DESIGN OF CONTROLLERS FOR NONLINEAR DYNAMICAL SYSTEMS

DISSERTATION/THESIS
SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

MASTER OF TECHNOLOGY
IN
CONTROL AND INSTRUMENTATION

Submitted by:

SREEVIDYA B

2K12/C&I/021

Under the supervision of

SUDARSHAN K VALLURU

DEPARTMENT OF ELECTRICAL ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bhawana Road, Delhi-110042

2014

DEPARTMENT OF ELECTRICAL ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bhawana Road, Delhi-110042

CERTIFICATE

I, Sreevidya B, Roll No. 2K12/C&I/021 student of M. Tech. (Control and Instrumentation), hereby declare that the dissertation/project titled "Design of Controllers for Nonlinear Dynamical Systems" under the supervision of Mr. Sudarshan K Valluru of Electrical Engineering Department, Delhi Technological University in partial fulfillment of the requirement for the award of the degree of Master of Technology has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi **SREEVIDYA B**

Date:

SUDARSHAN K VALLURU

SUPERVISOR

Associate professor

Department of Electrical Engineering

Delhi Technological University

ACKNOWLEDGMENT

I would like to express my gratitude towards all the people who have contributed their precious time and efforts to help me complete this project; without which it would not have been possible for me to complete this project. I would like to thank Professor. Madhusudan Singh and Mr. Sudarshan K Valluru, Department of Electrical Engineering, Delhi Technological for his guidance, support, motivation and encouragement from the start of this project to its completion.

SREEVIDYA B

M.Tech, Control &

Instrumentation

Roll. No. 2K12/C&I/021

Delhi Technological University

ABSTRACT

Practically almost all nonlinear dynamical systems are complex due to presence of unmodelled system dynamics and system parameters variations. Typically nonlinear dynamical system design is able to fulfill certain permissible performance level by adaptive control mechanism. However the adaptive control mechanism used for the control of nonlinear dynamical systems does not offer reasonable performance due to the ill-defined mathematical modelling. In this project work some benchmark nonlinear dynamical systems are being considered and their responses with conventional PID, Metaheuristic algorithms tuned linear and nonlinear PID controllers are being evaluated. The metaheuristic algorithm‘s efficacy are tested by using popular benchmark functions.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

m = Mass

β = Coefficient of damping

k = Coefficient of stiffness

x = Displacement

F = Force (input)

τ_m = Motor torque

τ_p = Torque applied on robotic manipulator

N = Gear ratio

θ_m = Pendulum angle

θ_p = Motor shaft position

L_a = Inductance of armature

R_a = Resistance of armature

K_b = Back emf constant

i_a = Current in armature

V = Voltage applied

g = Gravitational constant

l = Length of pendulum rod

ω = Angular velocity

CHAPTER 1

INTRODUCTION

Nonlinear control systems are control systems where nonlinearity plays a substantial role, either in the controlled process (plant) or in the controller itself. Nonlinearity in plants arises naturally as well as sometimes deliberately included, for desired performance. In electro mechanical, biological, aerospace, automotive and industrial control systems the above said nonlinearity can be seen frequently. Various control methods are used to minimize a pre-determined cost function of control objective in optimal systems. Nonlinear system stabilization has been a major area of interest for engineering research since a long time. Various nonlinear methods have been used overtime to increase the efficiency, accuracy, performance and robustness of nonlinear systems.

1.1 LITERATURE REVIEW

A number of methods can be found in literature for tuning of classical PID controller like Ziegler-Nichols tuning method [1], Cohen-Coon method, system identification technique etc. These tuning methods do work good for most of the plants and give satisfactory results for linear systems, but when the system is nonlinear, to increase the robustness and performance other methods of tuning PID parameters are to be devised. One of the latest tuning methods in this field is using metaheuristic algorithm based fine tuning of PID parameters. One of the metaheuristic algorithms which are found to be very efficient in tuning of PID is Genetic algorithm (GA). It is highly robust mainly because the efficiency of this algorithm does not depend upon the characteristics of the plant being controlled. Hence they can be used for a variety of linear and nonlinear plant control [2, 3, 4]. In the year 1995 James Kennedy and Russell Eberhart developed bio-inspired algorithm called particle swarm optimization (PSO) which was inspired by the social interaction [5] of birds and fishes [6]. Now PSO algorithm is being used for a number of applications like for example searching PID parameters to achieve the expected step response of the plant, controlling the plant for varying set point changes and disturbance interference [7, 8] and for tuning parameters of controller for various nonlinear systems like vehicle navigation systems, spacecraft altitude design stabilization system, PMDC motor, gantry crane system etc. [9, 10, 11, 12]. For some nonlinear systems a classical nonlinear PID controller may not give the best control and performance, in such cases a nonlinear PID controller acknowledges the nonlinearities and offers a wide range of operation. In a nonlinear PID controller the parameter values change according to the output response as the variable coefficients of the PID are dependent on the difference between the immediate output and the preferred output [5].Various types of nonlinear PID controllers have been designed in the past with different control laws as suitable for the nonlinear plant under consideration. A nonlinear PID controller is put forth in [13] for a superconducting magnetic energy storage (SMES) system which overcomes the limitation of linear PID and also increases the reliability and operating range for the power system containing the SMES system. Similarly a nonlinear controller was designed for a fixed bed bio reactor where the controller architecture consisted of a linearizing control law in addition to a proportional + derivative reduced order observer that detects the varying parameters of the bio reactor and controls the system [14].

1.2 OBJECTIVES

The Aim and objective of this project is to design controllers for various nonlinear dynamical systems

i. Design a linear PID controller with parameters tuned by metaheuristic algorithms like particle swarm optimizer and genetic algorithm.

ii. Design nonlinear PID controller to find best values of K_p, K_i and K_d values tuned by metaheuristic algorithms.

iii. Compare the performance of the above designed controllers with conventional PID controller as well as with each other for three nonlinear dynamical systems considered, namely- nonlinear Mass-Spring-Damper system, PMBDC motor with robotic manipulator system and Inverted Pendulum on progressing cart system

1.3 OUTLINE OF DISSERTATION

This Project work is divided into six chapters. Chapter 1 gives a brief introduction and review about the work already done in this field; also a brief explanation of the aims and objective of the project work has been explained. The second chapter analyses the nonlinear systems and mathematical modelling of the system is presented. Chapter 3 starts with an overview of the controllers to be designed and explains briefly about the metaheuristic algorithms used for the tuning. Chapter 4 tests the algorithms formulated with various benchmark functions, analyses and compares their performances. In Chapter 5 the simulation studies of nonlinear systems are given and Chapter 6 concludes the thesis with results, conclusions and further scope of the project.

CHAPTER 2
MATHEMATICAL MODELLING

2 . 1 NON-LINEAR MASS-SPRING-DAMPER SYSTEM

An oscillating system consists mainly of a kinetic energy storing element called the inertia or mass, a potential energy storing system i.e. stiffness and an energy dissipating element, damper. Non-linearity in the mass-spring-damper system are due to these components solely .The surface friction between adjacent moving parts can be caused by damping effect in vibratory system or it may occur due to plastic deformation and internal friction between layers of the material of the part. The two above causes may not be completely eliminated as they are uncontrollable as such. The third damping source is the use of mechanical viscous dampers and this type can be used with a known value (damping coefficient β) to get the required damping. Added to this in an actual practical system, the stiffening or weakening of the spring over time, as it compresses and elongates leads to a nonlinear response.

The system being considered here can be explained as follows consists of a total mass m that slides vertically up and down while being attached by a dash pot and a spring on its lower side to a rigid plane. The whole system consisting of the mass, spring and the dashpot can be diagrammatically represented as a simple parallel connection of dashpot and spring elements in series with mass

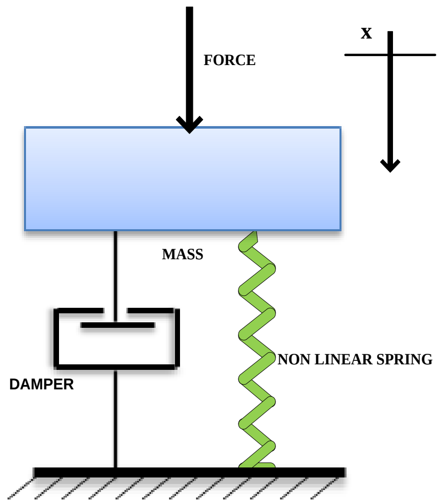

Figure 2.1 Physical representation of a Mass-Spring-Damper system

2.1.1 Differential equation

A simple nonlinear oscillatory system is generally described by the equation

$$m\frac{d^2x}{dt^2} + f\left(x, \frac{dx}{dt}, \beta, k\right) = 0 \tag{2.1}$$

Here the Mass or inertia of the system is denoted by m (lbf), damping coefficient by β (lbf/inch/sec), and k represents the spring coefficient measured in lbf/inch. The nonlinear load displacement curve of the spring is represented by the function f .if an external force F is acted upon the mass causing a displacement in the system then the equation changes to

$$m\frac{d^2x}{dt^2} + f\left(x, \frac{dx}{dt}, \beta, k\right) = F \tag{2.2}$$

Since we are considering a nonlinear spring, we are assuming that the spring stiffness function is given by

$$k = k_1x + k_2x^3 \tag{2.3}$$

Then the nonlinear differential equation for the system becomes

$$m\frac{d^2x}{dt^2} + k_1x + k_2x^3 + \beta\frac{dx}{dt} = F \tag{2.4}$$

2.1.2 State space modelling

Considering two state space variable for the system x_1 and x_2 such that

$$x_1 = x \tag{2.5}$$

$$x_2 = \frac{dx}{dt} \tag{2.6}$$

Then,

$$\dot{x_1} = \frac{dx}{dt} = x_2 \tag{2.7}$$

$$\dot{x_2} = \frac{d^2x}{dt^2} \tag{2.8}$$

Rearranging equation (2.4) we get

$$m\dot{x_2} + k_1x_1 + k_2x_1^3 + \beta x_2 = F \tag{2.9}$$

$$\dot{x_2} = -\frac{k_1x_1}{m} - \frac{k_2x_1^3}{m} - \frac{\beta x_2}{m} - \frac{F}{m} \tag{2.10}$$

Then we can write system compactly as

$$\dot{x} = f(x) + g(x)u + d \tag{2.11}$$

Where,

$$f(x) \triangleq \begin{bmatrix} x_2 \\ \frac{-k_1x_1}{m} - \frac{k_2x_1^3}{m} - \frac{\beta x_2}{m} \end{bmatrix}$$

$$g(x) \triangleq \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}$$

$$d \triangleq 0$$

The system was modelled in SIMULINK and is shown in Figure 2.2.The mass *m* is taken as 100 lbf and the spring stiffness coefficients were taken $k_1 = 10^5$ and $k_2 = 3 \times 10^{11}$. damping coefficient β is considered 100 lbf/ inch/sec.

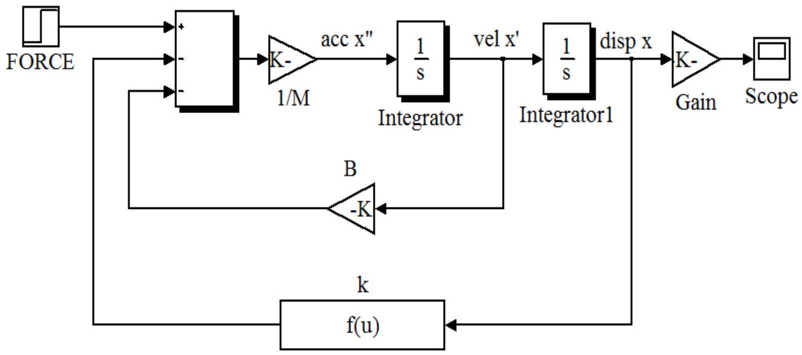

Figure 2.2 SIMULINK representation of Mass-Spring-Damper system

2 . 2 PMBDC MOTOR WITH ROBOTIC MANIPULATOR

Robotic systems are extensively being used in industries today. These robotic systems are used generally in medical, defense fields, state space maintenance etc. Precise control of position of the robotic manipulator is very necessary in these fields .Hence the demand for high level of flexibility and sophistication in joints and its mechanism has led to major research effort in control of robotic structures, particularly the robotic arms. The control of such robotic arms has to surpass many significant control problems like static deflection or vibrations which arise due to inherent nonlinearity or dynamic behavior of the material itself. These nonlinearities tend to have adverse effect on the end point accuracy, making the system slower and also complicate the control schemes.

Here a simple model of PMBDC motor controlled single joint robotic manipulator is taken. The robot manipulator's movement is controlled by a dc motor through a gear. The speed of the PMBDC motor is being adjusted by armature control method. The moment of inertia of the system has been neglected as it is considered very less in comparison to that of the robot manipulator. Gear train is assumed to have no backlash and the shafts rigid.

For the robotic manipulator a clockwise rotation is considered to be negative and counter clockwise as positive, vice a versa for the dc motor shaft rotation. Figure 2.3 PMBDC motor controlled single - link manipulator represents the system diagrammatically.

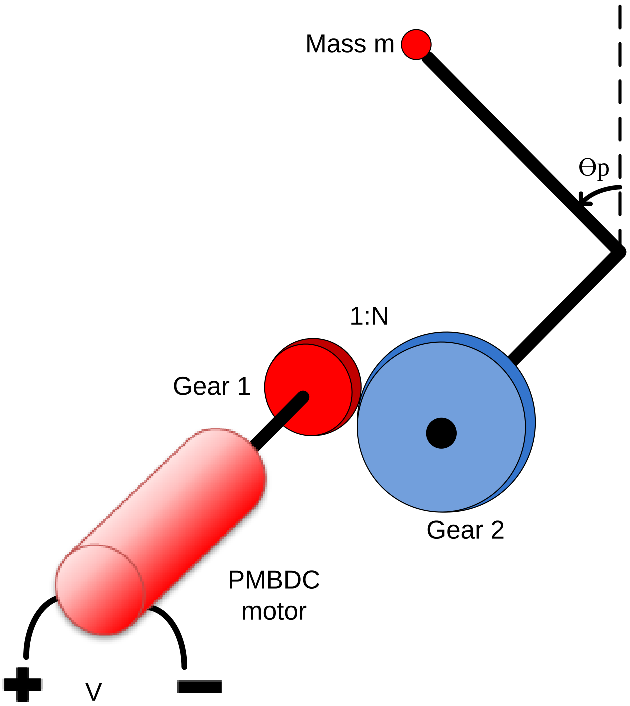

Figure 2.3 PMBDC motor controlled single - link manipulator

2.3.1 Differential equation

The torque developed (τ_m) by the motor is given by

$$\tau_m = K_m i_a \tag{2.12}$$

Where i_a is the armature current and K_m is the torque constant of the motor. The gear ratio N is given by

$$\frac{\theta_p}{\theta_m} = \frac{1}{N} \tag{2.13}$$

Since the gears are in contact

$$\theta_p \times arm\ gear\ radius = \theta_m \times motor\ gear\ radius$$

Since the gears are proportional to their respective number of teeth; the work done by the gears must be equal

$$\tau_p\theta_p = \tau_m\theta_m \tag{2.14}$$

Where τ_p is the torque applied to the robotic joint.

$$\tau_p = N\tau_m = NK_m i_a \tag{2.15}$$

By newton's second law equation of manipulator dynamics can be written as

$$I_a \frac{d^2\theta_p}{dt^2} = mglsin\theta_p + \tau_p \tag{2.16}$$

Substituting equation (15) in equation (16) and rearranging we get

$$ml^2 \frac{d^2\theta_p}{dt^2} = mglsin\theta_p + NK_m i_a \tag{2.17}$$

Here g is the gravitational constant having value 9.8 m/sec^2

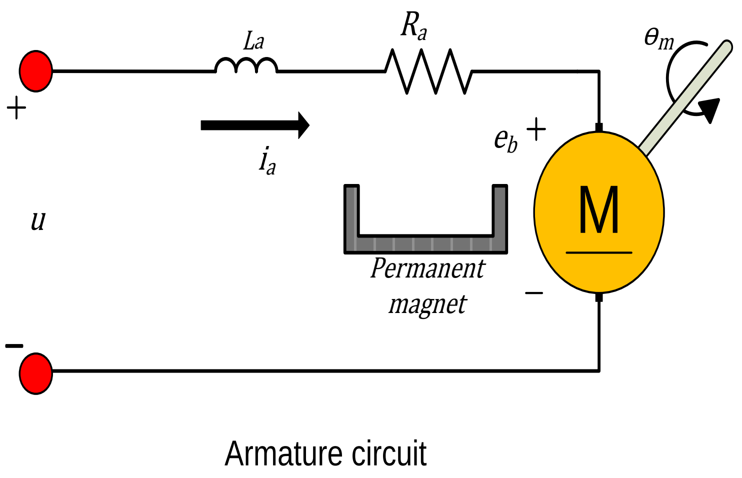

Figure 2.4 Voltage controlled PMBDC motor schematic

By applying Kirchoff's law in the armature circuit we get the equation

$$L_a \frac{di_a}{dt} + R_a i_a + e_b = V \tag{2.18}$$

Since,

$$e_b = K_b N \frac{d\theta_p}{dt} \tag{2.19}$$

The equation becomes

$$L_a \frac{di_a}{dt} + R_a i_a + K_b N \frac{d\theta_p}{dt} = V \quad (2.20)$$

Where,

L_a = Inductance of the armature

R_a = Resistance of the armature

K_b = Back emf constant

θ_p = position of motor shaft

i_a = current in armature

N = Gear ratio

V = Applied voltage

2.3.2 State space modelling

Model of single link robotic manipulator is a nonlinear single input single output (SISO) dynamical control system which is affine in the control input. Such a system can be represented by

$$\dot{x} = f(x) + g(x)u + d$$

$$y = h(x)$$

Assuming three state variables for the system x_1, x_2 and x_3, where

$$x_1 = \theta_p$$

$$x_2 = \frac{d\theta_p}{dt} = \omega_p$$

$$x_3 = i_a$$

Equations (2.17), (2.20) becomes as follows

$$ml^2\dot{x_2} = mglsinx_1 + NK_m x_3 \quad (2.21)$$

$$L_a\dot{x_3} + R_a x_3 + K_b N x_2 = V \quad (2.22)$$

Then,

$$\dot{x_1} = \frac{d\theta_p}{dt} = x_2 \tag{2.23}$$

$$\dot{x_2} = \frac{d^2\theta_p}{dt^2} = \frac{g}{l} sinx_1 + \frac{NK_m}{ml^2} x_3 \tag{2.24}$$

$$\dot{x_3} = \frac{u}{L_a} - \frac{K_b N}{L_a} x_2 - \frac{R}{L_a} x_3 \tag{2.25}$$

Hence it can be represented as,

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{g}{l} sinx_1 + \frac{NK_m}{ml^2} x_3 \\ -\frac{K_b N}{L_a} x_2 - \frac{R}{L_a} x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L_a} \end{bmatrix} V \tag{2.26}$$

$$y = x_1 \tag{2.27}$$

The values of parameters taken are

$l = 1m$

$m = 1Kg$

$N = 10$

$K_m = 0.1Nm/A$

$K_b = 0.1Vsec/rad$

$R_a = 10\Omega$

$L_a = 100mH$

The Simulink representation of the system is given in Figure 2.5

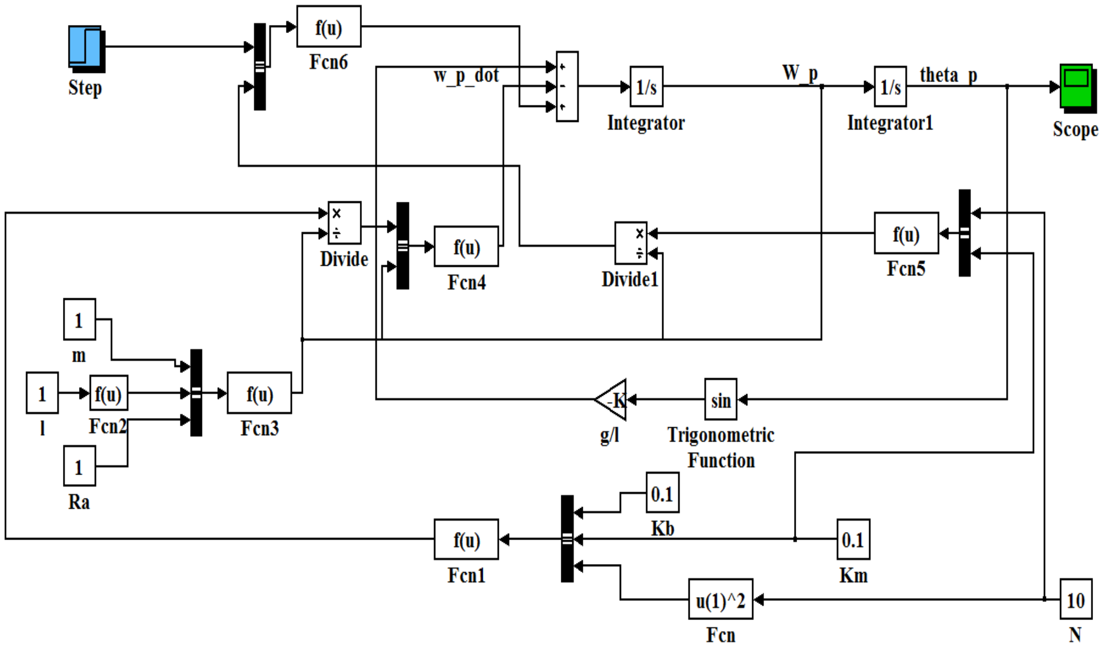

Figure 2.5 Simulink model of motor with robotic manipulator

2 . 3 Inverted Pendulum On Progressing Cart

The system under consideration here is a widely studied highly nonlinear dynamic control problem. Here a two dimensional version of inverted pendulum on progressing cart is analyzed. The pendulum movement is constrained to vertical plane as shown in the Figure 2.6 while the cart can move only in the horizontal plane. The control input F is used to control the motion and hence the displacement x of the car in the horizontal plane while indirectly controlling the angular position of the pendulum angle θ.

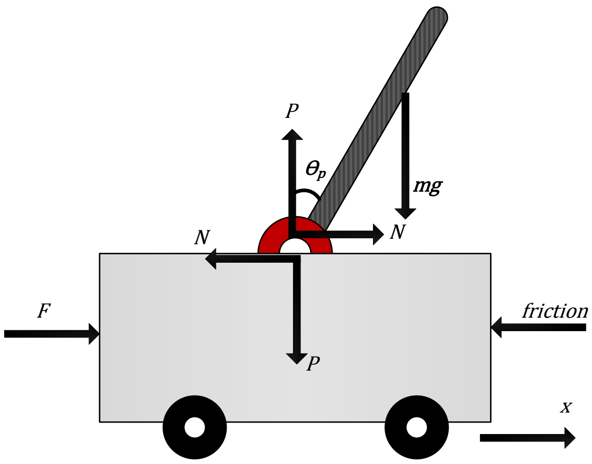

Figure 2.6 Schematic of inverted pendulum on a cart

2.2.1 Differential equation

Rotational motion of pendulum rod about its Centre of gravity is given by

$$I\frac{d^2\theta_p}{dt^2} = \boldsymbol{V}lsin\theta_p - \boldsymbol{H}lcos\theta_p \tag{2.28}$$

Horizontal motion of Centre of gravity of pendulum rod is described as

$$m\frac{d^2}{dt^2}\left(x + lsin\theta_p\right) = \boldsymbol{H} \tag{2.29}$$

Vertical motion of Centre of gravity of pendulum rod

$$m\frac{d^2}{dt^2}\left(lcos\theta_p\right) = V - mg \tag{2.30}$$

Horizontal motion of the cart

$$M\frac{d^2x}{dt^2} = F - \boldsymbol{H} \tag{2.31}$$

Differentiating equation (2.29) and combining with equation (2.31)

$$m\ddot{x} + ml\ddot{\theta}_p cos\theta_p - ml\dot{\theta}_p^{\ 2} sin\theta_p = F - M\ddot{x} \tag{2.32}$$

Differentiating equation (2.30) and combining with equation (2.32)

$$I\ddot{\theta}_p = \left(mg - ml\dot{\theta}_p^{\ 2} cos\theta_p - ml^2\ddot{\theta}_p sin\theta_p\right) lsin\theta_p - (F - M\ddot{x})lcos\theta_p \tag{2.33}$$

Substituting values from equation (2.32) in equation (2.33)

$$I\ddot{\theta}_p = mglsin\theta_p - ml^2\ddot{\theta}_p - m\ddot{x}lcos\theta_p \tag{2.34}$$

Denoting $a = \frac{1}{(m+M)}$ we can represent equation (2.32) as

$$\ddot{x} = -mal\ddot{\theta}_p cos\theta_p + mal\dot{\theta}_p^{\ 2} sin\theta_p + aF \tag{2.35}$$

Substituting equation (2.35) in equation (2.33)

$$\ddot{\theta}_p = \frac{mglsin\theta_p + m^2l^2a\dot{\theta}_p^{\ 2} sin\theta_p cos\theta_p - malcos\theta_p F}{I - m^2l^2acos^2\theta_P + ml^2} \tag{2.36}$$

Where,

$M = mass\ of\ the\ cart$

$m = mass\ of\ the\ \ pendulum$

$l = length\ of\ the\ pendulum$

$I = moment\ of\ inertia\ of\ the\ pendulum$

$x = cart\ position\ coordinate$

$\theta = pendulum\ angle\ measured\ from\ verticle$

2.2.2 State space modelling

Considering four state variables for the system x_1, x_2, x_3 and x_4 such that

$x_1 = \theta_P;$

$x_2 = \dot{\theta}_P;$

$x_3 = x;$

$x_4 = \dot{x};$

Then,

$$\dot{x_1} = x_2; \tag{2.37}$$

$$\dot{x_2} = \frac{g \sin x_1 - mlax_2^2 sinx_1 \cos x_1}{\frac{4l}{3} - mla \cos^2 x_1} - \frac{mla \cos x_1 F}{\frac{4ml^2}{3} - m^2 l^2 a \cos^2 x_1} \tag{2.38}$$

Substituting Equation (2.36) in Equation (2.35) we get

$$\ddot{x} = \frac{-mag \sin x_1 \cos x_1 + \frac{4mla}{3} x_2^2 \sin x_1 + \frac{4aF}{3}}{\frac{4}{3} - macos^2 x_1} \tag{2.39}$$

Then,

$$\dot{x_3} = x_4; \tag{2.40}$$

$$\dot{x_4} = \frac{-mag \sin x_1 \cos x_1 + \frac{4mla}{3} x_2^2 \sin x_1 + \frac{4aF}{3}}{\frac{4}{3} - macos^2 x_1} \tag{2.41}$$

The values used for simulation are-

$l = 0.3$ m;

$m = 0.2\ Kg;$

$M = 0.5\ Kg$

$I = 0.006\ Kg.m^2$

$g = 9.81$

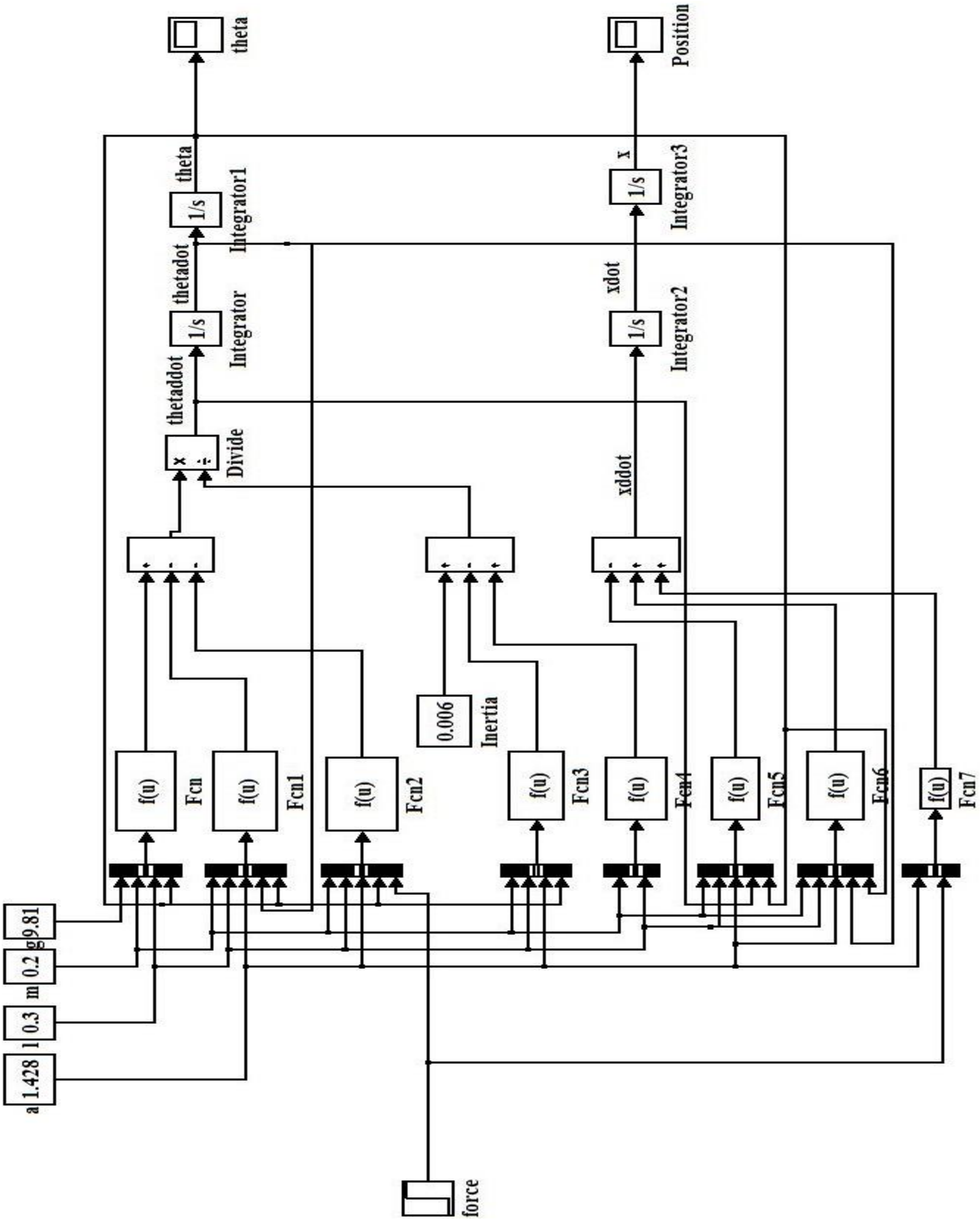

Figure 2.7 Simulink model of Inverted Pendulum

CHAPTER 3
DESIGN OF PID CONTROLLERS

PID controllers are the mostly widely used feedback controllers to enhance the performance of automatic control systems since it was developed by Nicholas Minorsky,a Russian born in 1885.It was his theoretical development for the application of automatic steering of ships that is called today as proportional plus integral plus derivative (PID) controller or simply three mode controller. It is interesting to note that more than 95% of controllers used in industries are either PID controllers or modified PID controllers [15]. This is because if mathematical model of a plant can be formulated then the parameters of PID controller can be derived with various tuning methods which satisfy the transient along with the steady state response of the system in a closed loop. However more it has been noted that more than 80% of the industrial PID controllers are not tuned properly this may be because of the complexity of the system or the high nonlinearity of the system which makes the tuning of the PID parameters cumbersome hence a lot schemes of PID tuning can be found proposed in the Research texts ranging from simple Ziegler Nicholas tuning method to adaptive tuning methods, which can be used for delicate and fine tuning of PID parameters.

3.1 CONVENTIONAL LINEAR PID CONTROLLER

A conventional linear PID controller is based on linear control theory. Here the PID controller compares the actual value of the plant input with the reference input (desired value), determines the deviation, and produces a control signal that will reduce the deviation to zero or a reduced value. The generated control signal c is the difference between the reference input and the output of the plant and the linear combination of integral, differential and proportional coefficients as shown;

$$c = K_p e + K_i \int e dt + K_d \frac{de}{dt}$$

$$e(t) = r(t) - y(t)$$

Where, c is the control signal to the system. $r(t)$ is the input (reference) and $y(t)$ is the system output, the difference between them yields the error e as shown in equation.K_p is the proportional gain, K_i is the integral gain and K_d is the derivative gain constant. Both the transient response as well as the steady state response is improved by the use PID controller. The integral term eliminate the offset error and helps in improving the steady state response by increasing the system type with additional pole at the origin. The transient response is improved by the derivative part, by adding an additional zero to the open loop transfer function [15]. The schematic of a linear PID controller is given

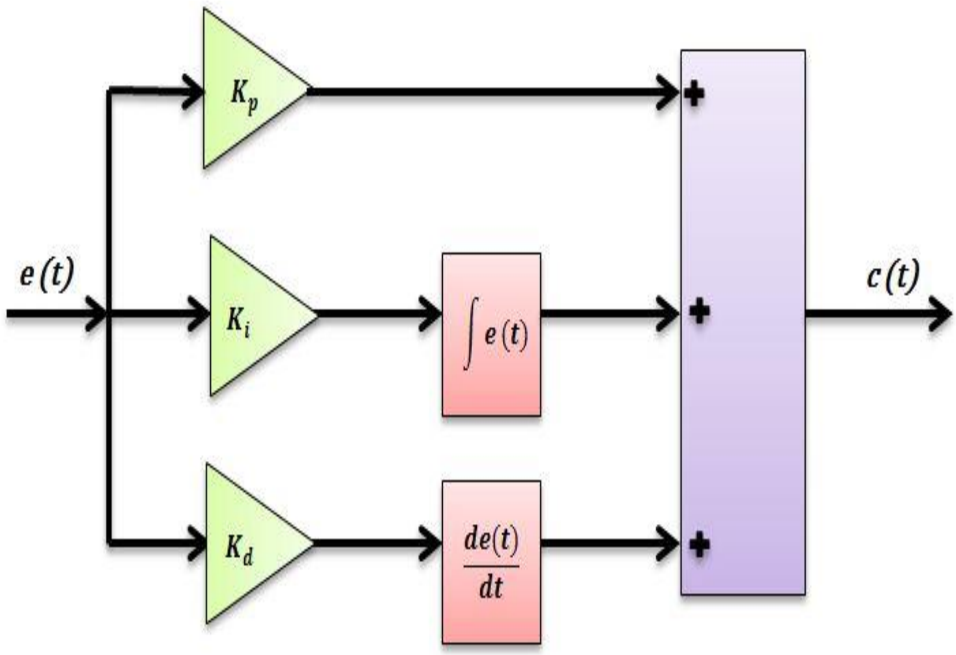

Figure 3.1 Conventional linear PID controller

3.2 NON-LINEAR PID CONTROLLER

The classical PID is based on linear theory and hence is effective only when used for simple practical linear processes. However when the system under consideration has more nonlinearity or the system parameters varies under a wide range, the performance of classical PID deteriorates considerably .This happens because the reference input given to the system are usually not continuous or smooth as they are subjected to the disturbance or noise signal, while the output of the system is required to be smooth and continuous [16]. Since the smooth continuous output is taken as the direct target of the output the inertia effect of the system is neglected which causes unanticipated overshoot oscillations when used practically.in addition to that the reference signals are usually un-differentiable signals which makes it difficult to obtain the differential signal of the error .The linear combination of traditional PID causes the conflict between the overshoot variable and the high speed. Initially a high value of K_p would increase the systems response but as the error starts reducing the K_p value must also change automatically so that there is no overshoot. Also when the error is decreasing and the rate of change in error is increasing the K_p must decrease gradually to reduce the overshoot and when the change of error is decreasing the K_p must gradually decrease to avoid the overshoot. But due to the linear combination of the PID this is not possible. Hence we need to find different nonlinear controllers to control such plants without nonlinearities. Here the non-linear PID controller comes into picture. The major failure of linear PID controller comes due to the constraints in the mathematical model i.e. Contradiction between the overshoot (increased gain) and the speed of response of the system. If a suitable law can be formulated for the system these limitations would be removed, thereby leading to a better control action for the system. For this a nonlinear module can be introduced as proposed in [13]

$$f(e,\alpha,\delta)=\begin{cases} |e|^{\alpha} sign(e) & |\gamma| \geq \delta \\ \dfrac{e}{\delta^{1-\alpha}} & |\gamma| < \delta \end{cases}$$

Where,

$$sign(e)=\begin{cases} 1, & e \geq 0 \\ -1, & e < 0 \end{cases}$$

e is the error signal ,δ describes the linear range of the function f.here function f can accommodate a greater range of nonlinear characteristics which is determined by α. Hence now control signal being generated by the PID controller takes the form [17]

$$c = K_p e f_p(e_p, \alpha_p, \delta_p) + K_i f_i(e_i, \alpha_i, \delta_i) \int e dt + K_d f_d(e_d, \alpha_d, \delta_d) \frac{de}{dt}$$

Where,

K_p, K_i and K_d are proportional, integral and derivative gains respectively for the system

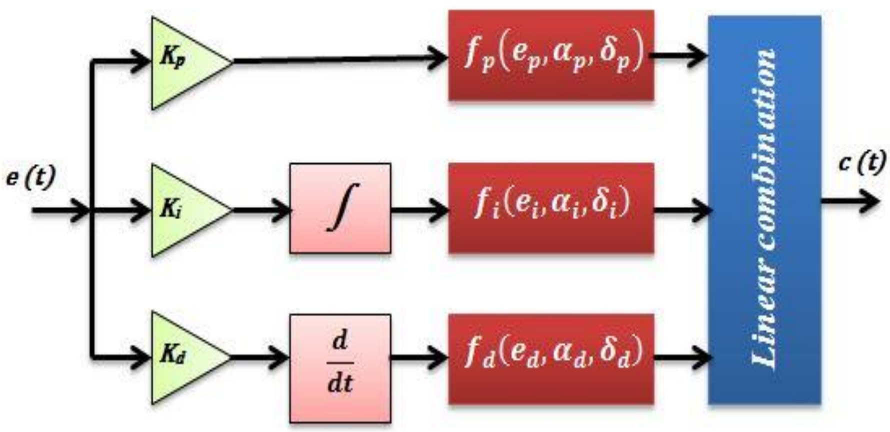

Figure 3.2 Nonlinear PID controller

The function $f(\alpha,\delta,e)$ denotes the rate of error feedback, to compensate the nonlinearity of the systems considered the value of α_p is taken in the range of $\alpha_p \epsilon [0,1]$ since we need to have lower gain when the error is high and vice versa [16]. The integral saturation problem of the integral term can be rectified by using α_i in the range of $\alpha_i \epsilon [-1,0]$. The value of differential term is chosen as $\alpha_d > 1$ so that when the steady state is reached the effect of the differential term is reduced. The values of $\delta_p, \delta_i\ and\ \delta_d$ are set by trial and error method comparing the simulation results.

3.3 METAHEURISTIC ALGORITHM TUNED PID CONTROLLERS

Metaheuristic algorithm are different from the traditional optimization algorithm by the fact that the later uses only a single point solution while an metaheuristic algorithm uses a sizeable population for finding an optimal solution. EA's can simultaneously evaluate a number of points scattered in the search space at the same time as they are inherently parallel in operation which enables them to consider as many points in the search space while not taking too much time to converge. Also they are more likely to converge to global optima rather than local optima. During an iteration of metaheuristic algorithm there is competitive selection of the better resulting individuals from the search space which gives better solution and the rest of the solutions are discarded off. Further some metaheuristic algorithms also reproduce or generate new points or offspring to improve the search space as well as solution. Off late several metaheuristic algorithms have been developed like Ant colony algorithm, firefly algorithm, Genetic algorithm, particle swarm optimizer algorithm etc. These algorithms differ from each other on the basis of variables to be tuned, offspring generation, and replacement mechanism [18]. Off all the metaheuristic algorithms used the most used ones are GA and PSO. They have been considered here for the tuning of PID controller for nonlinear systems because when using EA the restriction to the Eigen values of the nonlinear system matrix does not apply.

3.3.1 PSO algorithm

PSO is used for optimizing a wide range of objective functions because it is an extremely easy algorithm to work with. It is a metaheuristic algorithm proposed by Kennedy and Eberhart in the year 1995 which uses multiple agents (particles) to obtain optimal solutions and hence also called agent based algorithm. The algorithm is essentially an bio inspired algorithm i.e. the algorithm has been inspired by the flocking of birds and schooling of fishes, here a group of individuals (called particles) [6] together in a population or swarm try to search for optimal solution in a stochastic manner. The concept of social interaction by exchanging the experience of each other is introduced here which is governed by three factors –collision avoidance, velocity matching and centering of the flock It has been seen that the particle goes for faster convergence to local or global optima over small number of iterations than other EA's hence it has become a widely used algorithm to solve engineering problems. The algorithm starts with initialization of points in the search space [11]; each particle denotes a possible solution for the optimization of objective function. During each iteration all the particles in the search space discover a probable solution. After this the particle updates its position according to the velocity vector which also depends on its previous velocity and is decided taking in account the past local and global best solutions .now the best solutions are kept and particle moves towards local best solution attained by its fellow particles but also the global best. Hence if a particle has discovered a new best solution then all other particles would try to move toward it. The four important terms in PSO for the particles in the swarm are p_i (position), v_i (current velocity), Pb_i (local best position), Gb_j (global best position).each particle is updated according to the above four features in each iteration, assuming an cost function J to be minimized. The equation for new velocity of a particle is given by

$$v_{i,j}(n+1) = C \times \{wv_{i,j}(n) + c_1r_1[Pb_{i,j}(n) - p_{i,j}(n)] + c_2r_2[Gb_j(\text{n}) - p_{i,j}(n)]\}$$

Where, n is the number of iterations, w is inertia weight, c_1 and c_2 are the acceleration coefficients (called cognitive and social component respectively) and $r_1$and r_2 are two uniform random numbers between (0, 1). $v_{i,j}$ is the velocity of the j^{th} dimension of the i^{th} particle. The new position of the particle is updated by the equation

$$p_{i,j}(n+1) = p_{i,j}(n) + v_{i,j}(n+1)$$

The local best of every particle is updated according to the equation

$$Pb(n+1) = \begin{cases} Pb(n), & J(p(n+1)) \geq J(Pb(n)) \\ Pb(n+1), & otherwise \end{cases}$$

if the population size is denoted by s, then the global best found in the optimization is given by

$$Gb(n+1) = \min_{Pb} J\big(Pb_i(n+1)\big), 1 \leq i \leq s)$$

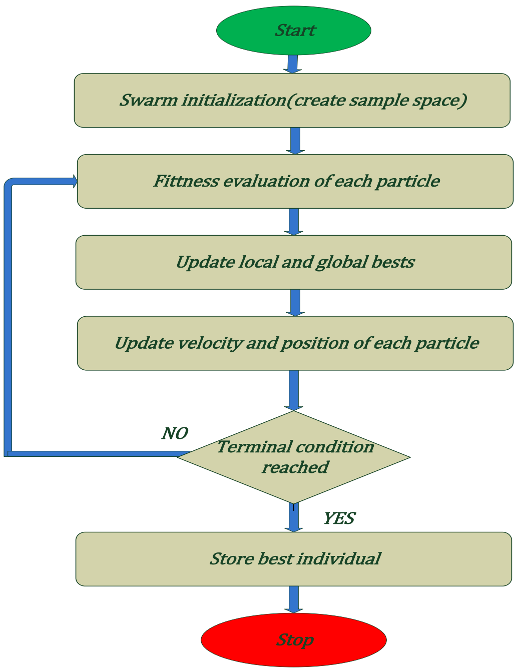

Figure 3.3 Flowchart for PSO algorithm

3.3.2 Tuning of PID with PSO

Tuning of PID can be efficiently done by using PSO algorithm based optimization technique. Particle swarm optimizer can be used to find the value of optimal PID parameters which will minimize the objective function (J).for this the knowledge of the system being considered should be known.an approximately similar model of the physical system can be made from the mathematical equations describing the model as shown in section 2.the PID is said to be tuned when the objective function defined for the system is close to zero. Every time tuning of the controller is done the following steps are performed [19]:

i. Initialization of various parameters of PSO is done, like value of c_1 and c_2,population size, dimension of problem is set

ii. Initial population of set of particles (K_p, K_i and K_d) values are generated

iii. Each individual is tested for fitness by evaluating the objective function

iv. The local best and global best values are stored

v. Iteration is started, with every iteration the velocity and position updation takes place for every particle along with of local and global best updation.

vi. Iteration stops when maximum value is reached and the value of (K_p, K_i and K_d) that results in the best performance is obtained

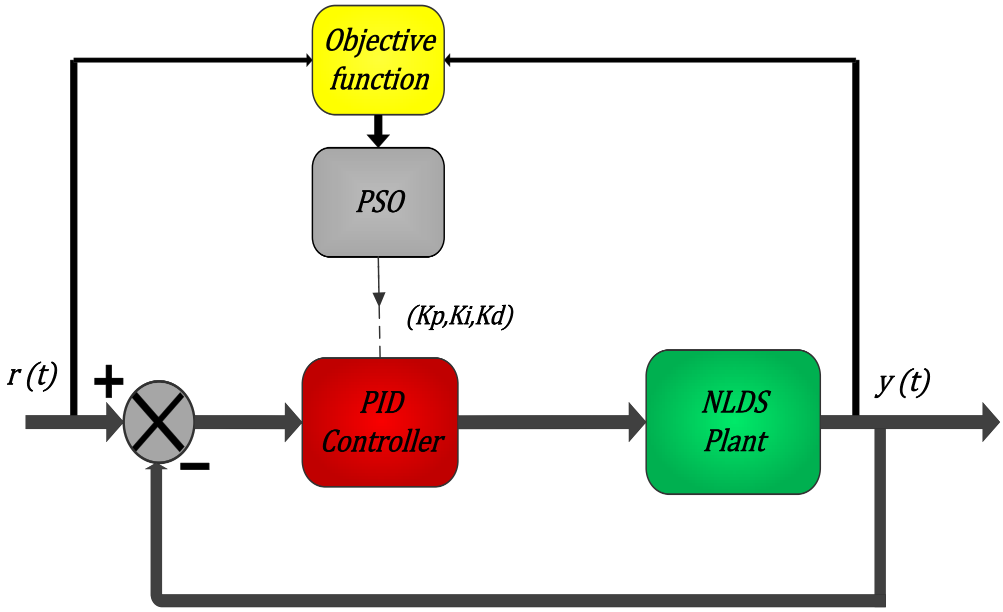

Figure 3.4 Block diagram of PSO-tuned PID controller

The various parameters used in the Particle swarm optimizer process for finding PID parameters for various systems are

Size of the swarm = 10

Dimension of the problem = 3

Maximum iteration =50

$Cognitive\ factor\ c_1$ = 2.05

$Social\ factor\ c_2$ = 2.05

Constriction factor C = 2

3.3.3 GA algorithm

Genetic algorithm is a metaheuristic algorithm found on stochastic global search method which was introduced in the US in the year 1970 by John Holland. The modifications in GA since have made it a much sought after algorithm for some type of optimization problems. This algorithm can be used for both single objective as well as multi objective functions .GA is based on natural selection hence it has no clue in the beginning what the correct answer for a given optimization problem is, it depends entirely on the environments response as well as the operators used repeatedly on the initial. The GA selects individuals from the population randomly and uses operations like crossover, mutation to create a new set population. Over the successive generations the best among the lot are selected, operators are worked upon on them and evaluated again until they evolve to the global optimal solution. Various processes involved in GA algorithm have been explained here [20].

i. Fitness evaluation and selection

All the individuals in the population are evaluated and then arranged in the increasing order of objective function value. Then portion of the population was selected having the desired performance based on whether minimization or maximization of the objective function was needed. Selection is the procedure to obtain parent chromosome or individuals needed to create a new offspring. There are many other types of selection procedures also like tournament selection, roulette selection etc. that can be used to create a new set of population

ii. Reproduction

Individuals having good performance are mated together to produce new generations, thus creating a new pool of generation having the same population size as before.

iii. Crossover

Crossover operator is applied on the new population which exchanges the information between any two parent individual selected earlier for reproduction. The number of times the cross over operator is used depends upon the probability of crossover as well as the population size.

iv. Mutation

The mutation operator randomly selects a particle from the population and alters the particle’s value. This operators is used in a very low probability as it may spoil the results drastically, so it is used only as complementary to crossover and reproduction genetic algorithm only requires the objective function to start its work, they do not need any derivative functions as such, hence they can be used for solving multiple, nonlinear or knowledge based problems very easily. They exploit probabilistic transition rules not deterministic ones hence they can converge to the optimal solution efficiently [2].

Hence the robust and simple structure of GA makes it suitable for complex optimization problems. A simple flowchart of Genetic Algorithm is given.

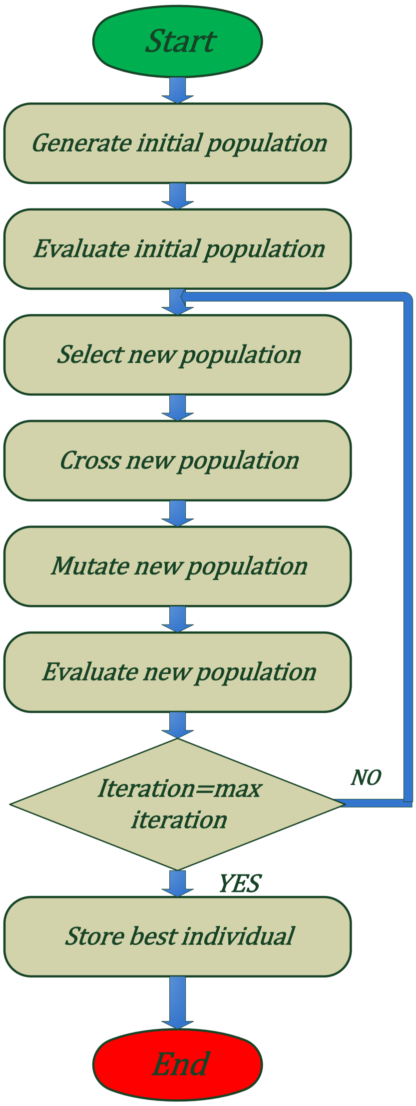

Figure 3.5 Flowchart for GA algorithm

3.3.4 Tuning of PID with GA

The objective of using Genetic algorithm here is to determine the optimal value of PID parameters (K_p, K_i and K_d) such that the objective functions (J) is minimized. The overall result of the finding of the optimal parameters would lead to fine tuning of the PID controller and simultaneously improvement in the transient as well as steady state response of the system under consideration. The objective function (J) becomes an instrument to evaluate the performance of PID controller with the determined value of gain parameters, resulting in an optimized controller with the best individual (parameter value). The involved in implementing GA for PID tuning is as follows [21]:

i. Creating an initial random population of individuals of a fixed size 's' i.e. set of K_p, K_i and K_d values are created of size's'. Thus the dimension of the problem is 3.
ii. Evaluating each set of individual for fitness
iii. Selecting the individuals showing the best fitness value
iv. Reproducing using one of the probabilistic methods
v. Performing crossover operation on the reproduced individuals
vi. Executing mutation operation with low probability
vii. Repeating step two till predefined conditions are met
viii. Take the best individual from the population obtained at the end

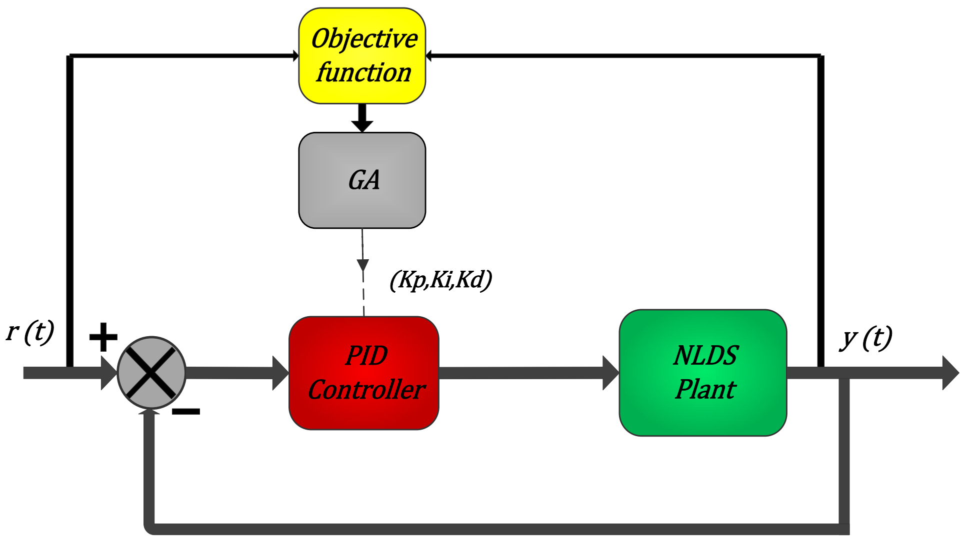

Figure 3.6 Block diagram of GA-tuned PID controller

The various parameters used in the Genetic Algorithm process for finding PID parameters for various systems are

Size of the population = 10

Dimension of the problem = 3

Mutation Rate = 0.2

Selection Rate = 0.5

Maximum Iteration = 100

3.4 Objective Function

The objective function used in the optimization algorithm of all the systems is given as

$$J = (r(n) - y(n))^2 + (max[r] - max[y])^2$$

Where,

$r(n)$ = n^{th} input of the system

$y(n)$ = n^{th} output of the system

CHAPTER 4
BENCHMARK FUNCTION EVALUATION

For testing the algorithms 4 benchmark functions were used namely Rosenbrock function, Camel hump function and Goldstein function [22]

4.1 ROSENBROCK FUNCTION

Rosenbrock function also called banana or valley function is a very popular problem for optimization algorithm.it is a unimodal function and its global minimum lies in a narrow parabolic valley. Even though the global minimum is easy to find the convergence to the minimum is actually difficult. The function is given as

$$f(x) = \sum_{i=1}^{d-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$$

The function is evaluated on the hypercube $x_i \in [-2.048, 2.048]$, for all i=1…d

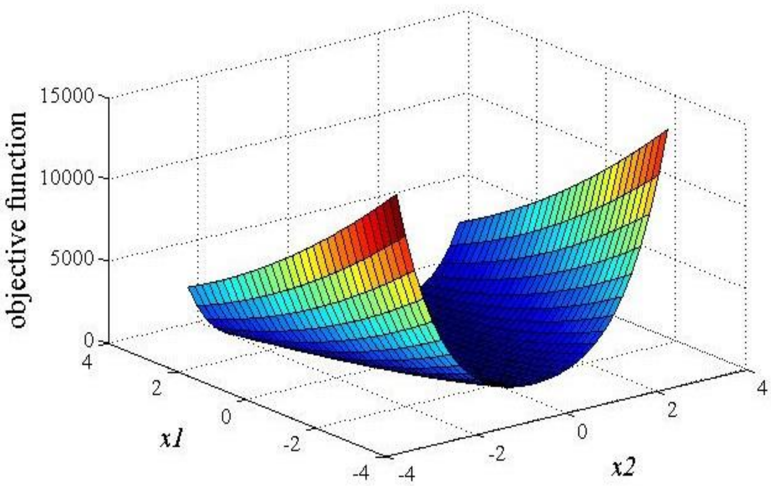

Figure 4.1 Rosenbrock function

4.2 SIX HUMP CAMEL BACK FUNCTION

Six Hump camel back function is a multimodal 2-dimensional benchmark function which has six minimum points out of which two are the global minima. It is given by

$$f(x_1, x_2) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right).x_1^2 + x_1.x_2 + (-4 + 4.x_2^2).x_2^2$$

The functions are evaluated in the region

$$-3.0 \leq x_1 \leq 3.0$$

$$-2.0 \leq x_2 \leq 2.0$$

The global minimum is found at $(x_1, x_2) = (0.0898, -0.7126), (-0.0898, 0.7126)$, and the value of global minima is -1.03164

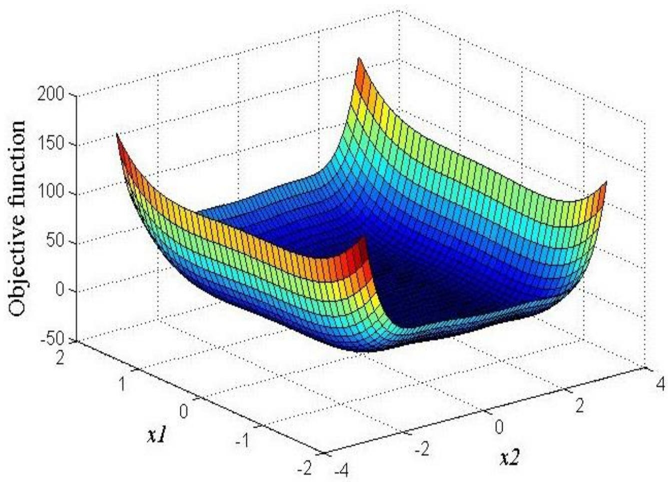

Figure 4.2 Six Hump Camel Back function

4.3 GOLDSTEIN-PRICE'S FUNCTION

The Goldstein price function is a continuous and slightly multi modal function having two variables. The variables are evaluated inside the bounds $-2 \leq x_i \leq 2$.the fuction is represented by

$$f(x_1, x_2) = \left(1 + (x_1 + x_2 + 1)^2 \times (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right)$$
$$\times \left(30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)\right)$$

The global minimum is at $(x_1, x_2) = (0, -1)$ with a value of f=3

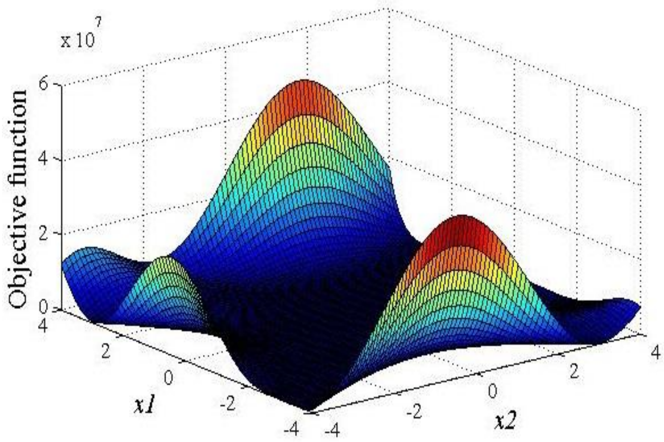

Figure 4.3 Goldstein-Price's function

4.4 RASTRIGIN FUNCTION

Rastrigin function is a highly multimodal function having several local minimum points. However, all the minimum points are distributed evenly. The function is given by

$$f(x) = 10d + \sum_{i=1}^{d} [x_i^2 - 10\cos(2\pi x_i)]$$

The function is evaluated in the region $x_i \in [-5.12, 5.12]$ for all i=1,2…d

The global minimum is at $x = (0,0, \dots .0)$ with value of objective function f=0

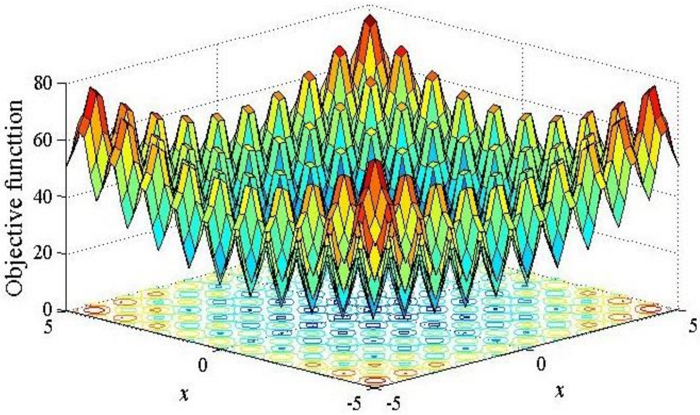

Figure 4.4 Rastrigin function

Table 4.1 Best solutions found for various Benchmark systems with metaheuristic algorithms

Benchmark functions / Values		Rosenbrock	Six Hump Camel Back	Goldstein-Price	Rastrigin
x_1		1	-0.0898	0	0
x_2		1	0.7126	-1	-
Global minimum(f)		0	-1.0316	3	0
GA algorithm	x_1	1.0410	-0.0895	$1.1657e^{-04}$	$8.388e^{-09}$
	x_2	1.0839	0.7128	-1.0001	-
	f	0	-1.0316	3	0
PSO algorithm	x_1	1	-0.0898	$1.2072e^{-10}$	$9.960e^{-10}$
	x_2	1	0.7126	-1	-
	f	0	-1.0316	3	0

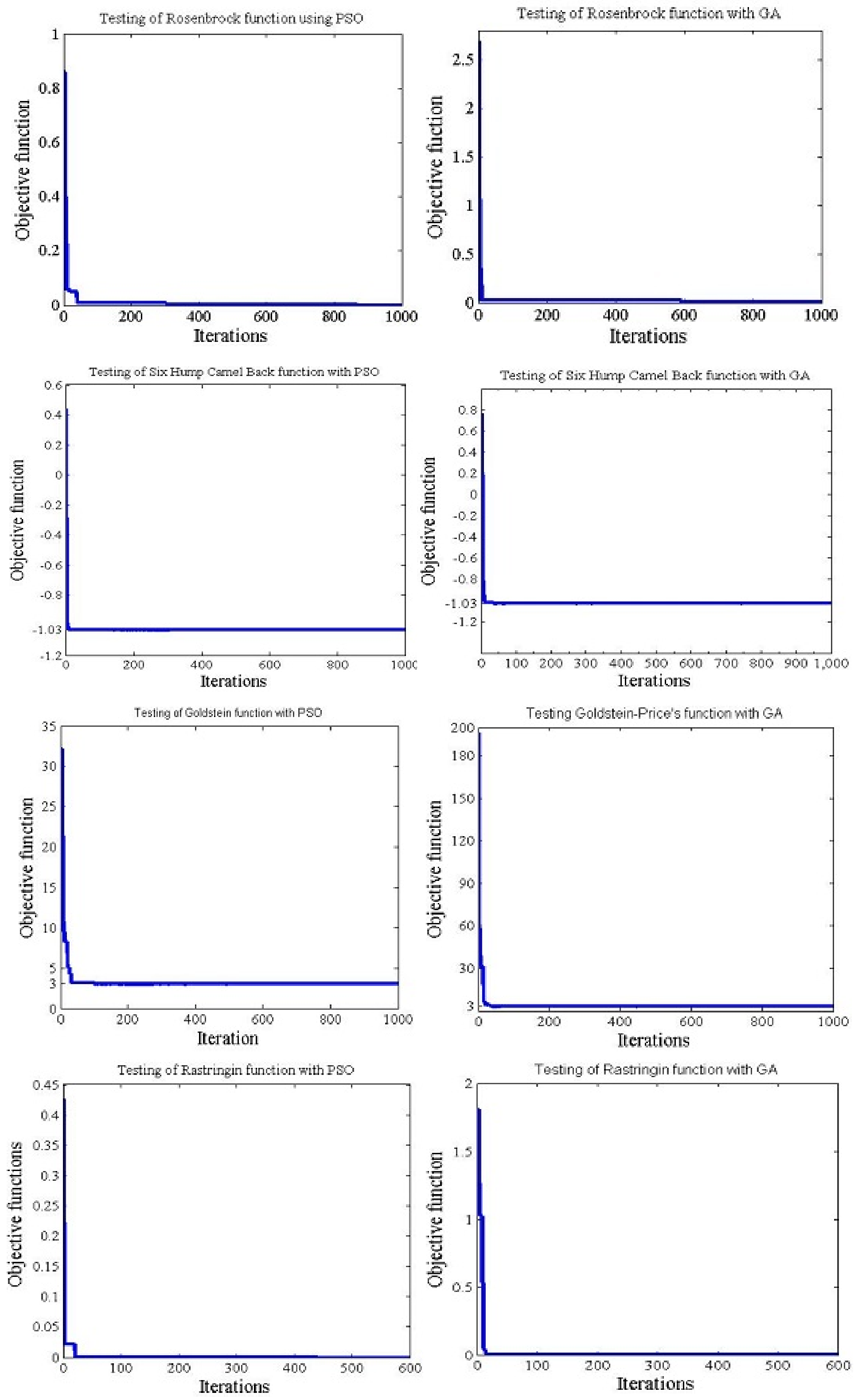

Figure 4.5 Performance Graph of Algorithms for various Benchmark function testing

CHAPTER 5
SIMULATION STUDIES OF NON-LINEAR SYSTEMS

The nonlinear dynamical systems modelled in section 2 were simulated in Simulink software and the following simulation results were obtained.

5.1 Simulation studies of Non-linear Mass-spring-Damper system

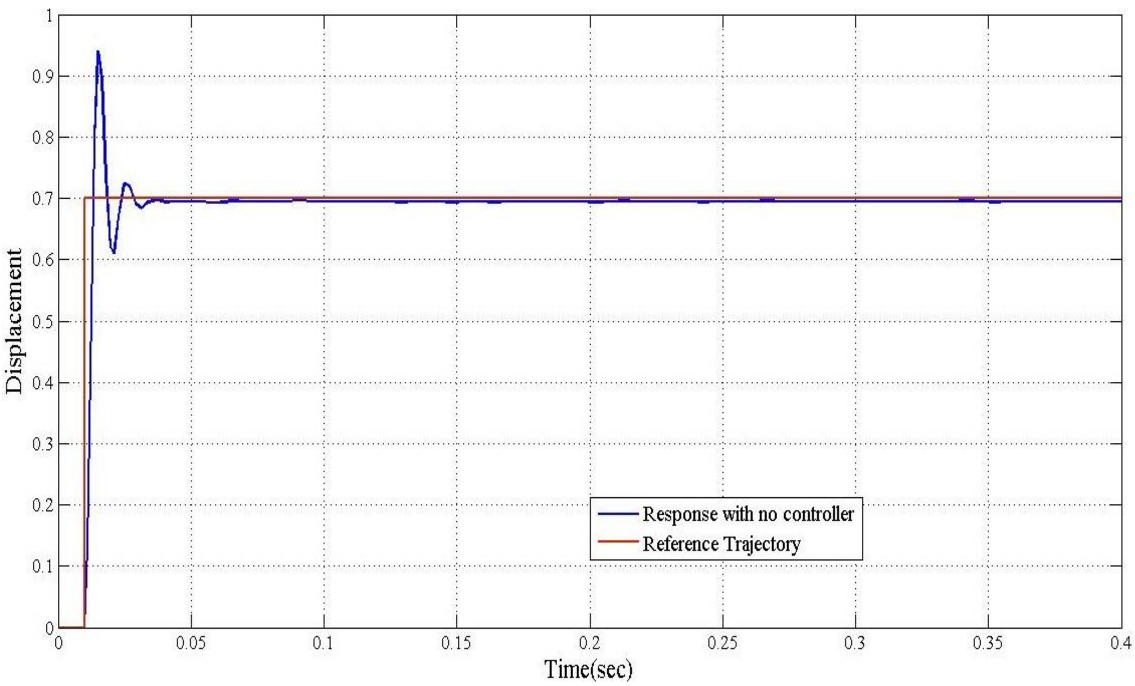

Figure 5.1 Response of Mass Spring Damper without controller action

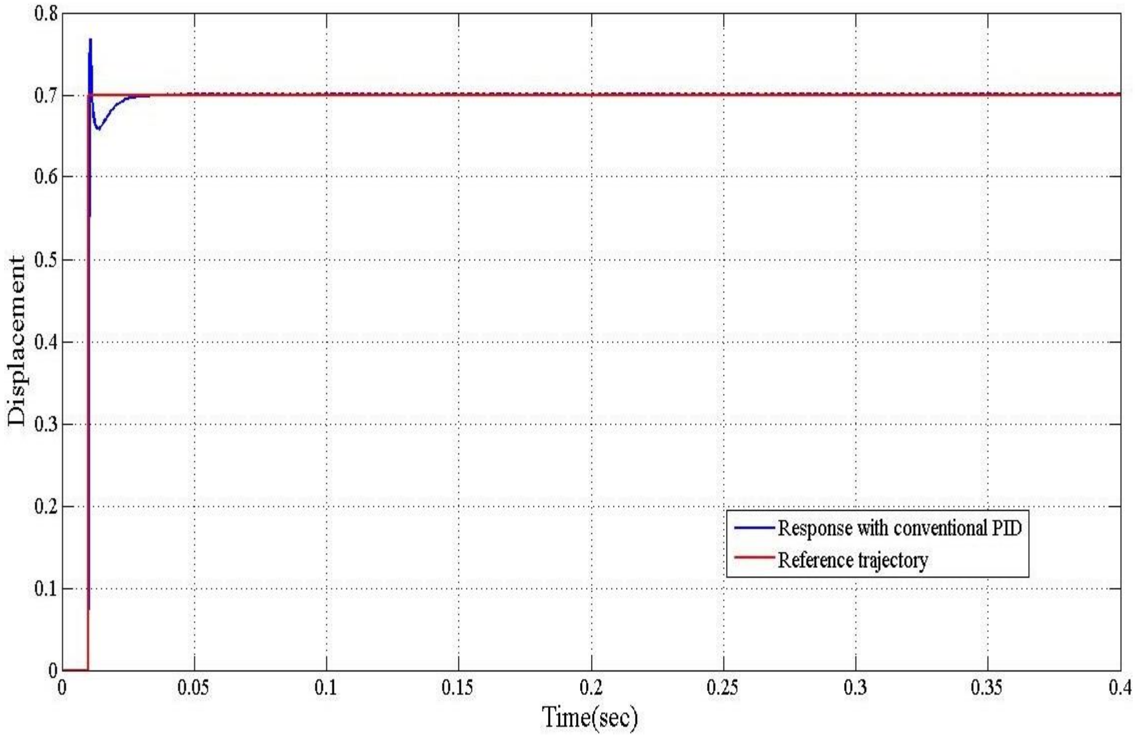

Figure 5.2 Response of Mass Spring Damper with conventional PID

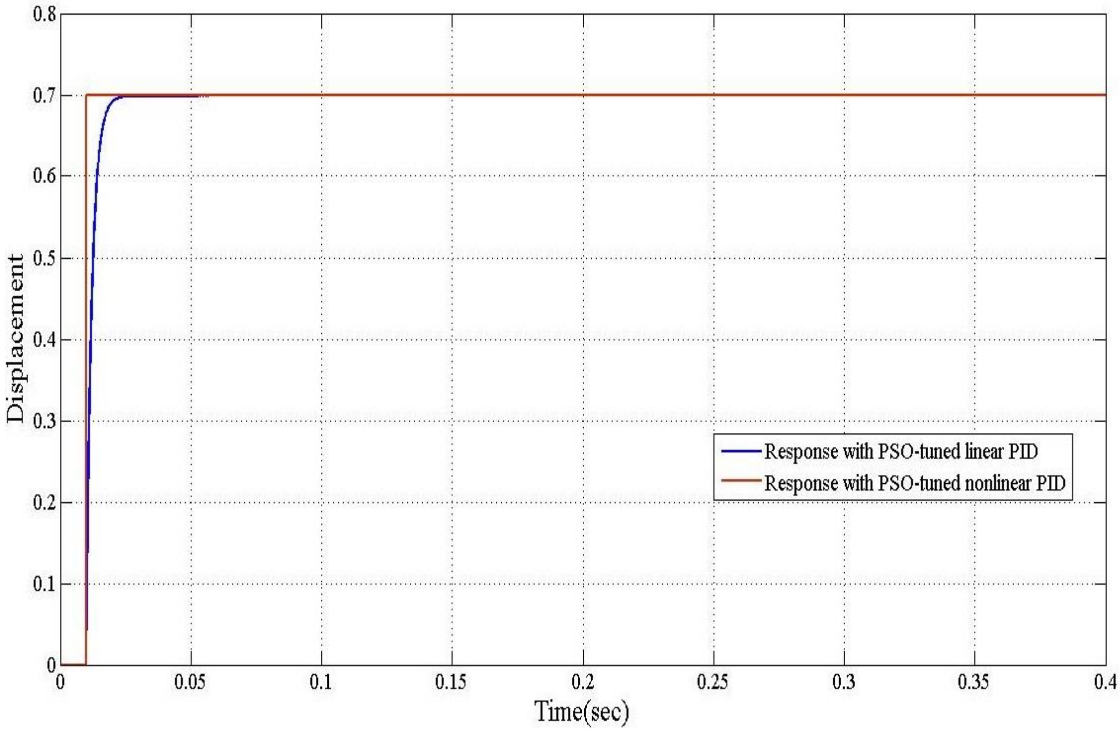

Figure 5.3 Response of Mass Spring Damper with linear PSO-PID

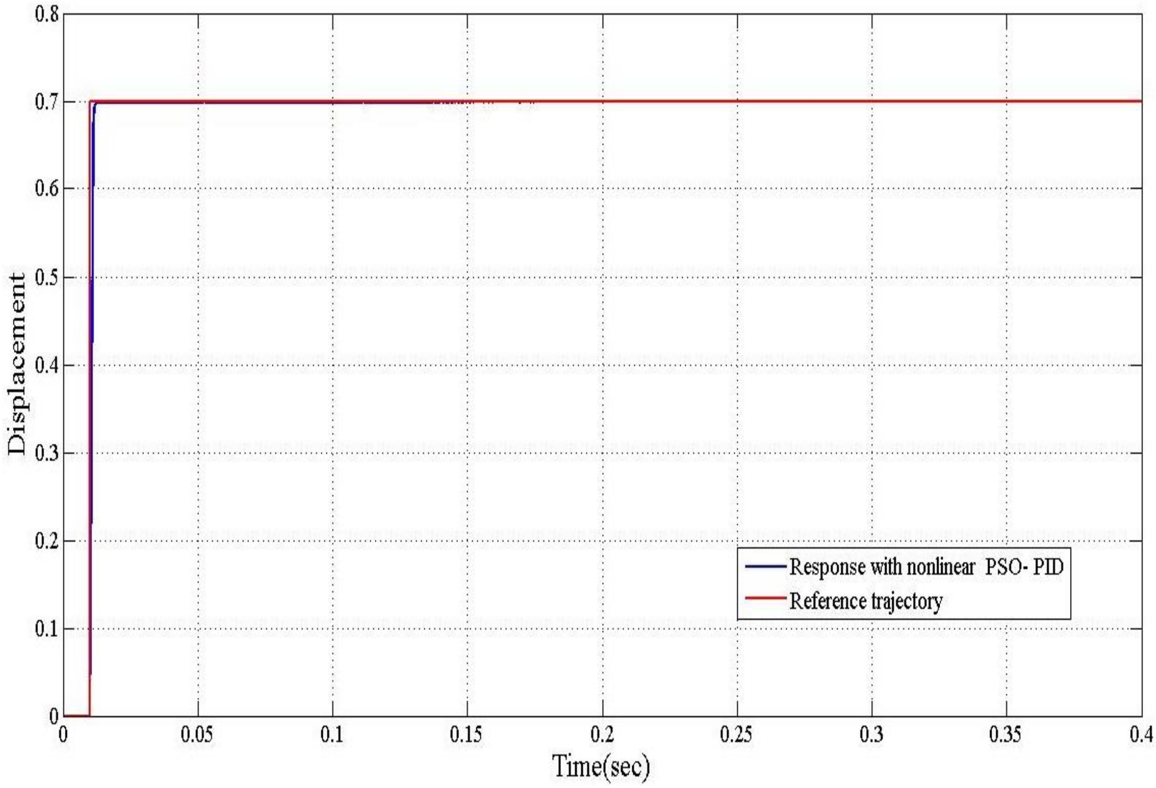

Figure 5.4 Response of Mass Spring Damper with nonlinear PSO- PID

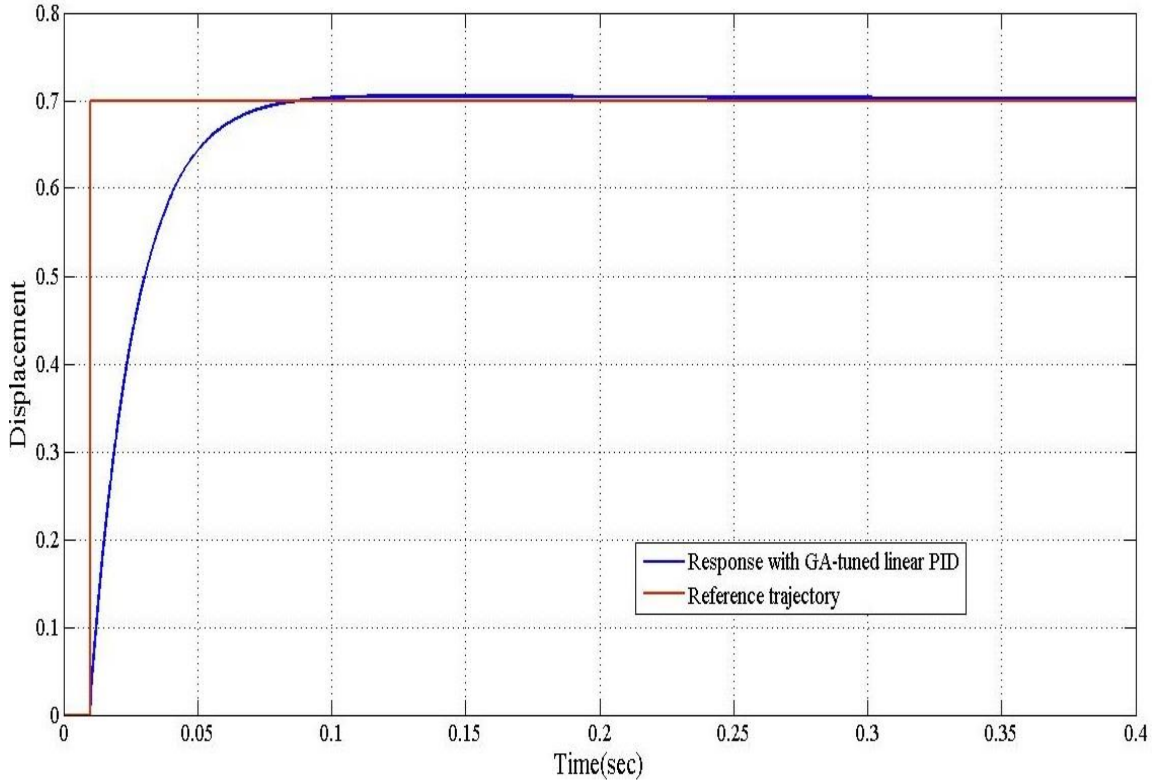

Figure 5.5 Response of Mass Spring Damper with linear GA-PID

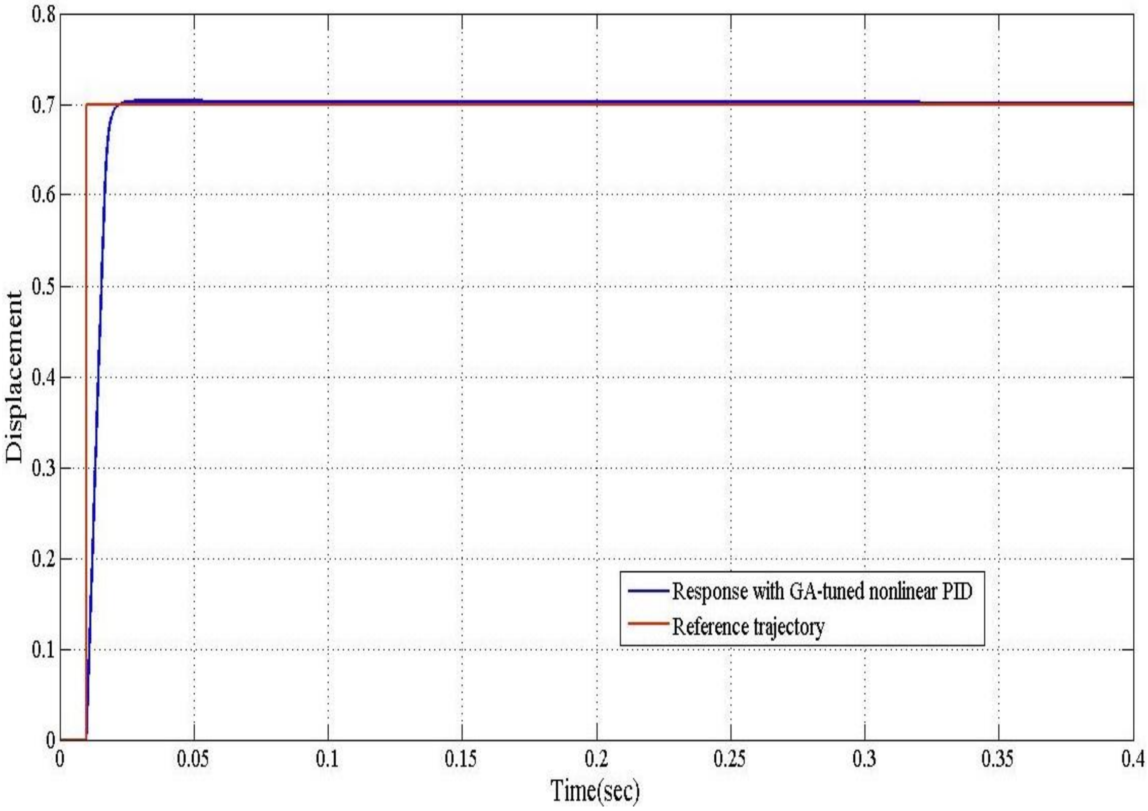

Figure 5.6 Response of Mass Spring Damper with nonlinear GA-PID

5.2 Simulation studies of PMBDC Motor with Robotic Manipulator

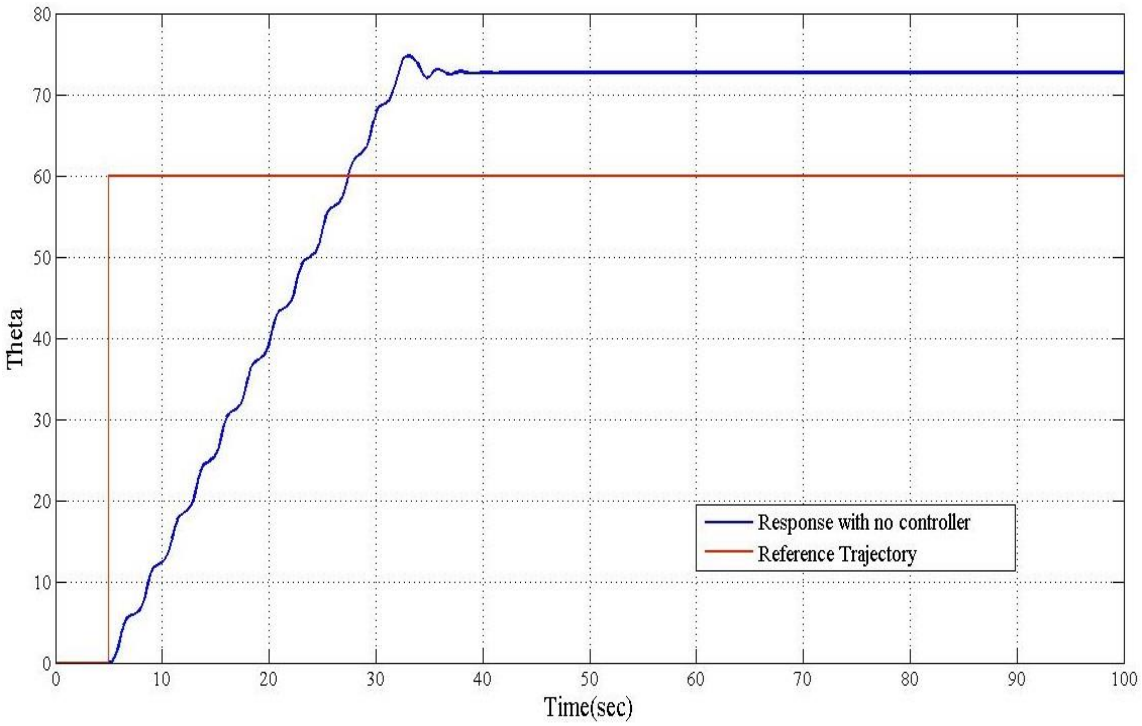

Figure 5.7 Response of PMBDC motor with robotic manipulator without controller action

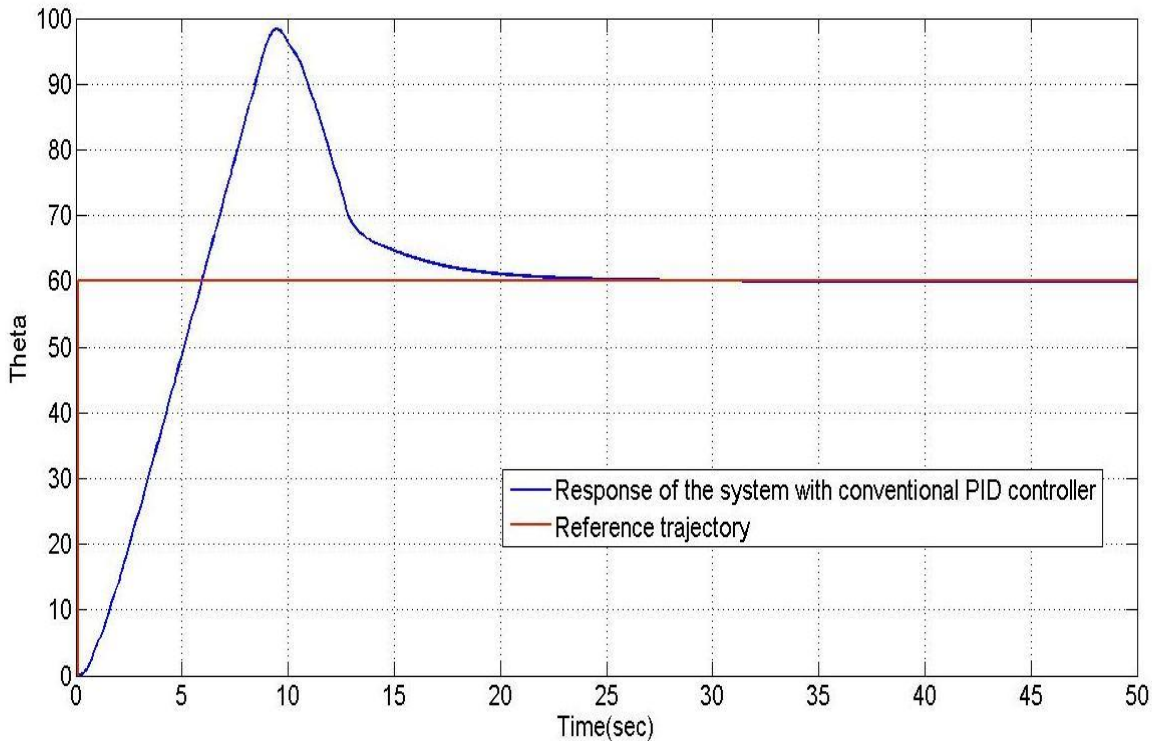

Figure 5.8 Response of PMBDC motor with robotic manipulator with conventional PID

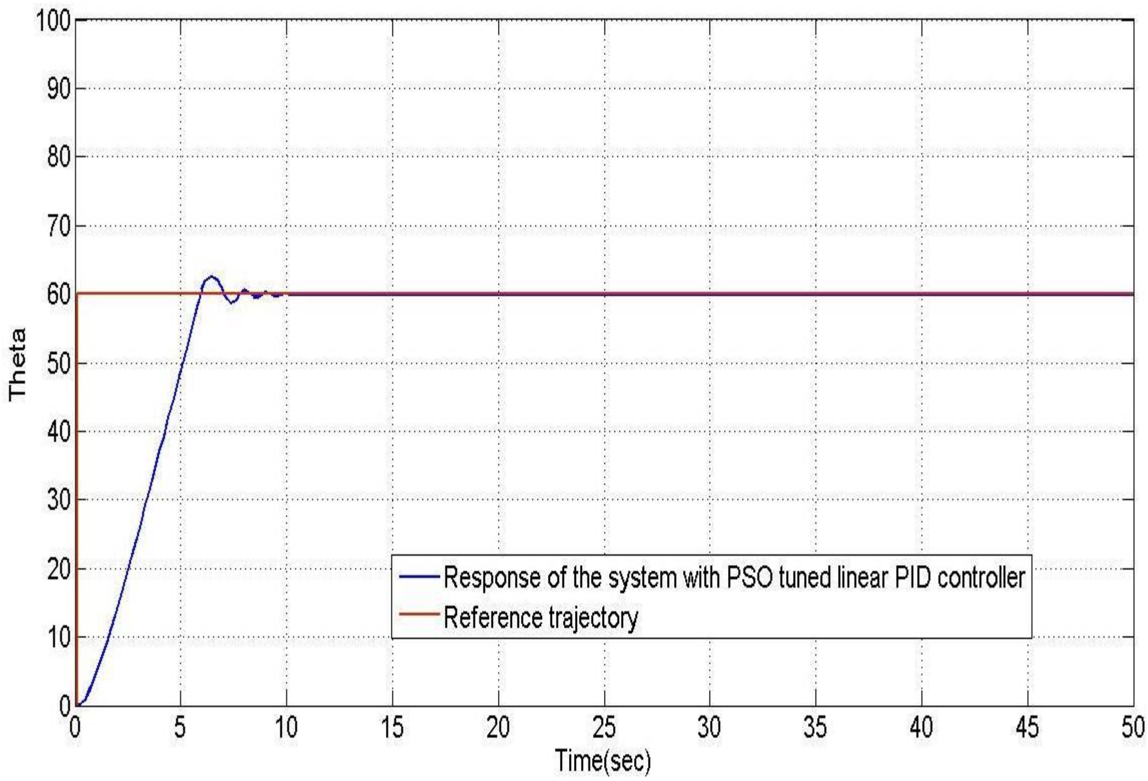

Figure 5.9 Response of PMBDC motor with robotic manipulator with linear PSO-PID

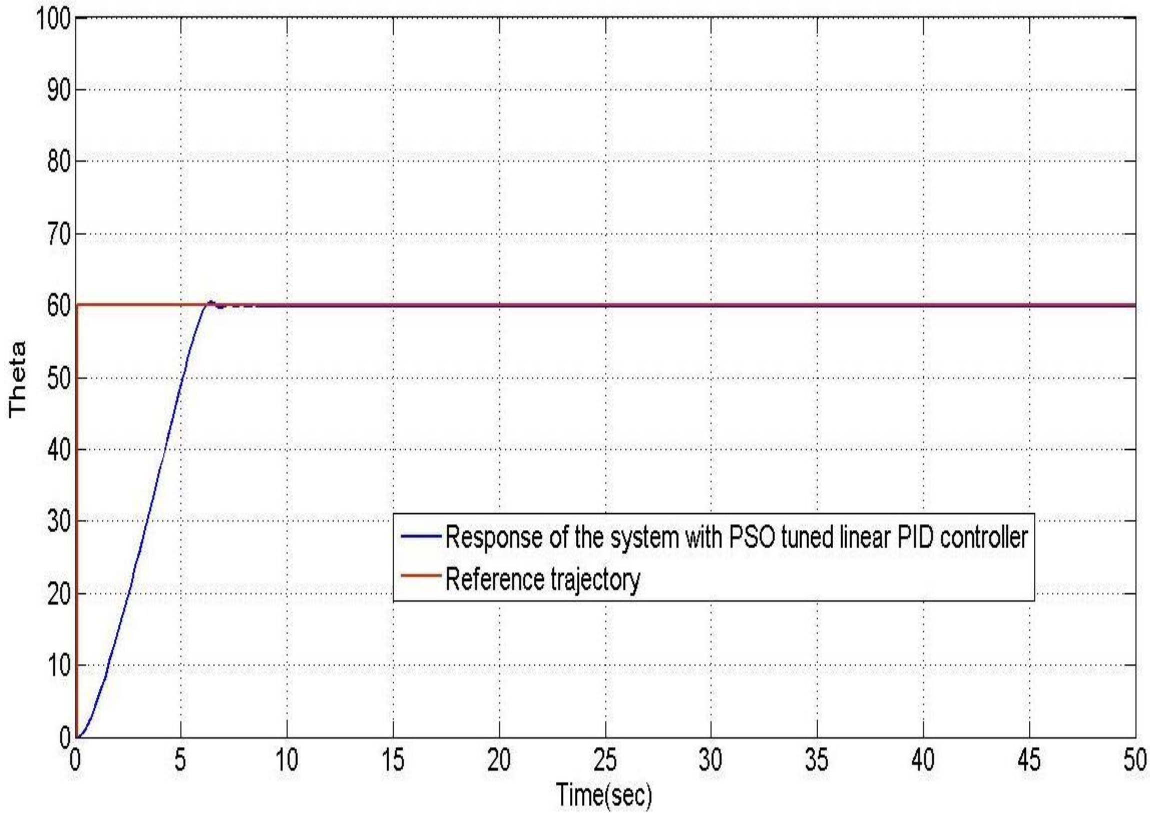

Figure 5.10 Response of PMBDC motor with robotic manipulator with nonlinear PSO-PID

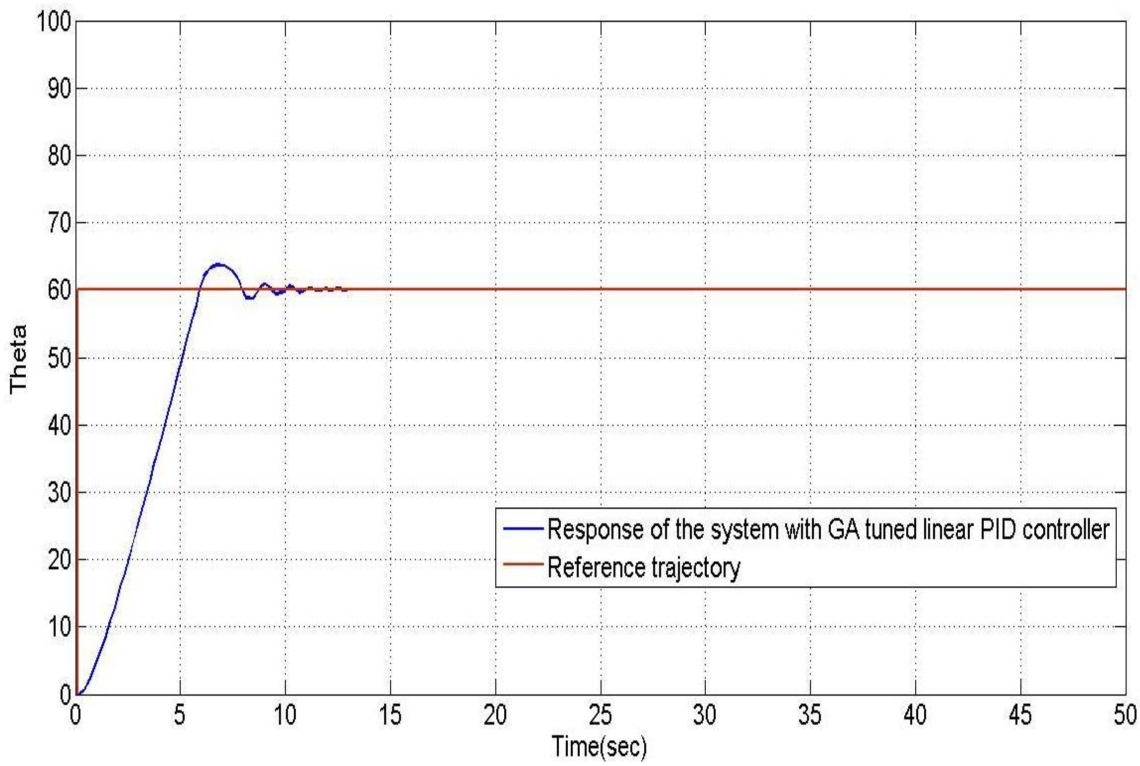

Figure 5.11 Response of PMBDC motor with robotic manipulator with linear GA-PID

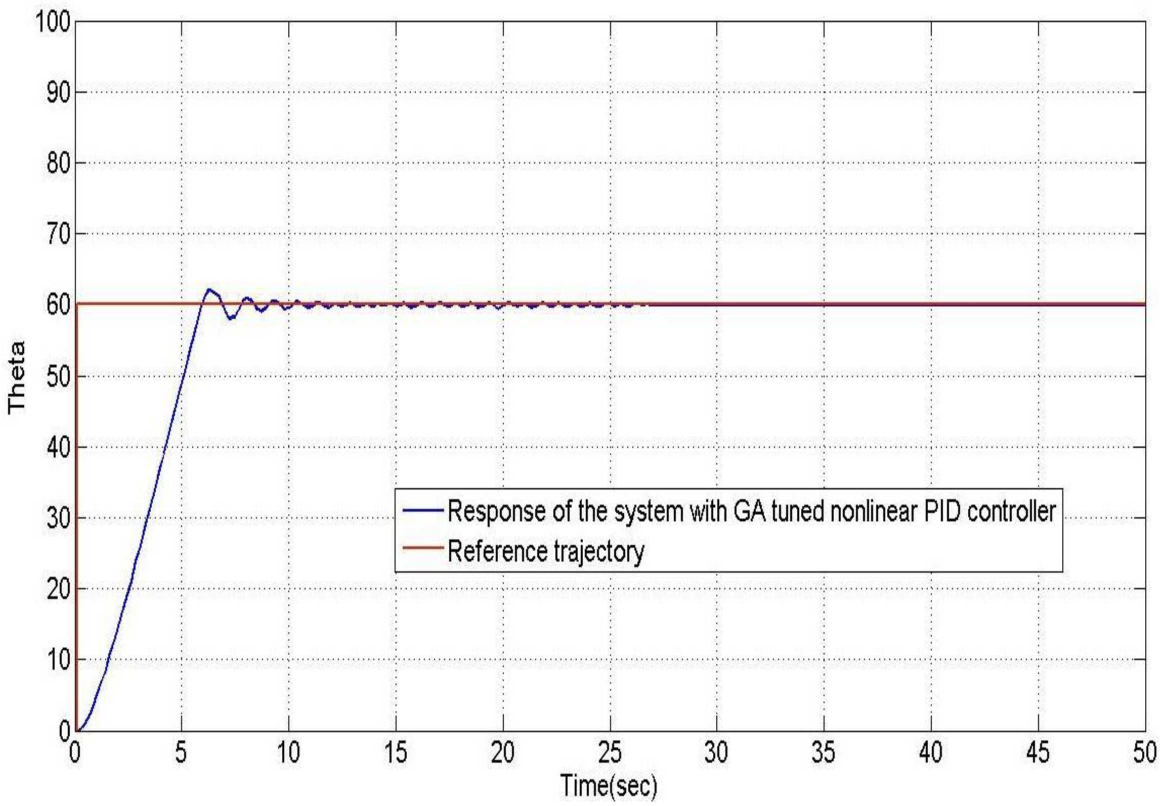

Figure 5.12 Response of PMBDC motor with robotic manipulator with nonlinear GA-PID

5.3 Simulation studies of Inverted Pendulum on progressing cart

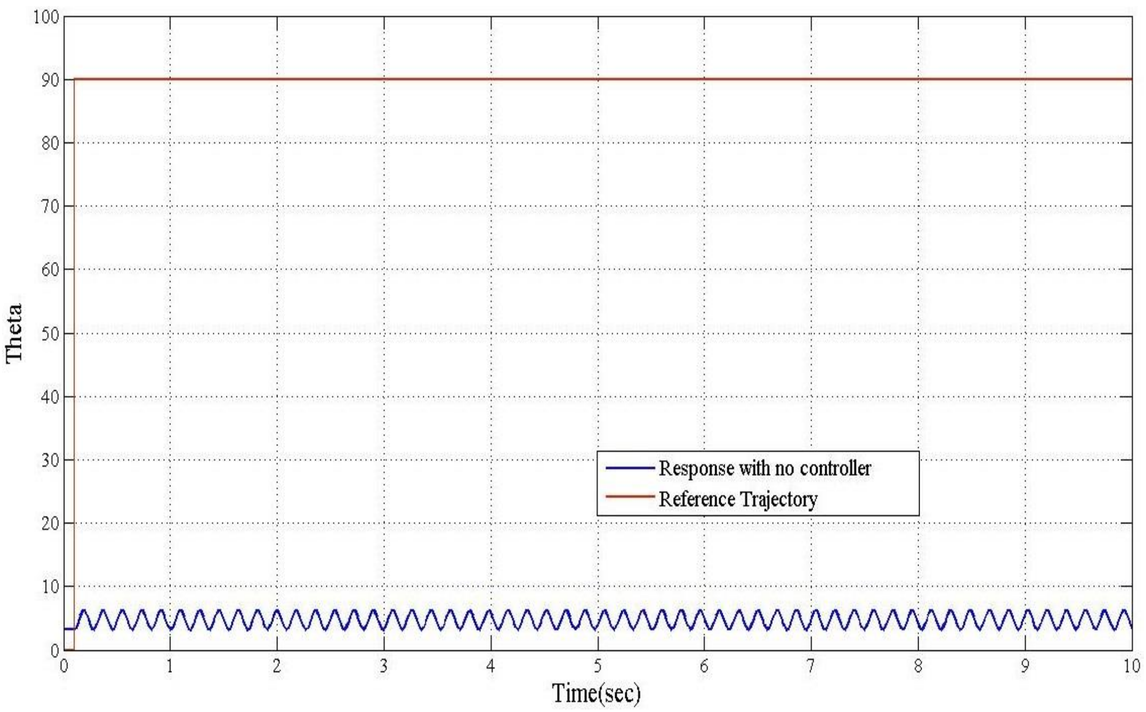

Figure 5.13 Response of Inverted Pendulum on a progressing cart without controller action

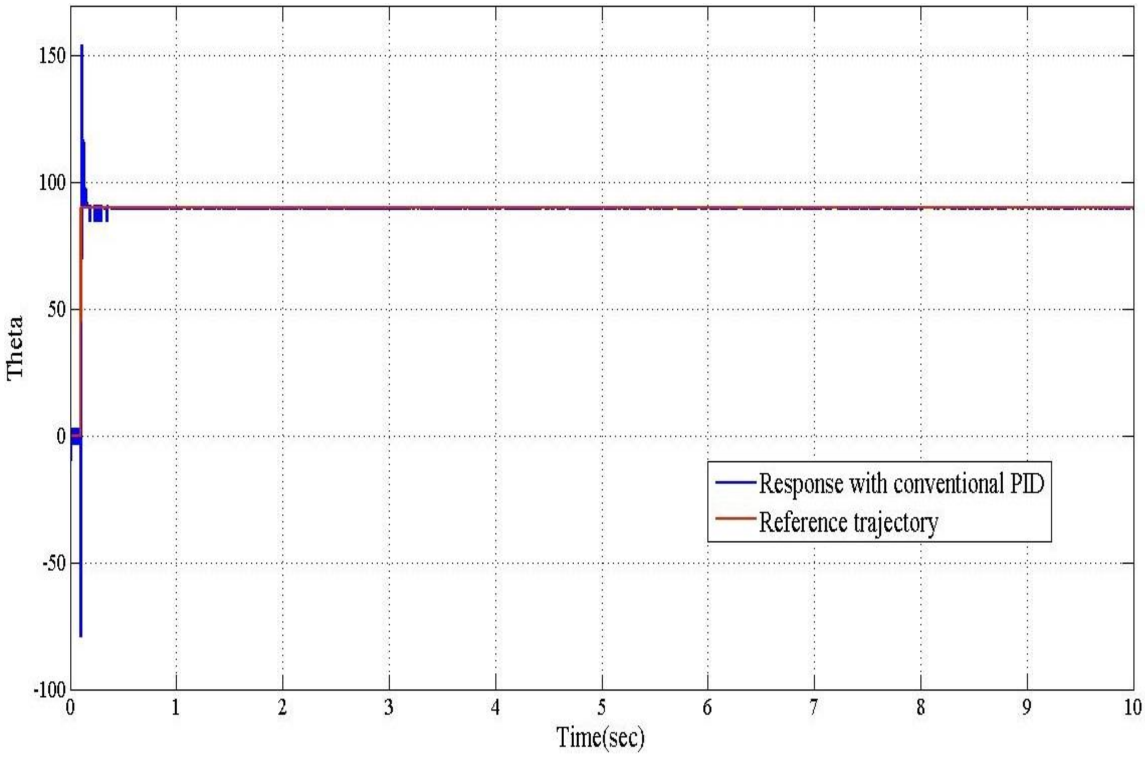

Figure 5.14 Response of Inverted Pendulum on a progressing cart with conventional PID

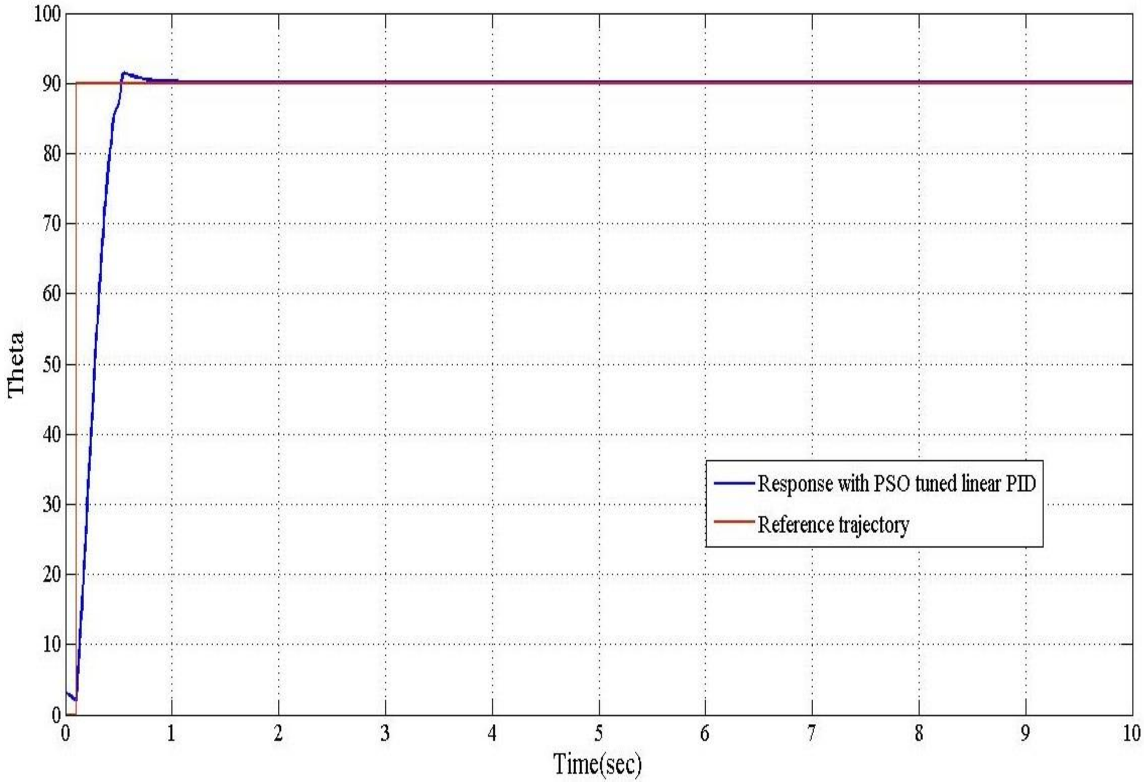

Figure 5.15 Response of Inverted Pendulum on a progressing cart with linear PSO –PID

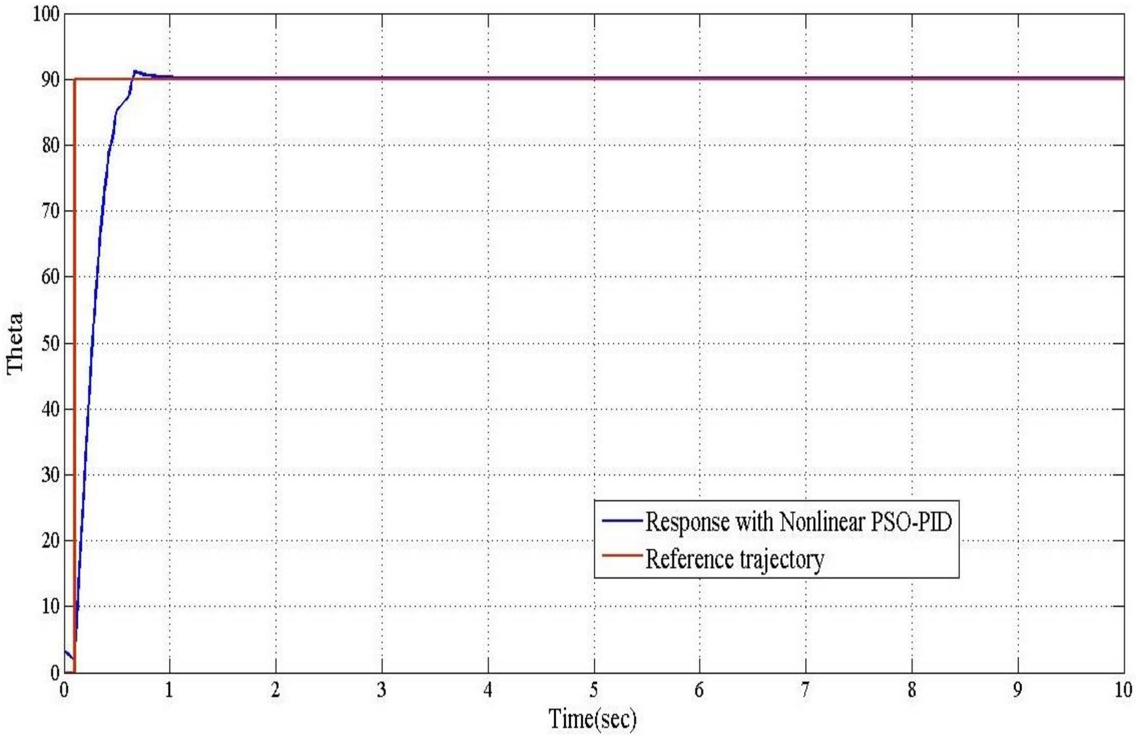

Figure 5.16 Response of Inverted Pendulum on a progressing cart with nonlinear PSO-PID

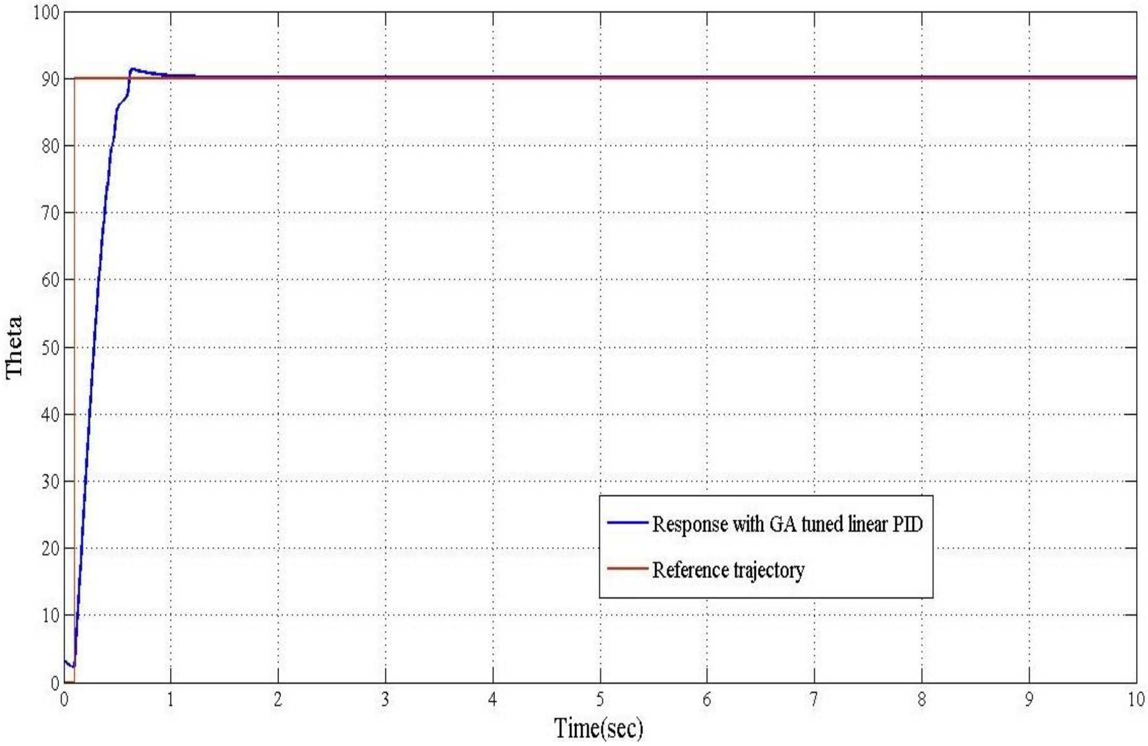

Figure 5.17Response of Inverted Pendulum on a progressing cart with linear GA-PID

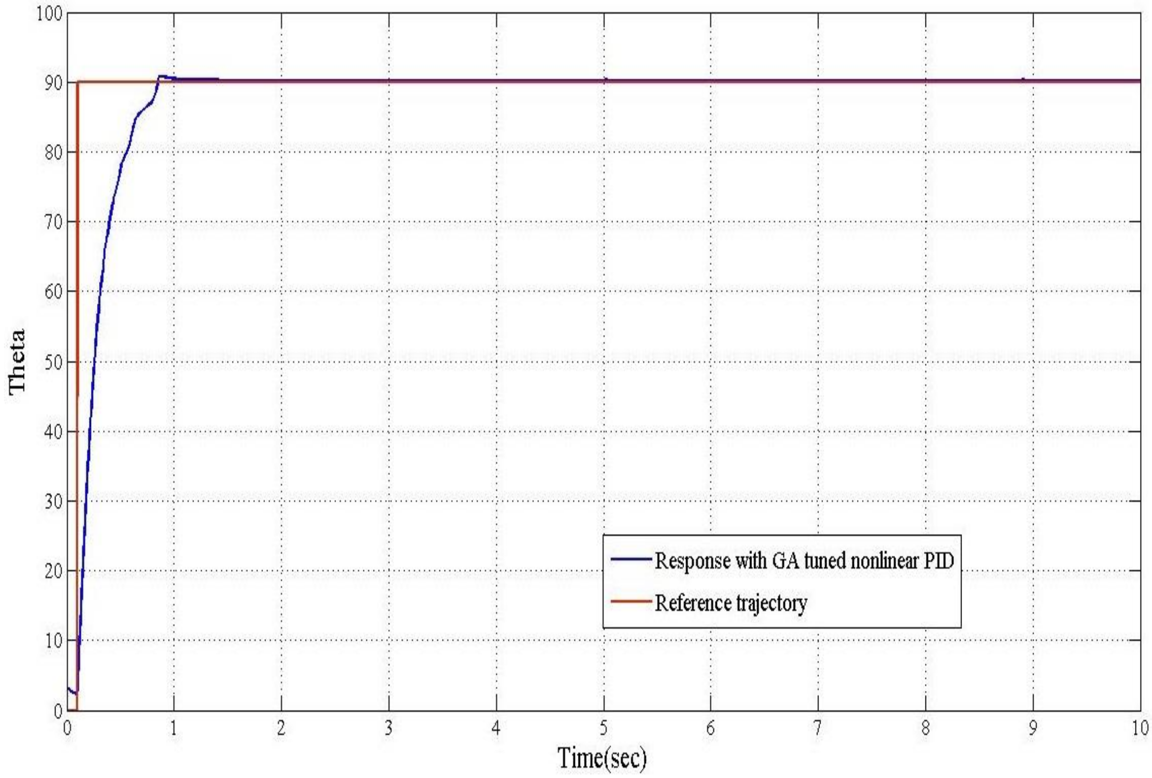

Figure 5.18 Response of Inverted Pendulum on a progressing cart with nonlinear GA-PID

CHAPTER 6
RESULTS AND CONCLUSION

6.1 RESULTS

The simulation results validate the performance of the system in terms of system overshoot, steady state error and settling time. The comparative results for mass spring damper system, PMBDC motor with robotic arm system and inverted pendulum on progressing cart system are shown in

Table 6.1,

Table and Table 6.3 respectively

Table 6.1 Performance evaluation of nonlinear mass-spring-damper with different controllers

Type of controller	**Nonlinear Mass Spring Damper System**		
	System overshoot (%)	**Steady state error**	**Settling time (sec)**
Without controller	34.919278	0.005039	0.0389
Conventional PID	9.7327	0.000025	0.0304
PSO-tuned linear PID	nil	0.000300	0.2043
PSO-tuned nonlinear PID	nil	0.000300	0.0128
GA-tuned linear PID	0.7628	-0.002175	0.4012
GA-tuned nonlinear PID	0.4736	-0.001272	0.3494

Table 6.2 Performance evaluation of PMBDC motor with robotic manipulator system with different controllers

Type of controller	PMBDC Motor with Robotic Manipulator System		
	System overshoot (%)	Steady state error	Settling time (sec)
Without controller	24.5546	-12.678	infinity
Conventional PID	63	-0.0005	28.199
PSO-tuned linear PID	4.3118	0.0536	8.3576
PSO-tuned nonlinear PID	0.7250	0.0100	6.6624
GA-tuned linear PID	6.2183	-0.0939	13.360
GA-tuned nonlinear PID	3.4760	0.0032	9.4083

Table 6.3 Performance evaluation of Inverted Pendulum on a Progressing Cart with different controllers

Type of controller	Inverted Pendulum on a progressing cart system		
	System overshoot (%)	Steady state error	Settling time (sec)
Without controller	nil	84.59584	infinity
Conventional PID	71.86526	-0.000290	0.3649
PSO-tuned linear PID	9.043406	-0.110326	1.0828
PSO-tuned nonlinear PID	1.171914	-0.140532	1.1965
GA-tuned linear PID	1.526151	-0.136794	1.3554
GA-tuned nonlinear PID	0.857396	-0.167234	1.5288

It can be observed from the results, that for the nonlinear dynamical systems the PSO tuned linear PID and GA tuned linear PID gives better performance than conventional PID. However when comparing between the GA tuned PID controller and PSO tuned PID controller the later gives better result of the two. In addition to this when a nonlinear controller is used the performance is improved more with little or no overshoot and better rising time than linear PSO-PID.

6.2 CONCLUSION

The simulation results and parameters computed shows that the performance of metaheuristic algorithm (PSO, GA) tuned linear proportional-integral-derivative (PID) controller is efficiently improved when compared with the performance of conventional PID for nonlinear dynamical systems. The system overshoot is considerably lowered with the use of proposed method of tuning. In addition to this it is shown that the use of a nonlinear PID controller instead of a linear PID controller improves the performance of the system with a decrease in both the overshoot as well as the settling time as compared to the response with a linear PID.

It is also shown in the performance evaluation of the two tuning method proposed in this project that tuning of PID with particle swarm optimizer gives the best performance. In the benchmark functions evaluation also the PSO was able to converge to global minima more efficiently and faster than GA algorithm. Thus overall a nonlinear PSO tuned PID controller gave the best result and performance for all the three nonlinear dynamic system considered in this project.

6.3 FURTHER SCOPE

Lot of research work is still going on in the field of optimization and tuning of controllers for nonlinear dynamical systems with metaheuristic algorithms. Real time implementation of such controllers can be extensively used in industries for automation and fine control of the plant and processes. Since PID controllers are the basic controllers used in almost all of industries hence, fine tuning of these controllers using the proposed method of PSO tuned nonlinear PID would be extremely helpful in increasing the performance, robustness and efficiency of the system. Modifications in the existing standard PSO algorithms and GA algorithms can be done to increase the performance of the optimization and parameter tuning.

REFERENCES

[1] J. G. Ziegler and N. B. Nichols, "Optimum Settings for Automatic Controllers," *ASME*, vol. 65, pp. 433-444, 1942.

[2] P. Wang and D. Kwok, "Auto-tuning of classical PID controllers using an advanced genetic algorithm," in *Proceedings of the 1992 International Conference on Power Electronics and Motion Control,Industrial Electronics, Control, Instrumentation, and Automation*, 1992.

[3] Zhenyu Yang and G. Pedersen, "Automatic tuning of PID controller for a 1-D levitation system using a genetic algorithm - a real case study," in *IEEE International Conference on Control Applications on Computer Aided Control System Design,IEEE International Symposium on Intelligent Control*, 2006.

[4] N. Mohammed, M. Xiuzhen and S. Enzhe, "Tuning of PID controller for diesel engines using genetic algorithm," in *IEEE International Conference on Mechatronics and Automation (ICMA)*, 2013.

[5] M. Salem and K. M.F, "Application of Biogeography based optimization in tuning a PID controller for nonlinear systems," in *International Conference on Complex Systems (ICCS)*, 2012.

[6] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," in *IEEE International Conference Proceedings on Neural Networks*, 1995.

[7] A. Oi, N. Chikashi, T. Matsui, H. Fujiwara, K. Matsumoto, H. Nishida, J. Ando and M. Kawaura, "Development of PSO-based PID tuning method," in *International Conference on Control, Automation and Systems*, 2008.

[8] S. Nema and P. Padhy, "PSO based PID tuning TITO process," in *IEEE International Conference on Computational Intelligence & Computing Research (ICCIC)*, 2012.

[9] S. Tsung-Ying, H. Cheng-Sen and T. Shang-Jeng, "Particle Swarm Optimizer based controller design for Vehicle Navigation System," in *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2008.

[10] R. Ahmed, H. Chaal and D. -W. Gu, "Spacecraft controller tuning using Particle Swarm Optimization," in *ICCAS-SICE*, 2009.

[11] A. Sharaf and A. El-Gammal, "An integral squared error -ISE optimal parameters tuning of modified PID controller for industrial PMDC motor based on Particle Swarm Optimization-PSO," in *IPEMC '09. IEEE 6th International on Power Electronics and Motion Control*, 2009.

[12] H. I. Jaafar, Z. Mohamed, A. F. Z. Abidin and Z. A. Ghani, "PSO-Tuned PID Controller for a Nonlinear Gantry Crane System," in *IEEE International Conference on Control System,Computing and Engineering*, Malaysia, 2012.

[13] P. Xiaotao, H. S. Shijie Cheng and Y. T. Shaorong Wang, "Nonlinear PID controller design for superconducting magnetic energy storage," in *39th International Universities Power Engineering Conference*, 2004.

[14] R. Aguilar, R. Martinez-Guerra and A. Poznyak, "Nonlinear PID controller for the regulation of fixed bed bioreactors," in *Proceedings of the 41st IEEE Conference on Decision and Control*, 2002.

[15] M. Tajjudin, N. Ishak, M. Rahiman and H. Ismail, "Model reference input for an optimal PID tuning using PSO," in *IEEE International Conference on Computing and Engineering (ICCSCE) Control System*, 2011.

[16] M. Shuhua and L. Wei, "Non-linear PID Control Method with Feed-forward in Smart Car," in *Chinese Control and Decision Conference*, 2009.

[17] G. Silva, A. Datta and S. Bhattacharyya, "New results on the synthesis of PID controllers," *IEEE Transactions on Automatic Control*, vol. 47, no. 2, pp. 241,252, 2002.

[18] L. Junli, M. Jianlin and Z. Guanghui, "Evolutionary Algorithms Based Parameters Tuning of PID Controller," in *Chinese Control and Decision Conference*, China, 2011.

[19] X.-z. Li, F. Yu and Y.-b. Wang, "PSO Algorithm Based Online Self-Tuning of PID Controller," in *International Conference on Computational Intelligence and Security*, 2007.

[20] D. Kwok and P. Wang, "Fine-tuning Of Classical PID Controllers Based On Genetic Algorithms," in *IEEE International Workshop on Emerging Technologies and Factory Automation*, 1992.

[21] N. F. Mohammed, M. Xiuzhen and E. Song, "Tuning of PID Controller for Diesel Engines Using Genetic Algorithm," in *International Conference on Mechatronics and Automation*, Takamatsu,Japan, 2013.

[22] J. Digalakis and K. Margaritis, "An experimental study of benchmarking functions for genetic algorithms," in *IEEE International Conference on Systems, Man, and Cybernetics*, 2000.

[23] M. A. Abido, "Optimal Design of Power System Stabilizers using Particle Swarm Optimization," *IEEE Transactions on Energy Conversion*, vol. 17, no. 3, pp. 406-413, 2002.

[24] Z. Bingul and O. Karahan, "Tuning of fractional PID controllers using PSO algorithm for robot trajectory control," in *IEEE International Conference on Mechatronics (ICM)*, 2011.

[25] C. Ping-Lin, M.-C. Yang and S. Tsung-Ying, "PSO-based on-line tuning PID controller for setpoint changes and load disturbance," in *IEEE Congress on Evolutionary Computation (CEC)*, 2011.

[26] X. Wang, R. Li, Y. Wang, Y. Peng and B. Qin, "Self-tuning PID Controller with Variable Parameters Based on Particle Swarm Optimization," in *Third International Conference on Intelligent System Design and Engineering Applications*, 2013.

[27] R. Kanojiya and P. M. Meshram, "Optimal tuning of PI controller for speed control of DC motor drive using particle swarm optimization," in *International Conference on Advances in Power Conversion and Energy Technologies (APCET)*, 2012.

[28] F. Hongqing and C. Long, "Application of an enhanced PSO algorithm to optimal tuning of PID gains," in *Chinese Conference on Control and Decision*, 2009.

[29] M. Korkmaz, O. Aydogdu and H. Dogan, "Design and performance comparison of variable parameter nonlinear PID controller and genetic algorithm based PID controller," in *International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*, 2012.

[30] A. Nazir, Gautham, R. Surajan and L. Binu, "simplified Genetic Algorithm for online tuning of PID controller in LabView," in *World Congress on Nature & Biologically Inspired Computing (NaBIC)*, 2009.

[31] Y. Mitsukura, T. Yamamoto and M. Kaneda, "A Genetic Tuning Algorithm of PID Parameters," in *IEEE International Conference on Computational Cybernetics and Simulation*, 1997.

[32] Y. Mitsukura, T. Yamamoto and M. Kaneda, "A design of self-tuning PID controllers using a genetic algorithm," in *Proceedings of the American Control Conference*, 1999.

[33] W.-H. Chen, D. Ballance, P. Gawthrop, J. Gribble and J. O'Reilly, "Nonlinear PID predictive controller," in *IEE Proceedings on Control Theory and Applications*, 1999.

[34] D. Pereira and J. O. P. Pinto, "Genetic algorithm based system identification and PID tuning for optimum adaptive control," in *Proceedings IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2005.

[35] Wei-You Cai; Wei-Min Qi; Qiao-ling Ji; Yuan-Chu Cheng, "A Design of Nonlinear Adaptive PID Controller Based on Genetic Algorithm," in *Chinese Control Conference (CCC)*, 2006.

[36] Y. Su, Dong Sun and B. Duan, "Design of an enhanced nonlinear PID controller," *Mechatronics (Sciencedirect)* , vol. 15, no. 8, pp. 1005,1024, 2005.

[37] S. Sivanandam and S. N. Deepa, Introduction to Genetic Algorithms, Springer, 2007.

[38] L. Behera and I. Kar, Intelligent Systems and Control: Principles and Applications, Oxford higher education, 2009.

[39] S. H. Żak, Systems and Control,Engineering & Technology,Oxford series in electrical and computer engineering, Oxford University Press, 2003.

[40] P. Zhang and H. W. Mingzhe Yuan , Self-Tuning PID Based on Adaptive Genetic Algorithms with the Application of Activated Sludge Aeration Process, Proceedings of the 6th World Congress on Intelligent Control and Automation, 2006.

[41] Liu, Li, Li, Huanzhou and Tang, Zhangguo, "PID tuning based on improved quantum genetic algorithm," in *Sixth International Symposium on Computational Intelligence and Design*, 2013.

[42] Li, Chengwei and Lian, Jiandong, "The Application of Immune Genetic Algorithm in PID Parameter Optimization for Level Control System," in *Proceedings of the IEEE International Conference on Automation and Logistics*, 2007.

[43] D. H. Kim, "Comparison of PID Controller Tuning of Power Plant Using Immune and Genetic Algorithms," in *nternational Symposium on Computational Intelligence for Mearurements and Application*, 2003.

[44] Khoie, Maryam, "PID Controller Tuning using Multi-Objective Optimization based on Fused Genetic-Immune Algorithm and Immune Feedback Mechanism," in *Proceedings of the 2011 IEEE International Conference on Mechatronics and Automat*, 2011.

[45] Fabiano, Claudio, Toledo, Motta, Carlos, São, Lima, João Miguel G and Arantes, Silva, "A Multi-Population Genetic Algorithm Approach for PID Controller," 2012.

[46] Das, Saptarshi, Pan, Indranil, Das, Shantanu and Gupta, Amitava, "Genetic Algorithm Based Improved Sub-Optimal Model Reduction in Nyquist Plane for Optimal Tuning Rule Extraction of PID and PI," 2011.

[47] S. Sastry, Nonlinear Systems: Analysis, Stability and Control, Springer, 1999.

[48] T. Kapitaniak, Chaos for Engineers: Theory, Applications, and Control, Springer, 2000.

[49] E. J. Davison, Benchmark Problems for Control System Design: Report of the IFAC Theory Committee, International Federation of Automatic Control, 1990.

[50] Tan, K K, Huang, S and Ferdous, R, "Robust self-tuning PID controller for nonlinear systems," *Journal of Process Control,* vol. 12, pp. 753-761, 2002.

[51] Chang, Wei-der, Hwang, Rey-chue and Hsieh, Jer-guang, "A self-tuning PID control for a class of nonlinear systems based on the Lyapunov approach," *Journal of Process Control,Elsevier,* vol. 12, pp. 233-242, 2002.

[52] Letting, L K, Munda, J L and Hamam, Y, "Optimization of a fuzzy logic controller for PV grid inverter control using S-function based PSO," *Solar Energy,Elsevier,* vol. 86, no. 5, pp. 1689-1700, 2012.

[53] Qais, Mohammed and Z. AbdulWahid, "A New Method for Improving Particle Swarm Optimization Algorithm (TriPSO)," in *IEEE*, 2013.

[54] Liu, Yuanning, Wang, Gang, Chen, Huiling, Dong, Hao, Zhu, Xiaodong and Wang, Sujing, "An Improved Particle Swarm Optimization for Feature Selection," *Journal of Bionic Engineering,Science Direct,* vol. 8, pp. 1-10, 2011.

[55] Li, W T, Shi, X W and Y. Q. Hei, "An Improved Particle Swarm Optimization Algorithm For Pattern Synthesis Of Phased Arrays," *Progress In Electromagnetics Research,PIER,* vol. 82, p. 319–332, 2008.

[56] Ho, S L, Yang, S Y, Ni, G Z and Wong, K F, "An Improved PSO Method With Application to Multimodal Functions of Inverse Problems," *IEEE Transactions On Magnetics,* vol. 43, no. 4, pp. 1597-1600, 2007.