ISLAMIC UNIVERSITY OF TECHNOLOGY, (IUT)

DESIGN AND SIMULATION OF An AIRCRAFT & MISSILE SYSTEM & OPTIMAL TUNING OF BOTH USING PID & DIFFERENTIAL ALGORITHM(DE)

Submitted By

MD.ZAHIDUL ISLAM SHAHIN (082451)

AHMED TASHFIN IFTEKHAR (082459)

TANVIR HASSAN MAHMUD(082473)

A Dissertation
Submitted in Partial Fulfillment of the Requirement for the ***Bachelor of Science in Electrical and Electronic Engineering***
Academic Year: 2011-2012

Department of Electrical and Electronic Engineering
Islamic University of Technology (IUT)
A Subsidiary Organ of OIC. Dhaka, Bangladesh.

CERTIFICATE OF RESEARCH

The thesis title "**DESIGN AND SIMULATION OF An AIRCRAFT & MISSILE SYSTEM & OPTIMAL TUNING OF BOTH USING PID & DIFFERENTIAL ALGORITHM(DE)**" has been accepted as satisfactory in partial fulfillment of the requirement for the Degree of Bachelor of Science in Electrical and Electronics Engineering on October, 2012.

Submitted By

____

Md.Zahidul Islam Shahin (082451)

____

Ahmed Tashfin Iftekhar (02459)

____

Tanvir Hassan Mahmud (082473)

Approved By:

HEAD OF THE DEPARTMENT

____

Prof. Dr. Md. Shahid Ullah

EEE Department ,Islamic University Of Technology

SUPERVISOR

____

Mr. Ashik Ahmed

Assistant Professor , EEE Department, Islamic University Of Technology

ACKNOWLEDGEMENT

First of all, we express our sincere gratitude to the Almighty Allah, Who has kindly given us the energy to accomplish the thesis work.

The project was carried out by the authors under the close supervision of Mr.Ashik Ahmed, Asst. Professor, Department of Electrical and Electronics Engineering, Islamic University of Technology (IUT).

The supervisor had dedicated his valuable time, which had ensured the completion of this project. We express our profound gratitude to the supervisor for his support, co-operation and guidance in accomplishing the project.
Without their co-operation, it would have been very difficult for us to complete the project successfully.

We would like to thank our family who has always been with us showing sympathy understanding & encouragement during the completion of this work. We are also grateful to all of our will-wishers, who provided their perpetual support towards getting this task done successfully.

Lastly, we offer our heartfelt felicitations and compliments to all of those who supported us in any respect during the completion of the thesis work.

ABSTRACT

This thesis tries to develop a model for the missile and for the aircraft simply using those dynamic force model and transfer function and to explore the potential of using soft computing methodologies such as Evolutionary algorithm(EA) –Differential algorithm(DE) in controllers and their advantages over conventional methods. PID controller, being the most widely used controller in industrial applications, needs efficient methods to control the different parameters of the plant. This thesis asserts that the conventional approach of PID tuning is not very efficient due to the presence of non-linearity in the system of the plant. The output of the conventional PID system has a quite high overshoot and settling time.

Two problem has been taken from a paper which has been considered as the dynamic model for missile and aircraft ,based on those problem the project progress.

The main focus of this project is to apply evolutionary algorithm such as differential algorithm to design and tuning of PID controller to get an output with better dynamic and static performance. The application of differential algorithm(DE) to the PID controller imparts it the ability of tuning itself automatically in an on-line process makes it give an optimum output by searching for the best set of solutions for the PID parameters.

The project also discusses the benefits and the short-comings of both the methods. The simulation outputs are the MATLAB results obtained for a step input .

This work is

Dedicated to

My lovely Mother

TABLE OF CONTENTS

CERTIFICATE OF RESEARCH II
ACKNOWLEDGEMENT III
ABSTRACT IV
CONTENTS VI
LIST OF FIGURES IX

Chapter -1 - INTRODUCTION

1.1 Introduction 1
1.2 control system 1
1.3 Advantage of control system 2
1.4 Analysis and Design Objectives 2
1.5 Transient Response 2
1.6 Steady state response 3
1.7 stability 3

Chapter-2- MODELLING OF A SISO SYSTEM

2.1 SISO system 4
2.2 State space modeling 4
2.2.1 General form of state space model 4
2.2.2 State space analysis 6
2.2.3 State space realizations 6
2.2.4 Converting transfer function to state space 6
2.3 Linear and Nonlinear SISO system 7
2.3.1 Linear SISO 7
2.3.2 Nonlinear SISO 8
2.4 SISO SYSTEM MODEL APPROACH 9
2.4.1 State variable model for a dynamic system 9

Chapter-3 -MISSILE DYNAMICS 10

3.1 Missile overview 10
3.2 Category of missile 10
3.2.1 Guided Missile control 10
3.3 Studies on the control of missiles in the literature 11
3.3.1 Guided Missile consideration 11
3.3.2 Missile type-Guided Missile 11
3.4 Missile Model 12
3.4.1 Missile Dynamic Model 12
3.4.2 Missile kinematics 13
3.4.3 Forces on a rocket in flight 13

3.4.4 Missile seeker showing geometry 14
3.4.5 Equations of motion of the Missile 14
3.4.6 Common Missile structure 15
3.5 Missile Modeling problem 16
3.5.1 Problem statement 16
3.5.2 Controlling of Missile 16
3.5.3 Transfer function of SISO 17
3.5.4 State space representation 18
3.5.6 Step response of missile transfer function 19
3.5.7 Analyzing the step response 19
3.5.8 Performance parameter of a system 20
3.6 Second orders system 20
3.6.1 Second order system characteristics 21
3.6.2 second order system response analysis 22
3.7 control system analysis and design focus 23
3.7.1 General description of a controlled system 23
3.7.2 Structure and components 23
3.7.3 General controlled system 24
Chapter-4 PID CONTROLLED SYSTEMS 25
4.1 Introduction and representation 25
4.2 Criteria of performance Evaluation 25
4.2.1 Time domain indices 26
4.3 Conventional PID controllers 26
4.3.1 Preliminary and background 26
4.4 Tuning of PID parameters 27
4.5 Using PID controller on missile control system 27
4.5.1 Output transfer function 28
4.5.2 Tuning via Trial and error method 28
4.5.3 Matlab simulation –PID 28
4.5.4 Step response using PID 29
4.5.5 Improvement of the system 29
Chapter-5 -SIMO SYSTEM 30
5.1 SIMO system 30
5.2 Problem statement 31
5.2.1 Parameter and other constraints 32
5.3 Modeling of a SIMO : Aircraft control system 32
5.4 Aircraft Equations of motion 33
5.5 Laplace transform of dynamic system 33
5.6 Configuration Aerodynamics 33
5.6.1 Configuration variables 33
5.7 Case problem(SISO system) 34

5.8 Simulation for the 1st Output 35
5.8.1 Transfer function via MATLAB 35
5.8.2 Step response of G1(s) 36
5.8.3 Tuning via PID for G1(s) 36
5.8.4 Simulation results 37
5.9 Simulation for the 2nd Output G2(s) 37
5.9.1 Transfer function from Matlab simulation 38
5.9.2 Step response of G2(s) 38
5.9.3 Tuning via PID controller G2(s) 39
CHAPTER-6 -OPTIMIZATION ALGORITHM AND DE 41
6.1 optimization algorithm 41
6.2 What is optimization? 41
6.3 optimization method 41
6.4 Evolutionary Algorithm(EA) 42
6.5 Some optimization algorithm(EA) 42
6.6 Proposed Algorithm –Differential Algorithm(DE) 42
6.6.1 Background and Literature Review 42
6.6.2 Basis of Differential Evolution algorithm 42
6.7 Differential Evolution Algorithm Optimization process 43
6.7.1 Initialization 43
6.7.2 Mutation 44
6.7.3 Crossover 44
6.7.4 Selection 45
6.8 Design of the Differential Evolution Algorithm Optimization Program 45
6.9 Basic parameter of DE optimization 46
6.10 optimization process flowchart 47
6.11 APPLICATION OF DE ON MISSILE SYSTEM(SISO) 48
6.11.1 Tuning of the SISO system via DE 48
6.11.2 Optimizing performance 48
6.11.3 STEP RESPONSE BEFORE DE OPTIMIZATION(SISO) 49
6.12 STEP RESPONSE after DE OPTIMIZATION(SISO) 50
6.12.1 Simulation result(step response using DE) 50
6.12.2 Bode plot 51
6.12.3 Root locus 52
6.12.4 Nyquist diagram 52
6.13 conclusion 53
APPENDIX-A 54 -60
REFERENCES 61-62

LIST OF FIGURES :

Figure 1.1: Simplified description of a control system

Figure 2.1: SISO SYSTEM block diagram

Figure 2.2: Graphic Representation of state space and state vectors

Figure 3.1 :Canard Controlled Missile

Figure 3.2 : Wing Controlled Missile

Figure 3.3 :Tail fins Missile

Figure 3.5 : Missile Geometry

Figure 3.6 : Common Missile structure

Figure 3.7 : Common missile structure

Figure 3.8 : Missile problem

Figure 3.9 : Step response of SISO

Figure 3.10 : General form of a second order system

Figure 3.11 : Characterizing a second order response

Figure 3.12 Response of a 2nd order system

Figure 3.13 : Simplified model of a control system

Figure 4.1: Basic block diagram of a conventional PID controller

Figure 4.2: PID controller

Figure 4.3: Step response after Using PID

Figure 5.1:Typical SIMO System

Figure 5.2: open loop flight control system

Figure 5.4 : Basic configuration of aerodynamics-some definitions

Figure 5.5: step response for G1(s)

Figure 5.6: Tuning of the G1(s)-simulation output using PID

Figure 5.7: step response of G2(s)

Figure 5.8: output of PID controller based G2(s)

Figure 6.1: Optimization process

Figure 6.2 : Flowchart of overall process of DE optimization

Figure 6.3: Step response of SISO before DE optimization

Figure 6.4: Step response of SISO after using DE

Figure 6.5: Bode plot of DE Simulation

Figure 6.6: Root lucas of the SISO system after DE

Figure 6.7: Nyquist Diagram of SISO system

CHAPTER 1

INTRODUCTION

1.1 Introduction

Control systems are an integral part of modern society. Numerous applications are all around us: The rockets fire, and the space shuttle lifts off to earth orbit; in splashing cooling water, a metallic part is automatically machined; a self-guided vehicle delivering material to workstations in an aerospace assembly plant glides along the floor seeking its destination. These are just a few examples of the automatically controlled systems that we can create.

We are not the only creators of automatically controlled systems; these systems also exist in nature. Within our own bodies are numerous control systems, such as the pancreas, which regulates our blood sugar. In time of "fight or flight," our adrenaline increases along with our heart rate, causing more oxygen to be delivered to our cells. Our eyes follow a moving object to keep it in view; our hands grasp the object and place it precisely at a predetermined location. Even the nonphysical world appears to be automatically regulated.

1.2 Control System

A control system consists of *subsystems* and *processes* (or *plants)* assembled for the purpose of obtaining a desired *output* with desired *performance,* given a specified *input.* Figure 1.1 shows a control system in its simplest form, where the input represents a desired output.

.

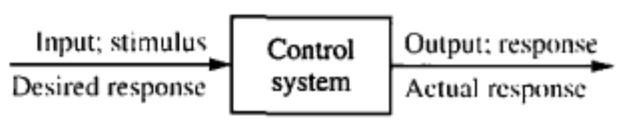


Figure 1.1: Simplified description of a control system

Two major measures of performance are apparent:
(1) the transient response
(2) the steady-state error

1.3 Advantages of Control Systems

With control systems we can move large equipment with precision that would otherwise be impossible. We can point huge antennas toward the farthest reaches of the universe to pick up faint radio signals; controlling these antennas by hand would be impossible.). We alone could not provide the power required for the load and the speed; motors provide the power, and control systems regulate the position and speed.

We build control systems for four primary reasons:

1. Power amplification

2. Remote control

3. Convenience of input form

4. Compensation for disturbances

1.4 Analysis and Design Objectives

Analysis is the process by which a system's performance is determined. For example, we evaluate its transient response and steady-state error to determine if they meet the desired specifications.

Design is the process by which a system's performance is created or changed. For example, if a system's transient response and steady-state error are analyzed and found not to meet the specifications, then we change parameters or add additional components to meet the specifications.

A control system is dynamic: It responds to an input by undergoing a transient response before reaching a steady-state response that generally resembles the input.

1.5 Transient Response

Transient response is important. In the case of an elevator, a slow transient response makes passengers impatient, whereas an excessively rapid response makes them uncomfortable. If the elevator oscillates about the arrival floor for more than a second, a disconcerting feeling can result. Transient response is also important for structural reasons: Too fast a transient response could cause permanent physical damage. We analyze the system for its existing transient response. Finally, we adjust parameters for the getting the desired transient response.

1.6 Steady-State Response

Another analysis and design goal focuses on the steady-state response. As we have seen, this response resembles the input and is usually what remains after the transients have decayed to zero.

1.7 Stability

Discussion of transient response and steady-state error is moot if the system does not have stability. In order to explain stability, we start from the fact that the total response of a system is the sum of the natural response and the forced response.

Natural response describes the way the system dissipates or acquires energy. The form or nature of this response is dependent only on the system, not the input. On the other hand, the form or nature of the forced response is dependent on the input.

Thus, for a linear system, we can write

Total response = Natural response +Forced response (1.1)

For a control system to be useful, the natural response must eventually approach zero, thus leaving only the forced response, or oscillate. In some systems, however, the natural response grows without bound rather than diminish to zero or oscillate. Eventually, the natural response is so much greater than the forced response that the system is no longer controlled. This condition, called instability, could lead to self-destruction of the physical device if limit stops are not part of the design.

Control systems must be designed to be stable. That is, their natural response must decay to zero as time approaches infinity, or oscillate.

CHAPTER-2

MODELLING OF A SISO SYSTEM

2.1 SISO SYSTEM :

SISO refers to single input-single output system. A system has only one single input and only one single output can be modeled as SISO System.

A simple block that can represent a SISO system is :

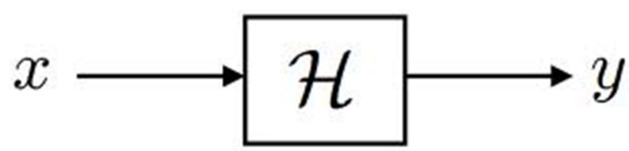


Figure 2.1: SISO SYSTEM block diagram

Where x is the input and y is the output. The transfer function is H.

2.2 STATE SPACE MODELING

2.2.1 General form of a state-space model

.

Linear independence:

A set of variables is said to be linearly independent if none of the variables can be written as a linear combination of the others. For example, given x1, x2, and x3, if x2 ¼ 5x1 þ 6x3, then the variables are not linearly independent, since one of them can be written as a linear combination of the other two.

System variable:

Any variable that responds to an input or initial conditions in a system.

State variables: The smallest set of linearly independent system variables such that the values of the members of the set at time t0 along with known forcing functions completely determine system variables for all t .

State vector : A vector whose elements are the state variables.

State space:. The n-dimensional space whose axes are the state variables. This is a new term and is illustrated in fig :2.2 .

State equations : A set of n simultaneous, first-order differential equations with n variables, where the n variables to be solved are the state variables.

Output equation : The algebraic equation that expresses the output variables of a system as linear combinations of the state variablesand the inputs. Now that the definitions have been formally stated, we define the

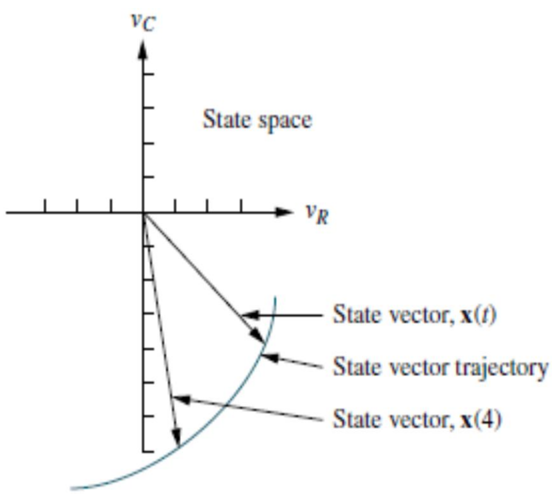


Fig 2.2: Graphic Representation of state space and state vectors

In general, a state-space representation has the form

$\dot{x} = Ax + Bu$ (2.1)

$y = Cx + Du$ (2.2)

where:

x is a column vector of dimension n, called the state vector,
u is a scalar signal, and the input of the system,
y is a scalar signal, and the output of the system,
A is an n × n matrix,
B is a column vector of dimension n,
C is a row vector of dimension n
D is a scalar.

2.2.2 STATE-SPACE ANALYSIS :

Some simple yet general results can be obtained by applying the Laplace transform to the
state-space model. The equations for the state-space model are

$x^{\cdot} = Ax + Bu$ (2.3)

$y = Cx + Du$

so that, in the Laplace domain

$sX(s) - x(0) = AX(s) + BU(s)$ (2.4)

$Y(s) = CX(s) + DU(s)$

where x(0) is the initial condition of the state vector (x(t) at t = 0), and X(s), U(s), and Y(s) are the Laplace transforms of x(t), u(t), and y(t), respectively. The first (vector) equation gives

$sX(s) - AX(s) = (sI - A)X(s) = x(0) + BU(s)$ (2.5)

where I is the identity matrix with dimension n×n. Therefore, the transform of the output is

$Y(s) = C(sI - A)-1x(0)$ (2.6)

2.2.3 STATE-SPACE REALIZATIONS :

A state-space realization for a transfer function H(s) is a set of A, B, C, D such that H(s) =C(sI − A)−1B + D. Before proceeding on this topic, we introduce the following definitions:

- a function of s is called a rational function of s if it is the ratio of two polynomials,
- a rational function of s is called proper if degN(s) _ degD(s),
- a rational function of s is called strictly proper if degN(s) < degD(s),

2.2.4 CONVERTING TRANSFER FUNCTION TO STATE SPACE:

Consider the differential equation

$$\frac{d^n y}{dt^n} + a_{n-1}\frac{d^{n-1}y}{dt^{n-1}} + \cdots + a_1\frac{dy}{dt} + a_0 y = b_0 u \qquad (2.7)$$

A convenient way to choose state variable is to choose the output, $y(t)$, and it's $(n-1)$ derivatives as the state variables. This choice is called phase variable choice. Choosing state variable, x_i, we get

$$x_1 = y$$

$$x_2 = \frac{dy}{dt}$$

$$x_3 = \frac{d^2y}{dt^2}$$ (2.8)

$$x_n = \frac{d^{n-1}y}{dt^{n-1}}$$

and differentiating both sides yields

$$\begin{aligned}
\dot{x}_1 &= \frac{dy}{dt} \\
\dot{x}_2 &= \frac{d^2y}{dt^2} \\
\dot{x}_3 &= \frac{d^3y}{dt^3} \\
&\vdots \\
\dot{x}_n &= \frac{d^ny}{dt^n}
\end{aligned}$$

--------------(2.9)

In vector matrix form we can write

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\ \vdots & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & -a_2 & -a_3 & -a_4 & -a_5 & \cdots & -a_{n-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ b_0 \end{bmatrix} u$$

…(2.11)

Finally the solution to the differential equation is $y(t), or\ x_1$ the output equation is

$$y = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix}$$

…………………………(2.12)

2.3 Linear and Nonlinear SISO system:

2.3.1 Linear SISO:

A *linear system* is one in which the output is jointly linear in the intitial condition for the system and the input to the system. In particular, a linear system has the property that if we apply an input $u(t) = \alpha u_1(t) + \beta u_2(t)$ with zero initial condition, the corresponding output will be $y(t) = \alpha y_1(t) + \beta y_2(t)$, where y_i is the output associated with the input u_i. This property is called linear *superposition*.

A differential equation of the form

$$\dot{x} = Ax + Bu \qquad x \in R^n, u \in R$$
$$y = Cx + Du \qquad y \in R \qquad \text{………………………………………(2.13)}$$

is a *single-input, single-output* (SISO) *linear differential equation*.

A linear system is characterized only by,

$$\dot{x} = Ax \qquad \text{……} \qquad (2.14)$$

is _stable if and only if all eigenvalues of A all have strictly negative real part and is unstable if any eigenvalue of it has strictly positive real part.

2.3.2 Nonlinear SISO :

A nonlinear system of the form

$$\dot{x} = f(x, u) \qquad x \in R^n, u \in R$$
$$y = h(x, u) \qquad y \in R \qquad . \qquad \text{……………………..(2.15)}$$

is a single-input, single-output (SISO) nonlinear system. It can be linearized about an equibrium point $x = x_e$, $u = u_e$, $y = y_e$ by defining new variables

$$z = x - x_e \qquad v = u - u_e \qquad w = y - h(x_e, u_e). \text{ ……………} \qquad (2.16)$$

The dynamics of the system near the equilibrium point can then be approximated by the linear system

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du \qquad \text{……………………………} \qquad (2.17)$$

where

$$A = \left.\frac{\partial f(x,u)}{\partial x}\right|_{x_e,u_e} \qquad B = \left.\frac{\partial f(x,u)}{\partial u}\right|_{x_e,u_e}$$

$$C = \left.\frac{\partial h(x,u)}{\partial x}\right|_{x_e,u_e} \qquad D = \left.\frac{\partial y(x,u)}{\partial u}\right|_{x_e,u_e} \qquad \text{.................} \quad (2.18)$$

The equilibrium point for a nonlinear system is locally stable if the real part of the eigenvalues of the linearization about that equilibrium point have strictly negative real part.

2.4 SISO SYSTEM MODEL APPROACH:

2.4.1 : State variable model for a dynamic system

This type of model consists of a set of simultaneous first-order differential equations, called the State equation:

$$\frac{dx}{dt} = [A]x + [B]i(t) \qquad \text{........}(2.19)$$

and the output equation is:

$$O(t) = [C]x + Di(t). \qquad \text{........................} \quad .(2.20)$$

Here we define:

x as the *state vector*,

A as the *system matrix* (square, N x N for N states),

B as the *input matrix* (N rows x 1 column for a single-input, single output (SISO) system,

C as the *output matrix* (one row x N columns for a SISO system),

D as the *feedforward matrix* (1 x 1 for a SISO system).

Chapter-3

MISSILE DYNAMICS

3.1 Missile overview

A missile can be defined as any object that can be thrown, projected or propelled toward a target . 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 . Obviously, it is primarily used as a weapon in order to give damage to the target.

3.2 Category of Missile

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

- Unguided Missiles
- Guided Missiles

Unguided missiles, whether initially or continuously propelled, can beoriented toward their targets only before they are fired. After firing, they get completely out of control.

❖ The guided missiles can be categorized into two groups depending on the operational range :

- Tactical Missiles
- Strategic, or Ballistic, or Cruise Missiles.

❖ Depending on their missions, the missiles can be divided into four subsets
- Surface-to-Surface Missiles (SSM)
- Surface-to-Air Missiles (SAM)
- Air-to-Air Missiles (AAM)
- Air-to-Surface Missiles (ASM)

3.2.1 **Guided Missile control**

Regarding the guided missiles, the guidance and control problem involves
four sequential stages:
- Dynamic Modeling
- Guidance
- Control
- Target Motion Estimation

3.3 STUDIES ON THE CONTROL OF MISSILES IN THE LITERATURE

3.3.1 Guided Missile consideration :

The mission of a missile control system, or a missile autopilot, is to ensure the stability, high performance and that the missile flies in accordance to the demands of the guidance law .

In most of the missile control systems, the controlled variables are selected to be the lateral acceleration components of the missile. Depending on the type of the guidance command, the controlled variables may also be the body rates (yaw, pitch and roll rates), body angles (yaw, pitch and roll angles), wind frame angles (angle of attack and side-slip angles), or their rates.In the guided missiles, the necessary steering actions are achieved by various motion control elements such as the ones listed below

- Aerodynamic Control Surfaces
- Thrust Vector Deflectors
- Side Jets or Reaction Jets

3.3.2 Missile type –Guided missile :

Aerodynamic control surfaces can be in the form of canards, wings, and tails depending on their locations on the missile .

❖ In the canard-controlled missiles, the control fins, or canards, are mounted on the nose part of the missile.They are light in weight and can be simply packaged..

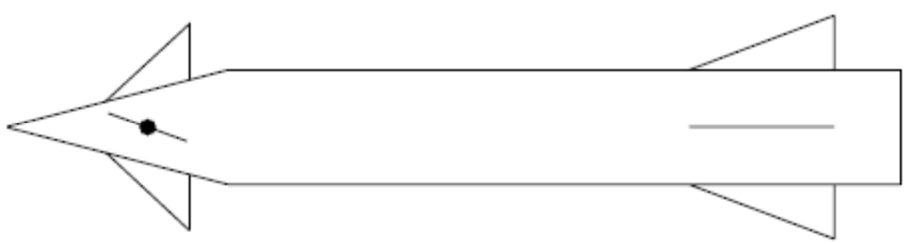


Fig 3.1 :Canard Controlled Missile

• In the wing-controlled missiles, the control fins are mounted near the mass center of the missile (CG) as shown . This type of control provides fast response of body lateral acceleration

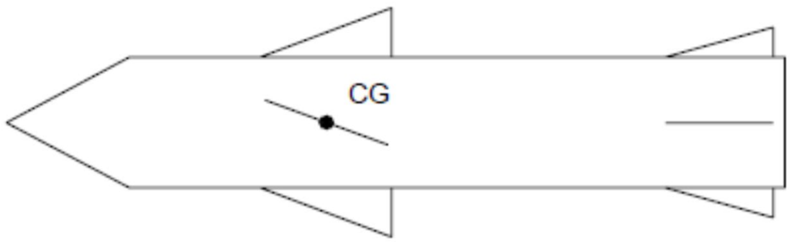


Fig3.2 : Wing Controlled Missile

❖ The third type of aerodynamic control surfaces are tail fins. As the name implies, these control surfaces are placed at the tail section of the missile as show.

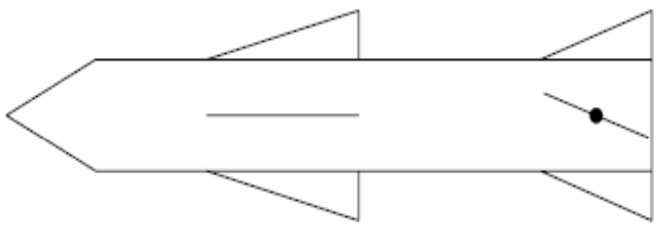


Fig 3.3 :Tail fins Missile

3.4 MISSILE MODEL

In this section, first, the missile model considered in this study is described. Next, completing the kinematic analysis of the considered missile model, its governing differential equations of motion are derived. Afterwards, the necessary transfer functions to be used in the control system design stage have been determined based on the derived equations of motion. Then, the control actuation system model is constructed. Eventually, the models for the measuring instruments and the wind are given.

3.4.1 Missile Dynamic Model

In this study, a missile is considered such that its body is combined of two parts which are connected to each other by means of a free-rotating bearing. In this structure, the front body carries the guidance and control sections of the entire missile while the rear body involves the fixed tail wings and the rocket motor that supplies the missile with thrust required in the boost phase.

3.4.2 MISSILE KINEMATICS

$$\begin{array}{ccccccc} & \vec{u}_3^{(0)} & & \vec{u}_2^{(m)} & & \vec{u}_1^{(n)} & \\ F_0 & \rightarrow & F_m & \rightarrow & F_n & \rightarrow & F_b \\ & \psi_1 & & \theta_1 & & \phi_1 & \end{array} \quad \text{……………………(3.1)}$$

Here, F_0 denotes the Earth-fixed reference frame whose origin is point Oe while *Fm* and *Fn* show the intermediate reference frames between *F0* and *Fb*. .Also,ψ1 , θ1 and φ1 stand for the Euler angles in the yaw, pitch and roll directions of body 1, respectively. As a general note for the reference frames, they are all taken to be orthogonal and right-handed.

3.4.3 FORCES ON A ROCKET IN FLIGHT :

Forces on a rocket in flight, rockets that must travel through the air are usually tall and thin at his shape gives a high ballistic coefficient and minimizes drag losses

Flying rockets are primarily affected by the following:

- Thrust from the engine(s)
- Gravity from celestial bodies
- Drag if moving in atmosphere
- Lift; usually relatively small effect except for rocket-powered aircraft

It can be shown that the net thrust of a rocket is:

$$F_n = \dot{m}\, v_e \qquad (3.2)$$

where:

$\dot{m}$ =propellant flow (kg/s or lb/s)
v_e =the effective exhaust velocity (m/s or ft/s)

Drag can be minimized by an aerodynamic nose cone and by using a shape with a high ballistic coefficient (the "classic" rocket shape—long and thin), and by keeping the rocket's angle of attack as low as possible.

3.4.4 MISSILE SEEKER SHOWING GEOMETRY:

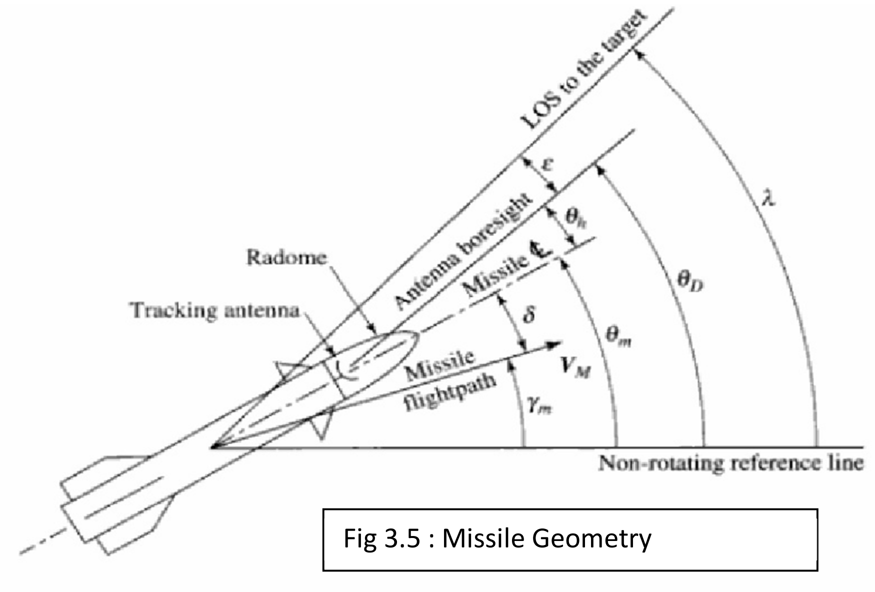


Fig 3.5 : Missile Geometry

3.4.5 Equations of Motion of the Missile :

Having completed the kinematic analysis, the equations of motion of the missile can be derived using the Newton-Euler equations, which express the force and moment balance on each body. The Newton-Euler equations for body are

$\overline{F_{A1}} + \overline{F_{21}} + m_1\bar{g} = m_1\,\overline{a_{c1/oe}}$ ....(3.3a)

$\overline{M_{A1}} + \overline{M_{21}} + \overline{r_{B/C1}} \times \overline{F_{21}} = \overline{J_{c1}}.\overline{\alpha_{b/0}} + \bar{\omega}_{b/0} \times J_{C1}.\dot{\omega}_{b/o}$ ....(3.3b)

Similarly, the Newton-Euler equations for body 2 are

$\overline{F_{A2}} + \overline{F_{r2}} + \overline{F_{12}} + m_2\bar{g} = m_2\,\overline{a_{c2/oe}}$ ....(3.4a)

$\overline{M_{A2}} + \overline{M_{12}} + \overline{M_{r2}} + \overline{r_{B/C2}} \times \overline{F_{12}} = \overline{J_{c2}}.\overline{\alpha_{b/0}} + \bar{\omega}_{b/0} \times J_{C2}.\dot{\omega}_{s/o}$ .... (3.4b)

$\bar{F}_{Ai}$, $\bar{M}_{Ai}$: Aerodynamic force and moment vectors acting on body i

$\bar{F}_{ij}$, $\bar{M}_{ij}$: Reaction force and moment vectors applied by body i on body j

$\bar{F}_{T2}$, $\bar{M}_{T2}$: Thrust force and thrust misalignment moment vectors acting on body 2

m_i : Mass of body i

3.4.6 **Common missile structure:**

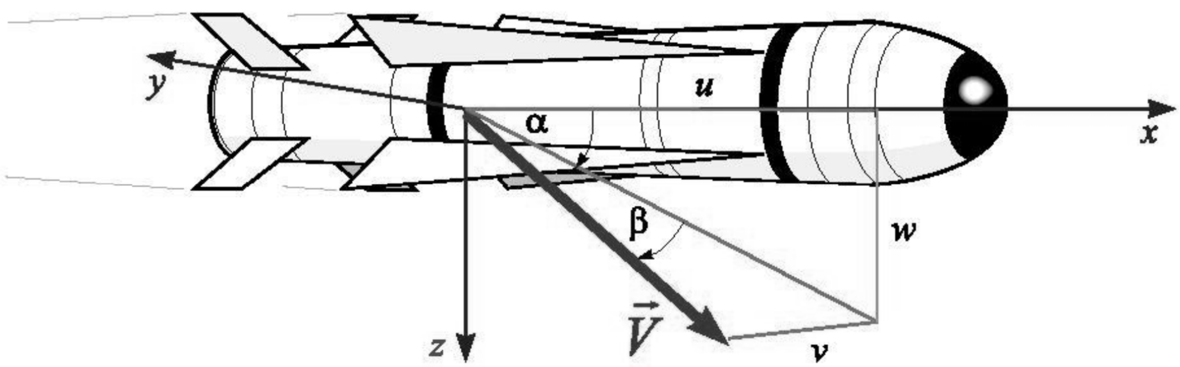


Fig 3.6 : Common Missile structure

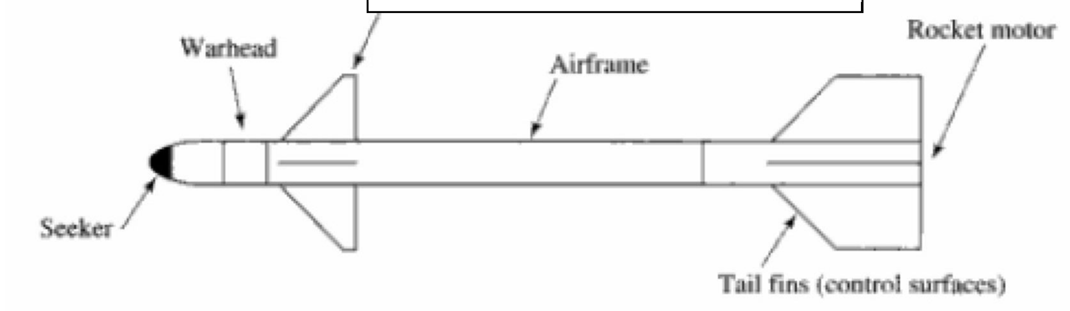


Fig 3.7 : Common missile structure

3.5 MISSILE MODELING PROBLEM

3.5.1 Problem Statement:

A missile in flight, as shown is subject to several forces: thrust, lift, drag, and gravity.The missile flies at an angle of attack, a, from its longitudinal axis, creating lift. For steering, the body angle from vertical, f, is controlled by rotating the engine at the tail. The transfer function relating the body angle, f, to the angular displacement, d, of the engine is of the form

$$\frac{\Phi(s)}{\delta(s)} = \frac{K_a s + K_b}{K_3 s^3 + K_2 s^2 + K_1 s + K_0} \quad \text{..........................(3.5)}$$

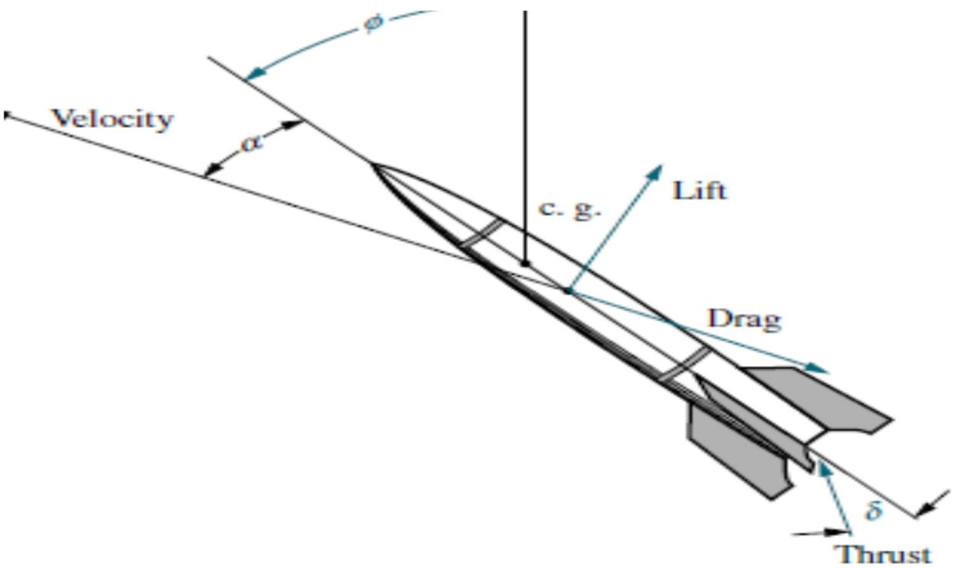


Fig 3.8 : Missile problem

3.5.2 CONTROLLING OF MISSILE:

Forces ------- (i) Thrust (ii)Lift (iii) Drag (iv) Gravity

Given parameter: $\alpha = angle\ of\ attack$

φ=body angle from vertical

$\delta = \text{angular displacement}$

3.5.3 Transfer Function of SISO:

Output = the body angle which is responsible for steering

Input = rotation of the tail / the angular displacement

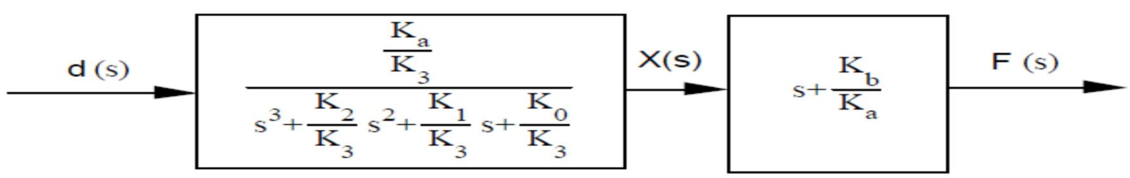
$$\boldsymbol{y(t) = \phi(t);}$$

$$\boldsymbol{u(t) = \delta(t);}$$

In Laplace domain :

$$\boldsymbol{T(s) = \frac{\phi(s)}{\delta(S)} = \frac{K_a s + k_b}{k_3 s^3 + k_2 s^2 + k_1 s + k_0}} \qquad \textbf{(3.6)}$$

The equivalent cascade transfer function is as shown below.



For the first box, $\dddot{x} + \frac{K_2}{K_3}\ddot{x} + \frac{K_1}{K_3}\dot{x} + \frac{K_0}{K_3}x = \frac{K_a}{K_3}\delta(t)$.

Selecting the phase variables as the state variables: $x_1 = x$, $x_2 = \dot{x}$, $x_3 = \ddot{x}$.

Writing the state and output equations:

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = x_3$$

$$\dot{x}_3 = -\frac{K_0}{K_3}x_1 - \frac{K_1}{K_3}x_2 - \frac{K_2}{K_3}x_3 + \frac{K_a}{K_3}\delta(t)$$

$$y = \phi(t) = \dot{x} + \frac{K_b}{K_a}x = \frac{K_b}{K_a}x_1 + x_2$$

In vector-matrix form,

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -\frac{K_0}{K_3} & -\frac{K_1}{K_3} & -\frac{K_2}{K_3} \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ \frac{K_a}{K_3} \end{bmatrix} \delta(t)\ ;\ y = \begin{bmatrix} \frac{K_b}{K_a} & 1 & 0 \end{bmatrix} x$$

………..(3.7)

3.5.4 **State space representation :**

$$\dot{x}=\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -\frac{k_0}{k_3} & -\frac{k_1}{k_3} & -\frac{k_2}{k_3} \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ \frac{k_a}{k_3} \end{bmatrix} \delta(t); \qquad \text{.........(3.8)}$$

$$\boldsymbol{y} = \begin{bmatrix} \frac{\boldsymbol{k_b}}{\boldsymbol{k_a}} & \mathbf{1} & \mathbf{0} \end{bmatrix} \mathrm{x}; \qquad \text{..........(3.9)}$$

$x = state\ vector$;

$\dot{x}$=derivative of state vector;

A=system matrix=$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -\frac{k_0}{k_3} & -\frac{k_1}{k_3} & -\frac{k_2}{k_3} \end{bmatrix}$ B=input matrix=$\begin{bmatrix} 0 \\ 0 \\ \frac{k_a}{k_3} \end{bmatrix}$ ....(3.10)

C=output matrix== $\begin{bmatrix} \frac{\boldsymbol{k_b}}{\boldsymbol{k_a}} & \mathbf{1} & \mathbf{0} \end{bmatrix}$

Y=output vector

3.5.5 **Transfer function Via MATLAB** : (appendix-A)

Given Data:

Ki=1; Kb=1.5; K3=70; K2=43;

K1=4;

Ko=2.5;

3.5.6 Step response of missile transfer function: MATLAB SIMULATION

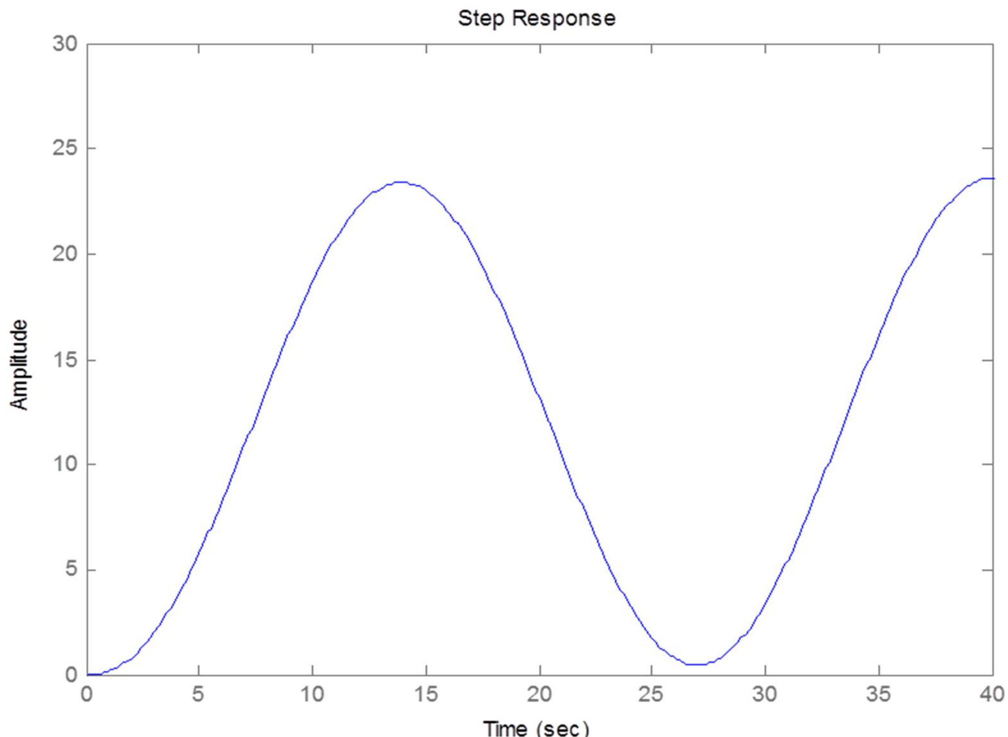


Figure 3.9 : Step response of SISO

3.5.7 ANALYZING THE STEP RESPONSE:

It is clearly shown from the figure

- ✓ system is unstable,
- ✓ It has high oscillation,
- ✓ high rise time,
- ✓ Steady state error is higher.

3.5.8 Performance parameter of a system:

Rise Time(Tr): It is defined as the time for waveform to go from 0.1 to 0.9 of its final value. Rise time is found as $T_r = \frac{2.31}{a} - \frac{0.11}{a} = \frac{2.2}{a}$

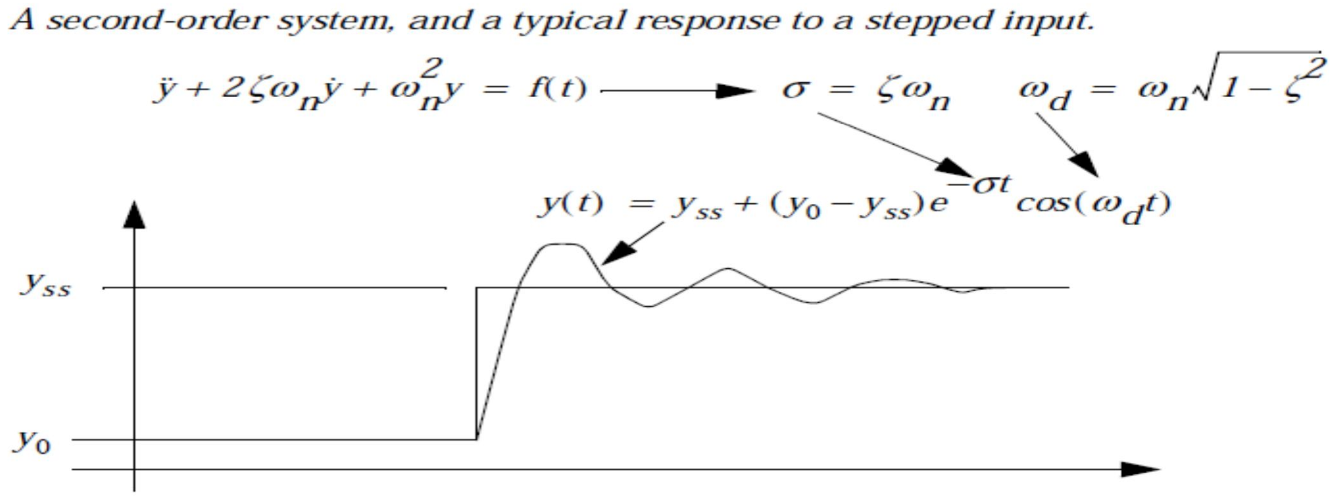
Settling Time(Ts): It is defined as the time response to reach and stay within 2% of its final value.It is found as $T_s = \frac{4}{a}$

Peak Time(Tp): It is inversely proportional to the imaginary part of the complex pole.

Percentage Overshoot(%OS): It is a function of only the damping ratio.

3.6 Second-order system:

A second-order system response typically contains two first-order responses, or a first order response and a sinusoidal component. A typical sinusoidal second-order response is shown in Figure. Notice that the coefficients of the differential equation include a damping coefficient and a natural frequency. These can be used to develop the final response, given the initial conditions and forcing function. Notice that the damped frequency of oscillation is the actual frequency of oscillation. The damped frequency will be lower than the natural frequency when the damping coefficient is between 0 and 1. If the damping coefficient is greater than one the damped frequency becomes negative, and the system will not oscillate because it is over damped



ω_n *Natural frequency of system - Approximate frequency of control system oscillations.*

ξ *Damping factor of system - If < 1 underdamped, and system will oscillate. If =1 critically damped. If > 1 overdamped, and never any oscillation (more like a first-order system). As damping factor approaches 0, the first peak becomes infinite in height.*

ω_d *The actual frequency of oscillation - It is below the natural frequency because of the damping.*

Figure3.10 : General form of a second order system

3.6.1 .SECOND ORDER SYSTEM CHARACTERISTICS:

Rise time is the time it takes to go from 10% to 90% of the total displacement, and is comparable to a first order time constant. The settling time indicates how long it takes for the system to pass within a tolerance band around the final value. The permissible zone shown is

2%, but if it were larger the system would have a shorter settling time. The period of oscillation can be measured directly as the time between peaks of the oscillation, the inverse is the damping frequency. The damped frequency can also be found using the time to the first peak, as half the period. The overshoot is the height of the first peak. Using the time to the first peak, and the overshoot the damping coefficient can be found

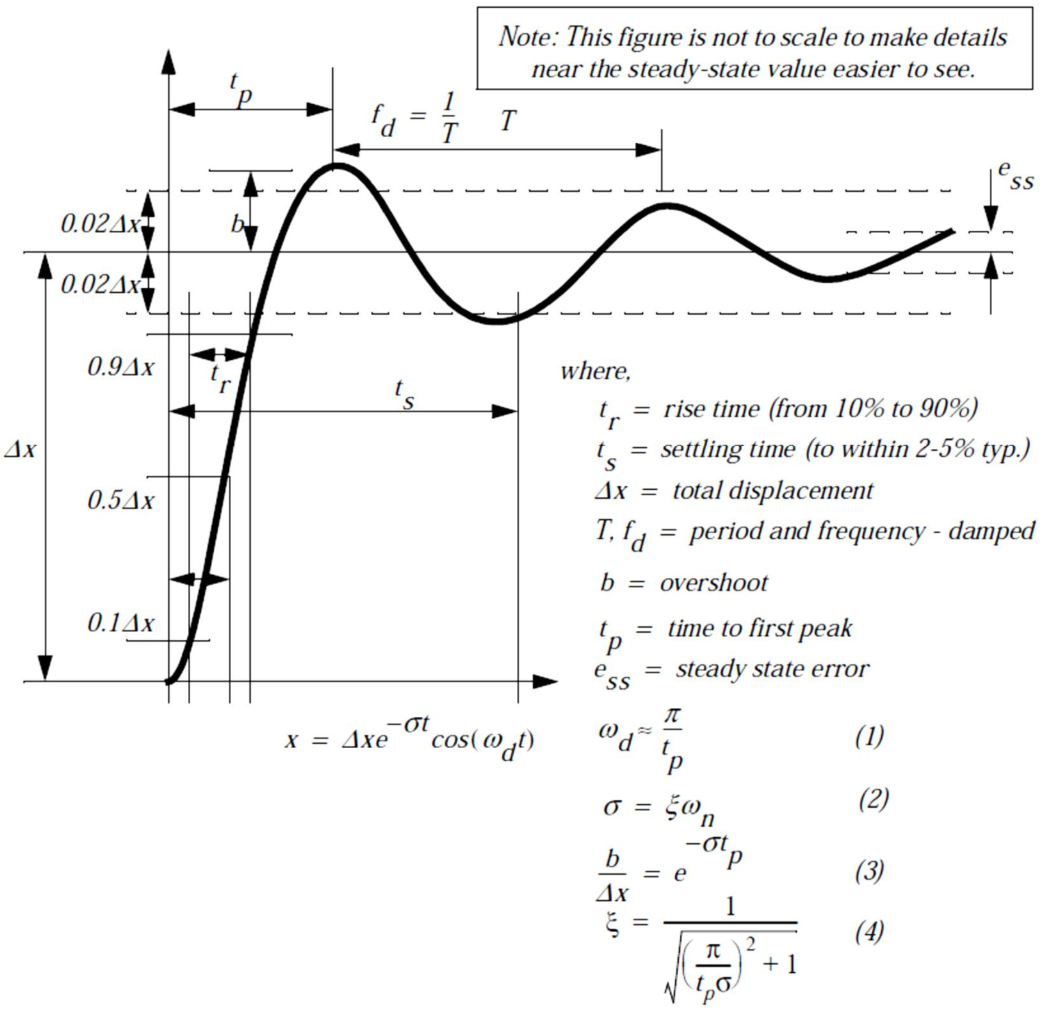


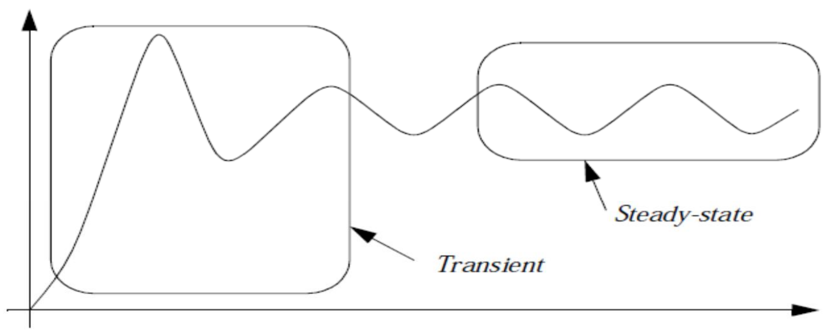
Figure3.11 : Characterizing a second order response

3.6.2 Second order system Response Analysis:

Up to this point we have mostly discussed the process of calculating the system response. As an engineer, obtaining the response is important, but evaluating the results is more important. The most critical design consideration is system stability. In most cases a system should be inherently stable in all situations. In other cases an unstable system may be the objective. Simple methods for determining the stability of a system are listed below:

1. If a step input causes the system to go to infinity, it will be inherently unstable.

2. A ramp input might cause the system to go to infinity; if this is the case, the system might not respond well to constant change.
3. If the response to a sinusoidal input grows with each cycle, the system is probably resonating, and will become unstable. Beyond establishing the stability of a system, we must also consider general performance. This includes the time constant for a first-order system, or damping coefficient and natural frequency for a second-order system.
Figure shows a system response. The transient effects at the beginning include a quick rise time and an overshoot. The steady-state response settles down to a constant amplitude sine wave.



Note: the transient response is predicted with the homogeneous solution. The steady state response in mainly predicted with the particular solution, although in some cases the homogeneous solution might have steady state effects, such as a non-decaying oscillation.

Fig 3.12 Response of a 2nd order system

3.7 Control system analysis and design focus:

- Producing the desired transient response
- Reducing steady state errors
- Achieving stability

3.7.1 General Description of a Controlled System:

3.7.2 Structure and components

A controlled system is composed of several elements. The plant is a part of the overall controlled system as shown in figure

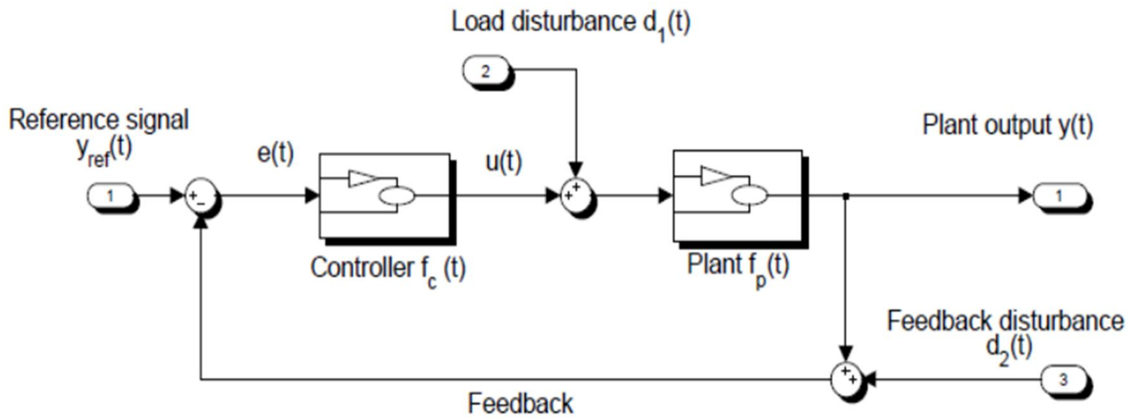


Figure3.13 : Simplified model of a control system

This system typology is designed as *feedback control* or *closed loop control*. It makes use of a feedback signal by means of which the desired value of the plant output is compared to the actual one. The error feeds the input signal of the controller that provides the plant with an action that is function of the error. An alternative typology is the *open loop control* that does not use a feedback signal. Open loop controllers are used when there are no important load disturbances to be suppressed that justify the implementation of the more expensive closed loop structure

3.7.3 General Controlled System:

Variables commonly used to describe the dynamics of the controlled sys-tem are the reference signal or set-point or desired value *yref* (*t*), the plant output *y*(*t*), the error between the plant output and the reference signal *e*(*t*) = *yref* (*t*) ¡ *y*(*t*), the control variable *u*(*t*), the load disturbance *dl*(*t*) and the feedback disturbance *df* (*t*).The main components of a SISO3 control system are: a *prefilter*, a *compensator*, the *plant* and *trans conductors* which are divided into *actuators* and *sensors*.

Chapter 4

PID CONTROLLED SYSTEMS

4.1 INTRODUCTION AND REPRESENTATION:

A general closed loop controller implements the control variable *u*(*t*) as a function of the error *e*(*t*). A PID controller bases its action on the sum of three values derived from the error: a proportional action, a derivative action and an integral action. The weights of the three different actions are the parameters of a PID controller. The tuning of a PID controller consists in the search of the parameters that can optimize a pre-specified performance index within some particular constraints. The typology of PID controller is widely spread in industry where more than 90% of all control loops are PID. The action taken by a PID controller on the plant can be expressed in the time domain as

$$u(t) = K\left(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d\frac{de(t)}{dt}\right) \quad \ldots\ldots\ldots\ldots\ldots..(4.1)$$

In the frequency domain PID controller can be expressed by the transfer function

$$G_{PID}(s) = K\left(1+\frac{1}{sT_i} + sT_d\right) \quad (4.2)$$

This equation is called standard or non-interacting form (Äström, Hagglund'1995)
This equation can also be written to highlight the three PID parameters as

$$G_{PID}(s) = k_3 s + k_1 + \frac{K_2}{S} \quad (4.3)$$

When $K_{1,} K_{2,} K_3$ are the weights of the proportional, integral and derivative actions.
An alternative representation is expressed by

$$G'_{PID}(s) = K'\left(1+\frac{1}{ST_{i'}}\right)(1+sT'_d) \quad (4.4)$$

4.2 **Criteria of Performance Evaluation:**

The problem of the evaluation of a control system performance has always been a central topic in control system theory. Constraints and requirements play an important role in the design of a controller.
With the emergence of evolutionary algorithms in control system engineering, the problem of the performance has become related to the evaluation of a solution. For each control system, a performance index should be calculated in order to express the quality of the controller
as a unique positive value.

4.2.1 Time domain indices

Several indices are proposed. I will limit the description to the indices used in this paper. A measure of the difference between the reference value and the actual plant output, intended as the area between the two curves, is given by the IAE and ITAE indices defined as follows. The Integral of Absolute Error is expressed by

$$\text{IAE}=\int_0^T |\ e(t)\ | dt \tag{4.5}$$

The integral of Time-weighted Absolute error is expressed by

$$\text{IAE}=\int_0^T t\ |e(t)|\ dt \tag{4.6}$$

where *T* is an arbitrarily chosen value of time so that the integral reaches a steady-state value.

4.3 **Conventional PID controllers:**

4.3.1 Preliminary and background

PID controllers are the most widely-used type of controller for industrial applications. They are structurally simple and exhibit robust performance over a wide range of operating conditions. In the absence of the complete knowledge of the process these types of controllers are the most efficient of choices. The three main parameters involved are Proportional (P), Integral (I) and Derivative (D). The proportional part is responsible for following the desired set-point, while the integral and derivative part account for the accumulation of past errors and the rate of change of error in the process respectively.

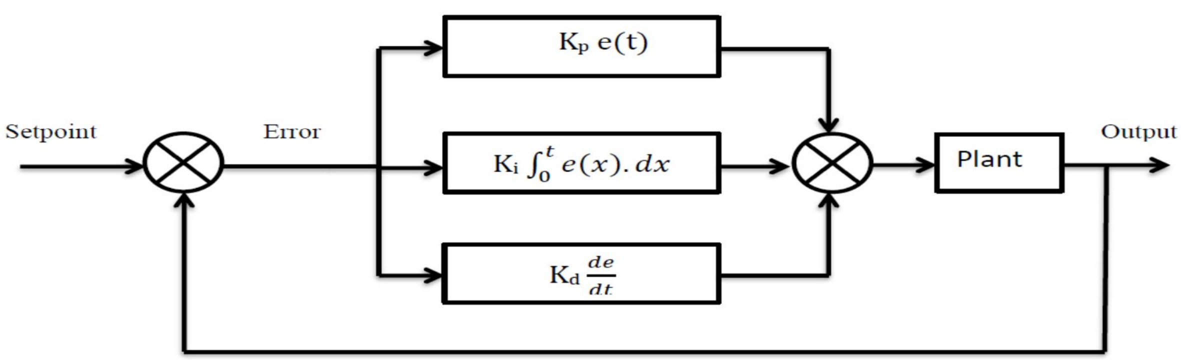


Figure 4.1: Basic block diagram of a conventional PID controller

For the PID controller presented in figure

The output of the PID controller

$$u(t) = K_p e(t) + K_i \int_0^t e(x)\,dx + K_d \frac{de(t)}{dt} \quad (4.7)$$

Error, e(t) = Set point - Plant output

4.4 Tuning of PID parameters:

Tuning of a PID controller refers to the tuning of its various parameters (P, I and D) to achieve an optimized value of the desired response. The basic requirements of the output will be the stability, desired rise time, peak time and overshoot. Different processes have different requirements of these parameters which can be achieved by meaningful tuning of the PID parameters. If the system can be taken offline, the tuning method involves analysis of the step input response of the system to obtain different PID parameters. But in most of the industrial applications, the system must be online and tuning is achieved manually which requires very experienced personnel and there is always uncertainty due to human error.

4.5 USING PID CONTROLLER ON MISSILE CONTROL SYSTEM:

MATLAB SIMULATION:

1. with unity feedback and close loop pole
2. Series connection of Gp and Gc
3. Negative feedback
4. Close loop transfer function

$$G_c(S) = k_p + \frac{k_I}{s} + k_D s = \frac{k_p s + k_D s^2 + k_I}{s} \quad (4.8)$$

Ultimate Transfer Function:

$$T(s) = \frac{G_p G_c}{1 + G_p G_c} \quad (4.9)$$

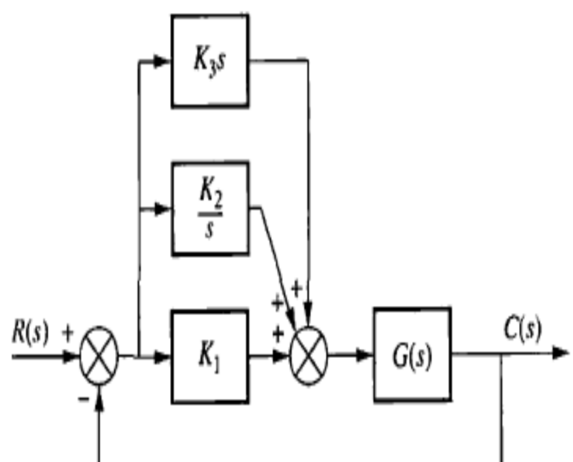


Figure 4.2: PID controller

4.5.1 OUTPUT TRANSFER FUNCTION via MATLAB(Appendix-A):

$$T(s)=\frac{300 s^3 + 550 s^2 + 250 s + 150}{70 s^4 + 343 s^3 + 554 s^2 + 252.5 s + 150} \tag{4.10}$$

4.5.2 TUNING via TRIAL AND ERROR METHOD:

1. Determine the gain kp,Ki and Kd
2. Simulate the system to meet all design requirements
3. Redesign and redesign until the requirement have not been met.

Advantages:
1. Improving Steady state error
2. Improving transient response

4.5.3 MATLAB SIMULATION –PID :

Tuning Value:
- ✓ ka=1;
- ✓ kb=1.5;
- ✓ k3=70;
- ✓ k2=43;
- ✓ k1=4;
- ✓ ko=2.5
- ✓ kp=100;
- ✓ ki=100;
- ✓ kd=300;

4.5.4 STEP RESPONSE USING PID :

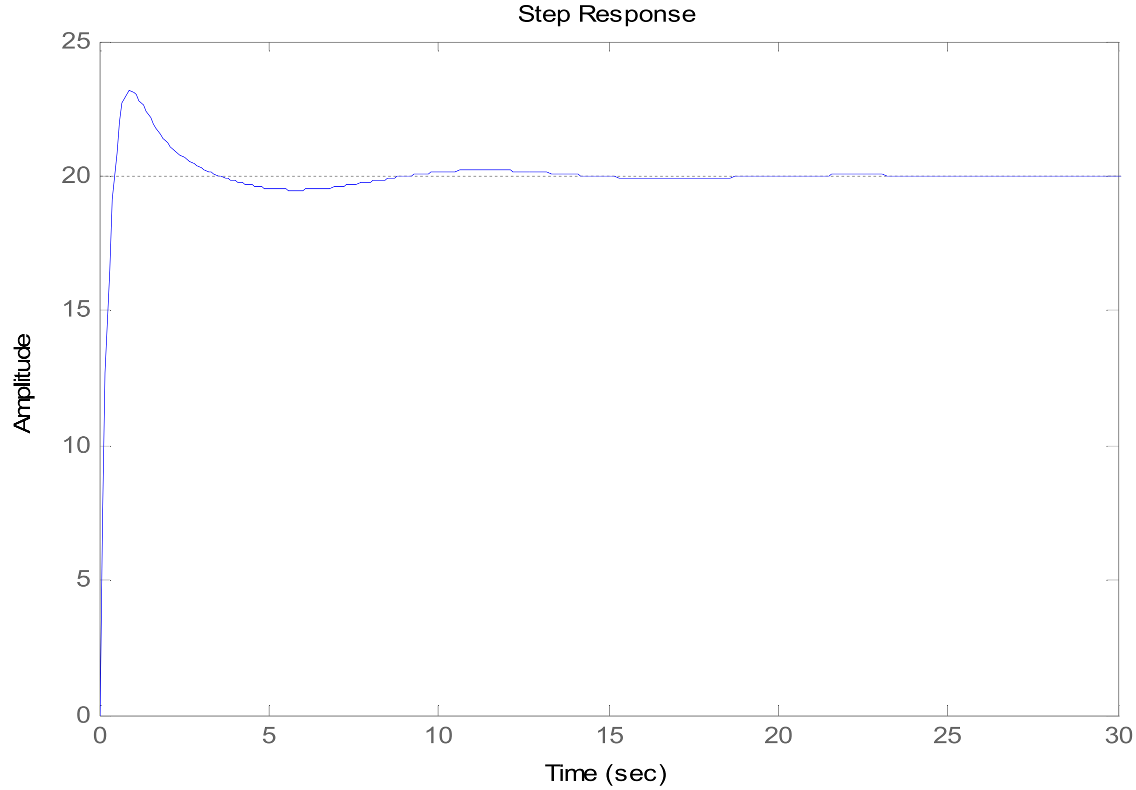


Fig 4.3: Step response after Using PID

4.5.5 Improvement of the system:

In this figure it is clear that

- ✓ oscillation is less,
- ✓ Rise time
- ✓ Fall time is small,
- ✓ Steady state error is lesser than the system without controller

CHAPTER -5

SIMO SYSTEM

5.1 SIMO SYSTEM:

SIMO (single input, multiple output) is technology for wireless communications in which multiple antennas are used at the destination (receiver). The antennas are combined to minimize errors and optimize data speed. The source (transmitter) has only one antenna.

In conventional wireless communications, a single antenna is used at the source, and another single antenna is used at the destination. In some cases, this gives rise to problems with multipath effects. When an electromagnetic field (EM field) is met with obstructions such as hills, canyons, buildings, and utility wires, the wavefronts are scattered, and thus they take many paths to reach the destination. The late arrival of scattered portions of the signal causes problems such as fading, cut-out (cliff effect), and intermittent reception (picket fencing). In digital communications systems such as wireless Internet, it can cause a reduction in data speed and an increase in the number of errors. The use of two or more antennas at the destination can reduce the trouble caused by multipath wave propagation.

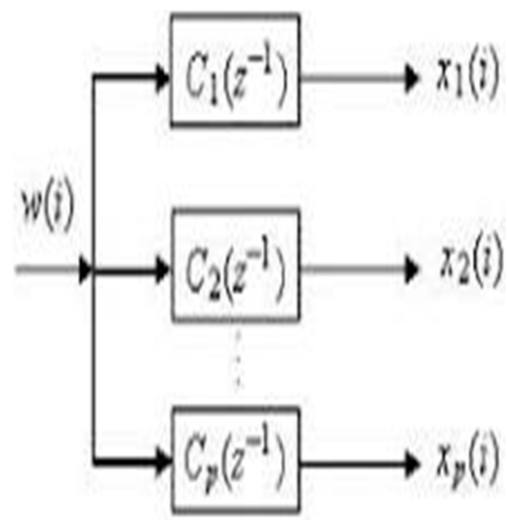


Fig 5.1:Typical SIMO System

SIMO technology has widespread applications in digital t area networks (WLANs), metropolitan area networks (MANs), and mobile communications.

An early form of SIMO, known as diversity reception, has been used by military, commercial, amateur, and shortwave radio operators at frequencies below 30 MHz since the First World War

5.2 PROBLEM STATEMENT:

Given the military aircraft shown in Figure , where normal acceleration, an, and pitch q are controlled by elevator deflection δ_e on the horizontal stabilizer and Canard deflection δ_{com} as shown in figure is used to effect a change in both δ_e and δ_c

The relationships are:

$$\frac{\delta_e(s)}{\delta_{\text{com}}(s)} = \frac{1/\tau}{s + 1/\tau}$$

$$\frac{\delta_c(s)}{\delta_{\text{com}}(s)} = \frac{K_c/\tau}{s + 1/\tau}$$

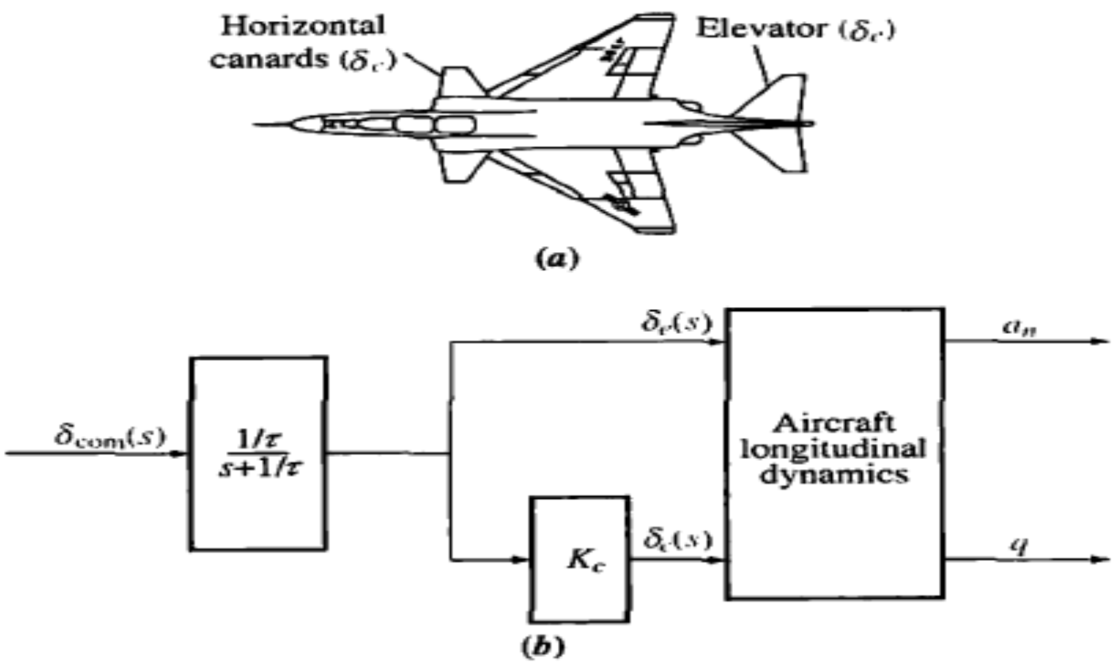


Figure 5.2: open loop flight control system

5.2.1 PARAMETER AND OTHER CONSTRAINTS:

.These deflections yield,via the aircraft control longitudinal dynamics ,a_n and q. The state equations describing the effect δ_{com} on $a_n$and q is given by …….

$$\begin{bmatrix} \dot{a_n} \\ \dot{q} \\ \dot{\delta_e} \end{bmatrix} = \begin{bmatrix} -1.702 & 50.72 & 263.38 \\ 0.22 & -1.418 & -31.99 \\ 0 & 0 & -14 \end{bmatrix} \begin{bmatrix} a_n \\ q \\ \delta_e \end{bmatrix} + \begin{bmatrix} -272.06 \\ 0 \\ 14 \end{bmatrix} \delta_{COM} \qquad (5.1)$$

And the transfer function for the two outputs is given,

$$\text{G}\mathbf{1}(\boldsymbol{s}) = \frac{\boldsymbol{A_n(s)}}{\boldsymbol{\delta_{com}(s)}} \qquad \textbf{(5.2)}$$

$$\text{G}\mathbf{2}(s) = \frac{\boldsymbol{Q}\ (s)}{\delta_{com}(s)} \qquad (5.3)$$

5.3 MODELLING OF SIMO:AIRCRAFT CONTROL SYSTEM:

Dynamics: The main dynamics is as like as missile and all the forces discussed previously exist here.

Pitch rate($\boldsymbol{q}$): It is simply the rate at which the aircraft can pitch - ie. rotate around the lateral axis, which is roughly the wings.

Normal Acceleration($\boldsymbol{a_n}$) : The forward acceleration normal to the surface.

Elevator deflection(δ_e): Deflection of horizontal stabilizer

Canard deflection (δ_c) :Deflection of the canards

5.4 Aircraft Equations of Motion :

Equation of motion depends on the following factors:

- 6 degrees of freedom
- Angular kinematics
- Euler angles
- Rotation matrix
- Angular momentum
- Inertia matrix

Angular Momentum and Rate:
Angular momentum and rate vectors are not necessarily aligned

Angular momentum ,$h = I\omega$
Where I=inertia
ω=angular momentum

5.5 LAPLACE Transform of dynamic system:

System equation

$$\triangleq X(t) = F \triangleq X(t) + \dot{G} \triangleq u(t) + L \triangleq w(t) \qquad (5.4)$$

Laplace transform of system equation:

$$s \triangleq X(s) - \triangleq X(0) = F \triangleq X(S) + G \triangleq u(S) + L \triangleq w(S) \qquad (5.5)$$

5.6 Configuration Aerodynamics :

5.6.1 Configuration Variables:

• Lift
– Effects of shape, angle, and
Mach number
– Stall
• Parasitic Drag
– Skin friction
– Base drag

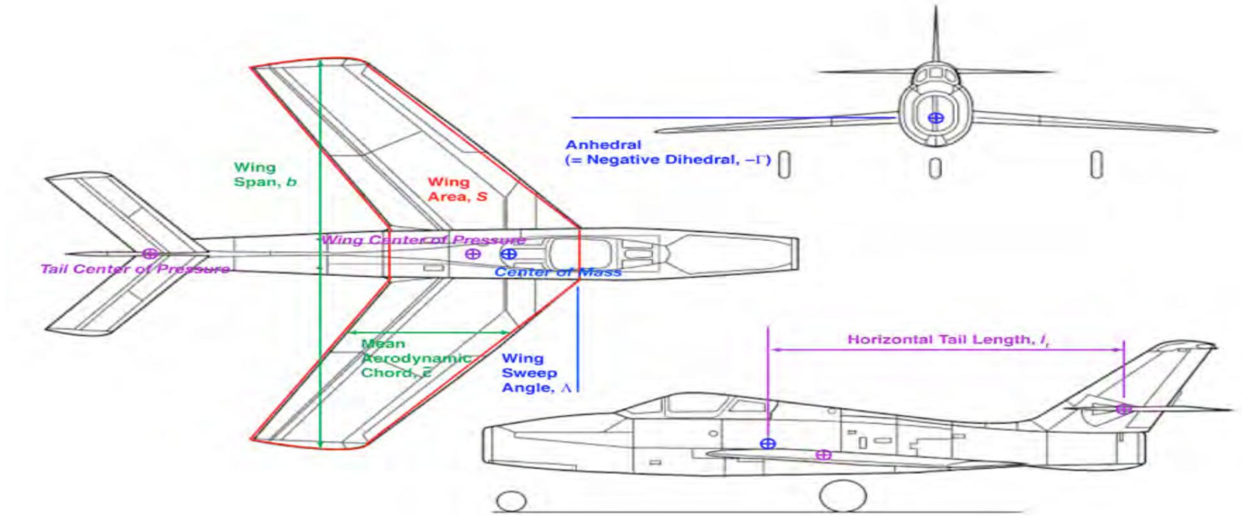


Figure5.4 : Basic configuration of aerodynamics-some definitions

5.7 Case Problem (SIMO System):

Input : commanded deflection , $\boldsymbol{\delta_{con}}$, which controls both the elevator deflection and canard deflection

$$\delta_{com} = \frac{1/\tau}{s+1/\tau} \qquad (5.6)$$

Output :

Normal acceleration, $\boldsymbol{a_n}$: The transfer function for this output is

$$\mathrm{G}\mathbf{1}(\boldsymbol{s}) = \frac{\boldsymbol{A_n(s)}}{\boldsymbol{\delta_{com}(s)}} \qquad \textbf{(5.7)}$$

Pitch rate: The transfer function for this output is

$$\mathrm{G}\mathbf{2}(s) = \frac{\boldsymbol{Q}\ (s)}{\delta_{com}(s)} \qquad (5.8)$$

MATRIX for the aircraft system =

$$\begin{bmatrix} \dot{a_n} \\ \dot{q} \\ \dot{\delta_e} \end{bmatrix} = \begin{bmatrix} -1.702 & 50.72 & 263.38 \\ 0.22 & -1.418 & -31.99 \\ 0 & 0 & -14 \end{bmatrix} \begin{bmatrix} a_n \\ q \\ \delta_e \end{bmatrix} + \begin{bmatrix} -272.06 \\ 0 \\ 14 \end{bmatrix} \delta_{COM} \qquad (5.9)$$

Where

$$A=\begin{bmatrix} -1.702 & 50.72 & 263.38 \\ 0.22 & -1.418 & -31.99 \\ 0 & 0 & -14 \end{bmatrix}$$

$$B=\begin{bmatrix} -272.06 \\ 0 \\ 14 \end{bmatrix}$$

D= 0;

5.8 SIMULATION FOR THE 1st OUTPUT

So the transfer function for the first output is:

$$G_1(s) = \frac{-272.06(s^2 + 1.8647s + 84.128)}{(s+14)(s-1.7834)(s+4.9034)}$$

5.8.1 Transfer function via MATLAB :(Appendix-A)

Matlab output: Transfer function: G1(s)

```
  -272.1 s^2 - 507.3 s - 2.289e004
  ---------------------------------
   s^3 + 17.12 s^2 + 34.94 s - 122.4
```

5.8.2 STEP RESPONSE OF G1(s):

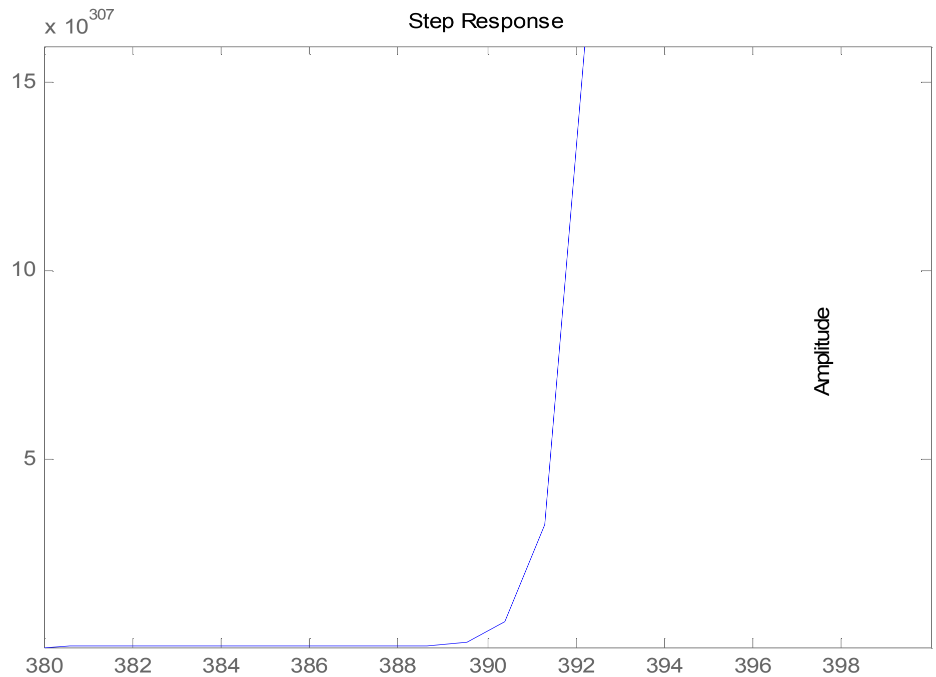


Figure 5.5: step response for G1(s)

From the figure the following criteria can be identified:

- Unstable system
- Rise time is so high

As the system is totally uncontrollable ,so we need to figure out the stability of the system to control the aircraft output.

5.8.3 Tuning via PID for G1(s):(Appendix-A)

After applying PID controller and several trial and error proces we found out the tuning parameter .

kp=-50; ki=-10;kd=-10;

u=-100;   t=0:0.001:20;

5.8.4 SIMUIATION RESULTS:

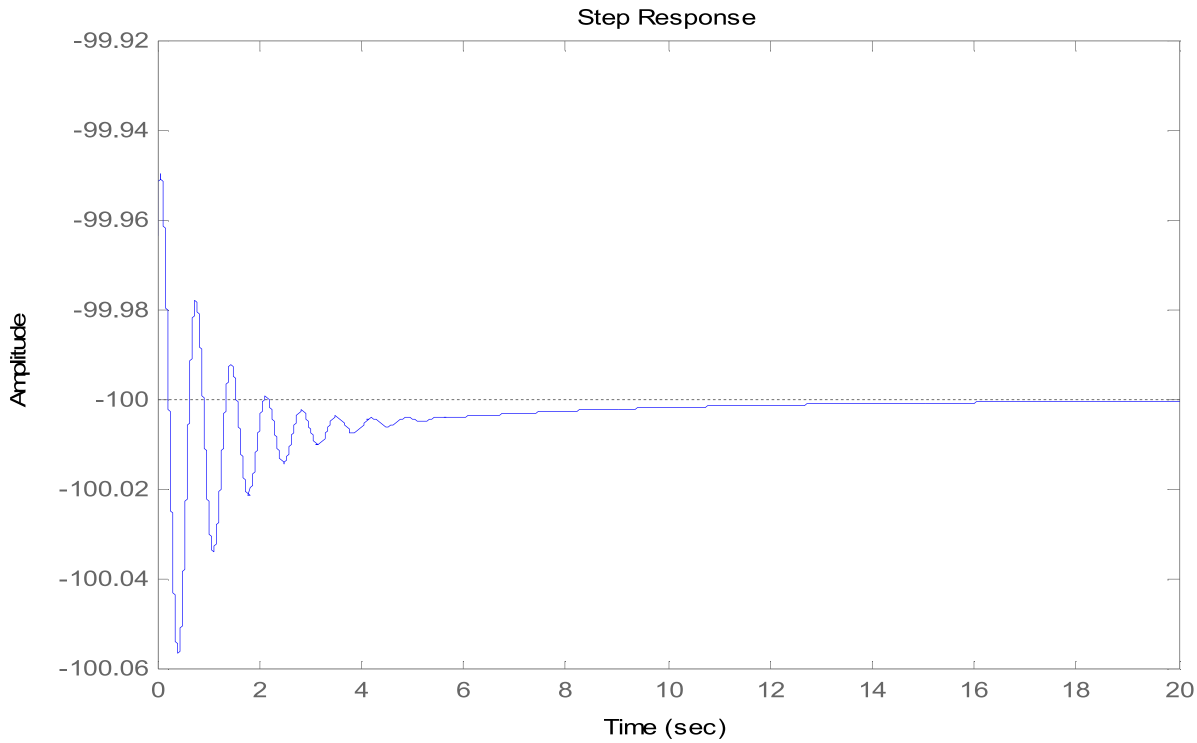


Figure5.6: Tuning of the G1(s)-simulation output using PID

Evaluation :

- ✓ The system is now controlled
- ✓ Steady state error minimized
- ✓ Natural response is controlled
- ✓ Overshoot less(%OS)
- ✓ Settling time less(Ts)
- ✓ Rise time better(Tr)
- ✓ Totally Stable system
- ✓ Aircraft can be controlled with this value of the PID controller parameter

5.9 SIMULATION FOR THE 2nd Output G2(S) :

For $G_2(s)$, $\mathbf{C_2} = (0,1,0)$, and

By manipulating the transfer function for the 2^{nd} Output is:

$$G_2(s) = \frac{-507.71(s+1.554)}{(s+14)(s-1.7834)(s+4.9034)}$$

5.9.1 Transfer function from matlab simulation:(Appendix-A)

Transfer function:G2(s)

-507.7 s - 789

----

s^3 + 17.12 s^2 + 34.94 s - 122.4

5.9.2 Step response of G2(S):

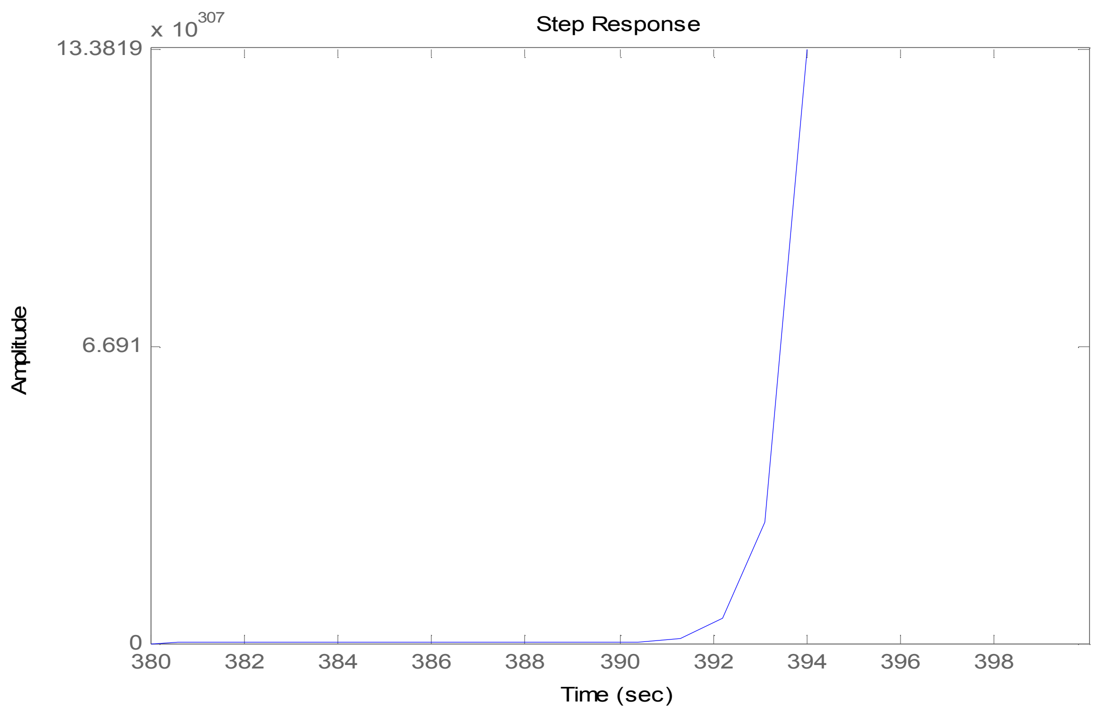


Figure 5.7: step response of G2(s)

From the above figure the following criteria can be identified:

- ❖ Unstable system
- ❖ Rise time is so high

As the system is totally uncontrollable ,so we need to figure out the stability of the system to control the aircraft output.

5.9.3 Tuning via PID controller for G2(s):

Simulation results:

kp=-50;

ki=-10;

kd=-10;

u=-100;

t=4:0.001:50

Output:

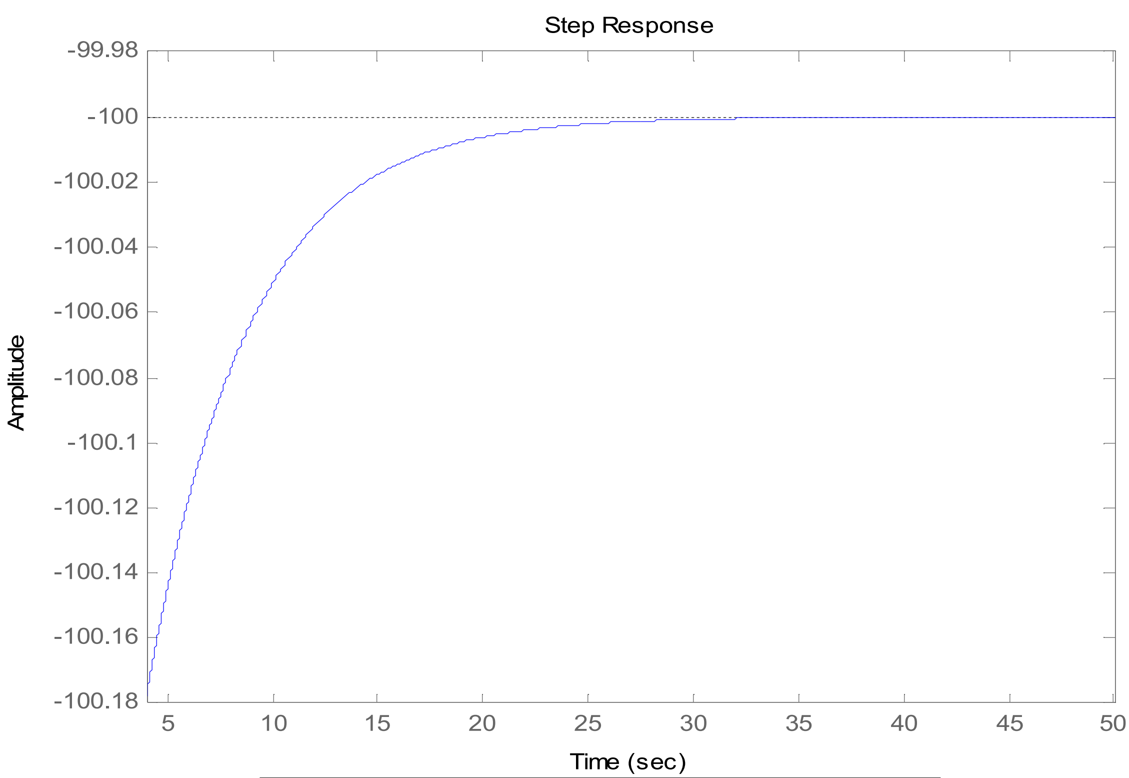


Figure5.8: output of PID controller based G2(s)

Evaluation:

- ✓ The system is now controlled
- ✓ Steady state error minimized
- ✓ Natural response is controlled
- ✓ Overshoot less(%OS)
- ✓ Settling time less(Ts)
- ✓ Rise time better(Tr)
- ✓ Totally Stable system
- ✓ Smooth controllable output
- ✓ Aircraft can be controlled with this value of the PID controller parameter

Chapter-6
OPTIMIZATION ALGORITHM AND DE

6.1 OPTIMIZATION ALGORITHM

In mathematics, statistics, empirical sciences, computer science, or management science, **mathematical optimization** (alternatively, **optimization** or **mathematical programming**) is the selection of a best element (with regard to some criteria) from some set of available alternative.

6.2 What is Optimization?

Optimization is an iterative process by which a desired solution (max/min) of the problem can be found while satisfying all its constraint or bounded conditions.

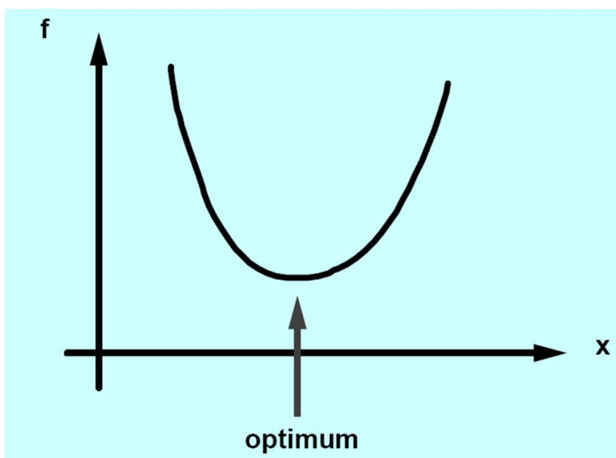


Figure 6.1: Optimization process

Optimization feature:

- ➢ Optimization problem could be linear or non-linear.
- ➢ Non –linear optimization is accomplished by numerical method.
- ➢ Search methods are used iteratively before a solution is achieved.
- ➢ The search procedure is termed as algorithm

6.3 OPTIMIZATION METHOD:

❑ Deterministic - specific rules to move from one iteration to next , gradient, Hessian.

- Direct Search – Use Objective function values to locate minimum

Gradient Based – first or second order of objective function

- Minimization objective function f(x) is used with –ve sign –f(x) for maximization problem.

- ❑ Stochastic – probalistic rules are used for subsequent iteration
- ❑ Optimal Design – Engineering Design based on optimization algorithm
- ❑ Lagrangian method – sum of objective function and linear combination of the constraints

6.4 EVOLUTIONARY ALGORITHM(EA):

Evolutionary algorithm is based on the Darwin‟s principle of „survival of the fittest strategy‟. An evolutionary algorithm begins with initialising a population of candidate solutions to a problem and then new solutions are generated by randomly varying those of initial population. All solutions are evaluated with respect to how well they address the task. Finally, a selection operation is applied to eliminate bad solutions.

.6.5 SOME OPTIMIZATION ALGORITHM(EA):

There exists so many optimization algorithm .Some optimization algorithm are:

1. Ant colony optimization algorithm (ACO)
2. SIMULATED ANNELYING
3. TABU SEARCH (TE)
4. GENETIC ALGORITHM
5. PARTICLE SWARM OPTIMIZATION (PSO)
6. INVASIVE WEED OPTIMIZATION(IWO)
7. DIFFERENTIAL EVOLUTION (DE)
8. BACTERIA AGEING

6.6PROPOSED ALGORITHM -DIFFERENTIAL EVOLUTION(DEA):

6.6.1 Background and Literature Review

A differential evolution algorithm (DEA) is an evolutionary computation method that was originally introduced by Storn and Price in 1995 There are also a number of significant advantages when using DEA, which were summarised by Price in .

- ➢ Ability to find the true global minimum regardless of the initial parameter values;

- Fast and simple with regard to application and modification;
- Requires few control parameters;
- Parallel processing nature and fast convergence;
- Capable of providing multiple solutions in a single run;
- Effective on integer, discrete and mixed parameter optimisation;
- Ability to find the optimal solution for a nonlinear constrained optimization problem with penalty functions.

6.6.2 Basis of Differential Evolution Algorithm

A DEA is a novel evolution algorithm as it employs real-coded variables and typically relies on mutation as the search operator.

DEA is a parallel direct search method that employs a population P of size NP, consisted of floating point encoded individuals or candidate solutions as shown in equation (6.1). At every generation G during the optimization process, DEA maintains a population P(G) of NP vectors of candidate solutions to the problem at hand. (6.1) Each candidate solution Xi is a D-dimensional vector, containing as many integer-valued parameters (6.2) as the problem decision parameters D.

$P^G = [X_1^G,,,,,,,,,,X_2^G,,,,,,,,,,,,X_{N_p}^G]$ ……………………………………(6.1)

Each solution candidate Xi is a D-dimensional Vector, containing as many as integer valued parameters(3.2) as the problem decision parameters D.

$X_i^{(G)} = \left[X_{1.i}^{(G)},,,,,\dots.X_{j.1}^{(G)},,,\dots..X_{D.1}^{(G)}\right], whree\ i = 1,..,,,..,N_p\,; j = 1,2,..,,,,,,,,,D$ ………………………….(6.2)

6.7 Differential Evolution Algorithm Optimization Process

6.7.1 Initialization :

In the first step of the DEA optimization process, the population of candidate solutions must be initialized. Typically, each decision parameter in every vector of the initial population is assigned a randomly chosen value from within its corresponding feasible bounds.

$x_{j,i}^{(G=0)} = x_j^{min} + rand_j[0,1].(x_j^{max} - x_j^{min})$……………………………………….....(6.3)

Where $i = 1,,,,,,, N_p and\ j = 1,,,,,,,,,,, D. x_{j.i}^{(G=0)}$ is the initial value(G=0) of the j^{th} parameter of the i^{th} individual vector. $x_j^{max} and\ x_j^{min}$ are the upper and lower bounds of the j^{th} parameter ,respectively.

6.7.2 Mutation

The DEA optimization process is carried out by applying the following three basic genetic operations; mutation, recombination (also known as crossover) and selection. After the population is initialized, the operators of mutation, crossover and selection create the population of the next generation P(G+1) by using the current population P(G). At every generation G, each vector in the population has to serve once as a target vector Xi(G), the parameter vector has index i, and is compared with a mutant vector. The mutation operator generates mutant vectors (Vi(G)) by perturbing a

Randomly selected vector(X_{r1}) with the difference of two randomly selected vectors(X_{r2} and X_{r3}).

$V_i^{(G)} = X_{r1}^{(G)} + F(\ X_{r2}^{(G)} - X_{r3}^{(G)}, i = 1, \dots \dots \dots \dots, N_p, \dots \dots \dots \dots \dots$(6.4)

Vector indices r1,r2,and r3 are randomly chosen, which r2,r3 and r3 $\in N_p$ and r1$\neq r2 \neq r3 \neq i$. X_{r1}, X_{r2} and X_{r3} are selected anew for each parent vector is a user defined constant known as the “Scaling mutation factor”, which is typically chosen from within the range [0,1].

6.7.3 Crossover

In this step, crossover operation is applied in DEA because it helps to increase the diversity among the mutant parameter vectors. At the generation *G*, the crossover operation creates trial vectors (*Ui*) by mixing the parameters of the mutant vectors (*Vi*) with the target vectors (*Xi*) according to a selected probability distribution.

$$U_i^{(G)} = u_{j,i}^{(G)} = \begin{cases} v_{j,i}^{(G)} & \text{if} \quad \text{rand}_j(0,1) \leq CR \quad or \quad j = s \\ x_{j,i}^{(G)} & \text{otherwise} \end{cases}$$

………..(6.5)

The crossover constant *CR* is a user-defined value (known as the “crossover probability”), which is usually selected from within the range [0,1]. The crossover constant controls the diversity of the population and aids the algorithm to escape from local optima. rand_j is a uniformly distributed random number within the range (0,1) generated anew for each value of *j*. *s* is the trial parameter with randomly chosen index {1,…,*D*}, which ensures that the trial vector gets at least one parameter from the mutant vector.

6.7.4 Selection

Finally, the selection operator is applied in the last stage of the DEA procedure. The selection operator chooses the vectors that are going to compose the population in the next generation. This operator compares the fitness of the trial vector and the corresponding target vector and selects the one that provides the best solution. The fitter of the two vectors is then allowed to advance into the next generation according to

$$X_i^{(G+1)} = \begin{cases} U_i^{(G)} & \text{if} \quad f(U_i^{(G)}) \le f(X_i^{(G)}) \\ X_i^{(G)} & \text{otherwise} \end{cases}$$

……………………(6.6)

6.8 Design of the Differential Evolution Algorithm Optimization Program

Given the basic optimization process of DEA and several variations of mutation operator strategies, DEA optimization program has been designed in this chapter using MATLAB. The overall procedure of the DEA optimization program has been described as follows:

Step 1: Set up all required parameters of the DEA optimization process by the user;

➢ Set up control parameters of the DEA optimization process that are population size (NP), scaling mutation factor (F), crossover probability (CR), convergence criterion (), number of problem

➢ variables (D), lower and upper bounds of initial population (xjmin and xjmax) and maximum number of iterations or generations (Gmax);

➢ Select a DEA mutation operator strategy;

Step 2: Set generation G = 0 for initialization step of DEA optimization process;

Step 3: Initialization step;

➢ Initialize population P of individuals according to equation (3.3) where each decision parameter in every vector of the initial population is assigned a randomly selected value from within its corresponding feasible bounds;

Step 4: Calculate and evaluate the fitness values of the initial individuals according to the problem's fitness function; Step 5: Rank the initial individuals according to their fitness; Step 6: Set iteration G = 1 for optimization step of DEA optimization process;

Step 7: Apply mutation, crossover and selection operators to generate new individuals;

➢ Apply mutation operator to generate mutant vectors (Vi(G)) according to equation (3.4) with a selected DEA mutation operator strategy in step 1;

➢ Apply crossover operator to generate trial vectors (Ui(G)) according to equation (3.5);

Step 8: Calculate and evaluate the fitness values of new individuals according to the problem's fitness function; Step 9: Rank new individuals by their fitness; Step 10: Update the best fitness value of the current iteration (gbest) and the best fitness value of the previous iteration (pbest)

Step 11: Check the termination criteria;

➢ If Xibest - Xi > orpbest - gbest> but the number of current generation remains not over the maximum number of generations G < Gmax, set G = G + 1 and return to step 7 for repeating to search the solution. Otherwise, stop to calculate and go to step 12;

Step 12: Output gbest of the last iteration as the best solution of the problem.

6.9 BASIC PARAMETER OF DE OPTIMIZATION:

❑ **Population Size (N_P)** :The population size of DEA should be moderate. As DEA may converge to local optimum if population size is very small due to its less diversity of discovery. On the other hand, if the population size is very large, DEA would require huge numbers of function evaluations for convergence, which needs tremendously high computation time.

❑ **Mutation Factor (F)**: Mutation factor is a real and constant factor that controls the amplification of the differential variation (Xr2(G) - Xr3(G)) in equation (3.4) and it affects the DEA convergence.

❑ **Crossover Probability (CR) :**

Crossover probability affects the number of variables to be changed in the trial vectors (Ui(G)) compare to the target vectors (Xi(G)). If the value of crossover probability is large, more variables are taken from the mutant vectors (Vi(G)) than the target vectors (Xi(G)).

❑ **Number of Problem Variables (D)** :The number of variables in the objective function depends on the problem size and affects the convergence speed of DEA.

❑ **Convergence Criterion (E)** :Convergence criterion compares two differences of the candidate solution population that are the difference between fitness function values of other members and the best member in the same iteration or the difference between fitness values of the best solution in present iteration and previous iteration.

6.10 OPTIMIZATION PROCESS FLOWCHART:

The flowchart for the DE optimization process is given below:

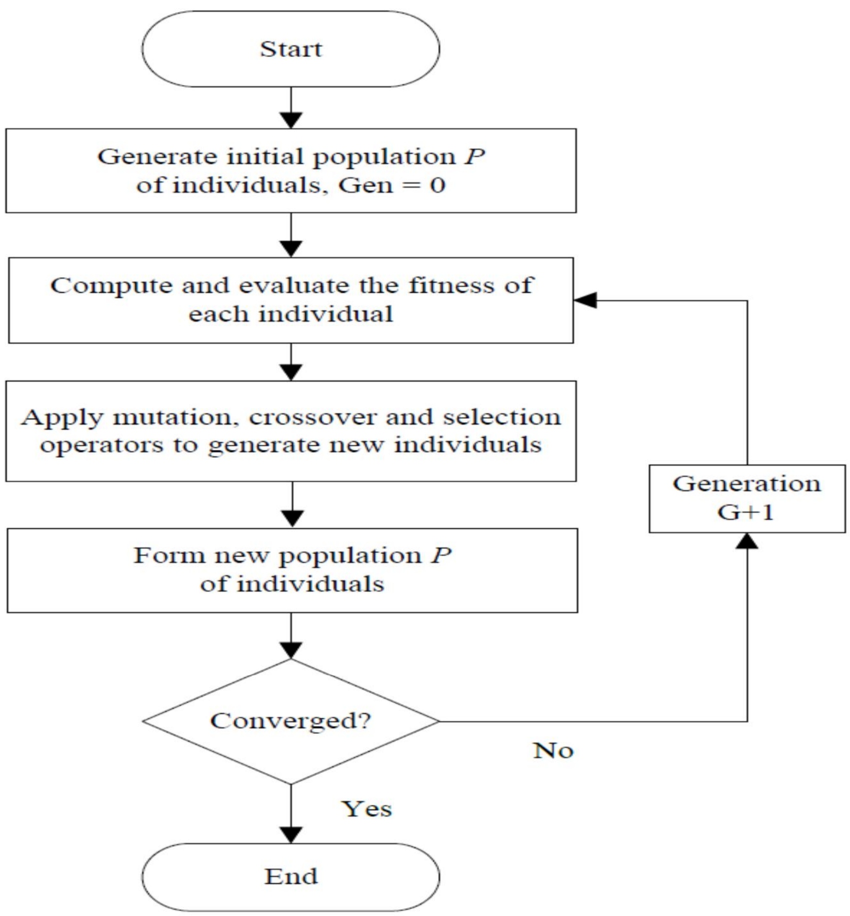


Figure6.2 : Flowchart of overall process of DE optimization

6.11 APPLICATION OF DE ON MISSILE SYSTEM(SISO):

- Before we have simulated missile system via trial and error method using PID controller
- Now we will apply evolutionary algorithm(EA) , Differential Evolution(DE) on that missile system.
- Objective is to find optimized value of Kp,ki and kd by using DE.

6.11.1 TUNING OF THE SISO SYSTEM VIA DE :

TRANSFER FUNCTION :(WITHOUT PID)

$$\boldsymbol{T(s) = \frac{\phi(s)}{\delta(S)} = \frac{K_a s + k_b}{k_3 s^3 + k_2 s^2 + k_1 s + k_0}}$$

TRANSFER FUNCTION:(WITH PID)

$$T_{PID}(s) = \frac{(k_a + k_p)s^3 + (k_a k_p + k_b k_p)s^2 + (k_a k_I + k_b k_p)s + k_b k_I}{k_3 s^4 + (k_a k_p + k_2)s^3 + (k_1 + k_a k_p + k_b k_D)s^2 + (k_o + k_a k_I + k_b k_p)s + k_b k_I}$$

6.11.2 OPTIMIZING PERFORMANCE

- Gain the stability of the system
- Oscillatory natural response minimized
- Error minimized
- Overshoot minimized
- Settling time minimized-----otherwise fault will destroy the system
- Rise time to be less

6.11.3STEP RESPONSE BEFORE DE OPTIMIZATION(SISO):(Appendix-A)

Simulation Result:

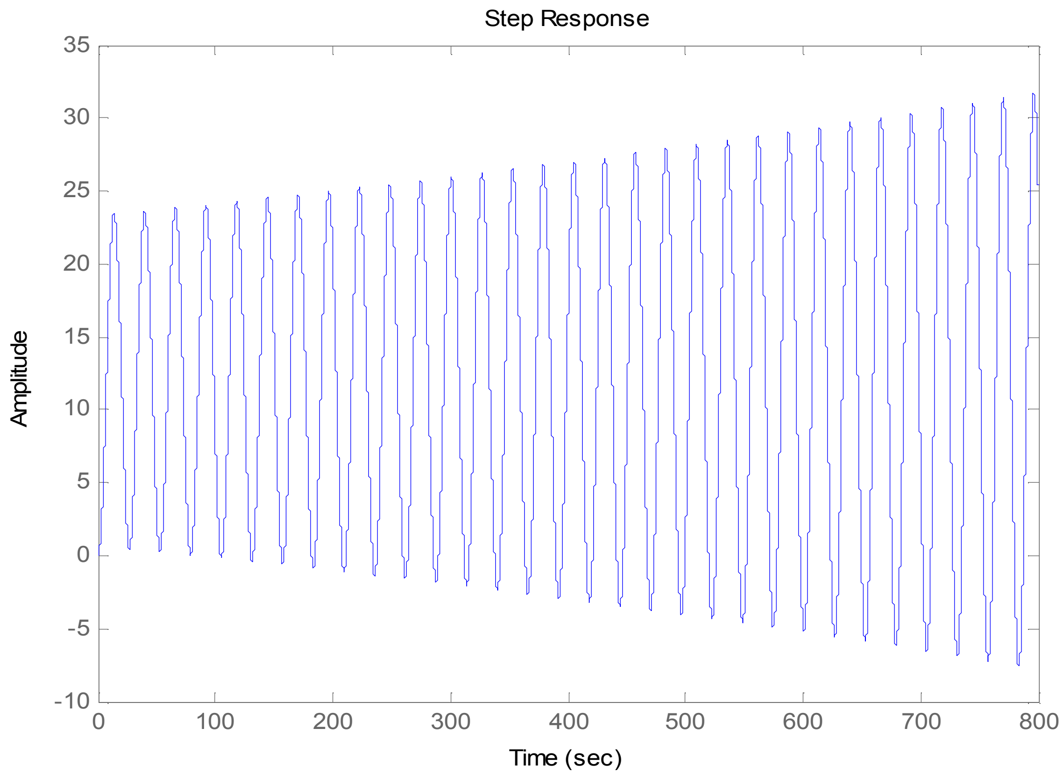


Figure 6.3: Step response of SISO before DE optimization

Performance Criteria :

- ✓ Undamped system
- ✓ Huge oscillation without bound
- ✓ Undamped sunusoid with radian frequency
- ✓ Natural response will not decay over time
- ✓ Unstable system
- ✓ Overshoot criteria is worst
- ✓ Rise time and settling time has no bound
- ✓ Not controllable

6.12 System response after DE optimization via MATLAB: (Appendix-A)

- Settling time is 55 sec
- System stable and less overshoot
- Population Size=20;
- No of Generation=500
- stop=50;
- kp_min=1.2 & kp_max=5
- Kd_min=11 & kd_max=15
- Ki_min=0.28 & ki_max=2.2
- t=0:.01:800;
- Iseed=1;

6.12.1 SIMULATION RESULT: (STEP RESPONSE USING DE)(Appendix-A)

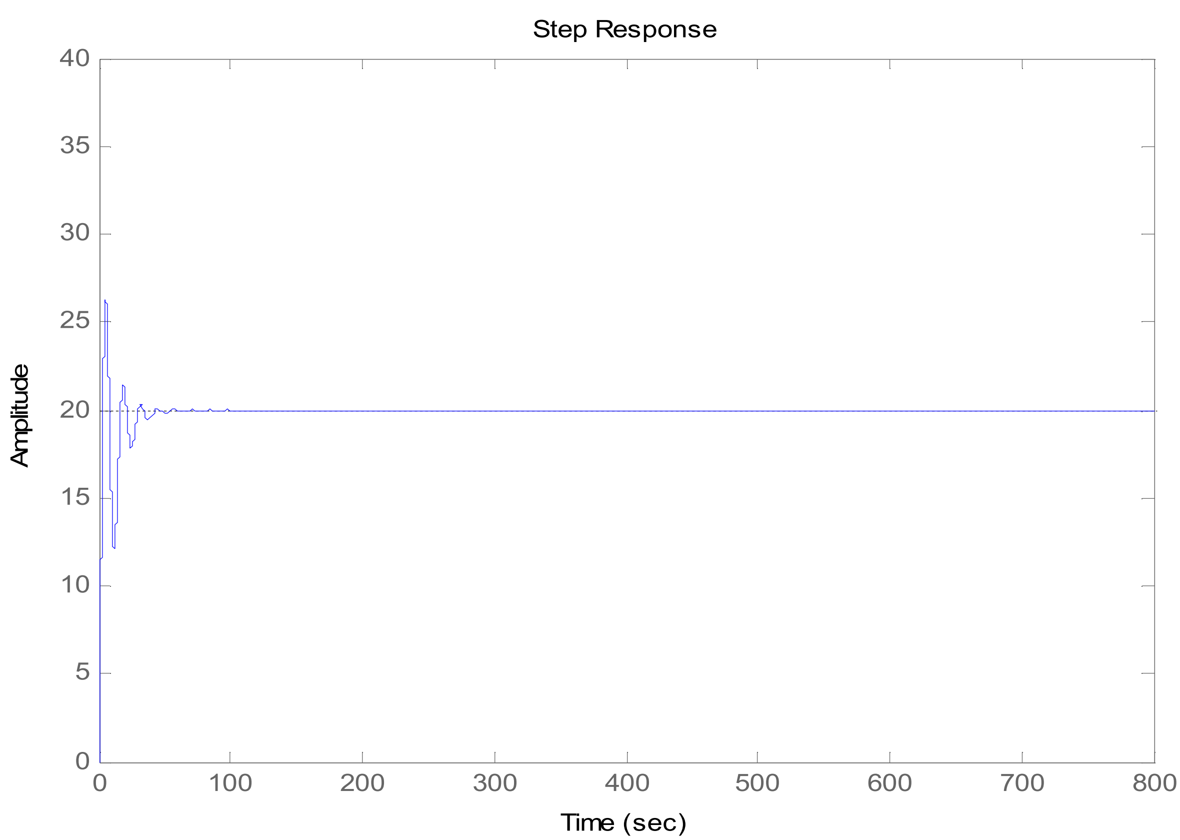


Figure 6.4: Step response of SISO after using DE

Evaluation:

- ✓ The system is now controlled
- ✓ Steady state error minimized
- ✓ Natural response is controlled
- ✓ Overshoot less(%OS)
- ✓ Settling time less(Ts)
- ✓ Rise time better(Tr)
- ✓ Totally Stable system
- ✓ Aircraft can be controlled with this value of the PID controller parameter

Compare with PID:

✓ Time saving
✓ Tuning with more perfect value
✓ Performance is better than PID
✓ In PID tuning does not guarantee the best optimize value but in DE, the optimized value can be obtained
✓ Less erroneous method

6.12.2 Bode Plot:

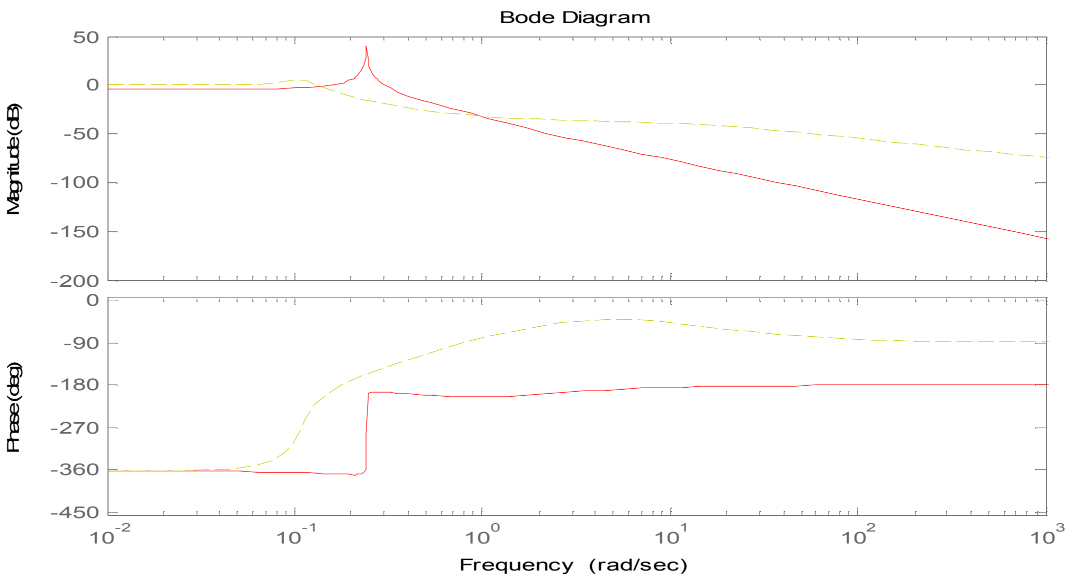


Figure 6.5: Bode plot of DE Simulation

❑ The upper part of the bode plot is for magnitude of the siso system and the green line is for unstable and uncontrolled system, but after optimization we obtained the red line system response

❑ The Lower part is for the frequency response of the system, here we can see the red line system response is better than the uncontrolled green line response.

6.13.3 Root locus :

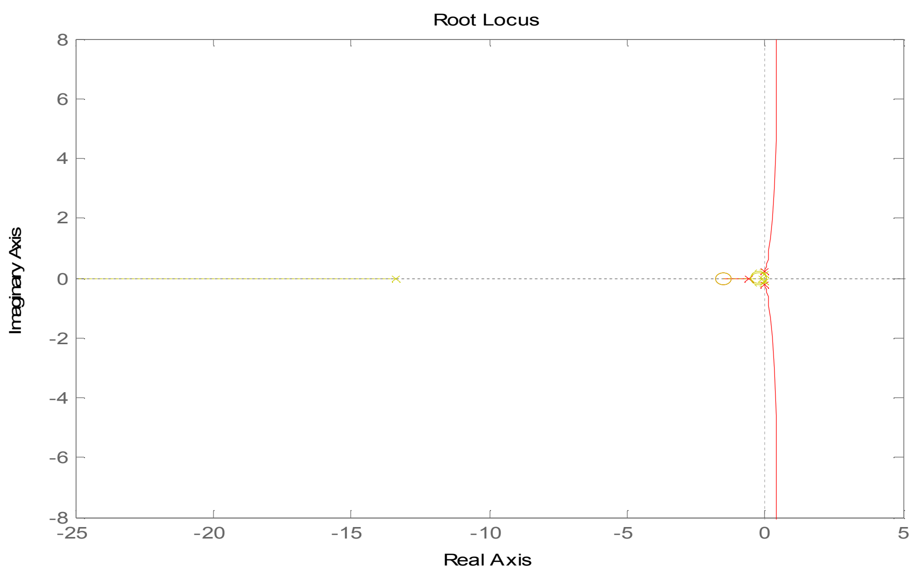


Figure 6.6: Root lucas of the SISO system after DE

➢ From the root Lucas plot ,it is visible that we found out the eugen value that follows the path and the green line follows the path.

➢ Root lucas provides the Phenomena of a given system for optimizing the system with a certain range of eigen value

➢ From the root lucas plot ,we can see that the system is stable and controllable.

6.12.4 NYQUIST DIAGRAM:

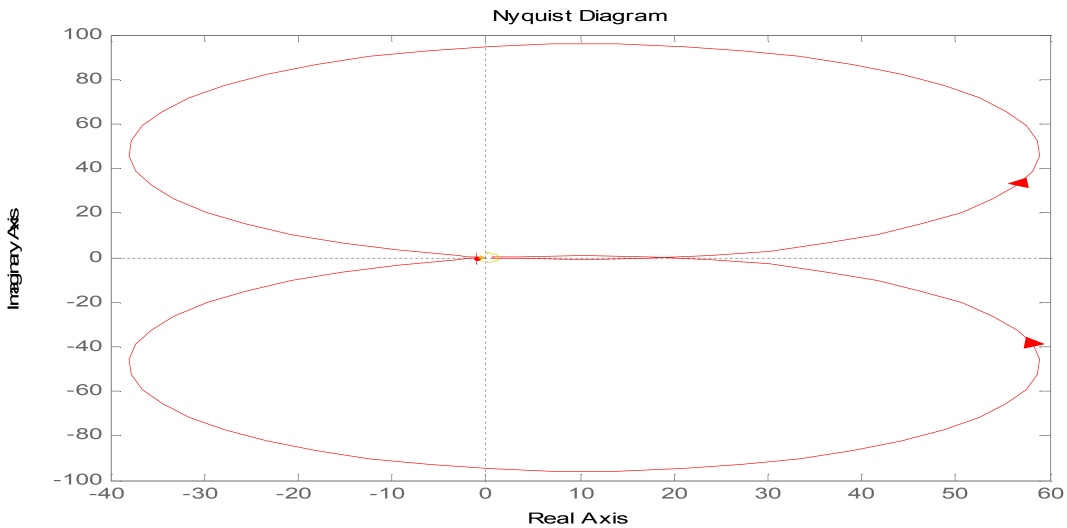


Figure 6.7: Nyquist Diagram of SISO system

6.13 CONCLUCION :

So we have simulated the output of the missile system by using the differential algorithm techniques, and by comparing with conventional PID controller ,DE can tune automatically the PID values which ensures the robustness and reliability of the system.

APPENDIX–A

MATLAB CODES:

A.1 TRANSFER FUNCTION FOR MISSILE

```
%MISSILE CONTROL ----An alternative approach MATLAB CODE%
clear all;
close all;
ka=1;
kb=1.5;
k3=70;
k2=43;
k1=4;
ko=2.5;
num1=[ka kb];
den1=[k3 k2 k1 ko];
Gp=tf(num1,den1);
u=20;
t=0:.1:100;
step(u*num1,den1,t)
axis([0 40  0 30])
```

A.2 THE PID CONTROLLED –TUNING FOR MISSILE

```
%MISSILE CONTROL ----An alternative approach MATLAB CODE%
clear all;
ka=1;
kb=1.5;
k3=70;
k2=43;
k1=4;
ko=2.5;
num1=[ka kb];
den1=[k3 k2 k1 ko];
Gp=tf(num1,den1)          %MISSILE TRANSFER FUNCTION%
kp=100;
ki=100;
kd=300;
num2=[kd kp ki];
den2=[1 0];
Gc=tf(num2,den2)         %PID CONTROLLER TRANSFER FUNCTION%
H1=series(Gp,Gc);        % Using series concept%
H=feedback(H1,1);        %       Using unity feedback%
T=tf(H)
[num,den]=tfdata(T,'v')  % Extracting numerator and denominator from transfer
u=20;
t=0:.1:100;
step(u*num,den,t)
axis([0 30 0 25]);
```

A.3 SIMO SYSTEM –TRANSFER FUNCTION FOR G1(S)

```
clear all;
close all;
A=[-1.702 50.72 263.38;0.22 -1.418 -31.99;0 0 -14];
B=[-272.06;0;14];
C2=[1 0 0];
D=0;
T2=ss(A,B,C2,D);
G2=tf(T2)
[num,den]=tfdata(G2,'v')
u=-100;
t=380:.01:400;
step(u*num,den,t)
```

A.4 PID CONTROLLER APPLIED TO G1(s)

```
% MILITARY AIRCRAFT CONTROL ---FOR TRANSFER FUNCTION G1(s) %
clear all;
close all;
A=[-1.702 50.72 263.38;0.22 -1.418 -31.99;0 0 -14]
B=[-272.06;0;14]
C1=[1 0 0];                 % For the G1 transfer function%
D=0;
T1=ss(A,B,C1,D);
G1=tf(T1)                  % Transfer function of G1(s)%
kp=-50;
ki=-10;
kd=-10;
num2=[kd kp ki];
den2=[1 0];
Gc=tf(num2,den2)          %Transfer function of PID controller%
H1=series(G1,Gc);
H2=feedback(H1,1);          % PID COntroller applied in G1%
T4=tf(H2)
[num1,den1]=tfdata(T4,'v');% Extracting numerator and denominator from
transfer function T1%
u=-100;
t=0:0.001:20;
step(u*num1,den1,t)
   %Output of PID controller on G1%
```

A.5 TRANSFER FUNCTION FOR G2(s)

```
clear all;
close all;
A=[-1.702 50.72 263.38;0.22 -1.418 -31.99;0 0 -14];
B=[-272.06;0;14];
C2=[0 1 0];
D=0;
T2=ss(A,B,C2,D);
G2=tf(T2)
[num,den]=tfdata(G2,'v')
u=-100;
t=380:.01:400;
step(u*num,den,t)
```

A.6 PID CONTROLLED –TUNING FOR G2(s)

```
% MILITARY AIRCRAFT CONTROL ---FOR TRANSFER FUNCTION G2(s) %
clear all;
close all;
A=[-1.702 50.72 263.38;0.22 -1.418 -31.99;0 0 -14]
B=[-272.06;0;14]
C2=[0 1 0];                % For the G1 transfer function%
D=0;
T1=ss(A,B,C1,D);
G1=tf(T1)                  % Transfer function of G1(s)%
kp=-50;
ki=-10;
kd=-10;
num2=[kd kp ki];
den2=[1 0];
Gc=tf(num2,den2)          %Transfer function of PID controller%
H1=series(G1,Gc);
H2=feedback(H1,1);         % PID COntroller applied in G2%
T4=tf(H2)
[num1,den1]=tfdata(T4,'v');% Extracting numerator and denominator from
transfer function T1%
u=-100;
t=4:0.001:50;
step(u*num1,den1,t)
   %Output of PID controller on G2%
```

A.7 DE SIMULATION MATLAB CODES

A.7.1 Arrange PID.m :

```
 %Arrange PID%
Kp=p(1,:);
Ki=p(1,:);
Kd=p(1,:);
```

A.7.2 Cost PID.m :

```
function J=cost_pid_mis(p,display)
if nargin ==1 , display=0; end

arrange_pid;

A_pid_mis= [0 1 0 0 ; 0 0 1 0 ; 0 0 0 1; -1.5/65*Ki -(1.5*Kp+1+Ki)/65 -
(1.5*Kd+3+Kp)/65 -(43/65+Kd)];
eigenvalues=eig(A_pid_mis)
eigenvalues(imag(eigenvalues)<0.01)=[];
% Maximize minimum damping ratio (+ve)
J = -min(-real(eigenvalues)./abs(eigenvalues));

if display==1
    disp('      Eigenvalues     Damping Ratio')
    disp([eigenvalues -real(eigenvalues)./abs(eigenvalues)])
end
```

A.7.3 Diffev_PID.m

```
function [J_best,x_best] =
diffev_pid_mis(x_min,x_max,Pop_Size,No_Gen,stop,F,lambda,Cross,Iseed)
%% Differential Evolution (DE2, 1995 paper)

d=length(x_min); % Number of parameters
[Rnd,Iseed]=ran(Iseed,d,Pop_Size);
x= x_min*ones(1,Pop_Size) + (x_max-x_min)*ones(1,Pop_Size).*Rnd;
for i=1:Pop_Size
    J(i)=cost_pid_mis(x(:,i));
end
[J_best,r]=min(J);
x_best=x(:,r);
diary('diffevo.txt')
disp(num2str([J_best x_best'],6))
diary off
```

```
for j=2:No_Gen
    for i=1:Pop_Size
        [Rnd,Iseed]=ran(Iseed,1,Pop_Size);
        [B,R]=sort(Rnd);
        R(R==i)=[];
        v(:,i)=x(:,i)+F*(x(:,R(1))-x(:,R(2)))+lambda*(x_best-x(:,i));

        for k=1:d
            v(k,v(k,:)>x_max(k))=x_max(k);
            v(k,v(k,:)<x_min(k))=x_min(k);
        end

        for k=1:d-1
            L=k;
            [Rnd,Iseed]=ran(Iseed,1,1);
            if Rnd > Cross
                break
            end
        end
        [Rnd,Iseed]=ran(Iseed,1,d);
        [B,n]=sort(Rnd);
        if mod(n(1)-1,d)<L
            u(:,i)=v(:,i);
        else
            u(:,i)=x(:,i);
        end

        J(j,i)=cost_pid_mis(u(:,i));
        if j>1 && J(j,i)<=min(J(:,i)) % == should be O.K.
            x(:,i)=u(:,i);
        end
    end
    [J_best(j),r]=min(min(J));
    x_best=x(:,r);
    diary('diffevo.txt')
    disp(num2str([J_best(j) x_best'],6))
    diary off
    if (j>stop && J_best(j)==J_best(j-stop)) % | now>stopat
        break
    end
end
```

A.7.4 missile_PID.m:

```
clc
clear all

% close_sys=tf([1 1.5], [70 43 4 2.5] );
% step(close_sys)

param_no=3;
x_min(1) = 1; x_max(1)= 10;
x_min(2) = 0.08; x_max(2)=0.15;
x_min(3) = 11; x_max(3)= 15;
```

```
x_min=x_min'; x_max=x_max';

Pop_Size=20;
No_Gen=500;
stop=50;
Iseed=1;

% % PSO
% weight=0.9; alpha=.99; N=30;
% [J_pso,x_pso] =
pso_pid(x_min,x_max,Pop_Size,No_Gen,stop,weight,alpha,N,Iseed);
% p=x_pso;

%DE
F=.9; lambda=.7; Cross=.9;
[J_diff,x_diff] =
diffev_pid_mis(x_min,x_max,Pop_Size,No_Gen,stop,F,lambda,Cross,Iseed);
p=x_diff;

Kp=p(1,1);
Ki=p(2,1);
Kd=p(3,1);

%ZN
% Kp=0.636;
% Ki=0.04922;
% Kd=1.029;

%TL
% Kp=0.47;
% Ki=0.0083;
% Kd=1.96;

%IM
% Kp=0.06;
% Ki=0.09;
% Kd=0.61;

%Paper_PSO
% Kp=3.26;
% Ki=0.13;
% Kd=13.61;

K3=70;
num= [1 1.5];
den=[K3 43 4 2.5];
A_mis=[0 1 0; 0 0 1; -2.5/K3 -4/K3 -43/K3];
B=[0 0 1/K3]';
C=[1.5 1 0];
D=0;
EIG1=eig(A_mis);
sys1=ss(A_mis,B,C,D)
```

```
A_pid_mis= [0 1 0 0 ; 0 0 1 0 ; 0 0 0 1; -1.5/K3*Ki -(1.5*Kp+1+Ki)/K3 -
(1.5*Kd+3+Kp)/K3 -(43/K3+Kd)];
B_pid_mis=[0 0 0 1/K3]';
C_pid_mis=[1.5*Ki Ki+1.5*Kp Kp+1.5*Kd Kd];
D_pid_mis=0;
EIG2=eig(A_pid_mis);
sys2=ss(A_pid_mis,B_pid_mis,C_pid_mis,D_pid_mis)

% bode(sys1,'r',sys2,'y--')
% rlocus(sys1,'r',sys2,'y:')
% nyquist(sys1,'r',sys2,'y--')

% break
sys=ss(A_mis,B,C,D);
step(sys)
hold on

% step(sys2)
% hold off

num1= [1 1.5];
den1=[K3 43 4 2.5];
numc=[Kd, Kp, Ki];
denc=[1 0];
numa=conv(num1,numc);
dena=conv(den1,denc);
[numac,denac]=cloop(numa,dena);
step(numac,denac)
hold off
```

A.7.5 ran.m

```
function  [R,ISEED]=ran(ISEED,n,m)
% Random number generator with a seed

IA=7141; IC=54773; IM=259200;
for i=1:n
    for j=1:m
        ISEED=mod(ISEED*IA+IC,IM);
        R(i,j)=ISEED/IM;
    end
end
```

REFERENCES

1. Özgören, M. K., Seminar Notes on Dynamics and Control of Guided Missiles, Middle East Technical University Continuing Education Center, February 21, 1991
2. Frarr, D. j., “Control of Missile Airframes”, British Aerospace Dynamics Group Report, 1979
3. Zarchan, P., Tactical and Strategic Missile Guidance, Vol. 157, Progress in Aeronautics and Astronautics, AIAA, Washington DC, 1994
4. Lin, C. F., Modern Navigation, Guidance and Control Processing,Prentice Hall Publication, Englewood Cliffs, New Jersey, 1991
5. Berglund, E., “Guidance and Control Technology”, RTO SCI Lecture Series on Technologies for Future Precision Strike Missile Systems, Atlanta, USA, pp. 1-10, March 2000
6. Tanrıkulu, Ö., “Non-Linear Flight Dynamics of Unguided Missiles”, Ph.D. Thesis, Middle East Technical University, Türkiye, 1999
7. Mahmutyazıcıoğlu, G., “Dynamics and Control Simulation of an inertially Guided Missile”, M.Sc. Thesis, Middle East Technical University, Türkiye, 1994.
8. Acar, Ş.U., “Trajectory Tracking by Means of Homing Guidance Methods”, M.Sc. Thesis, Middle East Technical University, Türkiye,1996
9. Tiryaki, K., “Polynomial Guidance Laws and Dynamic Flight Simulation Studies”, M.Sc. Thesis, Middle East Technical University,Türkiye, 2002
10. Şahin, K. D., “A Pursuit Evasion Game between an Aircraft and a Missile”, M.Sc. Thesis, Middle East Technical University, Türkiye, 2002
11. Vural, A. Ö., “Fuzzy Logic Guidance System Design for Guided Missiles”, M.Sc. Thesis, Middle East Technical University, Türkiye,2003
12. Mendanhall, M. R., Perkins, S. C. and Lesieutre, D. J., “Prediction of the Nonlinear Aerodynamic Characteristics of Maneuvering Missiles”, Journal of Spacecraft, Vol. 24, pp. 394-402, September-October 1987
13. McFarland, M. B. and Calise, A. J., “Neural Adaptive Nonlinear Autopilot Design for an Agile Anti-Air Missiles”, Proceedings of the AIAA Guidance, Navigation and Control Conference, San Diego, California, July 1996
14. Babu, K. R., Sarma, I. G. and Swamy, K. N., “Two Robust Homing Missile Guidance Laws Based on Sliding Mode Control Theory”, IEEE Proceedings, pp. 540-547, 1994
15. Mickle, M. C. and Zhu, J. J., “Skid to Turn Control of the APKWS Missile using Trajectory Linearization Technique”, Proceedings of the American Control Conference, Arlington, VA, pp. 3346-3351, June 25-27, 2001 215

16. Han, D., Balakrishnan, S. N. and Ohlmeyer, E. J., “Optimal Midcourse Guidance Law with Neural Networks”, Proceedings of the IFAC 15th Triennial World Congress, Barcelona, Spain, 2002.
17. Lin, C. L. and Chen, Y. Y., “Design of Advanced Guidance Law against High Speed Attacking Target”, Proceeding of National Science Council, ROC(A), Vol. 23, No. 1, pp. 60-74, 1999
18. Pastrick, H. L., Seltzer, S. M. and Warren, M. E., “Guidance Laws for Short-Range Tactical Missiles”, Journal of Guidance and Control,Vol. 4, No. 2, March- April 1981
19. 19. Wang, Q., Lin, C. F. and D’Souza, C. N., “Optimality-Based Midcourse Guidance”, Proceedings of the American Institute of Aeronautics and Astronautics, pp. 1734-1737, 1993
20. Wang-Xiao Kan, Sun Zhong-Liang, Wnglei, Feng Dong-qing. “Design and research based on fuzzy self-tuning PID using Matlab”. International Conference on
21. Advanced Computing theory and Engineering (2008).Liu Fan, Er Meng Joo. “Design for Auto-tuning PID Controller Based on Genetic Algorithms”. IEEE Conference on Industrial Electronics and Applications ICIEA2009
22. B. Nagaraj, S. Subha, B.Rampriya. “Tuning Algorithms for PID Controller Using Soft Computing Techniques.
23. http://www.math.ucdavis.edu/~kouba/CalcOneDIRECTORY/maxmindirectory/MaxMin.html
24. cavallo.a.de ,a parameter space design,journal of guidance ,control and dynamics vol:15,1992