‘Modern Control Systems.’ Addison Wesley Longman, 1998.
27. Derek Atherton. ‘Control Engineering’. Ventus Publishing ApS, 2009.
28. Roberth H. Bishop. ‘Modern Control Systems Analysis and Design Using Matlab And Simulink.’ Pearson, 1997.
29. Morries Driel. ‘Linear Control Systems Engineering.’ Mc Graw-Hill,1995.
30. The MathWorks,Inc. 1984-2010.
31. S.N. Sivanandam, S.Sumathi, S.N. Deepa. ‘Introduction to Fuzzy logic Using Matlab.’pringer 2007.
32. Ismail H. Atlas, Adel M. Sharaf. ‘A generalized direct approach for designing Fuzzy logic Controllers in Matlab/Simulink GUI environment.’ Intrnational Journal of Information Technology and Intelligent computing,Int.J.IT&IC no.4 vol. 1,2007.
33. Paolo Dadone. ‘Design Optimization of Fuzzy Logic Systems.’ PhD Dissertation: Submitted to the Faculty of the Virginia Polytechnic Institute and State University, May 18,2001 Blacksburg, Virginia.
34. Neeraj Gupta, Sanjay K. Jain. ‘Comparative Analysis of Fuzzy Power System Stabilizer Using Different Membership Functions’. International Journal of Computer and Electrical Engineering Vol.2, No.2, April 2010.
35. Nurbaiti Wahid, Mohd. Fua’ad Rahmat, Kamaruzaman Jussoff. ‘Comparative Assesment Using LQR and Fuzzy Logic Controller for a Pitch Control system’. European Journal of Scientific Research ISSN 1450-216X Vol. 42 No. 2 (2010), PP. 184-194.
36. J.-S. Roger Jang, Ned Gulley. ‘MATLAB Fuzzy Logic Toolbox User’s Guide’. Mathworks. Inc. 1997.
37. Jack Kluska. ‘Analytical Methods in Fuzzy Modelling and Control’. Springer,2009.
38. Sheroz Khan, Salami Femi Abdulazeez, Lawal Wahab Adetunji, AHM Zahirul Alam, Momoh Jimoh E. Salami, Shihab Ahmed Hameed, Aisha Hasan Abdalla and Mohd Rafiqul Islam. ‘Design and Implementation of an Optimal Fuzzy Logic Controller

Using Genetic Algorithm'. Journal of Computer Science ISSN 1549-3636. Science Publications,2008.

39. http://en.wikipedia.org/wiki/Type-2_Fuzzy_Sets_and_Systems.
40. Oscar Castillo, Patricia Melin. 'Type-2 Fuzzy Logic: Theory and Applications.' Springer-Verlag Berlin Heidelberg, 2008.
41. Simon J. Julier. Jeffrey K. Uhlmann. 'A New Extension of Kalman Filter to Nonlinear Systems.' The Robotics Research Group, Department of Engineering Science, The University of Oxford.
42. Michail Petrov. 'Fuzzy Logic Controller with Neural network Defuzzifier'. Sofia 13-15 October, INCON, 1997.

Author's Publications

1. Kisabo Aliyu. 'Autopilot for Intelligent Autonomous Aerospace Vehicles.' IFAC Workshop AGNFCS June 30- July 2, 2009, Samara-Russia.
2. Aliyu Bhar Kisabo. 'Modern Approaches to Aerospace Vehicle Navigation and Motion Control System Design.' The Scientific Conference of SUAI, Volume IV Proceedings of the session April 12, 2009.
3. Aliyu B. Kisabo. 'Optimal Approach to Aerospace Vehicle Navigation and Motion Control.' The Scientific Conference of SUAI, Volume IV, Proceedings of the session April 13, 2010.

Appendices

1. M-file for Simulink model of Equation of motion (Fig 2.2)

```
% Input of an Expendable Launch vehicle at the moment of time
 72seconds after launch (frozen time)

 M=1.642e6;%kg
 L_alpha=0.7;
 T_t=3.516e7;
 D=1.363e6;
 T_c=2.634e6;
 miu_c=3.4858;% 1/s^2
 miu_alpha=14.7805;% 1/rad
 g=9.81;% meters per second squared
 V=754.877;%m/sec
 V_w=20;%m/sec % High wind
 teta=0.05;%Radians
 delta=0.05; % 3 degrees
 x0=[0.05;0.017;0]';
```

2. M-file for Dirac's Investigation (Fig 3.2)

```
A=[0 1 0;14.7805 0 0.01958;-100.858 1 -0.1256];
B=[0;3.4858;20.42];
C=[1 0 0;0 1 0;0 0 0];
D=[0];

x0=[0.05;00.17;0];
```

3. M-file for Controllability and Observability investigation

```
% Investigation of Controllability and observability of ELV
  A=[0 1 0;14.7805 0 0.01958;-100.858 1 -0.1256];
  B=[0;3.4858;20.42];
  C=[1 0 0;0 1 0;0 0 0];
  D=[0];


 % Controllability
  c0=ctrb(A,B)
  Rank_Contr=rank(c0)
  if (rank(c0) == size(A,1)) % Logic to assess controllability

  disp('System is controllable.');
  else
  disp('System is NOT controllable.');
  end
  c0_2 = [B A*B A^2*B] % Check c0 via the formula



 % observability
 Ob = obsv(A,C)
 Rank_Obsv=rank(Ob)
 if (rank(Ob) == size(A,1)) % Logic to assess observability
 disp('System is observable.');
 else
```

```
disp('System is NOT observable.');
end

Ob_2 = [C C*A C*A^2] % Check ob via the formula
```

4. Computation for weighing matrix Q and R

$$q_{11}=\frac{1}{\theta^2{}_{\max}\ \deg}=\frac{1}{0.2^2}=2.5\deg^{-2}$$

$$q_{22}=\frac{1}{\dot{\theta}^2{}_{\max}\ \deg}=\frac{1}{0.5^2}=4\deg^{-2}$$

$$q_{33}=\frac{1}{\varphi^2{}_{\max}}=173.56rad^{-2}$$

Where,

$$\varphi=\frac{\dot{h}_{\max}}{v}.57.3=\frac{1m/s}{754.877\,m/s}.57.3=0.075906405\,rad$$

$$r_{11}=\frac{1}{\delta^2{}_{\max}\ \deg}=\frac{1}{3^2}=0.1\deg^{-2} \qquad R=0.1$$

Thus,

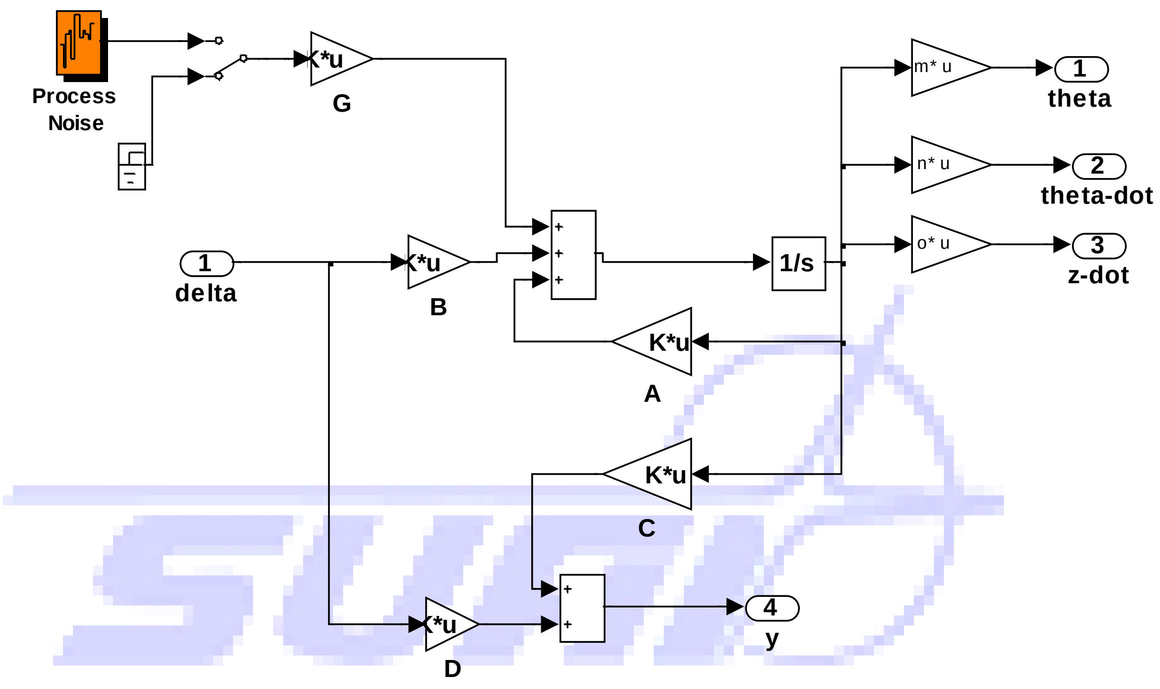
$$Q=\begin{bmatrix} q_{11} & 0 & 0\\ 0 & q_{22} & 0\\ 0 & 0 & q_{33}\end{bmatrix}=\begin{bmatrix} 2.5 & 0 & 0\\ 0 & 4 & 0\\ 0 & 0 & 173.6\end{bmatrix},$$

5. M-file code for LQR autopilot

```
A=[0 1 0;14.7805 0 0.01958;-100.858 1 -0.1256];
B=[0 0;3.4858 14.7805;20.42 -94.8557];
C=[1 0 0;0 1 0;0 0 0];
D=[0 0;0 0;0 0];
% defination of ELV in State-space
sys=ss(A,B,C,D);
r=[0.05;0;0];% Reference signal in radians
% weighing matrices
Q=[2.5 0 0;0 4 0;0 0 173.6];
R=[0.1 0;0 0.1];
% matlab function for computing LQR gain
[K,P,E]=lqr(sys,Q,R)
LQR_gain=K;
v=754.877; % m/sec velocity of ELV
```

```
 %state vector extraction
 m=[1 0 0];
 n=[0 1 0];
 o=[0 0 1];

%Simulink model
LQR_autopilot_act_2
```

6. Simulink model of ELV perturbed by process turbulence



7. Computation for Q_N and R_N

$$Q_N = \frac{2\sigma_w^2 L_T}{v^2} \qquad \text{where, } L_T = 580m \text{ (turbulence scale at height greater than } 580\,m\text{)}$$

and $\sigma_w = 4m$ (mean square value of wind velocity for severe storm)

thus,

$$Q_N = \frac{2\times(4)^2\times 580}{(754.877)^2} = 0.0326$$

$$R_N = \begin{bmatrix} r_{\Delta\theta} & 0 & 0 \\ 0 & r_{\Delta\dot{\theta}} & 0 \\ 0 & 0 & r_{\Delta\dot{z}} \end{bmatrix} \text{ where, } \quad r_{\Delta\theta} = \frac{2\sigma_{\Delta\theta}^2}{\alpha_{\Delta\theta}}, \quad r_{\Delta\dot{\theta}} = \frac{2\sigma_{\Delta\dot{\theta}}^2}{\alpha_{\Delta\dot{\theta}}} \quad \text{and} \quad r_{\Delta\dot{z}} = arbitrary\,.$$

Time constant inverse of filter device; $\alpha = \dfrac{1}{T_\Phi}$ and $T_\Phi \approx 0.1s$

Considering less expensive sensors with the following value of correlative function of error of measuring instrument:

$\sigma_{\Delta\theta} = 0.7rad \quad \sigma_{\Delta\dot{\theta}} = 0.7rad/s$

Thus,

$$r_{\Delta\theta} = r_{\Delta\dot{\theta}} = \frac{2\times(0.7)^2}{10} = 0.098 \approx 0.1$$

Hence,

$$R_N = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}$$

Considering more expensive sensors with the following value of correlative function of: error of measuring instrument:

$\sigma_{\Delta\theta} = 0.1rad \qquad \sigma_{\Delta\dot{\theta}} = 0.1rad/s$

Thus,

$$r_{\Delta\theta} = r_{\Delta\dot{\theta}} = \frac{2\times(0.1)^2}{10} = 0.002$$

Hence,

$$R_N = \begin{bmatrix} 0.002 & 0 & 0 \\ 0 & 0.002 & 0 \\ 0 & 0 & 0.002 \end{bmatrix}$$

$$P_0 = 0.0326 \times 0.002 = 6.52 \times 10^{-5}$$

8. The KF m-file code

```
A=[0 1 0;14.7805 0 0.01958;-100.858 1 -0.1256];
B=[0;3.4858;20.42];
C=[1 0 0;0 1 0;0 0 0];
D=0;


v=754.877; % m/sec

m=[1 0 0];
n=[0 1 0];
o=[0 0 1];

% Modelling Process disturbance & measurement noise for ELV
L=580;%200; % meters, at heights greater than 580m and for
thurnderstorm
sigma_w=4; % meters, for severe storm
G=[0;14.7805;-94.8557];
Q_N=(2*sigma_w^2*L)/v^2;
R_N=diag([0.002 0.002 0.002]); % measurement noise
```

9. M-file code for the realised LQG autopilot

```
 A=[0 1 0;14.7805 0 0.01958;-100.858 1 -0.1256];
B=[0;3.4858;20.42];
C=[1 0 0;0 1 0;0 0 0];
D=0;
%ELV is state-space defination
sys=ss(A,B,C,D);
% Reference signal
r=[0.05;0;0];% 3 deg
%velocity of ELV at 72 sec
v=754.877; % m/sec
% LQR weighing matrices
Q=[2.5 0 0;0 4 0;0 0 173.6]
R=0.1;

[K,P,E]=lqr(sys,Q,R);
LQR_gain=K;
%Extraction of state vectors
m=[1 0 0];
n=[0 1 0];
o=[0 0 1];
% Modelling Process disturbance & measurement noise for ELV
L=580;%200; % meters, at heights greater than 580m and for
thurnderstorm
sigma_w=4; % meters, for severe storm
G=[0;14.7805;-94.8557];
Q_N=0.0163;
%R_N=2*sigma_v^2/
R_N=diag([0.002 0.002 0.002]); % measurement noise
%simulink model
LQG_controlled_ELV
```

10. Matlab codes for open-loop Bode plot of ELV

```
  s=tf('s')
```

```
plant=(3.4858)/(s^2-14.7805)
figure
bode(plant),grid
margin([0 3.4858],[1  0  -14.7805]),grid
```

11. Matlab codes for open-loop Bode plot for PD controller of the ELV

```
 s=tf('s')
Kp=94.2988;
 Kd=5.6910;
 PD=(3.4858*Kd*s+3.4858*Kp)/(s^2-14.7805)
figure
 bode(PD,logspace(0,2)),grid
 margin([19.84 328.7],[1 0 -14.78]),grid
```

12. Matlab codes for open-loop Bode plot for PID controller of the ELV.

```
s=tf('s')

 Kp=111.5;
 Ki=52.2;
 Kd=19.8;

PID=(3.4858*Kd*s^2+3.4858*Kp*s+3.4858*Ki)/(s^3-14.7805*s)
figure
 bode(PID,logspace(0,2)),grid

 margin([69.02  388.7  182],[1  0  -14.7805 0]),grid
```

13. Matlab codes for computing bandwidth frequency of closed-loop PID control of the ELV.

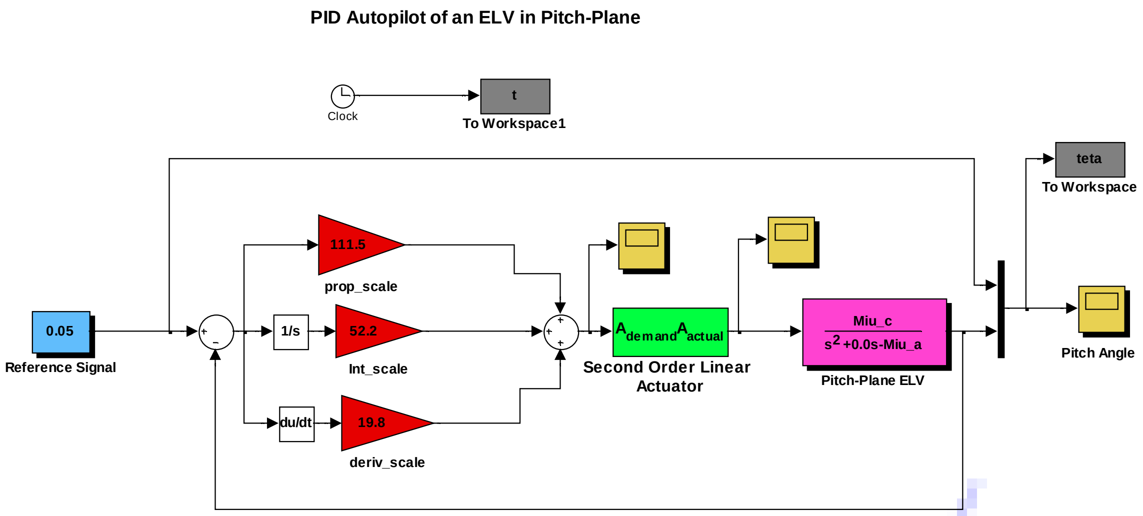
```
s=tf('s')

 Kp=111.5;
 Ki=52.2;
 Kd=19.8;

 PID=(3.4858*Kd*s^2+3.4858*Kp*s+3.4858*Ki)/(s^3+3.4858*Kd*s^2
 +3.4858*Kp*s-14.7805*s+3.4858*Ki)

 Fb=bandwidth(PID,-3)
```

14. Matlab codes for bar chart plot of Performance characteristic of designed autopilots.

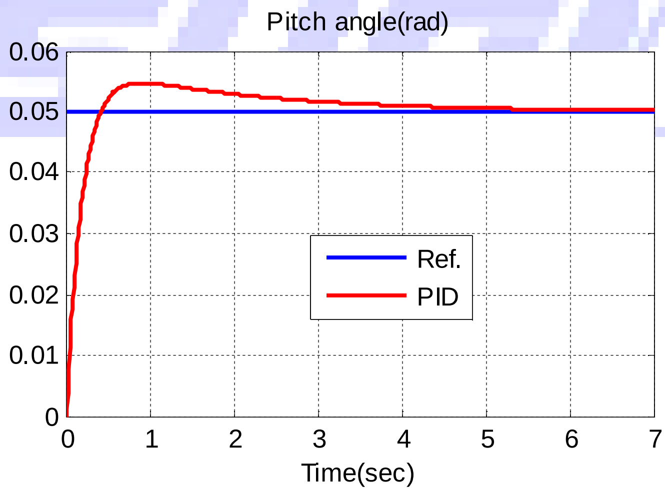
```
Y = [    4         10      20
        0.0007     0       0
        0.3        4.0     3
        0.12       0.45    0.25]


bar(Y,'grouped'),grid
```

15. Simulink model of the PID autopilot with a second order actuator.

Actuator properties; $\begin{array}{l}\omega = 150 rad/\sec \\ \xi = 0.707\end{array}$



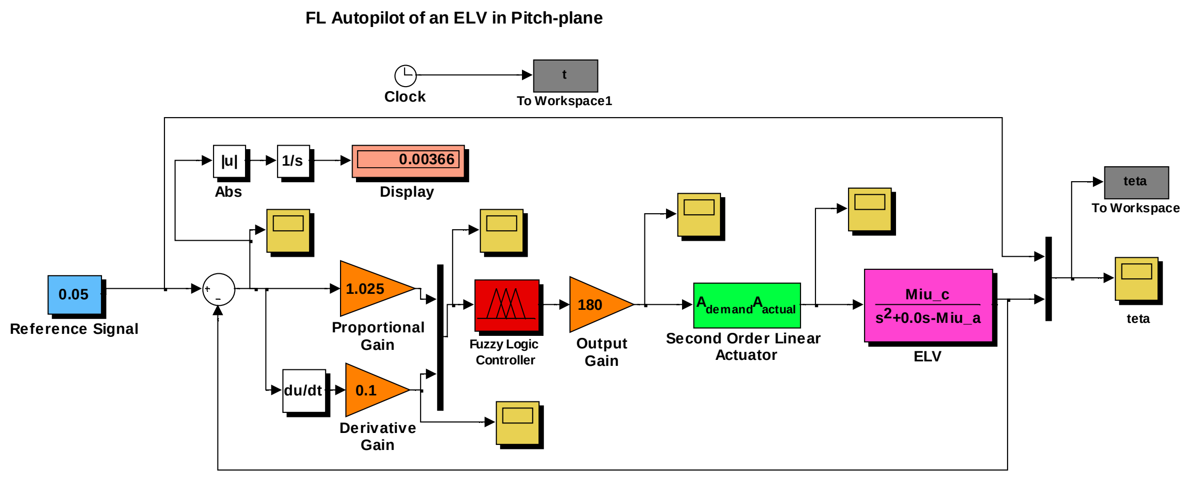
16. Set point command tracking of PID autopilot with actuator.



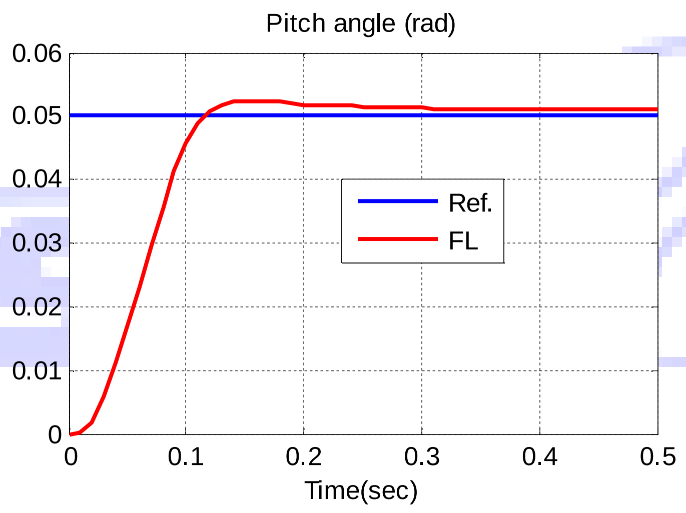
$$T_s = 4s \qquad T_r = 0.4s \qquad PO = 10\% \qquad SSE = 0$$

17. Simulink model of the FLC autopilot with a second order actuator.

Actuator properties; $\begin{array}{l}\omega = 150 Hz \\ \xi = 0.707\end{array}$



18. Set point command tracking of FLC autopilot with actuator.



$$T_s = 0.3s \qquad T_r = 0.12s \qquad PO = 4.8\% \qquad SSE = 0.001$$