Stability Control of Ball & Beam System Using Evolutionary Computational Techniques

Tayyab Ali

Reg No.: 390-FET/MSEE/F14

Supervised By:

Dr. Suheel Abdullah Malik

Department of Electrical Engineering

Faculty of Engineering and Technology

International Islamic University Islamabad

2017

Stability Control of Ball & Beam System Using Evolutionary Computational Techniques

Tayyab Ali

Reg No.: 390-FET/MSEE/F14

Supervised By:

Dr. Suheel Abdullah Malik

Department of Electrical Engineering

Faculty of Engineering and Technology

International Islamic University Islamabad

2017

PID controllers

Control theory.

MATLAB

Genetic algorithm

Stability Control of Ball & Beam System Using Evolutionary Computational Techniques

Tayyab Ali

(Reg No.: 390-FET/MSEE/F14)

This dissertation is submitted to Faculty of Engineering and Technology, International Islamic University Islamabad Pakistan for partial fulfillment of the degree of **MS Electronic Engineering** with specialization in **Control Systems** at the Department of Electrical Engineering.

Supervised By:

Dr. Suheel Abdullah Malik

Faculty of Engineering and Technology

International Islamic University Islamabad

Certificate of Approval

This is to certify that the work contained in this thesis entitled, "**Stability Control of Ball & Beam System Using Evolutionary Computational Techniques**" was carried out by **Mr. Tayyab Ali, Registration No. 390-FET/MSEE/F14** and it is fully adequate in scope and quality for the degree of MS Electronic Engineering.

Viva Voce Committee:

Supervisor

Dr. Suheel Abdullah Malik

Associate Professor, DEE, FET, IIU, Islamabad

External Examiner

Dr. Tanveer Ahmad Cheema

Professor, ISRA University, Islamabad

Internal Examiner

Dr. Ihsan ul Haq

Assistant Professor, DEE, FET, IIU, Islamabad

Chairman DEE, FET, IIU Islamabad

Dr. Suheel Abdullah Malik

Associate Professor, DEE, FET, IIU, Islamabad

Dean DEE, FET, IIU Islamabad

Dr. Muhammad Amir

Professor, DEE, FET, IIU, Islamabad

Declaration

I certify that research work titled **"Stability Control of Ball & Beam System Using Evolutionary Computational Techniques"** is my own work and has not been presented elsewhere for assessment. Moreover, the material taken from other sources has also been acknowledged properly.

Tayyab Ali

390-FET/MSEE/F14

Date: 25/07/17

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

In the name of Allah (SWT), the most beneficent and the most merciful

Dedicated

to

My caring Parents, Wife, Brothers, Sister and Respected Teachers

Acknowledgment

In the name of Allah (S.W.T), who gave me the courage and patience to carry out this research work. Peace and blessings of Allah be upon His last Prophet Muhammad (S.A.W) and all his Sahaba (R.A) who devoted their lives for the prosperity and spread of Islam.

By the grace of Almighty Allah, I would like to express my admiration for the assistance provided during the groundwork of this thesis. I would like to take this opportunity and give special thanks to my supervisor **Dr. Suheel Abdullah Malik** for his kind supervision and valuable guidance. It was not possible for me to complete this research work without his kindness, motivation and support. He helped and guided me in completing this dissertation. He gave me the most needed supervision and took keen interest in all the matters related to my thesis.

I would like to pay thanks to my friends and colleagues especially to Dr. Abdul Basit who helped me during my thesis. I pray to Almighty Allah for His blessing upon all those people who have cooperated with me during my research. May Allah help us in following the true spirit & principles of Islam written down in the Holy Qur'an and Sunnah (Aameen).

Tayyab Ali

390-FET/MSEE/F14

Abstract

The ball & beam system is one of the widely used benchmark for the stability analysis. In this system, aim is to change the ball position as desired by varying the beam angle. Ball and beam system exhibits an unstable open loop response. In order to make it stable, some control mechanism is employed by keeping in view the system's dynamics. In this thesis, controllers such as Proportional Integral- Derivative (PI-D) and Proportional Integral-Proportional Derivative (PI-PD) have been implemented for the stability control of ball & beam system. Moreover, Proportional-Derivative (PD), Proportional-Integral (PI) and Integral-Proportional (I-P) controller have also been investigated for the said purpose. The tuning of these controllers has been done using evolutionary computation techniques including Simulated Annealing (SA) and Cuckoo Search Algorithm (CSA) and Genetic algorithm (GA). Fitness function for each controller has been solved by using these evolutionary techniques and finally the optimum values of the controller parameters such as $\mathbf{K_P}$, $\mathbf{K_i}$ and $\mathbf{K_D}$ have been obtained. Simulations have been carried out using MATLAB/Simulink software. The transient performance of the evolutionary computation based controllers has been observed using four different performance indices such as Integral of time multiplied by absolute value of error (ITAE), Integral of absolute value of error (IAE), Integral of time multiplied by squared value of error (ITSE) and Integral of squared error (ISE). The rise time (t_r), steady-state error (e_{SS}), settling time (t_S) and % overshoot (os) and have been considered as performance parameters. The step response and set point tracking has been efficiently investigated. A comprehensive comparative analysis has been carried between various controllers which shows that PI-PD controller tuned with evolutionary computation techniques is very effective for the set point tracking of ball and beam system which yields transient performance with fairly small overshoot and settling time. Further, transient response of evolutionary computation based PI-PD controller has been compared with PID and I-PD controller which shows that CSA based PI-PD controller exhibits much better transient response than other controllers.

Table of Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Overview

The ball and beam system consists of a ball, beam, support block, motor, lever arm, gear system, belt pulley and position sensors. It is used as a benchmark for studying stability analysis of different control systems [1]. This system is widely used in control systems laboratories for the demonstration of stability control [2]. Ball and beam system is used to teach automatic control for doing scientific research in related areas. There are many modern and classical control methodologies which have been implemented for its stability. In ball and beam system, the aim is to control ball position on the beam by generating torque from the motor. The information from the position sensors is acquired by applying different controlling techniques, and then this information is compared with the reference signal. The difference is fed back to the designed controller to achieve the desired position by eliminating the error signal. Ball and beam system has additional nonlinearities when we practically implement this system in the laboratory [3]. These nonlinearities include backlash introduced by the DC motor and belt pulley and irregular rolling surface. It has two degrees of freedom [2][4]. One is beam rotating through its central axis and other is ball rolling up and down on the beam [5]. Feedback control is necessary for the stability of ball and beam system due to its unstable behavior. The main goal is to adjust the ball position according to the reference position by rejecting all unwanted disturbances. When ball position is fed back to the controller, a control signal is generated. This control signal is fed to the DC motor which generates the torque for the rotation of the beam to acquire the desired ball position [2].

In the past, different controllers such as PD, PID, LQR and Fuzzy controllers have been employed for the stability control of ball & beam system. Tuning of these controllers to achieve the optimum results is always a major task. In past, different tuning techniques have been explored to do this. The conventional and modern tuning methods include Ziegler-Nicholas method, Modified-Ziegler-Nicholas method, Hit and Trial method, Fuzzy Logic, and Cohen-Cool method etc. In the recent past, evolutionary computational techniques have been used and it has been found that these techniques are very useful for the control of different optimization problems. These techniques include Cuckoo Search Algorithm (CSA), Particle Swam Optimization (PSO), Genetic Algorithm (GA), Simulated Annealing (SA), Differential Evolution (DE) etc. Although different controllers have been utilized for the stability of ball and beam system but evolutionary computational based two degrees of freedom (2DOF) controllers such as Proportional Integral-Derivative (PI-D) and Proportional Integral-Proportional Derivative (PI-PD) controllers have not been explored yet for the control of the ball & beam system. PI-D and PI-PD controllers are basically modified versions of classical PID controller. Moreover Proportional Derivative (PD), Proportional Integral (PI) and Integral-Proportional (I-P) controller have also been implemented by using evolutionary computational techniques for comparisons. The evolutionary computational techniques which have been used in this thesis are Cuckoo Search Algorithm (CSA), Simulated Annealing (SA) and Genetic Algorithm (GA). Controllers have been tuned with these tuning techniques subjected to minimize four different performance indices including ISE, IAE, ITAE and ITSE. The comparative analysis has been done according to the different performance parameters of the controllers such as settling time (t_s), rise time (t_r), steady state error (e_{ss}) and % overshoot (os). It can be concluded from the comparative analysis that PI-PD controller tuned with each evolutionary computational technique is very effective for the stability control of ball and beam system as compared to other controllers. Moreover, an excellent set point tracking has also been achieved with PI-PD controller.

1.2 Problem statement

Ball and beam system is unstable and non-linear system [2] and it covers a wide range of practical applications like landing of airplanes, chemical reaction control, rocket balancing during its launch and control the speed of induction motors etc. In past, various controlling schemes have been used for the stability control of ball on the mounted beam. However, evolutionary computational based controllers including PI-D and PI-PD have not been explored yet. These two degrees of freedom controllers (PI-D and PI-PD) are quite different from classical PID controller. In two degrees of freedom (2DOF) controlling schemes, one controller is in the feedback path whereas the other controller is in the feed-forward path. Feedback characteristics and closed-loop characteristics can be adjusted independently for the improvement of steady state and transient response of the system [6]. In this research, PI-D and PI-PD controller have been explored. Moreover tuning of these controllers has been done with evolutionary computation techniques including Cuckoo Search Algorithm (CSA), Genetic Algorithm (GA) and Simulated Annealing (SA). However, evolutionary computation based PI, IP and PD controllers have also been implemented. Performance comparison has been presented to analyze the transient and steady state performance of each proposed controller. A comprehensive comparative analysis of each controller with each tuning technique has also been done.

1.3 Goals and objectives

1. Study of the different control methods of ball and beam system.
2. Deduction of a mathematical model for the ball and beam system.
3. Design & modeling of controllers such as PI, I-P and PD.
4. Design and modeling of two degrees of freedom controllers such as PI-D and PI- PD.
5. To derive fitness function for each proposed controller
6. Tuning of proposed controllers using evolutionary computing techniques like Genetic Algorithm (GA), Cuckoo Search Algorithm (CSA) and Simulated Annealing (SA).
7. Implementation of tuned controllers in MATLAB/Simulink software.
8. Comparative analysis between different controllers implemented with each proposed tuning technique.

1.4 Thesis Organization

In Chapter 1, introduction to ball & beam system has been given. In Chapter 2, a detailed literature overview of stability control of ball & beam system has been provided. In this chapter, a summary of implemented controllers along with tuning algorithms has been presented. In Chapter 3, mathematical model for the proposed ball and beam system has been derived. In Chapter 4, our propose controllers such as P, PI, I-P, PD, PI-D and PI-PD have been explained. Finally these controllers have been applied to the proposed ball and beam system whose mathematical modeling has been deduced in Chapter 3. Chapter 5 will give a description of evolutionary computational algorithms including Cuckoo Search Algorithm (CSA), Simulated Annealing (SA) and Genetic Algorithm (GA). In Chapter 6, fitness functions for each proposed controller have been derived.

Furthermore, different performance indices have also been discussed in Chapter 6. In Chapter 7, controllers have been designed and implemented in MATLAB/Simulink. Simulation results and discussions have also been provided. A detailed comparative analysis has also been presented for the evaluation of the controllers. In Chapter 8, conclusions and future trends have been summarized.

CHAPTER 2

Literature Review

The stability control of ball & beam system is always a difficult and challenging task for the researchers. Many controlling schemes have been presented in past to make this system stable and to get better transient responses. A summary of literature survey has been provided in this section.

Asymptotically stable PD controller has been designed for the stability control the ball & beam system. By introducing non linear compensators into traditional PD controller, a new controller has been constructed as presented in [1].

PID controller is very effective for the control of industrial appliances. Simple Internal Model Control (SIMC) based PID controller has also been implemented for the ball and beam system. Moreover H–infinity based PID controller has also been designed. Nyquist plot has been used to analyze the convergence of both controllers. From software and hardware results, it has been concluded that both SIMC and H-infinity based PID controller are very effective for ball and beam system [3].

Linear Quadratic Regulator (LQR) has been designed using Genetic Algorithm (GA) for the ball and beam system. Jacobean Linearization (JL) method has been utilized for the linearization of the system around a particular operating point. Due to sensor noise, a state observer has also been presented to observe the motion of ball accurately. Finally, a non-model based PID controller and model-based hybrid PID-LQR have been implemented. The experimental results are summarized and it has been concluded that model based control strategies (LQR) are better than non-model based controllers for the nonlinear ball and beam system [4].

Tracking controller for ball & system has also been implemented using nonlinear servomechanism theory of nonlinear systems. The designed controller was compared with the non-linear servo-regulator design. The simulation results reveal that proposed design has tremendous tracking property with excellent control [7].

Fuzzy controller has been designed for the stabilization of ball and beam system. By using Lyapnouv's stability theory, a new stability criterion has been derived. Finally it has been suggested that designed approach can be used for fuzzy controllers having both negative and positive grades of membership [8].

A controller has also been devised by evolving a feed forward neural network to control the ball & beam system. That network has been implemented with genetic algorithm. This genetic approach can also be used to control other non linear systems very easily and efficiently [9].

Lyapnouv direct method has been used to implement the control of ball and beam system. In this method, an energy function has been defined by using potential and kinetic energy of the system. That energy function is also known as candidate Lyapnouv function. This Lyapnouv function should be strictly positive. When the Lyapnouv function is known, then find out its derivative, that should be strictly negative. If it is, then system will be stable [10].

Dynamic responses of fuzzy logic controller, PID controller and single input fuzzy logic controller (SIFLC) have been compared. It has been observed that SIFLC can stabilize the ball and beam system more efficiently. After certain trials, it has been analyzed that transient period of this controller is better than conventional FLC and PID in the sense that systems response comprises of less % overshoot with zero steady state error [11].

STF-LQR control approach has been introduced for the ball & beam system. By using physical laws, complete mathematical model has been derived. In order to balance the un-modeled dynamics, a linearize state space model is deduced that is predicted by using STF. By using discrete time LQR approach, finally we can design a desired controller [12].

PSO algorithm based PID controller has also been implemented for the stability control of ball & beam system. In PSO, different trials are being done. After successive trials, the best trial is selected. Finally, results of PSO algorithm and ITAE equation method for PID tuning have been compared [13].

PID controller has been designed for ball and beam system by using evolutionary computational techniques including Differential Evolution and Genetic Algorithm (DE and GA). Optimum PID controller has been designed with GA and DE. But it has been suggested that DE based PID is effective with respect to performance and speed of convergence. It has also been observed that DE based PID exhibits faster disturbance rejection as compared to GA based PID. MATLAB tools have been used to achieve all the results of controllers [14].

A model predictive control (MPC) approach has been used in order to obtain non-linear control law which satisfies input constraints. Primal-dual interior point algorithm has been designed as an optimization solver. Experimental results have been compiled for the comparison purpose by considering three control methods; one saturated LQR and two different MPCs [15].

Improved Ant Colony Optimization (ACO) based fuzzy controller has also been implemented for the real-time ball & beam system. ACO is basically a natured inspired algorithm which can be used for the optimum tuning of the controllers. Experimental and simulation results tell us that proposed controls scheme is much better than conventional ACO in terms of convergence speed and accuracy [16].

PID controller has also been designed using different tuning techniques for the control of 1st order, 2nd order (like ball and beam system) and 3rd order systems. Simulation results have been done for the comparative analysis and it has been observed that PID control reveals satisfactory transient response[17].

Robust self-tuning PID controller has been implemented for nonlinear systems. There are different types of non-linear systems exist around us, this paper demonstrates that robust self-tuning PID controller can also be used to stabilize non-linear systems [18].

A fuzzy cascade controller has also been employed for the ball and beam system. The idea of parallel genetic algorithms has been explored here. Basically this is an application of Artificial Intelligence (AI). It has been proved that proposed algorithm is applicable to real-time systems (like ball & beam system) [19].

I-PD and PID controller have been designed using Particle Swarm Optimization (PSO) and H-infinity method respectively. After simulation results, it has been suggested that H-Infinity based PID controller has much better transient and steady state response as compared to I-PD controller [20].

"Output Feedback Control of a Ball and Beam System Based on Jacobean Linearization (JL) under Sensor Noise" has also been implemented. In this paper, "feedback sensor noise" has been considered as a control problem. Since sensor noise always lead to system failure, therefore author has designed a robust output feedback controller to minimize the noise coming from the sensor. The controller has gain-scaling factor. Simulations have been carried out and they reveal that output feedback controller minimized the sensor noise by increasing gain-scaling factor [21].

PD and Fuzzy logic controller have also been used simultaneously for the stability control of ball and beam system. Fuzzy Logic controller (FLC) has been implemented for the outer-loop whereas PD controller has been utilized for inner-loop. Further, a PID controller has also been implemented for the comparison purpose. Simulation results reveal that FLC is much better than classical PID controller in terms of transient response [22].

T-S Fuzzy model based dynamic model has also been derived for the ball & beam system. Moreover, an adaptive dynamic surface control (DSC) has been designed for the control of real-time ball and beam system subjected to different uncertainties. Lyapunov theorem is applied to conserve the stability of the overall system. Simulations have also been carried out. Experimental and simulation results reveal that transient performance of the suggested controller is much better than classical dynamic surface controller (DSC) [23].

A pair of decoupled fuzzy sliding-mode controllers (DFSMCs) has been explored for the real-time ball & beam system. Ant colony optimization (ACO) has been employed for the optimum tuning of the proposed controller. The experimental and simulation results reveal that proposed controlling scheme along with tuning algorithm have provided much better transient response than conventional schemes [24].

Dynamic and static sliding-mode controllers have been successfully implemented for the control of ball and beam system. These two controllers have been designed using complete model of the ball & beam system. Then, same two controllers have been designed using simplified model of the system. It has been observed that proposed two controllers designed with complete model of the system are better than other two controllers in terms of transient response. Moreover, experimental results tell us that these two controllers have been reduced the chattering effect which is normally associated with sliding mode controllers [25].

A model reference adaptive controller has been designed by using MIT rule for the control of real-time ball and beam system. According to the reference model, MIT rule has been developed using gradient theory. A linearized model of proposed ball & beam system is considered for the controller design in MATLAB. Results indicate that proposed adaptive controller is much better in terms of transient performance [26].

Adaptive system control based on Genetic Algorithm (GA) has been presented for the ball & beam system. GA is an evolutionary computational algorithm in which random solutions are generated and evaluated. The solution which is better than all is considered as optimum. Basically this solution represents the gain parameters of the controller. Results reveal that proposed control scheme has given the satisfactory response [27].

The Inverse Lyapunov (IL) approach with energy-shaping technique has also been used to design a controller for ball & beam. By using a shaped Lyapunov function, a stabilizing controller is obtained. The invariance theorem of LaSalle has been utilized for the analysis of close loop system. Simulations results tell us that proposed controller is very effective for the ball & beam system [28].

A model based adaptive control algorithm has also been developed for the stability control of ball & beam system. The proposed algorithm is based on the error metric. Singularity problem has been overcome by improving the algorithm. Experimental results have been compiled and it has been concluded that proposed controlling scheme is very useful for the control of ball & beam system [29].

CHAPTER 3

Transfer Function of Ball & Beam System and Controllers

The stability control of ball & beam system covers many modern and conventional design methods. There are different types of ball and beam systems. The ball and beam system which has been considered in this thesis is given in Figure 3.1. In ball and beam system, the beam is mounted on the output shaft of an electrical motor. The beam will be twisted about its center axis when an electrical control signal will be applied to the motor amplifier. The main objective is to adjust the position of the ball on the beam automatically by altering angle of the beam. In this chapter, the transfer function of ball and beam system is deduced. Further, description of different controlling schemes has also been given in detail.

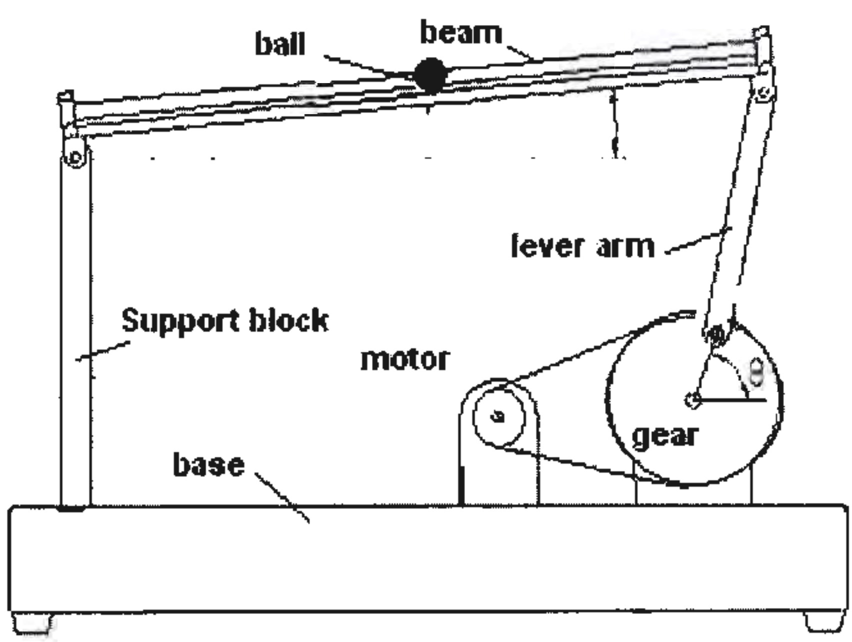

Figure 3.1: Ball and Beam System

3.1 Transfer Function of Ball and Beam System

Figure 3.2 represents the schematic diagram of the ball & beam system used in this Thesis. Different parameters of the ball & beam system along with their values are given in Table 3.1.

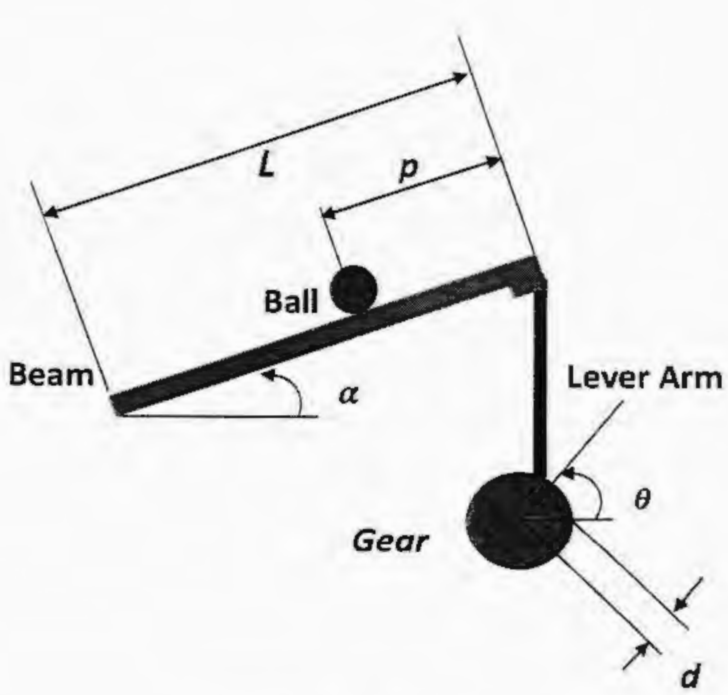

Figure 3.2: Schematic of Ball and Beam System

Transfer function of ball & beam system (**G_{BB}(s)**) between gear angle (**θ(s)**) and ball position (**P(s)**) has been derived in this section. Simplified transfer function of ball and beam system can be represented by Figure 3.3.

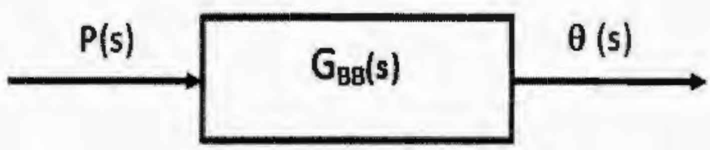

Figure 3.3: Basic Transfer Function block for Ball and Beam System

The ball's linear acceleration along the mounted beam can be expressed as Lagrangian equation of motion [3][20].

$$\left(\frac{J}{R^2}+m\right)\ddot{p} + mg\sin\alpha - mp\dot{\alpha}^2 \qquad (3.1)$$

where

α = Beam angle
J = Ball's moment of inertia
R = Radius of the ball

p = Position of the ball

m = Mass of the ball

g = Gravitational acceleration

Linearization of equation 3.1 about beam angle $\alpha = 0$ gives

$$\left(\frac{J}{R^2} + m\right)\ddot{p} = -\, mg\alpha \tag{3.2}$$

also $$\alpha = \left(\frac{d}{L}\right)\theta \tag{3.3}$$

where ***d*** represents the distance between the center of the gear and joint of the lever arm and L is the beam length

Substituting 3 into 2,

$$\left(\frac{J}{R^2} + m\right)\ddot{p} = -\, mg\left(\frac{d}{L}\right)\theta \tag{3.4}$$

Taking Laplace transform of 4 yields as

$$G_{BB}(s) = \frac{P(s)}{\theta(s)} = -\frac{mgd}{L\left(\frac{J}{R^2}+m\right)}\frac{1}{s^2} \tag{3.5}$$

$G_{BB}(s)$ represents the transfer function of the ball and beam system.

Table 3.1: Parameters of Ball and Beam System

Symbolic parameters	Description of Parameters	Values
L	Length of the beam	1m
R	Radius of the Ball	0.015 m
d	Lever arm offset	0.04 m
g	Gravitational acceleration	-9.8m/s^2
J	Moment Inertia of the Ball	2mR^2/5 kgm^2
m	Mass of the ball	0.11 kg

Substituting the values from Table 1 yields,

$$G_{BB}(s) = \frac{P(s)}{\theta(s)} = \frac{0.28}{s^2} \tag{3.6}$$

$\mathbf{G_{BB}}$(s) is the open loop transfer function [ratio of ball position P(s) to gear angle θ(s)] of overall ball and beam system. From 3.6, it can be concluded that plant's transfer function is representing a "double integrator".

3.2 Controllers

A controller is a device which makes an unstable system as a stable one. The output date is compared with the reference data to generate an error signal **(e(t))**. Error signal will derive the controller. Basic task of controller is to minimize this error signal. For a minimum value of error, plant will run smoothly without any disturbance. The stability control of ball and beam system has been investigated in this thesis. Due to open loop unstable behavior of ball and beam system, it requires a controller to act as a stable system. In past, many controllers have been used to do this but nobody has used evolutionary computational based two degrees of freedom controllers like PI-D and PI-PD. We have explored theses controllers by using evolutionary computation techniques including Cuckoo Search Algorithm (CSA), Simulated Annealing (SA) and Genetic Algorithm (GA). Moreover evolutionary computation based PD, I-P and PI controller has also been implemented for the investigation of stability control of ball and beam system.

3.2.1 Proportional (P) controller

Proportional controller can stabilize only 1st order unstable process. P controller contains proportional term only which will lead to forced oscillations. Rise time can be reduced with the P controller but steady state error cannot be eliminated. The Figure 3.4 represents the of P controller's block diagram with proposed system.

where

$G_P(s)$ is the transfer function of P controller

$G_{BB}(s)$ is the transfer function of the ball and beam system

Y(s) represents the output of the overall closed loop system

R(s) is the input to the system

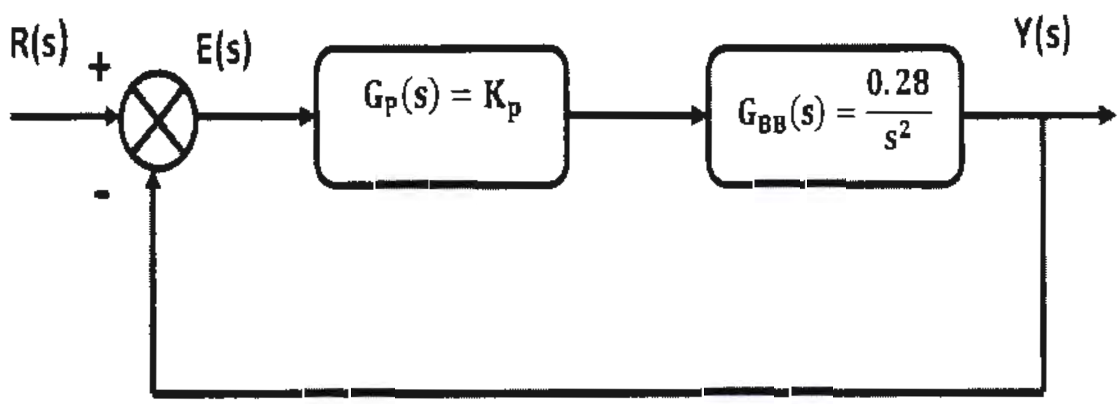

Figure 3.4: Block diagram of P controller

The expression for the closed-loop transfer function can be written as:

$$\frac{Y(s)}{R(s)} = \frac{G_P(s).G_{BB}(s)}{1+G_P(s).G_{BB}(s)} \tag{3.7}$$

$$\frac{Y(s)}{R(s)} = \frac{0.28*\frac{K_P}{s^2}}{1+\left(0.28*\frac{K_P}{s^2}\right)} \tag{3.8}$$

$$\frac{Y(s)}{R(s)} = \frac{0.28K_P}{s^2+0.28K_P} \tag{3.9}$$

3.2.2 Proportional Integral (PI) controller

Proportional Integral (PI) controller have two controllers (P and I) connected in feed forward path. Presence of integral term in PI controller will tend to decrease the rise time along with the elimination of steady state error. It has some disadvantages like more settling time and overshoot. Figure 3.5 represents the PI controller's block diagram.

where

$G_{PI}(s)$ represents the transfer function of PI controller

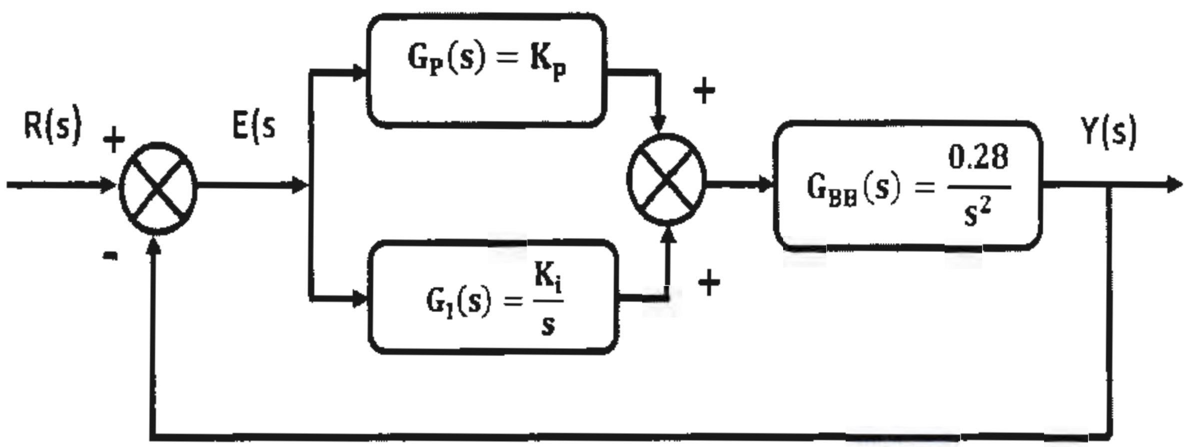

Figure 3.5: Block Diagram of PI Controller

The expression for the closed-loop transfer function can be written as:

$$\frac{Y(s)}{R(s)} = \frac{G_{PI}(s).G_{BB}(s)}{1+G_P(s).G_{BB}(s)} \tag{3.10}$$

After solving equation 3.10,

$$\frac{Y(s)}{R(s)} = \frac{0.28K_p s+0.28K_i}{s^3+0.28K_p s+0.28K_i} \tag{3.11}$$

3.2.3 Proportional Derivative (PD) Controller

Proportional Derivative (PD) controller has P and D controllers both in feed-forward path. Presences of Derivative controller will tend to reduce both settling time and overshoot. PD controller can be used for the control of moving objects such as rockets, flying & underwater vehicles and ships etc. Figure 3.6 shows the PD controller's block diagram.

where

$G_{PD}(s)$ represents the transfer function of PD controller

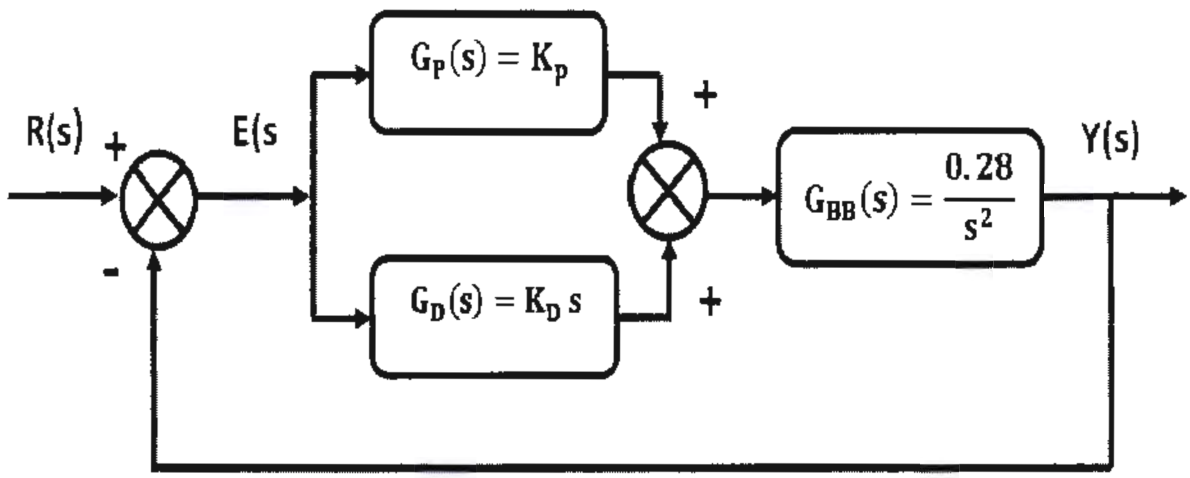

Figure 3.6: Block diagram of PD controller

The expression for the closed-loop transfer function can be written as:

$$\frac{Y(s)}{R(s)} = \frac{G_{PD}(s).G_{BB}(s)}{1+G_{PD}(s).G_{BB}(s)} \tag{3.12}$$

After solving equation 3.12,

$$\frac{Y(s)}{R(s)} = \frac{0.28K_D s+0.28K_P}{s^2+0.28K_D s+0.28K_P} \tag{3.13}$$

3.3 Two Degrees of Freedom controllers (2DOF)

Two degrees of freedom controller (2DOF) provides both feed-back and close-loop characteristics which can be controlled independently for the improvement of steady state and transient response of the system [6]. Following 2DOF configurations have been implemented in this thesis.

1. I-P controller
2. PI-D controller
3. PI-PD controller

3.3.1 Integral-Proportional (I-P) Controller

The Integral-Proportional (I-P) controller is the modified version of typical Proportional Integral (PI) controller [30]. The I-P controller can remove the disadvantages of PI controller like slow response and high starting overshoot [31]. In this controller, Proportional controller is connected in feedback path whereas Integral controller is connected in feed forward path. This configuration will introduce a zero in the overall system's transfer function which will minimize the overshoot. Figure 3.7 shows the I-P controller's block diagram.

where

$G_P(s)$ represents the transfer function of P controller

$G_I(s)$ represents the transfer function of I controller

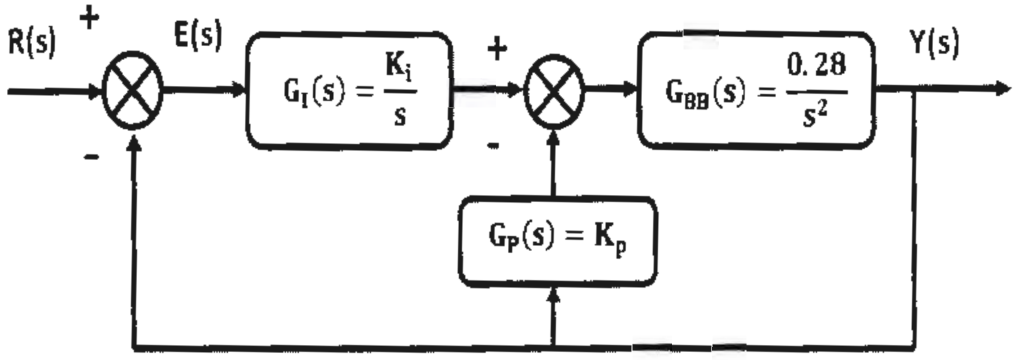

Figure 3.7: Block diagram of Integral-Proportional (I-P) controller

After applying blocks diagram reduction techniques on Figure 3.7, simplified block diagram can be represented as Figure 3.8.

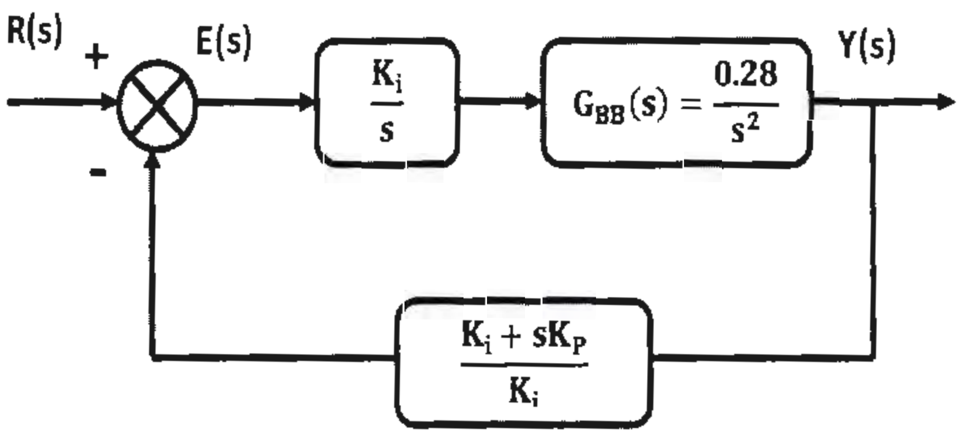

Figure 3.8: Simplified block diagram of Integral-Proportional (I-P) controller

The expression for the closed-loop transfer function can be written as:

$$\frac{Y(s)}{R(s)} = \frac{[(K_i/s)G_{BB}(s)]}{1+[(K_i*G_{BB}(s)][K_i+sK_P]/s^2} \tag{3.14}$$

After solving equation 3.14,

$$\frac{Y(s)}{R(s)} = \frac{0.28K_i}{s^3+0.28K_p s+0.28K_i} \tag{3.15}$$

3.3.2 Proportional Integral-Derivative (PI-D) controller

Proportional Integral-Derivative (PI-D) controller is a controlling scheme in which D controller is connected in feedback path whereas PI controller in connected in feed-forward path. Figure 3.9 represents the PI-D controller's block diagram.

where

$G_{PI}(s)$ is transfer function of PI controller

$G_D(s)$ is the transfer function of D controller

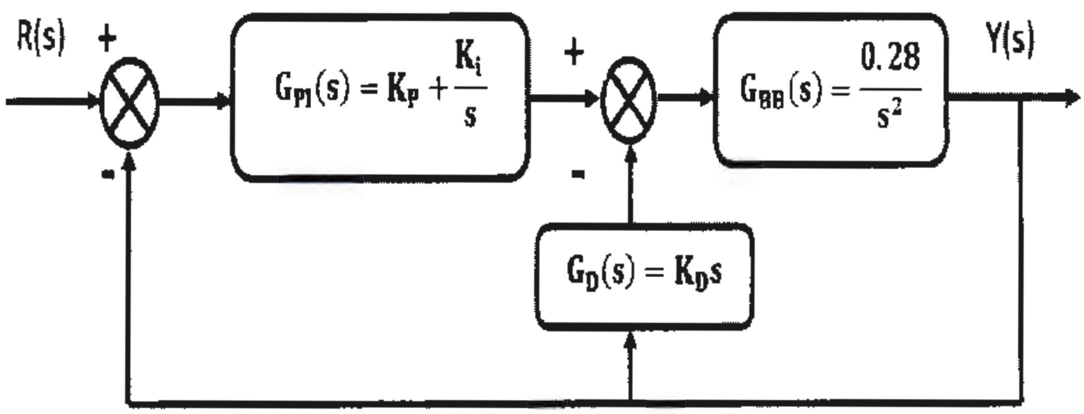

Figure 3.9: Block diagram of Proportional Integral-Derivative (PI-D) controller

After applying blocks diagram reduction techniques on Figure 3.9, simplified block diagram can be represented as Figure 3.10.

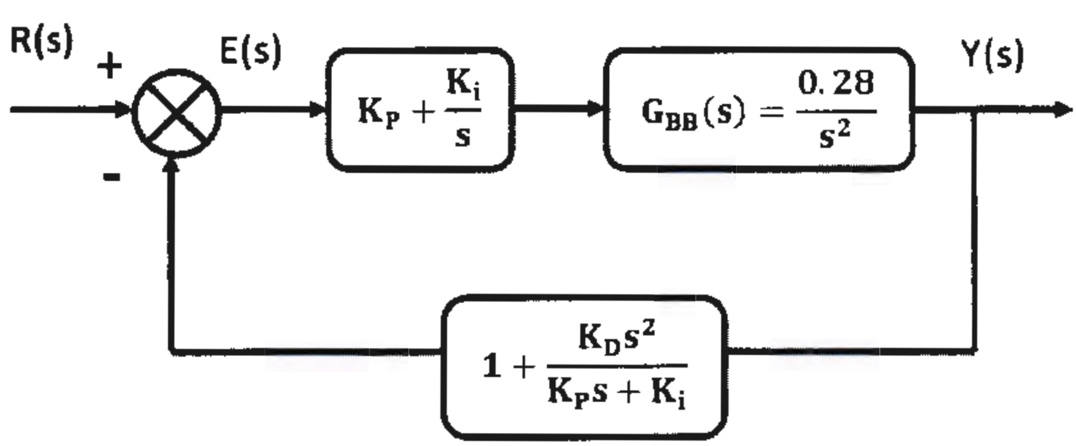

Figure 3.10: Simplified block diagram of Proportional Integral-Derivative (PI-D) controller

The expression for the closed-loop transfer function can be written as:

$$\frac{Y(s)}{R(s)}=\frac{\left(\frac{sK_P+K_i}{s}\right)\left(\frac{0.28}{s^2}\right)}{1+\left(\frac{sK_P+K_i}{s}\right)\left(\frac{0.28}{s^2}\right)\left(1+\frac{K_Ds^2}{K_Ps+K_i}\right)} \quad (3.16)$$

After solving equation 3.16,

$$\frac{Y(s)}{R(s)} = \frac{0.28K_P s+0.28K_i}{s^3+0.28K_D s^2+0.28K_p s+0.28K_i} \tag{3.17}$$

3.3.3 Proportional Integral-Proportional Derivative (PI-PD) controller

Proportional Integral-Proportional Derivative (PI-PD) controller is a controlling scheme in which PD controller appears in feedback path whereas PI controller is connected in the feed-forward path [32]. Figure 3.11 represents the block diagram of PI-PD controller with ball and beam system.

where

$G_{PD}(s)$ is the transfer function of PD controller

$G_{PI}(s)$ is the transfer function f PI controller

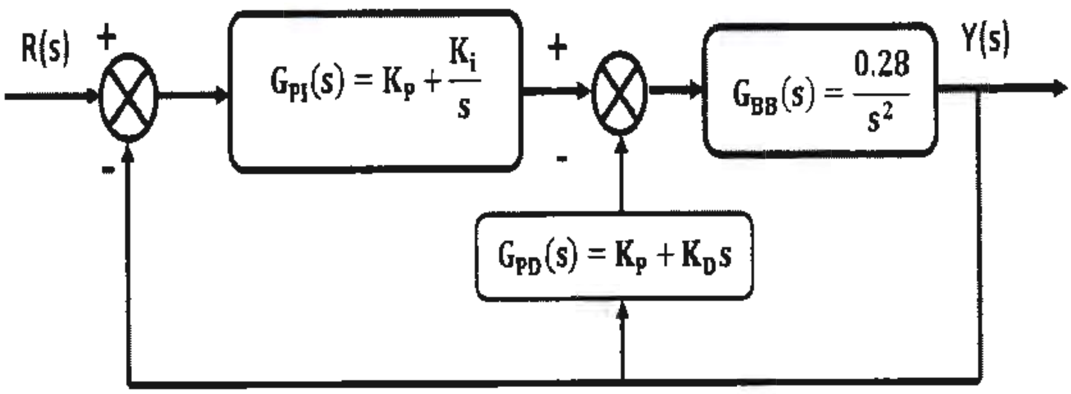

Figure 3.11: Block diagram of Proportional Integral-Proportional Derivative (PI-PD) controller

After applying blocks diagram reduction techniques on Figure 3.11, simplified block diagram can be represented as Figure 3.12.

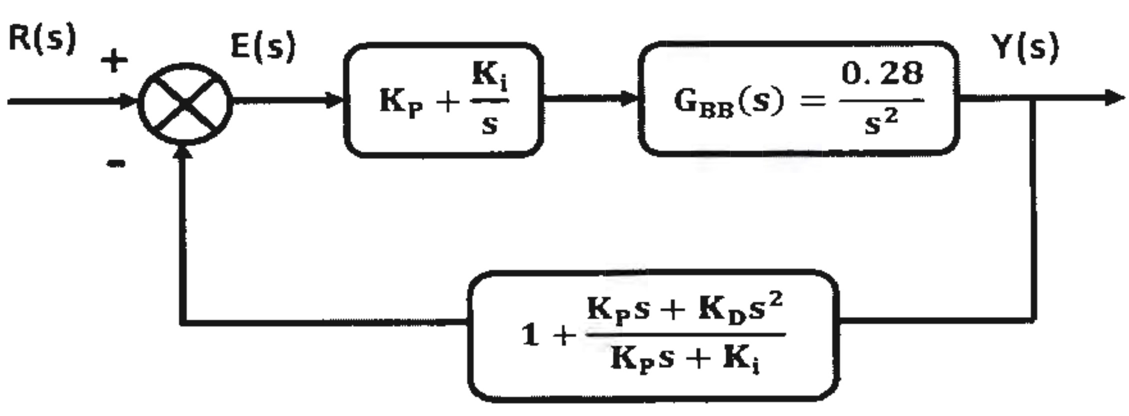

Figure 3.12: Simplified block diagram of Proportional Integral-Proportional Derivative (PI-PD) controller

The expression for the closed-loop transfer function can be written as:

$$\frac{Y(s)}{R(s)} = \frac{\left(\frac{sK_P+K_i}{s}\right)\left(\frac{0.28}{s^2}\right)}{1+\left(\frac{sK_P+K_i}{s}\right)\left(\frac{0.28}{s^2}\right)\left(1+\frac{K_P s+K_D s^2}{K_P s+K_i}\right)} \tag{3.18}$$

After solving equation 3.18,

$$\frac{Y(s)}{R(s)} = \frac{0.28K_P s+0.28K_i}{s^3+0.28K_D s^2+0.56K_p s+0.28K_i} \tag{3.19}$$

3.4 Characteristics of P, I and D controller

Proportional, Integral and Derivative controllers are fundamental blocks of the proposed controllers as described above. The characteristics of these controllers have been tabulated in Table 3.2.

Table 3.2: Characteristics of P, I and D controller

Controller	Settling time	Rise time	Steady state error	% Overshoot
[illegible]	Small Change	Decrease	Decrease	Increase
[illegible]	Increase	Decrease	Eliminate	Increase
[illegible]	Decrease	Small Change	Small Change	Decrease

CHAPTER 4

Tuning Techniques

4.1 Tuning

The process of acquiring the optimum parameters of a controller to achieve the desired control response is known as tuning. There are many types of tuning techniques such as Cohen-cool method, Ziegler Nicholas method, modified Ziegler Nicholas method, Manual tuning method, Skogestad's tuning method and evolutionary computational techniques. Evolutionary computational techniques have also been explored for the tuning of different controllers such as PID, I-PD, PI and IP etc. It has been observed that these evolutionary techniques have given satisfactory results in terms of steady state and transient response.

4.2 Evolutionary Computational Techniques

Evolutionary computational techniques can be used for the tuning of a controller. These techniques have proved their excellence in recent past by solving different types of engineering optimization problems. There are various evolutionary techniques such as Simulated Annealing (SA), Cuckoo Search Algorithm (CSA), Particle Swam Optimization (PSO), Differential Evolution (DE) and Genetic Algorithm (GA) etc. In this thesis, PI-D and PI-PD controller have been implemented for the control of ball and beam system. To achieve required performance specifications, we have to choose optimum controller's parameters (K_P, K_i, K_D). In evolutionary computation, a fitness function is obtained for each controller and it is minimized. For this purpose, different solution sets are tested. The solution set which minimizes the fitness function is considered as optimum solution set and it is applied to the controller. Figure 4.1 represents the block diagram for the controller's tuning with evolutionary computational techniques.

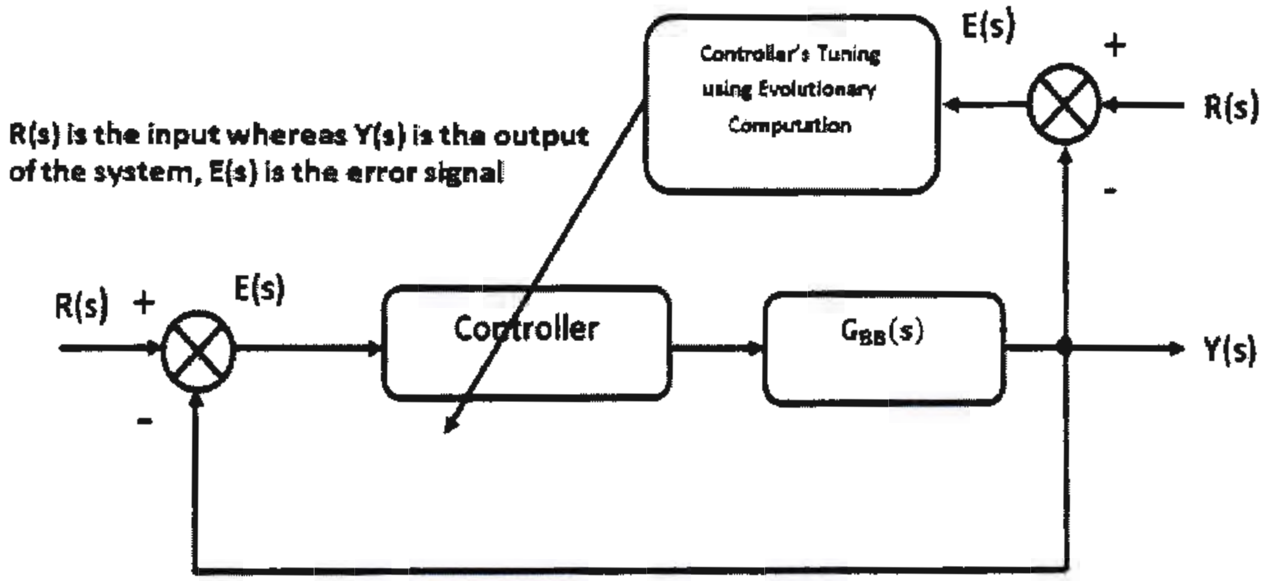

Figure 4.1: Tuning of a controller using Evolutionary Computation Techniques

In this thesis, Simulated Annealing (SA), Genetic Algorithm (GA) and Cuckoo Search Algorithm (CSA) have been utilized for the tuning of proposed controllers.

4.2.1 Cuckoo Search Algorithm (CSA)

Cuckoo Search Algorithm CSA is an evolutionary computational algorithm which was invented in 2009 by Deb and Yang [33]. It is one of the Swarm Intelligence based algorithm. According to this algorithm, each egg available in the nest is considered as a solution whereas a cuckoo's egg corresponds to a new solution [34-35].CSA has been successfully utilized in the past to solve different engineering problems [36-41]. Figure 4.2 represents the flow chart diagram of Cuckoo Search Algorithm (CSA).

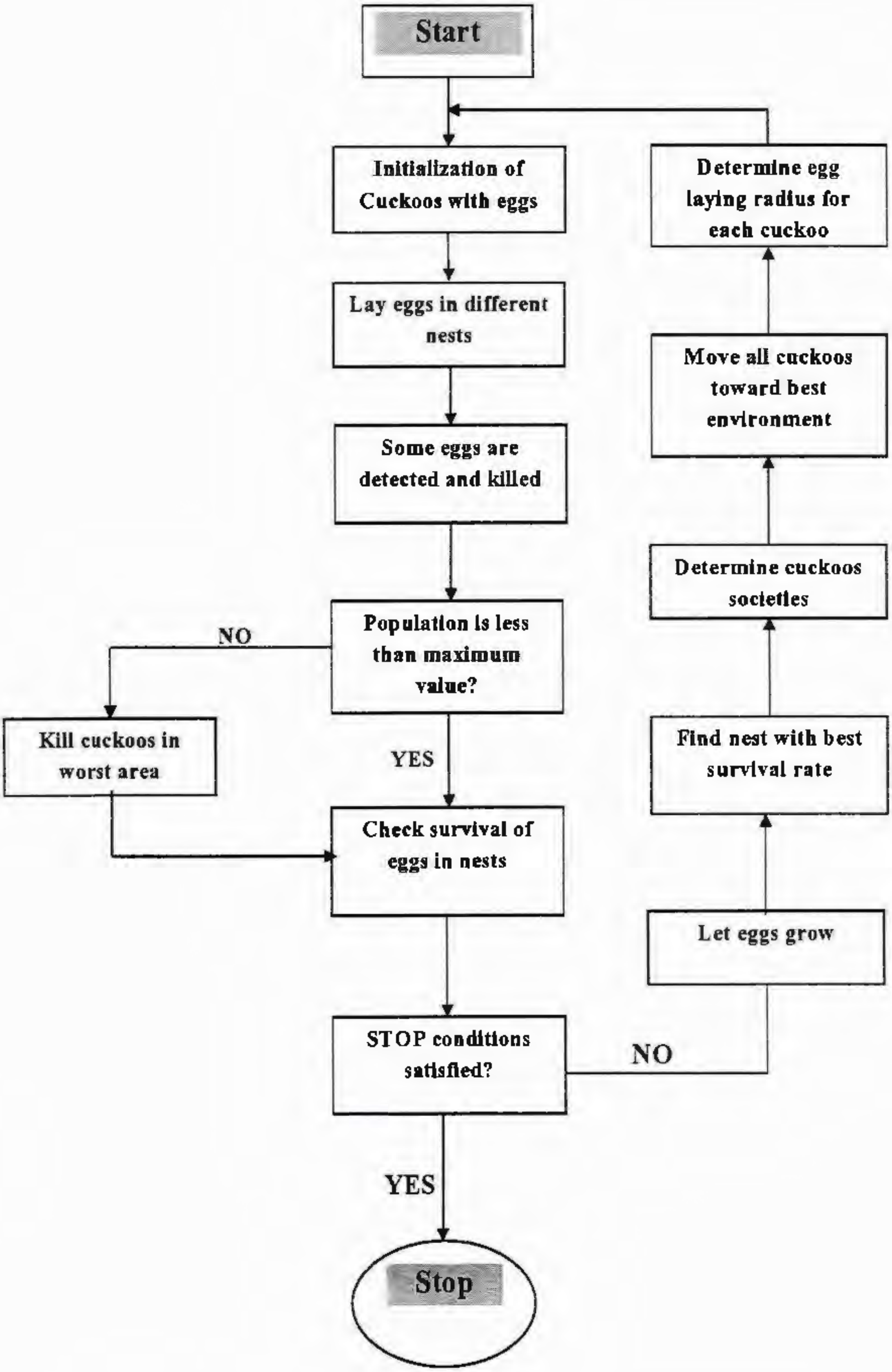

Figure 4.2: Flow chart of Cuckoo Search Algorithm (CSA)

Following are the main steps involved in the Cuckoo Search Algorithm (CSA).

1. First of all, cuckoos are initialized with eggs. Remember that these eggs are representing the solution set.
2. After this, cuckoos lay eggs in different nests. Some of the eggs are detected and killed.
3. If the eggs population is less than the maximum value, then check out the survival of the eggs in nests. Otherwise, kill cuckoos in worst area and then check survival of eggs.
4. If the stopping criterion has been achieved, then save the best solution set (best eggs) and stop the algorithm. If stopping criterion hasn't been satisfied, let the eggs grow up and follow the step 5 & step 6.
5. Find out the nests having best survival rate and then determine cuckoos societies.
6. Move all the cuckoos towards best atmosphere and find out the egg laying radius for each cuckoo.
7. Repeat the first four steps (1-4) of the algorithm as given above.

4.2.2 Genetic Algorithm (GA)

Genetic Algorithm (GA) is a nature inspired heuristic algorithm based upon Darwin's theory [42]. It was invented by John Holland in the early 1970's. It has been used for the optimization of different engineering problems [43-51]. GA is a stochastic algorithm and has been provoked by evolutionary genetics [52]. In this technique, we create random chromosomes (solutions) to find out resulting error. The chromosome which will give the minimum error is declared as optimum. The basic idea is to create new random generations till you will find out best solution. Figure 4.3 shows the flow chart of Genetic Algorithm (GA).

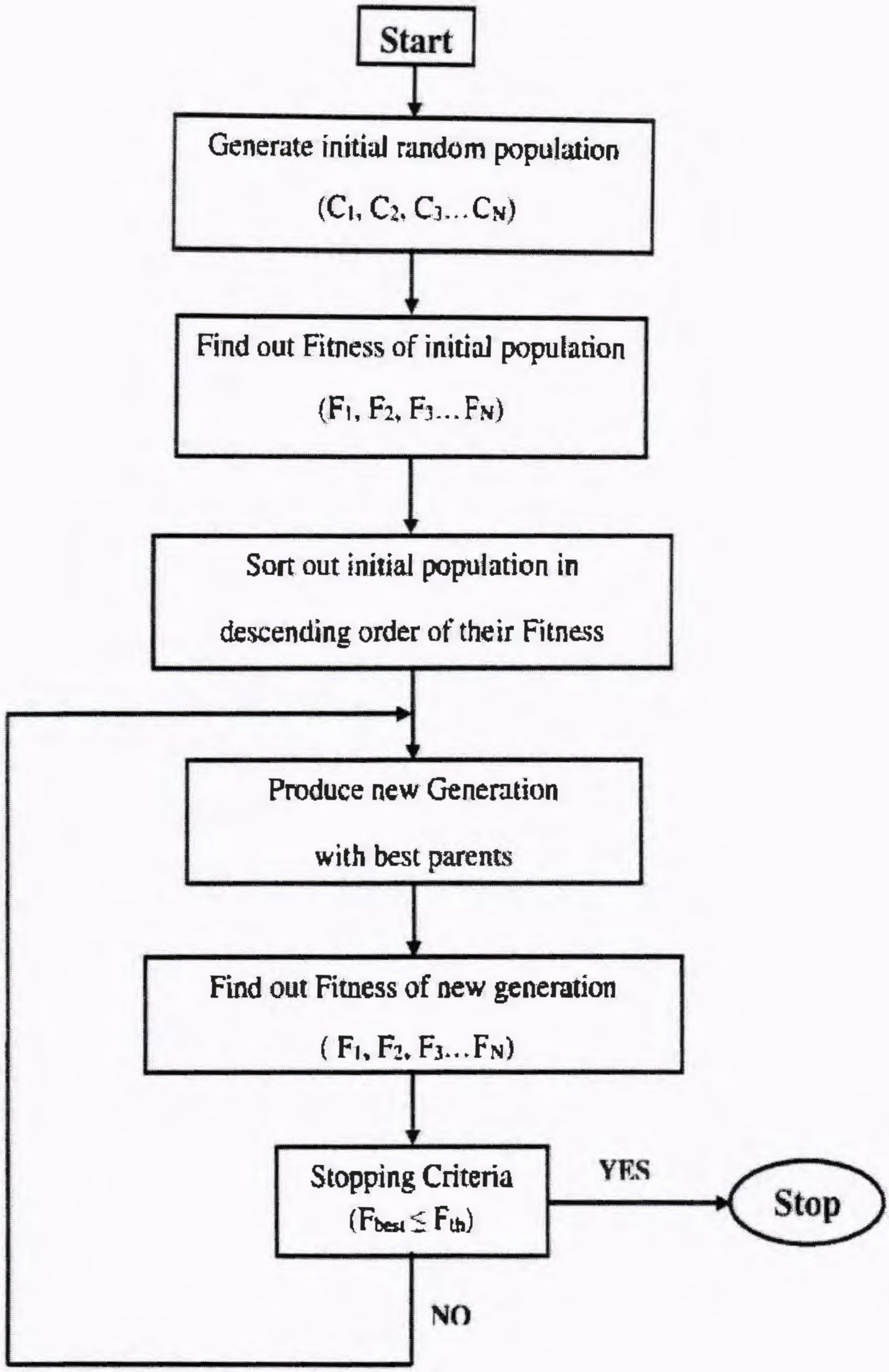

Figure 4.3: Flow Chart of Genetic Algorithm (GA)

Genetic Algorithm (GA) comprises of following major steps.

1. In GA, first of all we create N number of random chromosomes ($C_1, C_2, C_3 \ldots\ldots C_N$) by using some random function.
2. After creation of chromosomes, we find out fitness of corresponding chromosomes ($F_1, F_2, F_3 \ldots\ldots F_N$).
3. Sort out all chromosomes of initial population in descending order of their fitness to move F_{best} at the top.
4. If $F_{best} \leq F_{th}$, it means that we have found the best chromosome. Now save the chromosome and its fitness. Stop the algorithm.
5. If stopping criteria does not meet then produce new generation by using initial population.
6. Do the mutation if required (means that if the fitness is not improving).
7. Repeat step 2 and 3. If best fitness of new generation is equal to or less than the threshold value, save the best chromosome stats and terminate the process.
8. Else repeat the step 5, 6 and 7.

4.2.3 Simulated Annealing (SA)

Simulated Annealing (SA) is a heuristic algorithm which was invented in 1983 by Kirkpatrick et al. [53]. Kirkpatrick explore the idea of Metropolis (1953). Simulated Annealing (SA) for the physical annealing process of a solid and optimization problem are similar to each other. The minimum energy of the solid material is equivalent to the minimum value of cost (objective) function. SA is a global optimization technique which can easily distinguishes between different local minima. This is a very beneficial property for the researcher to optimize their problems with global minimum. SA has been successfully used to solve different problems [54-59]. Figure 4.4 represents the flow chart diagram of SA algorithm.

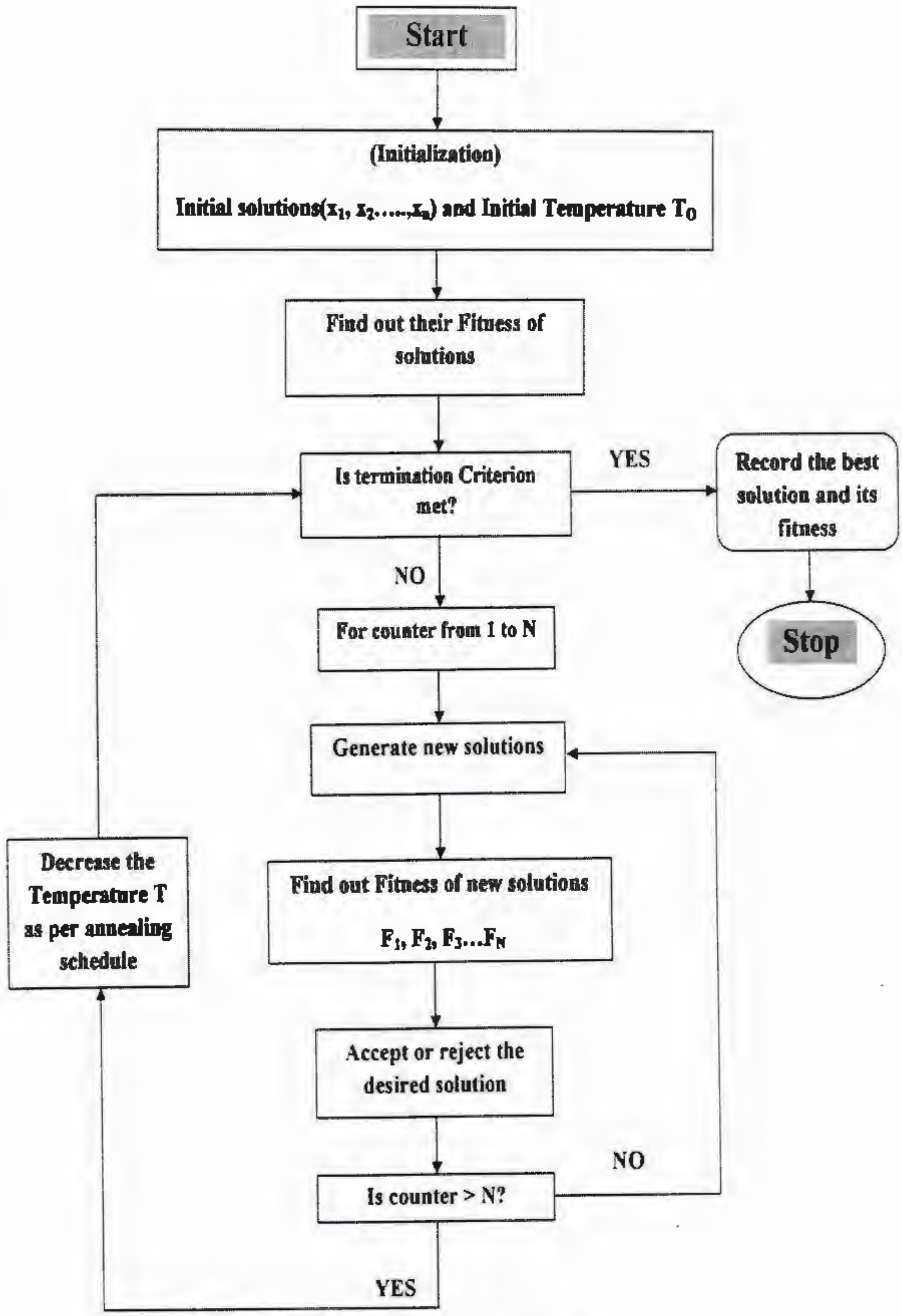

Figure 4.4: Flow chart of Simulated Annealing (SA)

The major steps involved in Simulated Annealing (SA) are given below.

1. Create the random initial population (x_1, x_2,……x_n) and define an initial temperature T (where n represent the length of solution set).
2. Find out the fitness of initial population (F_1, F_2,…..F_n).
3. If the termination criteria met, record the best solution with its fitness and stop the algorithm. If the termination criteria don't meet, then follow the next steps.
4. Start a counter from **1:M** and randomly generate the new solutions. Find out their fitness (where M is any positive number). Then decrease the temperature T as per annealing schedule and monitor the solutions.
5. If the termination criteria met, record the best solution with its fitness and stop the algorithm. If the termination criteria do not satisfy, then follow the steps 4.

CHAPTER 5

Formulations for the Fitness Functions of Controllers

5.1 Fitness Function

Optimum controller's parameters (K_P, K_i, K_D) can be obtained by using evolutionary computational techniques. For this purpose, fitness function for a particular controller is derived mathematically. The word fitness is associated with evolutionary theory and it tells how much the solution is better one. A fitness function calculates the fitness of a particular solution. Fitness function returns the optimum parameters of the controller with minimum error (best fitness). Fitness function is tested by using different set of solutions. These solutions are generated randomly by evolutionary computation techniques such as SA, CSA and GA. After acquiring optimum solution, evolutionary algorithm is stopped and solution is recorded.

5.2 Derivation of Error for PD controller

Figure 5.1 represents the PD controller applied to the proposed ball and beam system. Fitness function for the PD controller can be obtained as given below.

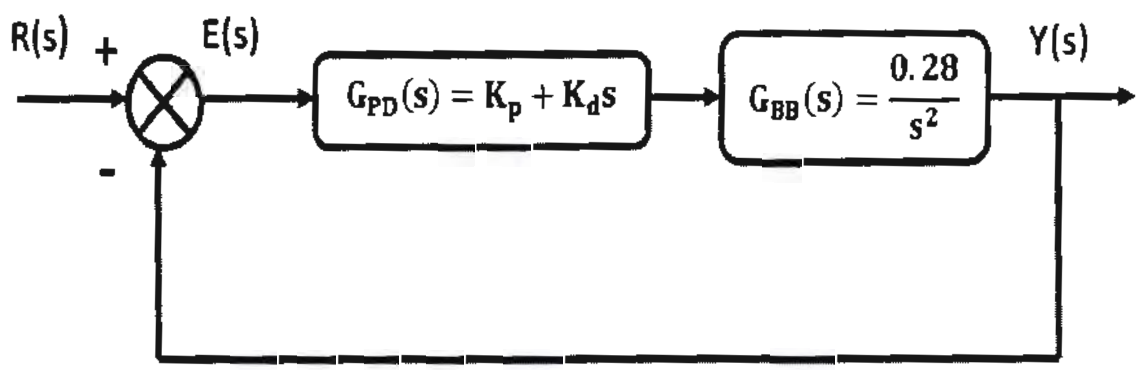

Figure 5.1: PD controller applied to Ball and Beam System

Let E(s) is defined as:

$$E(s) = R(s) - Y(s) \tag{5.1}$$

$$R(s) = \frac{1}{s} \tag{5.2}$$

$$\frac{Y(s)}{R(s)} = \frac{G_{PD}(s).G_{BB}(s)}{1+G_{PD}(s).G_{BB}(s)} \tag{5.3}$$

By solving equation 5.1, 5.2 and 5.3 we obtain,

$$E(s) = \frac{s}{s^2+0.28K_Ds+0.28K_P} \tag{5.4}$$

By taking the inverse Laplace transform of equation 5.4, e(t) can be written as:

e(t) = exp(-(7*kd*t)/50)*(cosh(t*((49*kd^2)/2500-(7*kp)/25)^(1/2))-(7*kd*sinh(t*((49*kd^2)/2500 - (7*kp)/25)^(1/2)))/(50*((49*kd^2)/2500 - (7*kp)/25)^(1/2)))

The detailed derivation of E(s) is given in Appendix A.

5.3 Derivation of Error for PI controller

Figure 5.2 represents the PI controller applied to the proposed ball and beam system. Fitness function for the PI controller can be obtained as given below.

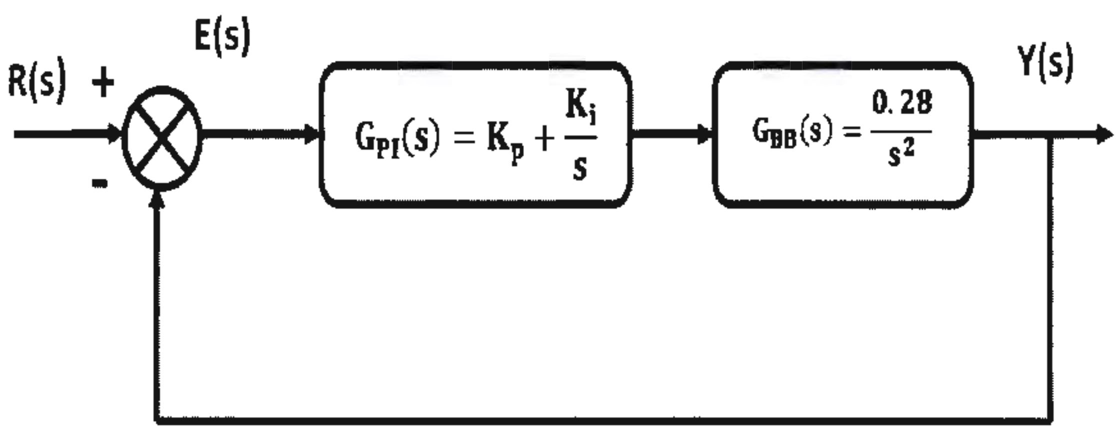

Figure 5.2: PI controller applied to Ball and Beam System

Let E(s) is defined as:

$$E(s) = R(s) - Y(s) \tag{5.5}$$

$$R(s) = \frac{1}{s} \tag{5.6}$$

$$\frac{Y(s)}{R(s)} = \frac{G_{PI}(s).G_{BB}(s)}{1+G_P(s).G_{BB}(s)} \tag{5.7}$$

By solving equation 5.5, 5.6 and 5.7 we obtain,

$$E(s) = \frac{s^2}{s^3 + 0.28K_p s + 0.28K_i} \quad (5.8)$$

By taking the inverse Laplace transform of equation 5.8, e(t) can be written as:

e(t) = 25*sum((r3^2*exp(r3*t))/(75*r3^2 + 7*kp), r3 in RootOf (s3^3 + (7*kp*s3)/25 + (7*ki)/25, s3))

The detailed derivation of E(s) is given in Appendix B.

5.4 Derivation of Error for I-P controller

Figure 5.3 represents the I-P controller applied to the proposed ball and beam system. Fitness function for the I-P controller can be obtained as given below.

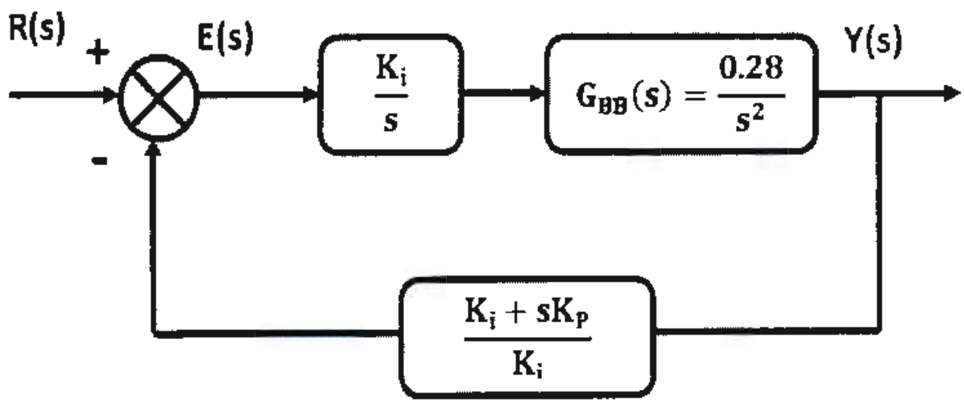

Figure 5.3: I-P controller applied to Ball and Beam System

Let E(s) is defined as:

$$E(s) = R(s) - Y(s) \quad (5.9)$$

$$R(s) = \frac{1}{s} \quad (5.10)$$

$$\frac{Y(s)}{R(s)} = \frac{\left(\frac{K_i}{s}\right)\left(\frac{0.28}{s^2}\right)}{1 + \left[\left(\frac{K_i}{sK_i}\right)\left(\frac{0.28}{s^2}\right)\right][K_i + sK_p]} \quad (5.11)$$

By solving equation 5.9, 5.10 and 5.11 we obtain,

$$E(s) = \frac{s^2 + 0.28K_p}{s^3 + 0.28K_p s + 0.28K_i} \quad (5.12)$$

By taking the inverse Laplace transform of equation 5.12, e(t) can be written as:

e(t) = 7*kp*sum(exp(r3*t)/(75*r3^2 + 7*kp), r3 in RootOf(s3^3 + (7*kp*s3)/25 + (7*ki)/25, s3)) + 25*sum((r3^2*exp(r3*t))/(75*r3^2 + 7*kp), r3 in RootOf(s3^3 + (7*kp*s3)/25 + (7*ki)/25, s3))

The detailed derivation of E(s) is given in Appendix C.

5.5 Derivation of Error for PI-D controller

Figure 5.4 represents the PI-D controller applied to the proposed ball and beam system. Fitness function for the PI-D controller can be obtained as given below.

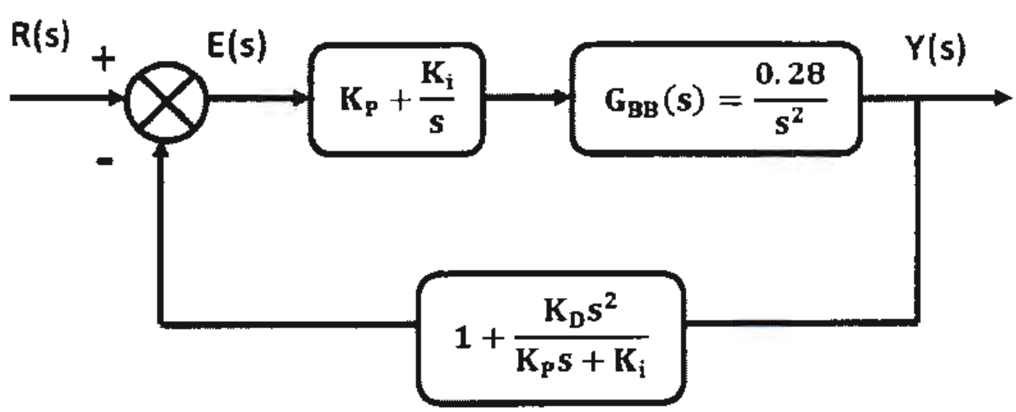

Figure 5.4: PI-D controller applied to Ball and Beam System

Let E(s) is defined as:

$$E(s) = R(s) - Y(s) \tag{5.13}$$

$$R(s) = \frac{1}{s} \tag{5.14}$$

$$\frac{Y(s)}{R(s)} = \frac{\left(\frac{sK_P+K_i}{s}\right)\left(\frac{0.28}{s^2}\right)}{1+\left(\frac{sK_P+K_i}{s}\right)\left(\frac{0.28}{s^2}\right)\left(1+\frac{K_D s^2}{K_P s+K_i}\right)} \tag{5.15}$$

By solving equation 5.13, 5.14 and 5.15 we obtain,

$$E(s) = \frac{s^2+0.28K_D s}{s^3+0.28K_D s^2+0.28K_P s+0.28K_i} \tag{5.16}$$

By taking the inverse Laplace transform of equation 5.16, e(t) can be written as:

e(t) = 7*kd*sum((r3*exp(r3*t))/(75*r3^2 + 14*kd*r3 + 7*kp), r3 in RootOf(s3^3 + (7*kd*s3^2)/25 + (7*kp*s3)/25 + (7*ki)/25, s3)) + 25*sum((r3^2*exp(r3*t))/(75*r3^2 + 14*kd*r3 + 7*kp), r3 in RootOf(s3^3 + (7*kd*s3^2)/25 + (7*kp*s3)/25 + (7*ki)/25, s3))

The detailed derivation of E(s) is given in Appendix D.

5.6 Derivation of Error for PI-PD controller

Figure 5.5 represents the PI-PD controller applied to the proposed ball and beam system. Fitness function for the PI-PD controller can be obtained as given below.

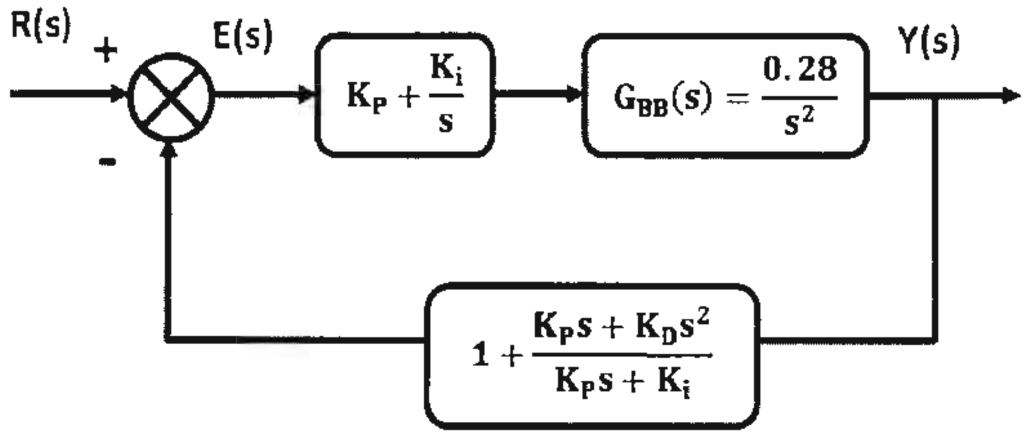

Figure 5.5: PI-PD controller applied to Ball and Beam System

Let E(s) is defined as:

$$E(s) = R(s) - Y(s) \tag{5.17}$$

$$R(s) = \frac{1}{s} \tag{5.18}$$

$$\frac{Y(s)}{R(s)} = \frac{\left(\frac{sK_P+K_i}{s}\right)\left(\frac{0.28}{s^2}\right)}{1+\left(\frac{sK_P+K_i}{s}\right)\left(\frac{0.28}{s^2}\right)\left(1+\frac{K_P s+K_D s^2}{K_P s+K_i}\right)} \tag{5.19}$$

By solving equation 5.17, 5.18 and 5.19 we obtain,

$$E(s) = \frac{s^2+0.28K_D s+0.28K_p}{s^3+0.28K_D s^2+0.56K_p s+0.28K_i} \tag{5.20}$$

By taking the inverse Laplace transform of equation 5.20, e(t) can be written as:

e(t) = 7*kd*sum((r3*exp(r3*t))/(75*r3^2 + 14*kd*r3 + 14*kp), r3 in RootOf(s3^3 + (7*kd*s3^2)/25 + (14*kp*s3)/25 + (7*ki)/25, s3)) + 7*kp*sum(exp(r3*t)/(75*r3^2 + 14*kd*r3 + 14*kp), r3 in RootOf(s3^3 + (7*kd*s3^2)/25 + (14*kp*s3)/25 + (7*ki)/25, s3)) + 25*sum((r3^2*exp(r3*t))/(75*r3^2 + 14*kd*r3 + 14*kp), r3 in RootOf(s3^3 + (7*kd*s3^2)/25 + (14*kp*s3)/25 + (7*ki)/25, s3))

The detailed derivation of E(s) is given in Appendix E

5.7 Performance Indices

To achieve an optimum response of the system, we have minimized the time domain based performance indices which are related to the error in the system. The minimization of error is done by using different tuning techniques as discussed in Chapter 4. A performance index measures the system's performance which is associated with different parameters like and steady state error (e_{ss}), settling time (t_s), overshoot (os) and rise time (t_r) [60].

The performance indices which have been used in this research work are ISE, IAE, ITAE and ITSE. Theses performance indices are calculated over some defined time interval T. The time (T) is chosen in such a way that it will cover much of the transient response of the system. If a system has a response similar to 2nd order system, then best choice of T will be t_s (settling time). The description of each performance index is given below.

5.7.1 Integral of squared error (ISE)

Integral of squared error (ISE) criterion is a very useful performance index, which can be calculated as,

$$\mathbf{ISE} = \int_0^T \mathbf{e^2(t)dt} \tag{5.21}$$

where e(t) is given in previous section.

Due to square of the error function, both negative and positive values of the error are penalized. In 2nd order systems, ISE will have a minimum value when damping ratio will be approximately 0.5. ISE is not sensitive to parameter variations. In addition, the ISE criterion is has an advantage of being easy to deal with mathematically.

5.7.2 Integral of absolute value of error (IAE)

A fairly useful performance index is the Integral of absolute value of error (IAE) criterion, which is given as,

$$\mathbf{IAE} = \int_0^T \mathbf{|e(t)|dt} \tag{5.22}$$

Due to the magnitude of the error, this integral expression will be increased for both negative and positive values of the error but it exhibits a very good under-damped response. It has poor sensitivity but slightly better than ISE criterion. In 2nd order systems, IAE will have a minimum value when damping ratio will be approximately 0.7.

5.7.3 Integral of time multiplied by absolute value of error (ITAE)

Integral of time multiplied by absolute value of error (ITAE) is an error criterion in which long-duration transients are penalized. It can be calculated as,

$$\mathbf{ITAE} = \int_0^T t|e(t)|dt \tag{5.23}$$

ITAE is relatively more selective than ISE or IAE. ITAE criterion generally gives minor oscillations and overshoots as compared to IAE or ISE.

5.7.4 Integral of time multiplied by squared value of error (ITSE)

Integral of time multiplied by squared value of error (ITSE) can be written as,

$$\mathbf{ITSE} = \int_0^T te^2(t)dt \tag{5.24}$$

This is generally desirable to the step response patterns when used as a basis for optimum design. It gives a transient response in which large initial errors are weighted lightly whereas late occurring errors are penalized heavily.

Equations 5.21-5.24 have been solved using evolutionary computation such as Cuckoo Search Algorithm (CSA), Genetic Algorithm (GA) and Simulated Annealing (SA) to find out the controller coefficients corresponding to the minimum error function.

CHAPTER 6

Simulation and Results

In this chapter, step responses and set point tracking with proposed controllers has been achieved. For simulations purpose, MATLAB/Simulink software has been utilized. In all simulations unit step input is taken as reference position. The system response has been investigated with each controller by following the reference input signal.

6.1 Open-Loop response of Ball and Beam System

Figure 6.1 represents the schematic diagram for the open loop step response of ball and beam system.

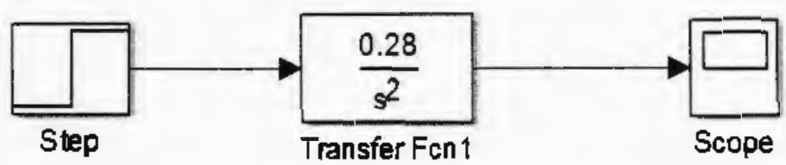

Figure 6.1: Schematic diagram for the open loop step response of Ball & Beam System

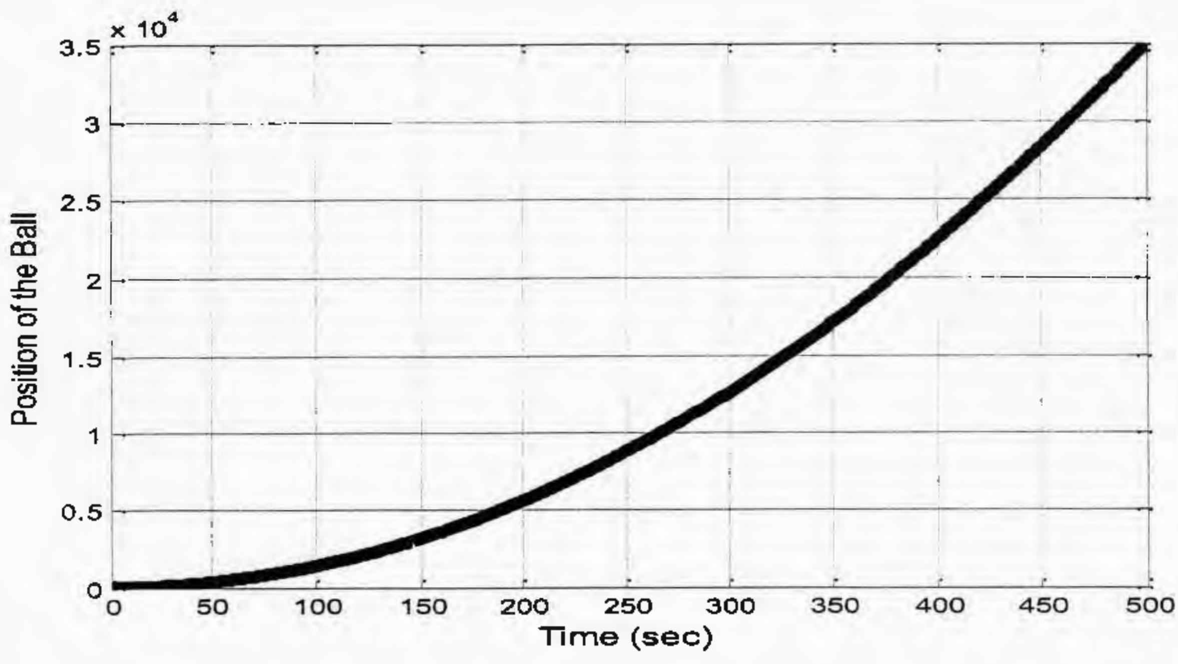

Figure 6.2: Open Loop step Response of Ball & Beam System

Figure 6.2 shows the open loop step response of the ball and beam system. It can be observed that the response is growing with time i.e. an unstable response. It can be suggested that this system needs some control mechanism to behave as a stable system.

6.2 Step Response of I-P and PI controller

In this section, step responses of PI and I-P controller have been provided. Figure 6.3 represents the schematic diagram for step response of ball and beam system with I-P controller whereas Figure 6.4 represents the step response of the system achieved with the I-P controller.

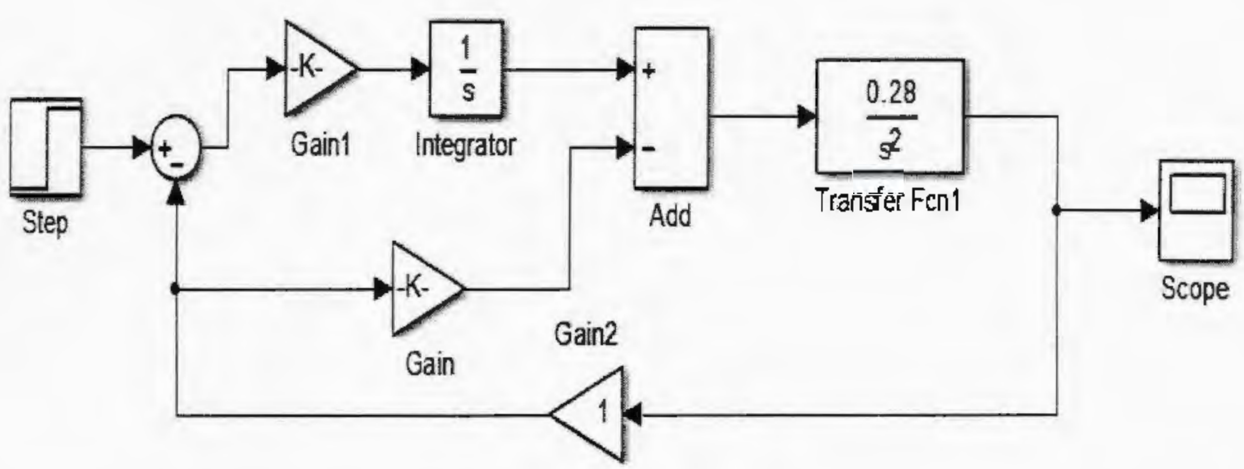

Figure 6.3: Schematic diagram for step response of Ball and Beam System with I-P controller

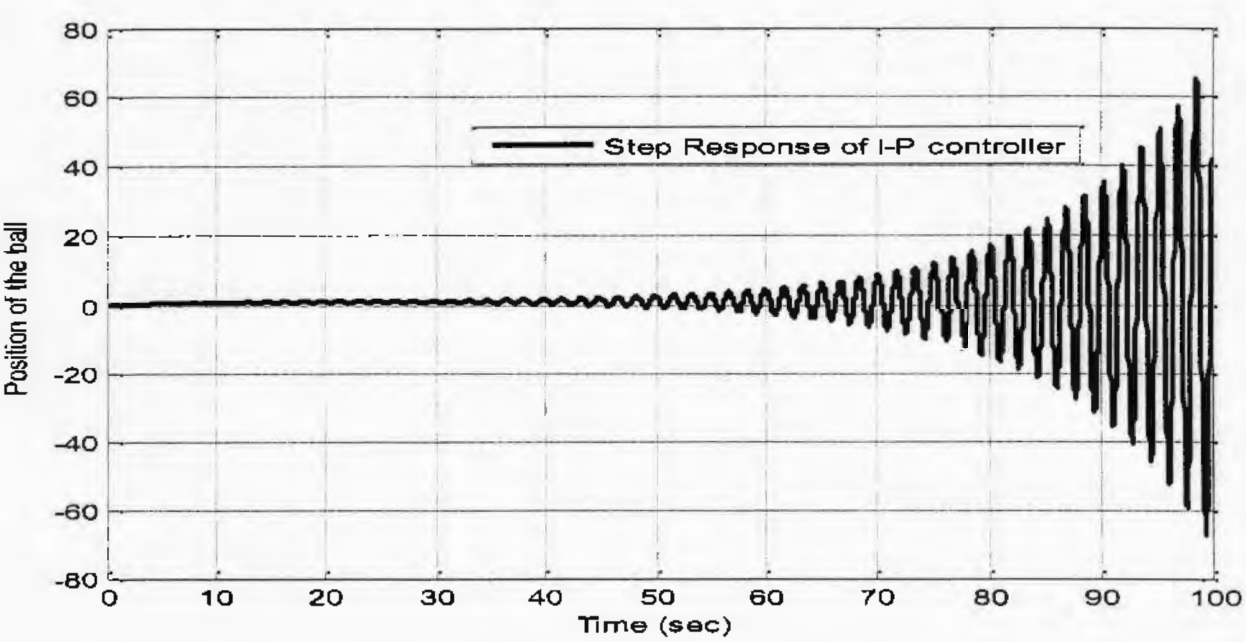

Figure 6.4: Step response of I-P controller

From the results of Figure 6.4, it can be observed that I-P controller yields an unstable response. It can be suggested that I-P controller could not be used for the position control of ball and beam system used in this research. Figure 6.5 represents the schematic diagram for step response of ball and beam system with PI controller whereas Figure 6.6 represents the step response of PI controller.

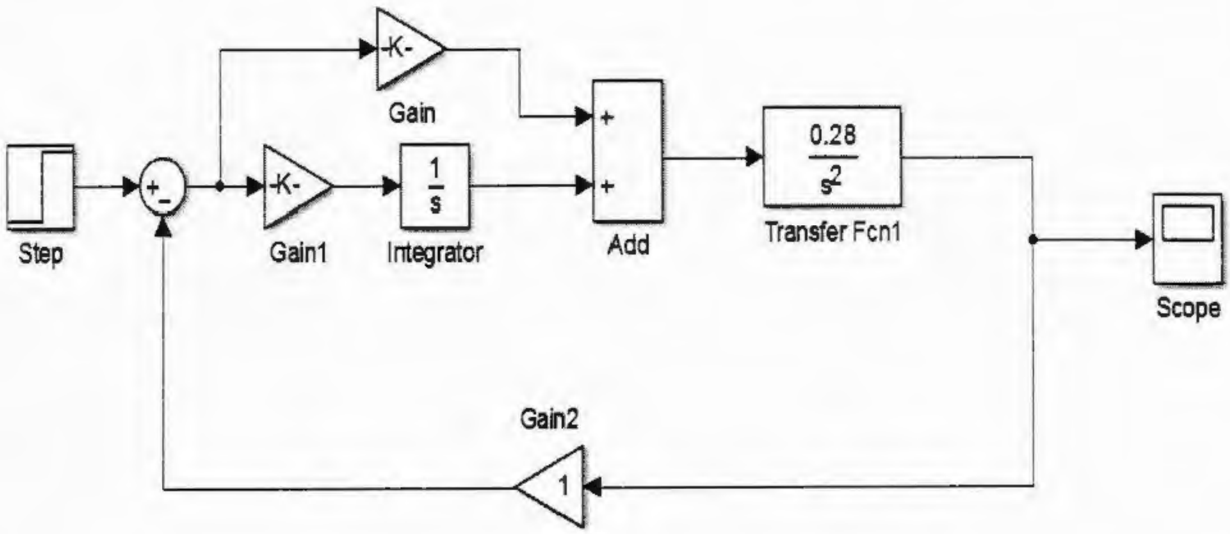

Figure 6.5: Schematic diagram for step response of ball and beam system with PI controller

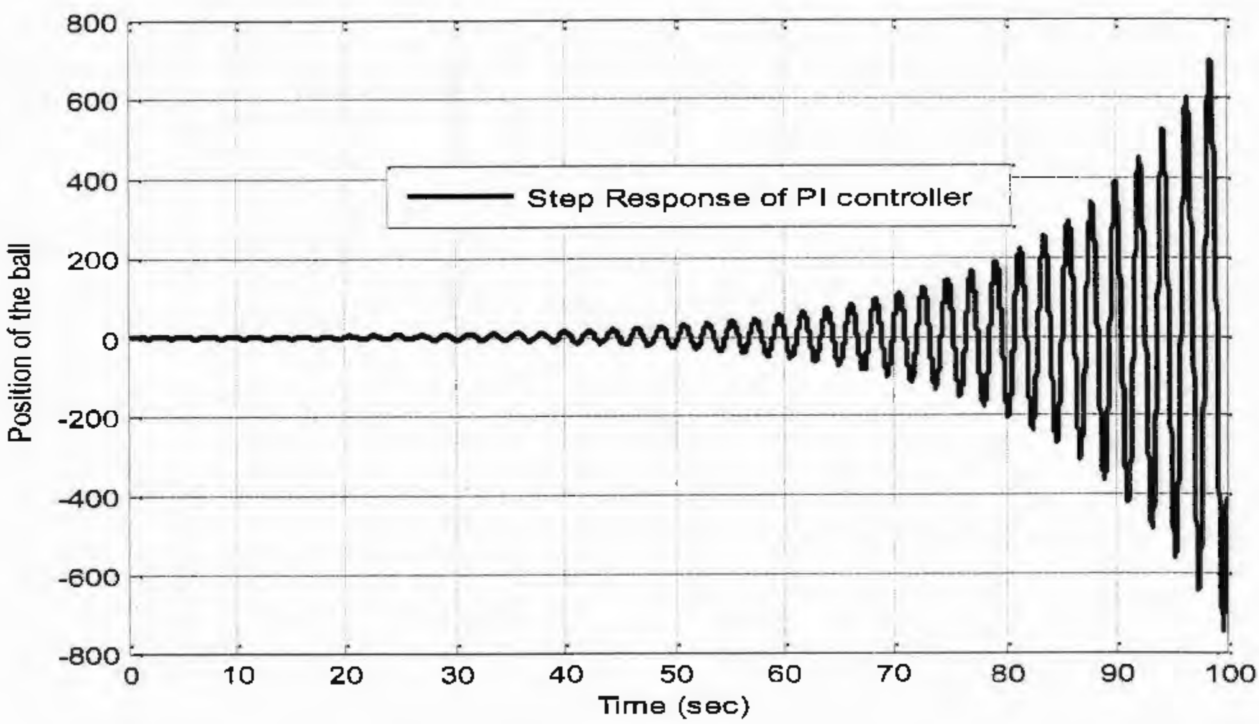

Figure 6.6: Step response of PI controller

From the results of Figure 6.6, it can be observed that PI controller also exhibits an unstable response. That's why PI controller could not be used for the stability control of ball and beam system.

6.3 Step Response of PD controller tuned with SA and CSA

In this section, PD controller with Simulated Annealing (SA) and Cuckoo Search Algorithm (CSA) has been implemented. Optimum parameters of the PD controller have been obtained successfully by using SA and CSA. MATLAB settings for the proposed tuning techniques have been summarized in Table 6.1 and Table 6.2.

Table 6.1: MATLAB OPTIMTOOL settings for Simulated Annealing (SA)

Parameters	Annealing Function	Temperature Update Function	Initial Temperature	Start Point	Lower bound/Upper bounds
ISE	Fast Annealing	Exponential Temperature Update	100	[5 3]	[5 3] / [50 30]
ITSE	Fast Annealing	Exponential Temperature Update	100	[5 3]	[5 3] / [50 30]
IAE	Fast Annealing	Exponential Temperature Update	100	[5 3]	[5 3] / [50 30]
ITAE	Fast Annealing	Exponential Temperature Update	100	[5 3]	[5 3] / [50 30]

Table 6.2: MATLAB settings for Cuckoo Search Algorithm (CSA)

[illegible]	[illegible]	Discovery rate of alien eggs	[illegible]	Upper bounds
[illegible]	25	0.25	3	60
[illegible]	25	0.25	3	60
IAE	25	0.25	3	60
ITAE	25	0.25	3	60

Tuning parameters of PD controller with SA and CSA have been provided in the Table 6.3. Figure 6.7 represents the schematic diagram for step response of ball and beam system with PD controller.

Table 6.3: Parameters of PD controller using SA and CSA

[illegible]	Tuning with SA		Tuning with CSA	
	K_p	K_d	K_p	K_d
[illegible]	49.99	21.82	36.45	20.73
[illegible]	36.88	20	48.7304	49.4589
[illegible]	49.91	30	42.1214	45.0019
[illegible]	50	29.95	58.5385	39.7783

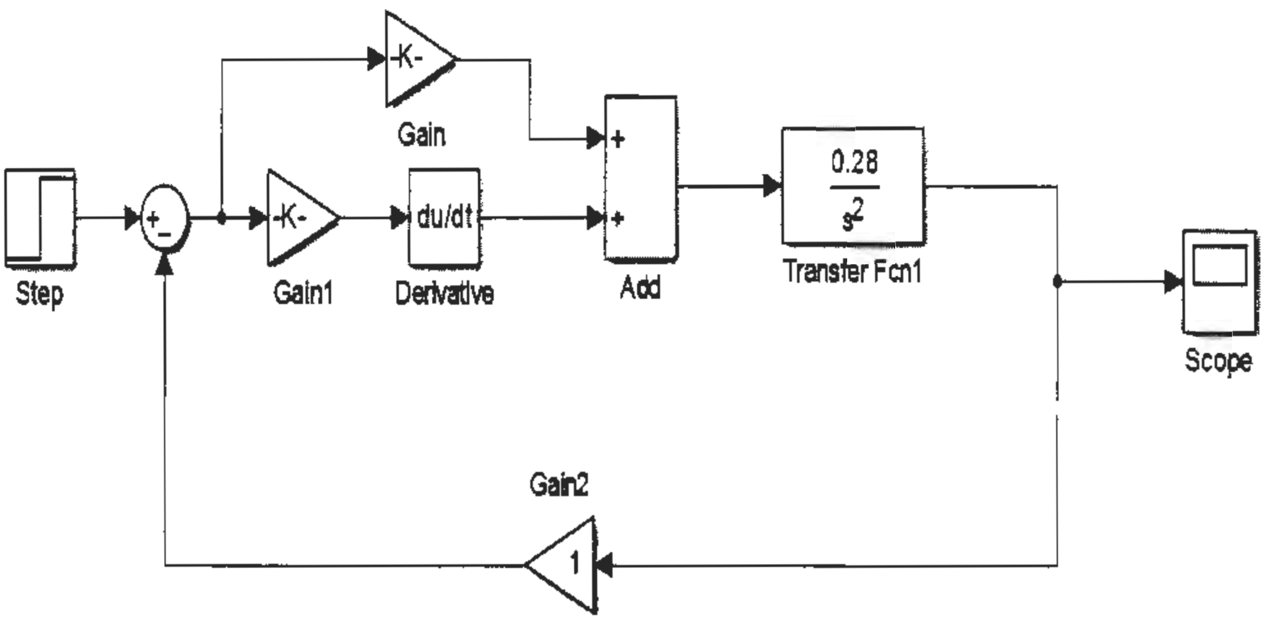

Figure 6.7: Schematic diagram for step response of Ball and Beam System with PD controller

Step responses of PD controller tuned with SA and CSA are given in Figure 6.8 and Figure 6.9 respectively.

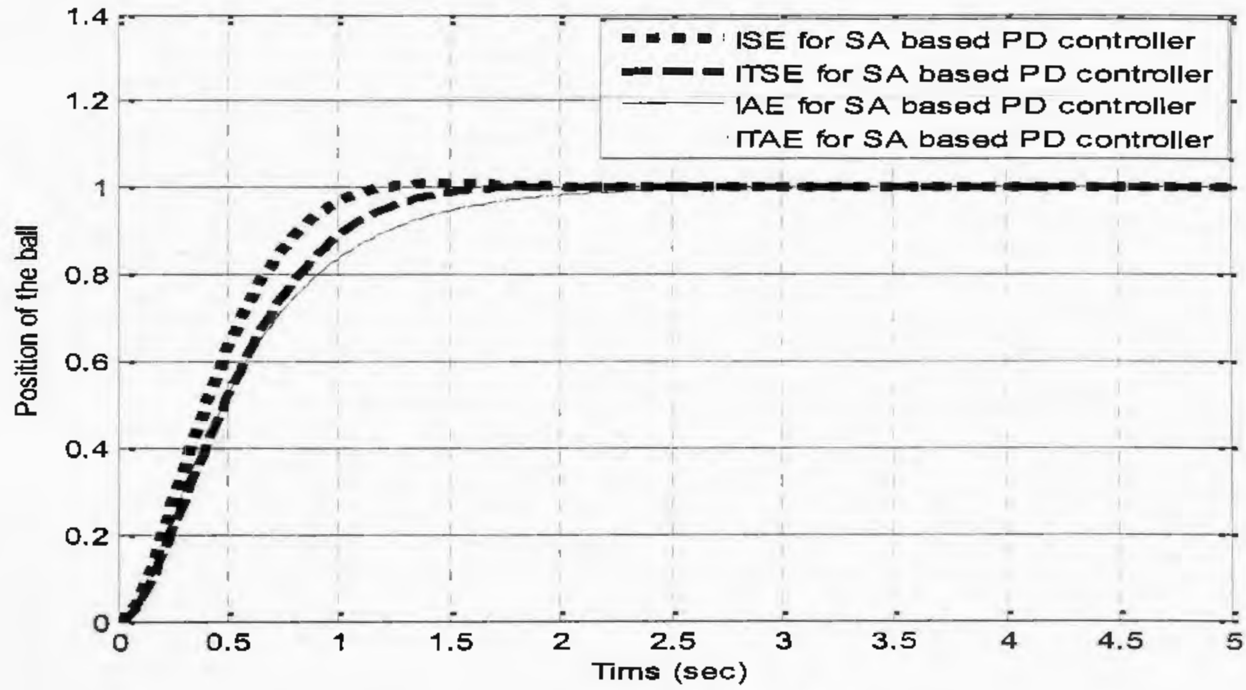

Figure 6.8: Step response of SA based PD controller

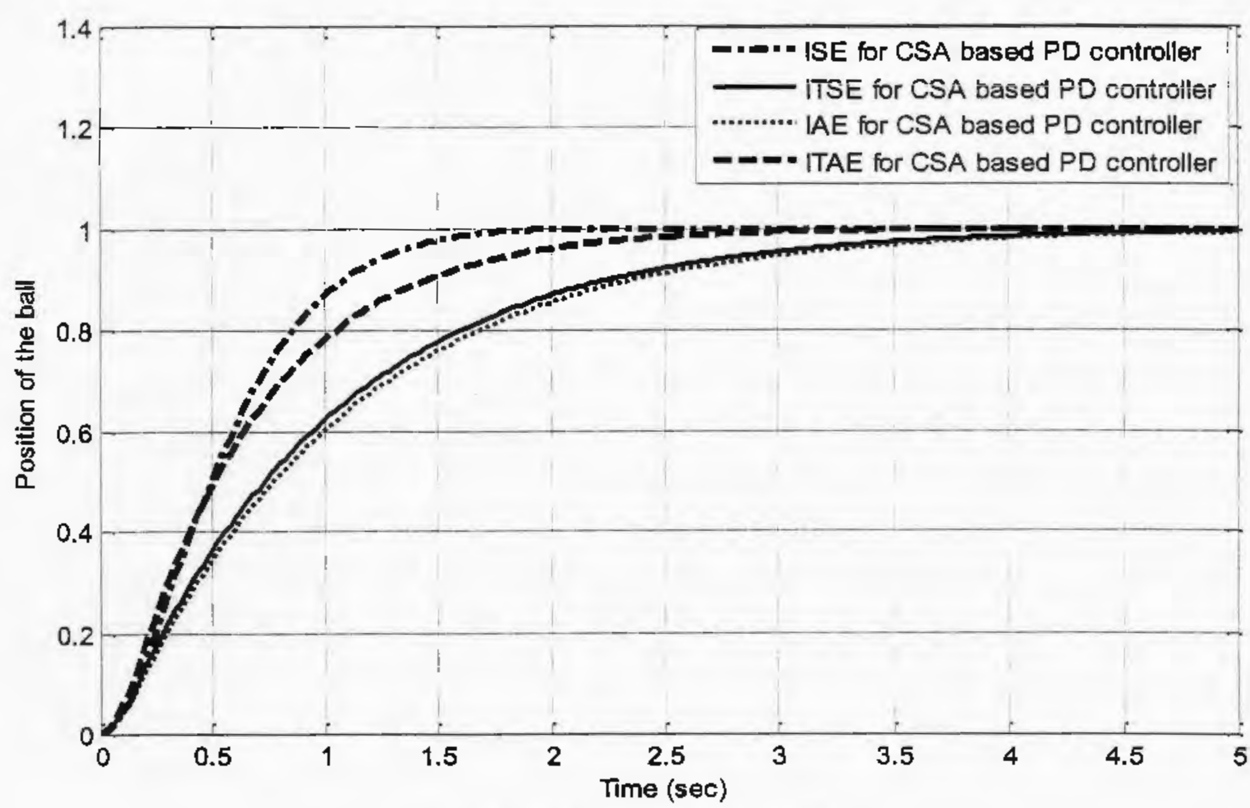

Figure 6.9: Step response of CSA based PD controller

In Table 6.4 and Table 6.5, the corresponding transient responses of SA based PD (SA-PD) and CSA based PD (CSA-PD) controller have been summarized respectively.

Table 6.4: Performance comparison of SA based PD controller

Performance Parameter / Performance Index	rise time (sec)	settling time (sec)	% overshoot	s-s error
ISE	0.6748	1.0563	0.9053	0
ITSE	0.8614	1.3937	0.2456	0
IAE	1.0746	1.9482	4.4409e-14	0
ITAE	1.0696	1.9387	4.4409e-14	0

Table 6.5: Performance comparison of CSA based PD controller

Performance Parameter / Performance Index	rise time (sec)	settling time (sec)	% overshoot	s-s error
ISE	0.9193	1.5300	0.0489	0
ITSE	2.0812	3.7706	0	0
IAE	2.1827	3.9572	6.6613e-14	0
ITAE	1.3079	2.3862	2.2204e-14	0

It is observed from the results that SA-PD controller yields less value of rise and settling time than CSA-PD controller. The % overshoot is negligible with both evolutionary techniques (SA and CSA). The steady state error is also zero with both SA and CSA. The rise time of SA based PD controller subjected to minimize ISE has been reduced to **27%** as compared to CSA based PD controller. Further, the settling time of SA based PD controller subjected to minimize ISE has been reduced to **31%** as compared to CSA based PD controller. Finally, it can be concluded that SA based PD controller is relatively better than CSA based PD controller in terms of rise and settling time.

6.4 Step Response of PI-D controller with SA and CSA

In this section, PI-D controller with Simulated Annealing (SA) and Cuckoo Search Algorithm (CSA) has been implemented. Optimum parameters of the PI-D controller have been obtained successfully by using SA and CSA. Tuning parameters have been given in Table 6.6.

Table 6.6: Parameters of PI-D controller using SA and CSA

[redacted]						
[redacted]	59.9	15.5	14.82	42.32	5.94	16.77
	57.4	6.81	21	56.96	3.74	27.6
	59.8	5.63	17.04	42.95	3.84	18.06
	59.74	6	15.64	48.75	21.59	13.04

Figure 6.10 represents the schematic diagram for step response of ball and beam system with PI-D controller.

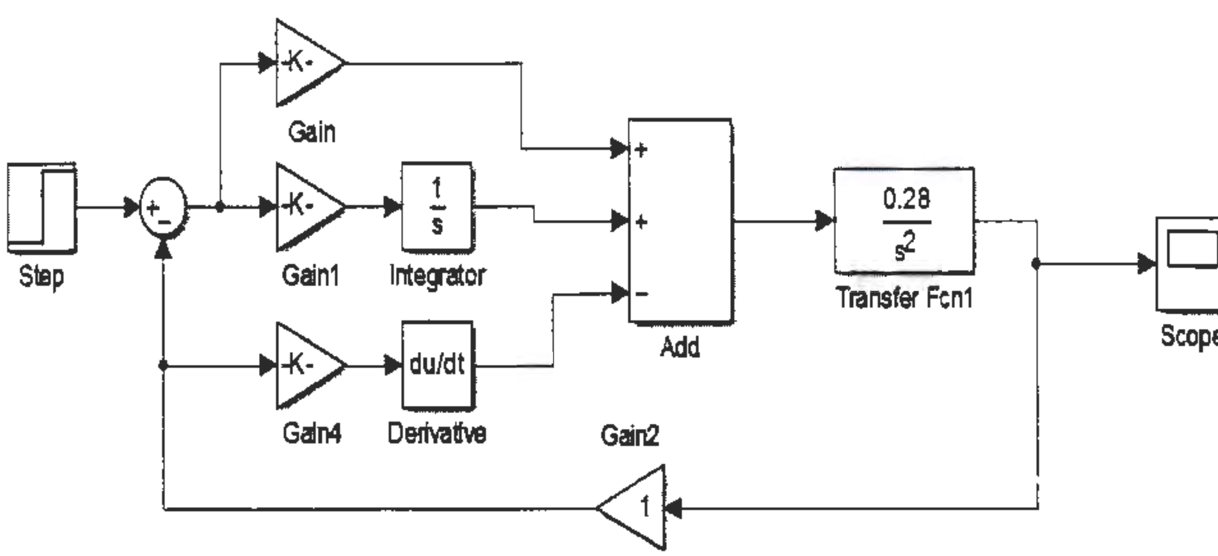

Figure 6.10: Schematic diagram for step response of Ball and Beam System with PI-D controller

Step responses of PI-D controller tuned with SA and CSA are provided in Figure 6.11 and Figure 6.12 respectively. Furthermore, the results of PI-D controller with different performance indices IAE, ISE, ITAE, and ITSE are also presented.

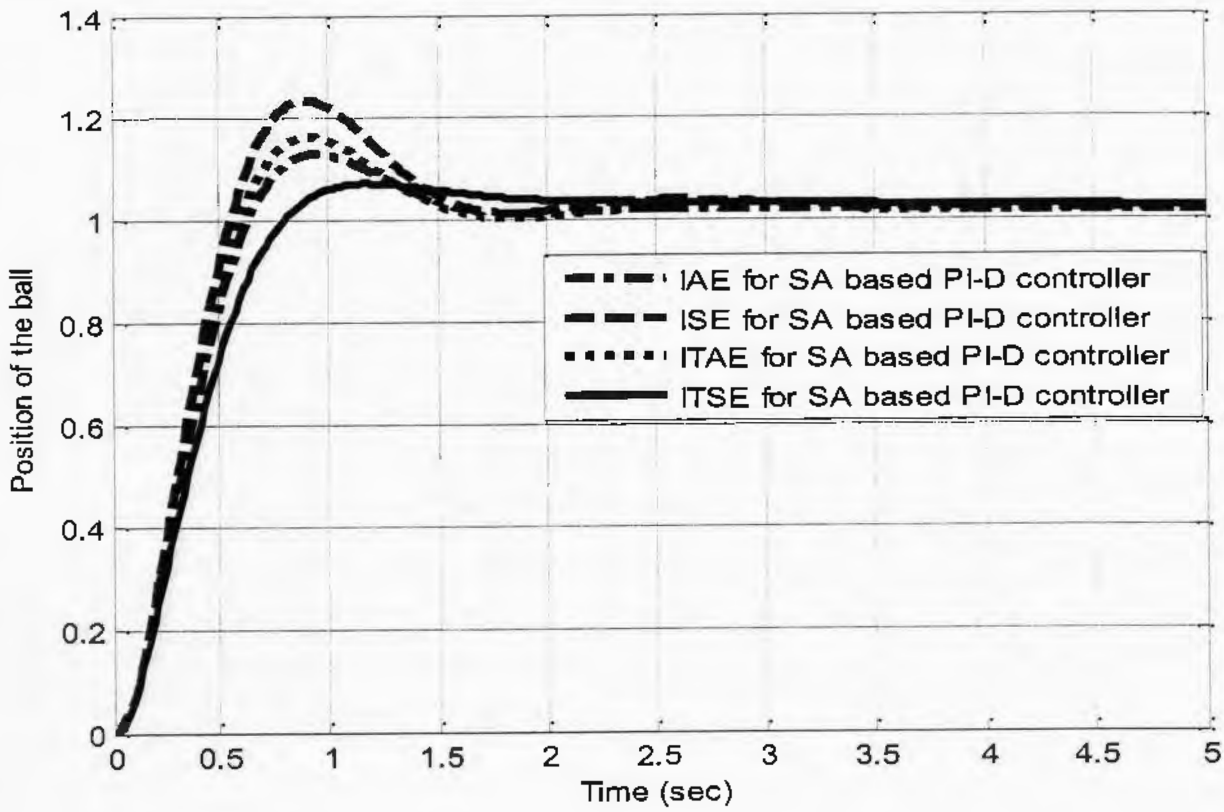

Figure 6.11: Step response of SA based PI-D controller

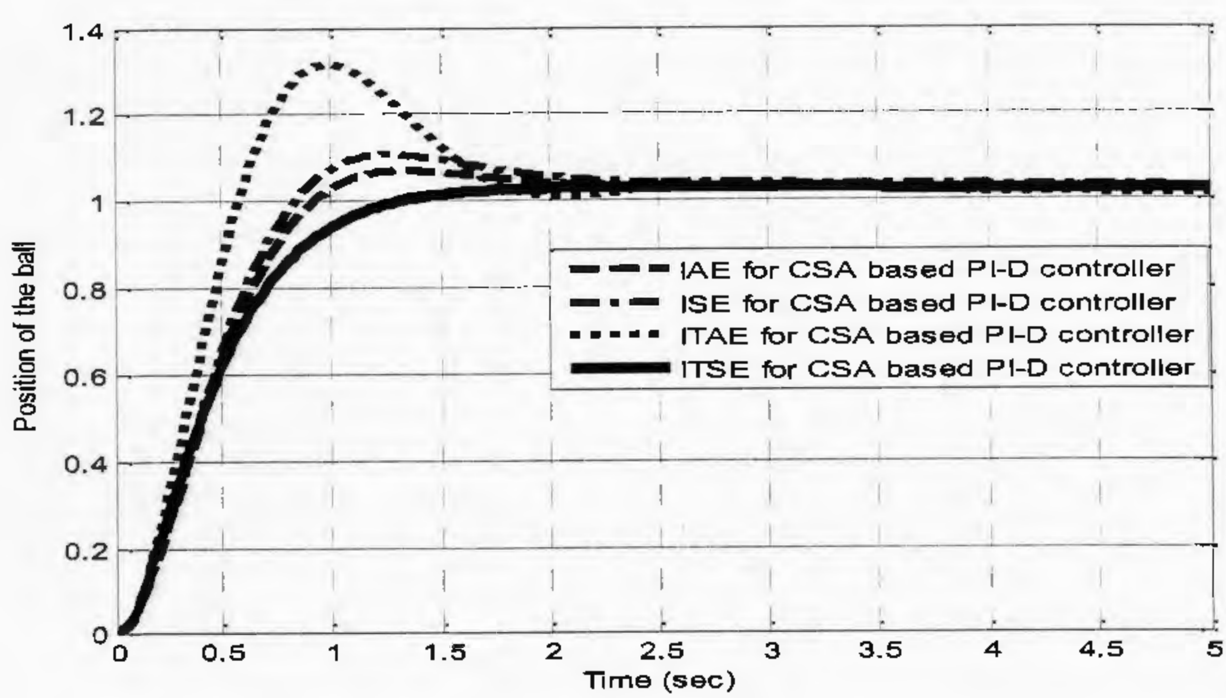

Figure 6.12: Step response of CSA based PI-D controller

In Table 6.7 and Table 6.8, the corresponding transient responses of SA based PI-D and CSA based PI-D controller have been summarized respectively.

Table 6.7: Performance comparison of SA based PI-D controller

[illegible]	Rise Time (sec)	Settling Time (sec)	% Overshoot	[illegible]
[illegible]	0.3478	4.71	23.5697	7.5273e-14
[illegible]	0.5203	7.27	7.3	1.9945e-07
[illegible]	0.4262	3.62	13.13	1.8496e-06
[illegible]	0.4027	3.3099	16.44	9.2260e-07

Table 6.8: Performance comparison of CSA based PI-D controller

[illegible]	Rise Time (sec)	Settling Time (sec)	% Overshoot	[illegible]
[illegible]	0.5610	7.8904	10.63	2.1872e-08
[illegible]	0.7530	8.2990	2.8997	3.8944e-05
[illegible]	0.6066	7.9092	6.9062	3.8114e-06
[illegible]	0.3918	3.8608	31.59	1.1102e-16

It is observed from the results that SA based PI-D controller yields less value of rise and settling time than CSA based PI-D controller. The steady state error is negligible with both evolutionary techniques (SA and CSA). The % overshoot (os) of CSA based PI-D controller subjected to minimize ITSE has been reduced to **60%** as compared to SA based PI-D controller. Further, the rise time of SA based PI-D controller subjected to minimize ISE has been reduced to **38%** as compared to CSA based PD controller. Moreover, the settling time of SA based PI-D controller subjected to minimize ITAE has been reduced to **14%** as compared to CSA based PD controller. It can be concluded from the results that SA based PI-D controller subjected to minimize ISE & ITAE is relatively better than CSA based PI-D controller in terms of rise and settling time. Further, It can also be concluded that CSA based PI-D controller subjected to minimize ITSE is relatively much better than SA based PI-D controller in terms of % overshoot (os).

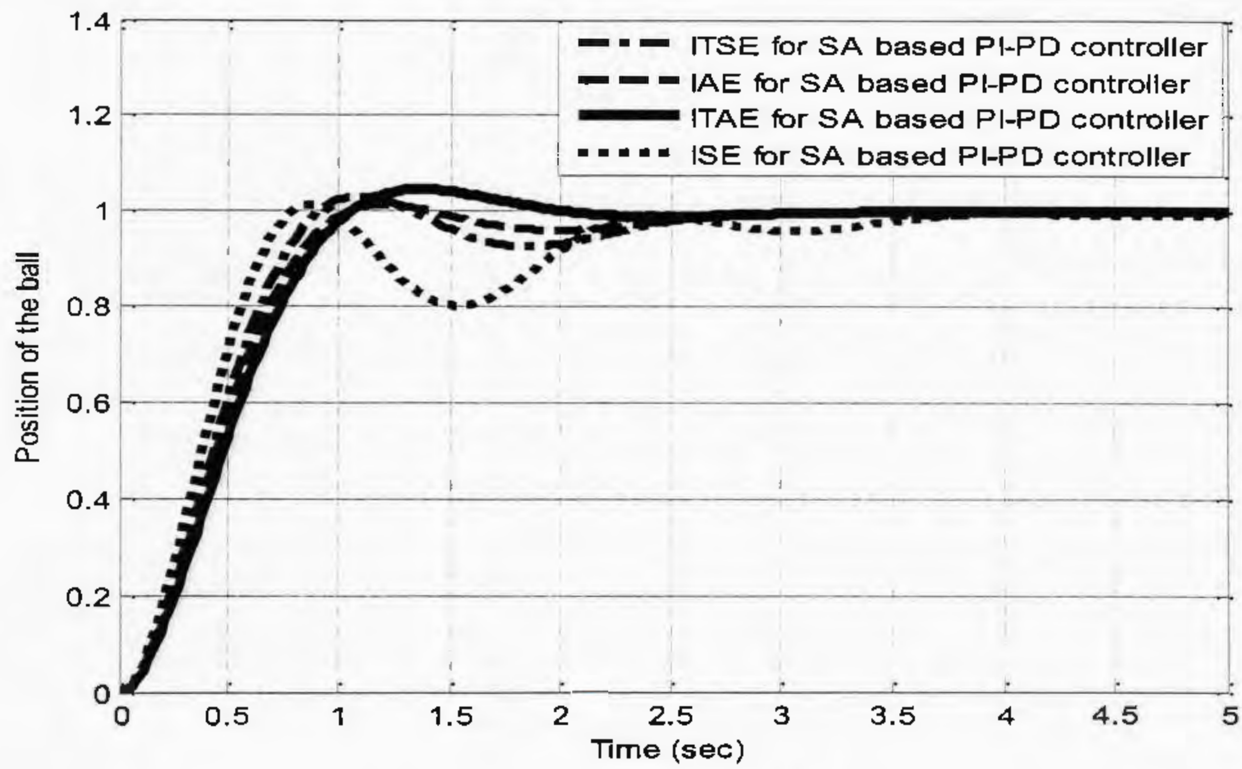

Figure 6.14: Step response of SA based PI-PD controller

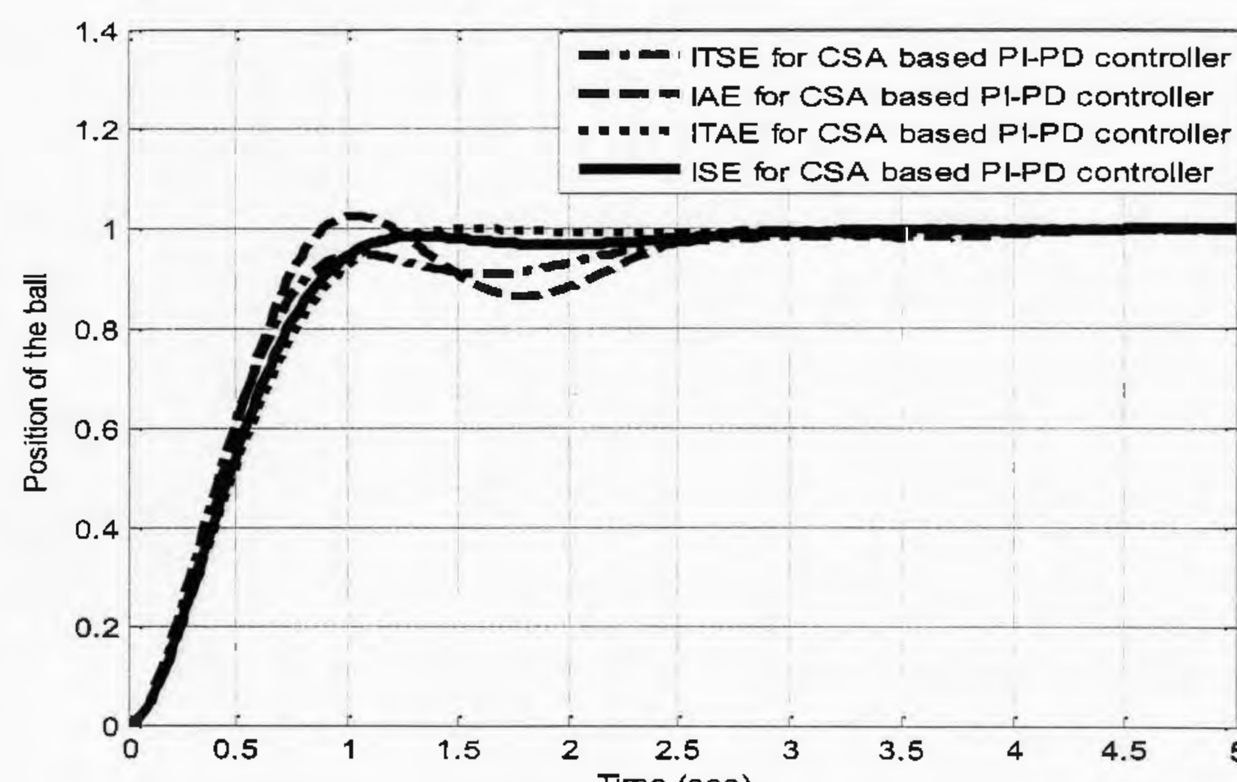

Figure 6.15: Step response of CSA based PI-PD controller

Transient response performance comparison of SA based PI-PD and CSA based PI-PD controller has been provided in Table 6.10 and Table 6.11 respectively.

Table 6.10: Performance comparison of SA based PI-PD controller

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	0.4908	3.50	1.3538	2.5535e-15
[illegible]	0.5795	2.50	3.14	8.8818e-16
[illegible]	0.6378	2.5043	1.3956	2.2204e-16
[illegible]	0.6783	1.7565	4.57	1.2212e-15

Table 6.11: Performance comparison of CSA based PI-PD controller

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	0.7189	2.5050	0	9.9920e-16
[illegible]	0.6592	2.7452	2.2204e-14	2.5535e-15
[illegible]	0.5681	2.5592	2.61	9.9920e-16
[illegible]	0.7660	1.2092	0.0086	7.7716e-16

It can be observed from the results that SA based PI-PD controller yields relatively less values of rise time than CSA-PD controller. However, CSA based PI-PD controller yields relatively lower values of settling time SA based PI-PD controller The steady state error is negligible with both evolutionary techniques (SA and CSA). The % overshoot (os) of CSA based PI-PD controller is relatively much better than SA based PI-PD controller. Further, the rise time of SA based PI-PD controller subjected to minimize ISE has been reduced to **31.7%** as compared to CSA based PI-PD controller. Moreover, the settling time of CSA based PI-PD controller subjected to minimize ITAE has been reduced to **31%** as compared to SA based PI-PD controller. It can be concluded from the results that SA based PI-PD controller subjected to minimize ISE is relatively better than CSA based PI-PD controller in terms of rise time. Further, it can also be concluded that CSA based PI-PD controller subjected to minimize ITAE is relatively much better than SA based PI-PD controller in terms of settling time.

In Table 6.12, comparison of our proposed CSA-PI-PD controller is made with PSO based I-PD and H-infinity based PID controller [20].

Table 6.12: Performance comparison of CSA-PI-PD controller with I-PD and PID controller [20]

Performance Parameters	Rise Time (sec)	Settling time (sec)	% Overshoot
[illegible] ISE	0.72	2.51	0
[illegible] ITAE	0.66	2.74	0
[illegible]	0.57	2.56	2.6
[illegible] ITAE	0.77	1.21	0
[illegible]	*0.1*	*4.3*	*20.7*
[illegible] based I-PD controller	*6.8*	*19.9*	*0*

It has been observed that CSA-PI-PD controller yields relatively smaller overshoot and setting time. CSA-PI-PD controller subjected to minimize ITAE has reduced the rise time up to **88%** and settling time up to **72%** as compared to PSO based I-PD controller. Moreover, CSA-PI-PD controller subjected to minimize ISE has reduced settling time up to **41%** with zero % overshoot as compared to H-infinity based PID controller. Finally it can be concluded that CSA based PI-PD controller subjected to minimize ISE and ITAE is very effective for the control of ball and beam system than H-infinity based PID and PSO based I-PD controller.

6.6 Set point tracking of ball & beam system using GA based PI-PD controller

In this section, PI-PD controller with Genetic Algorithm (GA) has been implemented for set point tracking of ball and beam system. This is to verify the effectives of PI-PD controller for the control of ball and beam system. For this purpose, we have changed the beam length and investigated the set point tracking response of GA based PI-PD controller. Table 6.13 represents the new and older value of beam length.

Table 6.13: Updated Beam Length

Older value of beam length (L)	New value of beam length (L)
1m	0.4m

The new expression for the transfer function of ball and beam system can be written as:

$$G_{BB}(s)= \frac{P(s)}{\theta(s)} = \frac{0.7}{s^2} \quad (6.1)$$

In all simulations different set points (10cm, 20cm, 30cm, 20cm and 10cm) are taken as reference positions respectively. Figure 6.16 shows the open loop set point tracking of ball and beam system. It can be observed that the system's response is unstable so it requires some control mechanism to act as a stable system.

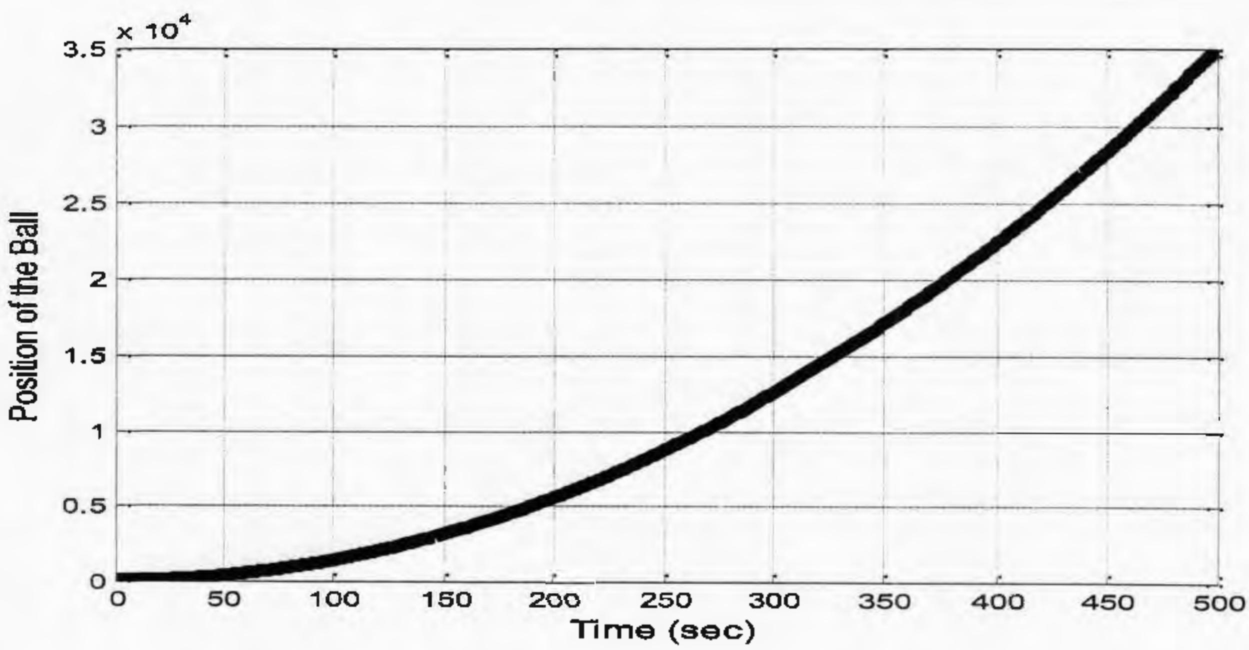

Figure 6.16: Open-loop step set point tracking response of Ball and Beam System

Implementation of PI-PD controllers with GA for set point tracking response is also provided in this section. Table 6.14 represents the OPTIMTOOL settings for the Genetic Algorithm (GA). In Table 6.15, tuning parameters of PI-PD controller with GA been provided. Figure 6.17 represents the schematic diagram for set point tracking response of ball and beam system by PI-PD controller.

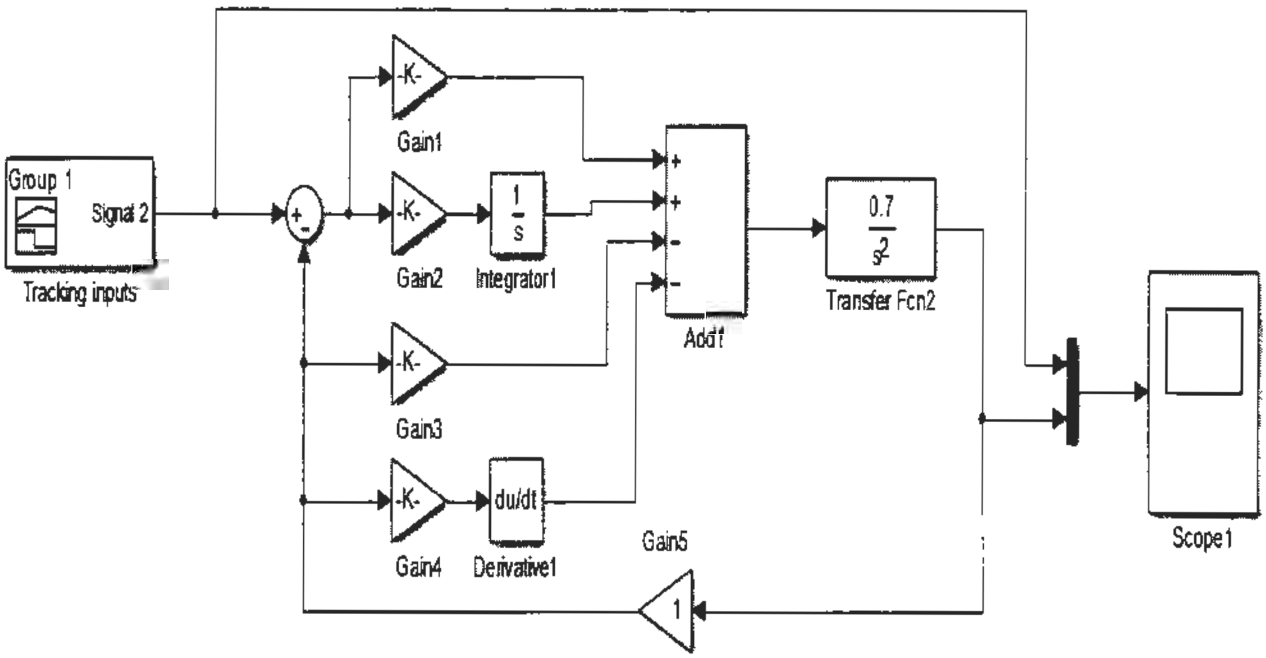

Figure 6.17: Schematic diagram for set point tracking of Ball and Beam System with PI-PD controller

Table 6.14: GA-OPTIMTOOL settings for Genetic Algorithm (GA) in MATLAB

Performance Index	Population Type	Fitness Scaling	Selection Function	Mutation	Crossover	Lower Bound	Upper Bound	Population size
ISE	Double Vector	Proportional	Stochastic Uniform	Adaptive Feasible	Two Point	[5,3]	[50,30]	10
ITSE	Double Vector	Proportional	Stochastic Uniform	Adaptive Feasible	Two Point	[5,3]	[50,30]	10
[illegible]	Double Vector	Proportional	Stochastic Uniform	Adaptive Feasible	Two Point	[5,3]	[50,30]	10
[illegible]	Double Vector	Proportional	Stochastic Uniform	Adaptive Feasible	Two Point	[5,3]	[50,30]	10

Table 6.15: PI-PD controller's tuning by using Genetic Algorithm (GA)

Performance Index	K_P	K_i	K_D
ISE	17.261	29.99	4.96
ITSE	14.19	29.99	6.68
IAE	13.99	29.99	7.55
ITAE	14.46	29.99	9.21

Figure 6.18 and 6.19 shows the output responses of GA based PI-PD controller for IAE and ISE respectively. Similarly Figure 6.20 and 6.21 shows the output responses of GA based PI-PD controller for ITSE and ITAE respectively.

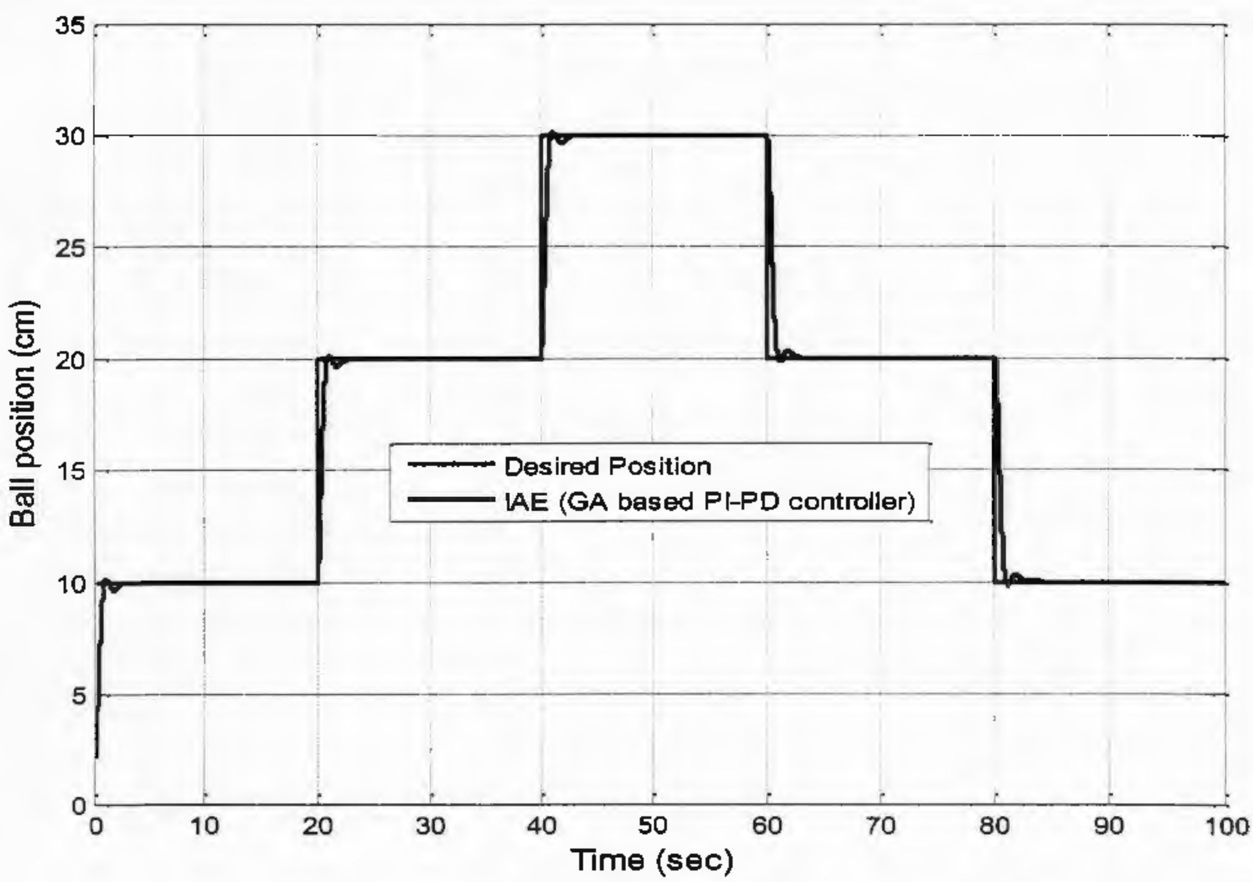

Figure 6.18: Set point tracking of GA based PI-PD controller with IAE

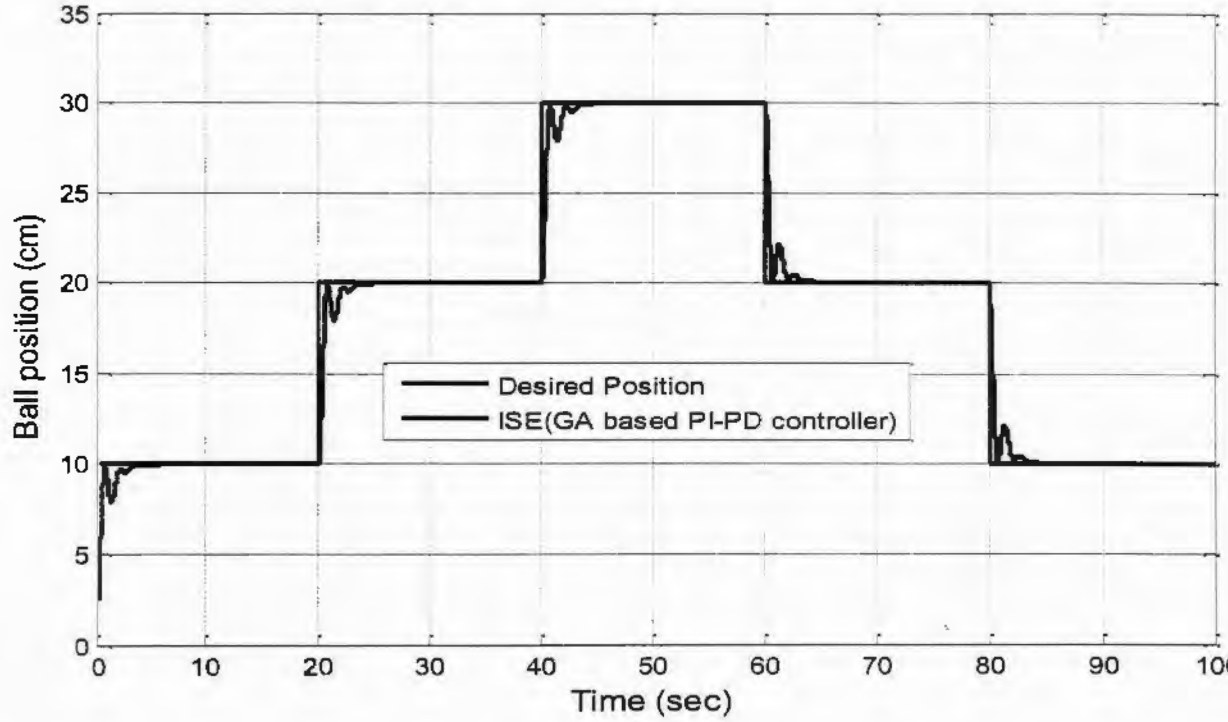

Figure 6.19: Set point tracking of GA based PI-PD controller with ISE

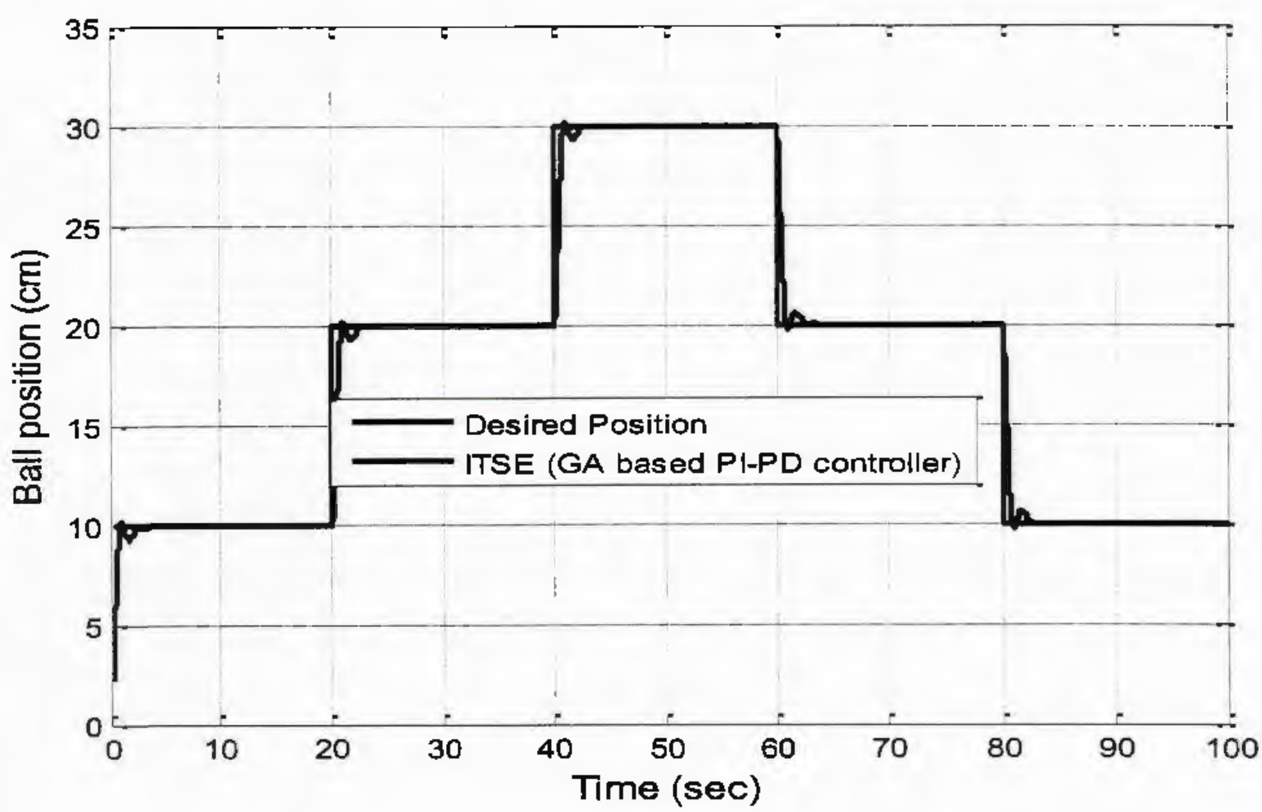

Figure 6.20: Set point tracking of GA based PI-PD controller with ITSE

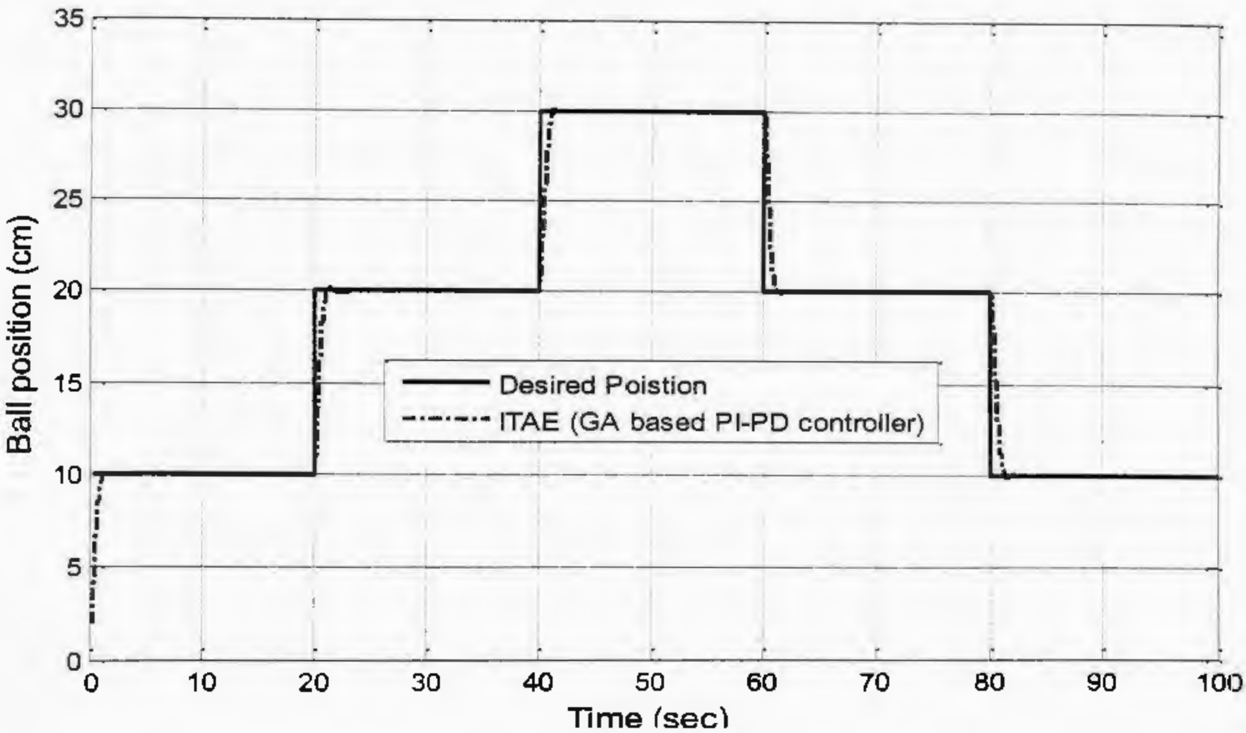

Figure 6.21: Set point tracking of GA based PI-PD controller with ITAE

Table 6.16 represents the transient response performance of GA based PI-PD controller with each performance index. It can be observed from the results of Table 6.16 that GA based PI-PD controller subjected to minimize ISE exhibits relatively less rise time along with zero % overshoot. Moreover it can be observed that GA based PI-PD controller subjected to minimize ITAE gives lower value of settling time. Finally it can be concluded that GA based PI-PD controller yields much better transient response in terms of steady state error (e_{ss}.), settling time (t_s), overshoot (os) and rise time (t_r).

Table 6.16: Comparison of GA based PI-PD controller with different performance indices

Performance Parameter / Performance Index	settling time (sec)	s-s error	rise time (sec)	% overshoot
IAE	2.30	0	0.60	1.73
ITAE	1.08	0	0.70	1
ISE	3.24	0	0.46	0
ITSE	2.33	0	0.55	1.90

Furthermore Figure 6.22 shows the comparison of GA based PI-PD controller with SIMC based PID and H-infinity controller as used in [3]. The performance comparison is also provided in Table 6.17.

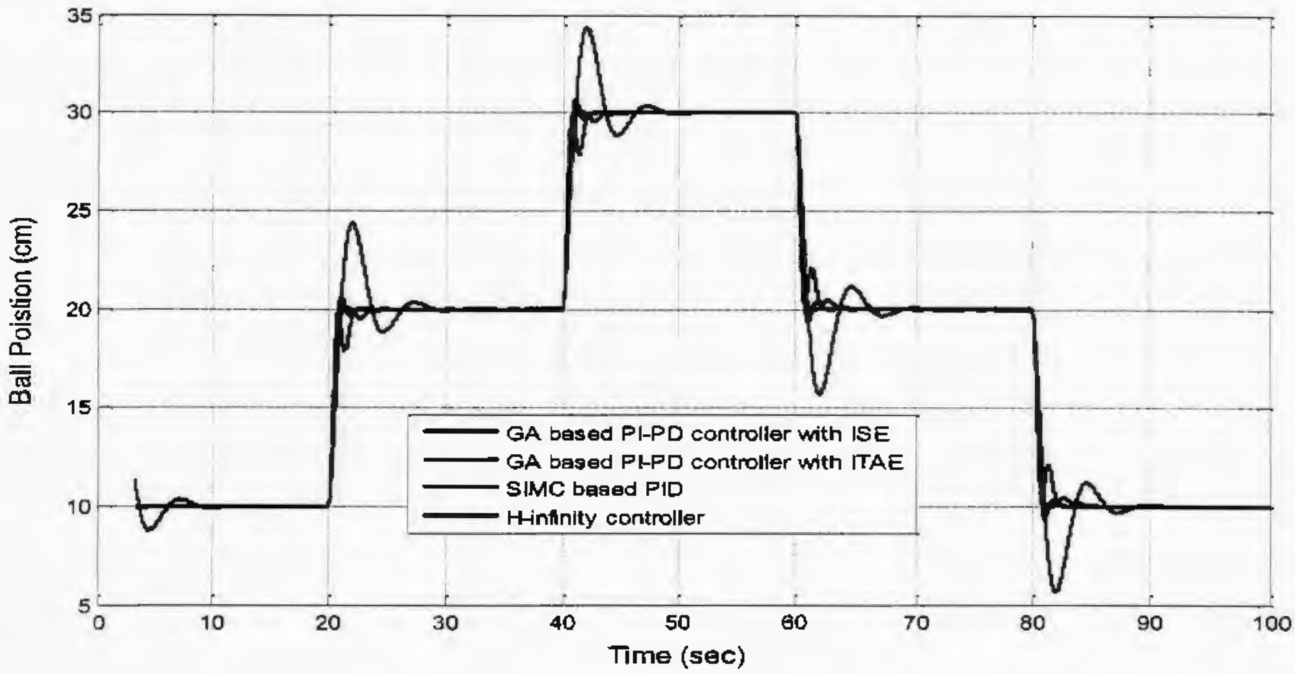

Figure 6.22: Set point tracking of PI-PD controller vs. SIMC based PID and H-infinity controller

Table 6.17: Comparison of GA based PI-PD with SIMC based PID and H-infinity controller [3]

Performance Parameter / Controller		settling time (sec)	s-s error	% overshoot	rise time (sec)
GA based PI-PD	ISE	3.24	0	0	0.46
	ITAE	1.08	0	1	0.70
SIMC based PID controller		10.9	0	40	1
H-infinity controller		3.7	0	6.7	1.1

Figure 6.22 and Table 6.17 reveal that GA based PI-PD controller yields very negligible % overshoot (os) with lower value of rise time (t_r) and settling time (t_S) as compared to SIMC based PID and H-infinity controller. GA based PI-PD controller subjected to minimize ISE has reduced the rise time up to **54%** and settling time up to **70%** as compared to SIMC based PID controller. SIMC based PID controller exhibits **40%** overshoot (os) whereas GA based PI-PD controller subjected to minimize ISE has completely removed the overshoot. Similarly GA based PI-PD controller subjected to minimize ITAE has reduced the rise time up to **36%** and settling time up to **71%** as compared to H-infinity controller. H-infinity controller exhibits **6.7%** overshoot whereas GA based PI-PD controller subjected to minimize ITAE has reduced the % overshoot up to **85%**. These results reveal that how much effective is the PI-PD controller with evolutionary computational techniques.

CHAPTER 7

Conclusions and Future Work

7.1 Conclusions

The stability control of the ball & beam system has been analyzed by using PD, PI, I-P, PI-D and PI-PD controllers. Evolutionary computation algorithms including Cuckoo Search Algorithm (CSA), Simulated Annealing (SA) and Genetic Algorithm (GA) have been employed to find out optimum values of controller parameters. Furthermore, four differnt perforamnces indices including ISE, IAE, ITAE and ITSE have been used for the evaluation. Simulations have beem carried out in the MATLAB-Simulink software. The step response has been investigated. It has been observed from the simulation results that both PI-D and PI-PD controllers have given satisfactoy results with each proposed tuning technique. PI-D controller exhibits more settling time (t_s), % overshoot (os) and steady-state error (e_{ss}) than PI-PD controller. Thus it can be concluded that SA and CSA based PI-PD controller is very usefull and proficient for the stability control of ball & beam system as compared to PI-D controller. The comparison of our proposed CSA-PI-PD controller is made with PSO based I-PD and H-infinity based PID controller [9]. It has been observed that CSA-PI-PD controller yields relatively smaller overshot and setting time. CSA-PI-PD controller subjected to minimize ITAE has reduced the rise time up to 88% and settling time up to 72% as compared to PSO based I-PD controller. Moreover, CSA-PI-PD controller subjected to minimize ISE has reduced settling time up to 41% with zero overshoot as compared to H-infinity based PID controller.

The set point tracking response of ball and beam system has also been investigated using PI-PD controller. Genetic Algorithm (GA) has been utilized to find out optimum gain parameters of the PI-PD controller. All four above mentioned performance indices have been used for the evaluation of the controller. Simulation results reveal that proposed PI-PD controller with any performance index is very effective as comapred to SIMC based PID and H-infinty controllers [2]. PI-PD controller tuned by GA has less overshoot (os), settling time(t_S) and rise time (t_r). PI-PD controller has very negligible % overshoot which is very reliable for the plant dynamics. Similarly rise time and settling time have been reduced so that sytem will adjust to deired position within very small time intrerval. Finally it can be concluded from the results that PI-PD controller tuned by Genetic Aalgorithm (GA) is much efficient and valuable for the set point tracking of ball & beam systems which is very supportive for the control of many engineering problems in the future. Thus, it can be suggested that evolutionary computation based PI-PD controller is very victorious for the stability control of ball and beam system.

7.2 Future Recommendations

In future, PI-D, PI-PD, I-PD and other possible configurations may be explored for the efficient control of different plants other than ball and beam system. Moreover, heuristic computational techniques such as Artificial Bee Colony (ABC), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO) and Differential Evolution (DE) may be used to find out optimum gain parameters of the controllers.

References

[1] W. Yu and F. Ortiz, "Stability analysis of PD regulation for ball and beam system," in *Proceedings of 2005 IEEE Conference on Control Applications, 2005. CCA 2005.*, 2005, pp. 517–522.

[2] W. Wang, "Control of a ball and beam system," *M. Tech Thesis, Univ. Adelaide, Aust.*, 2007.

[3] S. Sathiyavathi and K. Krishnamurthy, "PID control of ball and beam system-A real time experimentation," *J. Sci. Ind. Res.*, vol. 72, pp. 481–484, 2013.

[4] M. Keshmiri, A. F. Jahromi, A. Mohebbi, M. H. Amoozgar, and W.-F. Xie, "Modeling and control of ball and beam system using model based and non-model based control approaches," *Int. J. smart Sens. Intell. Syst.*, vol. 5, no. 1, pp. 14–35, 2012.

[5] E. V. Kumar, J. Jerome, and G. Raaja, "State Dependent Riccati Equation based Nonlinear Controller Design for Ball and Beam System," *Procedia Eng.*, vol. 97, pp. 1896–1905, 2014.

[6] K. Ogata and Y. Yang, "Modern control engineering," 1970.

[7] J. Huang and C.-F. Lin, "Robust nonlinear control of the ball and beam system," in *American Control Conference, Proceedings of the 1995*, 1995, vol. 1, pp. 306–310.

[8] H. K. Lam, F. H. F. Leung, and P. K. S. Tam, "Design of a fuzzy controller for stabilizing a ball-and-beam system," in *Industrial Electronics Society, 1999. IECON'99 Proceedings. The 25th Annual Conference of the IEEE*, 1999, vol. 2, pp. 520–524.

[9] Q. Wang, M. Mi, G. Ma, and P. Spronck, "Evolving a neural controller for a ball-and-beam system," in *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on*, 2004, vol. 2, pp. 757–761.

[10] C. Aguilar-Ibañez, "On the stabilization of the ball and beam system using a direct Lyapunov method," in *Electrical Engineering, Computing Science and Automatic Control, CCE, 2009 6th International Conference on*, 2009, pp. 1–6.

[11] S. S. Abdullah, M. Amjad, M. I. Kashif, and Z. Shareef, "A simplified intelligent controller for ball and beam system," 2010.

[12] Z.-H. Pang, G. Zheng, and C.-X. Luo, "Augmented state estimation and LQR control for a ball and beam system," in *Industrial Electronics and Applications (ICIEA), 2011 6th IEEE Conference on*, 2011, pp. 1328–1332.

[13] M. A. Rana, Z. Usman, and Z. Shareef, "Automatic control of ball and beam system using particle swarm optimization," in *Computational Intelligence and Informatics (CINTI), 2011 IEEE 12th International Symposium on*, 2011, pp. 529–534.

[14] K. T. Prasad and Y. V Hote, "Optimal PID controller for Ball and Beam system," in *Recent Advances and Innovations in Engineering (ICRAIE), 2014*, 2014, pp. 1–5.

[15] N. Hara, M. Takahashi, and K. Konishi, "Experimental evaluation of model predictive control of ball and beam systems," in *American Control Conference (ACC), 2011*, 2011, pp. 1130–1132.

[16] Y.-H. Chang, C.-W. Chang, H.-W. Lin, and C. W. Tao, "Fuzzy controller design for ball and beam system with an improved ant colony optimization," *World Acad. Sci. Eng. Technol.*, vol. 56, pp. 616–621, 2009.

[17] M. Shahrokhi and A. Zomorrodi, "Comparison of PID controller tuning methods," *Dep. Chem. Pet. Eng. Sharif Univ. Technol.*, 2013.

[18] K. K. Tan, S. Huang, and R. Ferdous, "Robust self-tuning PID controller for nonlinear systems," *J. Process Control*, vol. 12, no. 7, pp. 753–761, 2002.

[19] S.-K. Oh, H.-J. Jang, and W. Pedrycz, "The design of a fuzzy cascade controller for ball and beam system: A study in optimization with the use of parallel genetic algorithms," *Eng. Appl. Artif. Intell.*, vol. 22, no. 2, pp. 261–271, 2009.

[20] D. D. Kumar, B. Meenakshipriya, and S. S. Ram, "Design and Comparsion of PID and PSO based I-PD Controller for Ball and Beam System," 2016.

[21] H. Do Kim, K. H. Oh, and H. L. Choi, "Output Feedback Control of a Ball and Beam System Based on Jacobian Linearization under Sensor Noise," in *Advanced Engineering Forum*, 2012, vol. 2, pp. 85–90.

[22] M. Amjad, M. I. Kashif, S. S. Abdullah, and Z. Shareef, "Fuzzy logic control of ball and beam system," in *Education Technology and Computer (ICETC), 2010 2nd International Conference on*, 2010, vol. 3, pp. V3-489.

[23] Y.-H. Chang, W.-S. Chan, and C.-W. Chang, "TS fuzzy model-based adaptive dynamic surface control for ball and beam system," *IEEE Trans. Ind. Electron.*, vol. 60, no. 6, pp. 2251–2263, 2013.

[24] Y.-H. Chang, C.-W. Chang, C.-W. Tao, H.-W. Lin, and J.-S. Taur, "Fuzzy sliding-mode control for ball and beam system with fuzzy ant colony optimization," *Expert Syst. Appl.*, vol. 39, no. 3, pp. 3624–3633, 2012.

[25] N. B. Almutairi and M. Zribi, "On the sliding mode control of a ball on a beam system," *Nonlinear Dyn.*, vol. 59, no. 1, pp. 221–238, 2010.

[26] P. Jain and M. J. Nigam, "Real time control of ball and beam system with model reference adaptive control strategy using MIT rule," in *Computational Intelligence and Computing Research (ICCIC), 2013 IEEE International Conference on*, 2013, pp. 1–4.

[27] M. A. Marra, B. E. Boling, and B. L. Walcott, "Genetic control of a ball-beam system," in *Control Applications, 1996., Proceedings of the 1996 IEEE International Conference on*,

1996, pp. 608–613.

[28] C. Aguilar-Ibañez, M. S. Suarez-Castanon, and J. de J. Rubio, "Stabilization of the ball on the beam system by means of the inverse Lyapunov approach," *Math. Probl. Eng.*, vol. 2012, 2012.

[29] L. Márton and B. Lantos, "Stable adaptive ball and beam control," in *Mechatronics, 2006 IEEE International Conference on*, 2006, pp. 507–512.

[30] S. A. Bhatti, S. A. Malik, and A. Daraz, "Comparison of PI and IP controller by using Ziegler-Nichols tuning method for speed control of DC motor," in *Intelligent Systems Engineering (ICISE), 2016 International Conference on*, 2016, pp. 330–334.

[31] T. Sreekumar and K. S. Jiji, "Comparison of Proportional-Integral (PI) and Integral-Proportional (IP) controllers for speed control in vector controlled induction Motor drive," in *Power, Control and Embedded Systems (ICPCES), 2012 2nd International Conference on*, 2012, pp. 1–6.

[32] S. Nema and P. Kumar Padhy, "Identification and cuckoo PI-PD controller design for stable and unstable processes," *Trans. Inst. Meas. Control*, vol. 37, no. 6, pp. 708–720, 2015.

[33] X.-S. Yang and S. Deb, "Engineering optimisation by cuckoo search," *Int. J. Math. Model. Numer. Optim.*, vol. 1, no. 4, pp. 330–343, 2010.

[34] A. Narwal and B. R. Prasad, "A novel order reduction approach for LTI systems using Cuckoo search optimization and stability equation," *IETE J. Res.*, vol. 62, no. 2, pp. 154–163, 2016.

[35] K.-L. Du and M. N. S. Swamy, "Search and Optimization by Metaheuristics," 2016.

[36] V. Tiwari, "Face recognition based on cuckoo search algorithm," *Image (IN).*, vol. 7, no. 8, p. 9, 2012.

[37] İ. Durgun and A. R. Yildiz, "Structural design optimization of vehicle components using cuckoo search algorithm," *Mater. Test.*, vol. 54, no. 3, pp. 185–188, 2012.

[38] P. Dash, L. C. Saikia, and N. Sinha, "Comparison of performances of several Cuckoo search algorithm based 2DOF controllers in AGC of multi-area thermal system," *Int. J. Electr. Power Energy Syst.*, vol. 55, pp. 429–436, 2014.

[39] S. Burnwal and S. Deb, "Scheduling optimization of flexible manufacturing system using cuckoo search-based approach," *Int. J. Adv. Manuf. Technol.*, vol. 64, no. 5–8, pp. 951–959, 2013.

[40] P. Dash, L. C. Saikia, and N. Sinha, "Comparison of performances of several FACTS devices using Cuckoo search algorithm optimized 2DOF controllers in multi-area AGC," *Int. J. Electr. Power Energy Syst.*, vol. 65, pp. 316–324, 2015.

[41] A. Y. Abdelaziz and E. S. Ali, "Cuckoo search algorithm based load frequency controller design for nonlinear interconnected power system," *Int. J. Electr. Power Energy Syst.*, vol. 73, pp. 632–643, 2015.

[42] M. A. Adnan, M. A. Razzaque, I. Ahmed, and I. F. Isnin, "Bio-mimic optimization strategies in wireless sensor networks: A survey," *Sensors*, vol. 14, no. 1, pp. 299–345, 2013.

[43] N. Thomas and D. P. Poongodi, "Position control of DC motor using genetic algorithm based PID controller," in *Proceedings of the World Congress on Engineering*, 2009, vol. 2, pp. 1–3.

[44] R. Dimeo and K. Y. Lee, "Boiler-turbine control system design using a genetic algorithm," *IEEE Trans. energy Convers.*, vol. 10, no. 4, pp. 752–759, 1995.

[45] A. G. Bakirtzis, P. N. Biskas, C. E. Zoumas, and V. Petridis, "Optimal power flow by enhanced genetic algorithm," *IEEE Trans. power Syst.*, vol. 17, no. 2, pp. 229–236, 2002.

[46] T. T. Chow, G. Q. Zhang, Z. Lin, and C. L. Song, "Global optimization of absorption chiller system by genetic algorithm and neural network," *Energy Build.*, vol. 34, no. 1, pp. 103–109, 2002.

[47] M. V Sadasivarao and M. Chidambaram, "PID controller tuning of cascade control systems using genetic algorithm," *J. Indian Inst. Sci.*, vol. 86, no. 7, pp. 343–354, 2006.

[48] T. L. Seng, M. Bin Khalid, and R. Yusof, "Tuning of a neuro-fuzzy controller by genetic algorithm," *IEEE Trans. Syst. Man, Cybern. Part B*, vol. 29, no. 2, pp. 226–236, 1999.

[49] K.-J. Han, H.-S. Cho, D.-H. Cho, and H.-K. Jung, "Optimal core shape design for cogging torque reduction of brushless DC motor using genetic algorithm," *IEEE Trans. Magn.*, vol. 36, no. 4, pp. 1927–1931, 2000.

[50] A. Rahideh, T. Korakianitis, P. Ruiz, T. Keeble, and M. T. Rothman, "Optimal brushless DC motor design using genetic algorithms," *J. Magn. Magn. Mater.*, vol. 322, no. 22, pp. 3680–3687, 2010.

[51] M. Jaiswal and M. Phadnis, "Speed control of DC motor using genetic algorithm based PID controller," *Int. J. Adv. Res. Comput. Sci. Softw. Eng. ISSN*, vol. 2277, 2013.

[52] S. A. Malik, I. M. Qureshi, M. Amir, and I. Haq, "Nature inspired computational technique for the numerical solution of nonlinear singular boundary value problems arising in physiology," *Sci. World J.*, vol. 2014, 2014.

[53] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science (80-.).*, vol. 220, no. 4598, pp. 671–680, 1983.

[54] A. Van Breedam, "Improvement heuristics for the vehicle routing problem based on simulated annealing," *Eur. J. Oper. Res.*, vol. 86, no. 3, pp. 480–490, 1995.

[55] R. Tavakkoli-Moghaddam, N. Safaei, and Y. Gholipour, "A hybrid simulated annealing for capacitated vehicle routing problems with the independent route length," *Appl. Math. Comput.*, vol. 176, no. 2, pp. 445–454, 2006.

[56] W. L. Goffe, G. D. Ferrier, and J. Rogers, "Global optimization of statistical functions with simulated annealing," *J. Econom.*, vol. 60, no. 1–2, pp. 65–99, 1994.

[57] R.-E. Precup, R.-C. David, E. M. Petriu, S. Preitl, and M.-B. Radac, "Fuzzy control systems with reduced parametric sensitivity based on simulated annealing," *IEEE Trans. Ind. Electron.*,

vol. 59, no. 8, pp. 3049–3061, 2012.

[58] D. P. Kwok and F. Sheng, "Genetic algorithm and simulated annealing for optimal robot arm PID control," in *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, 1994, pp. 707–713.

[59] M. G. Park, J. H. Jeon, and M. C. Lee, "Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing," in *Industrial Electronics, 2001. Proceedings. ISIE 2001. IEEE International Symposium on*, 2001, vol. 3, pp. 1530–1535.

[60] M.-T. Vakil-Baghmisheh and M. Salim, "The design of PID controllers for a Gryphon robot using four evolutionary algorithms: a comparative study," *Artif. Intell. Rev.*, vol. 34, no. 2, pp. 121–132, 2010.

Appendix A

Formulation of the Fitness Function for PD controller:

Let E(s) is defined as:

$$E(s) = R(s) - Y(s) \quad \text{(A-1)}$$

$$\frac{Y(s)}{R(s)} = \frac{G_{PD}(s).G_{BB}(s)}{1+G_{PD}(s).G_{BB}(s)} \quad \text{(A-2)}$$

$$\frac{Y(s)}{R(s)} = \frac{(K_P+K_D s)(\frac{0.28}{s^2})}{1+(K_P+K_D s)(\frac{0.28}{s^2})} \quad \text{(A-3)}$$

$$\frac{Y(s)}{R(s)} = \frac{(0.28K_D s+0.28K_P)/s^2}{(s^2+0.28K_D s+0.28K_P)/s^2} \quad \text{(A-4)}$$

$$Y(s) = \frac{0.28K_D s+0.28K_P}{s^2+0.28K_D s+0.28K_P} * R(s) \quad \text{(A-5)}$$

Putting Y(s) into equation (6.1),

$$E(s) = R(s) - \frac{0.28K_D s+0.28K_P}{s^2+0.28K_D s+0.28K_P} * R(s) \quad \text{(A-6)}$$

$$E(s) = R(s) * (1 - \frac{0.28K_D s+0.28K_P}{s^2+0.28K_D s+0.28K_P}) \quad \text{(A-7)}$$

$$E(s) = R(s) * (\frac{s^2+0.28K_D s+0.28K_P-0.28K_D s-0.28K_P}{s^2+0.28K_D s+0.28K_P}) \quad \text{(A-8)}$$

$$E(s) = R(s) * (\frac{s^2}{s^2+0.28K_D s+0.28K_P}) \quad \text{(A-9)}$$

Since $R(s) = \frac{1}{s}$

$$E(s) = \frac{s}{s^2+0.28K_D s+0.28K_P} \quad \text{(A-10)}$$

By taking the inverse Laplace transform of equation A-10, **e(t)** can be written as:

e(t)=exp(-(7*kd*t)/50)*(cosh(t*((49*kd^2)/2500-(7*kp)/25)^(1/2))-
7*kd*sinh(t*((49*kd^2)/2500 - (7*kp)/25)^(1/2)))/(50*((49*kd^2)/2500 - (7*kp)/25)^(1/2)))

(A-11)

Appendix B

Formulation of the Fitness Function for PI controller:

Let E(s) is defined as:

$$E(s) = R(s) - Y(s) \tag{B-1}$$

$$\frac{Y(s)}{R(s)} = \frac{G_{PI}(s).G_{BB}(s)}{1+G_P(s).G_{BB}(s)} \tag{B-2}$$

$$\frac{Y(s)}{R(s)} = \frac{(K_P+\frac{K_i}{s})(\frac{0.28}{s^2})}{1+(K_P+\frac{K_i}{s})(\frac{0.28}{s^2})} \tag{B-3}$$

$$\frac{Y(s)}{R(s)} = \frac{[0.28(sK_P+K_i)/s^3]}{1+[0.28(sK_P+K_i)/s^3]} \tag{B-4}$$

$$\frac{Y(s)}{R(s)} = \frac{0.28K_P s+0.28K_i}{s^3+0.28K_p s+0.28K_i} \tag{B-5}$$

$$Y(s) = \frac{0.28K_P s+0.28K_i}{s^3+0.28K_p s+0.28K_i} * R(s) \tag{B-6}$$

Putting Y(s) into equation (6.11),

$$E(s) = R(s) - \frac{0.28K_P s+0.28K_i}{s^3+0.28K_p s+0.28K_i} * R(s) \tag{B-7}$$

$$E(s) = R(s) * (1 - \frac{0.28K_P s+0.28K_i}{s^3+0.28K_p s+0.28K_i}) \tag{B-8}$$

$$E(s) = R(s) * (\frac{s^3+0.28K_p s+0.28K_i-0.28K_P s-0.28K_i}{s^3+0.28K_p s+0.28K_i}) \tag{B-9}$$

$$E(s) = R(s) * (\frac{s^3}{s^3+0.28K_p s+0.28K_i}) \tag{B-10}$$

Since $R(s) = \frac{1}{s}$

$$E(s) = \frac{s^2}{s^3+0.28K_p s+0.28K_i} \tag{B-11}$$

By taking the inverse Laplace transform of equation B-11, **e (t)** can be written as:

e(t)= 25*sum((r3^2*exp(r3*t))/(75*r3^2 + 7*kp), r3 in RootOf (s3^3 + (7*kp*s3)/25 + (7*ki)/25, s3)) (B-12)

Appendix C

Formulation of the Fitness Function for I-P controller:

Let E(s) is defined as:

$$E(s) = R(s) - Y(s) \quad \text{(C-1)}$$

$$\frac{Y(s)}{R(s)} = \frac{(\frac{K_i}{s})(\frac{0.28}{s^2})}{1+\left[\left(\frac{K_i}{sK_i}\right)\left(\frac{0.28}{s^2}\right)\right][K_i+sK_p]} \quad \text{(C-2)}$$

$$\frac{Y(s)}{R(s)} = \frac{\frac{0.28K_i}{s^3}}{1+\left[\left(\frac{0.28}{s^3}\right)\right][K_i+sK_p]} \quad \text{(C-3)}$$

$$\frac{Y(s)}{R(s)} = \frac{\frac{0.28K_i}{s^3}}{s^3+0.28K_ps+0.28K_i]/s^3} \quad \text{(C-4)}$$

$$\frac{Y(s)}{R(s)} = \frac{0.28K_i}{s^3+0.28K_ps+0.28K_i} \quad \text{(C-5)}$$

$$Y(s) = \frac{0.28K_i}{s^3+0.28K_ps+0.28K_i} * R(s) \quad \text{(C-6)}$$

Putting Y(s) into equation (6.22),

$$E(s) = R(s) - \frac{0.28K_i}{s^3+0.28K_ps+0.28K_i} * R(s) \quad \text{(C-7)}$$

$$E(s) = R(s)(1 - \frac{0.28K_i}{s^3+0.28K_ps+0.28K_i}) \quad \text{(C-8)}$$

$$E(s) = R(s) * (\frac{s^3+0.28K_ps+0.28K_i-0.28K_i}{s^3+0.28K_ps+0.28K_i}) \quad \text{(C-9)}$$

$$E(s) = R(s) * (\frac{s^3+0.28K_ps}{s^3+0.28K_ps+0.28K_i}) \quad \text{(C-10)}$$

Since $R(s) = \frac{1}{s}$

$$E(s) = \frac{s^2 + 0.28K_p}{s^3 + 0.28K_p s + 0.28K_i} \tag{C-11}$$

By taking the inverse Laplace transform of equation C-11, **e(t)** can be written as

e(t) =7*kp*sum(exp(r3*t)/(75*r3^2 + 7*kp), r3 in RootOf(s3^3 + (7*kp*s3)/25 + (7*ki)/25, s3)) + 25*sum((r3^2*exp(r3*t))/(75*r3^2 + 7*kp), r3 in RootOf(s3^3 + (7*kp*s3)/25 + (7*ki)/25, s3)) (C-12)

Appendix D

Formulation of the Fitness Function for PI-D controller:

Let E(s) is defined as:

$$E(s) = R(s) - Y(s) \tag{D-1}$$

$$\frac{Y(s)}{R(s)} = \frac{\left(\frac{sK_P+K_i}{s}\right)\left(\frac{0.28}{s^2}\right)}{1+\left(\frac{sK_P+K_i}{s}\right)\left(\frac{0.28}{s^2}\right)\left(1+\frac{K_D s^2}{K_P s+K_i}\right)} \tag{D-2}$$

$$\frac{Y(s)}{R(s)} = \frac{0.28K_P s+0.28K_i}{s^3+0.28K_D s^2+0.28K_p s+0.28K_i} \tag{D-3}$$

$$Y(s) = \frac{0.28K_P s+0.28K_i}{s^3+0.28K_D s^2+0.28K_p s+0.28K_i} * R(s) \tag{D-4}$$

Putting Y(s) into equation (6.33),

$$E(s) = R(s) - \frac{0.28K_P s+0.28K_i}{s^3+0.28K_D s^2+0.28K_p s+0.28K_i} * R(s) \tag{D-5}$$

$$E(s) = R(s) * \left(1 - \frac{0.28K_P s+0.28K_i}{s^3+0.28K_D s^2+0.28K_p s+0.28K_i}\right) \tag{D-6}$$

$$E(s) = R(s) * \left(\frac{s^3+0.28K_D s^2+0.28K_p s+0.28K_i-0.28K_P s-0.28K_i}{s^3+0.28K_D s^2+0.28K_p s+0.28K_i}\right) \tag{D-7}$$

$$E(s) = R(s) * \left(\frac{s^3+0.28K_D s^2}{s^3+0.28K_D s^2+0.28K_p s+0.28K_i}\right) \tag{D-8}$$

Since $R(s) = \frac{1}{s}$

$$E(s) = \frac{s^2+0.28K_D s}{s^3+0.28K_D s^2+0.28K_p s+0.28K_i} \tag{D-9}$$

By taking the inverse Laplace transform of equation D-9, **e(t)** can be written as

e(t)=7*kd*sum((r3*exp(r3*t))/(75*r3^2 + 14*kd*r3 + 7*kp), r3 in RootOf(s3^3 + (7*kd*s3^2)/25 + (7*kp*s3)/25 + (7*ki)/25, s3)) + 25*sum((r3^2*exp(r3*t))/(75*r3^2 + 14*kd*r3 + 7*kp), r3 in RootOf(s3^3 + (7*kd*s3^2)/25 + (7*kp*s3)/25 + (7*ki)/25, s3))

(D-10)

Appendix E

Formulation of the Fitness Function for PI-PD controller:

Let E(s) is defined as:

$$E(s) = R(s) - Y(s) \tag{E-1}$$

$$\frac{Y(s)}{R(s)} = \frac{(\frac{sK_P+K_i}{s})(\frac{0.28}{s^2})}{1+\left(\frac{sK_P+K_i}{s}\right)\left(\frac{0.28}{s^2}\right)(1+\frac{K_P s+K_D s^2}{K_P s+K_i})} \tag{E-2}$$

$$\frac{Y(s)}{R(s)} = \frac{0.28s+0.28K_i}{s^3+0.28K_D s^2+0.56K_p s+0.28K_i} \tag{E-3}$$

$$Y(s) = \frac{0.28s+0.28K_i}{s^3+0.28K_D s^2+0.56K_p s+0.28K_i} * R(s) \tag{E-4}$$

Putting Y(s) into equation (6.42),

$$E(s) = R(s) - \frac{0.28K_P s+0.28K_i}{s^3+0.28K_D s^2+0.56K_p s+0.28K_i} * R(s) \tag{E-5}$$

$$E(s) = R(s) * (1 - \frac{0.28K_P s+0.28K_i}{s^3+0.28K_D s^2+0.56K_p s+0.28K_i}) \tag{E-6}$$

$$E(s) = R(s) * (\frac{s^3+0.28K_D s^2+0.56K_p s+0.28K_i-0.28K_P s-0.28K_i}{s^3+0.28K_D s^2+0.56K_p s+0.28K_i}) \tag{E-7}$$

$$E(s) = R(s) * (\frac{s^3+0.28K_D s^2+0.28K_p s}{s^3+0.28K_D s^2+0.56K_p s+0.28K_i}) \tag{E-8}$$

Since $R(s) = \frac{1}{s}$

$$E(s) = (\frac{s^2+0.28K_D s+0.28K_p}{s^3+0.28K_D s^2+0.56K_p s+0.28K_i}) \tag{E-9}$$

By taking the inverse Laplace transform of equation E-9, **e (t)** can be written as

e(t) =7*kd*sum((r3*exp(r3*t))/(75*r3^2 + 14*kd*r3 + 14*kp), r3 in RootOf(s3^3 + (7*kd*s3^2)/25 + (14*kp*s3)/25 + (7*ki)/25, s3)) + 7*kp*sum(exp(r3*t)/(75*r3^2 + 14*kd*r3 + 14*kp), r3 in RootOf(s3^3 + (7*kd*s3^2)/25 + (14*kp*s3)/25 + (7*ki)/25, s3)) + 25*sum((r3^2*exp(r3*t))/(75*r3^2 + 14*kd*r3 + 14*kp), r3 in RootOf(s3^3 + (7*kd*s3^2)/25 + (14*kp*s3)/25 + (7*ki)/25, s3))

(E-10)