ADDIS ABABA UNIVERSITY

ADDIS ABABA INSTITUTE OF TECHNOLOGY

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

Particle Swarm Optimization (PSO) tuned Linear Quadratic Gaussian (LQG) Controller Design for Surface to Air Missile Guidance System

A thesis Submitted to the School of Electrical and Computer Engineering of Addis Ababa Institute of Technology, School of Graduate Studies, Addis Ababa University In partial fulfillment of the Requirement for the Degree of

Masters of Science in Control Engineering

By

Getasew Mekonnen

Advisor: Dereje Shiferaw (Ph.D.)

June, 2018

Addis Ababa, Ethiopia

ADDIS ABABA UNIVERSITY

ADDIS ABABA INSTITUTE OF TECHNOLOGY

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

Particle Swarm Optimization (PSO) tuned Linear Quadratic Gaussian (LQG) Controller Design for Surface to Air Missile Guidance System

By

Getasew Mekonnen

APPROVED BY BOARD OF EXAMINERS

_____	_____
Chairman, Department of Graduate Committee	Signature
Dr. Dereje Shiferaw	_____
Advisor	Signature
_____	_____
Internal Examiner	Signature
_____	_____
External Examiner	Signature

DECLARATION

I, the undersigned, declare that this thesis is my original work, has not been presented for a degree in this or other universities, all sources of materials used for this thesis work have been fully acknowledged.

Name: Getasew Mekonnen

Signature: _____

Place: Addis Ababa Institute of Technology, Addis Ababa University, Addis Ababa, Ethiopia

This thesis has been submitted for examination with my approval as a university advisor

Dereje Shiferaw (Ph.D.) _____

Advisor's Name Signature

ACKNOWLEDGEMENT

I wish to express my sincere appreciation to my advisor**, Dereje Shiferaw (Ph.D.)** for his insight and support throughout the duration of this thesis and my colleague Habtamu Abraham (M.Sc.).

ABSTRACT

Missile guidance system is a well-known nonlinear control engineering area of research. Many technologies have been developed to improve control performance, robustness and to overcome environmental disturbances. This thesis employs a particle swarm optimization (PSO) algorithm to solve the weighting matrices selection problem of linear quadratic Gaussian (LQG) for controlling surface to air Missile guidance system. One of the major challenges in the design of LQG for real time applications is the optimal choice of the state and input weighting matrices (Q and R) respectively, which play a vital role in determining the performance and optimality of the controller. Commonly, trial and error approach is employed for selecting the weighting matrices, which not only burdens the design but also results in non-optimal response. Hence, to choose the elements of Q and R matrices optimally, a PSO algorithm is formulated and applied in the design of linear quadratic regulator (LQR) and linear quadratic Gaussian (LQG) for control of surface to air Missile. It is also intended to produce better robustness with the help of particle swarm optimization (PSO) algorithm as an optimization tool.

Indeed, the system's mathematical model has been developed and also the properties of the uncontrolled system have been analyzed. The model developed shows that the Missile system considered is a 2x2 MIMO (multiple input multiple output) system. Since the system is MIMO, the interaction of the inputs with the outputs has been analyzed using relative gain array (RGA) analysis and frequency domain analysis of the system transfer functions. Also, the condition number of the missile system is calculated and it has small value i.e. "1" which implies there is no control problem for the plant. Then, optimal state feedback controllers have been developed. Here, LQR and LQG controllers are developed. The performance of the controllers designed by manual tuning and PSO-tuning has been analyzed and compared. Besides the performance, the robustness of the controllers developed has been analyzed. The robustness analysis is done by evaluating the singular values of the loop gains using singular value decomposition (SVD) at a certain frequency and for a specified frequency range.

Finally, comparative analysis between the designed controllers is carried out. The proposed PSO tuned design methodology has resulted good Performance i.e. The PSO tuned LQR and LQG controller improved the steady state error and peak response from (0.1043 to 0), (1.043 to 1) and (0.1723 to 0), (1.1723 to 1) m/s^2 respectively in Y direction. In addition, the PSO tuned design has also resulted improvements in robustness of the control systems i.e. The PSO and manually tuned LQG PM is $65.001^0$and 30.3122^0 respectively. Indeed, the loop transfer recovery (LTR) approach is employed at the input to recover the robustness of the manual linear quadratic Gaussian (LQG) controller, which resulted in the improvement of the robustness at the input i.e. the singular value increases or altered from -17.345dB to -5.625dB. The thesis has also suggested further research work in the control of Missile system.

Key words: PSO, LQR, Kalman filter, LQG, LTR, RGA, Surface to air Missile.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

SAM	Surface to Air Missile
AAM	Air to Air Missile
ASM	Air to Surface Missile
SSM	Surface to Surface Missile
LQG	Linear Quadratic Gaussian
LQR	Linear Quadratic Regulator
PSO	Particle Swarm Optimization
RGA	Relative Gain Array
MIMO	Multiple Input Multiple Output
SISO	Single Input Single Output
SVD	Singular Value Decomposition
AI	Artificial Intelligence
GM	Gain Margin
PM	Phase Margin
LTR	Loop Transfer Recovery
KF	Kalman Filter
VLSAM	Vertical Launch Surface to Air Missile

CHAPTER ONE

1. INTRODUCTION

1.1 General

A missile can be defined as any object that can be thrown, projected or propelled toward a target [1]. In other words, a missile is a projectile carrying a payload (usually a warhead) which is guided onto a target by manual or automatic means [2]. Obviously, it is primarily used as a weapon in order to give damage to the target.

Missiles can be classified into different categories. Depending on how they are oriented toward the target, the following are two major categories [1]:

- Unguided missiles, or strategic missiles
- Guided missiles (also called guided munitions), or tactical missiles

Unguided Missiles: Unguided missiles, which includes ballistic missiles, follow the natural laws of motion under gravity to establish a ballistic trajectory [1]. Unguided missiles, whether initially or continuously propelled, can be oriented toward their targets only before they are fired. After firing, they get completely out of control. Therefore, they can be used with acceptable effectiveness only for short distance and stationary targets; because, for moving targets or for targets at longer distances, the hitting accuracy drops more and more due to various reasons such as aiming errors, crosswinds, curvature and rotation of the Earth, etc. The unguided missiles are especially ineffective against moving targets unless such targets are particularly close and slowly moving [2].

Guided Missiles: In the guided class of missiles belong the aerodynamic guided missiles. That is, those missiles that use aerodynamic lift to control its direction of flight. An aerodynamic guided missile can be defined as an aerospace vehicle, with varying guidance capabilities, that is self-propelled through the atmosphere for the purpose of inflicting damage on a designated target. Stated another way, an aerodynamic guided missile is one that has a winged configuration and is usually fired in a direction approximately towards a designated target and subsequently receives steering commands from the ground guidance system (or its own, onboard guidance, system) to improve its accuracy [1].

Typically, guided missiles are homing missiles, which include the following: (1) a propulsion system, (2) a warhead section, (3) a guidance system, and (4) one or more sensors (e.g., radar, infrared, electro optical, lasers). Movable control surfaces are deflected by commands from the guidance system in order to direct the missile inflight; that is, the guidance system will place the missile on the proper trajectory to intercept the target through the controller [1].

Depending on their method of launching, the missiles can be divided into four subsets [3]:

1. Surface-to-Surface Missiles (SSM)
2. Surface-to-Air Missiles (SAM)
3. Air-to-Air Missiles (AAM)
4. Air-to-Surface Missiles (ASM)

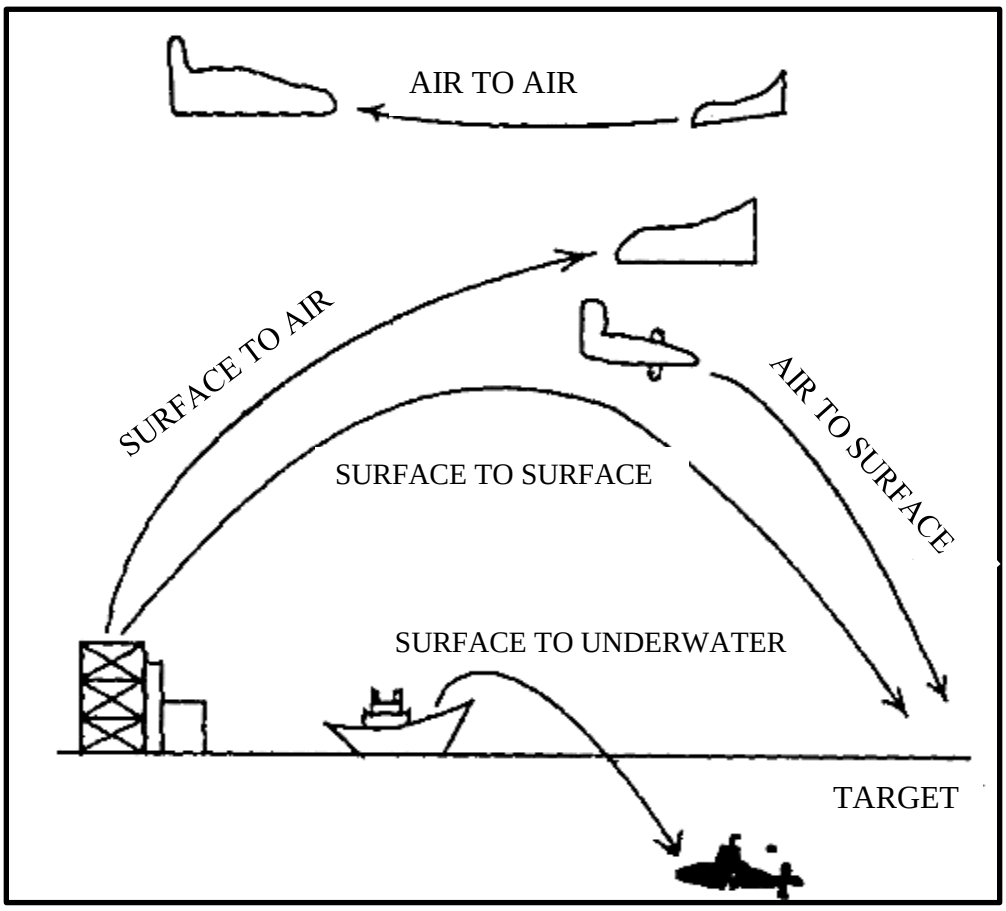


Fig.1. 1: Missile classification by method of launching.

1. SSM: - are common ground-to-ground ones though these may also be launched from a ship to another ship. Underwater weapons which are launched from a submarine also come under this class of missiles.

2. SAM: - are essential complement of modern air defense systems along with anti-aircraft guns which are used against hostile aircraft. It launches from the ground to attack the enemy in the air.

3. AAM: - are for airborne battle among fighter/bomber aircraft. These are usually mounted under the wings or fuselage of the aircraft and are fired at enemy airborne targets by the pilot through the press of a button. In his decision to launch a missile at a particular moment, the pilot is aided by a computer and radar network onboard as well as from ground- based data link.

4. ASM: - is thrown from an air platform such as an aircraft or helicopter toward a surface target.

Regarding the guided missiles, the guidance and control problem involves three sequential stages

- Dynamic Modeling
- Guidance
- Control

In the dynamic modeling stage, the missile is modeled so as to get the relationships among the selected input and the output variables. Then, the guidance algorithm is developed in order to guide the missile toward the intended target for an expected interception. Once the guidance algorithm is constructed, the next step is to design a control system based on the dynamic model of the missile so that it obeys the command signals generated by the guidance unit. The last stage is the estimation of the kinematic parameters of the target. For this task, it is conventional to use a state estimator algorithm such as Kalman filter or fading memory filter.

In order to design a control algorithm, it is a primary requirement to obtain the relationships among the forces/moments acting on the missile and the kinematic state, i.e., position, velocity and acceleration of the missile. The external forces and moments acting on a missile are those generated by the aerodynamic effects including control surfaces, the propulsion including control thrusters and the gravity. As the results of these effects, the components of the position vector of the missile along the downrange, cross range and altitude directions change as well as the yaw, pitch and roll attitudes.

1.1.1 Missile Guidance

Applied to missiles, as guided objects, the goal of guidance is to intercept a target. It means that at a certain moment of time a missile position should coincide with a target position and a target velocity should be sufficient to destroy a target. The goal of guidance, expressed mathematically precisely or by using a "humanitarian language," should be supported by an adequate rule that is able to realize this goal [4].

Among external factors influencing an object's behavior, the target information is the most important one. It has been pointed out that the two basic categories of targets are moving and stationary targets. Missile targets are classified into two broad classes: air targets, usually aircraft or other missiles; and surface targets, which include ships and various objects on the ground. To be destroyed the targets must be detected, identified, and tracked by the missile or associated equipment. All guided missiles launched to engage moving targets use units that observe or sense the target. The point of observation may vary. It can be observed from the missile or a station outside the missile. In this thesis I used homing guidance system i.e. the target is observed from the missile [4].

Homing Guidance: - Homing guidance is used to describe a missile system that can sense the target by some means, and then guide itself to the target by sending commands to its own control surfaces. Homing systems may be classified in three general groups as follows [1]:

1. Active homing system: - the target is illuminated and tracked by equipment on board the missile itself (Fig.1.2, top). That is, the missile carries the source of radiation on board in addition to the radiation sensor. In an active radar homing system, for example, both the radar transmitter and the receiver are contained within the missile. Actively guided missiles have the advantage of launch-and-leave; i.e., they can be launched and forgotten. Disadvantages of the active homing system are additional weight, higher cost, and susceptibility to jamming, since the radiation it emits can reveal its presence.

2. A semiactive homing system:-is one that selects and chases a target by following the energy from an external source, such as a tracking radar, reflecting from the target (Fig.1.2, middle). This illuminating radar may be ground-based, ship-borne, or airborne. Semiactive homing requires the target to be continuously illuminated by the external radar at all times during the flight of the missile.

3. A passive homing system (Fig.1.2, bottom), one that is designed to detect the target by means of natural emanations or radiation such as heat waves, light waves, and sound waves. Thus, passive homing guidance systems are based on the use of the characteristic radiation from the target itself as a means of attracting the missile, for example, as in infrared homing systems. In other words, the target acts as a lure.

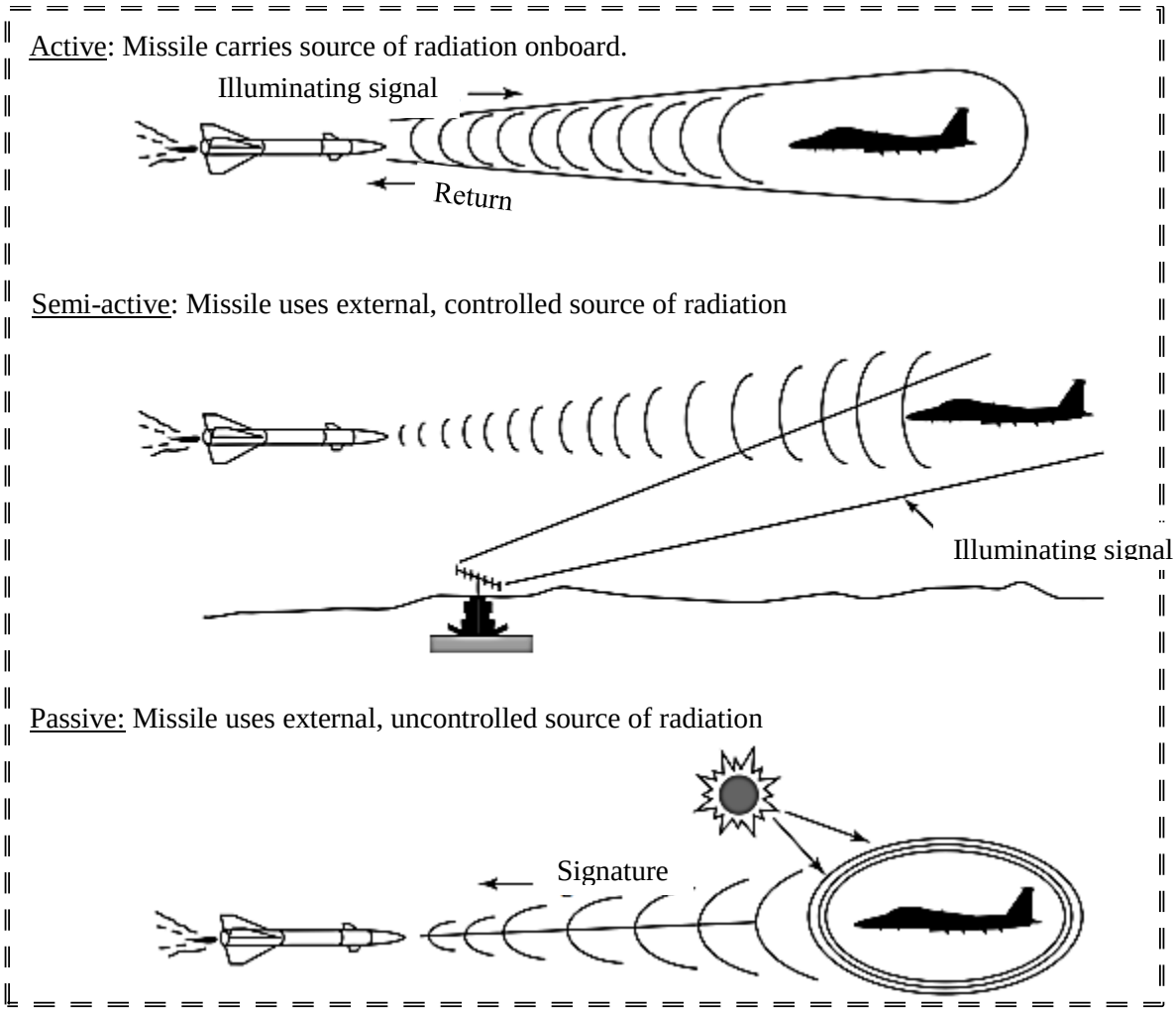


Fig.1. 2: Homing missile guidance types.

1.1.2 Missile Control

In missiles the control function is to ensure stability of the missile and implement the guidance signals received from external sources or generated onboard. The control, after processing the guidance signals, actuates the aerodynamic surfaces on thrust vector to generate turn of the missile speed and direction as required.

The guidance system is to detect whether the missile is flying above or below, to the left or right, of the required path. It obtains these deviations or errors and sends signals to the control system to reduce these errors to zero. The task of the control system therefore is to manoeuvre the missile quickly and efficiently making use of these signals.

In order to appreciate controls we shall briefly describe the motion of the missile as a free body. The missile has a total of six degrees of freedom of movement. Out of this, three degrees are translational or linear about the three axes viz., x, y and z; while the other degrees are rotational movement about three axes termed as pitch, yaw and roll. Pitch is the turn of missile when it climbs up or down. Yaw is its turn to left or right. The roll is when the missile rotates about its longitudinal axis, which is also called roll axis [3].

1.1.3 Control Technologies

One of the most important parts of a missile is the control system, because no matter how sophisticated the guidance system may be or how clever the autopilot is in compensating for the undesirable aerodynamic characteristics, they will be useless if the controls do not generate the required control forces to enable the demanded maneuvers. Traditionally, these control forces have been generated using moveable aerodynamic surfaces; however, increasing demand for more maneuverability has developed other control techniques, such as thrust vector techniques, reaction control system or vernier rockets. In the following sections, aerodynamic control and thrust vector control methods will be explained. But in this, thesis aerodynamic control method is used [5].

Aerodynamic control: **-** This method can be used when the missile is moving in the atmosphere above a certain minimum speed. In this flat aerodynamic surfaces called control surfaces provided on the body of the missile are deflected relatively with respect to the body to generate local differential force leading to a moment acting on the body and resulting in its rotation about a particular axis. Depending on the location of the control surface along the longitudinal &is of the missile, they are termed as canard control (nose end location), moving wing control (middle location), or tail control.

Thrust vector control: - Thrust vector control is a technique whereby the moment required to turn the missile is generated by deflecting the primary thrust from centerline**.** This system is useful when there is not adequate velocity of missile (immediately after launch) or when the vehicle is in low density atmosphere or space.

1.2 Literature Review

Saptarshi Das, Kaushik Halder, "An attitude control system for the canard missile control" [6], has been proposed in this paper as a fusion of linear quadratic Gaussian, loop transfer recovery and linear quadratic tracking is explained. The controller guarantees set point tracking, rejects process

noise and measurement noise but the robustness of the linear quadratic Gaussian controller at the output and the input is not analyzed.

Jimenez, Fernando, "Linear quadratic Gaussian controller design using loop transfer recovery for a flexible missile model" [7], In this study, a Linear Quadratic Gaussian Controller is developed for the model of a tail controlled surface-to-air missile without taking in consideration the uncertainties present in the model. All of the required Time Domain Specifications for the missile autopilot are met by the LQG design. The controller does not saturate the tail deflection actuator rate capabilities nor destabilize the high frequency flexible body modes of the missile. Nominal Performance is met by the LQG design, but Robust Performance and Robust Stability are not met. In short, the LQG controller meets the design goal at the nominal operating condition but it becomes sensitive with parameter variation. Some linear combination of the uncertainties might destabilize the system. To improve the Robust Stability of the system, Loop Transfer Recovery is then applied to the LQG design by the injection of fictitious noise at different uncertainties. All of the required Time Domain Specifications for the missile autopilot are met by the LQGLTR design. The controller does not saturate the tail deflection actuator rate capabilities nor destabilize the high frequency flexible body modes of the missile. Robust Stability is also met but as a consequence of losing Nominal Performance. Robust Performance is not met.

The paper is highly focused on the stability and robustness of the system; however it has not proposed anything about optimization of the control effort of the controller designed.

Raziye tekin, "Design, modeling, guidance and control of a vertical launch surface to air missile"[5], The aims of this study is to obtain a nonlinear model for a generic hybrid controlled VLSAM, and analyze the constructed system under the effect of high angle of attack flight characteristics by 6 DOF simulations and linearization. The dynamic modeling of VLSAM is carried out by implementing the well-known Newton-Euler equations with rigid body assumptions including thrust vector forces and moments.

Missile dynamics are separated into three planes, pitch, yaw and roll after linearization. Controllers are designed as the missile is a skid-to-turn missile. Controllers design is also challenging as the system parameters are changing rapidly. Body rate, angle and acceleration autopilots are designed with classical approaches.

Seid Miad Zandavi, "Surface-to-air missile path planning using genetic and PSO algorithms" [], Optimization algorithms use various mathematical and logical methods to find optimal points. Given the complexity of models and design levels, this paper proposed a heuristic optimization model for surface-to-air missile path planning in order to achieve the maximum range and optimal height based on 3DOF simulation. The proposed optimization model involves design variables based on the pitch programming and initial pitch angle (boost angle). In this optimization model, the paper used genetic and particle swarm optimization (PSO) algorithms. Simulation results indicated that the genetic algorithm was closer to reality but took longer computation time. PSO algorithm offered acceptable results and shorter computation time, so it was found to be more efficient in the surface-to-air missile path planning. This paper focused on path planning using the

PSO algorithm. But in this thesis, I will use PSO algorithm to find optimum values of Q and R parameters of LQG controller.

In this thesis, particle swarm optimization (PSO) algorithm tined Linear Quadratic Gaussian (LQG) Controller for surface to air missile guidance system is designed, thus the control function is to ensure stability of the missile and implement the guidance signals received from external sources or generated onboard. The control, after processing the guidance signals, actuates the aerodynamic surfaces on thrust vector to generate turn of the missile speed, acceleration and direction as required with the presence of the plant disturbance and measurement noises. Therefore, the particle swarm optimization (PSO) tuned system is expected to have a good performance and robustness with minimum possible control effort compared to the conventionally or manually tuned control system.

1.3 Statement of the Problem

One of the major issues of LQG design for real time application is the optimal choice of the state and input weighting matrices (Q and R) respectively, which play a vital role in determining the performance and optimality of the controller. Commonly, trial and error approach is employed for selecting the weighting matrices, which not only burdens the design but also results in non-optimal response. Hence, to choose the elements of Q and R matrices optimally, a particle swarm optimization (PSO) algorithm is formulated and applied for control of surface to air missile guidance system.

Hence, in this thesis the conventional LQG design problem will be reformulated as an optimization problem and will be solved using particle swarm optimization algorithm for the guidance of surface to air missile system.

Therefore, in this thesis observer based state feedback controllers (LQGs) are designed to stabilize the system with optimal control effort. In the design of the LQG the selection of the weighting matrices has been under taken by using an AI technique particle swarm optimization algorithm. In addition the robustness of the control systems will be analyzed.

1.4 Objectives

1.4.1 General objective:

The general objective of this thesis is to design Particle Swarm optimization (PSO) tuned Optimal Linear Quadratic Gaussian (LQG) Controller for a Missile guidance control system; that enables the Missile to successfully hit the target.

1.4.2 Specific objectives:

The specific objectives of these thesis are as follows:

- To design the dynamics of Missile system and to select a suitable model.
- To analyze the interactions of the input-output pairs of the system.
- To design LQG controller by manually tuning the weighing matrices.
- To find the optimized values of the weighting matrices (Q and R) using PSO
- To design PSO tuned LQG controller and manually tuned LQG controller.
- To design LQG/LTR controller, if the designed controller is not robust.

- To compare PSO tuned LQG controller performance with that of manually tuned LQG controller.
- To test performance analysis of the LQG controller with respect to optimality and robustness.

1.5 Methodology

The primary task to meet the general objective stated above is modeling the Missile system. The basic equations defining the airframe dynamics are non-linear, however, since the non-linearity's are "structured" (in the sense that the states are of quadratic form) a novel approach (quadratic state vector approach) of expressing this non-linear dynamics in state space form is done that considers a locally linearized state space model that lends itself to better known linear techniques of the modern control theory (LQG).

The design of the optimal LQG proposed includes finding the optimal values of the weighting matrices i.e. Q and R matrices. These optimal values are found using an intelligent procedure particle swarm optimization. Linear Quadratic Gaussian controller is the main controller that regulates the acceleration of the missile towards the target. LQG has two main parts, The Kalman filter and the Linear Quadratic Regulator (LQR).

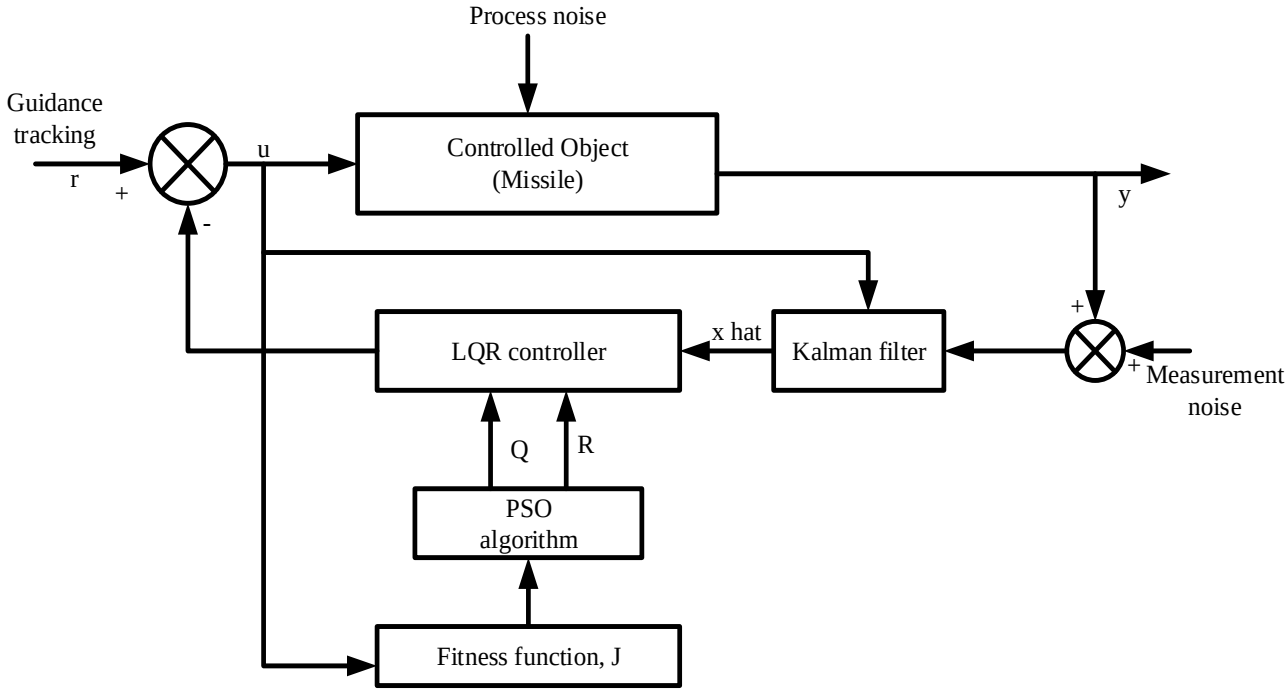


Fig.1. 3: PSO tuned LQG controller for a missile guidance system.

Here in Fig.1.3. it can be seen that Linear Quadratic Gaussian (LQG) controller composed of Kalman Filter (which will estimate all the state of system), followed by Linear Quadratic Regulator (LQR) (which is responsible for controlling the response of system with optimal tuning of Q and R parameters using an intelligent procedure particle swarm optimization) along with control input 'u' process noise is also applied to system. External white Gaussian noise is

added to plant because plant is stochastic with some unknown noise. Measurement noise is also added to system in the output.

The separation principle in linear quadratic gaussian controller designs the kalman estimator and the state feedback gain independently i.e. Kalman filter takes feedback from the output of the system and the input control signal to the Missile system. Then it estimates the states with the presence of process disturbance and measurement noises. After the states are estimated they will be used to control the plant with state feedback gains in the LQR which gives the overall LQG controller.

Thus, in this thesis particle swarm optimization algorithm tuned linear quadratic gausssian controller and manually tuned LQG controller are designed.

After the designed step the two controllers (i.e. PSO tuned LQG and manually tuned LQG), their performances will be compared. The performances to be compared are the time domain specifications such as overshoot, settling time, rise time and steady state error. At the end, Robustness of the two controllers will be analyzed and compared.

1.6 Organization of the Thesis

Chapter one: Reviews the literature on the control of missile guidance system and necessity of particle swarm optimization algorithm to get the optimum values of state control vector (Q) and control effort matrix (R). Methodology and objective along with brief description of the work is presented.

Chapter two: Deals with the mathematical modeling of Missile system. In this chapter Linearization of nonlinear terms is done using a novel approach quadratic state vector technique.

Chapter three: Discusses the general overview of linear quadratic regulator and kalman filter which gives linear quadratic gaussian controller. Indeed the intelligent procedure particle swarm optimization is briefly described. Flow chart and basic steps of PSO algorithm are included. Basic parameter Values of PSO are given that are used to get minimized error value.

Chapter four: Presents the analysis of the MIMO Missile system and the design of LQR and LQG controllers using PSO tuning and manual tuning approach. Then the designed controllers are analyzed and compared. Indeed this chapter presents the analysis of the robustness of the controllers designed using singular value decomposition of return ratio matrix at a specific frequency and within a defined range of frequency. Finally, the robustness of the controllers is compared with each other.

Chapter five: Discusses the conclusions drawn and the recommendation on further work for the best control of the Missile guidance system to achieve the desired target with high precision.

CHAPTER TWO

2. MATHEMATICAL MODELING OF MISSILE SYSTEM

The dynamic equations of motion can be found from Newton's 2nd law of motion for rigid bodies which states that time rate of change of the momentum is equal to the net force applied on the body and time rate of change of the angular momentum is equal to the net moment applied on the body.

2.1 Rigid-Body Equations of Motion

In this section we will consider a typical missile and derive the equations of motion according to Newton's laws. In deriving the rigid-body equations of motion, the following assumptions will be considered:

A. Rigid Body: - A rigid body is an idealized system of particles. Furthermore, it will be assumed that the body does not undergo any change in size or shape.

Translation of the body results in that every line in the body remains parallel to its original position at all times. Consequently, the rigid body can be treated as a particle whose mass is that of the body and is concentrated at the center of mass. In assuming a rigid body, the aeroelastic (is the science which studies the interactions among inertial, elastic, and aerodynamic forces.) effects are not included in the equations. With this assumption, the forces acting between individual elements of mass are eliminated. Furthermore, it allows the airframe motion to be described completely by a translation of the center of gravity and by a rotation about this point. In addition, the airframe is assumed to have a plane of symmetry coinciding with the vertical plane of reference. The vertical plane of reference is the plane defined by the missile x_b- and z_b-axes as shown in Fig. 2.1. The y_b-axis, which is perpendicular to this plane of symmetry, is the principal axis, and the products of inertia I_{xy} and I_{yz} vanish.

B. Aerodynamic Symmetry in Roll: - The aerodynamic forces and moments acting on the vehicle are assumed to be invariant with the roll position of the missile relative to the free-stream velocity vector. Consequently, this assumption greatly simplifies the equations of motion by eliminating the aerodynamic cross coupling terms between the roll motion and the pitch and yaw motions. In addition, a different set of aerodynamic characteristics for the pitch and yaw is not required.

C. Mass: - A constant mass will be assumed, that is, $dm/dt \cong 0$.

In addition, the following assumptions are commonly made:

D. The missile equations of motion are written in the body-axes coordinate frame.

E. A spherical Earth rotating at a constant angular velocity is assumed.

F. The vehicle aerodynamics are nonlinear.

G. The undisturbed atmosphere rotates with the Earth.

H. The winds are defined with respect to the Earth.

I. An inverse-square gravitational law is used for the spherical Earth model.

J. The gradients of the low-frequency winds are small enough to be neglected.

Furthermore, in the present development, it will be assumed that the missile has six degrees of freedom (6-DOF). The six degrees of freedom consist of (1) three translations, and (2) three rotations, along and about the missile (x_b, y_b, z_b) axes. These motions are illustrated in Fig.2.1, the translations being (u, v, w) and the rotations (P, Q, R). In compact form, the translation and rotation of a rigid body may be expressed mathematically by the following equations [1]:

$$\text{Translation: } \sum F = m\mathbf{a}. \tag{2.1}$$

$$\text{Rotations: } \sum \tau = \frac{d}{dt}(\mathbf{r} \times m\mathbf{V}) \tag{2.2}$$

Where, $\sum \tau$ is the net torque on the system.

2.2 Notation and Convention

Conventions and notations for vehicle body axes systems as well as the forces, moments and other quantities are shown in Fig.2.1 and defined in (Table 2.1).

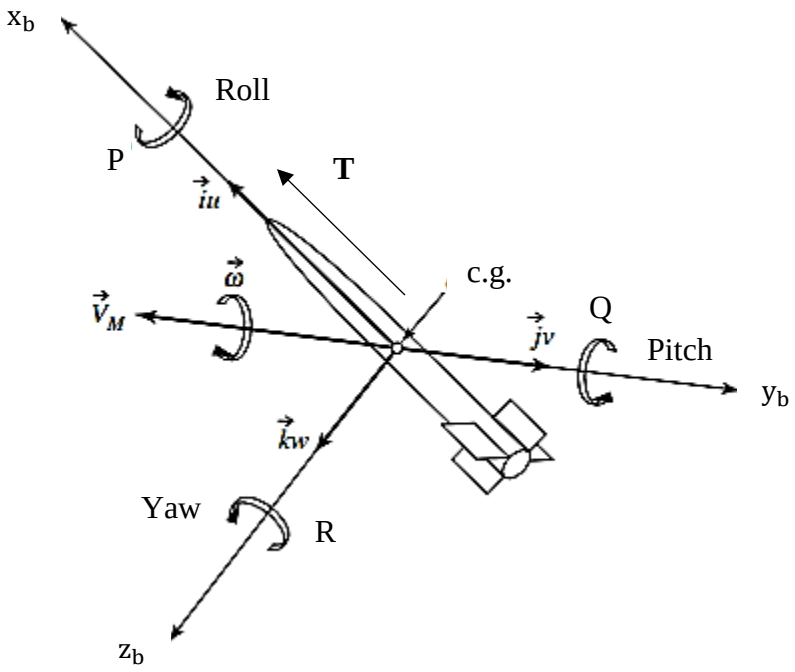


Fig.2. 1: Representation of the missile's six degrees of freedom [1].

The variables shown in Fig.2.1 are defined as:

M - Mass of a vehicle.

T – Thrust.

Table 2. 1: Motion variables

Vehicle Body Axes System	Roll axis x	Pitch axis y	Yaw axis z
Angular rates	p	q	r
Component of vehicle linear velocity along each axis	u	v	w
Component of aerodynamic forces acting on vehicle along each axis	X	Y	Z
Moments acting on vehicle about each axis	L	M	N
Moments of inertia about each axis	$\boldsymbol{I}_{xx}$	$\boldsymbol{I}_{yy}$	$\boldsymbol{I}_{zz}$
Products of inertia	$\boldsymbol{I}_{yz}$	$\boldsymbol{I}_{zx}$	$\boldsymbol{I}_{xy}$
Longitudinal and lateral acceleration	$\boldsymbol{a}_x$	$\boldsymbol{a}_y$	$\boldsymbol{a}_z$
Euler angles	ϕ	θ	Ψ
Gravity along each axis			g
Vehicle thrust along the body axis	T		

2.3 Euler's Equations of Motion

2.31 Translational and Rotational Dynamics

The linear velocity of the missile **V** can be broken up into components u, v, and w along the missile **(**x_b, y_b, z_b) body axes, respectively. Mathematically, we can write the missile vector velocity,V_m in terms of the components as:

$$V_m = u\mathbf{i} + v\mathbf{j} + w\mathbf{k}, \tag{2.3}$$

Where, (i, j, k) are the unit vectors along the respective missile body axes. These components are illustrated in Fig.2.1.

In a similar manner, the missile's angular velocity vector **ω** can be broken up into the components P, Q, and R about the **(**x_b, y_b, z_b) axes, respectively, as follows:

$$\omega = P\mathbf{i} + Q\mathbf{j} + R\mathbf{k}, \tag{2.4}$$

Where, P is the roll rate, Q is the pitch rate, and R is the yaw rate. Therefore, these linear and rotational velocity components constitute the 6-DOF of the missile. As stated in the beginning of this section, the rigid-body equations of motion are obtained from Newton's second law, which states that the summation of all external forces acting on a body is equal to the time rate of the momentum of the body, and the summation of the external moments acting on the body is equal to the time rate of change of moment of momentum (angular momentum). Specifically, Newton's laws of motion were formulated for a single particle. Assuming that the mass m of the particle is multiplied by its velocity V, then the product

$$\mathbf{P} = m\mathbf{V} \tag{2.5}$$

is called the linear momentum. Thus, the linear momentum is a vector quantity having the same direction and sense as **V**. For a system of ***n*** particles, the linear momentum is the summation of the linear momenta of all particles in the system.

$$\mathbf{P} = \sum_{\mathbf{i=1}}^{\mathbf{n}} (m_i \mathbf{V_i}) = m_1 V_1 + m_2 \mathbf{V_2} + + m_n V_n \tag{2.6}$$

Where, ***i*** denotes the ***i***th particle, and ***n*** denotes the number of particles in the system. Note that the time rates of change of linear and angular momentum are referred to an absolute or inertial reference frame. For many problems of interest in airplane and missile dynamics, an axis system fixed to the Earth can be used as an inertial reference frame. Mathematically, Newton's second law can be expressed in terms of conservation of both linear and angular momentum by the following vector equations.

$$\sum \mathbf{F} = \frac{d(\mathbf{m}\mathbf{V}_m)}{dt}\Big]_I \tag{2.7}$$

$$\sum \mathbf{M} = \frac{d\mathbf{H}}{dt}\Big]_I \tag{2.8}$$

Where, ***m*** is the mass, **H** the angular momentum, and the **sy**mbol $]_I$, indicates the time rate of change of the vector with respect to inertial space. Note that Eqn. (2.7) is simply

$$\mathbf{F} = \frac{d\mathbf{p}}{dt} \tag{2.9}$$

or

$$\mathbf{F} = m\left(\frac{dV}{dt}\right) = m\mathbf{a}. \tag{2.10}$$

Equations (2.7) and (2.8) can be rewritten in scalar form, consisting of three force equations and three moment equations as follows:

$$F_x = \frac{d(mu)}{dt}, \quad F_y = \frac{d(mv)}{dt}, \quad F_z = \frac{d(mw)}{dt} \tag{2.11}$$

Where, F_x, F_y, F_z and *u, v, w* are the components of the force and velocity along the missile's x_b, y_b and z_b axes, respectively. Normally, these force components are composed of contributions due to (1) aerodynamic, (2) propulsive, and (3) gravitational forces acting on the missile. In a similar manner, the moment equations can be expressed as follows:

$$L = \frac{dH_x}{dt}, M = \frac{dH_y}{dt}, N = \frac{dH_z}{dt} \tag{2.12}$$

Where, L, M, N are the roll moment, pitch moment, and yaw moment, respectively, and H_x, H_y, and H_z are the components of the moment of momentum along the body x, y, and z axes, respectively.

Let us summarize the various forces, moments, and axes used in developing the missile 6-DOF equations of motion.

Force: $F = F_x\mathbf{i} + F_y\mathbf{j} + F_z\mathbf{k}$, (2.13)

Velocity: $V = u\mathbf{i} + v\mathbf{j} + w\mathbf{k}$, (2.14)

Where, u, v, w are the linear velocity components along the (x, y, z) axes, respectively.

Moment of External Forces: $\sum M = L\mathbf{i} + M\mathbf{j} + N\mathbf{k}$, (2.15)

Where, L is the rolling moment, M is the pitching moment, and N is the yawing moment.

Angular Momentum: $H = H_x\mathbf{i} + H_y\mathbf{j} + H_z\mathbf{k}$, (2.16)

Where, H_x , H_y , H_z are the components of the angular momentum along the (x, y, z) axes, respectively.

Angular Velocity: $\omega = P\mathbf{i} + Q\mathbf{j} + R\mathbf{k}$, (2.17)

Where, **P** is the roll rate, **Q** is the pitch rate, and **R** is the yaw rate. (***i***=unit vector along the x-axis**,** ***j***=unit vector along the y-axis, and ***k***=unit vector along the z-axis).

We now wish to develop an expression for the time rate of change of the velocity vector with respect to the Earth. Before we do this, we note that in general, a vector **A** can be transformed from a fixed (e.g., inertial) to a rotating coordinate system by the relation:

$$\left(\frac{d\mathbf{A}}{dt}\right)_{fixed(X',Y',Z')} = \left[\frac{dA}{dt}\right]_{rot.(X,Y,Z)} + \omega \times A, \quad (2.18a)$$

or

$$\left(\frac{d\mathbf{A}}{dt}\right)_{inertial} = \left[\frac{dA}{dt}\right]_{body} + \omega \times V_M, \quad (2.18b)$$

Where, **ω** is the angular velocity of the missile body coordinate system (x, y, z) relative to the fixed (inertial) system (x', y', z'), and × denotes the vector cross product. Normally, the missile's linear velocity V_M is expressed in the Earth-fixed axis system, so that Eqn. (2.18a) can be written in the form

$$\left(\frac{dV_M}{dt}\right)_E = \left[\frac{dV_M}{dt}\right]_{rot.coord} + \omega \times V_M, \quad (2.19)$$

Where, **ω** is the total angular velocity vector of the missile with respect to the Earth. In terms of the body axes, we can write the force equation in the form

$$F = m\left[\frac{dV_M}{dt}\right]_{body} + m(\omega \times V_M). \quad (2.20)$$

The first part on the right-hand side of Eqn.(2.20) can be written as

$$\left(\frac{dV_M}{dt}\right)_{rot.coord} = \left(\frac{du}{dt}\right)\mathbf{i} + \left(\frac{dv}{dt}\right)\mathbf{j} + \left(\frac{dw}{dt}\right)\mathbf{k} \quad (2.21)$$

Where, (du/dt) = forward (or longitudinal) acceleration,

(dv/dt) = right wing (or lateral) acceleration,

(dw/dt) = downward (or vertical) acceleration,

and the vector cross product as

$$\omega \times V_M = \begin{bmatrix} i & j & k \\ P & Q & R \\ u & v & w \end{bmatrix} = (wQ - vR)\mathbf{i}+(uR - wP)\mathbf{j}+(vP-uQ)\mathbf{k} \quad (2.22)$$

Next, from Eqn. (2.7) we can write the sum of the forces as

$$\sum \Delta F = \sum \Delta F_x \; \boldsymbol{i} + \sum \Delta F_y \; \boldsymbol{j} + \sum \Delta F_z \; \boldsymbol{k} \quad (2.23)$$

Equating the components of Eqns. (2.21), (2.22), and (2.23) yields the missile's linear equations of motion. Thus, for a missile with an $(x_b - z_b)$ plane of symmetry (see rigid-body assumption A) we have:

$$\sum \Delta F_x = m(\dot{u} + wQ - vR) = X+T \quad (2.24)$$

$$\sum \Delta F_y = m(\dot{v} + uR - wP)=Y \quad (2.25)$$

$$\sum \Delta F_z = m(\dot{w} - uQ + vp) = Z + mg \quad (2.26)$$

From Eqn. (2.8) we can obtain in a similar manner the equations of angular motion. However, before we develop these equations, an expression for **H** is needed.

In general, the moment about an arbitrary point O (center of mass) of the momentum, **p** = m**V** of a particle is [8].

$$\mathbf{H} = \mathbf{r} \times m\mathbf{V}=m\mathbf{r} \times (\boldsymbol{\omega}\times\mathbf{r}) \quad (2.27)$$

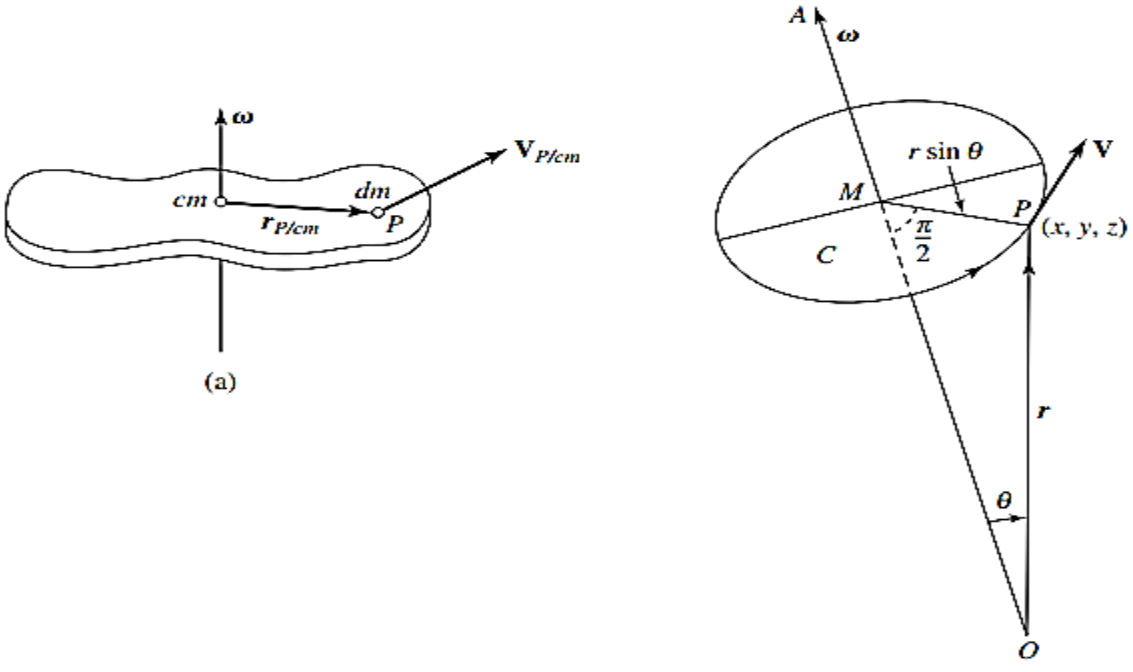


Fig.2. 2: General rigid body with angular velocity vector **ω** about its center of mass [1].

The moment of momentum (or angular momentum) of the entire body about O (at the center) is therefore,

$$\mathbf{H} = \sum \mathbf{r} \times m\mathbf{V} = \sum m\mathbf{r} \times (\boldsymbol{\omega} \times \mathbf{r}) = \sum m[\boldsymbol{\omega}(\mathbf{r}.\mathbf{r}) - \mathbf{r}(\mathbf{r}.\boldsymbol{\omega})]$$

(Note that this result was obtained using the formula a × (b×c) = (a· c)b – (a·b)c). This equation can also be written as

$$\mathbf{H} = (\sum m\mathbf{r}^2)\boldsymbol{\omega} - \sum m\mathbf{r}(\mathbf{r}.\boldsymbol{\omega})$$

From Fig.2.3a, the total moment of momentum can be written as

$$\mathbf{H} = \sum \delta\mathbf{H} = \sum(\mathbf{r} \times \mathbf{V}_{cm})dm + \sum[\mathbf{r} \times (\boldsymbol{\omega} \times \mathbf{r})]dm \qquad (2.28)$$

Note that the velocity $\mathbf{V}_{cm}$ (the velocity at the center of mass (cm) of the missile) is constant with respect to the summation and can be taken outside the summation sign. Thus,

$$\mathbf{H} = \sum \mathbf{r}dm \times \mathbf{V}_{cm} + \sum[\mathbf{r} \times (\boldsymbol{\omega} \times \mathbf{r})]dm \qquad (2.29a)$$

The first term on the right-hand side of Eqn. (2.29a) is zero. Therefore, we have simply:

$$\mathbf{H} = \sum[\mathbf{r} \times (\boldsymbol{\omega} \times \delta\mathbf{r})]dm \qquad (2.29b)$$

and

$$\mathbf{H} = \int \mathbf{r} \times (\boldsymbol{\omega} \times \mathbf{r})]dm \qquad (2.29c)$$

Performing the vector operations in Eqn. (2.29c) and noting that

$$\boldsymbol{\omega} = P\boldsymbol{i} + Q\boldsymbol{j} + R\boldsymbol{k}$$

$$\mathbf{r} = x\boldsymbol{i} + y\boldsymbol{j} + z\boldsymbol{k}$$

We have,

$$\boldsymbol{\omega} \times \mathbf{r} = \begin{bmatrix} \boldsymbol{i} & \boldsymbol{j} & \boldsymbol{k} \\ P & Q & R \\ x & y & z \end{bmatrix}$$

$$= (zQ - yR)\mathbf{i} + (xR - zP)\mathbf{j} + (yP - xQ)\mathbf{k}. \qquad (2.30a)$$

Finally,

$$\mathbf{r} \times (\boldsymbol{\omega} \times \mathbf{r}) = \begin{bmatrix} \boldsymbol{i} & \boldsymbol{j} & \boldsymbol{k} \\ x & y & z \\ (zQ - yR) & (xR - zP) & (yP - xQ) \end{bmatrix}$$

$$= \mathbf{i}[(y^2 + z^2)P - xyQ - xzR] + \mathbf{j}[(z^2 + x^2)Q - yzR - xyP] + \mathbf{k}[(x^2 + y^2)R - xzP - yzQ]. \qquad (2.30b)$$

Substituting Eqn. (2.30b) into Eqn. (2.29c), we have

$$\mathbf{H} = \int \mathbf{i}[(y^2 + z^2)P - xyQ - xzR]dm + \int \mathbf{j}[(z^2 + x^2)Q - yzR - xyP]dm +$$

$$\int \mathbf{k}[(x^2 + y^2)R - xzP - yzQ]dm, \qquad (2.30c)$$

Where the $\int(y^2+z^2)$ is defined as the moment of inertia, $\boldsymbol{I}_x$, and $\int xydm$ is defined as the product of inertia, $\boldsymbol{I}_{xy}$. The remaining integrals in Eqn. (2.38c) are similarly defined. By proper positioning of the body axis system, one can make the products of inertia $\boldsymbol{I}_{xy} = \boldsymbol{I}_{yz}$ equal to 0. This will be true if we can assume that the x-y plane is a plane of symmetry of the missile. Consequently, Eqn. (2.30c) can be rewritten in component form as follows:

$$\mathrm{H_x} = \mathrm{P}\int(\mathrm{y}^2+\mathrm{z}^2)\,\mathrm{dm} - \mathrm{R}\int \mathrm{xzdm} = \mathrm{P}\mathbf{I}_\mathrm{x} - \mathrm{R}\mathbf{I}_\mathrm{xz}, \tag{2.31a}$$

$$\mathrm{H_y} = \mathrm{Q}\,\mathrm{P}\int(\mathrm{x}^2+\mathrm{z}^2)\,\mathrm{dm} = \mathrm{Q}\mathbf{I}_\mathrm{y}, \tag{2.31b}$$

$$\mathrm{H_z} = \mathrm{R}\int(\mathrm{x}^2+\mathrm{y}^2)\,\mathrm{dm} - \mathrm{P}\int \mathrm{xzdm} = \mathrm{R}\mathbf{I}_\mathrm{z} - \mathrm{P}\mathbf{I}_\mathrm{xz}, \tag{2.31c}$$

From Eqn. (2.8), we note that the time rate of **H** is required. Now, since **H** can change in magnitude and direction, Eqn. (2.8) can be written as

$$\sum \Delta\mathbf{M} = \left(\frac{d\mathbf{H}}{dt}\right) + \boldsymbol{\omega} \times \mathbf{H} \tag{2.32}$$

$d\mathbf{H}/dt$ is,

$$\frac{\mathrm{dH_x}}{\mathrm{dt}} = \left(\frac{\mathrm{dP}}{\mathrm{dt}}\right)\mathbf{I}_\mathrm{x} - \left(\frac{\mathrm{dR}}{\mathrm{dt}}\right)\mathbf{I}_\mathrm{xz}, \tag{2.33a}$$

$$\frac{\mathrm{dH_y}}{\mathrm{dt}} = \left(\frac{\mathrm{dQ}}{\mathrm{dt}}\right)\mathbf{I}_\mathrm{y}, \tag{2.33b}$$

$$\frac{\mathrm{dH_z}}{\mathrm{dt}} = \left(\frac{\mathrm{dR}}{\mathrm{dt}}\right)\mathbf{I}_\mathrm{z} - \left(\frac{\mathrm{dP}}{\mathrm{dt}}\right)\mathbf{I}_\mathrm{xz}, \tag{2.33c}$$

Since initially we assumed a rigid body with constant mass, the time rates of change of the moments and products of inertia are zero. The vector cross product in (2.32) is

$$\boldsymbol{\omega} \times \mathbf{H} = \begin{bmatrix} \boldsymbol{i} & \boldsymbol{j} & \boldsymbol{j} \\ P & Q & R \\ H_x & H_y & H_z \end{bmatrix}$$

$$= \left(\mathrm{QH_z} - \mathrm{RH_y}\right)\mathrm{i} + (\mathrm{RH_x} - \mathrm{PH_z})\mathrm{j} + (\mathrm{PH_y} - \mathrm{QH_x})\mathrm{k} \tag{2.34}$$

We can write an equation for the summation of all moments in the form,

$$\sum \Delta\mathbf{M} = \mathbf{i}\sum \Delta\mathrm{L} + \mathbf{j}\sum \Delta\mathrm{M} + \mathbf{k}\sum \Delta\mathrm{N} \tag{2.35}$$

Equating the components of Eqns. (2.33), (2.34), and (2.35) and substituting for H_x,H_y, and H_z from Eqn. (2.31) yields the angular momentum equations.

$$\sum \Delta \mathbf{L} = \dot{P}\, I_x + QR(I_z - I_y) - (\dot{R} + PQ)I_{xz}, \tag{2.36a}$$

$$\sum \Delta \mathbf{M} = \dot{Q}I_y + (I_x - I_z)PR + (P^2 - R^2)I_{xz}, \tag{2.36b}$$

$$\sum \Delta \mathbf{N} = \dot{R}I_z + (I_y - I_x)PQ - (\dot{P} - QR)I_{xz}, \tag{2.36c}$$

$$Here(.) = is\ the\ derivative\ operator$$

Where, dP/dt is the roll acceleration, dQ/dt is the pitch acceleration, and dR/dt is the yaw acceleration. The set of Eqns. (2.24 - 2.26) and (2.36a – 2.36c) represents the complete 6-DOF missile equations of motion. Specifically, Eqns (2.24 – 2.26) describe the translations, and equations 2.36 describe the rotation of body. The set of Eqns (2.24 – 2.26) and (2.36) are six simultaneous nonlinear equations of motion, with six variables *u*, *v*, *w*, *P*, *Q*, and *R*, which completely describe the behavior of a rigid body [1]. Note thatI_x, I_y, $and\ I_{xz}$ are constant for a given rigid body because of our choice of coordinate axes. Due to the usual symmetry of the missile about the *x*-*y* plane, the products of inertia that involve *y* are usually omitted, and the moment equations rewritten as follows [1], [9].

$$\Delta \mathbf{L} = \dot{P}I_x + QR(I_z - I_y) \tag{2.37a}$$

$$\Delta \mathbf{M} = \dot{Q}I_y + (I_x - I_z)PR \tag{2.37b}$$

$$\Delta \mathbf{N} = \dot{R}I_z + (I_y - I_x)PQ \tag{2.37c}$$

2.4 Control Surfaces

The function of a guided missile's control system, which is an integral part of the guidance system, is to make certain that the missile follows the prescribed trajectory, that is, to detect whether the missile is flying too high or low, or too far to the right or left. The guidance system measures these errors and sends signals to the control system to reduce these errors to zero.

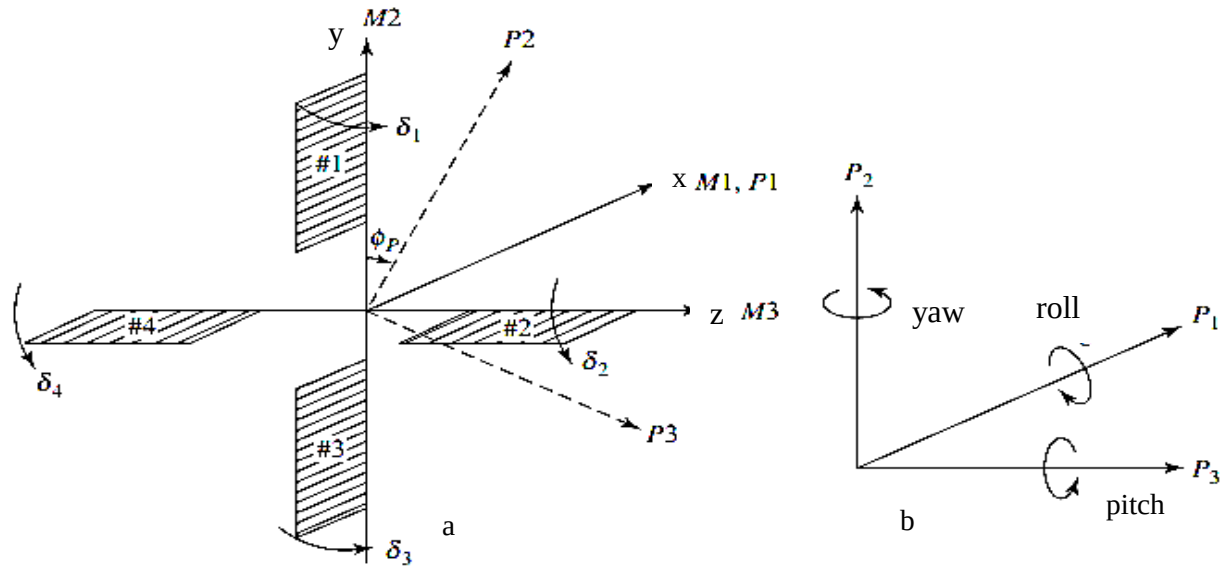


Fig.2. 3: Control surfaces and Fin deflection convention [1].

For the purposes of the present discussion, it will be assumed that the missile is tail-controlled by four fins, which have no downwash interference from the control surfaces. At this point, it is appropriate to define the terms *elevators, rudders, and ailerons*. Commonly, aerodynamically guided missiles have two axes of symmetry that is, arranged in a cruciform configuration. If the missile has four control surfaces as shown in Fig.2.3, then we will define surfaces 2 and 4 as elevators, and 1 and 3 as rudders.

Referring to Fig. 2.3, if 2 and 4 are mechanically linked, then a servo must impart the same rotation to both these surfaces and call elevators. The same argument applies to surfaces 1 and 3, which we call rudders. Furthermore, if surfaces 2 and 4 each have their own servo, they can act as ailerons (i.e., one can move clockwise while the other can move counterclockwise) [1],[10]. The majority of tactical missiles are designed in a cruciform configuration, thus enabling them to maneuver with ease horizontally and vertically. In a cruciform configuration, the two horizontal lifting surfaces are deflected equally by the fin control actuation system. The same concept applies to the vertical surfaces.

In essence, the actuator consists of the control surfaces (or fins) and associated servomechanisms, and is used to change the missile's attitude and trajectory or flight path. Therefore, the function of the four fin actuators is to move the control surfaces in accordance with commands from the three autopilots. The autopilot outputs are virtual fin deflection commands shown in Fig.2.3b. In Fig.2.3b, the roll autopilot is along the P1axis, while the pitch and yaw axes are along the P3 and P2 axes, respectively; the corresponding positive fin deflection commands are indicated by the corresponding δP's. The four real fins are located in the missile or M-frame, which is shown in Fig.2.3a and is rotated from the autopilot axis system (P) by an angle.φ_p

Figure 2.3a defines the control surface convention. Here the control surfaces are numbered as shown and the deflections $(\delta_1, \delta_2, \delta_3, \delta_4)$ are taken to be positive if clockwise, looking outwards along the individual hinge axis. Thus

Aileron deflection**:** $\boldsymbol{\xi} = \frac{1}{4}(\delta_1 + \delta_2 + \delta_3 + \delta_4)$

Elevator deflection**:** $\boldsymbol{\eta} = \frac{1}{2}(\delta_1 + \delta_3)$

Rudder deflection**:** $\boldsymbol{\zeta} = \frac{1}{2}(\delta_2 + \delta_4)$

2.5 Linearization of Missile Dynamics Equation

Equations (2.24 to 2.26) represent the force equations of a generalized rigid body and describe the translational motion of its center of gravity (c.g), since the origin of the vehicle body axes is assumed to be co-located with the body c.g. Separating the derivative terms and after some algebraic manipulation and neglecting the disturbances, Eqns. (2.24 to 2.26) may be written in a vector form as:

$$\frac{d}{dt}\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} uq \\ ur \\ vp \\ vr \\ wp \\ wq \end{bmatrix} + \begin{bmatrix} \overline{X} + \overline{Y} \\ \overline{Y} \\ \overline{Z} \end{bmatrix} \tag{2.38}$$

Where, $\overline{X} = \frac{X}{m}$, $\overline{Y} = \frac{Y}{m}$, $\overline{Z} = \frac{Z}{m}$, $\overline{T} = \frac{T}{m}$.

Note that the states (*u, v, w, p, q, and r)* appear as "quadratic terms (form)''.

Equations (2.37a to 2.37c) represent the moment equations of a generalized rigid body and describe the rotational motion about the body axes through its c.g. separating the derivative terms and after some algebraic manipulation and neglecting the disturbances, Eqns. (2.37a to 2.37c) may be written in a vector form as:

$$\frac{d}{dt}\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 0 & 0 & \overline{I_x} \\ 0 & \overline{I_y} & 0 \\ \overline{I_z} & 0 & 0 \end{bmatrix} \begin{bmatrix} pq \\ pr \\ qr \end{bmatrix} + \begin{bmatrix} \overline{L} \\ \overline{M} \\ \overline{N} \end{bmatrix} \tag{2.39}$$

$$\text{Where, } \overline{I_x} = \frac{\overline{I_y} - \overline{I_z}}{\overline{I_x}}, \quad \overline{I_y} = \frac{\overline{I_z} - \overline{I_x}}{\overline{I_y}}, \quad \overline{I_z} = \frac{\overline{I_x} - \overline{I_y}}{\overline{I_z}}$$

$$\overline{L} = \frac{L}{I_x}, \quad \overline{M} = \frac{M}{I_y}, \quad \overline{N} = \frac{N}{I_z}.$$

The selection of the particular order of the terms in the "quadratic-state vectors"

$[uq\ ur\ vp\ vr\ wp\ wq]^T$of Eqn.(2.38) and $[P^2\ pq\ pr\ q^2\ qr\ r^2]^T$ of Eqn.(2.39) are discussed [6],[11].

Combining Eqns. (2.38) and (2.39), we obtain the full 6^{th} order rigid body dynamics state equations as:

$$\frac{d}{dt}\begin{bmatrix} x_1^{[1]} \\ x_2^{[1]} \end{bmatrix} = \begin{bmatrix} [C] & [0] \\ [0] & [H] \end{bmatrix} \begin{bmatrix} x_1^{[2]} \\ x_2^{[2]} \end{bmatrix} + \begin{bmatrix} [I] & [0] \\ [0] & [I] \end{bmatrix} \begin{bmatrix} u_1^{[1]} \\ u_2^{[1]} \end{bmatrix} \tag{2.40}$$

Where,

$$x_1^{[1]} = [u\ \ v\ \ w]^T,$$

$$x_2^{[1]} = [p\ \ q\ \ r]^T$$

$$x_1^{[2]} = [uq\ ur\ vp\ vr\ wp\ wq]^T,$$

$$x_2^{[2]} = [pq\ \ pr\ qr]^T,$$

$$u_1^{[1]} = \left[\overline{X} + \overline{Y} \;\; \overline{Y} \;\; \overline{Z}\right]^T,$$

$$u_2^{[1]} = \left[\overline{L} \;\; \overline{M} \;\; \overline{N}\right]^T,$$

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

$$[H] = \begin{bmatrix} 0 & 0 & \overline{I_x} \\ 0 & \overline{I_y} & 0 \\ \overline{I_z} & 0 & 0 \end{bmatrix}$$

Equations (2.40) is complete non-linear description of the full 6-DOF autopilot model. In fact, these equations contain quadratic terms in states and will be classed as the quadratic dynamic model. This type of model is required when autopilot design is undertaken for a missile executing high g- or high angle of attack manoeuvres, and (u, v, w, p, q, r) are not small. A more detailed consideration of the algebraic structure of this type of dynamic systems is given in [9], [11].

2.5.1 Linearized model for a two – axis symmetrical airframe

It is assumed that $\overline{X}$, $\overline{Y}$, $\overline{Z}$, $\overline{L}$, $\overline{M}$, and $\overline{N}$ are functions of *u*, *v*, *w*, *p*, *q*, *r*, ξ, η, ζ and neglecting the rolling action because most of the time rolling is not necessary in the missile. Only the lateral directional dynamics control is enough to achieve the target. Using first order linearization about the nominal values$u_0, v_0, w_0, p_0, q_0, r_0,\ \xi_0, \eta_0, and\ \zeta_0$, and defining the aerodynamic derivative as:

$$\overline{X_u} = \frac{\partial \overline{X}}{\partial u},\ \overline{X_v} = \frac{\partial \overline{X}}{\partial v},\ \overline{X_w} = \frac{\partial \overline{X}}{\partial w},\ \overline{X_p} = \frac{\partial \overline{X}}{\partial p},\ \overline{X_q} = \frac{\partial \overline{X}}{\partial q},\ \overline{X_r} = \frac{\partial \overline{X}}{\partial r},$$

$$\overline{X_\eta} = \frac{\partial \overline{X}}{\partial \eta}, \overline{X_\zeta} = \frac{\partial \overline{X}}{\partial \zeta},$$

$$\overline{Y_u} = \frac{\partial \overline{Y}}{\partial u},\ \overline{Y_v} = \frac{\partial \overline{Y}}{\partial v},\ \overline{Y_w} = \frac{\partial \overline{Y}}{\partial w},\ \overline{Y_p} = \frac{\partial \overline{Y}}{\partial p},\ \overline{Y_q} = \frac{\partial \overline{Y}}{\partial q},\ \overline{Y_r} = \frac{\partial \overline{Y}}{\partial r},$$

$$\overline{Y_\eta} = \frac{\partial \overline{y}}{\partial \eta}, \overline{Y_\zeta} = \frac{\partial \overline{Y}}{\partial \zeta}$$

$$\overline{Z_u} = \frac{\partial \overline{Z}}{\partial u},\ \overline{Z_v} = \frac{\partial \overline{Z}}{\partial v},\ \overline{Z_w} = \frac{\partial \overline{Z}}{\partial w},\ \overline{Z_p} = \frac{\partial \overline{Z}}{\partial p},\ \overline{Z_q} = \frac{\partial \overline{Z}}{\partial q},\ \overline{Z_r} = \frac{\partial \overline{Z}}{\partial r},$$

$$\overline{Z_\eta} = \frac{\partial \overline{Z}}{\partial \eta}, \overline{Z_\zeta} = \frac{\partial \overline{Z}}{\partial \zeta}$$

$$\overline{L_u} = \frac{\partial \overline{L}}{\partial u},\ \overline{L_v} = \frac{\partial \overline{L}}{\partial v},\ \overline{L_w} = \frac{\partial \overline{L}}{\partial w},\ \overline{L_p} = \frac{\partial \overline{L}}{\partial p},\ \overline{L_q} = \frac{\partial \overline{L}}{\partial q},\ \overline{L_r} = \frac{\partial \overline{L}}{\partial r},$$

$$\overline{L_\eta} = \frac{\partial \overline{L}}{\partial \eta}, \overline{L_\zeta} = \frac{\partial \overline{L}}{\partial \zeta}$$

$$\overline{M_u} = \frac{\partial \overline{M}}{\partial u}, \ \overline{M_v} = \frac{\partial \overline{M}}{\partial v}, \ \overline{M_w} = \frac{\partial \overline{M}}{\partial w}, \ \overline{M_p} = \frac{\partial \overline{M}}{\partial p}, \ \overline{M_q} = \frac{\partial \overline{M}}{\partial q}, \ \overline{M_r} = \frac{\partial \overline{M}}{\partial r},$$

$$\overline{M_\eta} = \frac{\partial \overline{M}}{\partial \eta}, \overline{M_\zeta} = \frac{\partial \overline{M}}{\partial \zeta}$$

$$\overline{N_u} = \frac{\partial \overline{N}}{\partial u}, \ \overline{N_v} = \frac{\partial \overline{N}}{\partial v}, \ \overline{N_w} = \frac{\partial \overline{N}}{\partial w}, \ \overline{N_p} = \frac{\partial \overline{N}}{\partial p}, \ \overline{N_q} = \frac{\partial \overline{N}}{\partial q}, \ \overline{N_r} = \frac{\partial \overline{N}}{\partial r},$$

$$\overline{N_\eta} = \frac{\partial \overline{N}}{\partial \eta}, \overline{N_\zeta} = \frac{\partial \overline{N}}{\partial \zeta}$$

The six equations of motion of an airframe using (Eqn.2. 40) can thus be written as:

$$\Delta\dot{u} = r_0\Delta v + v_0\Delta r - q_0\Delta w - w_0\Delta q + \left(\overline{X_u}\Delta u + \overline{X_v}\Delta v + \overline{X_w}\Delta w + \overline{X_p}\Delta p + \overline{X_q}\Delta q + \overline{X_r}\Delta r + \overline{X_\eta}\Delta\eta + \overline{X_\zeta}\Delta\zeta\right) + \Delta\overline{T}$$

$$\Delta\dot{v} = p_0\Delta w + w_0\Delta p - r_0\Delta u - u_0\Delta r + \left(\overline{Y_u}\Delta u + \overline{Y_v}\Delta v + \overline{Y_w}\Delta w + \overline{Y_p}\Delta p + \overline{Y_q}\Delta q + \overline{Y_r}\Delta r + \overline{Y_\eta}\Delta\eta + \overline{Y_\zeta}\Delta\zeta\right)$$

$$\Delta\dot{w} = q_0\Delta u + u_0\Delta q - p_0\Delta v - v_0\Delta p + \left(\overline{Z_u}\Delta u + \overline{Z_v}\Delta v + \overline{Z_w}\Delta w + \overline{Z_p}\Delta p + \overline{Z_q}\Delta q + \overline{Z_r}\Delta r + \overline{Z_\eta}\Delta\eta + \overline{Z_\zeta}\Delta\zeta\right)$$

$$\Delta\dot{p} = \overline{I_x}(q_0\Delta r + r_0\Delta q) + \left(\overline{L_u}\Delta u + \overline{L_v}\Delta v + \overline{L_w}\Delta w + \overline{L_p}\Delta p + \overline{L_q}\Delta q + \overline{L_r}\Delta r + \overline{L_\eta}\Delta\eta + \overline{L_\zeta}\Delta\zeta\right)$$

$$\Delta\dot{q} = \overline{I_y}(r_0\Delta p + p_0\Delta r) + \left(\overline{M_u}\Delta u + \overline{M_v}\Delta v + \overline{M_w}\Delta w + \overline{M_p}\Delta p + \overline{M_q}\Delta q + \overline{M_r}\Delta r + \overline{M_\eta}\Delta\eta + \overline{M_\zeta}\Delta\zeta\right)$$

$$\Delta\dot{r} = \overline{I_z}(p_0\Delta q + q_0\Delta p) + \left(\overline{N_u}\Delta u + \overline{N_v}\Delta v + \overline{N_w}\Delta w + \overline{N_p}\Delta p + \overline{N_q}\Delta q + \overline{N_r}\Delta r + \overline{N_\eta}\Delta\eta + \overline{N_\zeta}\Delta\zeta\right)$$

This equations can be written in a matrix notation as:

$$\begin{bmatrix}\Delta\dot{u}\\ \Delta\dot{v}\\ \Delta\dot{w}\\ \Delta\dot{p}\\ \Delta\dot{q}\\ \Delta\dot{r}\end{bmatrix} = \begin{bmatrix} \overline{X_u} & (r_0 + \overline{X_v}) & (-q_0 + \overline{X_w}) & \overline{X_p} & (-w_0 + \overline{X_q}) & (w_0 + \overline{X_r}) \\ (-r_0 + \overline{Y_u}) & \overline{Y_v} & (p_0 + \overline{Y_w}) & (w_0 + \overline{Y_p}) & \overline{Y_q} & (-u_0 + \overline{Y_r}) \\ (q_0 + \overline{Z_u}) & (-p_0 + \overline{Z_v}) & \overline{Z_w} & (-v_0 + \overline{Z_p}) & (u_0 + \overline{Z_q}) & \overline{Z_r} \\ \overline{L_u} & \overline{L_v} & \overline{L_w} & \overline{L_p} & (\overline{I_x}r_0 + \overline{L_q}) & (\overline{I_x}q_0 + \overline{L_r}) \\ \overline{M_u} & \overline{M_v} & \overline{M_w} & (\overline{I_y}r_0 + \overline{M_p}) & \overline{M_q} & (\overline{I_y}p_0 + \overline{M_r}) \\ \overline{N_u} & \overline{N_v} & \overline{N_w} & (\overline{I_z}q_0 + \overline{N_p}) & (\overline{I_z}p_0 + \overline{N_q}) & \overline{N_r} \end{bmatrix} \begin{bmatrix}\Delta u\\ \Delta v\\ \Delta w\\ \Delta p\\ \Delta q\\ \Delta r\end{bmatrix}$$

$$+\begin{bmatrix} \overline{X_\eta} & \overline{X_\zeta} \\ \overline{Y_\eta} & \overline{Y_\zeta} \\ \overline{Z_\eta} & \overline{Z_\zeta} \\ \overline{L_\eta} & \overline{L_\zeta} \\ \overline{M_\eta} & \overline{M_\zeta} \\ \overline{N_\eta} & \overline{N_\zeta} \end{bmatrix} \begin{bmatrix} \Delta\eta \\ \Delta\zeta \end{bmatrix} \tag{2.41}$$

2.5.2 Incorporation of Accelerometer and Gyro Measurement Model

Generally, not all state variables in the state equation are accessible or measurable. The common measurement variables, in most missiles, is the acceleration components ($a_x, a_y, and\ a_z$).

The acceleration components measured at point O (where O is at a distance of, $d_x, d_y, and\ \ d_z$from the central point of gravity, c.g., along x-, y- and z-axis, respectively), may be written as:

$$a_x = \dot{u} + qw - rv - d_x(q^2 + r^2) + d_y(pq - \dot{r}) + d_z(pr + \dot{q}) \tag{2.42a}$$

$$a_y = \dot{v} + ru - pw + d_x(pq + \dot{r}) - d_y(p^2 + r^2) + d_z(qr - \dot{p}) \tag{2.42b}$$

$$a_z = \dot{w} + pv - qu + d_x(pr - \dot{q}) + d_y(qr + \dot{p}) - d_z(p^2 + q^2) \tag{2.42c}$$

If the accelerometer are mounted along x-axis (i.e. $d_y, d_z = 0$) which is usually the case, and most of the time in missile the only lateral acceleration is needed to control. Thus, neglecting the longitudinal acceleration eqns. (2.42a – 2.42c) reduce to

$$a_y = \dot{v} + ru - pw + d_x(pq + \dot{r}) = \overline{Y} + d_x(pq + \dot{r}) \tag{2.43a}$$

$$a_z = \dot{w} + pv - qu + d_x(pr - \dot{q}) = \overline{Z} + d_x(pr - \dot{q}) \tag{2.43b}$$

Note that the right hand side of Eqns. (2.43a) and (2.43b) come directly from eqns. (2.24 to 2.26).

Linearizing Eqns. (2.43a) and (2.43b) and using the relationship of Eqn. (2.41) gives us:

$$\Delta y(t) = [H]\Delta x(t) \tag{2.44}$$

$\Delta y(t) = \begin{bmatrix} \Delta a_y & \Delta a_z \end{bmatrix}^T$,is the output vector.

[H]=

$$\begin{bmatrix} \overline{Y_u} + \overline{N_u}d_x & \overline{Y_v} + \overline{N_v}d_x & \overline{Y_w} + \overline{N_w}d_x & \left[\overline{Y_p} + (q_0 + \overline{I_z}q_0 + \overline{N_p})d_x\right] & \left[\overline{Y_q} + (p_0 + \overline{I_z}p_0 + \overline{N_q})d_x\right] & \overline{Y_r} + \overline{N_r}d_x \\ \overline{Z_u} - \overline{M_u}d_x & \overline{Z_v} + \overline{M_v}d_x & \overline{Z_w} + \overline{M_w}d_x & \left[\overline{Z_p} + (r_0 - \overline{I_y}r_0 + \overline{M_p})d_x\right] & \overline{Z_q} - \overline{M_q}d_x & \left[\overline{Z_r} + (p_0 - \overline{I_y}p_0 - \overline{M_q})d_x\right] \end{bmatrix}$$

is the state output matrix [9].

Finally, the overall mathematical model of the missile system using the state space representations obtained on Eqns. (2.41), and (2.44) we obtain the overall system state space representation.

$$\begin{bmatrix}\dot{u}\\ \dot{v}\\ \dot{w}\\ \dot{p}\\ \dot{q}\\ \dot{r}\end{bmatrix}=\begin{bmatrix}X_u & 0 & 0 & X_p & 0 & 0\\ 0 & Y_v & 0 & 0 & 0 & -u_0+Y_r\\ 0 & 0 & Z_w & 0 & u_0+Z_q & 0\\ L_u & 0 & 0 & L_p & 0 & 0\\ 0 & 0 & M_w & 0 & M_q & 0\\ 0 & N_v & 0 & 0 & 0 & N_r\end{bmatrix}\begin{bmatrix}u\\ v\\ w\\ p\\ q\\ r\end{bmatrix}+\begin{bmatrix}0 & 0\\ 0 & Y_{zeta}\\ Z_{eta} & 0\\ 0 & 0\\ M_{eta} & 0\\ N_{eta} & N_{zeta}\end{bmatrix}\begin{bmatrix}\eta\\ \zeta\end{bmatrix} \quad (2.45a)$$

$$Y=\begin{bmatrix}0 & 0 & Y_v+N_v*d_x & 0 & Y_q-N_q*d_x & Y_r\\ 0 & Z_w+M_w*d_x & 0 & 0 & 0 & Z_r+M_r*d_x\end{bmatrix}\begin{bmatrix}u\\ v\\ w\\ p\\ q\\ r\end{bmatrix} \quad (2.45b)$$

Now, substituting the parameters that is used in most of missiles and papers [9].

$$\begin{bmatrix}\dot{u}\\ \dot{v}\\ \dot{w}\\ \dot{p}\\ \dot{q}\\ \dot{r}\end{bmatrix}=\begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0\\ 0 & -3 & 0 & 0 & 0 & -500\\ 0 & 0 & -3 & 0 & 500 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & -1 & 0 & -3 & 0\\ 0 & 1 & 0 & 0 & 0 & -3\end{bmatrix}\begin{bmatrix}u\\ v\\ w\\ p\\ q\\ r\end{bmatrix}+\begin{bmatrix}0 & 0\\ 0 & 180\\ -180 & 0\\ 0 & 0\\ -500 & 0\\ 0 & -500\end{bmatrix}\begin{bmatrix}\eta\\ \zeta\end{bmatrix} \quad (2.46a)$$

$$Y=\begin{bmatrix}0 & 0 & -3.5 & 0 & 1.5 & 0\\ 0 & 2.5 & 0 & 0 & 0 & 1.5\end{bmatrix}\begin{bmatrix}u\\ v\\ w\\ p\\ q\\ r\end{bmatrix} \quad (2.46b)$$

Where,

$U = [\eta \quad \zeta] = [eta \quad zeta]$

Therefore, the final system model obtained verifies that the system is linear and MIMO (multiple input multiple output) with two inputs elevator deflection $eta, (\eta)$ and Rudder Rudder deflection $zeta, (\zeta)$ and two lateral acceleration outputs $a_y \; and \; a_z$.

CHAPTER THREE

3. LINEAR QUADRATIC GAUSSIAN CONTROLLER AND PARTICLE SWARM OPTIMIZATION ALGORITHM

3.1 Linear Quadratic Gaussian Controller

LQG Control, is a robust control method since noise in the state and output equations is explicitly considered. It is assumed that the plant dynamics are linear, measurement noise and disturbance signals (process noise) are stochastic with known statistical properties.

Linear quadratic Gaussian regulators are observer based state feedback controllers. In this case, the optimal state observer is the Kalman filter and the optimal state feedback controller is the linear quadratic regulator (LQR). In the design of LQG, the state estimator and state feedback controller are designed independently. In control theory this principle is known as separation principle.

That is, we have a plant model

$$\dot{x} = Ax + Bu + w \tag{3.1a}$$

$$y = Cx + v \tag{3.1b}$$

w and v, are the disturbance (process noise) and measurement noise inputs respectively, which are usually assumed to be uncorrelated zero-mean Gaussian stochastic processes.

The LQG control problem is to find the optimal control $u(t)$ which minimizes

$$J = E\left\{\lim_{T\to\infty}\frac{1}{T}\int_0^T [x^T Qx + u^T Ru]dt\right\} \tag{3.2}$$

Where, Q and R, are appropriately chosen constant weighting matrices (design parameters) such that Q = $Q^T \geq 0$ and R = $R^T > 0$. The name LQG arises from the use of a linear model, an integral Quadratic cost function, and Gaussian white noise processes to model disturbance signals and noise [12].

The solution to the LQG problem, known as the Separation Theorem. It consists of first determining the optimal control to a deterministic linear quadratic regulator (LQR) problem. Namely, the above LQG problem without w and v. It happens that the solution to this problem can be written in terms of the simple state feedback law,

$$u(t) = -K_r x(t) \tag{3.3}$$

Where, K_r is state feedback gain matrix. The next step is to find an optimal estimate $\tilde{x}$ of the state *x*, so that $E = \{[x - \tilde{x}]^T[x - \tilde{x}]\}$ is minimized. The optimal state estimate is given by a Kalman filter and is independent of Q and R. The required solution to the LQG problem is then found by replacing *x by* $\tilde{x}$, to give

$$u(t) = -K_r \tilde{x}(t).$$

LQG problem and its solution can be separated into two distinct parts, as illustrated in Fig.3.1

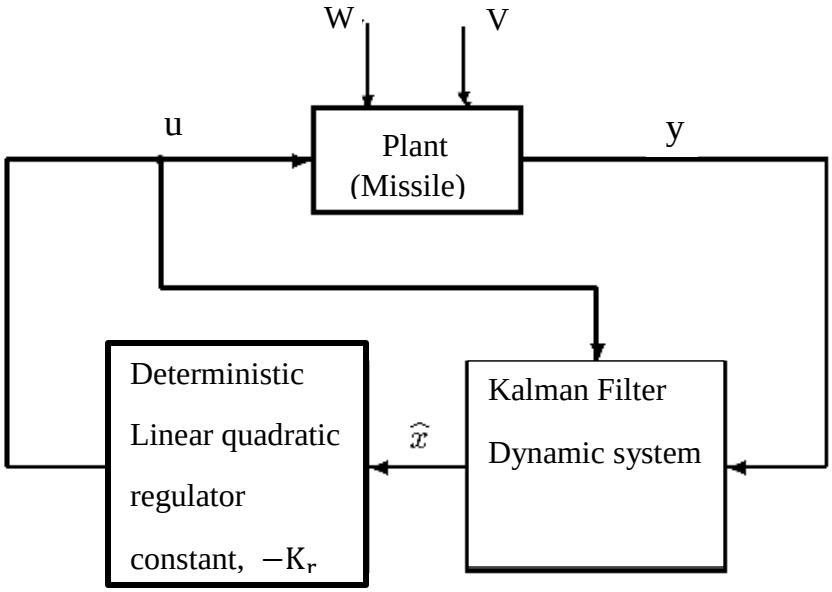


Fig.3. 1: The Separation Theorem [13].

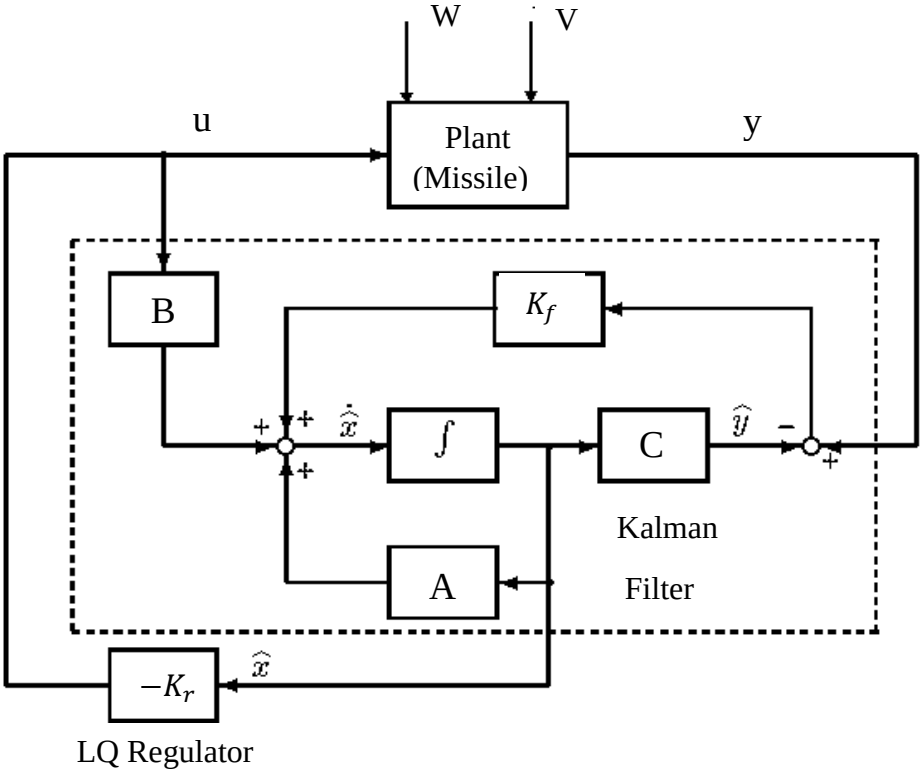


Fig.3. 2: The LQG controller and noisy plant [13].

The necessary equation to find the optimal state-feedback matrix Kr and the Kalman filter are:

3.1.1 Optimal state feedback: The LQR problem, where all the states are known, is the deterministic initial value problem: given the system $\dot{x} = Ax + Bu$ with a non-zero initial state $x(0)$, find the input signal $u(t)$ which takes the system to the zero state ($x = 0$) in an optimal manner, i.e. by minimizing the deterministic cost,

$$J_r = \int_0^\infty [x(t)^T Q x(t) + u(t)^T R u(t)] dt \tag{3.4}$$

The optimal solution (for any initial state) is $u(t) = -K_r x(t)$, where

$$K_r = R^{-1} B^T P$$

and $P = P^T \geq 0$ is the unique positive-semidefinite solution of the algebraic Riccati equation

$$A^T P + AP - PBR^{-1} B^T P + Q = 0 \tag{3.5}$$

Cheap control: The term "cheap control" here means that control effort is inexpensive and one can use any large control signals to ensure the dynamic behavior of the system. In this case, the weight on u(t) can be made very small, i.e. R is very small; and Q, should be very large.

Expensive control: In contrast to the cheap control strategy, with an expensive control strategy, the control cost is assumed to be quite large. Therefore, the control signal u(t) should be made as small as possible. In this case, a large R should be used and; Q should be very small.

3.1.2 Kalman filter: The Kalman filter has the structure of an ordinary state-estimator or observer, as shown in Fig.3.2, with,

$$\dot{\tilde{x}} = A\tilde{x} + Bu + K_f(y - C\tilde{x}) \tag{3.6}$$

The optimal choice of K_f, which minimizes $E = \{[x - \tilde{x}]^T [x - \tilde{x}]\}$, is given by

$$K_f = YC^T {W_n}^{-1}$$

Where $Y = Y^T \geq 0$ is the unique positive-semidefinite solution of the algebraic Riccati equation [13].

$$YA^T + AY - YC^T V^{-1} CY + W_d = 0 \tag{3.7}$$

Kalman filter is a well-known recursive state estimator for linear systems. By using recursive calculations, the Kalman filter is capable of estimating the state of a linear dynamic system from a series of noisy measurements.

In short, the optimal compensator (LQG) design process is the following:

I. Design an optimal regulator for a linear plant assuming full-state feedback (i.e. assuming all the state variables are available for measurement) and a quadratic objective function (such as that given by Eq. (3.4)). The regulator is designed to generate a control input, u(t), based upon the measured state-vector x(t).
II. Design a Kalman filter for the plant assuming a known control input, u(t), a measured output, y(t),and white noises, w and v, with known power spectral densities. The Kalman filter is designed to provide an optimal estimate of the state-vector, $\tilde{x}$(t).

III. Combine the separately designed optimal regulator and Kalman filter into an optimal compensator, which generates the input vector, u(t) ,based upon the estimated state-vector, $\tilde{x}(t)$, rather than the actual state-vector, x(t), and the measured output vector, y(t).

3.2 Robustness of LQG and Loop Transfer recovery

As described in the [12], [13], [14] the standard LQG problem can be obtained as the solution of the optimal state estimation problem and the optimal state feedback problem, and both of these two sub problems have good robustness, but the LQG has not. Doyle (1978) (in a paper entitled "Guaranteed Margins for LQG Regulators" with a very compact abstract which simply states "There are none") [15]. He showed, by example, that there exist LQG combinations with arbitrarily small gain margins. To overcome this, the loop transfer recovery (LTR) design procedure allows one to design a full modified LQG control system, and to approach the good robustness properties exhibited by either optimal state feedback or a kalman filter. Hence there are two versions of LTR [13], one approaches good robustness of the optimal state feedback control, the approaches the good robustness of the Kalman filter. The first version consists the following two steps:

1. Design an optimal state feedback system: Using the weighting matrices Q and R to design a controller by the standard LQR design method, which is the same as Eqns. (3.4 and 3.5). According to [11] good robustness properties come automatically at this stage.
2. Synthesize a Kalman filter in the following way: Set $\mathbf{W_d = qI}$, and $\mathbf{W_n = I}$, where q is a positive real number. Then q is increased, when q is large enough, the robustness of step (1) can be recovered. W_d and W_n are the power spectral density matrices.

Second version is

1. Design a Kalman filter: Using the weighting matrices W and V to design a kalman filter which is the same as Eqns. (3.6 and 3.7). According to [11] good robustness properties come automatically at this stage.
2. Synthesize an optimal state feedback system in the following way: Set **Q = qI**, and **R = I**, where q is a positive real number. Then q is increased, when q is large enough, the robustness of step (1) can be recovered.

There are two methods for the application of the loop recovery in a feedback loop.

Define G and Φ as:

$$G = C(SI - A)^{-1}B$$

$$\Phi = (SI - A)^{-1}$$

Method 1: Recovery at plant output

- ✓ The open loop transfer matrix obtained by breaking the LQG loop at point 4 of Fig.3.3 is the KF open loop transfer matrix $C\Phi K_f$. Since u acts the same way to system and KF, it is convenient to assume u = 0, y = 0.
- ✓ The open loop transfer matrix obtained by breaking the LQG loop at point 2 of Fig.3.3 is GK_{LQG} . It can be made to approach $C\Phi K_f$ by using the above LTR procedure.

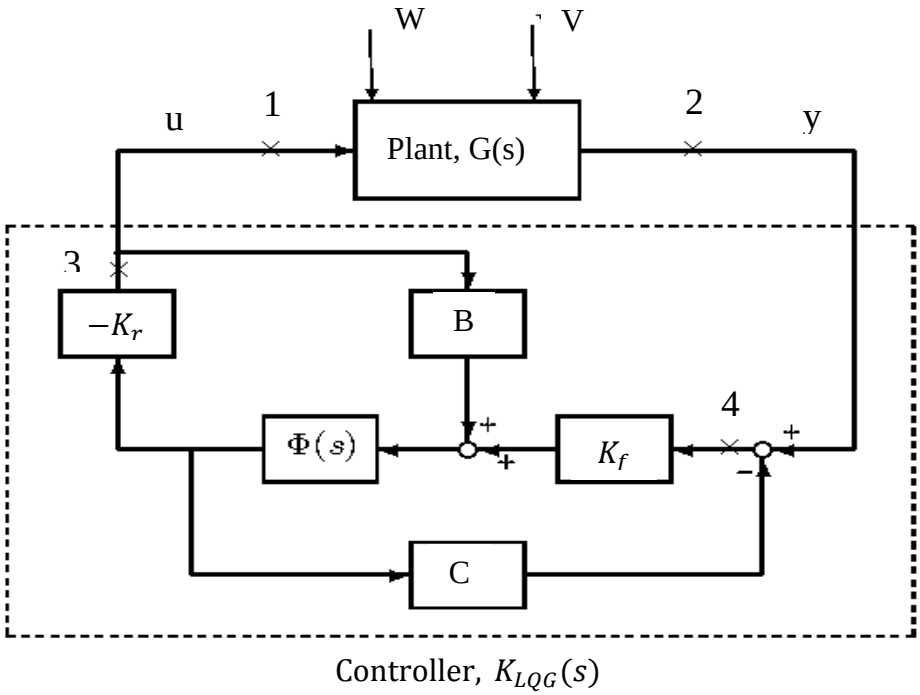


Fig.3. 3: LQG-controlled plant [13].

Method 2: Recovery at plant input

- ✓ The open loop transfer matrix obtained by breaking the LQG loop at point 3 of Fig.3.3 is the LQR open loop transfer matrix $K_r\Phi B$. In this analysis, it is convenient to assume state estimation error is zero.
- ✓ The open loop transfer matrix obtained by breaking the LQG loop at point 1 of Fig.3.3 is $K_{LQG}G$. It can be made to approach $K_r\Phi B$ by using the above LTR procedure.

In this thesis, the robustness of the designed controller will be explained in chapter 4.i.e.robustness at the input and output will be compared at different frequencies by finding the maximum and minimum singular values of the return ratio matrix and also the loop transfer recovery approach will be applied if the designed LQG controller is not robust as compared to LQR controller.

3.3 Particle Swarm Optimization Algorithm

3.3.1 Introduction to PSO

Particle swarm optimization is one of the most popular nature-inspired metaheuristic optimization algorithm developed by James Kennedy and Russell Eberhart in 1995 [16]. The principle of PSO optimization technique is to search for the best solution in n-dimension in the search space. That can be done by a collaborative share between each individual particle in a swarm and it is used for optimizing nonlinear control systems for its satisfactory results.

Particle swarm optimization (PSO) is inspired by social and cooperative behavior displayed by various species to fill their needs in the search space. The algorithm is guided by personal experience (Pbest), overall experience (Gbest) and the present movement of the particles to decide their next positions in the search space. Further, the experiences are accelerated by two factors

c1and c2, and two random numbers generated between [0, 1] whereas the present movement is multiplied by an inertia factor w [17][18].

The initial population (swarm) of size N and dimension D is denoted as $X = [X_1,\ X_2,.\ .\ .X_N]^T$, where T denotes the transpose operator. Each individual (particle) $X_i\ (i = 1,2 \dots N)$ is given as $X_i = \left[X_{i,1}, X_{i,2}, \dots X_{i,D}\right]$. Also, the initial velocity of the population is denoted as $V = [V_1, V_2, \dots\ V_N]^T$. Thus, the velocity of each particle $X_i\ (i = 1,2 \dots N)$ is given as $V_i = \left[V_{i,1}, V_{i,2}, \dots V_{i,D}\right]$. The index i varies from 1 to N whereas the index j varies from 1 to D. The detailed algorithms of various methods are described below for completeness.

$$V_{i,j}^{k+1} = w \times V_{i,j}^{K} + c_1 \times r_1 \times \left(P_{best\ i,j}^{K} - X_{i,j}^{K}\right) + c_2 \times r_2 \times \left(G_{best\ j}^{K} - X_{i,j}^{K}\right) \tag{3.8}$$

$$X_{i,j}^{K+1} = X_{i,j}^{K} + V_{i,j}^{k+1} \tag{3.9}$$

Where,

w = inertia factor

c_1 = cognitive factor

c_2 = social factor

$r_1\ and\ r_2$ = uniform random numbers in the interval (0 1).

$X_{i,j}^{K}$ = particle position

$V_{i,j}^{K}$ = particle velocity

$P_{best\ i,j}^{K}$ = Best "remembered" individual particle position

$G_{best\ j}^{K}$ = Best "remembered" swarm position

In equation (3.8) $P_{best\ i,j}^{K}$ represents personal best j^{th} component of i^{th} individual whereas, $G_{best\ j}^{K}$ represents j^{th} component of the best individual of population up to iteration k. Figure (1) shows the search mechanism of PSO in multidimensional search space. How the PSO algorithm is passes the local minimum and approaches the global minimum is more explained [16].

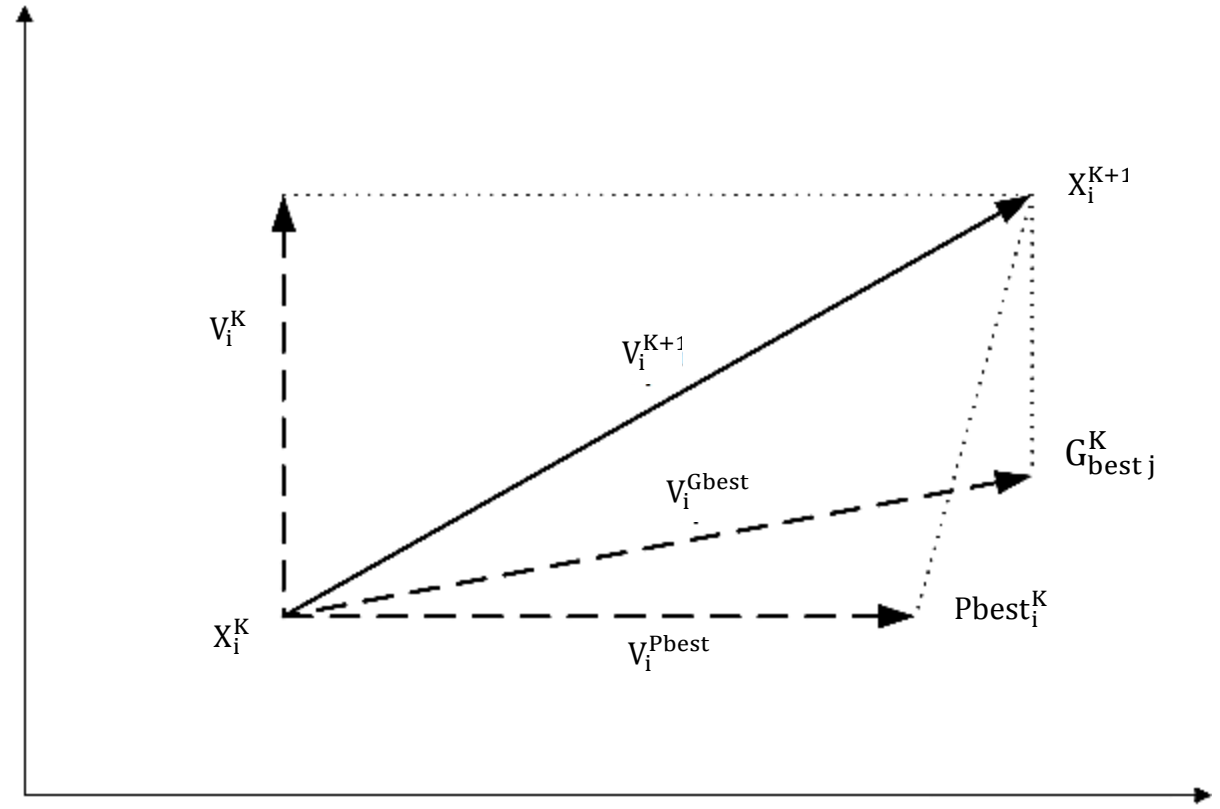


Fig.3. 4: PSO search mechanism in multidimensional search space.

3.3.2 Basic steps of PSO algorithm

1. Set parameters w, c_1, c_2 of PSO.

2. Initialize population (array) of particles with random positions (X) and velocities (V) on "d" dimensions in the problem space.

3. Set iteration k = 1.

4. For each particle, evaluate the desired optimization fitness function in "d" variables. i.e. $J_i^k = J\left(X_i^k\right)$,$\forall i$ and find the index of the best particle b.

5. Compare particle's fitness evaluation with particles *Pbest*, if current value is better than *Pbest*, then set *Pbest* value equal to the current value, and the *Pbest* location equal to the current location in d-dimensional space i.e. select $P_{best\ i}^K = X_i^K$, $\forall i$.

6. Compare fitness evaluation with the population's overall previous best. If the current value is better than *Gbest*, then reset *Gbest* to the current particle's array index and value. i.e. $G_{best}^K = X_b^K$.

7. Update velocity and position of particles

$V_{i,j}^{k+1} = w \times V_{i,j}^K + c_1 \times rand() \times \left(P_{best\ i,j}^K - X_{i,j}^K\right) + c_2 \times rand() \times \left(G_{best\ j}^K - X_{i,j}^K\right)$
$\forall i\ and\ \forall j$

$$X_{i,j}^{K+1} = X_{i,j}^{K} + V_{i,j}^{k+1} \quad \forall i \; and \; \forall j$$

8. Evaluate fitness $J_i^{k+1} = J\left(X_i^{k+1}\right) \forall i$ and find the index of the best particle b1.

9. Update *Pbest* of population $\forall i$

If $J_i^{k+1} < J_i^k$, then $P_{best\,i}^{K+1} = X_i^{K+1}$ else $P_{best\,i}^{K+1} = P_{bset\,i}^{K}$.

10. Update *Gbest* of population

If $J_{b1}^{k+1} < J_b^k$, then $G_{best}^{K+1} = P_{best\,b1}^{K+1}$ and set *b=b1* else $G_{best}^{K+1} = G_{best}^{K}$.

11. If *k < maxite then k=k+1* and go to step 4 else go to step 12

12. Print optimum solution as $\boldsymbol{G_{best}^{K}}$

A detailed flowchart of PSO considering the above steps is shown in Fig.3.5.

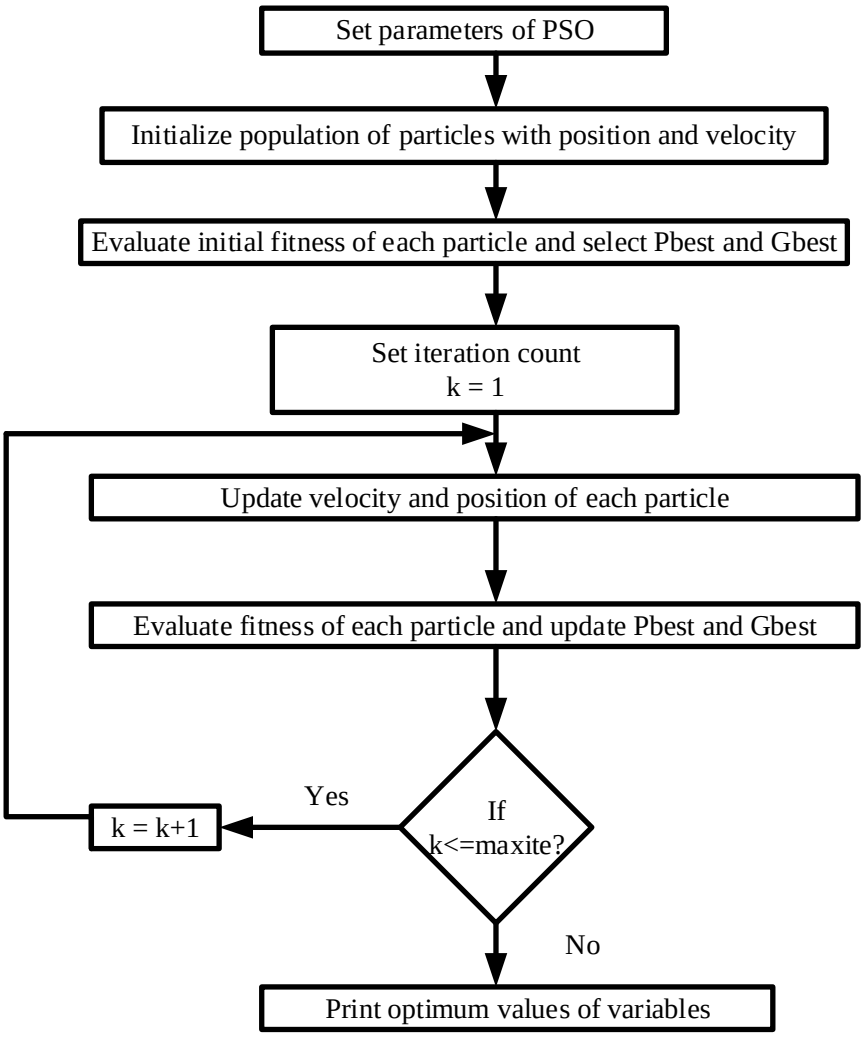


Fig.3. 5: Flowchart of PSO.

3.3.3 Parameters of PSO algorithm and problem definition

The most common parameters used are [18]:

- Inertial weight: 0.9 to 0.4
- Acceleration factors ($c_1 and\ c_2$): 1.5 to 2.5
- Population size: 10 to 100
- Maximum iteration (Maxite): 50 to 10000
- Initial velocity: 10 % of position

Problem Formulation of Optimization Problem: The convergence of the optimization algorithms towards the global optimal solution is characterized by the fitness function. Some of the commonly used fitness functions to design controllers are integral of the absolute error (IAE). The objective function (OF) or fitness function i.e. the cost function in linear quadratic regulator problem is a minimization problem. Thus, my aim is to find the optimum value of Q (positive semi definite) and R (positive definite) that minimizes the cost function or the error.

So, I did in such manner; given the missile model then design LQR controller but the state control vector (Q) and control effort parameter (R) is tuned by the intelligent method particle swarm optimization. So, I have four by four Q matrix and two by two R matrix and both of them are diagonal. This diagonal variables are tuned in each iteration and I calculate the state feedback gain matrix and then I find the step response of the system. After this I minimize the error subtracting the feedback from the reference.

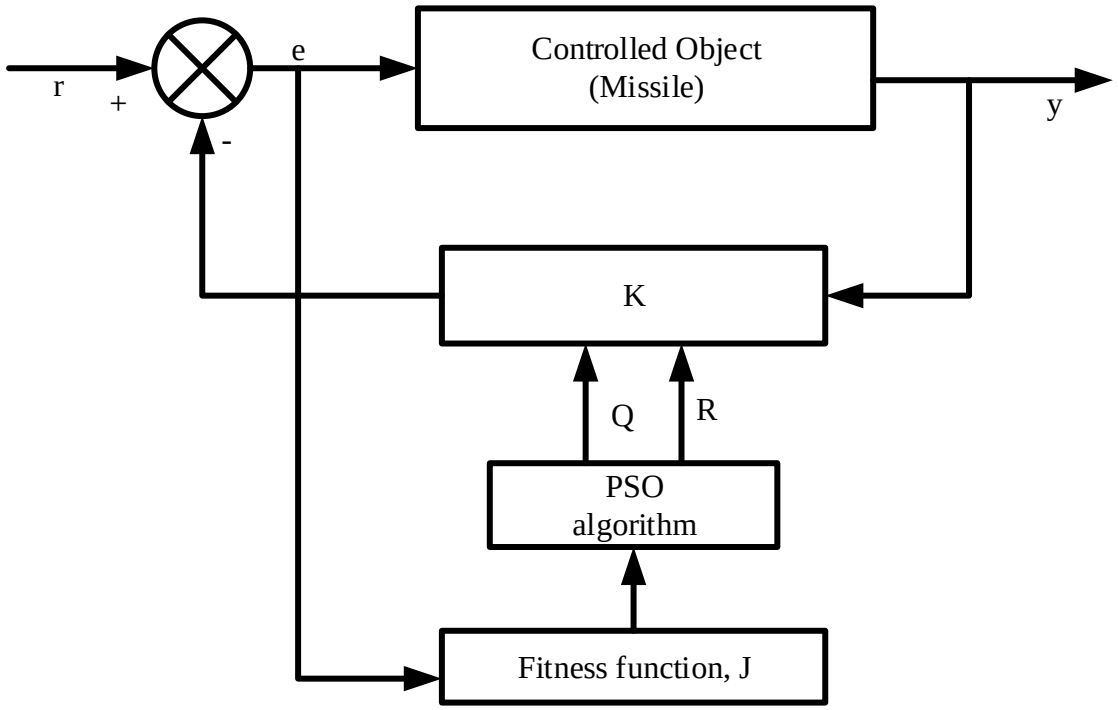


Fig.3. 6: The block diagram showing the LQR optimization using Particle swarm optimization.

CHAPTER FOUR

4. SIMULATION STUDIES AND ANALYSIS OF RESULTS

In this chapter, system analysis and design of state feedback controllers for the Missile system that stabilizes and gives good performance for the system, will be discussed. The state feedback controllers designed are the linear quadratic regulators (LQR) and the linear quadratic Gaussian regulators (LQG).

The system analysis is undertaken before the design of the controllers since, the Missile system is 2x2 MIMO, the effect of each pair of inputs on each output will be analyzed and the appropriate input-output loop will be selected. While designing the controllers, different methodologies have been followed, the conventional design procedure (by tuning the weighting matrices manually) and the proposed methodology (using particle swarm optimization to tune the weighting matrices). Finally the controllers designed with different methodology will be compared with each other and with the control objective that have been set.

4.1 Minimal Realization

Before, I move to the system analysis and controller design the system controllability and observability must be checked. For a system,

$$\dot{x} = Ax + Bu \tag{4.1}$$

$$y = Cx + Du \tag{4.2}$$

Controllability: The state equation (4.1) or the pair (A, B) is said to be controllable if for any initial state $x(0) = x_0$ and any final state x_1, there exists an input that transfers x_0 to x_1 in a finite time. Otherwise (4.1) or (A. B) is said to be uncontrollable.

Observability: The state equation (4.2) is said to be observable if for any unknown initial state x(0), there exists a finite $t_1 > 0$ such that the knowledge of the input u and the output y over $[0,t_1]$ suffices to determine uniquely the initial state x(0). Otherwise, the equation is said to be unobservable.

Thus, checking the controllability and observability by MATLAB™ ***(ctrb(A,B) and obsv(A,C))***. The system is neither controllable nor observable or the controllability and observability matrix is not full rank. So, to design a controller the system must be controllable and observable and the solution is minimize the uncontrollable and unobservable states.

Minimal realization: A state equation (A, B, C, D) is a minimal realization of a proper rational function g(s) if and only if (A, B) is controllable and (A, C) is observable or if and only if [19]:

Dimension **A** = degree g(s) $\qquad g(s) = \frac{n(s)}{d(s)}$

More generally, controllable and observable state equations and coprime fractions contain essentially the same information and either description can be used to carry out analysis and design or it represents the actual system.

Therefore, using Matlab^{TM} function ***minreal(system)***(*minreal* implies minimal realization) we obtained the minimal realization of the system that is controllable and observable and the state reduces from six to four state which implies there was two uncontrollable states

4.2 Missile system analysis and stability

The model developed has shown that the system is 2-input 2-output system. This makes the system to have two manipulated variables and two control inputs. The model of the Missile system is given by the state-space representation as in Eqns. (2.46a and 2.46b). In order to determine the poles and zeros of the system, the missile model can be represented by a 2x2 transfer matrix function G(S) given as:

$$\mathrm{G(s)} = \begin{bmatrix} \mathrm{G11(s)} & \mathrm{G12(s)} \\ \mathrm{G21(s)} & \mathrm{G22(s)} \end{bmatrix} \tag{4.3}$$

Using Matlab command *G(s) = tf(ss(A,B,C,D)), gives*

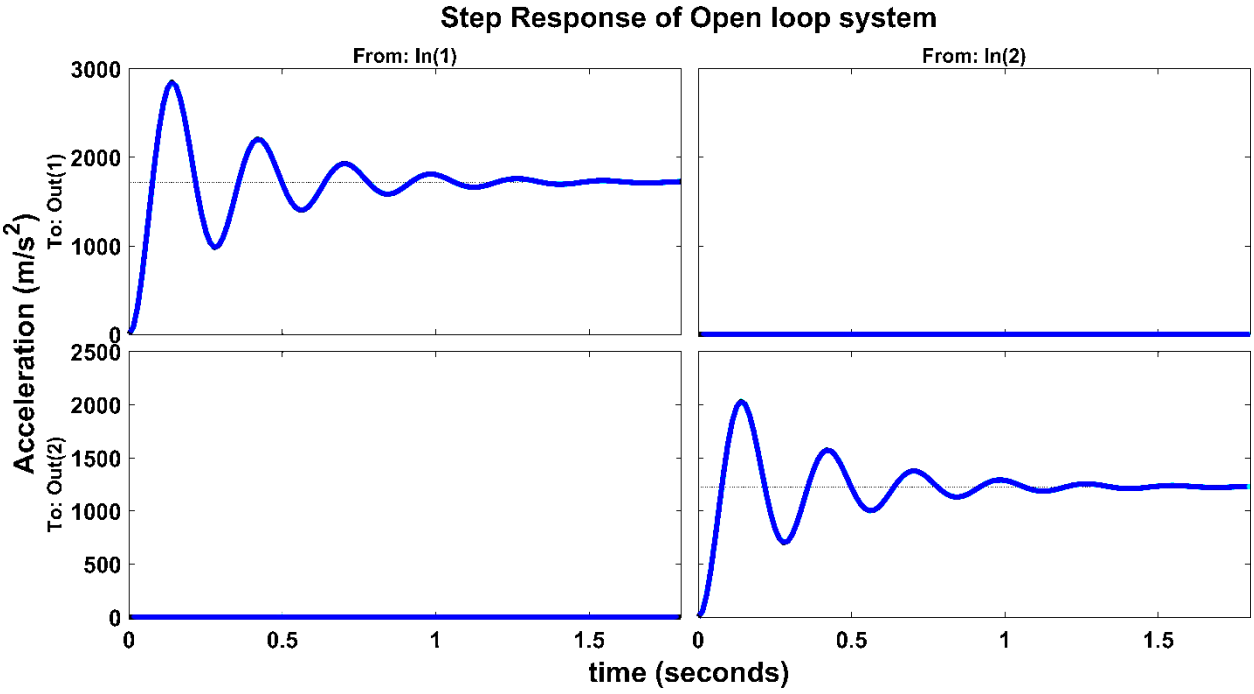
$$\mathrm{G11(s)} = \frac{-120 * S^3 + 874190 * S^2 + 5.188\mathrm{e}06 * \mathrm{S} + 4.453\mathrm{e}08}{S^4 + 12 * S^3 + 1054 * S^2 + 6108 * S + 2.591e05}$$

$$\mathrm{G12(s)} = 0$$

$$\mathrm{G21(s)} = 0$$

$$\mathrm{G22(s)} = \frac{-300 * S^3 + 6.226\mathrm{e}05 * S^2 + 3.594\mathrm{e}06 * \mathrm{S} + 3.178\mathrm{e}08}{S^4 + 12 * S^3 + 1054 * S^2 + 6108 * S + 2.591e05}$$

The open-loop response of the system to a step input can be used to verify the unstable and non-minimum phase or used to show the system performance fulfills or not the specification or property of the system. Fig.4.1 shows the system has low performance. Thus, we must design a controller that will give good performance that minimizes this large amount of error.



Acceleration in the Y = in(1), out(1) and acceleration in Z = in(2), out(2)

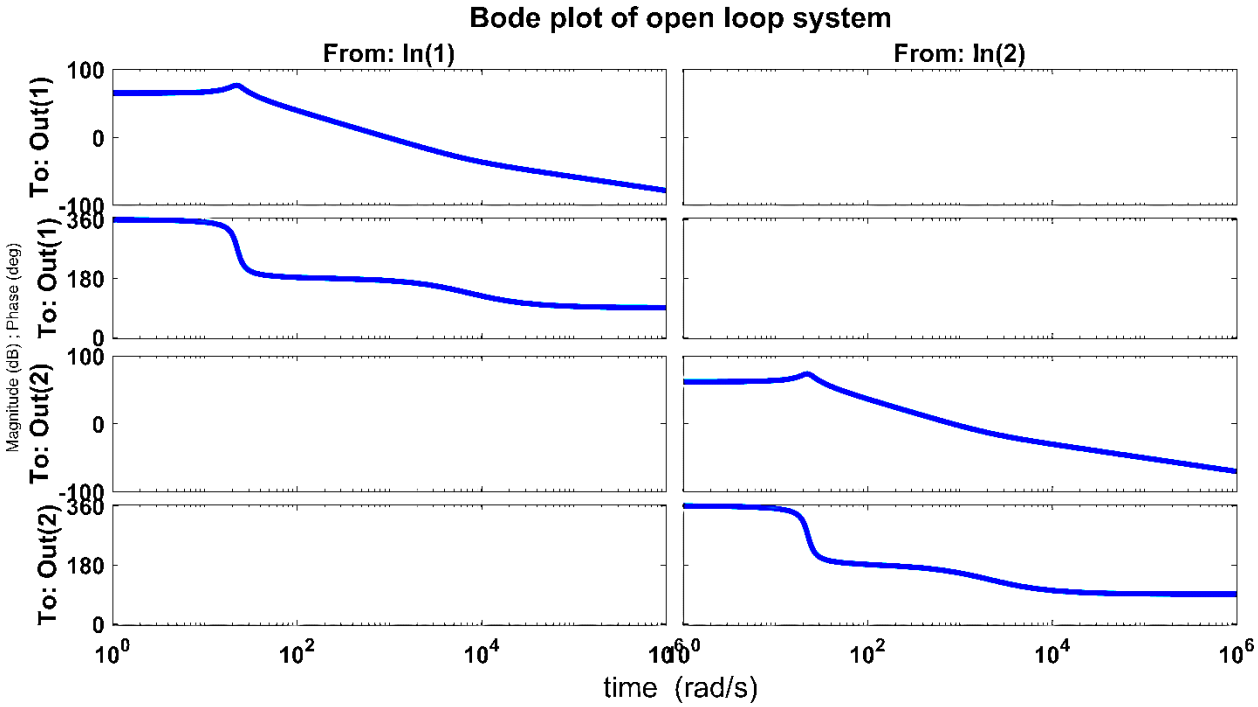
Fig.4. 1: Open-loop response of the linear Missile model for unit-step input.

4.3 MIMO system interaction and coupling

One of the most challenging aspects of MIMO systems control is the interaction between different inputs and outputs. In general, each input will have an effect on every output of the system (outputs are coupled). In the Missile control system also the control action on elevator deflection for lateral acceleration in Y direction not only affects the acceleration in the Y direction but, it will also have effect on the acceleration in the Z direction and the control action on rudder deflection for acceleration in the Y direction not only affect the acceleration in the Z direction but, it will affect in the Z direction.

In order to approach this MIMO control design problem as a multiple SISO control design problems, we should either decouple the system using different decoupling techniques or justify that the system is already decoupled. A certain MIMO system is totally decoupled when the non-diagonal elements of the square transfer matrix functions are zero. In other words, if the transfer matrix is a diagonal matrix the system is said to be decoupled [13].

Now, In the case of the Missile system, the diagonal elements of the transfer functions in the transfer matrix are zero. According to the previous definition that has been given for decoupled MIMO systems, in general, the system can be approached as a multiple SISO problem, because of diagonal transfer functions are zero. Thus the plant in this cases is sufficiently diagonal, which makes it easier to control. A diagonally dominant plant has a transfer matrix in which the transfer functions on the diagonal are greater in magnitude than the off-diagonal elements.



Magnitude and phase in the Y = in(1), out(1) and magnitude and phase in Z = in(2), out(2)

Fig.4. 2: Bode plots of the four transfer functions in the transfer matrix.

Thus, from the bode plot of the above figure, the magnitude of the two diagonal transfer functions of the transfer matrix are greater than the off-diagonal (the magnitude is zero) elements at all frequencies. Therefore this shows that the Missile is diagonal plant. If a transfer matrix is diagonal, it is possible to design a good controller by considering each input-output pair as a separate loop.

4.3.1 Relative Gain array

The relative gain array (RGA), R(s), can be useful for establishing the best input-output pairings for use in the control of a multivariable system. An important issue in the controller design is the appropriate selection of input-output pairs. One way of choosing the pairing is thus the relative gain array (RGA). The RGA will show the measure of process interaction and gives indication of control loop pairings. The array, R(s) is defined as the element by element product (the Hadamard or Schur product) of the transfer function matrix, G(s) and the transpose of the inverse of this matrix [13].

$$\mathrm{R}(0) = \mathrm{G}(0).* \mathrm{G}^{-T}(0) \tag{4.4}$$

Now, for the Missile system the transfer functions matrix is two by two and the RGA is calculated as:

$$\mathrm{R(s)} = \begin{bmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{bmatrix} = \begin{bmatrix} \lambda_{11} & 1-\lambda_{11} \\ 1-\lambda_{11} & \lambda_{11} \end{bmatrix} \tag{4.5}$$

Where,

$$\lambda_{11} = \frac{1}{1 - \frac{G12(0)G21(0)}{G11(0)G22(0)}}$$

Evaluating G(0) from equation (4.3), we get

$$\lambda_{11} = \frac{1}{1 - \frac{0 * 0}{(1.718 * 10^3) * (1.226 * 10^3)}}$$

$$\lambda_{11} = 1$$

Therefore, the Relative gain array is

$$R(0) = \begin{bmatrix} \lambda_{11} & 1 - \lambda_{11} \\ 1 - \lambda_{11} & \lambda_{11} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

From the RGA result obtained above $\lambda_{11} = \lambda_{12} = 1$ indicates that coupled interaction does not affect the pairing of U_1 with Y_1 and $U_2$with Y_2. Therefore it is a clear that we should select the two loops since we can sufficiently manipulate the outputs Y_1 and Y_2 using the input control signals $U_1$and U_2.

On the other hand the results $\lambda_{12} = \lambda_{21} = 0$ indicate that the output Y_1 with input U_2 and Y_2 with U_1 are loosely or not related. Thus, no control on the outputs $Y_1$and Y_2 using the inputs U_2 and U_1 respectively.

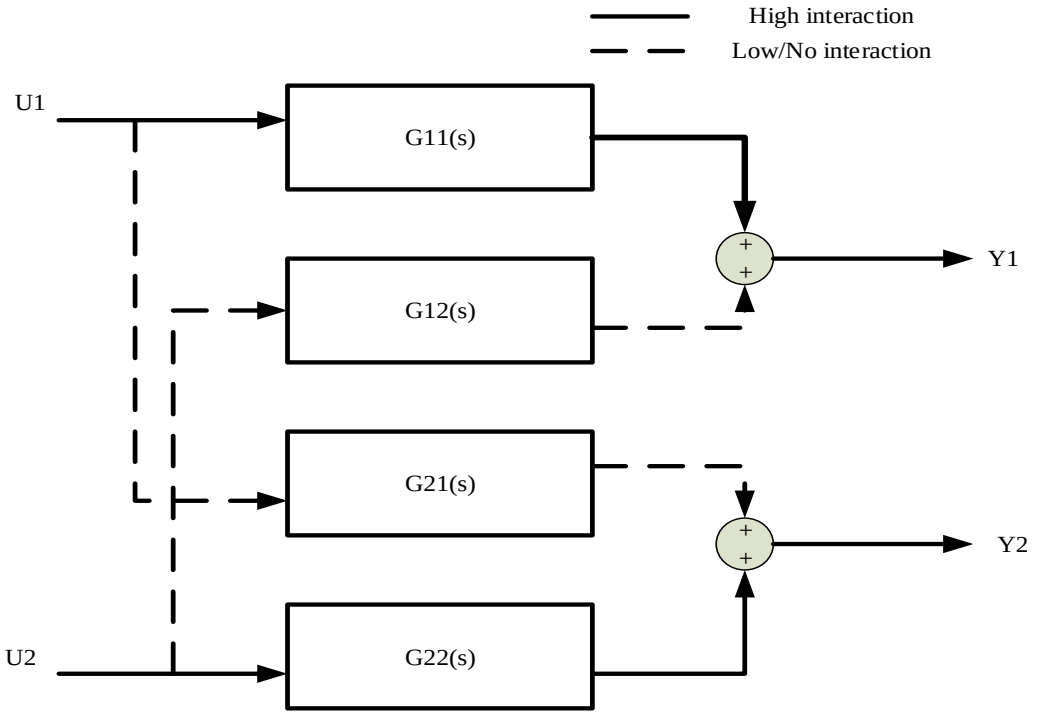


Fig.4. 3: Input-output pairing and interaction.

4.3.2 Condition Number

The condition number has been used as an input-output controllability measure, and in particular it has been postulated that a large condition number indicates sensitivity to uncertainty. This is not true in general, but the reverse holds; if the condition number is small, then the multivariable effects of uncertainty are not likely to be serious.

Condition number of a matrix is the ratio between the maximum and minimum singular values.

$$\gamma(G) \triangleq \frac{\overline{\sigma}(G)}{\underline{\sigma}(G)}$$

Where, $\overline{\sigma}(G)$ is the maximum singular value of G.

$\underline{\sigma}(G)$ = is the minimum singular value of G.

A matrix with a large condition number is said to be ill-conditioned. A large condition number may mean that the plant has a large minimized condition number, or equivalently, it has large RGA-elements which indicate fundamental control problems. Now to check the above plant (missile) has control problems or easy to control (if the condition number is small) the condition number of G(s) is calculated below:

Note that for a triangular matrix or diagonal matrix, the RGA is always the identity matrix and $\gamma(G)$ is always 1 [13].

In the above calculation, I get the value of RGA value that is identity.

RGA(G) = $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ since the plant is diagonal there are no interactions so RGA is "I" and the minimized condition number is 1. Thus, the condition number of the plant is small which indicates there is no control problem or it is easy to control the missile which gives confidence for the controller that I want to design.

4.4 Controller Design Objectives

The ultimate goal of every feedback controller design effort is to achieve a desired behavior of the closed loop system by means of appropriately processing measurements of the states from the physical plant, G(S) and feeding back the control signals, U(s).

In controller design, the term performance refers to the closed loop system's behavior. Performance requirements can be formulated either in the time domain or in the frequency domain. Typical requirements of time domain performance are often defined in terms of the response the system to a step input of size one. The primary objective for the design of the controller is to obtain the fastest time response possible while maintaining stability robustness. The following time domain specifications are to be considered in the design of the linear quadratic Gaussian controller for the missile.

Settling Time $T_s < 0.4\ sec$

Steady State Error $e_{ss} < 0.01$ (Due to a unit step command)

Overshoot $O_v \leq 4\%$ (Due to a unit step command)

Now, the conventional design procedure (by tuning the weighting matrices manually) and the proposed methodology (using PSO to tune the weighting matrices) will be discussed. Finally the controllers designed with different methodology will be compared with each other and with the control objective that have been set.

The first procedure is keeping R matrix constant (R=diag([1 1])) and checking for different Q values. Here for the initial value of Q I can choose the best trail method for finding Q. Thus, Q will be

$$\mathrm{Q} = \mathrm{diag}([10.0 * \mathrm{Q_i}\ 10.0 * \mathrm{Q_i}\ 10.0 * \mathrm{Q_i}\ 10.0 * \mathrm{Q_i}]).$$

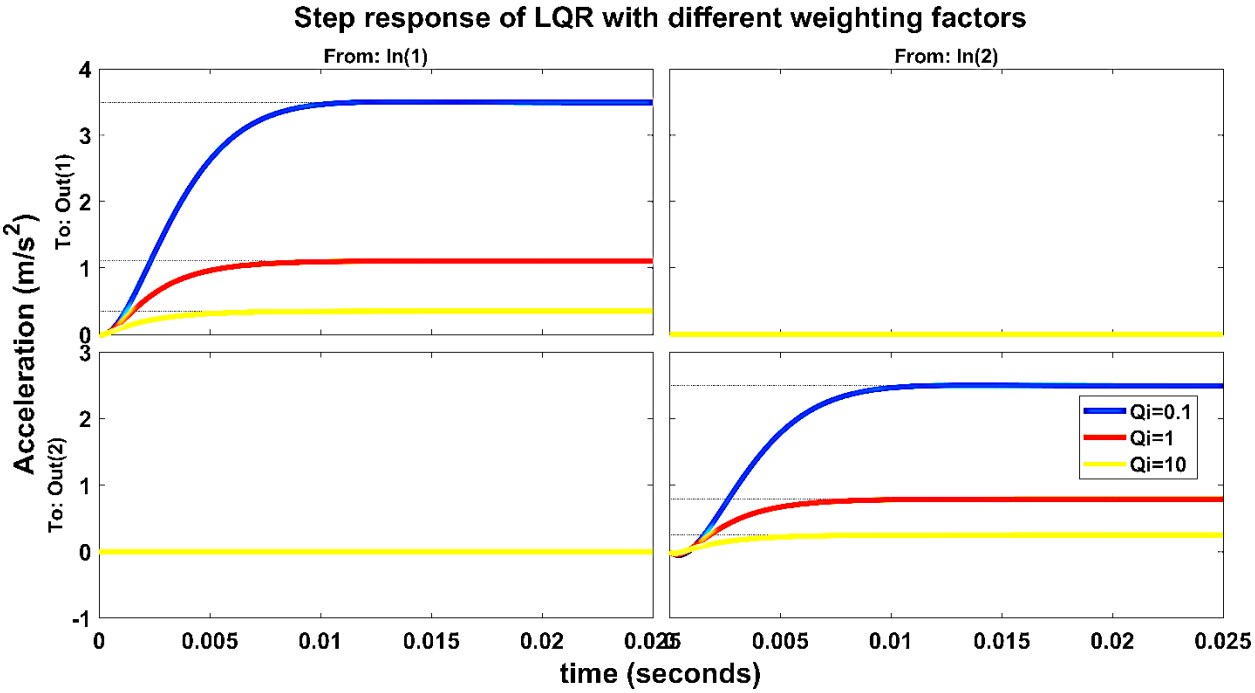
$$\mathrm{Q_i} = [0.1\ 1\ 10]$$

In the second case keeping Q matrix constant i.e. choosing the best Q value from the above and introducing different R values. Thus

$$\mathrm{R} = \mathrm{diag}([R_i\ R_i])$$

$$R_i = [0.1\ 1\ 10]$$

From the above the controllability matrix of the Missile plant has full rank, which makes it fully controllable. So, we can design LQR controller and we can find the state feedback gain matrix using the Matlab^{TM} function, *K = lqr(A,B,Q,R).*(*lqr* implies linear quadratic regulator).



Acceleration in the Y = in(1), out(1) and acceleration in Z = in(2), out(2)

Fig.4. 4: Step response of LQR for fixed R and different Q matrices.

For the diagonally interacting input-output pairs i.e. $G_{11}(s)$ and $G_{22}(s)$ the performance of the LQR for unit step input shown in Fig.4.4 has been summarized in (Table 4.1) and (Table 4.2).

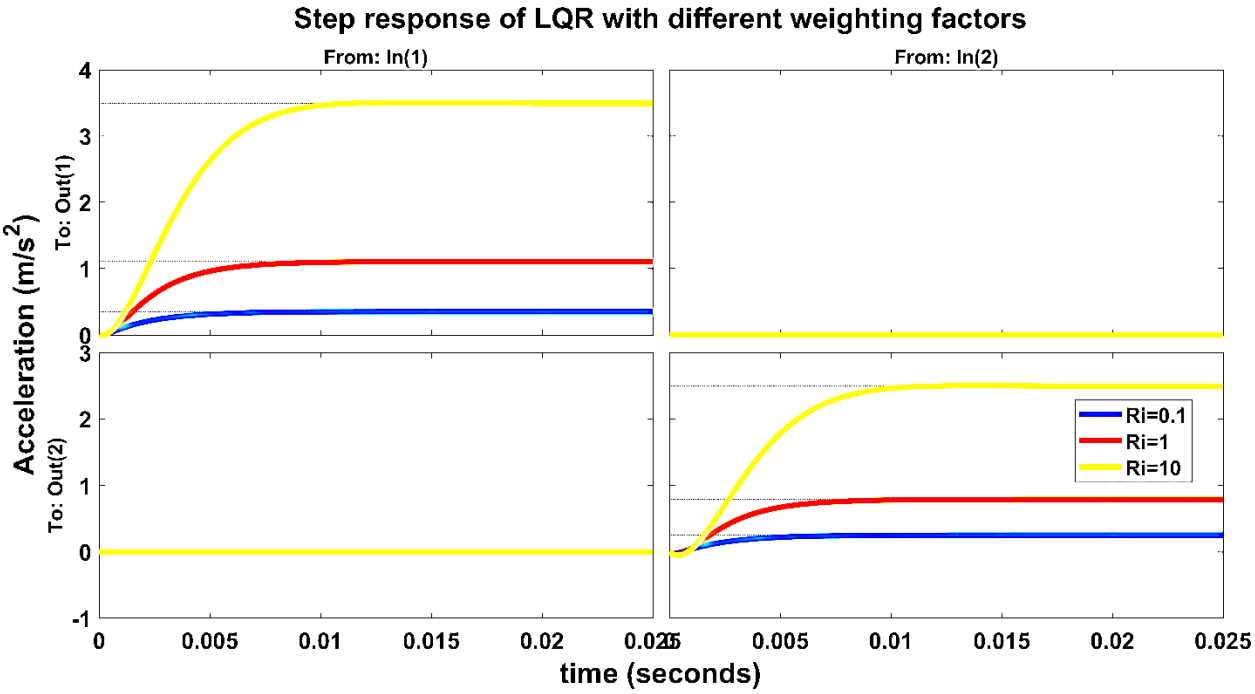
Table 4. 1: The performance of LQR for different values of Q (acceleration in the Y – axis).

acceleration y(i)	R=eye(2) Q_i	Rise time (t_r)	Settling time (t_s)	Steady state error (e_{ss})	Peak response (y_p)
1	0.1	0.0056	0.0092	-2.4925	3.5014
2	1	0.0051	0.0088	-0.1043	1.1043
3	10	0.0047	0.0086	0.6508	0.3492

Table 4. 2: The performance of LQR for different values of Q (acceleration in the Z– axis).

acceleration z (i)	R=eye(2) Qi	Rise time (t_r)	Settling time (t_s)	Steady state error (e_{ss})	Peak response (y_p)
1	0.1	0.0055	0.0094	-1.4924	2.4924
2	1	0.0047	0.0090	0.2120	0.7880
3	10	0.0046	0.0087	0.7508	0.2492

From the above (Table 4.1, the step response of the acceleration in the Y direction) and (Table 4.2 the step response of the acceleration in the Z direction) we can observe that the rise time and settling time show an improvement as the value of Q_i increases. The best peak response and steady state error results also are resulted at the value of $Q_i = 1$. Therefore, the best results are obtained at Qi = 1. For the off diagonally related input-output pairs i.e. G12 and G21 the desired characteristic is that the inputs should produce zero output since, it is required remove the effect of the inputs on the other outputs. Thus, the LQR designed has produced the responses observed in Fig.4.4 and the output is zero as the desired response.



Acceleration in the Y = in(1), out(1) and acceleration in Z = in(2), out(2)

Fig.4. 5: Step response of LQR for fixed Q and different R matrices.

For the diagonally interacting input-output pairs i.e. $G_{11}(s)$ and $G_{22}(s)$ the performance of the LQR for unit step input shown in Fig.4.4 has been summarized in (Table 4.3) and (Table 4.4).

Table 4. 3: Performance of LQR for different values of Ri (acceleration in the Y-axis)

Acceleration y (i)	Q=1 Ri	Rise time (t_r)	Settling time (t_s)	Steady state error (e_{ss})	Peak response (y_p)
1	0.1	0.0047	0.0085	0.6508	0.3492
2	1	0.0048	0.0088	-0.1043	1.1043
3	10	0.0056	0.0092	-2.4925	3.5014

Table 4. 4: Performance of LQR for different values of Ri (acceleration in the Z-axis)

Acceleration z (i)	Q=1 Ri	Rise time (t_r)	Settling time (t_s)	Steady state error (e_{ss})	Peak response (y_p)
1	0.1	0.0046	0.0087	0.7508	0.2492
2	1	0.0047	0.0090	0.2120	0.7880
3	10	0.0055	0.0094	-1.4924	2.4924

From the above (Table 4.3, the step response of the acceleration in the Y direction) and (Table 4.4 the step response of the acceleration in the Z direction) we can observe that the rise time and settling time show an improvement as the value of Ri decreases. The best peak response and steady state error results also are resulted at the value of Ri=1. Therefore, the best results are obtained at

Ri=1. For the off diagonally related input-output pairs i.e. G12 and G21 the desired characteristic is that the inputs should produce zero output since, it is required to remove the effect of the inputs on the other outputs. Thus, the LQR designed has produced the responses observed in Fig.4.5 and the output is zero as the desired response.

4.5 Particle swarm optimization tuned LQR

One of the main challenges in the design of LQR for real time applications is the optimal choice of state and input weighting matrices (Q and R), which play a vital role in determining the performance and optimality of the controller. Commonly, trial and error approach (above) is employed for selecting the weighting matrices, which not only burdens the design but also results in non-optimal response. Hence, to choose the elements of Q and R matrices optimally, PSO algorithm is formulated and applied for the control of Missile guidance system.

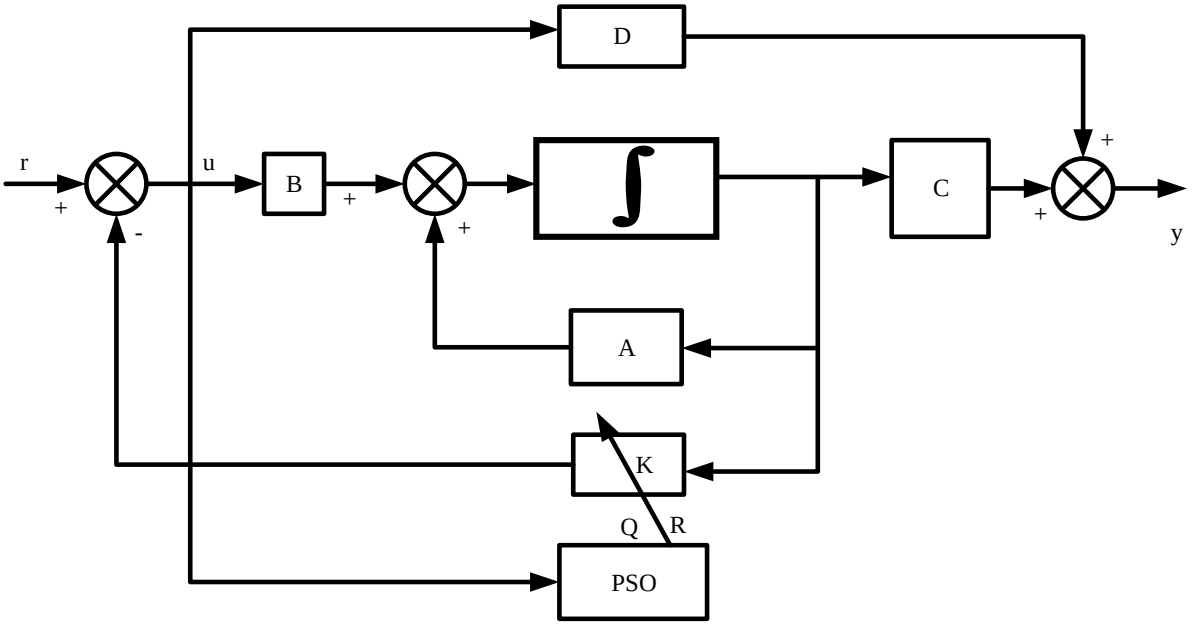


Fig.4. 6: The schematic of proposed method for tuning LQR using PSO algorithm.

In this thesis, particle swarm optimization algorithm is implemented by writing a code in MATLAB™, which was very hard task.

To implement the PSO optimization, first the fitness function must be defined. The fitness function is the objective function that needs to be minimized. In this case the objective function contains the function that represents the error of the response of the system to a unit step input. Beside the fitness function the number of variables to be optimized should also be specified. The variables to be optimized are the diagonal elements of the Q and R matrices.

After appying the particle swarm optimization I get the values of Q and R, state feedback gain matrix, and the poles of the system as follows:

Table 4. 5: Optimal parameters of PSO tuned LQR

Parameters of system	Values
Weighting matrix (Q)	$\begin{bmatrix} 9.4740 & 0 & 0 & 0 \\ 0 & 9.4276 & 0 & 0 \\ 0 & 0 & 5.1159 & 0 \\ 0 & 0 & 0 & 5.9292 \end{bmatrix}$
Weighting matrix (R)	$\begin{bmatrix} 0.7731 & 0 \\ 0 & 1.5255 \end{bmatrix}$
Optimal gain matrix (Kpso)	$\begin{bmatrix} 0.0000 & -3.4762 & -2.6344 & -0.0000 \\ 2.4787 & 0.0000 & -0.0000 & -2.2075 \end{bmatrix}$
Eigen values of the system (eig(A))	$1.0e{+}03* \begin{bmatrix} -0.7779 + 0.1386i \\ -0.7779 - 0.1386i \\ -1.2477 + 0.0000i \\ -0.7012 + 0.0000i \end{bmatrix}$

After finding the best Q and R values, the controller gain matrix can be found using LQR Matlab™ function *K = lqr(A,B,Q,R)* and, using the above optimized value of Q and R we get the following response. Using the results obtained the LQR is designed and the response has been shown in Fig.4.7b and compared to the manually tuned LQR. Fig.4.7a shows how the fitness fnction (error) is minimized as the iteration increases in the PSO agorithm.

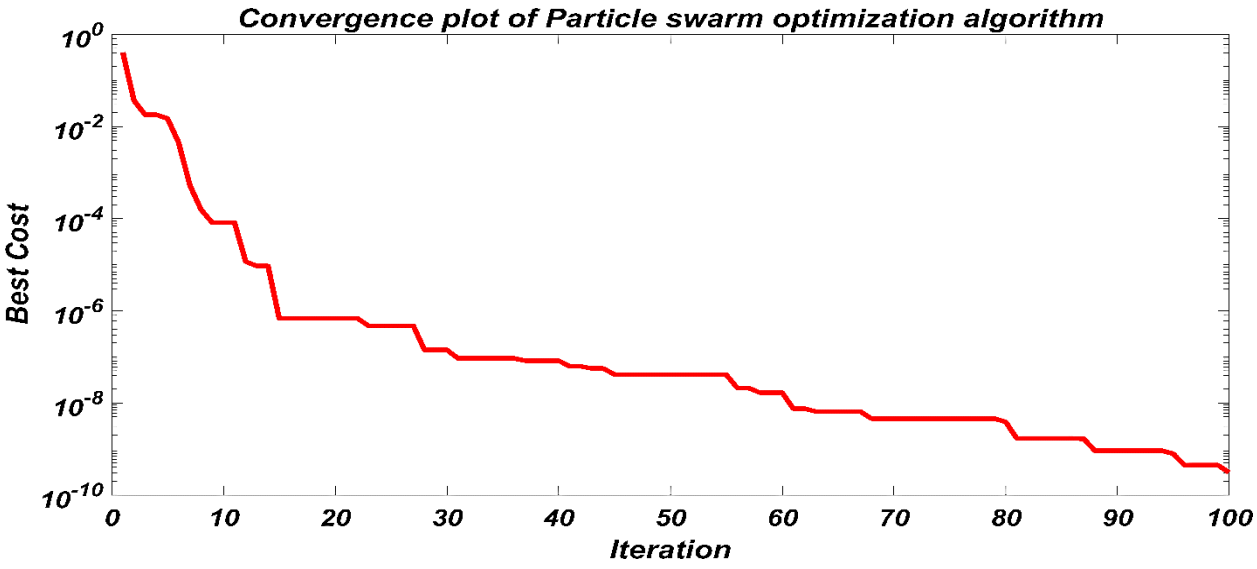
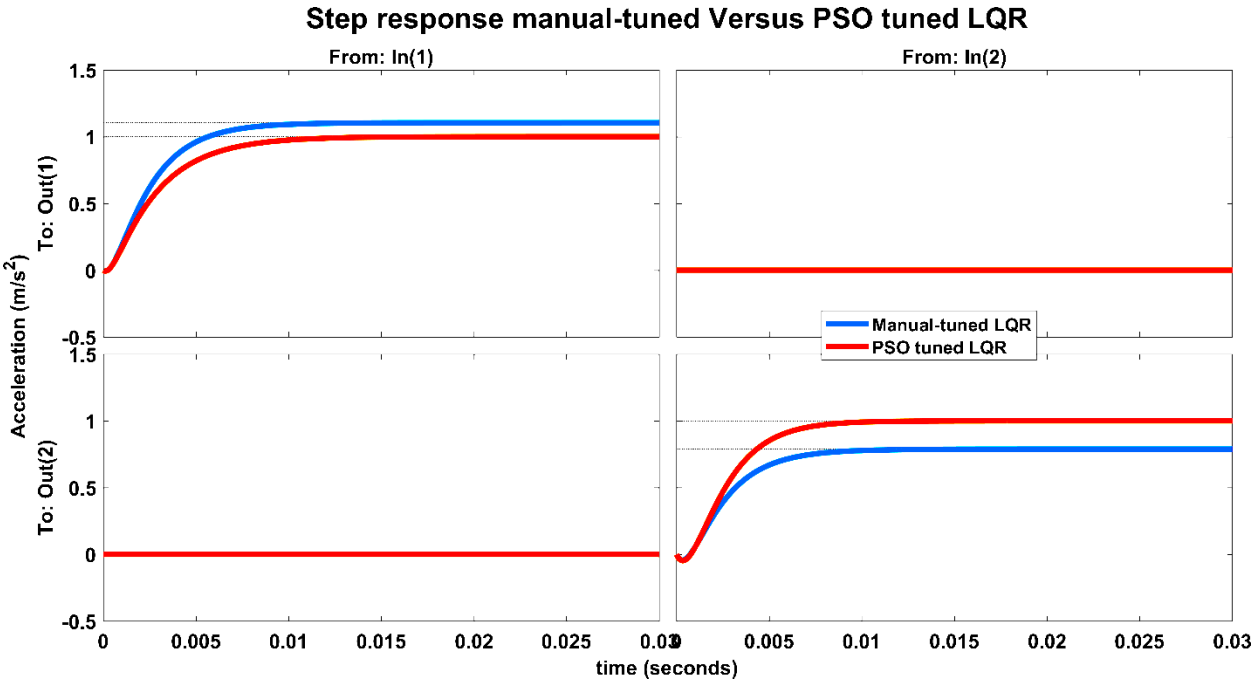


Fig.4.7a. Convergence plot of PSO using Semilog.



Acceleration in the Y = in(1), out(1) and acceleration in Z = in(2), out(2)

Fig.4. 7b: Unit step response of PSO tuned LQR Vs Manually tuned LQR.

Table 4. 6: The performance of Manual tuned LQR and PSO tuned LQR (acceleration in the Y-axis direction)

Acceleration in Y	Manual tuned LQR	PSO tuned LQR
Rise time (t_r)	0.0048	0.0058
Settling time (t_s)	0.0088	0.0106
Steady state error (e_{ss})	-0.1043	0
Peak response (y_p)	1.1043	1

From the above (Table 4.6), we can be observe in the time domain specifications i.e in steady state error and peak respnose the performance of paricle swarm optimization tuned LQR have best performance over the manually tuned LQR controller. The rise time and settling time of the particle swarm optimization tuned LQR have shown slight increase compared to the manually tuned LQR or in settling time and rise time response the manually tuned LQR has good performance even though the result obtained satisfies the control objective.

Table 4.7:The performance of manual tune LQR and PSO tuned LQR (acceleration in theZ-axis)

Acceleration in Z	Manual tuned LQR	PSO tuned LQR
Rise time (t_r)	0.0047	0.0045
Settling time (t_s)	0.0090	0.0085
Steady state error (e_{ss})	0.2120	0
Peak response (y_p)	0.7880	1

From the above (Table 4.7), we can be observe that the performance of paricle swarm optimization tuned LQR has best performance over the manually tuned LQR controller in all time domain specifications. Therefore, from the above table we conclude that the rise time, settling time, seady state error and peak response are well improved by using paricle swarm optimization algorithm than manually tuned LQR. Thus, the PSO algorithm give best optimized values of Q and R parameters and also the control objective is achieved with improvement of the overall system.

For the off diagonally related input-output pairs i.e. G12 and G21 the desired characteristic is that the inputs should produce zero output since, it is required remove the effect of the inputs on the other outputs. Thus, the LQR designed has produced the responses observed in Figure 4.7 and the output is zero as the desired response.

But LQR is design when the following two conditions are met.

- The model is perfectly known and
- All the states should be available for measurement.

In the case of Missile guidance system, the model of the system has its own level of uncertainty. This is due to the rejected model components during the linearization process. And also employing measurement techniques and devices to extract measurement each and every state makes the control system quite expensive. Thus, this problem can be addressed by another controller design approach known as the Linear Quadratic Gaussian (LQG) Regulator.

4.6 Linear Quadratic Gaussian Regulator (LQG) Design

In LQG the Kalman filter estimates the states of the system by taking the control signal and the output of the system. The estimation of the states is undertaken in the presence of both process disturbance and measurement noises.

The Kalman filter considered here is steady state Kalman filter; it is a linear estimation and is associated with time invariant systems. In such a case, the estimation error will also be a stationary white noise with a constant optimal covariance matrix, since the process disturbance and measurement noise vectors are assumed to be stationary white Gaussian noises.

To design the Kalman filter, the measurement noise and process noise covariance matrices need to be specified. The measurement noise covariance can be obtained from the expected value of rms noise from each sensor which is available from manufacturer's specifications. Whereas, the process noise covariance are more difficult to specify, values are often determined by trial-and-error with simulation studies.

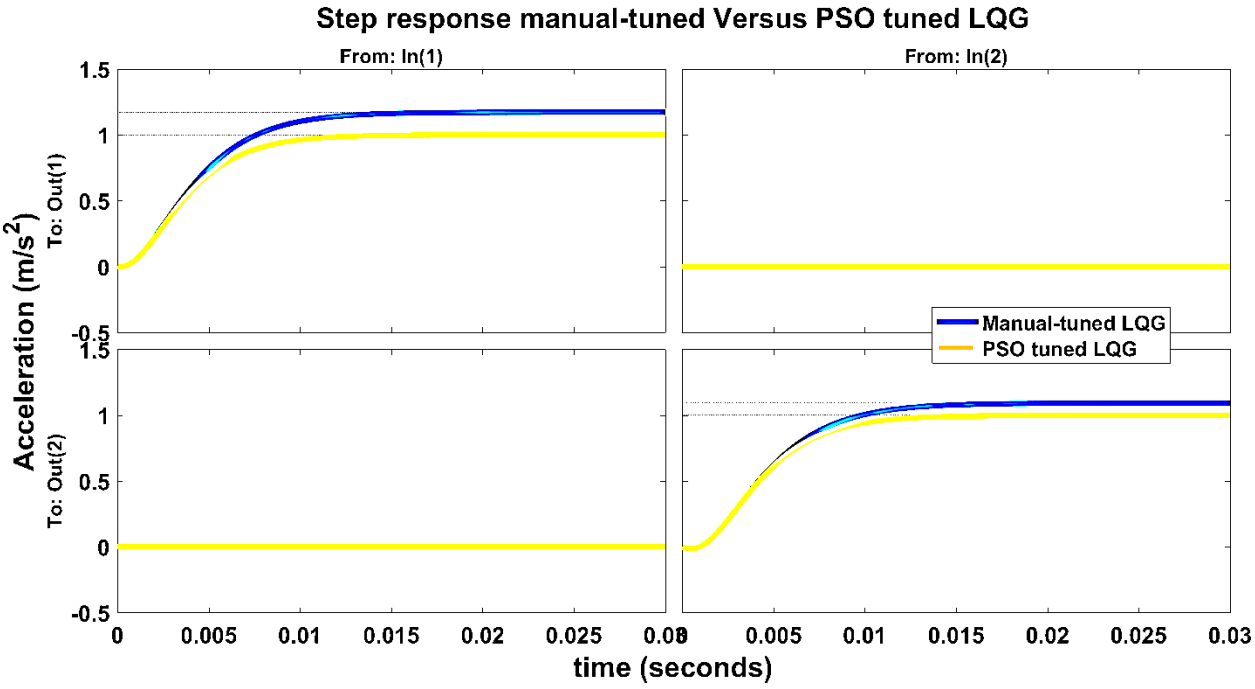
Generally, if the model of the plant has high uncertainties or the plant is operated constantly under the influence of external factors then we make the process disturbance covariance matrix to have a higher value than the measurement noise covariance matrix. If the plant model is well known but there exists a noise due to measuring devices, then we take higher values for the measurement noise covariance.

The sufficient conditions will be met if the system with state dynamics A, and observation matrix C, is obseravable. The plant is observable if the rank of the observability matrix has a full rank and this is fulfilled.

The process noise covariance matrix (W_d) is a positive semi-definite matrix and the measurement noise covariance matrix (V_n) is a positive definite matrix.

Since nonlinearities appear as process noise for all the state variables of the linear plant model, the process noise coefficient matrix is assumed to be an identity matrix, i.e. $W = I$. The linear Kalman filter is to be designed using the spectral densities of process and measurement noise, W, V, such that the exact state-vector, $\tilde{x}(t)$, is accurately estimated. After some trial and error, I select four by four diagonal matrix, W = 1300*I, and two by two diagonal matrix, V = 25*I, (I is identity matrix), and the Kalman filter gain will be obtained.

The algebraic Riccati equation for the steady state kalman filter is solved and the Kalman filter gains are easily obtained by using the MATLAB function ***lqe(A, W_d,C, W_d, V_n)*** (*lqe* implies linear quadratic estimator)**.** Once the gains (L) are determined then combine with the state feedback gain (K) and using MATLAB^{TM} function command the linear quadratic gaussian (LQG) is designed as follows.



Acceleration in the Y = in(1), out(1) and acceleration in Z = in(2), out(2)

Fig.4. 8: Unit step response of Manual and PSO tuned LQG.

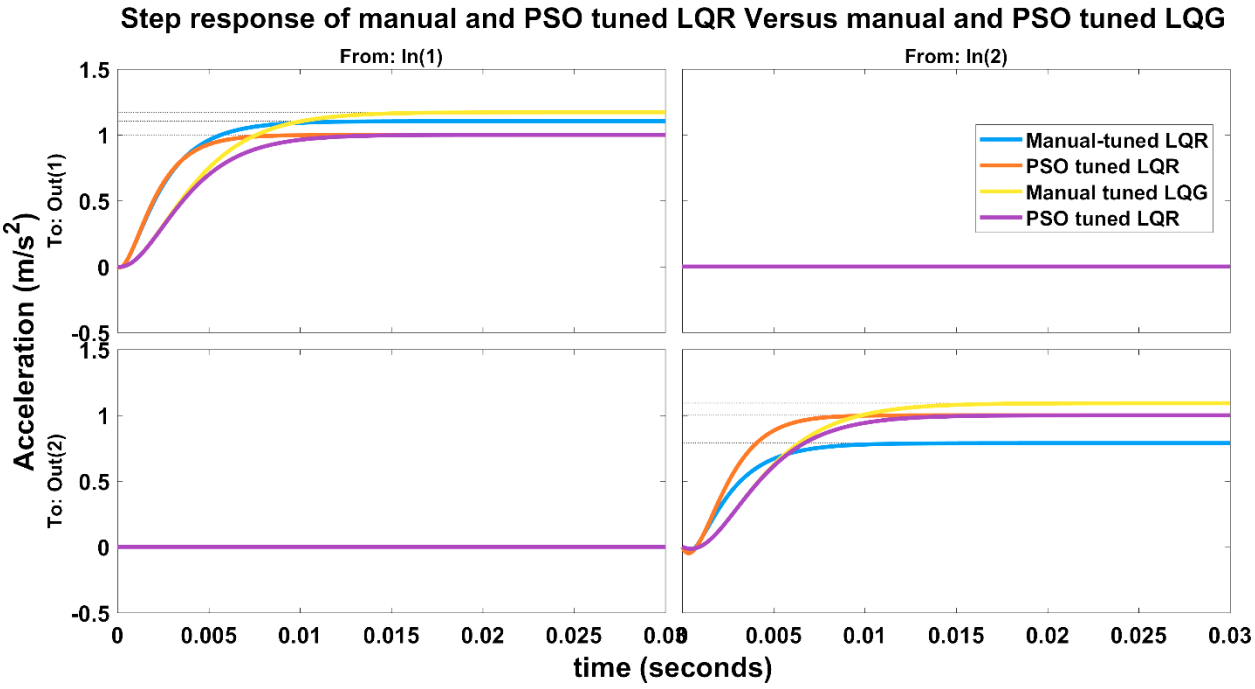
Table 4. 8:The performance of Manual tuned LQG and PSO tuned LQG (acceleration in the Y axis)

Acceleration in Y	Manual tuned LQG	PSO tuned LQG
Rise time (t_r)	0.0072	0.0064
Settling time (t_s)	0.0127	0.0114
Steady state error (e_{ss})	-0.1723	0
Peak response (y_p)	1.1723	1

Table 4. 9 Performance of Manual tuned LQG and PSO tuned LQG (acceleration in the Z axis)

Acceleration in Z	Manual tuned LQG	PSO tuned LQG
Rise time (t_r)	0.0074	0.0068
Settling time (t_s)	0.0136	0.0125
Steady state error (e_{ss})	-0.0918	0
Peak response (y_p)	1.0918	1

Therefore, as it has been noted from the above (Table 4.8 and 4.9) the performance of paricle swarm optimization tuned LQG has best performance over the manually tuned LQG controller in all time domain specifications in both accelerations. Therefore, from the above table we conclude that the rise time, settling time, seady state error and peak response are well improved by using paricle swarm optimization algorithm than manually tuned LQG. Thus, the PSO algorithm give best optimized values of Q and R parameters and also the control objective is achieved with improvement of the overall system.



Acceleration in the Y = in(1), out(1) and acceleration in Z = in(2), out(2).

Fig.4. 9: Unit step response of manual and PSO tuned LQR Versus manual and PSO tuned LQG.

Table 4. 10: The performance of Manual and PSO tuned LQR Versus manual and PSO tuned LQG (acceleration in the Y direction)

Acceleration in Y	Manual tuned LQR	PSO tuned LQR	Manual tuned LQG	PSO tuned LQG
Rise time (t_r)	0.0048	0.0058	0.0072	0.0064
Settling time (t_s)	0.0088	0.0106	0.0127	0.0114
Steady state error (e_{ss})	0.1043	0	-0.1723	0
Peak response (y_p)	1.1043	1	1.1723	1

From the above (Table 4.10), we can be observe that in the time domain specifications i.e in steady state error and peak respnose the performance of paricle swarm optimization (PSO tuned LQR and LQG have best performance over the manually tuned LQR and LQG controller. The rise time and settling time of the particle swarm optimization tuned LQR have shown slight increase compared to the manually tuned LQR or in settling time and rise time response the manually tuned LQR has good performance.But in the LQG contoller the PSO tuned LQG has best performance over the manually tuned LQG controller even though the result obtained satisfies the control objective.

The maually tuned LQR and PSO tuned LQG have a slightly better performance in the transient response. As far as the steady state response is concerned, the PSO tuned LQG and LQR have better response.

Table 4. 11: The performance of Manual and PSO tuned LQR Versus manual and PSO tuned LQG (acceleration in the Z-axis direction)

Acceleration in Z	Manual tuned LQR	PSO tuned LQR	Manual tuned LQG	PSO tuned LQG
Rise time (t_r)	0.0047	0.0045	0.0074	0.0068
Settling time (t_s)	0.0090	0.0085	0.0136	0.0125
Steady state error (e_{ss})	0.2120	0	-0.0918	0
Peak response (y_p)	0.7880	1	1.0918	1

Therefore, as it has been noted from the above (Table 4.11) the performance of paricle swarm optimization (PSO) tuned LQG and LQR have best performance over the manually tuned LQG and LQR controller in all time domain specifications in both accelerations. Therefore, from the above table we conclude that the rise time, settling time, seady state error and peak response are well improved by using paricle swarm optimization algorithm than manually tuned LQG. Thus, the PSO algorithm give best optimized values of Q and R parameters and also the control objective is achieved with improvement of the overall system.

4.7 Robustness of Linear Quadratic Gaussian (LQG) controller

The goal of any controller design is that the overall system is stable and satisfies desired performance requirements. These requirements should be satisfied at least when the controller is applied to the nominal plant, that is, nominal stability and nominal performance is required. In

practice the real plant is not equal to the model. Even though there are uncertainties in the dynamics of any real system we try to quantify the unmodeled dynamics.

When the controller designed for a nominal plant model is implemented on the real system, there are no guarantees on the resulting performance of the system. Even requirements as basic as stability may not be met. The deviation from the expected behavior of the system clearly depends on the accuracy of the model. Because models are never perfect, robustness analysis is necessary to determine the possibility of instability or inadequate performance in the face of uncertainty of the plant dynamics. A controller should guarantee stability, proper response to commands, and reduction of response perturbations caused by disturbance inputs.

Now, robustness of the controllers designed in the previous chapter will be analyzed. In short, robustness is the measure of how well the controller will perform if it is implemented in the actual Missile guidance system.

Before the analysis of robustness, the cases which may make the system not robust will be discussed. Such cases are known as the model uncertainties. Consequently, the multivarable feedback relation along the process noise and measurement noise will be derived then the robustness of the system will be analyzed using singular value analysis techniques and the different control systems will be compared with one another.

The model uncertainty in the Missile system considered in this thesis can arise due to different cases. Some of the main sources of uncertainty are listed as follows:

- Model parameter uncertainty: - Since perfect system identification is impossible, uncertainties occur in all models of physical systems.
- Neglected high frequency dynamics: - The model behavior at high frequencies cannot be identified, and therefore it is not modeled.
- Non-linearity: - this occurs when we linearize the nonlinear model of missile dynamic equations.
- Changing operating conditions:- changes in environmental conditions (eg.air pressure, Temperature) when the missile motion runs from one atomosphere level to the other.

Previously, no reasonable papers were published in the analysis of robustness of LQG controller. But the one that published in 1978 by famous scholar John C. doyle wrote “guaranteed stability margins for LQG regulators” and has concluded that there are none and the he concluded that there exist LQG combinations with arbitrarily small gain margins [15]. After this many analysis on the robustness of LQG ccontroller is undertaken specially in flight industries.

It is known that closed loop or feedback control systems are more robust and less susceptible to process disturbance or measurement noise. To analyze robustness of a certain closed loop system, the return difference matrix [I+G(s)K(s)] or return ratio matrix [G(s)K(s)] must be identified. Where, G(s) and K(s) are the transfer function matrices of the plant and controller respectively. The larger the return difference of the feedback loop, the greater will be the robustness when compared to the corresponding open-loop system [20].

Consider a linear, time-invariant, multivariable feedback control system with the block diagram shown below in Fig.4.10. that will be analzed along with the controller designed for real usage. The control system consists of a feedback controller with transfer matrix, K(s) and a plant with transfer matrix, G(s), with desired output,Y(s) the Process noise, W(s), and measurement noise,V(s).

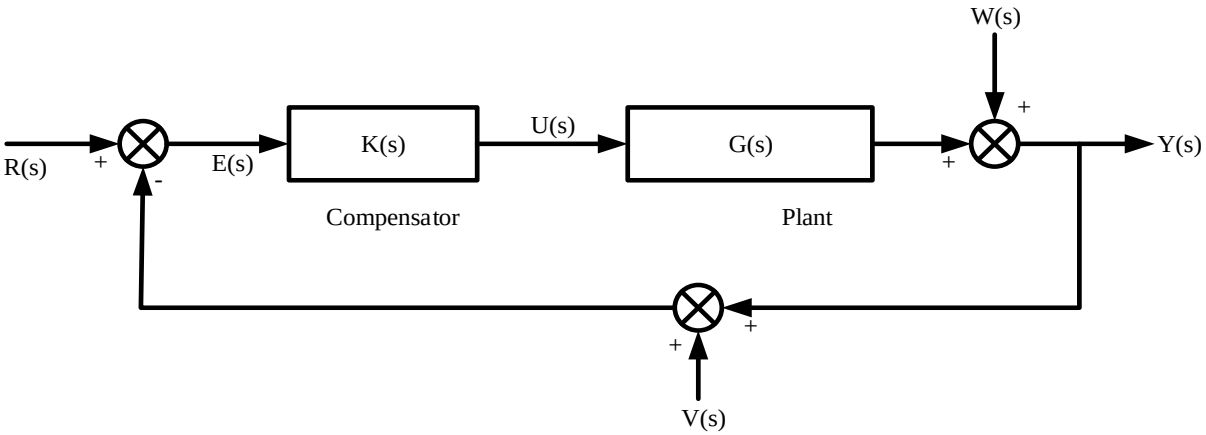


Fig.4. 10: Multivariable feedback control system.

Now, from the above Fig.4.10. the followig guiding equatons in multivariable control system can be derived.

$$Y(s) = G(s)K(s)E(s) + W(s) \quad (4.6)$$

and

$$E(s) = R(s) - [Y(s) + V(s)]$$

Substituting into the eqn. (4.6)

$$Y(s) = G(s)K(s)\,[R(s) - (Y(s) + V(s))] + W(s)$$

$$Y(s) = G(s)K(s)R(s) - G(s)K(s)\,[Y(s) + V(s)] + W(s)$$

$$Y(s) = G(s)K(s)R(s) - G(s)K(s)Y(s) - G(s)K(s)V(s) + W(s)$$

Now, rearranging the above terms

$$Y(s) + G(s)K(s)Y(s) = G(s)K(s)R(s) - G(s)K(s)V(s) + W(s)$$

$$Y(s)[I + G(s)K(s)] = G(s)K(s)R(s) - G(s)K(s)V(s) + W(s)$$

$$Y(s) = [I + G(s)K(s)]^{-1}G(s)K(s)R(s) - [I + G(s)K(s)]^{-1}G(s)K(s)V(s) + [I + G(s)K(s)]^{-1}W(s) \quad (4.7a)$$

Using the matrix identity property

$$[I + G(s)K(s)]^{-1} = I - [I + G(s)K(s)]^{-1}G(s)K(s)$$

Equation 4.7 becomes

$$Y(s) = [I - [I + G(s)K(s)]^{-1}]R(s) - [I - [I + G(s)K(s)]^{-1}]V(s)$$

$$+ [I + G(s)K(s)]^{-1}W(s) \quad (4.7b)$$

The control signal U(s) becomes

$$U(s) = K(s)E(s)$$

But, $E(s) = R(s) – [Y(s) + V(s)]$

and $Y(s) = G(s)U(s) + W(s)$

Now, substituting E(s) into U(s)

$$U(s) = K(s) [R(s) – Y(s) – V(s)]$$

and substituting Y(s) into the above equation

$$U(s) = K(s)R(s) – K(s)[G(s)U(s) + W(s)] – K(s)V(s)$$

$$U(s) = K(s)R(s) – K(s)G(s)U(s) – K(s)W(s) – K(s)V(s)$$

Rearranging the same terms, we get

$$U(s) + K(s)G(s)U(s) = K(s)R(s) – K(s)W(s) – K(s)V(s)$$

$$U(s)[I + K(s)G(s)] = K(s)R(s) – K(s)W(s) – K(s)V(s)$$

$$U(s) = [I + K(s)G(s)]^{-1}K(s)R(s) - [I + K(s)G(s)]^{-1}K(s)W(s) –$$

$$[I + K(s)G(s)]^{-1}K(s)V(s) \quad (4.8)$$

Sensitivity: $\mathbf{S} = [I + G(s)K(s)]^{-1}$ (4.9)

Complementary Sensitivity: $\mathbf{T} = [I - [I + G(s)K(s)]^{-1}]$ (4.10)

Thus, from the above Eqns.(4.7b) and (4.8) one can obsereve that the sensitivity of the output of the system, Y(s) with respect to the process disturbance, w(s) and measurement noise, v(s) depend on the sensitivity function given by $[I + G(s)K(s)]^{-1}$ and also the sensitivity of the input control signal, U(s) depends on the sensitivity function $[I + K(s)G(s)]^{-1}$. The larger the elements of these two matrices, the larger will be the sensitivity of the output and input to process and measurement noise. Robustness is inversely proportional to sensitivity. Therefore the robustness of the output is measured by the matrix [I+G(s)K(s)] called the return difference matrix at the output. Similarly, robustness of the input is measured by the matrix [I+K(s)G(s)] called the return difference matrix at the input.

Note: The return difference matrix [I+G(s)K(s)] is the inverese of sensitivity.

Return ratio matrix can be defined to analyze robustness [20]. Return ratio matrix is the product of the transfer function matrix of the plant and the controller. Return ratio matrix at the output is defined as G(s)K(s) and return ratio matrix at the input is defined as K(s)G(s).

To simplify the measure of robustness, we can assign scalar measure of robustness than dealing with return difference matrix or return ration matrix. In order to accomplish this we can assign the scalar values by defining matrix norm.

$$\|\mathrm{M}\| = \delta$$

M is a complex matrix with **n** rows and **m** columns. δ is positive square root of the eigen value of the matrix $\mathrm{M^H M}$ if $\mathrm{n} \geq \mathrm{m}$ or $\mathrm{MM^H}$ if $\mathrm{n} \leq \mathrm{m}$. The term $\mathrm{M^H}$ is the hermitian of M defined by the transpose of the complex conjugate of M. The ascending order of the positive square roots of the eigen values $\mathrm{M^H M}$ or $\mathrm{MM^H}$ are called the singular values of M.

The singular value decomposition (SVD) algorithm gives the singular value of a matrix M.

Any matrix M can be written as

$$\mathrm{M} = \mathrm{U\Sigma V^H}$$

U = eigen vectors of $\mathrm{M^H M}$ and it is unitary matrix ($\mathrm{UU^*} = \mathrm{I}$).

V = eigen vector of $\mathrm{MM^H}$ and it is unitary matrix ($\mathrm{VV^*} = \mathrm{I}$).

$\mathbf{\Sigma} = \sqrt{diag(eig(\mathrm{M^H M \text{ or } MM^H}))}$ i.e the singular values of M along the diagonal and zeros everywhere else.

Singular values are very important to analyze property of multivariable feedback system. In the analysis of robustness the largest and smallest singular values of the return difference matrix as providing upper and lower bounds on the scalar return difference matrix. Singular values also extends the SISO Bode magnitude plot into the MIMO (the plot versus frequency of the singular values of the transfer-function matrix). Which has very important role in the analysis of MIMO systems.

4.7.1 Robustness of Designed Controllers at the Output

For analyzing robustness, we can treat the largest and smallest singular values of a return difference (or return ratio) matrix at the input and output as providing the upper and lower bounds on the scalar return difference (or return ratio).

To maximize robustness with respect to the process noise, it is clear from Eqn.(4.7b) that we should minimize the singular values of the sensitivity matrix, $[\mathrm{I} + \mathrm{G(s)K(s)}]^{-1}$,which implies minimizing the largest singular $\delta_{max}[\,[\mathrm{I} + \mathrm{G(s)K(s)}]^{-1}]$ or maximizing the singular values of the return difference matrix at the output, i.e. maximizing $\delta_{min}[[\mathrm{I} + \mathrm{G(s)K(s)}]^{-1}]$, The latter requirement is equivalent to maximizing the smallest singular value of the return ratio at the output, $\delta_{min}[\mathrm{G(s)K(s)}]$.

Similarly, minimizing the sensitivity to the measurement noise requires minimizing the largest singular value of the matrix $[\mathrm{I} - [\mathrm{I} + \mathrm{G(s)K(s)}]^{-1}]$ which is equivalent to minimizing the largest singular value of the return ratio at the output, $\delta_{max}[\mathrm{G(s)K(s)}]$.

The following conditions on the singular values of the return ratio at output result when we analyze robustness with respect to process disturbance w(s) and measurement noise v(s) independently,

I. To maximize robustness with respect to process disturbance w(s) δ_{min}[G(s)K(s)] should be maximized.
II. To maximize robustness with respect to measurement noise v(s) δ_{max}[G(s)K(s)] should be minimized.

4.7.2 Loop Gain Limitations and Solutions

Sensitivity versus Complelnentary Sensitivity: In the above the sensitivity function **S**, Eqn.(4.9), and the complementary sensitivity function **T**,Eqn.(4.10) were introduced. To prevent disturbances and minor modeling errors from affecting the output too much, the sensitivity function S should be small.

To minimize the influence of measurement noise and to prevent modeling errors from destroying stability, the complementary sensitivity T should be small. But, there is obviously a severe conflict here since

$$\mathbf{S + T = I}$$

We conclude that S and T cannot both be small at the same frequency. (They can however both have a large magnitude, since complex nunlbers are added.) In most cases the conflict is resolved by requiring [21]:

- **S** to be small at low frequencies (and possibly some other frequency where disturbances are concentrated).
- **T** to be small at high frequencies.

Now, the conditions stated in (I) and (II) conflict to each other as I descibed. However, measurement noise usually has predominantly high frequency content (i.e.more peaks in the power spectrum at high frequencies). Therefore, we can threat robustness for process disturbance and measurement noise in low frequency and high frequency respectively. Thus, it is compromised by maximizing δ_{min}[G(s)K(s)] at low frequency and minimizing δ_{max}[G(s)K(s)] at high frequency, [14], [20], [21].

To analyze robustness, the transfer function value of the return ratio matrix at nominal frequencies is calculated. The frequency range selected for the robustness analysis of this missile system is 0.001rad/s to 1000 rad/s. Thus, the nominal frequencies are selected as low (0.001 rad/s), medium (10 rad/s) and high frequency (1000 rad/s). The transfer function value of the return ratio matrix is calculated by the inbuilt MATLAB^{Tm} function ***M=evalfr(sys,w) (****evalfr* imples evaluate at freqency *w)* , where 'sys' represents the transfer function and '*w*' represents the frequency at which the values are evaluated.

Now, evaluating the return difference matrix values for maually and PSO tuned LQR, LQG controller in different ferquncies, we get:

Table 4. 12: Manual and PSO tuned LQR,LQG values of G(jw)K(jw) for different frequncies

Frequncies	Manual tuned LQR	PSO tuned LQR
w1 = 0.001rad/sec	$M1 = \begin{bmatrix} 0.9994 & 0 \\ 0 & 0.9994 \end{bmatrix}$	$M1 = \begin{bmatrix} 0.9994 & 0 \\ 0 & 0.9993 \end{bmatrix}$
w2 = 10rad/sec	$M2 = \begin{bmatrix} 0.9890 & 0 \\ 0 & 0.9890 \end{bmatrix}$	$M2 = \begin{bmatrix} 0.9897 & 0 \\ 0 & 0.9885 \end{bmatrix}$
w3 = 1000rad/sec	$M3 = \begin{bmatrix} 0.7414 & 0 \\ 0 & 0.7414 \end{bmatrix}$	$M3 = \begin{bmatrix} 0.7368 & 0 \\ 0 & 0.7193 \end{bmatrix}$

Frequncies	Manual tuned LQG	PSO tuned LQG
w1 = 0.001rad/sec	$M1 = \begin{bmatrix} 252.6756 & 0 \\ 0 & 193.3750 \end{bmatrix}$	$M1 = \begin{bmatrix} 294.6463 & 0 \\ 0 & 210.1519 \end{bmatrix}$
w2 = 10rad/sec	$M2 = \begin{bmatrix} 16.4696 & 0 \\ 0 & 12.6471 \end{bmatrix}$	$M2 = \begin{bmatrix} 18.6302 & 0 \\ 0 & 13.4376 \end{bmatrix}$
w3 = 1000rad/sec	$M3 = \begin{bmatrix} 0.4799 & 0 \\ 0 & 0.2511 \end{bmatrix}$	$M3 = \begin{bmatrix} 0.5127 & 0 \\ 0 & 0.2537 \end{bmatrix}$

In the above by evaluating the return ratio matrix, a complex matrix **M** is obtained. After this using the singular value decomposition, the complex matrix **M** is decomposed and the singular values are obtained.The commands ***Σ= svd(M)***(*svd* is singular value decomposition ***Σ***is singular value) or ***Σ= sqrt(eig(M'*M))*** (*sqrt* implies square root and *eig* is eigen value) can be used to evaluate the singular values.

In the case of the Missile system since it has 2-outputs, the singular value matrix contains two values (δmin, δmax). Based on these values we can analyze the robustness of the control systems in different frequency ranges. The singular values of the manually and PSO tuned LQR and LQG is evaluated using the above MATLAB^{TM} commands. and the following results are obtained.

Table 4. 13: The singular values of the designed controllers for different frequncies

Design Methods	w = 0.001rad/sec		w = 10rad/sec		w = 1000rad/sec	
	δ_{min}	δ_{max}	δ_{min}	δ_{max}	δ_{min}	δ_{max}
Manual tuned LQR	0.9994	0.9994	0.9890	0.9890	0.7414	0.7414
PSO tuned LQR	0.9993	0.9994	0.9885	0.9897	0.7193	0.7368
Manual tuned LQG	193.3750	252.6750	12.6471	16.4696	0.2511	0.5127
PSO tuned LQG	210.1519	294.6463	13.4376	18.6308	0.2537	0.4799

In the above, I breifly descibed that the output robustness is analyzed with respect to process disturbance, w(s) and measurement noise, v(s). Since the process disturbance, w(s) is predominant in the lower frequency range, the singular values obtained at the lower frequency range will be used for analysis. Whereas, the measurement noise, v(s) is observed well at the higher frequency range. Thus the singular values at the higher frequency range will be used for analysis.

As, it has been noted in (Table 4.13) the output robustness of the desgned controllers with respect to the process disturbance is compared. In the results, the one with largest δ_{min} has a better robustness. Thus it can be observed that LQG has a better robustness compared to LQR.

The PSO tuned LQG has the best robustness to process disturbance compared to the other three controllers. The manually tuned LQG has the next better robustness while manually tuned LQR and PSO tuned LQR follow.

In the LQG design, the particle swarm optimization (PSO) algorithm tuning has managed to improve the robustness of the manually tuned LQG. Although in the case of the LQR design, the manually tuned LQR has better robustness. Therefore, the output of the system with PSO tuned LQG is least sensitive to process disturbance. For the case of LQR design, the manually tuned LQR is less sensitive to process disturbance or model uncertainties compared to PSO tuned LQR.

For output robustness with respect to measurement noise, the LQG controllers tend to have better measurement noise rejection. For better measurement noise rejection the system must have the smallest δ_{max} at higher frequency range. As it can be observed from the table the LQG controllers have very small values of δ_{max} which shows that the LQG designed has better noise rejection capability compared to LQG. This is due to the ability of the kalman filter being able to extract the states from a noisy measurement signal.

In comparing manual tuning with PSO tuning, in both LQG and LQR case the PSO tuned LQR and LQG have better robustness to measurement noise than manually tuned LQG and LQR. Which implies, the particle swarm optimization algorithm based tuning of the weighting matrices Q and R has managed to improve the measurement noise rejection capability

4.7.3 Robustness of Designed Controllers at the Input

For designing a control system in which the input vector is the least sensitive to process and measurement noise, we can derive appropriate conditions on the singular values of the return ratio at the plant's input, [K(s)G(s)]. In equation 4.8 the sensitivity matrix and return difference matrix of the input control signal U(s) have been defined as $[\mathrm{I} + \mathrm{K(s)G(s)}]^{-1}$ and [I+K(s)G(s)] respectively.

For the robustness of the system input with respect to process disturbance, w(s) and measurement noise, v(s) the singular value of the sensitivity matrix should be minimized. This implies to, minimizing the largest singular value δmax of the sensitivity matrix. In another terms the return difference matrix can be maximized which is maximizing the minimum singular value in the return difference matrix [20].

The following conditions on the singular values of the return ratio at input result when we analyze robustness with respect to process disturbance, w(s) and measurement noise, v(s).

I. The input robustness with respect to process noise or model uncertainty is improved if δ_{min}[K(s)G(s)] is maximized.
II. The input robustness with respect to measurement noise is improved if δ_{max}[K(s)G(s)] is maximized.

Thus, there is no conflict in achieving the robustness of the plant's input to process and measurement noise.

Following similar procedures as the output robustness, by analyzing the return ratio matrices at the nominal frequencies and then evaluating the singular values the robustness can be compared.

Table 4. 14: The singular values of the designed controllers for different frequncies.

Design Methods	w = 0.001rad/sec		w = 10rad/sec		w = 1000rad/sec	
	δ_{min}	δ_{max}	δ_{min}	δ_{max}	δ_{min}	δ_{max}
Manual tuned LQR	0.9994	0.9994	0.9890	0.9890	0.7414	0.7414
PSO tuned LQR	0.9993	0.9994	0.9885	0.9897	0.7193	0.7368
Manual tuned LQG	193.3750	252.6750	12.6471	16.4696	0.2511	0.5127
PSO tuned LQG	210.1519	294.6463	13.4376	18.6308	0.2537	0.4799

As it has been noted in (Table 4.14) the input robustness of the desgned controllers with respect to the process disturbance is compared. In the results, the one with largest δ_{min} has a better robustness. Thus it can be observed that PSO tuned LQG has a better robustness compared to manually tuned LQG. Whereas, in the case of LQR the manually tuned LQR has a better robustness to process disturbance compared to the PSO tuned LQR.

For input robustness with respect to measurement noise, the manually and PSO tuned LQR controllers tend to have better robustness over the manually tuned LQG and PSO tuned LQG controllers.

This result directly justifies the loop shaping approach (Loop Transfer recovery), which states that to recover the robustness of Linear quadratic gaussian (LQG) controller at the input we must minimize the kalman filter uncertainty to approach the LQR robustness at the input. Because an LQR-controlled system has good robustness at the plant inputs [13].

4.7.4 Loop Transfer Recovery at the Input

From the above discussion the linear quadratic gaussian has good robustness over the linear quadratic regualator at the output. But, this is not true at the input. So, we must recover the robustness of the manually tuned LQG controller (the robustness is not good compared to the others) at the input. To recover the of robustness of linear quadratic gaussian (LQG) controller let us see how robust such a compensator is by studying the return ratio at the plant'sinput.

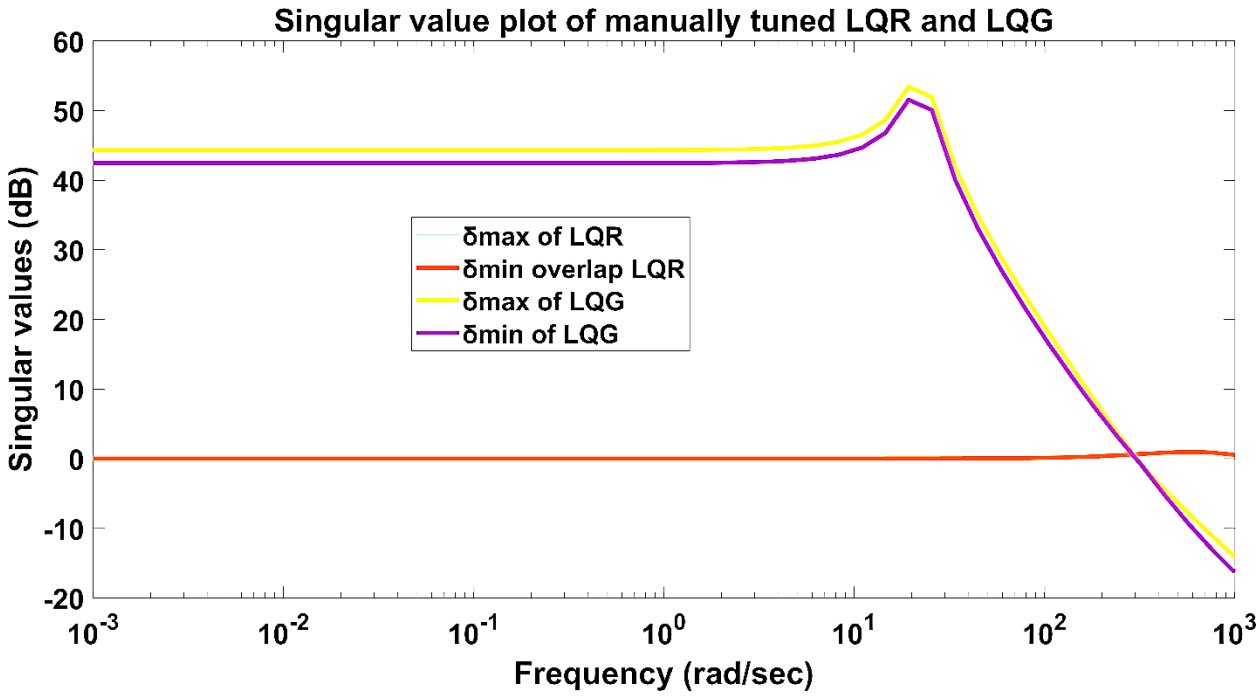


Fig.4. 11: Singular values of the return ratio matrix,K(s)G(s),at the plant's input of compensated system and full-state feedback system. with process noise spectral density w = 1300.

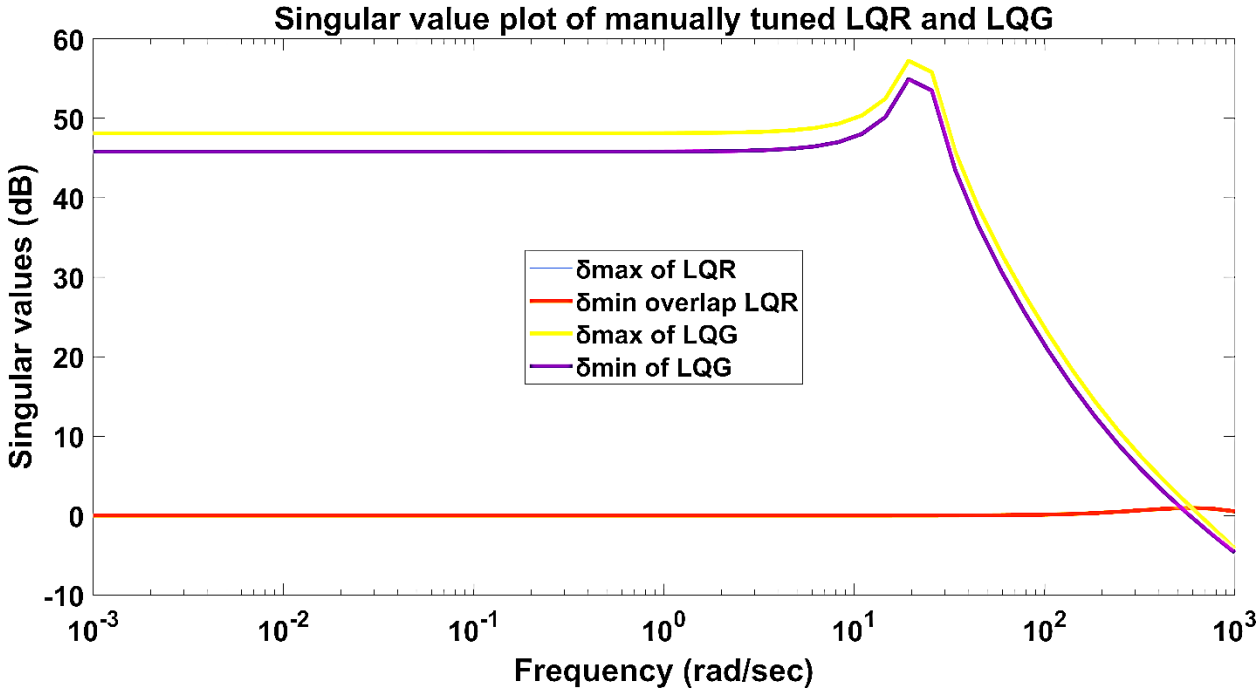


Fig.4. 12: Comparison of singular values of return ratio matrix at the plant's input of linear quadratic regulator and LQG controller for loop transfer recovery with process noise spectral density w = $13*10^5$.

From the above Fig.4.11 that there is a large difference between the smallest singular values of the linear quadratic gaussian controller and linear quadratic regulator systems, indicating that the LQG

controller system is much less robust than the LQR. For recovering the linear quadratic gaussian controller robustness at the plant's input, we re-design the Kalman filter using $w_d = \rho I$,where ρ is a scaling parameter for the process noise spectral density. As ρ is increased, say, from 1300 to $13*10^5$, the return ratio of the linear quadratic gaussian (LQG) controller approaches that of the linear quadratic regulator controller over a larger range of requencies, as seen in the singular value plot of Fig. 4.12.

The re-designed Kalman filter with w = $13*10^5$ recovers the robustness of the linear quadratic gaussian (LQG) controller in some extent at the input i.e. the minimum singular value increases from -17.345dB to -5.625dB which approaches to the minimum singular value of linear quadratic regulator which implies the robustness of linear quadratic gaussian controller recovered in as the process noise covariance increases, but there is gurantee for the performance of the missile system. By choosing a large value of the process noise spectral density for loop transfer recovery, the Kalman filter poles comes very close to the imaginary axis and becomes the dominant pole configuration, thereby playing havoc with the performance. Hence, there is a contradiction in recovering both performance and robustness with the same Kalman filter, and a compromise must be made between the two.

4.7.5 Gain Margin and Phase Margin

The singular values of the return difference matrix at the output in the frequency domain, δ[I+G(jw)K(jw)], can be used to estimate the gain and phase margins of a multivariable system. By plotting singular value plots against frequency, for all frequencies, ω, in the frequency range of interest.

The way in which multivariable gain and phase margins are defined with respect to the singular values is as follows: take the smallest singular value, δ_{min}, of all the singular values of the return difference matrix at the output, and find a real constant, a,such that $\boldsymbol{\delta_{min}[I + G(jw)K(jw)] > a,}$ for all frequncies, $\boldsymbol{w}$ in the frequncy range of interest.

The gain and phase margins can be defined as follows:

$$Gain\ Margin = 1/(1 \pm a) \tag{4.11}$$

$$Phase\ Margin = \pm 2 \sin^{-1}\left(a/2\right) \tag{4.12}$$

The singular value spectrum of the return ratio at the plant's input for frequency range $10^{-3}\ to\ 10^3$ rad/sec is calculated and ploted in MATLAB™.

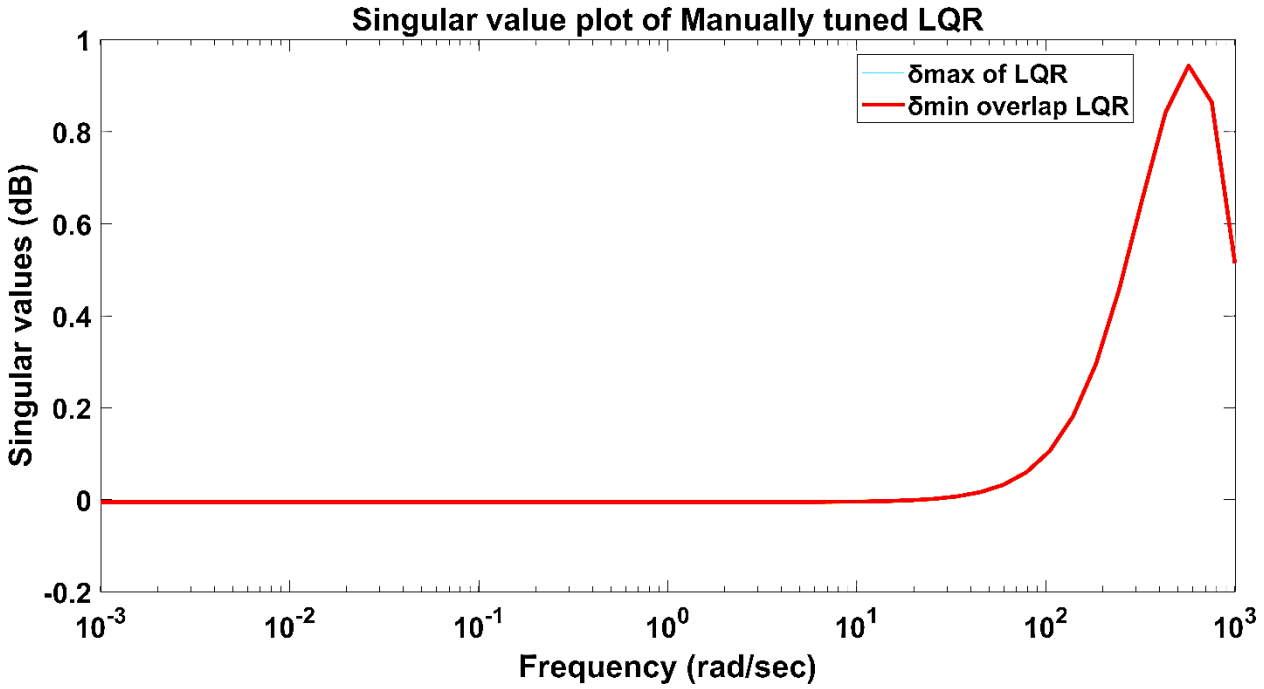


Fig.4. 13: Singular values of the return ratio matrix, K(s)G(s) of manually tuned LQR.

From the above, Fig.4.13 shows the overlapped singular values of the return difference ratio of the Missile control systems. It can be observed that the smallest singular value of manually tuned LQR reaches minimum of 0.5625dB at frequency of 1000 rad/sec.

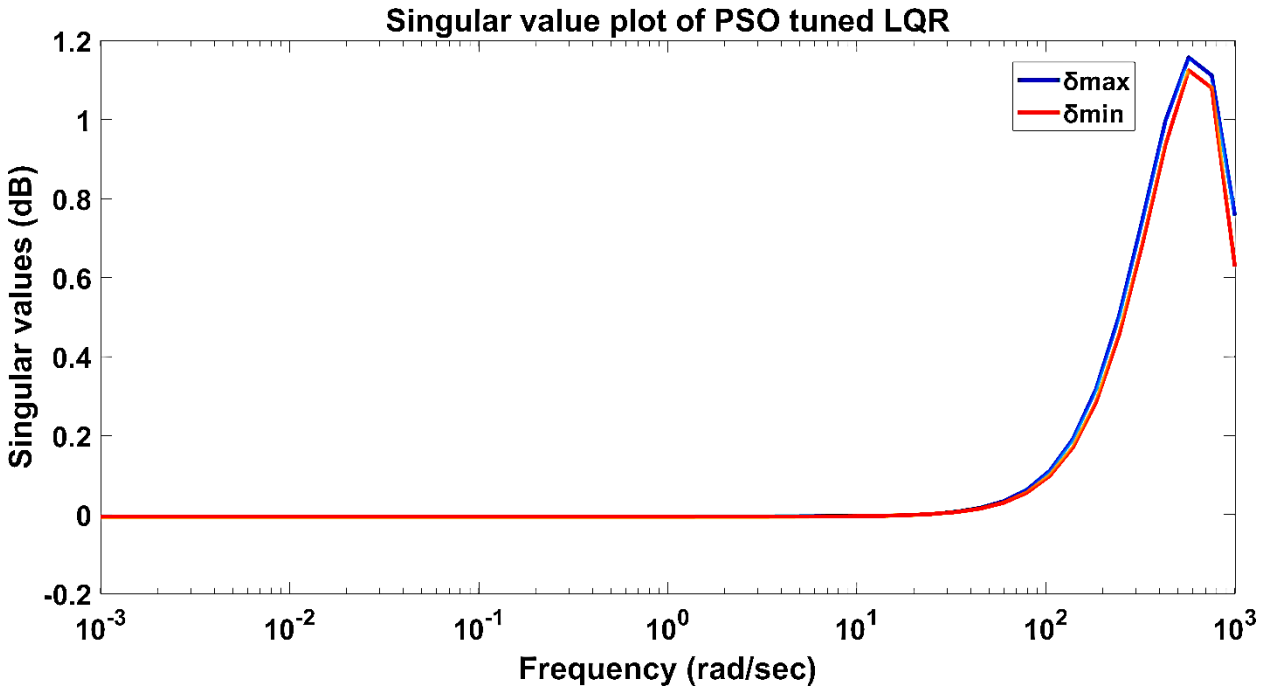


Fig.4. 14: Singular values of the return ratio matrix, K(s)G(s) of PSO tuned LQR.

From the above, Fig.4.14 shows the singular values of the return difference ratio of the Missile control systems. It can be observed that the smallest singular value of manually tuned LQR reaches minimum of 0.6223dB at frequency of 1000 rad/sec.

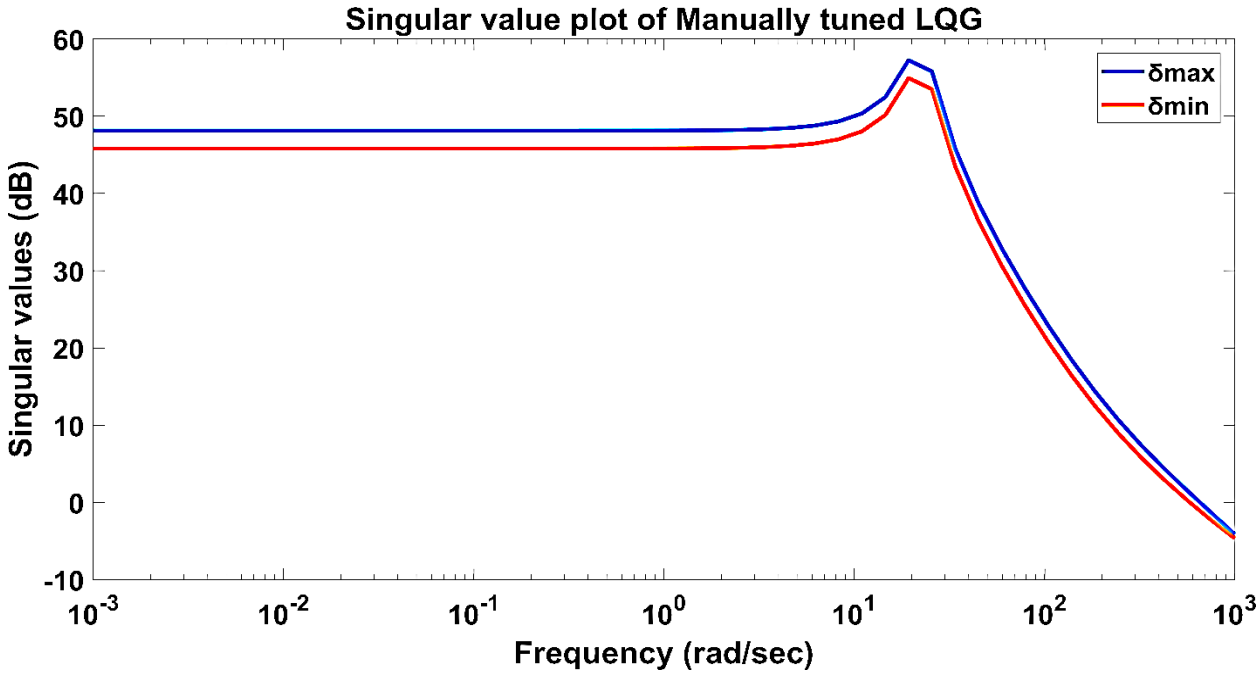


Fig.4. 15: Singular values of the return ratio matrix, K(s)G(s) of manually tuned LQG.

From the above, Fig.4.15 shows the singular values of the return ratio of the Missile control systems. It can be observed that the smallest singular value of manually tuned LQG reaches minimum of -5.625dB at frequency of 1000 rad/sec.

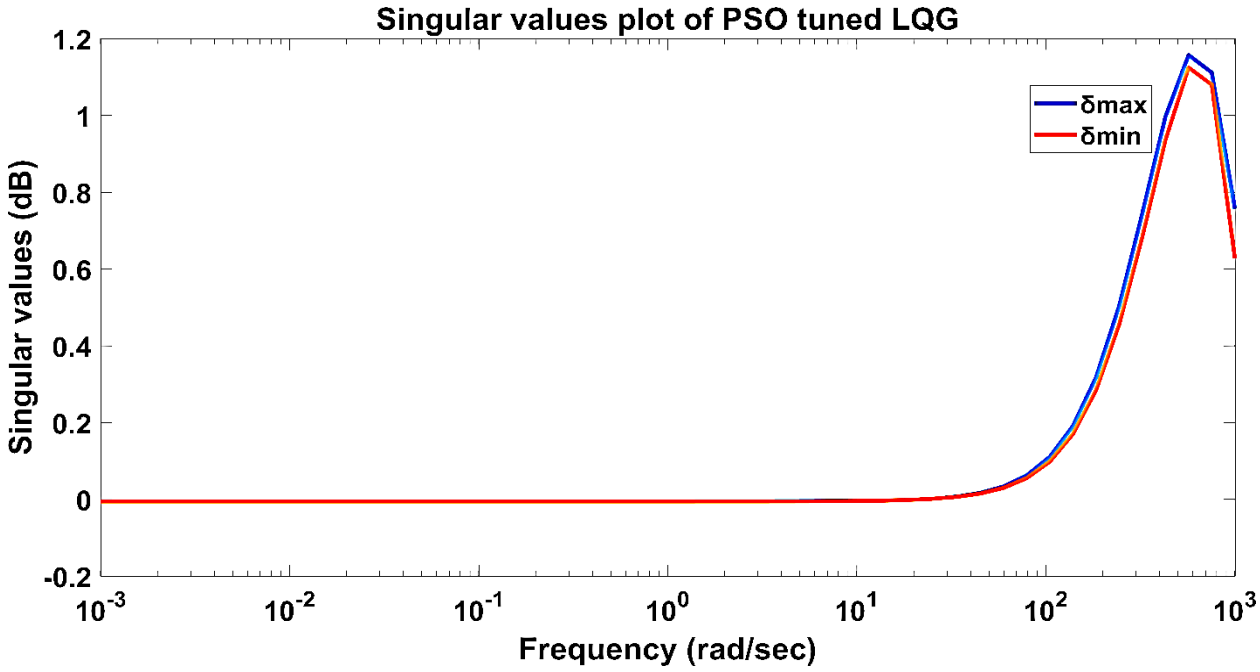


Fig.4. 16: Singular values of the return ratio matrix, K(s)G(s) of PSO tuned LQG.

From the above, Fig.4.16 shows the singular values of the return difference ratio of the Missile control systems. It can be observed that the smallest singular value of PSO tuned LQR reaches minimum of 0.625dB at frequency of 1000 rad/sec.

After the minimum singular value in dB is identified, the values are converted from dB scale to normal magnitude measure. Given the relation $a = 10^{(\delta dB/20)}$. Where, δdB is the minimum singular value reached.

Table 4. 15: Gain margin and Phase margin of the four controllers

Designed controllers	δ_{min}dB	a	Gain Margin (dB)	Phase Margin (deg)
Manually tuned LQR	0.5625	1.0669	0.4838	64.47778
PSO tuned LQR	0.6223	1.0741	0.4821	64.96608
Manually tuned LQG	-5.625	0.5229	0.6566	30.3122
PSO tuned LQG	0.6250	1.0746	0.4820	65.0001

As it hase been observed in (Table 4.15) the gain margin (GM) of manually tuned LQG exceeds the GM of the other three controllers. On the other hand, concerning on the tuning methodologies, the particle swarm optimization (PSO) tuning has relatively a better pahse margin (PM) in both LQR and LQG case compared to manual tuning.

The present analysis indicates that the control system for the Missile control system can tolerate an appreciable variation in the phase of the return difference matrix. Therefore, the above designed controllers have good phase margins except the manually tuned LQG controller.

The narrow range of gain margin indicates that the control system for the missile control system cannot tolerate an appreciable variation in the gain of the return difference matrix [G(s)K(s)] before its eigen value cross into the right half s-plane.

CHAPTER 5

5. CONCLUSION AND RECOMMENDATION

5.1 Conclusions

In this thesis, a Linear Quadratic Gausian Controller is developed for the model of surface-to-air missile that has been developed with some assumptions and it has been shown that, the system is 2-input and 2-output MIMO system. Before the controllers are designed the appropriate control loop was selected by the RGA analysis. From the RGA analysis it has been found that the system is diagonal i.e.u_1 controls y_1 and u_2 controls y_2 without any contol signal interaction. Indeed,the condition number of the missile is calculated and it has small value i.e. "1" which implies there is no control problem for plant.

Linear quadratic gaussian (LQG) and linear quadratic regulator (LQR) are designed by tuning the weighting matrices manually and using an intelligent procedure particle swarm optimization (PSO) algorithm. From the designed controllers the PSO tuned LQR and LQG has shown a better performance over the manually tuned LQR and LQG controllers.

By using the LQG controller, the desired performance is achieved. The Kalman filter was used to minimize the effect of the disturbances introduced both in the plant and sensors. The followings were the main results obtained.

- The PSO tuned LQR controller has better performance over the manually tuned LQR controller in both steady state error and peak response along with Y axis acceleration.
- The PSO tuned LQR controller has better performance over the manually tuned LQR controller in all time domain specifications i.e. rise time, settling time, steady state error and peak response along with Z axis acceleration.
- The PSO tuned LQG controller has better performance over the manually tuned LQG controller in all time domain specifications i.e. rise time, settling time, steady state error and peak response along with both Y and Z axis accelerations
- The PSO tuned LQG controller has better robustness over the manually tuned LQG controller at the output. which implies the PSO tuned LQG is least sensitive to process disturbance or model uncertainties.
- The appreciable variation in the gain of the return difference matrix [G(s)K(s)] that the system can tolerate before it becomes unstable in the case of PSO and manually tuned LQR has a value of 0.4821dB and 0.4838 respectively. In the case of PSO and manually tuned LQG the gain is 0.4820 and 0.6566 which is quite very small.
- The appreciable variation in the phase of the return difference matrix [G(s)K(s)] that the system can tolerate before it becomes unstable in the case of PSO and manually tuned LQR have a value of 64.9660^0 and 64.4778^0 respectively. In the case of PSO tuned LQG the PM is 65.001^0. In this case those controllers have good phase margin value.
- The weighting matrices (Q and R) of the LQR design have small effect in improving the robustness of the LQG.

- In fact the PSO tuning slightly improved the robustness of both LQG and LQR in the measurement noise than manually tuned LQG and LQR. Indeed, the PSO tuned has also improved the robustness in process disturbance.
- The PSO tuned design has removed the tedious process of selecting the elements of the weighting matrices in trial and error method which is the concern of this thesis and perfectly removed the manually method.
- Loop transfer recovery is applied at the plant input to recover the robustness of linear quadratic gaussian (LQG) controller. But, the robustness is recovered in some extent.
- The minimum singular value of LQG increases from -17.345dB to -5.625dB which approaches to the minimum singular value of linear quadratic regulator which implies the robustness of linear quadratic gaussian controller is altered.
- At the last, LQG in both tuning methodologies is not the best controller for the Missile control system as far as the robustness of the controllers is concerned.

5.2 Recommendations

In spite of, which tuning method was followed, the robustness of the LQG regulator is not on the expected level. In order to make LQG more robust controller for the Missile system, the plant model should be modeled well, which is close enough to the actual plant model. This is achieved by modeling the system considering all the physical phenomena's that could possibly be present in the Missile system.

To design the kalman fiter the process and measurement covariance matrices are insereted using trial and error method which is very onerous. As far as this covariance drops, the estimate errors becomes large. Therefore, optimization method or another method that finds the best estimate of the noise and that filters out the noise shall be used.

The use of more sophisticated techniques such as $\mathbf{H_2}$, $\mathbf{H_\infty}$ optimal control and $\boldsymbol{\mu}$-synthesis may allow the designer to meet the design specifications as well as all of the performance and robustness tests.

REFERNCES

[1] G. M. Siouris, *Missile Guidance and Control Systems*, Springer, First Edition., vol. 57, no. 6 2004.

[2] B. Ozkan, "Dynamic Modeling Guidance and Control of Homing Missile," *Thesis*, September, 2005.

[3] A. Porter, "Guided missiles," *Nature*, vol. 177, no. 4518, 1990.

[4] R. Yanushevsky, *Guidance of Unmanned Aerial Vehicles*, Taylor & Francis Group LLC, First Edition, 2011.

[5] Raziye Tekin, "Design, Modeling, Guidance and Control of a Vertical Launch Surface to Air missile," *Thesis* September, 2010.

[6] S. Das and K. Halder, "Missile attitude control via a hybrid LQG-LTR-LQI control scheme with optimum weight selection," *1st Int. Conf. Autom. Control. Energy Syst. - 2014, ACES 2014*, 2014.

[7] F. Jimenez, "*Linear quadratic Gaussian controller design using loop transfer recovery for a flexible missile model* ", Monterey, california, Naval postgraduate School, 1993.

[8] J. L. Meriam. and L.G.Kraige, *Engineering Mechanics And Statics*, John Wiley & Sons,Inc., 5th Edition, 2002.

[9] F. A. Faruqi and T.Lan.Vu, "Mathematical Models for a Missile Autopilot Design," *Defense. School. DSTO Systems Sciences Laboratory*, 2002.

[10] E.H.J. pallet, AMRAes, and S.Coyle, *Automatic Flight Control*, Blackwell Science Ltd., Fourth Edition, 2007.

[11] F. A. Faruqi, "The Algebraic Structure of Quadratic and Bilinear systems," *Defense School, DSTO Systems Sciences Laboratory*, 2003.

[12] Y. Chen, P. Atherton and D. Xue, *Linear Feedback Control Analysis and Design with MATLAB*, the society for industrial and applied mathematics, First Edition, 2007.

[13] Sigurd Skogestad, *Multivariable feedback control Analysis and design*, John Wiley & Sons, Second Edition, Aug 29, 2001.

[14] V.L.Syrmos, Frank L. Lewis, Draguna Vrabie, *Optimal Control*, John Wiley & Sons Inc., Third Edition, no. 2. 2012.

[15] J. Doyle, *Guaranteed margins for LQG regulators*, IEEE Transactions on Automatic Control, Volume: 23, Issue: 4, Aug 1978.

[16] Y. S. Russell C. Eberhart, *Particle swarm optimization : Development , applications and resources*, February, 2015.

[17] M. Clerc, *Particle Swarm Optimization*, ISTE, Ltd, First Edition, 2006.

[18] A. Lazinica, *Particle Swarm Optimization*, Croatia: In- Tech, First Edition, 2009.

[19] C. Chen, *Linear System Theory*, Oxford university press.Inc, Third Edition, 1999.

[20] A. Tewari, *Modern Control Design With Matlab And Simulink*. Kanpur, India, John Wiley & Sons Inc., First Edition, 2002.

[21] T. Glad. and L. Ljung, *Multivariable and Nonlinear Methods*, London: Taylor & Francis, First Edition, 2000.

APPENDICES

Appendix A: Missile system Parameters

$$Y_v = -3,\ Y_\zeta = 180,\ N_v = 1,\ N_r = -3,\ N_\zeta = -500,$$

$$Z_w = -3,\ Z_\eta = -180,\ M_w = -1,\ M_q = -3,,\ M_\eta = -500,$$

$$u_0 = 500,\ d_x = 0.5,$$

$$X_u = X_p = X_{xi} = X_{eta} = X_{zeta} = Y_p = Y_r = Y_{xi} = Y_{eta} = 0$$

$$L_u = L_p = L_{xi} = M_u = M_v = M_{zeta} = N_w = N_p = N_{xi} = N_{eta} = 0$$

$$Z_q = Z_r;\ Z_{eta} = -Y_{zeta};$$

Appendix B: MATLAB^{TM} code

```
%% State space model of missile
A = [0 0 0 0 0 0; 0 -3 0 0 0 -500; 0 0 -3 0 500 0; 0 0 0 0 0 0 ; 0 0 -1 0 -3 0; 0 1 0 0 0 -3];
B = [0 0; 0 180; -180 0; 0 0 ; -500 0; 0 -500];
C = [0 0 -3.5 0 1.5 0; 0 2.5 0 0 0 1.5];
D = [0 0; 0 0];
sys = ss(A,B,C,D)
Co = ctrb(A,B); % Checking the controllability of the plant
rank(Co)
if rank(Co)<rank(A);
disp('the system is not controllable')
else
disp('the system is controllable')
end
% check the observability of the system
Ob=obsv(A,C);
rank(Ob)
if rank(Ob)<rank(A);
disp('the system is not observable')
```

```
else
disp('the system is observable')
end
%% The plant is neither controllable nor observable. Thus, minimal realization of the system is done to extract the uncontrollable and unobservable states.
mineral(sys)      % Gives the controllable and observable state space representation
A = [-3 0 0 -500;0 -3 500 0;0 -1 -3 0;1 0 0 -3];
B = [0 180;-180 0;-500 0;0 -500];
C = [0 -3.5 1.5 0;2.5 0 0 1.5];
D = [0 0;0 0];
sys = ss(A,B,C,D);
% open loop step response
Op = step(sys)
title('Open-Loop Step Response')
% Bode plot
bode(sys)
title(‘bode plot of the open loop system')
% Manual LQR design
%% Qi = [0.1 1 10]
Q = [0.1 1 10];
R = eye(2);
for i = 1:numel(Q)
  %   Qi = diag([10*Q(i) 10*Q(i) 10*Q(i) 10*Q(i)]);
   Kc = lqr(A,B,Qi,R);
  syscl = ss(A - B*Kc,B,C,D);
  [Y,t] = step(syscl);
  r = ones(numel(t),2);            % Unit Inputs U1 and U2
  [y,~,~] = lsim(syscl,r,t);       % Simulate time response of dynamic system to arbitrary inputs
  stepResults = stepinfo(Y,t);   % Step response characteristics (Rt, St, others)
```

```
  ayq(i,1) = stepResults(1,1).RiseTime;
  ayq(i,2) = stepResults(1,1).SettlingTime;
  ayq(i,3) = r(1) - y(end,1);          % Steady state error of U1
  ayq(i,4) = stepResults(1,1).Peak;
   azq(i,1) = stepResults(2,2).RiseTime;
  azq(i,2) = stepResults(2,2).SettlingTime;
  azq(i,3) = r(1) - y(end,2);
  azq(i,4) = stepResults(2,2).Peak;
  end
%p.Title.String = 'Step response of LQR with different normalized weighting factors';
%%  Ri = [0.1 1 10]
Q = eye(4)*10;
R = [0.1 1 10];
for i = 1:numel(R)
    Ri = diag([R(i) R(i)]);
    Kc = lqr(A,B,Q,Ri);
  syscl = ss(A - B*Kc,B,C,D);
  [Y1,t] = step(syscl);
  r = ones(numel(t),2);                % Unit Inputs U1 and U2
  [y1,~,~] = lsim(syscl,r,t);          % Simulate time response of dynamic system to arbitrary inputs
  stepResults1 = stepinfo(Y1,t);    % Step response characteristics (Rt, St, others)
  ayr(i,1) = stepResults1(1,1).RiseTime;
  ayr(i,2) = stepResults1(1,1).SettlingTime;
  ayr(i,3) = r(1) - y1(end,1);       % Steady state error of U1
  ayr(i,4) = stepResults1(1,1).Peak;
  azr(i,1) = stepResults1(2,2).RiseTime;
  azr(i,2) = stepResults1(2,2).SettlingTime;
```

```
  azr(i,3) = r(1) -  y1(end,2);
  azr(i,4) = stepResults1(2,2).Peak;
end
%p.Title.String = 'Step response of LQR with different normalized weighting factors';
```

The PSO code is omitted for privacy.