

# **Performance Evaluation of Different Control Schemes using Various Tuning Methods**

*A Dissertation*

*Submitted in partial fulfillment of the requirements for the award of degree of*

## **Master of Engineering In Electronic Instrumentation and Control**



**Submitted by**

**Rajeev Kumar**

Roll No: 801151021

**Under the Guidance of**

**Dr. Sunil K. Singla**

Assistant Professor (SG)

**Mr. Vikram**

Lecturer

**Department of Electrical and Instrumentation Engineering  
Thapar University**

(Established under the section 3 of UGC act, 1956)

Patiala, 147004, Punjab, India

July 2013

## DECLARATION

I hereby certify that the work is being presented in the dissertation entitled “**Performance Evaluation of Different Control Schemes using Various Tuning Methods**” in fulfillment of award of degree of **Master of Engineering in Electronics Instrumentation and Control** submitted in Electrical and Instrumentation Engineering Department, Thapar University, Patiala is an authentic record of my own work carried under the supervision of Dr. Sunil Kumar Singla, Assistant Professor and Mr. Vikram, Lecturer, Department of Electrical and Instrumentation Engineering, Thapar University, Patiala, Punjab.

**Date:** 3/07/2013

*Rajeev Kumar*

**Rajeev Kumar**

**801151021**

I certify that the above statement made by the student is correct to the best of my knowledge and belief.

**Date:**

*Sunil K. Singla*

**Dr. Sunil K. Singla**  
Assistant Professor (SG)  
Department of Electrical and  
Instrumentation Engineering  
Thapar University, Patiala  
Punjab

*Vikram*  
3/7/2013

**Mr. Vikram**  
Lecturer  
Department of Electrical and  
Instrumentation Engineering  
Thapar University, Patiala  
Punjab

Countersigned By

*S. Ghosh*

**Dr. Smarajit Ghosh**  
Professor & Head of Department  
Department of Electrical  
Instrumentation Engineering  
Thapar University, Patiala  
Punjab

*SSK*  
**Dr. S.K. Mohapatra**  
Dean of Academic Affairs  
Thapar University, Patiala  
Punjab

## **ABSTRACT**

A control system is a device, or set of devices, that manages, commands, directs or regulates the behavior of other device or system to achieve the desired result. Controller design and choosing a particular control configuration/scheme is an important task in a particular control application. Different control configurations/schemes are used in process control for the purpose of controlling a particular application. These are feedback, feedback plus feed forward, cascade and cascade plus feed forward. The objective of feed-forward control is to measure disturbances and compensate error before the controlled variable deviates from the set point. The combined feed forward plus feedback control, can significantly improve performance over simple feedback control whenever a major disturbance can be measured before it affects the process output. Cascade controllers have a distinct advantage over other kinds of controller due to its ability to combine both feedback and feed-forward controls. Combining cascade and feed forward control techniques facilitates additional improvement in control loop performance. For obtaining desired closed loop response different tuning methods have been used. Tuning of PID controller for first order, second order and third order process has been done using Cohen-Coon, Direct Synthesis, Ziegler Nichols, Relay Auto tuning, Modulus Optimum, and Computational Optimization methods. In this work the performance evaluation of different control schemes using various conventional tuning methods such as Ziegler Nichols, Relay auto tuning, Modulus Optimum, and Computational Optimization have been compared. PID controller tuning using Particle Swarm Optimization (PSO) has also been done in this dissertation. The performance comparison between the different methods has been done using the parameters such as rise time, settling time, peak time, overshoot and steady state error. The proposed schemes have been implemented using MATLAB 2011b.

## **ACKNOWLEDGEMENT**

First of all, I would like to thank my mother and father for their unwavering support. Without them I certainly would not where I am today, it would not have been possible for me to quest for post graduate studies and objective for greater affairs in my vacation. I am proud to dedicate this dissertation and all the good things in my life to them.

I would like to express my gratitude towards Dr. Sunil K. Singla, Assistant Professor and Mr. Vikram, Lecturer department of Electrical and Instrumentation Engineering, Thapar University, Patiala for his guidance and support throughout the preparation of this report. I am thankful to Dr. Smarajit Ghosh, Head of Department, Electrical and Instrumentation Engineering, Thapar University, Patiala for his encouragement and support.

I am thankful to all the faculty members and staff members of department of Electrical and Instrumentation Engineering, Thapar University for their support during my academic years.

This section will look incomplete if I fail to thank all my near and dear friends and my family members who stood beside me, understood my academic goals and helped me to achieve it. Last but not by any means least, my heartiest thanks to all the persons, who made me what I am today; word fails to express my feelings for them.

RAJEEV KUMAR

# TABLE OF CONTENTS

| <b>Particulars</b>                                      | <b>Page No.</b> |
|---|-----------------|
| Declaration   | II              |
| Abstract  | III             |
| Acknowledgement   | IV              |
| List of Figures   | VII-IX          |
| List of Tables  | X               |
| <b>CHAPTERS</b>   |                 |
| <b>1. Introduction</b>                                  | <b>1-10</b>     |
| 1.1 Overview  | 1               |
| 1.2 Types of Control Systems                            | 3               |
| 1.2.1 Open loop control system                          | 3               |
| 1.2.2 Closed loop control system                        | 3               |
| 1.3 Types of Control Schemes                            | 4               |
| 1.3.1 Feedback control                                  | 6               |
| 1.3.2 Feed forward control                              | 6               |
| 1.3.3 Feedback plus Feed forward control                | 7               |
| 1.3.4 Cascade control                                   | 8               |
| 1.3.5 Cascade plus Feed forward control                 | 9               |
| 1.4 Objective of Dissertation                           | 10              |
| <b>2. Literature Review</b>                             | <b>12-24</b>    |
| <b>3. PID Controller Tuning</b>                         | <b>25-32</b>    |
| 3.1 Direct Synthesis Method                             | 25              |
| 3.2 Tuning Rules for First Order plus Dead Time Process | 26              |
| 3.2.1 Ziegler-Nichols open loop method                  | 26              |
| 3.2.2 Cohen-Coon parameters                             | 27              |
| 3.3 Ziegler-Nichols Tuning Method                       | 28              |
| 3.4 Relay Auto Tuning Method                            | 28              |
| 3.5 Computational Optimization Method                   | 30              |

|   |  |              |
|---|--|--------------|
| 3.6   | Modulus Optimum Method                           | 31           |
| 3.7   | Performance Index                                | 31           |
| <b>4. Particle Swarm Optimization</b>         |  | <b>33-38</b> |
| 4.1   | Origin   | 33           |
| 4.2   | Concept of PSO                                   | 34           |
| 4.3   | Working of PSO Algorithm                         | 36           |
| 4.4   | Fitness Function                                 | 38           |
| <b>5. Simulations Results and Discussions</b> |  | <b>39-65</b> |
| 5.1   | First order system without delay                 | 39           |
| 5.2   | First order system with delay                    | 41           |
| 5.3   | Second order system without delay                | 44           |
| 5.4   | Second order system with delay                   | 46           |
| 5.5   | Third order system without delay                 | 49           |
| 5.6   | Third order system with delay                    | 52           |
| 5.7   | Third order plant by PSO                         | 54           |
| 5.8   | Third order plant with different control schemes | 59           |
| <b>6. Conclusions and Future Scope</b>        |  | <b>66</b>    |
| <b>References</b>                             |  | <b>68</b>    |
| <b>Publications</b>                           |  | <b>76</b>    |

## LIST OF FIGURES

|                   |   |    |
|-------------------|---|----|
| <b>Figure 1.1</b> | Block diagram of closed loop PID controller for a plant                         | 2  |
| <b>Figure 1.2</b> | PID controller structure  | 2  |
| <b>Figure 1.3</b> | General block diagram of open loop control system                               | 3  |
| <b>Figure 1.4</b> | General block diagram of closed-loop control system                             | 5  |
| <b>Figure 1.5</b> | Block diagram for the feedback control system                                   | 6  |
| <b>Figure 1.6</b> | Block diagram of feed forward control   | 7  |
| <b>Figure 1.7</b> | Block diagram of feedback plus feed forward control                             | 8  |
| <b>Figure 1.8</b> | Block diagram of cascade control  | 9  |
| <b>Figure 1.9</b> | Block diagram of cascade plus feed forward control                              | 10 |
| <b>Figure 3.1</b> | Block diagram of relay auto tuning scheme                                       | 29 |
| <b>Figure 3.2</b> | (a) Plant response under relay feedback with the PID regulator disabled         | 29 |
|                   | (b) Relay response under relay feedback with the PID regulator disabled         | 30 |
| <b>Figure 3.3</b> | PID controller design with computational optimization approach                  | 30 |
| <b>Figure 4.1</b> | Concept of modification of a searching point by PSO                             | 37 |
| <b>Figure 4.2</b> | Flowchart of the PSO based PID control system                                   | 38 |
| <b>Figure 5.1</b> | Unit step response for the plant using Direct Synthesis tuning method           | 40 |
| <b>Figure 5.2</b> | Unit step response for the plant using Computational Optimization tuning method | 40 |
| <b>Figure 5.3</b> | Unit step response for the plant using Cohen Coon tuning method                 | 42 |
| <b>Figure 5.4</b> | Unit step response for the plant using Ziegler-Nichols tuning method            | 42 |
| <b>Figure 5.5</b> | Unit step response for the plant using Relay Auto tuning method                 | 43 |
| <b>Figure 5.6</b> | Unit step response for the plant using Computational Optimization tuning method | 43 |
| <b>Figure 5.7</b> | Unit step response for the plant using Ziegler-Nichols tuning method            | 45 |
| <b>Figure 5.8</b> | Unit step response for the plant using Relay Auto tuning method                 | 45 |

|                    |   |    |
|--------------------|---|----|
| <b>Figure 5.9</b>  | Unit step response for the plant using Computational Optimization tuning method | 46 |
| <b>Figure 5.10</b> | Unit step response for the plant using Ziegler-Nichols tuning method            | 47 |
| <b>Figure 5.11</b> | Unit step response for the plant using Relay Auto tuning method                 | 48 |
| <b>Figure 5.12</b> | Unit step response for the plant using Computational Optimization tuning method | 48 |
| <b>Figure 5.13</b> | Unit step response for the plant using Ziegler-Nichols tuning method            | 50 |
| <b>Figure 5.14</b> | Unit step response for the plant using Relay Auto tuning method                 | 50 |
| <b>Figure 5.15</b> | Unit step response for the plant using Modulus Optimum tuning method            | 51 |
| <b>Figure 5.16</b> | Unit step response for the plant using Computational Optimization tuning method | 51 |
| <b>Figure 5.17</b> | Unit step response for the plant using Ziegler-Nichols tuning method            | 53 |
| <b>Figure 5.18</b> | Unit step response for the plant using Relay Auto tuning method                 | 53 |
| <b>Figure 5.19</b> | Unit step response for the plant using Computational Optimization tuning method | 54 |
| <b>Figure 5.20</b> | Step response with ZN and PSO when population size is 20 and iteration is 2     | 56 |
| <b>Figure 5.21</b> | Step response with ZN and PSO when population size is 20 and iteration is 3     | 56 |
| <b>Figure 5.22</b> | Step response with ZN and PSO when population size is 20 and iteration is 4     | 57 |
| <b>Figure 5.23</b> | Step response with ZN and PSO when population size is 25 and iteration is 2     | 57 |
| <b>Figure 5.24</b> | Step response with ZN and PSO when population size is 25 and iteration is 3     | 58 |
| <b>Figure 5.25</b> | Step response with ZN and PSO when population size is 25 and iteration is 4     | 58 |
| <b>Figure 5.26</b> | MATLAB Simulink diagram for feed forward plus feedback control by ZN method     | 59 |

|                    |   |    |
|--------------------|---|----|
| <b>Figure 5.27</b> | MATLAB Simulink diagram for cascade control by ZN method                            | 60 |
| <b>Figure 5.28</b> | MATLAB Simulink diagram for cascade plus feed forward control by ZN method          | 60 |
| <b>Figure 5.29</b> | MATLAB Simulink diagram for feed forward plus feedback control by RA method         | 61 |
| <b>Figure 5.30</b> | MATLAB Simulink diagram for cascade control by RA method                            | 61 |
| <b>Figure 5.31</b> | MATLAB Simulink diagram for cascade plus feed forward control by RA method          | 62 |
| <b>Figure 5.32</b> | Unit step response for feed forward plus feedback control by using ZN and RA method | 64 |
| <b>Figure 5.33</b> | Unit step response for cascade control by using ZN and RA method                    | 64 |
| <b>Figure 5.34</b> | Unit step response for cascade plus feed forward control by using ZN and RA method  | 65 |

## LIST OF TABLES

|                   |  |    |
|-------------------|--|----|
| <b>Table 3.1</b>  | Ziegler-Nichols Open-Loop Tuning Parameters                            | 27 |
| <b>Table 3.2</b>  | Cohen-Coon Tuning Parameters   | 27 |
| <b>Table 3.3</b>  | Tuning parameters for Ziegler Nichols closed loop ultimate gain method | 28 |
| <b>Table 3.4</b>  | PID Controller Parameters by Modulus Optimum Method                    | 31 |
| <b>Table 5.1</b>  | PID tuning parameters a first order system without delay               | 39 |
| <b>Table 5.2</b>  | Various performance specifications for Direct Synthesis and CO method  | 39 |
| <b>Table 5.3</b>  | PID tuning parameters for a first order system with delay              | 41 |
| <b>Table 5.4</b>  | Various performance criteria for first order plant with delay          | 41 |
| <b>Table 5.5</b>  | PID tuning parameters for second order plant without delay             | 44 |
| <b>Table 5.6</b>  | The performance criteria for second order plant without delay          | 44 |
| <b>Table 5.7</b>  | PID tuning parameters for second order plant with delay                | 46 |
| <b>Table 5.8</b>  | Performance criteria for second order plant with delay                 | 47 |
| <b>Table 5.9</b>  | PID tuning parameters for third order plant without delay              | 49 |
| <b>Table 5.10</b> | Performance criteria for a third order plant without delay             | 49 |
| <b>Table 5.11</b> | PID tuning parameters for third order plant with delay                 | 52 |
| <b>Table 5.12</b> | Performance criteria for third order plant with delay                  | 52 |
| <b>Table 5.13</b> | The result of PID parameters for third order plant                     | 55 |
| <b>Table 5.14</b> | The result of PID performance criteria                                 | 55 |
| <b>Table 5.15</b> | The results of PID tuning parameters                                   | 62 |
| <b>Table 5.16</b> | The results of PID performance criteria                                | 63 |
| <b>Table 5.17</b> | Comparison of performance indices                                      | 63 |

## Introduction

---

### 1.1 Overview

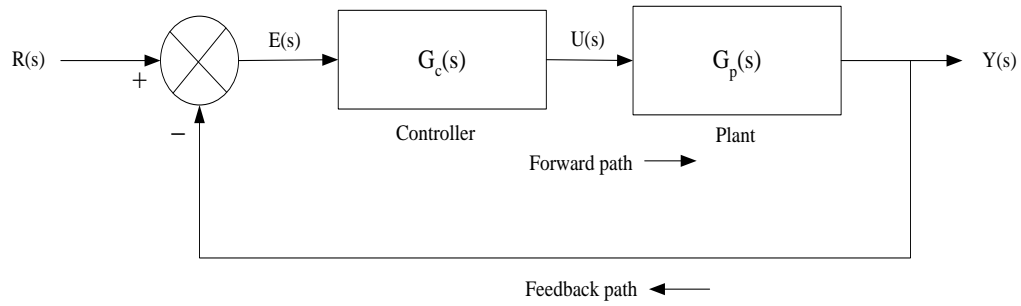
Proportional integral and derivative (PID) controller is most commonly used in industrial control systems. PID controller has three principal control effects. The proportional (P) action gives a change in the input (manipulated variable) directly proportional to the error signal. The integral (I) action gives a change in the input proportional to the integral of error, and its main purpose is to eliminate offset, whereas the derivative (D) action is used to speed up the response or to stabilize the system and it gives a change in the input proportional to the derivative of the error signal. The overall controller output is the sum of the contributions from these three terms [1]. The general form of the PID controller is given below in equation (1.1) [2].

$$U(t)=K_p e(t)+K_i \int e(t)dt+K_D \frac{de(t)}{dt} \quad (1.1)$$

In frequency domain, equation (1.1) can be rewritten as follows [3]:

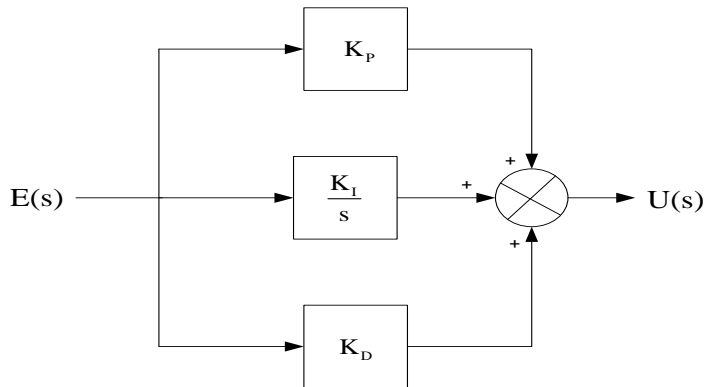
$$U(s)=\left( K_p s + \frac{K_i}{s} + K_D s \right) E(s) \quad (1.2)$$

The signal  $u(t)$  represent the input to the plant,  $e(t)$  represents the tracking error which is the difference between the desired input value and the actual output.  $K_p$ ,  $K_i$  and  $K_D$  are the proportional gain, integral gain and derivative gain respectively. The block diagram of closed loop PID controller for a process is shown in Figure 1.1. The goal of PID controller design is to determine a set of controller parameters ( $K_p$ ,  $K_i$ , and  $K_D$ ) to meet a given set of closed loop system performance requirements such as rise time, peak time, settling time [4].



**Figure 1.1:** Block diagram of closed loop PID controller for a plant

A PID controller improves the transient response of a system by reducing the overshoot, and by shortening the settling time of a system. The roles of the separate parts of the controller are as follows: The proportional part of the controller reduces error responses to disturbances. The integral term eliminates the steady state error and the derivative term dampens the dynamic response and thereby improves stability of the system. The proportional part of the PID controller estimates the system at present; the integral part takes the past into account and the derivative part estimates what will happen in the future [5]. Figure 1.2 shows that PID controller structure in parallel form:



**Figure 1.2:** PID controller structure [6]

A continuous-time PID controller is given by [7].

$$G_c(s) = \left( K_P + \frac{K_I}{s} + K_D s \right) \quad (1.3)$$

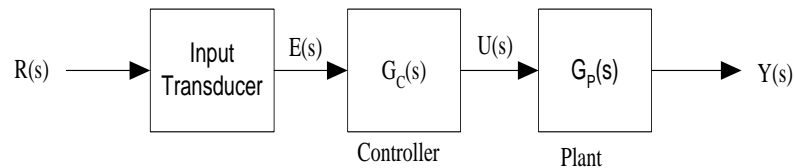
$$G_c(s) = \left( K_P + \frac{K_I}{s} + \frac{K_D s}{1 + T_n s} \right) \quad (1.4)$$

## 1.2 Types of Control Systems

There are two types of control systems namely:

### 1.2.1 Open loop control system

Open loop control system is a system without feedback. That is, in this type of system, sensing of the actual output and comparing of this output (through feedback) with the desired input doesn't take place. The system on its own is not in a position to give the desired output and it cannot take into account the disturbances. In these systems, the changes in output can be corrected only by changing the input manually. A general block diagram of open-loop system is shown in Figure 1.3[8].



**Figure 1.3:** General block diagram of open loop control system [8]

Examples of open loop control system:

1. Automatic washing machine
2. Bread toaster
3. Coffee server
4. Dryer
5. Traffic light controller

### **Advantages of open loop control system**

The various advantages of open loop control system are,

1. Such systems are simple to construct and are cheap.
2. Very much convenient when output is difficult to measure.
3. Such systems are easy from maintenance point of view.
4. Open loop systems are generally stable.

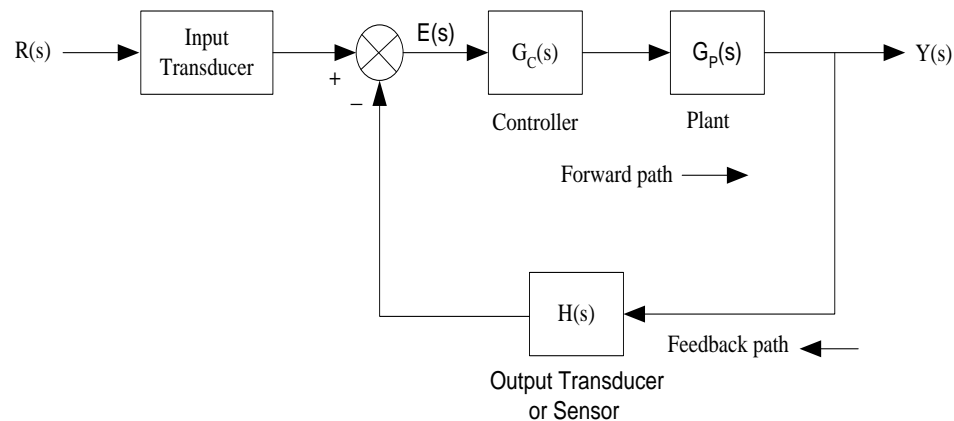
### **Disadvantages of open loop control system**

The various disadvantages of open loop control system are,

1. Such systems are inaccurate and unreliable because accuracy of such systems is totally dependent on the accurate precalibration of the controller.
2. Such systems give inaccurate results if there are variations in the external environment i.e. such systems cannot sense environmental changes.
3. To maintain the quality and accuracy, recalibration of the controller is necessary, time to time.

### **1.2.2 Closed loop control system**

A closed loop control system is a system with feedback. A normal system becomes a closed loop control by including a feedback. In a closed loop system the controlled variable (output) is sensed at every instant of time, feedback and compared with the desired input resulting in an error signal. This error signal directs the control elements in the system to do the necessary corrective action such that the output of the system is obtained as desired. The feedback takes into account the disturbances also and makes the corrective action. These control systems are accurate, stable and less affected by noise. But these are sophisticated and hence costly. They are also complicated to design for stability, give oscillatory response and feedback brings down the overall gain of the control system. A general block diagram of closed-loop control system [8].



**Figure 1.4:** General block diagram of closed-loop control system [8]

Examples of closed loop control system:

1. Human being
2. Automatic control system

### **Advantages of closed loop control system**

The advantages of closed loop control system are,

1. Accuracy of such system is always very high because controller modifies and manipulates the actuating signal such that error in the system will be zero.
2. Such system senses environment changes, as well as internal disturbances and accordingly modifies the error.
3. In such system, there is reduced effect of nonlinearities and distortions.

### **Disadvantages of closed loop control system**

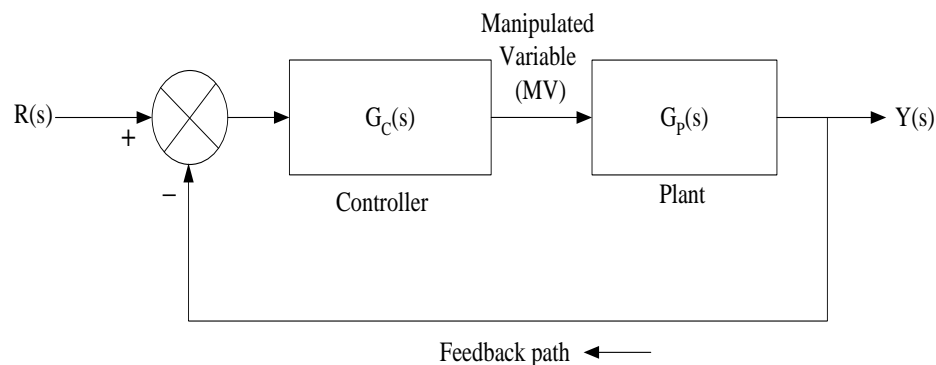
The disadvantages of closed loop control system are,

1. Such system are complicated and time consuming from design point of view and hence costlier.
2. Due to feedback, system tries to correct the error from time to time. Tendency to overcorrect the error may cause oscillations without bound in the system.

## 1.3 Types of Control Schemes

### 1.3.1 Feedback control

A system that maintains a prescribed relationship between the output and the reference input by comparing them and using the difference as a means of control is called a feedback control system. Block diagram for the feedback control system is shown in Figure 1.5. It is a process in which information about the past or the present influences the same phenomenon in the present or future. Feedback control may be defined as the use of difference signals, determined by comparing the actual values of system variables to their desired values, as a means of controlling a system [9]. An everyday example of a feedback control system is an automobile speed control, which uses the difference between the actual and the desired speed to vary the fuel flow rate. Since the system output is used to regulate its input, such a device is said to be a closed-loop control system.

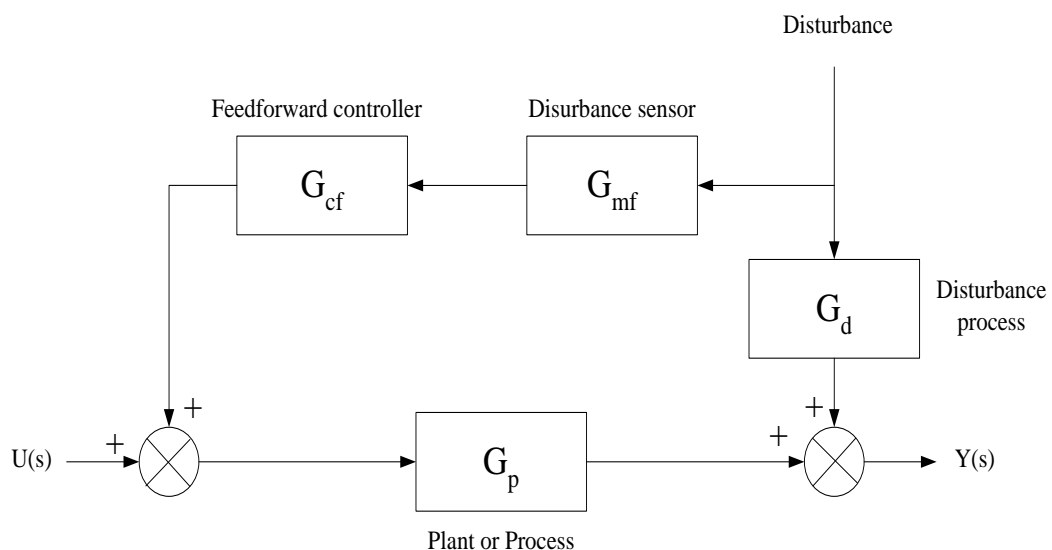


**Figure 1.5:** Block diagram for the feedback control system

### 1.3.2 Feed forward control

The objective of feed-forward control is to measure disturbances and compensate for them before the controlled variable deviates from the set point. The basic idea is to measure a disturbance directly and take control action to eliminate its impact on the process output [10]. How well the scheme will work depends on the accuracy of the process and disturbance models used to describe the system dynamics. Consequently,

feed forward control is normally used in conjunction with feedback control. The feedback controller is used to compensate for any model errors; unmeasured disturbances etc. and ensure offset free control. Feed forward control is always used along with feedback control because a feedback control system is required to track set point changes and to suppress unmeasured disturbances that are always present in any real process. Feed forward control can be used very successfully to improve a control loop's response to disturbances. Feed forward control reacts the moment a disturbance occurs, without having to wait for a deviation in process variable [11]. If any process control loop is subject to large, measurable disturbances, it can benefit greatly from feed forward control. Block diagram of feed forward control system is shown in Figure 1.6.

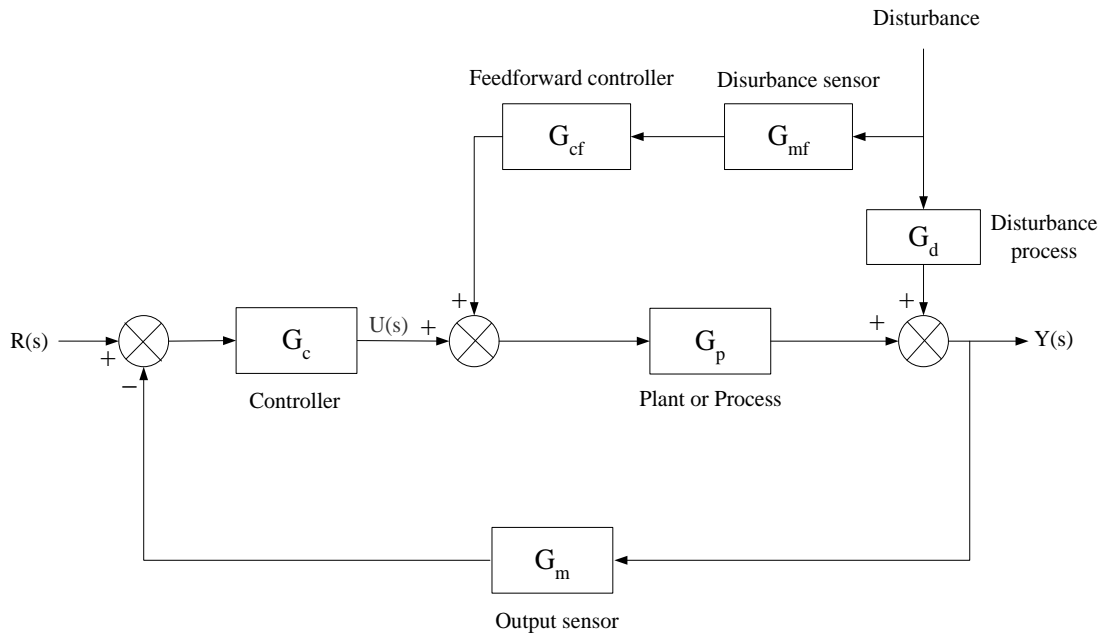


**Figure 1.6:** Block diagram of feed forward control [11]

### 1.3.3 Feedback plus Feed forward control

The combined feed forward plus feedback control, as seen in Figure 1.7 can significantly improve performance over simple feedback control whenever a major disturbance can be measured before it affects the process output. Feed forward control is one of the most widely used advanced control techniques in the process control industry. Combined feed forward plus feedback control can significantly improve performance over simple

feedback control whenever there is a major disturbance that can be measured before it affects the process output. Feed forward can compensate for load upsets before they are detected by the feedback control system as an error. In contrast, feedback controllers can only react to correct for a load upset after an error is detected between the process variable and the set point. Properly tuned feedback plus feed forward controllers can reduce load disturbances to controlled process variable measurement by a factor of 10, better than feedback control alone [12]. The basic principle of feed forward control is to measure the disturbances as they occur, and to make adjustments to the process demand signal of the feedback controller, thus, preventing the disturbance from upsetting the process variable signal being controlled.

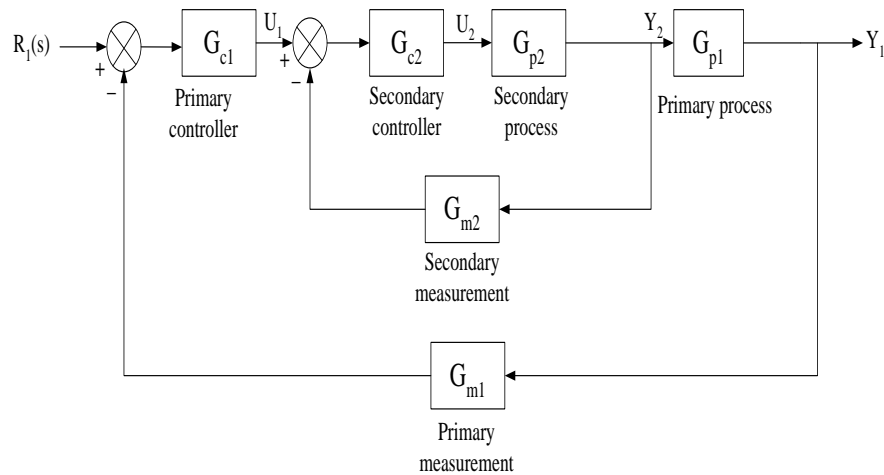


**Figure 1.7:** Block diagram of feedback plus feed forward control [13]

### 1.3.4 Cascade control

The conventional cascade scheme has two distinct functions with two control loops: an inner loop with fast dynamic to eliminate input disturbances, and an outer loop to regulate output performance. Conventionally, they are tuned in a sequential manner. First, the outer loop controller is put on manual and the inner loop controller is tuned.

Subsequently, the inner loop controller is commissioned and the outer loop controller is tuned to complete the tuning process. If the control performance achieved is unsatisfactory, the entire sequence must be repeated [14]. Block diagram for cascade control system is shown in Figure 1.8. Cascade control can improve control system performance over single-loop control whenever either: (1) Disturbances affect a measurable intermediate or secondary process output that directly affects the primary process output that we wish to control; or (2) the gain of the secondary process, including the actuator, is nonlinear. In the first case, a cascade control system can limit the effect of the disturbances entering the secondary variable on the primary output. In the second case, a cascade control system can limit the effect of actuator or secondary process gain variations on the control system performance. Such gain variations usually arise from changes in operating point due to set point changes or sustained disturbances

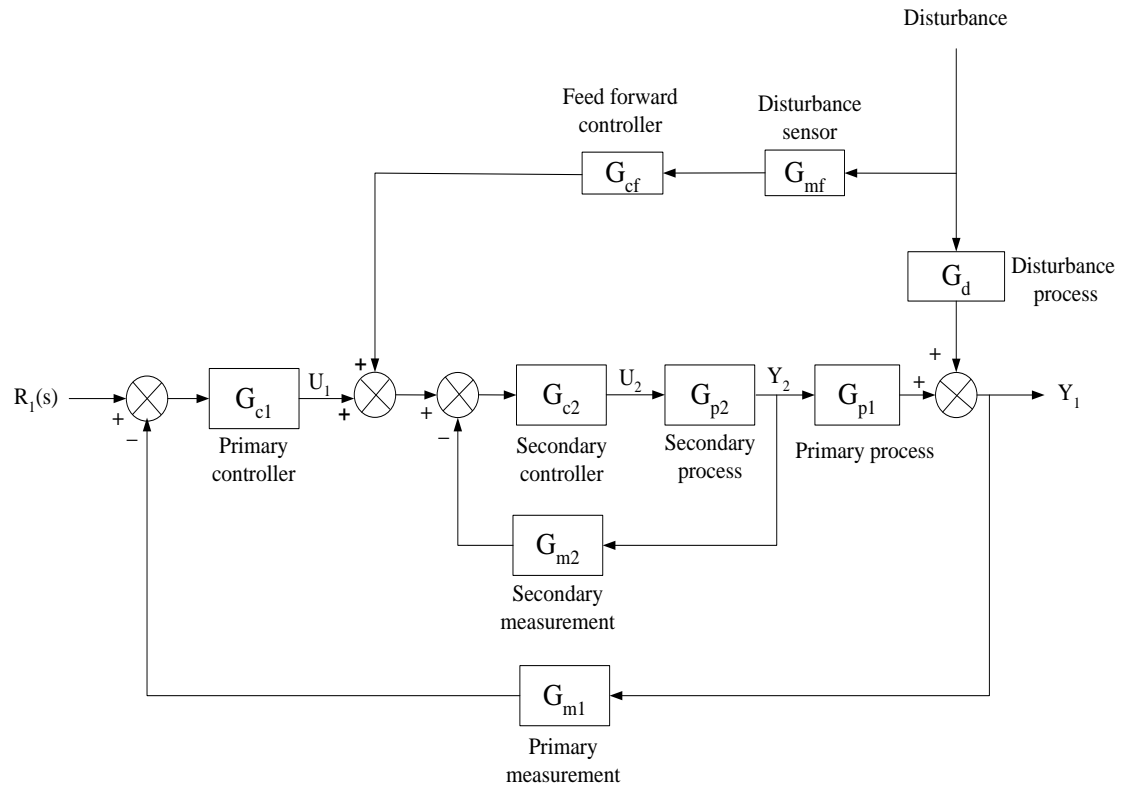


**Figure 1.8:** Block diagram of Cascade control [13]

### 1.3.5 Cascade plus Feed forward control

Combining cascade and feed forward control techniques facilitates additional improvement in control loop performance. The feed forward controller is used to reject feed flow rate disturbances. Cascade controllers have a distinct advantage over other kinds of controller due to its ability to combine both feedback and feed-forward controls.

While feed-forward loops have the potential to adjust the controlled variables to the ideal states, the feedback loops in a mixed system check deviations to make sure the system is on track. The Figure 1.9 below is an example of a mixed cascade control. Cascade, the secondary controller provides precise delivery of the manipulated variable calculated by the feed forward model [14].



**Figure 1.9:** Block diagram of Cascade plus Feed forward control [14]

## 1.4 Objective of Dissertation

The objectives of the dissertation titled “**Performance Evaluation of Different Control Schemes using Various Tuning Methods**” are

1. Tuning of PID controller for first order, second order and third order process using Cohen-Coon, Direct Synthesis, Ziegler Nichols, Relay Auto tuning, Modulus Optimum, and Computational Optimization method.

2. Adding delay to the given transfer function and tuning the PID controller with various methods.
3. Implementing different control schemes such as feedback, feedback plus feed forward, cascade and cascade plus feed forward.
4. Performance evaluation of different control schemes using various conventional tuning methods.
5. Tuning of PID controller for third order plant using Particle Swarm Optimization (PSO).

# CHAPTER 2

## Literature Review

---

Many researches have worked and are still working in the field of PID controller. The contribution of different researches in this field has been mentioned below.

R.E. Brown, G.N. Maliotis and J.A. Gibby [15] proposed a PID self-tuning controller for aluminum rolling mill. The mill process has been modeled as a second order system with one input and one output. The model parameters were estimated on-line using a recursive least squares estimation algorithm. The controller has been designed using pole-placement design technique. The advantage of the proposed scheme is its easy implementation and applicability to complex processes involving unknown parameters, presence of time delays, time varying process dynamics and stochastic disturbances.

Roman W.Swiniarski [16] presented a neural auto-tuner PID controller for real-time applications using Fourier transformation of temporal patterns. The neural auto-tune gives the responses of closed loop relay feedback in the controlled system and produces the updated parameters for PID controller. The advantages of using neural networks create a new chance to provide faster more robust adaptive controllers in case of model uncertainty and disturbances. The neural auto-tuner process temporal patterns, gives the discretized response of closed loop control system with relay feedback, and computes new optimal regulator parameters. Due to the ability of fast, robust processing of temporal oscillatory patterns, the examined neural adaptive controller may find some future application in process control.

Roman W.Swiniarski [17] proposed neuromorphic adaptive PID controller tuner process in real-time using pattern recognition technique. The neuromorphic auto-tuner based on temporal pattern processing in real time and produced the controller parameters. The advantages of using neural networks create a new chance to provide faster more

robust adaptive controllers in case of model uncertainty and disturbances. The auto-tuner analyses the error temporal pattern using discretized step response of open loop control system. In this neuromorphic PID tuning has been compared with Ziegler-Nichols tuning method. The neuromorphic auto-tuner shows a great advantages in perspective real time application due to first temporal pattern analysis.

David W. Augustine and K.S.P Kumar [18] proposed a self-tuning PID controller for control of small to medium sized diesel engines. The proposed controller resembles a parameter adaptive controller when it has been in the tuning phase and like a closed-loop PID controller when it was not in tuning phase. The model used to represent the diesel engine contains nonlinearities and widely ranging parameters. Computer simulations were used to verify that the tuned system's phase and gain margins meet design requirements. The suitable values for the three gains in the PID compensator were usually found by trial and error. The estimator used in the self-tuning PID controller was a discrete square root (DSQRT) filter.

M Zhuang and D.P. Atherton [19] proposed tuning of PID controller with Integral performance criteria. Ziegler-Nichols method was used to determining the PIDF controller parameters. The integral performance criteria provide a good system step response and better methods of achieving controller parameters. The theoretical results were given for a simple plant with time delay and they use these to check the numerical computations which have been achieved using a pade approximation. The advantage of using an integral performance criterion is better closed-loop response of a control system than heuristic tuning methods. The disadvantages of the Integral performance criterion(ISE) was the fact that its minimization often results in a relatively oscillatory step response since the large error which must occur for small time contribute significantly to the performance index.

P. Wang and D.P KWOK [20] proposed auto-tuning of classical PID controllers using an advantaged genetic algorithm. GA do not need analytical performance evaluation so they can use multiple settling time together as well as even knowledge-

based performance indices. The program has been written in c language and compiled by TURBO C. By comparing the results obtained by ZN and GA, GA consistently behaved better than the ZN. The GA based fine tuning approaches have many advantages such as good robustness, simple mechanics, global optimization, bounded random exploration, general information-driven property, large group searching, mixed multi-objective criteria as well as intrinsically similarity to natural world.

H.P Hong, S.J Park, S.J Han, K.Y Cho, Y.C Lim, J.K Park and T.G Kim [21] proposed auto-tuning of PID controller using Fuzzy logic to overcome the disadvantages of the existing auto-tuning PID controllers. The proposed technique has been based on the settling time of the process. The advantages of using Fuzzy logic is that it does not have to use test signals for modulating, it can be applied to a wide range of plants and it robust since it is not affected by the process characteristics. Simulation results reveal that the proposed controller has been tuned automatically to the optimal PID parameters under various operating conditions.

Cheng Ling and Thomas F.Edgar [22] proposed Fuzzy gain scheduling technique that involves implementing several linear controllers over a partitioned process space. The final controller output has been obtained by commutating the weighted average of contributions from all rules. The disadvantages of the proposed technique are the large number of rules required to perform process control. The proposed technique has been described the similarities between this techniques, which we call fuzzy gain scheduling (GS) method. In this FGS performed better than GS. Based on the analysis of PID tuning rules, they present an alternate scheme we call model-based Fuzzy gain scheduling (MFGS) for PID controllers. Based on the simulations, they have concluded that the proposed technique does not always produce a satisfactory set of controller parameters, but the performance of a PID controller with the model-based gain schedule approaches a controller adaptively tuned with the ITAE tuning rules.

Vladimir Bobal and Marek Kubalik [23] proposed self-tuning controller for temperature control of a thermo analyzer. The controller parameters have been derived on

the basis of pole-placement approach. The controller has been designed especially for process control without overshoot of the process output. The proposed technique has been verified with simulation result. The advantage of proposed technique is relatively simple, sufficiently robust and suitable for control of large class of systems.

Robert P.Copeland and Kuldip S.Rattan [24] designed a Fuzzy logic supervisor for PID control of unknown systems, using Ziegler-Nichols tuning rule. Since Ziegler-Nichols methods provide nominal parameters values for PID controller, the desired system response has not been realized without additional tuning. The proposed technique reduced the amount of additional tuning. The addition of a Fuzzy logic supervisor improves the response of a Ziegler-Nichols PID controlled system. Gradual increases in the controller gains, as the system error approaches zero, provide improved system operation.

G.Calsev and R.Gorez [25] proposed an iterative method for automatic closed-loop tuning of PID controllers. The proposed technique has been tested on various open-loop stable plants, with transfer function of order up to nine, with or without dead time and appositive Zero. The iterative method has been compared with Ziegler-Nichols tuning method. From simulation results it has been concluded that the proposed method is easy to use and it leads to control systems with robust stability and performances.

H.J Cho, K.B and B.H Wang proposed [26] genetic algorithms for generating fuzzy rules in fuzzy-PID hybrid control structures. Fuzzy rules in a mixed control structure must be different according to the plant and the conditions in which it is operated. Since fuzzy rule generation is difficult and time consuming procedure, it is required to have a systematic method for constructing appropriate rules. GA's are an efficient and robust tool for generating fuzzy rules in fuzzy-PID hybrid control structures. GA's can construct a set of fuzzy rules that optimize multiple criteria.

J.M. Sousa, R.Babuska and H.B Verbruggen [27] proposed internal model controller (IMC) with a fuzzy model for air-conditioning system. The proposed technique presented an identification procedure for a Takagi-Sugeno Fuzzy-model, which has been

based on fuzzy clustering. The result obtained by proposed technique has been compared with the results obtained with a PID controller. The response obtained with the PID controller was slower than in the case of IMC. Disadvantages of IMC are that its control action is more oscillatory.

Yongling Zheng, Longhua Ma, Liyan Zhang and Qian [28] proposed a simple but feasible robust PID (RPID) tuning using particle swarm optimization (PSO). Minimax criterion guarantees a set of controller parameters to be most optimal to the worst operating condition that may occur within the uncertainty ranges of model parameters. To execute the minimax criterion, a co-evolutionary algorithm based on particle swarm optimization has been proposed. Standard PSO was not sensitive to population sizes so that PSO requires smaller population sizes in comparison to standard GA in general. Therefore, the PSO co-evolutionary algorithm for minimax problem is more economic compared to GA. They implement RPID algorithm in C++ code and compared the different resources required by GA minimax search algorithm and PSO minimax search algorithm. The PSO is an efficient search algorithm suitable for nonlinear global optimization.

Francesco Cupertino, Vincenzo Giordano, David Naso, Luigi Salvatore, and Biagio Turchiano [29] proposed optimization of fuzzy controllers for industrial manipulators via genetic algorithms. The Proposed technique describes a design procedure for the decentralized fuzzy control of a 5-dof robotic manipulator based on genetic algorithms (GAs). GA-tuned fuzzy controllers guarantee better performances than PID in a wide range of operating conditions. The result obtained reveal that the described automatic tuning process much easier. Fuzzy controller ensure better performances than PID controller in different operating conditions, also in presence of unknown external load torque and coupling effects among the joints, in terms of trajectory tracking and current response.

Damir Vrancic, Birgitta Kristiansson and Stanko Strmcnik [30] proposed a reduced MO tuning method for PID controllers. The disturbance rejection magnitude

optimum (DRMO) tuning method for PID controllers provides a relatively fast and non-oscillatory disturbance rejection. The MO method results were a very good closed-loop response for a large class of process models frequently encountered in the process and chemical industries. The modified MO tuning method is called disturbance rejection magnitude optimum (DRMO) method. Advantages of the MO and DRMO tuning methods do not offer any additional tuning parameters. A disadvantage of the proposed method is that the calculations of criteria values require process frequency-response or identification of the process transfer function. Therefore, the calculation of the controller parameters cannot be performed from the measured characteristic areas in time-domain like in original MO and DRMO methods.

Huseyin Atakan Varol and Zafer Bingul [31] proposed a new PID tuning technique using Ant algorithm (AA). In the tuning process, the following cost functions were employed: (I) Integral absolute error (IAE) (II) Integral square error (ISE) (III) A new proposed cost function called reference based error with minimum control effort (RBEMCE). The result obtained from proposed technique has been compared with Ziegler-Nichols (ZN), integral model control (IMC) and Iterative feedback tuning (IFT) methods. The advantage of proposed technique has been its faster settling time, less or no overshoot and higher robustness.

Faisal A.Mohamed and Heikki N. Koivo [32] proposed speed control of power generation plants driven by diesel engine systems with genetic algorithm self-tuning PID controller. Speed control of power generation plants driven by diesel prime-movers was difficult because of a dead time and changes in parameters. Genetic algorithm self-tuning PID controller based on indirect estimation of the dead time has been proposed resulting in fast response at the start up and quick recovery, when a disturbance occurs. The proposed controller controls the system even if the system has time delay or load variations.

A.A Khan and Nishkam Rapal [33] proposed a fuzzy PID controller which can be tuned by carrying the tuning rules from PID domain. The advantage of proposed

technique has been its better performance in terms of rise time, and small overshoot. Step response of the fuzzy PID controller gives low overshoot, less settling time, less rise time and low integral absolute error than the conventional PID controllers. The proposed design and tuning procedure has been suitable for application in industrial environment because of its simplicity.

Junfeng Chen, Ziwu Ren, and XInnan Fan [34] proposed particle swarm optimization with adaptive mutation and its application research in tuning of PID parameters. The standard PSO algorithm has premature and local convergence phenomenon when solving complex optimization problem. To resolve this problem an improved particle swarm optimization has been proposed. The results shows that this approach has been effective and the designed controller has more excellent performance than the controllers designed by the PSO algorithm and the standard genetic algorithm (SGA). From the testing results it can be seen that the proposed IPSO method can avoid the premature convergence problem effectively, and also has good optimization performance. The results shown that the response performance of the IPSO-PID controller is indeed more excellent than the PSO-PID and SGA-PID controller.

M. Willjuice Iruthayarajan and S. Baskar [35] proposed optimization of PID parameters using GA and PSO. PID controller has been designed by minimizing various performance measures such as ISE, IAE, and ITAE for two linear systems. The performance of RGA and PSO has been compared with respect to time response specifications, computation time and statistical performance in 20 independent trials. The performance of PSO was compared with real coded GA with sufficient number of populations by fixing maximum number of functional evaluation as stopping criteria. The performance measured such as ISE, IAE, and ITAE were considered for designing PID controller's two different linear systems. ITAE was preferable for quick settling time.

X.G Duan, H.X Li and Hua Deng [36] proposed a tuning method for fuzzy PID controller based on internal model control (IMC) theory. The parameters of fuzzy PID controller can be designed analytically and automatically. With the help of simulation,

the proposed technique has been shown to be very effective in the set-point tracking control for a first order plus dead time process. The proposed technique has been compared with conventional counterpart; proposed technique is more robust to the process variation.

Dongyun Wang and Guan Wang [37] proposed parameters optimization of fuzzy controller based on improved particle swarm optimization (PSO). The tuning of the scaling factors of the fuzzy PID controller for the inverted pendulum has been investigated. The advantage of the PSO algorithm has been easy implementation, robustness to control parameters, and computational efficiency when compared with other mathematical techniques. The PSO can be easily applied to nonlinear and non-continuous optimization problems. Chaotic sequence is used to update weight and the position of the particles.

Salah Kermiche and H.A Abbassi [38] proposed fuzzy PID controller parameters optimized by genetic algorithms. Fuzzy inference system has been adopted to determine the values of the weight that multiplies the set-point for the proportional action. This contribution introduces genetic algorithms to change the shapes of the membership of the fuzzy controller by changing their parameters. The advantage of fuzzy logic controller has been its simple implement, easy to adapt to process set-point operation. It is robust to the disturbance that may affect the controlled process.

YANG BOLI, Wan-Zhou, and YANG Feng [39] proposed PSO-PID tuning method for time-delay process. The proposed method has been used to design PID controllers for stable first order and high order systems with time delay. The proposed method originates from process with small time delay: however, it is still effective even if the time delay is quite large. Simulation results shown that the proposed method gives significantly better dynamic performances than IMC-PID method. The proposed is still effective even if the time delay is quite large compared with other algorithms and strategies for designing PID controllers, the proposed method gives better dynamic performance.

Liu Fan and Meng Joo [40] proposed a method of designing proportional-integral-derivative (PID) controller based on genetic algorithms (GA). Ziegler-Nichols tuning formula has been used for predicting the range of gain coefficients of GA-PID controller. The proposed method has been used for optimizing the gain coefficients of PID controller has considerably improved the performance of PID controller. The performance of the proposed controller was evaluated by simulation on a second-order and a third order control system. Simulation results demonstrate that the proposed controller performs ZN-PID and fuzzy PID controllers in terms of dynamics and statics characteristic.

Guohan Lin and Guofan Liu [41] proposed PID controller tuning using adaptive genetic algorithms. The proposed techniques have been compared with traditional optimizations methods. Simulations with different process show that the gains obtained using proposed technique provide better performance than those obtained by the classical Ziegler-Nichols (ZN) method classical genetic algorithms (CGA) method. The basic structure of the proposed controller is similar to the CGA PID controller. The PID gains obtained by the ZN tuning methods are used to calculate the initial gains of the proposed controller. Advantage of proposed technique is simple in structure and its computational task is so small that the online adaptation is easy.

H.M. Asifa and S.R.Vaishnav [42] proposed a particle swarm optimization (PSO) method for determining the optimal proportional-integral-derivative (PID) controller parameters to improve the step response of a third order system. The simulation has been done using MATLAB. The result has been compared with the performance of PID controller tuned using conventional methods like Ziegler-Nichols, Tyreus Luyben and internal model control. The proposed method is better than the conventional methods. The PSO based PID controller has superior features like easy implementation, stable convergence characteristic and good computational efficiency. The advantage of proposed method is improving the steady state error, reduction in peak overshoot and settling time.

GAN Shu-chuan and Guo Hui [43] proposed the research of PID self-tuning based on fuzzy genetic algorithm. The proposed method has been used to optimize the PID parameters of temperature controller in heating furnace. After the genetic algorithm executed 200 iterations, the presetting value of temperature was set 60 degree in two intelligent nodes. The experiment indicated that the PID controller base on the fuzzy genetic algorithm obtained the less error, the less adjustment time and the less change in temperature.

Mazidah Tajjudin, Ramli Adnan, Norlela Ishak, Mohd and Hashimah Ismail [44] proposed an optimal PID tuning PSO. The proposed method has been utilized to search for optimum  $K_P$ ,  $K_I$  and  $K_D$  that will minimized some objective functions typically IAE, ITSE. The proposed method presents simulation results on a model reference input for a PSO-tuned PID to achieve better output performance. Simulation was done on a simple first-order model towards three reference model with different time constant. The proposed structure has been proved to be superior in terms of consistency in results and the control signal also provides better accuracy.

Geetha. M, Balajee. K. A and Jovitha Jerome [45] proposed an optimal tuning of virtual feedback PID controller for a continuous stirred tank reactor (CSTR) using particle swarm optimization (PSO) algorithm. The proposed method has been based on the optimal tuning of virtual feedback PID control for a CSTR system using PSO algorithm for minimum integral square error (ISE) condition. The optimized PID tuning parameters of the CSTR system has been obtained using Ziegler-Nichols tuning, genetic algorithm and particle swarm optimization. The results from all the three tuning techniques have been compared against each other. By comparing all the three methods, it was found that PSO algorithm was the best implemented.

Y.C. Kim, L.H. Keel and S.P. Bhattacharyya [46] proposed an approach to directly control the transient response of linear time-invariant systems. They proved that the principal characteristic ratio and generalized time constant, namely the two parameters  $a$  and  $r$  can be used to independently characterize or control the system

overshoot to a step input and the speed of response, respectively. These formulas have been used to develop a procedure to design a state feedback plus feed forward controller for a minimum phase plant that provides “good” transient response, namely one with specified overshoot and specified rise time or speed of response.

J.S Wang and C.S. George Lee [47] presented the utilization of a self-adaptive recurrent neuro-fuzzy control as a feed forward controller and a proportional-plus-derivative (PD) control as a feedback controller for controlling an autonomous underwater vehicle (AUV) in an unstructured environment. A systematic self-adaptive learning algorithm, consisting of a mapping- constrained agglomerative clustering algorithm for the structure learning and a recursive recurrent learning algorithm for the parameter learning, has been developed to construct the recurrent neuro-fuzzy system to model the inverse dynamics of an AUV with fast learning convergence. Computer simulations of the proposed recurrent neuro-fuzzy control scheme and its performance comparison with some existing controllers have been conducted to validate the effectiveness of the proposed approach.

Serdar Yuksel, Haitham Hindi and Lara Crawford [48] proposed an Optimal Tracking with Feedback-Feed forward Control Separation over a Network. The controller consists of a central decision maker and an on-site controller, which were connected through a discrete noiseless channel. The reference path has been available noncausally to the central decision maker and the on-site controller has access to noisy observations from the plant and the reference information provided by the central decision maker. The proposed method provided the optimal control using dynamic programming and show that the optimal controller can be separated into a non-causal feed forward term plus a feedback term.

A. E. Leon, J. A. Solsona, C. Busada, H. Chiacchiarini and M. I. Valla [49] proposed a novel control strategy for Voltage Source Converters (VSCs). The proposed method consists of a nonlinear feedback controller based on feedback linearization plus feed forward compensation of the load current. Energy function and a quadrature current

have been considered as outputs, such that zero dynamics was avoided. In order to improve the performance of the whole system and to diminish the output capacitor size an estimate of the load current has been feed forwarded. Simulation results have been performed in presence of parameters uncertainties and noisy measurements. The main advantages are: outputs are chosen such that complete feedback linearization has been obtained avoiding zero dynamics. In addition, feed forward compensation allows improving the performance when a capacitor value is fixed.

Wei Tang, Qian Feng, Mengxiao Wang, Qian Hou and Leilei Wang [50] proposed a novel control method based on expert system (ES). A feed forward and feedback plus PID control scheme based on expert system for DO (dissolved oxygen) value control has been employed, by which different control rules were made according to the derivative of DO values in different stages. This control strategy has been embedded into a wastewater aerobics treatment process control system running in China and resulted in notable profits. The advantage of proposed method are simple structure, short settling time, high precision, good stability and small overshoot.

I.S Baruch and R.G Guerra [51] proposed a Fuzzy-Neural Multimodal (FNMM) identification and control system for decentralized control of distributed parameter anaerobic wastewater treatment digestion bioprocess, carried out in a fixed bed and a recirculation tank. The distributed parameter analytical model of the digestion bioprocess has been reduced to a lumped system using the orthogonal collocation method, applied in three collocation points, which were used as centers of the membership functions of the fuzzyfied space variable of the plant. The states of the proposed FNMM identifier have been implemented by a direct feedback-feed forward hierarchical FNMM controller. The advantage of proposed method results good convergence, and precise reference tracking outperforming the optimal control.

Wei Tang, Zhongfeng Wang, Qian Feng and Mengxiao Wang [52] proposed a fuzzy expert control strategy based on DCS. A feed forward and feedback plus PID control scheme based on expert system (ES) for dissolved oxygen (DO) value control

was designed, by which different control rules were made according to the derivative of DO values in different stages. Based on SIMATIC PCS7, the control system has been designed for the aerobic section of APMP wastewater treatment. The application indicates that the proposed control method was feasible and valuable. The advantages of proposed method has been its simple structure, short settling time, high precision, good stability and small overshoot. The ES does not rely on the mathematical model. The energy consumption of APMP process was greatly reduced.

E.H Gurban and G.D Andreescu [53] proposed a PID Controller Tuning for Greenhouse Climate with Feedback-Feed forward Linearization and Decoupling. The proposed method presented an equivalent greenhouse climate model based on feedback feed-forward compensation technique responsible for linearization, decoupling and disturbance compensation of the greenhouse complex model. Based on this equivalent model reduced to integral plus dead time decoupled processes, a comparison study of associated PI/PID controllers employing different tuning techniques has been performed by simulation. The system responses using these tuning methods for set point step change in temperature and absolute humidity were compared by simulation, the smallest settling time has been obtained for Ziegler-Nichols PID tuning rules, and the smallest overshoot for Internal Model Control based IAE/ISE optimal tuning.

In this chapter we have summarized the tuning of PID controller with different tuning methods such as Ziegler-Nichols, Particle Swarm Optimization, Genetic Algorithm, Fuzzy Logic etc. Our main focus on Particle Swarm Optimization method in which we saw that tuning of PID controller for higher order plant has been done with large number of iteration. But in this dissertation tuning of PID controller for higher order plant has been done with less number of iteration.

## CHAPTER 3

# PID Controller Tuning

---

Tuning a controller is a method of determining the parameters of a PID controller for a given system. A PID controller is described by three parameters;  $K_P$ ,  $K_I$  and  $K_D$ . Different tuning methods have been discussed below.

### 3.1 Direct Synthesis Method

This is a model based tuning technique. It uses an identified process model in conjunction with a user specified closed loop response characteristic [54].

Given first order process:

$$G_p(s) = \frac{K_P}{\tau_p s + 1} \quad (3.1)$$

Then according to direct synthesis method the controller transfer function is given by:

$$\begin{aligned} G_C(s) &= \frac{\frac{1}{\lambda s + 1}}{\frac{K_P}{\tau_p + 1} \left( 1 - \frac{1}{\lambda s + 1} \right)} \\ &= \frac{\tau_p s + 1}{K_P \lambda s} \end{aligned} \quad (3.2)$$

Also,

$$G_C(s) = \frac{\tau_p}{K_P \lambda} \left( \frac{\tau_p s + 1}{\tau_p s} \right) \quad (3.3)$$

Where,  $\lambda$  is a filter tuning parameters. The larger value of  $\lambda$  results faster closed loop response while smaller value of  $\lambda$  results slower closed loop response. By comparing equation (3.1) and (3.3) we get,

$$K_C = \frac{\tau_P}{K_P \lambda}$$

$$\tau_I = \tau_P$$

## 3.2 Tuning Rules for First Order plus Dead Time Processes

### 3.2.1 Ziegler-Nichols open-loop method [13]

Ziegler and Nichols proposed tuning parameters for a process that has been identified as integrator plus time delay based on the open loop process step response.

$$g_p(s) = \frac{K e^{-\theta s}}{s}$$

Since first order plus time delay processes have a maximum slope of  $K = \frac{K_P}{\tau_P}$  at  $t = \theta$

for a unit step input change, these same rules can be used for first order plus time delay processes,

$$g_p(s) = \frac{K_P e^{-\theta s}}{\tau_P s + 1}$$

Their recommended tuning parameters, which should give roughly quarter-wave damping, are shown in Table 3.1.

**Table 3.1:** Ziegler-Nichols Open-Loop Tuning Parameters [13]

| Controller | $K_C$  | $\tau_I$     | $\tau_D$     |
|------------|--|--------------|--------------|
| P          | $\frac{1}{K\theta}$ or $\frac{\tau_p}{K_p\theta}$      | -----        | -----        |
| PI         | $\frac{0.9}{K\theta}$ or $\frac{0.9\tau_p}{K_p\theta}$ | 3.3 $\theta$ | -----        |
| PID        | $\frac{1.2}{K\theta}$ or $\frac{1.2\tau_p}{K_p\theta}$ | 2 $\theta$   | 0.5 $\theta$ |

### 3.2.2 Cohen-Coon parameters [13]

The method developed by Cohen and Coon (1953) is based on a first-order plus time-delay process model. The tuning parameters as a function of the model parameters are shown in Table 3.2. The major problem with the Cohen-Coon parameters is that they tend to be not very robust; i.e., a small change in the process parameters can cause the closed-loop system to become unstable.

**Table 3.2:** Cohen-Coon Tuning Parameters [13]

| Controller | $K_C$  | $\tau_I$  | $\tau_D$                                       |
|------------|--|---|--|
| P          | $\frac{\tau_p}{K_p} \left[ 1 + \frac{\theta}{3\tau_p} \right]$                 | -----   | -----  |
| PI         | $\frac{\tau_p}{K_p\theta} \left[ 0.9 + \frac{\theta}{1.2\tau_p} \right]$       | $\frac{\theta \left[ 30 + 3 \frac{\theta}{\tau_p} \right]}{9 + 20 \frac{\theta}{\tau_p}}$ | -----  |
| PID        | $\frac{\tau_p}{K_p\theta} \left[ \frac{4}{3} + \frac{\theta}{4\tau_p} \right]$ | $\frac{\theta \left[ 32 + 6 \frac{\theta}{\tau_p} \right]}{13 + 8 \frac{\theta}{\tau_p}}$ | $\frac{4\theta}{11 + 2 \frac{\theta}{\tau_p}}$ |

### 3.3 Ziegler Nichols Tuning Method

The most popular tuning methodology was proposed by Ziegler and Nichols in 1942 [55]. The closed-loop tuning method requires the determination of the ultimate gain and ultimate period. This can be achieved by adjusting the controller gain till the system undergoes sustained oscillations at the ultimate gain or critical gain ( $K_u$ ), whilst maintaining the integral time constant at infinity and the derivative time constant at zero. The Ziegler-Nichols tuning method is based on the determination of process inherent characteristics such as the process gain, process time constant and process dead time. These characteristics are used to determine the controller tuning parameters.

**Table 3.3:** Tuning parameters for Ziegler Nichols closed loop ultimate gain method [56].

| Controller | $K_p$      | $T_I$      | $T_D$       |
|------------|------------|------------|-------------|
| P          | $0.5 K_u$  | -----      | -----       |
| PI         | $0.45 K_u$ | $0.83 P_u$ | -----       |
| PID        | $0.6 K_u$  | $0.5 P_u$  | $0.125 P_u$ |

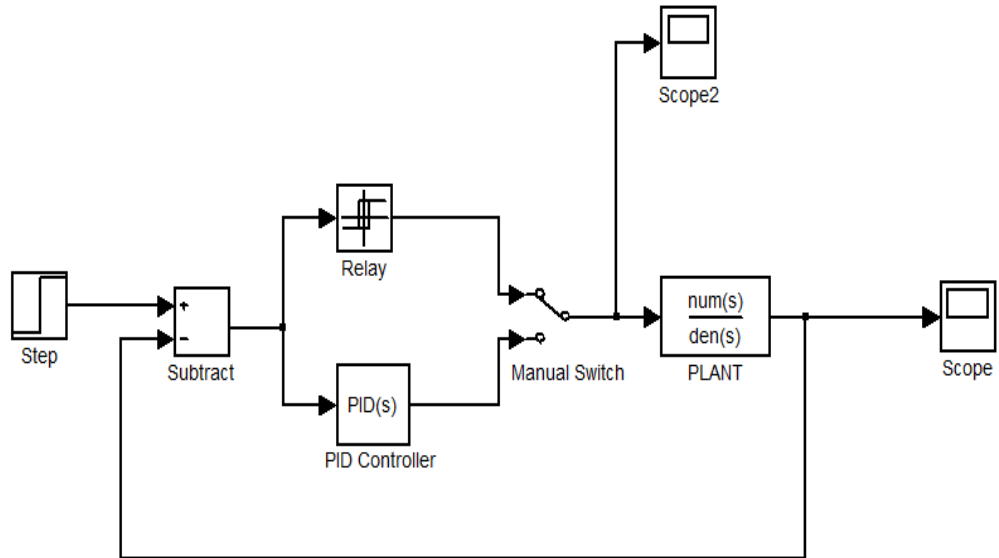
### 3.4 Relay Auto Tuning Method

Relay-based auto tuning is a simple way to tune PID controllers that avoids trial and error, and minimizes the possibility of operating the plant close to the stability limit [57]. Block diagram of simple feedback auto tuning system shown in Figure.3.1. Auto tuning is based on the idea of using an on/off controller (called a relay controller). Initially, the plant oscillates without a definite pattern around the nominal output value until a definite and repeated output response can be identified. When the desired response pattern has been reached the oscillation period ( $P_u$ ) and the amplitude (A) of the plant response can be measured and used for PID controller tuning. In fact, the ultimate gain can be computed as:

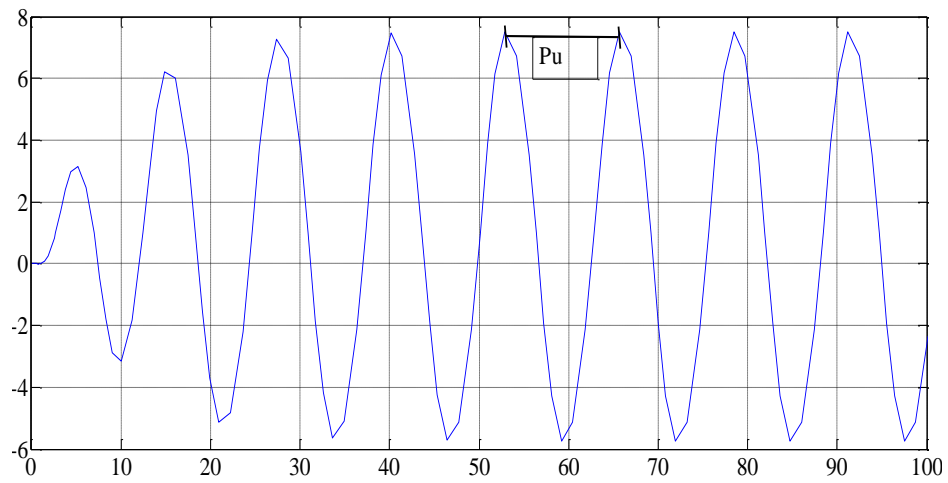
$$K_u = \frac{4h}{\Pi A} \quad (3.4)$$

Where, h=amplitude of the PID controller output

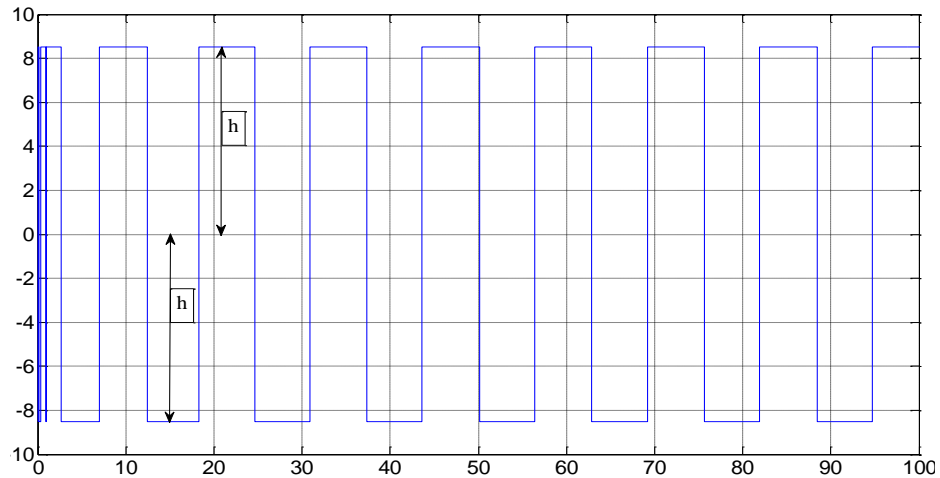
A=amplitude of the plant response



**Figure 3.1:** Block diagram of relay auto tuning scheme



(a)



(b)

**Figure 3.2:** (a) Plant response under relay feedback with the PID regulator disabled

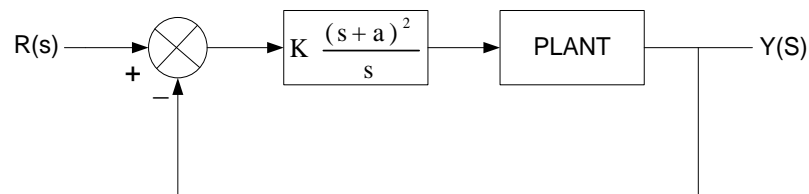
(b) Relay response under relay feedback with the PID regulator disabled

### 3.5 Computational Optimization Method

In this method an optimal set (or optimal sets) of values of PID controller to satisfy the transient response specifications is required to be obtained [58]. The PID controller with computational optimization approach has been shown in Figure 3.3. The objective is to find the combination of gain ‘K’ and ‘a’ such that the closed-loop system will have minimum rise time, settling time, peak time and overshoot. For designing the PID controller first specify the region to search for appropriate K and a. The values of K and a must specify

$$0.4 \leq K \leq 1 \quad (3.5)$$

$$0.08 \leq a \leq 0.3 \quad (3.6)$$



**Figure 3.3:** PID controller design with computational optimization approach

### 3.6 Modulus Optimum Method

The modulus optimum (MO) method for optimization of regulators can be applied in a wide variety of cases in the control field. Modulus Optimum (MO) method is based on the transfer function of set point  $G_{ref}(s)$  [59]. In ideal case the transfer function would be  $G_{ref}(s) = 1$ , i.e. step response of process variable is equal to set point. In frequency domain it corresponds with condition given in equation (3.7).

$$G_{ref}(j\omega) = 1 \Rightarrow |G_{ref}(j\omega)| = A_{ref}(\omega) = 1 \quad (3.7)$$

This condition cannot be satisfied in reality; however it can be proven that control process ends the fastest when amplitude characteristics  $A_{ref}(j\omega)$  will be flat at first and then monotonically decrease. The setting of PID parameters  $K_p$ ,  $T_I$  and  $T_D$  by MO method is sorted in the Table 3.4 for practical use and it depends on the type of controlled plant.

**Table 3.4:** PID Controller Parameters by Modulus Optimum Method

| Model of Controlled plant  | $K_p$               | $T_I$       | $T_D$                      |
|--|---------------------|-------------|----------------------------|
| $\frac{k}{(T_1s+1)(T_2s+1)(T_3s+1)}$ <p>Where,</p> $T_1 \geq T_2 \geq T_3$ | $\frac{T_1}{2kT_3}$ | $T_1 + T_2$ | $\frac{T_1T_2}{T_1 + T_2}$ |

### 3.7 Tuning Based on Minimization of Various Performance Indices

The followings are some commonly used criteria based on the integral error for a step set point or disturbance response [60].

The integral of absolute error (IAE) performance criteria is

$$J_{IAE} = \int_0^{\infty} |e(t)| dt \quad (3.8)$$

IAE gets the absolute value of the error to remove negative error components. Generally IAE is good for simulation studies.

The integral square error (ISE) performance criteria is

$$J_{ISE} = \int_0^{\infty} |e^2(t)| dt \quad (3.9)$$

ISE squares the error to remove negative error components. ISE discriminates between over-damped and under damped Systems.

The integral of time multiplied by the absolute error (ITAE) performance criteria is

$$J_{ITAE} = \int_0^{\infty} t |e(t)| dt \quad (3.10)$$

The ITAE weights the error with time and hence emphasizes the error values later on in the response rather than the initial large errors.

The integral of time multiplied square error criterion (ITSE) performance criteria is

$$J_{ITSE} = \int_0^{\infty} t |e^2(t)| dt \quad (3.11)$$

The ITSE weights the square error with time and much more emphasizes the error values rather than the ITAE performance criteria [61].

# Particle Swarm Optimization (PSO)

---

### 4.1 Origin

Particle Swarm Optimization (PSO) was developed by Dr.Kennedy and Dr.Eberhart in 1995. It is a kind of global optimization technology and belongs to the category of swarm intelligence method [62]. The PSO is a population based search algorithm based on the simulation of the social behavior of birds, bees or a school of fishes. This algorithm originally intends to graphically simulate the graceful and unpredictable choreography of a bird folk. Each individual within the swarm is represented by a vector in multidimensional search space. This vector has also one assigned vector which determines the next movement of the particle and is called the velocity vector. The PSO algorithm also determines how to update the velocity of a particle. Each particle updates its velocity based on current velocity and the best position it has explored so far; and also based on the global best position explored by swarm [63]. PSO has been found to be simple, effective and robust in solving problems with nonlinearity, non-differentiability, and multidimensional optimization [64]. The PSO uses a simple mechanism that imitates their swarm behaviors to guide the particles to search for globally optimal solutions. Hence it is also a population based iterative algorithm. Owing to its simplicity of implementation and ability to quickly converge to a reasonably good solution, the PSO has been successfully applied in solving many real-world optimization problems [65]. Particle swarm optimization has been enormously successful in several and various industrial domains. It has been used across a wide range of engineering applications. These applications can be summarized around domains of robotics, image and signal processing, electronic circuits design, communication networks, but more especially the domain of plant control design [66].

## 4.2 Concept of PSO

Original concept of PSO came from the study of simulating behavior of bird flocking to look for food. A possible solution for each problem can be represented as a particle that is just like a bird flocking in a  $D$ -dimensional searching space. Each individual particle has a fitness value that is evaluated by a fitness function to pick a good experience for itself and population respectively. The particles of population is initialized randomly first. A particle changed its searching direction based on two values or experiences during each iteration. The first one is the best searching experience of individual so far and it is called  $pbest$ . Another one is the best result obtained so far by any particle in the population and it is called  $gbest$  [67]. Imagine a source of insects or a school of fish. If one sees a desirable path to go (e.g. food, protection etc.) the rest of the swarm will be able to follow quickly even if they are on the opposite side of the swarm. On the other hand, in order to facilitate felicitous exploration of the search space, typically one wants each particle to have a certain level of randomness in their movement, so that the movement of the swarm has certain explorative capability the swarm should be influenced by the rest of the swarm but also should independently explore to a certain extent. PSO uses a population of solution, called particles which fly through the search space, with directed velocity vectors to find better solution. Each particle keeps track of its coordinates in the problem space which are associated with the best solution (fitness) it has achieved so far. The fitness value is also stored. This value is called  $pbest$  (personal best). Another ‘best’ value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the immediate neighborhood of the particle. This location is called  $lbest$  (local best). When a particle takes all the population as its topological neighbors, the best value is called  $gbest$  (global best). PSO concept consists of, at each time step, changing the velocity (accelerating) of each particle towards its  $pbest$  and  $lbest$  locations. Acceleration is weighted by a random term, with separate random numbers being generated for acceleration toward  $pbest$  and  $lbest$  locations. Consider an optimization problem of  $D$  variables. A swarm of  $M$  particles is initialized in which each particle is assigned a random position in the  $D$ -dimensional hyperspace such that each particle’s position

corresponds to a candidate solution for the optimization problem. Let denote a particle's position (coordinate) and  $V$  denote the particle's flight velocity over a solution space. Each individual  $X$  in the swarm is scored using a scoring function that obtains a score (fitness value) representing how good it solves the problem. The best previous position of a particle is  $Pbest$ . The index of the best particle among all particles in the swarm is  $Gbest$ . Each particle records its own personal best position ( $Pbest$ ), and knows the best positions found by all particles in the swarm ( $Gbest$ ). Then, all particles that fly over the  $D$ -dimensional solution space are subject to updated rules for new positions, until the global optimal position is found [68]. Each particle represents a candidate solution to the problem and it has its own position and velocity. For an  $N$ -dimensional problem, the position and velocity can be specified by a  $M \times N$  matrix as follows [69]:

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1N} \\ x_{21} & x_{22} & \cdots & x_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ x_{M1} & x_{M2} & \cdots & x_{MN} \end{bmatrix}$$

$$V = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1N} \\ v_{21} & v_{22} & \cdots & v_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ v_{M1} & v_{M2} & \cdots & v_{MN} \end{bmatrix}$$

Where  $X$  and  $V$  are the position matrix and the velocity matrix respectively. Number of particles in the swarm is denoted by  $M$ . Each row of the position matrix represents a possible solution to the optimization problem [70]. The velocity of each particle depends on the distance of the current position to the positions that resulted in good fitness values. To update the velocity matrix at each iteration, every particle should know its personal best and the global best position vectors. The personal best position vector defines the position at which each particle attained its best fitness value up to the present iteration [71]. The personal best positions can be defined by the matrix

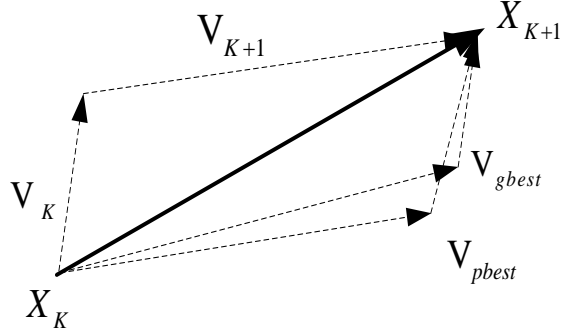
$$Pbest = \begin{bmatrix} pbest_{11} & pbest_{12} & \cdots & pbest_{1N} \\ pbest_{21} & pbest_{22} & \cdots & pbest_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ pbest_{M1} & pbest_{M2} & \cdots & pbest_{MN} \end{bmatrix}$$

Each row in the *Pbest* matrix represents the corresponding particle's personal best position. The global best position vector defines the position in the solution space at which the best fitness value was achieved by all particles, and is defined by [72].

$$Gbest = [gbest_1 \quad gbest_2 \quad \cdots \quad gbest_N]$$

### 4.3 Working of PSO Algorithm

As stated before, PSO simulates the behavior of bird flocking. Suppose the following scenario: a group of birds is randomly searching for food in an area. There is only one piece of food in the area being searched. Not all the birds know where the food is. However, during every iteration, they learn via their inter-communications, how far the food is. Therefore, the best strategy to find the food is to follow the bird that is nearest to the food. PSO learned from this bird-flocking scenario, and used it to solve optimization problems. In PSO, each single solution is a "bird" in the search space. We call it "particle". All of particles have fitness values which are evaluated by the fitness function (the cost function to be optimized), and have velocities which direct the flying of the particles. The particles fly through the problem space by following the current optimum particles. PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. During every iteration, each particle is updated by following two "best" values. The first one is the position vector of the best solution (fitness) this particle has achieved so far. The fitness value is also stored. This position is called *pbest*. Another "best" position that is tracked by the particle swarm optimizer is the best position, obtained so far, by any particle in the population. This best position is the current global best and is called *gbest* [73].



$X_K$  : Current searching point

$X_{K+1}$  : Modified searching point

$V_K$  : Current velocity

$V_{K+1}$  : Modified velocity

$V_{pbest}$  : Velocity based on pbest

$V_{gbest}$  : Velocity based on gbest

**Figure 4.1:** Concept of modification of a searching point by PSO [74]

The velocity and position matrix are updated such that each particle takes the path of a damped oscillatory movement toward its personal best and the global best positions in every iteration. The velocity and position matrix is updated according to following rules [75].

$$v_{m,n}^{(t+1)} = w * v_{m,n}^{(t)} + c_1 * rand() * (pbest_{m,n}^{(t)} - x_{m,n}^{(t)}) + c_2 * rand() * (gbest_n^{(t)} - x_{m,n}^{(t)}) \quad (4.1)$$

$$X^{(t+1)} = X^{(t)} + V^{(t)} \quad (4.2)$$

Where,  $m=1, 2, 3, \dots, M$

$n=1, 2, 3, \dots, N$

$t$  and  $t+1$  denote the time index of the current and next iterations respectively. Cognitive coefficient  $c_1$  and social coefficient  $c_2$  are constants known as acceleration coefficients, and  $rand()$  uniformly distributed random numbers in the range  $[0, 1]$ .  $w$  is the inertia weight factor.

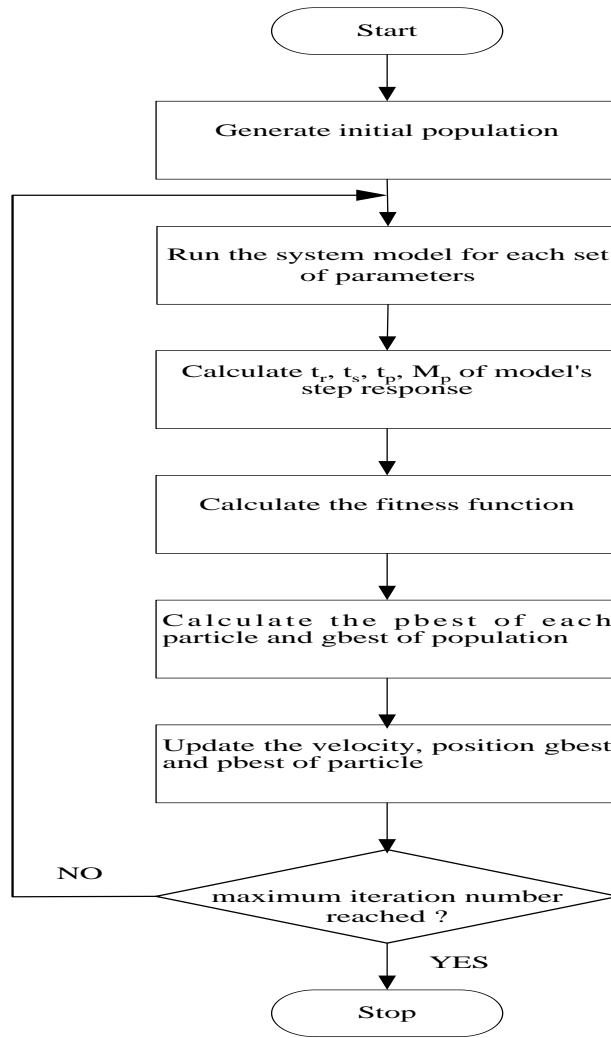


Figure 4.2: Flowchart of the PSO based PID control system

#### 4.4 Fitness Function

The Fitness function is important to be properly defined .In this work for minimization of rise time, settling time, and overshoot. The value of fitness function defined by optimization algorithm would be the minimum. The fitness function is defined by

$$f = \sqrt{(t_r)^2 + (t_s)^2 + (m_p)^2}$$

Where,  $t_r$ = rise time,  $t_s$ = settling time,  $m_p$ =peak overshoot

# CHAPTER 5

## Simulation Results and Discussions

### 5.1 First order system without delay

The transfer function of the plant is taken as:  $G_p(s) = \frac{2}{5s+1}$

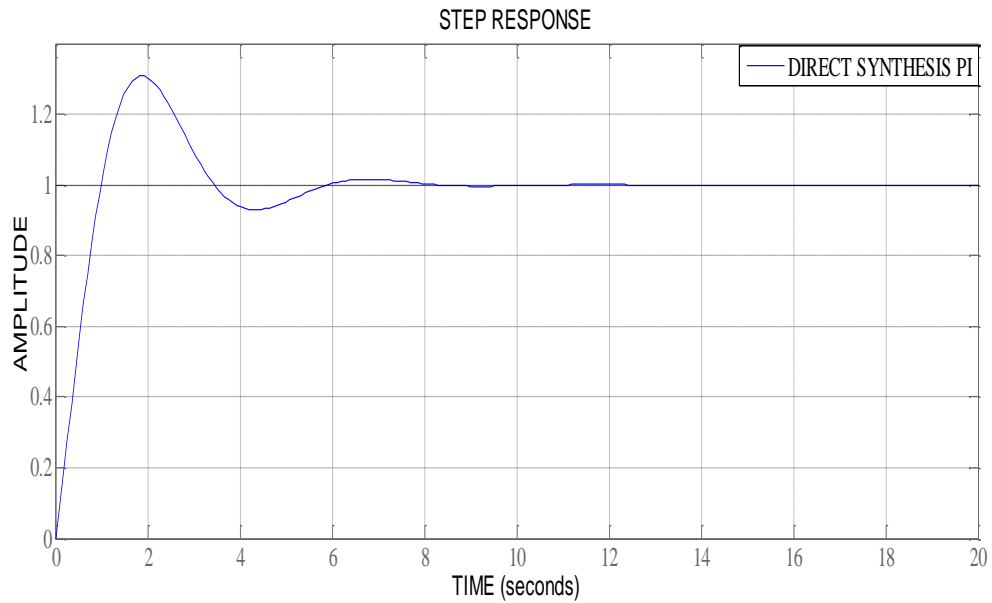
**Table 5.1:** PID tuning parameters for a first order system without delay

| TUNING METHOD                   | Controller | K <sub>P</sub> | K <sub>I</sub> | K <sub>D</sub> |
|---------------------------------|------------|----------------|----------------|----------------|
| Direct Synthesis                | PI         | 2.5            | 5              | 0              |
|                                 | PID        | NOT APPLICABLE |                |                |
| Computational Optimization (CO) | PI         | NOT APPLICABLE |                |                |
|                                 | PID        | 1.2            | 0.36           | 0.3            |

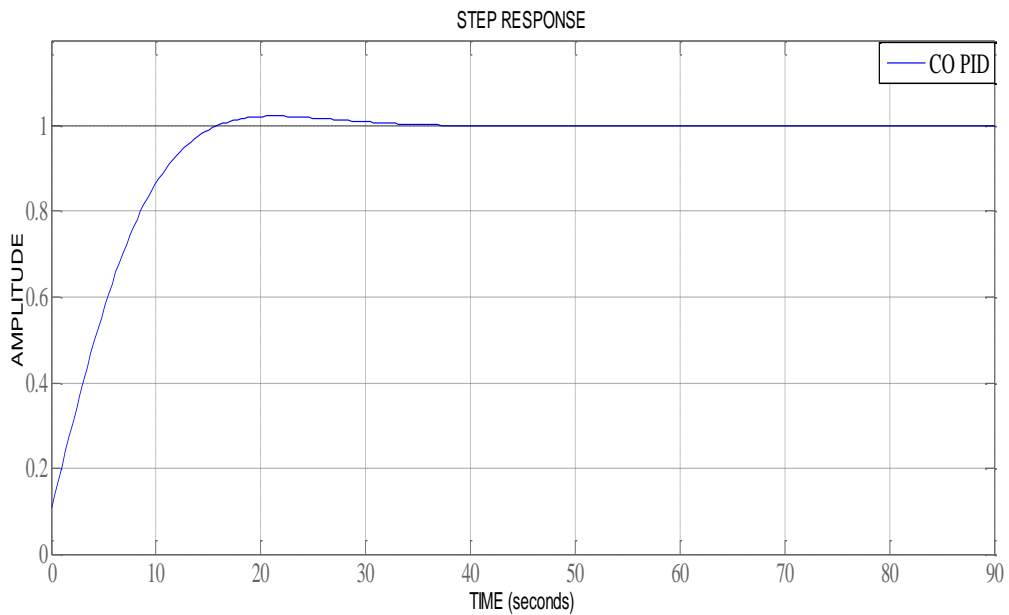
**Table 5.2:** Various performance specification for Direct Synthesis and CO method

| TUNING METHOD                   | Controller | Rise time( $t_r$ ) in sec | Settling time( $t_s$ ) in sec | Peak time( $t_p$ ) in sec | Overshoot (%) | Steady state error ( $e_{ss}$ ) in sec | IAE    | ISE    |
|---------------------------------|------------|---------------------------|-------------------------------|---------------------------|---------------|--|--------|--------|
| Direct Synthesis                | PI         | 0.7682                    | 5.4783                        | 1.8361                    | 30.7595       | 0                                      | 0.1    | 0.425  |
|                                 | PID        | NOT APPLICABLE            |                               |                           |               |  |        |        |
| Computational Optimization (CO) | PI         | NOT APPLICABLE            |                               |                           |               |  |        |        |
|                                 | PID        | 10.3417                   | 25.3851                       | 21.1786                   | 2.263         | 0                                      | 4.6296 | 2.6436 |

Simulation results show that tuning a PI controller with Direct Synthesis method is used when the small value of rise time ( $t_r$ ), settling time ( $t_s$ ) peak time ( $t_p$ ), integral absolute error (IAE) and integral squared error (ISE) is required. Computational Optimization (CO) is applicable only when less overshoot is required.



**Figure 5.1:** Unit step response for the plant using Direct Synthesis tuning method



**Figure 5.2:** Unit step response for the plant using Computational Optimization tuning method

Unit step response for the plant using Direct Synthesis method and Computational Optimization is shown in Figures 5.1 and 5.2 respectively. From above response Computational Optimization method is better than Direct Synthesis method for first order plant without delay.

## 5.2 First order system with delay

The transfer function of the plant is taken as:  $G_p(s) = \frac{2}{5s+1}e^{-3s}$

**Table 5.3:** PID tuning parameters for a first order system with delay

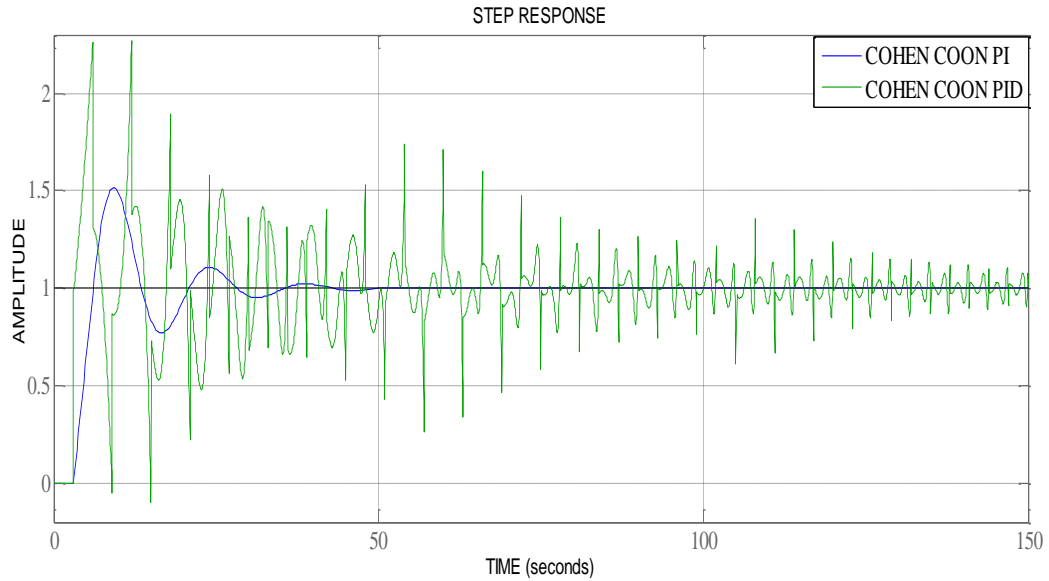
| TUNING METHOD              | Controller | K <sub>p</sub> | K <sub>i</sub> | K <sub>D</sub> |
|----------------------------|------------|----------------|----------------|----------------|
| Cohen Coon                 | PI         | 0.7916         | 0.1742         | 0              |
|                            | PID        | 1.2361         | 0.4120         | 2.4310         |
| Ziegler Nichols            | PI         | 0.7500         | 0.0757         | 0              |
|                            | PID        | 1              | 0.1666         | 1.5000         |
| Relay Auto tuning          | PI         | 0.5742         | 0.0663         | 0              |
|                            | PID        | 0.7656         | 0.1473         | 0.9945         |
| Computational Optimization | PI         | NOT APPLICABLE |                |                |
|                            | PID        | 1.2000         | 0.3600         | 0.3000         |

**Table 5.4:** Various performance criteria for first order plant with delay

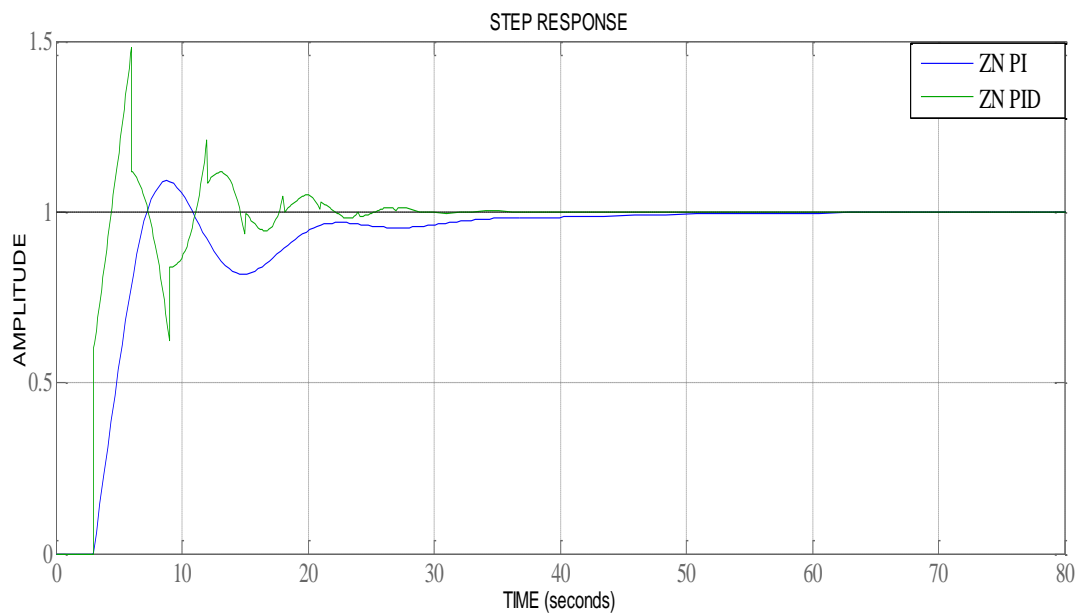
| TUNING METHOD              | Controller | Rise time(t <sub>r</sub> ) in sec | Settling time(t <sub>s</sub> ) in sec | Peak time(t <sub>p</sub> ) in sec | Overshoot (%) | Steady state error (e <sub>ss</sub> ) in sec |
|----------------------------|------------|-----------------------------------|---------------------------------------|-----------------------------------|---------------|--|
| Cohen Coon                 | PI         | 2.4624                            | 39.6939                               | 9.2353                            | 51.50740      | 0  |
|                            | PID        | 0.0008                            | 362.9955                              | 11.9995                           | 126.8031      | 0  |
| Ziegler Nichols            | PI         | 3.2034                            | 34.5245                               | 8.7541                            | 9.07410       | 0  |
|                            | PID        | 1.0518                            | 21.3943                               | 5.9999                            | 47.9566       | 0  |
| Relay Auto tuning          | PI         | 4.9185                            | 36.1287                               | 74.2355                           | 0             | 0  |
|                            | PID        | 2.1000                            | 18.2595                               | 6.0000                            | 12.8404       | 0  |
| Computational Optimization | PI         | NOT APPLICABLE                    |                                       |                                   |               |  |
|                            | PID        | 6.2166                            | 32.8463                               | 15.9186                           | 15.5132       | 0  |

Simulation results show that PI controller is better than PID controller for first order plant with delay. Tuning a PI controller with Cohen-Coon method is applicable when less rise

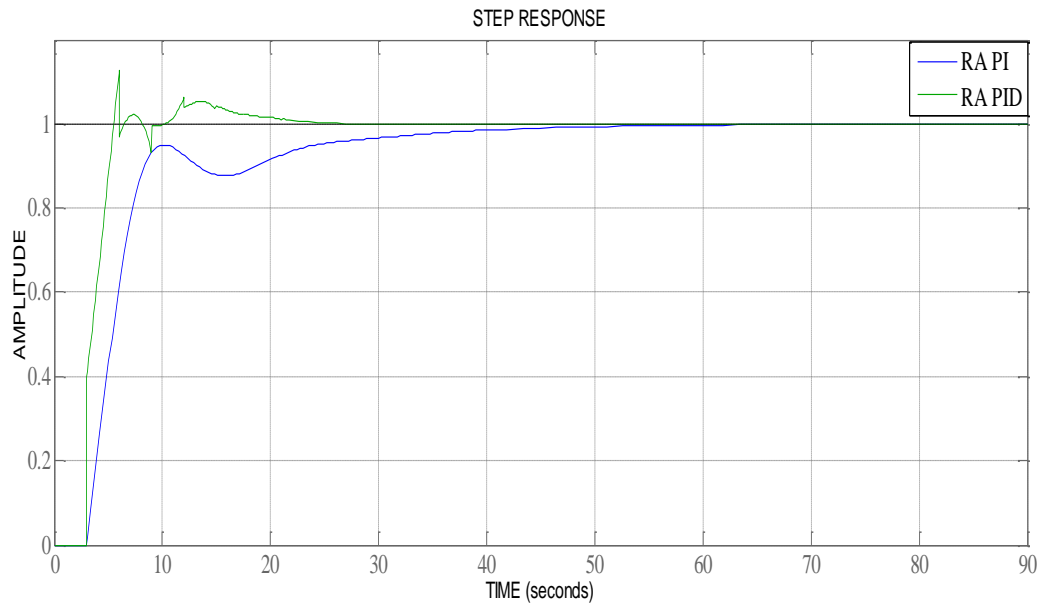
time ( $t_r$ ) is required. When less settling time ( $t_s$ ), peak time ( $t_p$ ) required ZN method is used. When we no overshoot is required RA method is used.



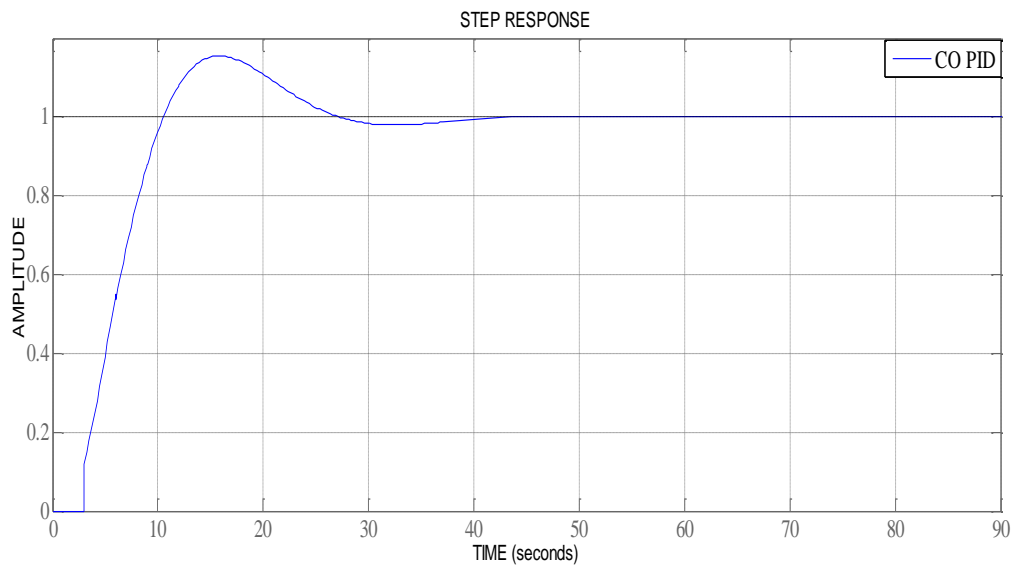
**Figure 5.3:** Unit step response for the plant using Cohen Coon tuning method



**Figure 5.4:** Unit step response for the plant using Ziegler-Nichols tuning method



**Figure 5.5:** Unit step response for the plant using Relay Auto tuning method



**Figure 5.6:** Unit step response for the plant using Computational Optimization tuning method

Figures 5.3 to 5.6 shows tuning of PID controller by Cohen-Coon, Ziegler-Nichols, Relay Auto tuning and Computational Optimization methods respectively for the first order plant with delay. From above Figures it can be concluded that PI controller is better than PID controller for first order plant with delay.

### 5.3 Second order system without delay

The transfer function of the plant is taken as:

$$G_p(s) = \frac{-0.5189S + 1.462}{0.1858S^2 + 0.8627S + 1}$$

**Table 5.5:** PID tuning parameters for second order plant without delay

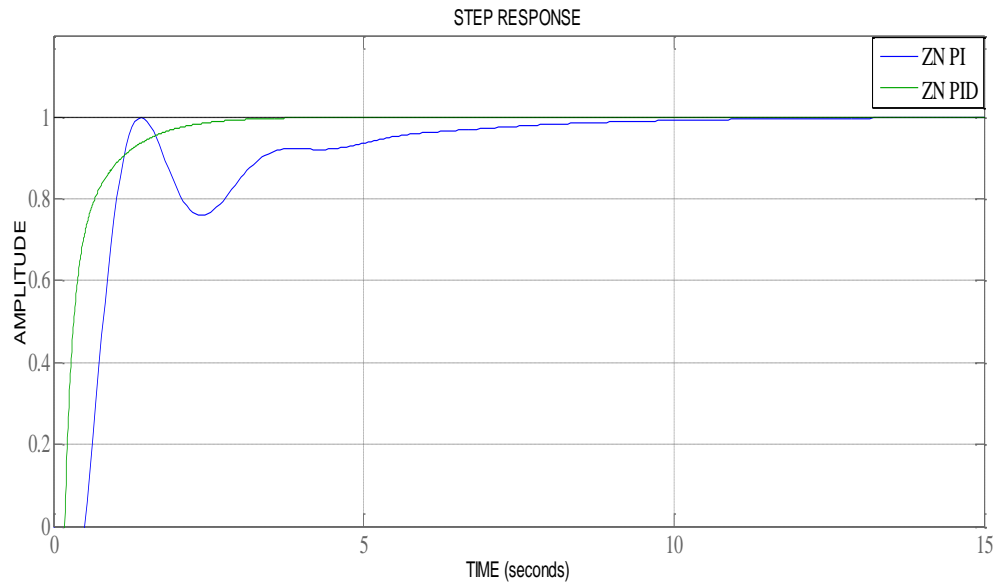
| TUNING METHOD              | Controller | K <sub>p</sub> | K <sub>i</sub> | K <sub>D</sub> |
|----------------------------|------------|----------------|----------------|----------------|
| Ziegler Nichols            | PI         | 0.7483         | 0.4642         | 0              |
|                            | PID        | 0.9977         | 1.0316         | 0.2412         |
| Relay Auto tuning          | PI         | 0.6727         | 0.4505         | 0              |
|                            | PID        | 0.8970         | 1.0010         | 0.2009         |
| Computational Optimization | PI         | NOT APPLICABLE |                |                |
|                            | PID        | 1.2000         | 0.3600         | 0.3000         |

**Table 5.6:** The performance criteria for second order plant without delay

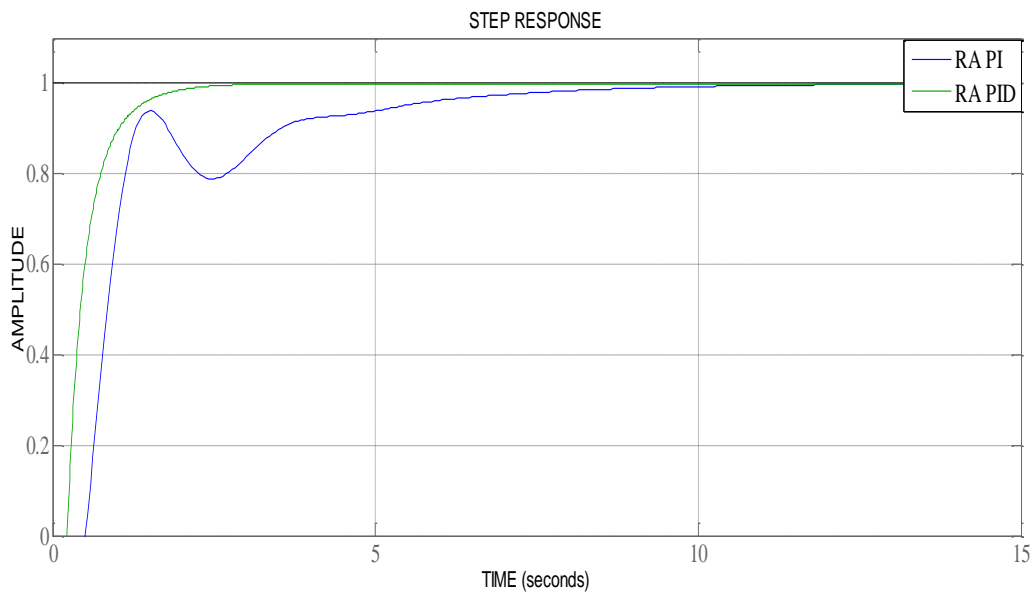
| TUNING METHOD              | Controller | Rise time(t <sub>r</sub> ) in sec | Settling time(t <sub>s</sub> ) in sec | Peak time(t <sub>p</sub> ) in sec | Overshoot (%) | Steady state error (e <sub>ss</sub> ) in sec | IAE     | ISE     |
|----------------------------|------------|-----------------------------------|---------------------------------------|-----------------------------------|---------------|--|---------|---------|
| Ziegler Nichols            | PI         | 0.5660                            | 7.2295                                | 1.4136                            | 0             | 0  | 1.4735  | 0.94018 |
|                            | PID        | 0.4583                            | 1.4323                                | 0                                 | 0             | 0  | 0.66304 | 0.77196 |
| Relay Auto tuning          | PI         | 0.7174                            | 7.0977                                | 15.6217                           | 0             | 0  | 1.5183  | 0.95163 |
|                            | PID        | 0.6676                            | 1.3873                                | 0                                 | 0             | 0  | 0.68331 | 0.71653 |
| Computational Optimization | PI         | NOT APPLICABLE                    |                                       |                                   |               |  |         |         |
|                            | PID        | 0.1384                            | 15.2586                               | 0                                 | 0             | 0  | 6.3333  | 2.7896  |

Simulation results show that PID controller is better than PI controller for second order plant without delay. Tuning a PID controller with RA method is applicable when less

settling time ( $t_s$ ) and integral squared error (ISE) is required. When less rise time ( $t_r$ ) is required CO method is used. ZN method is used when less integral absolute error (IAE) is required.



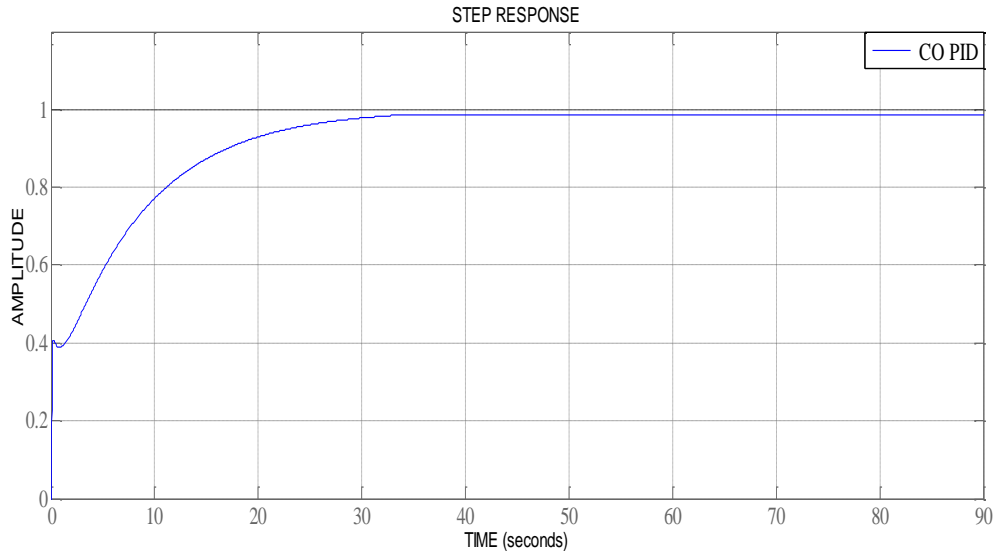
**Figure 5.7:** Unit step response for the plant using Ziegler-Nichols tuning method



**Figure 5.8:** Unit step response for the plant using Relay Auto tuning method

Unit step response obtained by tuning of PID controller by Ziegler-Nichols, Relay Auto tuning and Computational Optimization is shown in Figures 5.7 to 5.9. The above

response shows that PI and PID controller is applicable for better response for second order plant without delay.



**Figure 5.9:** Unit step response for the plant using Computational Optimization tuning method

## 5.4 Second order system with delay

The transfer function of the plant is taken as:

$$G_P(s) = \frac{-0.5189S + 1.462}{0.1858S^2 + 0.8627S + 1} e^{-3S}$$

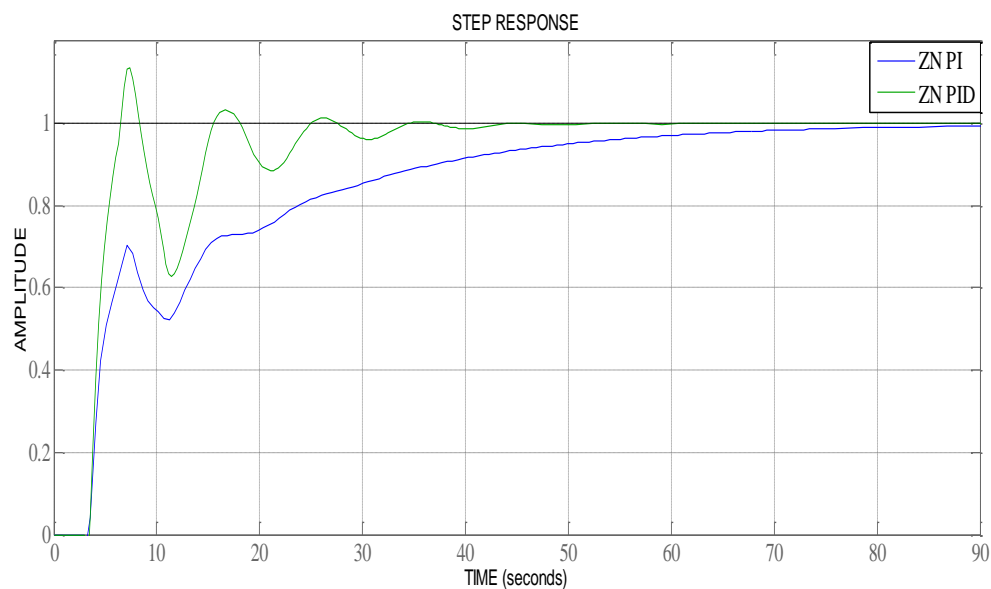
**Table 5.7:** PID tuning parameters for second order plant with delay

| TUNING METHOD              | Controller | $K_p$          | $K_I$  | $K_D$  |
|----------------------------|------------|----------------|--------|--------|
| Ziegler Nichols            | PI         | 0.3309         | 0.0471 | 0      |
|                            | PID        | 0.4412         | 0.1046 | 0.4655 |
| Relay Auto tuning          | PI         | 0.3936         | 0.0586 | 0      |
|                            | PID        | 0.2624         | 0.0652 | 0.2642 |
| Computational Optimization | PI         | NOT APPLICABLE |        |        |
|                            | PID        | 1.2000         | 0.3600 | 0.3000 |

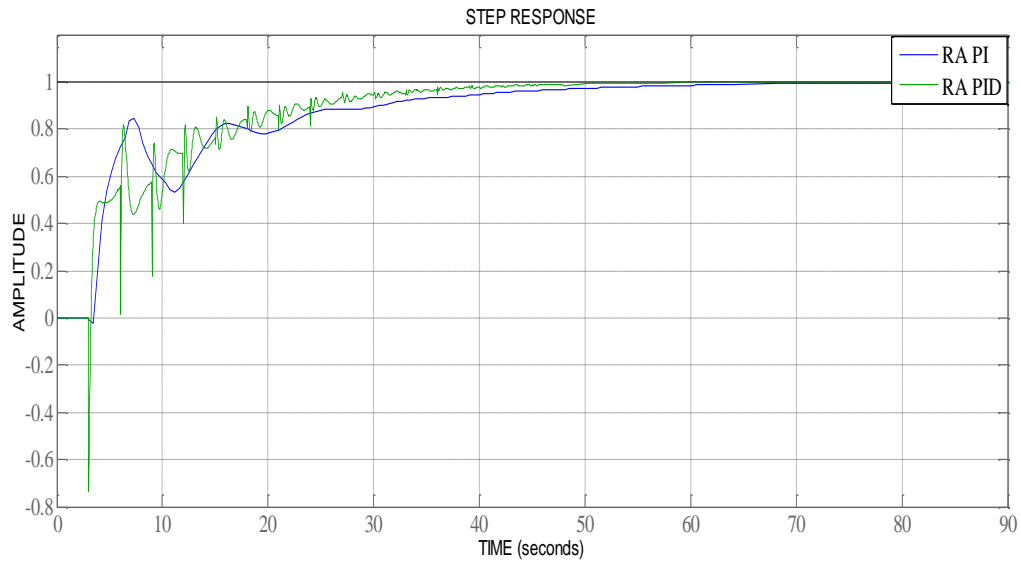
**Table 5.8:** Performance criteria for second order plant with delay

| TUNING METHOD              | Controller | Rise time( $t_r$ ) in sec | Settling time( $t_s$ ) in sec | Peak time( $t_p$ ) in sec | Overshoot (%) | Steady state error ( $e_{ss}$ ) in sec |
|----------------------------|------------|---------------------------|-------------------------------|---------------------------|---------------|--|
| Ziegler Nichols            | PI         | 33.2036                   | 66.0667                       | 127.5329                  | 0             | 0                                      |
|                            | PID        | 2.2628                    | 32.5634                       | 7.3758                    | 13.5442       | 0                                      |
| Relay Auto tuning          | PI         | 27.0073                   | 54.5930                       | 210.6971                  | 0             | 0                                      |
|                            | PID        | 19.1985                   | 36.1991                       | 113.3194                  | 0             | 0                                      |
| Computational Optimization | PI         | NOT APPLICABLE            |                               |                           |               |  |
|                            | PID        | 2.8614                    | 39.0624                       | 6.3484                    | 16.5723       | 0                                      |

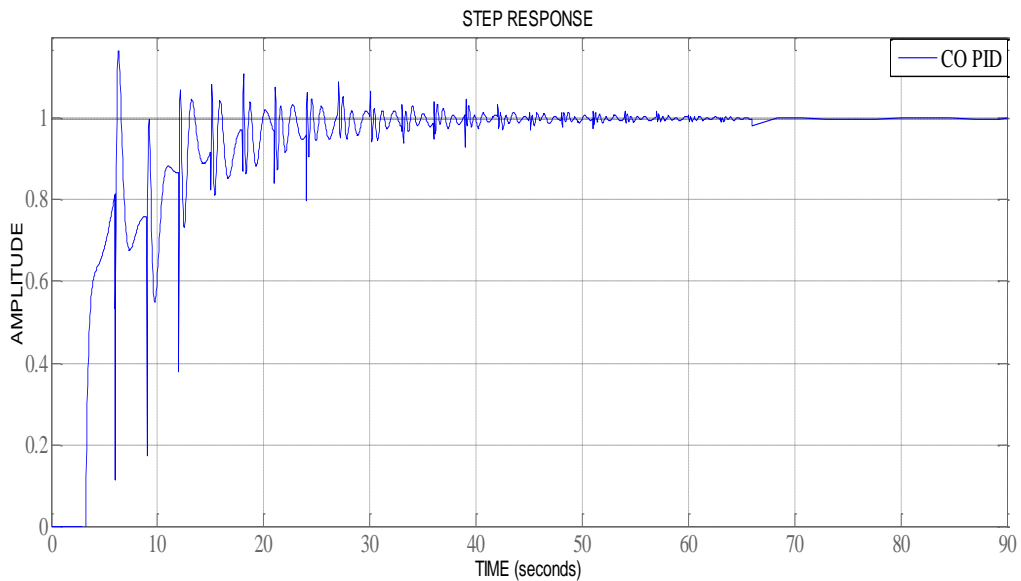
Simulation results show that PID controller is better than PI controller for second order plant with delay. Tuning a PID controller with ZN method is applicable when less rise time ( $t_r$ ) and settling time ( $t_s$ ) is required. For less peak time ( $t_p$ ) CO method is used.



**Figure 5.10:** Unit step response for the plant using Ziegler-Nichols tuning method



**Figure 5.11:** Unit step response for the plant using Relay Auto tuning method



**Figure 5.12:** Unit step response for the plant using Computational Optimization tuning method

Figures 5.10 to 5.12 shows the unit step response for second order plant with delay. From above Figure it is clear that when we use PID controller for tuning of second order plant with delay there is more distortion in the response so we prefer PI controller for this type of plant.

## 5.5 Third order system without delay

The transfer function of the plant is taken as:

$$G_p(s) = \frac{6}{48s^3 + 44s^2 + 12s + 1}$$

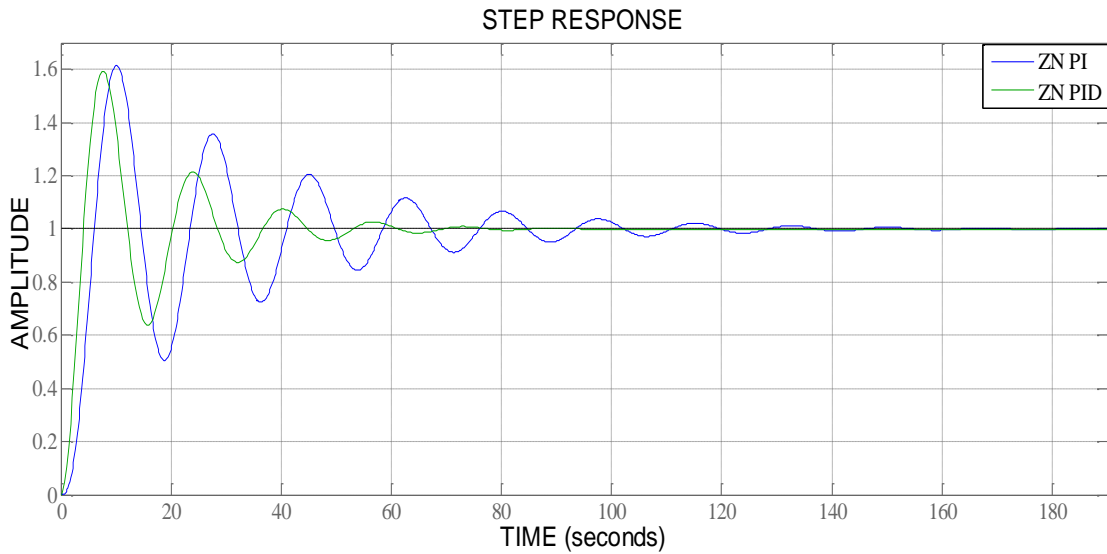
**Table 5.9:** PID tuning parameters for third order plant without delay

| TUNING METHOD              | Controller | K <sub>P</sub> | K <sub>I</sub> | K <sub>D</sub> |
|----------------------------|------------|----------------|----------------|----------------|
| Ziegler Nichols            | PI         | 0.7515         | 0.0675         | 0              |
|                            | PID        | 1.0020         | 0.2500         | 1.6733         |
| Relay Auto tuning          | PI         | 0.7354         | 0.0692         | 0              |
|                            | PID        | 0.9805         | 0.1539         | 1.5620         |
| Modulus Optimum            | PI         | NOT APPLICABLE |                |                |
|                            | PID        | 0.2500         | 0.0250         | 0.6000         |
| Computational Optimization | PI         | NOT APPLICABLE |                |                |
|                            | PID        | 0.4600         | 0.0529         | 0.8000         |

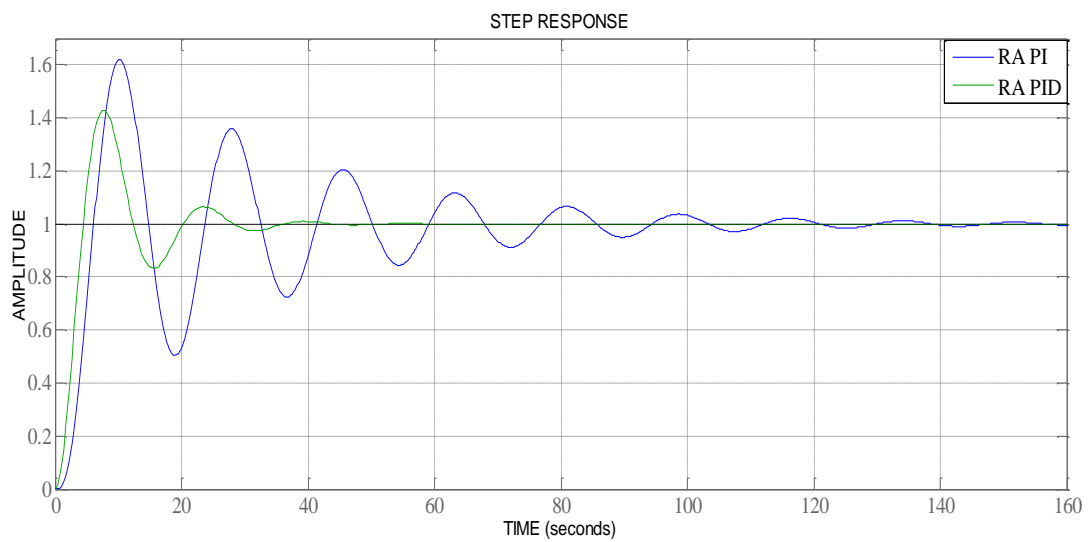
**Table 5.10:** Performance criteria for a third order delay plant without delay

| TUNING METHOD              | Controller | Rise time(t <sub>r</sub> ) in sec | Settling time(t <sub>s</sub> ) in sec | Peak time(t <sub>p</sub> ) in sec | Overshoot (%) | Steady state error (e <sub>ss</sub> ) in sec | IAE    | ISE    |
|----------------------------|------------|-----------------------------------|---------------------------------------|-----------------------------------|---------------|--|--------|--------|
| Ziegler Nichols            | PI         | 3.5685                            | 116.105                               | 10.0737                           | 0             | 0  | 2.4691 | 7.3001 |
|                            | PID        | 2.7766                            | 58.4976                               | 7.8134                            | 59.2726       | 0  | 0.6666 | 4.113  |
| Relay Auto tuning          | PI         | 3.5941                            | 117.284                               | 10.2201                           | 61.9451       | 0  | 2.4069 | 7.4101 |
|                            | PID        | 2.9935                            | 33.1324                               | 7.5034                            | 42.8670       | 0  | 1.0831 | 2.827  |
| Modulus Optimum            | PI         | NOT APPLICABLE                    |                                       |                                   |               |  |        |        |
|                            | PID        | 10.7301                           | 17.6936                               | 28.2871                           | 0.0839        | 0  | 6.6667 | 4.3345 |
| Computational Optimization | PI         | NOT APPLICABLE                    |                                       |                                   |               |  |        |        |
|                            | PID        | 6.4775                            | 22.1642                               | 13.9401                           | 8.6943        | 0  | 3.9382 | 3.3679 |

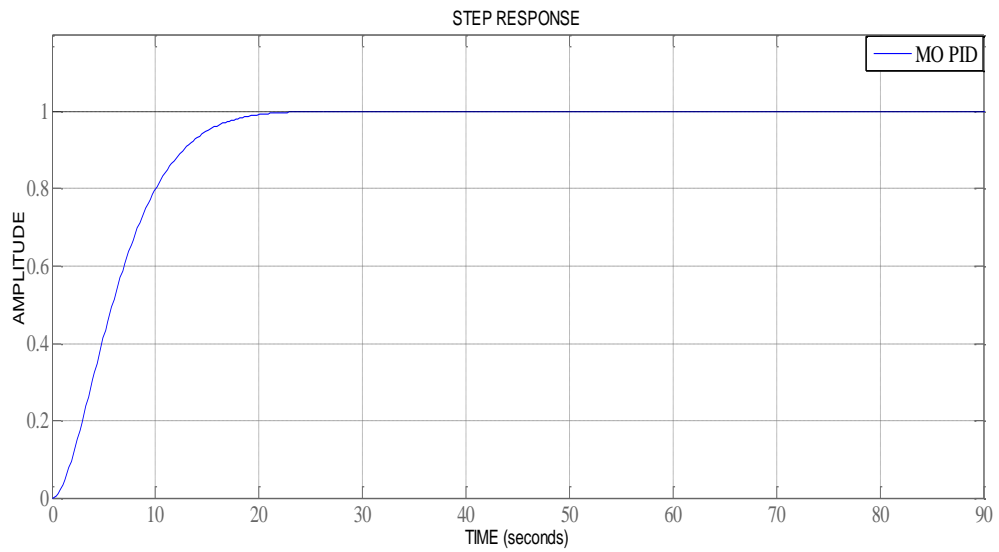
Simulation results show that tuning a PID controller is better than PI controller for third order plant without delay. PID controller with Ziegler Nichols (ZN) tuning method results in less rise time ( $t_r$ ), peak time ( $t_p$ ), and integral squared error (ISE). The Relay Auto tuning method is applicable when less ISE is required while Modulus Optimum (MO) tuning method is applicable when less settling time ( $t_s$ ) and less overshoot is required and Computational Optimization (CO) method is helpful when the desired closed loop specifications are decided by the designer.



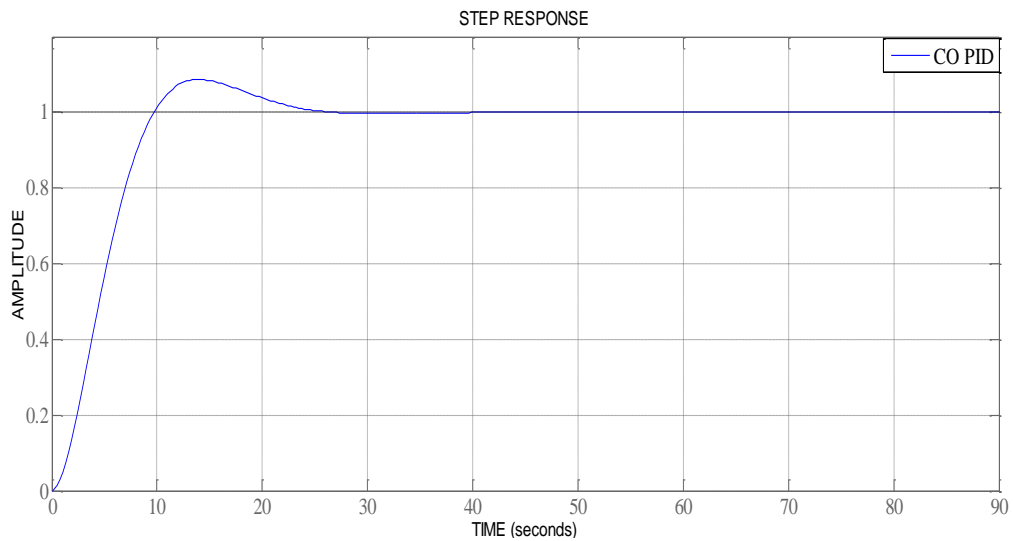
**Figure 5.13:** Unit step response for the plant using Ziegler-Nichols tuning method



**Figure 5.14:** Unit step response for the plant using Relay Auto tuning method



**Figure 5.15:** Unit step response for the plant using Modulus Optimum tuning method



**Figure 5.16:** Unit step response for the plant using Computational Optimization tuning method

Unit step response for the third order plant using Ziegler-Nichols, Relay Auto tuning, Modulus Optimum and Computational Optimization is shown in Figures from 5.13 to 5.16 respectively. The above response shows that it is better to use PID controller than PI controller for third order plant without delay.

## 5.6 Third order system with delay

The transfer function of the plant is taken as:

$$G_P(s) = \frac{6}{48s^3 + 44s^2 + 12s + 1} e^{-3s}$$

**Table 5.11:** Tuning parameters for third order plant with delay

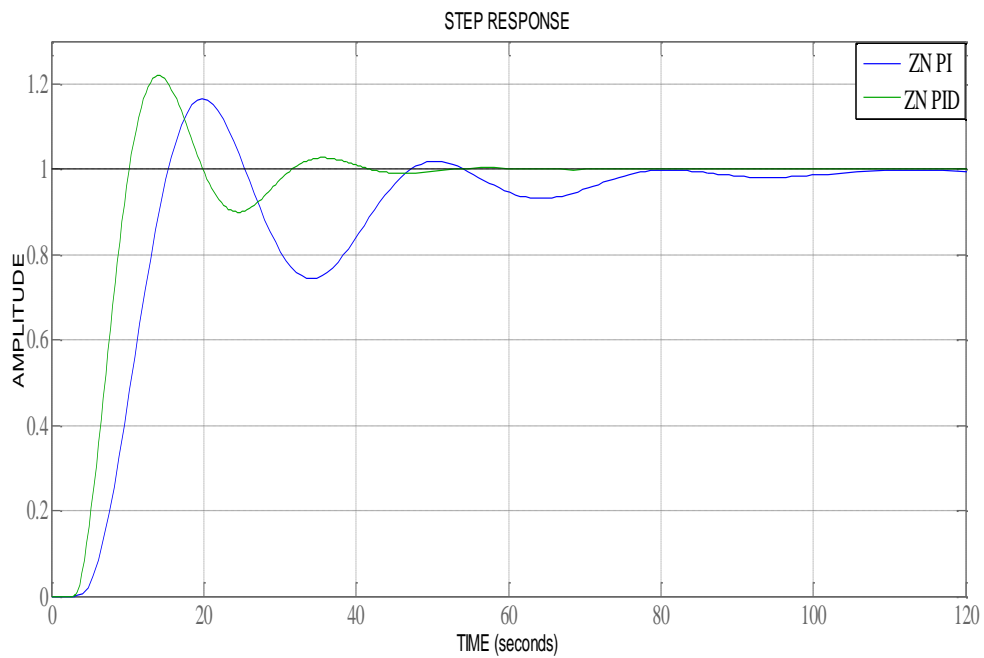
| TUNING METHOD              | Controller | K <sub>p</sub> | K <sub>i</sub> | K <sub>D</sub> |
|----------------------------|------------|----------------|----------------|----------------|
| Ziegler Nichols            | PI         | 0.2421         | 0.0120         | 0              |
|                            | PID        | 0.3228         | 0.0266         | 0.9785         |
| Relay Auto tuning          | PI         | 0.2342         | 0.0120         | 0              |
|                            | PID        | 0.3123         | 0.0267         | 0.9124         |
| Computational Optimization | PI         | NOT APPLICABLE |                |                |
|                            | PID        | 0.3800         | 0.0361         | 0.7000         |

**Table 5.12:** Performance criteria for third order plant with delay

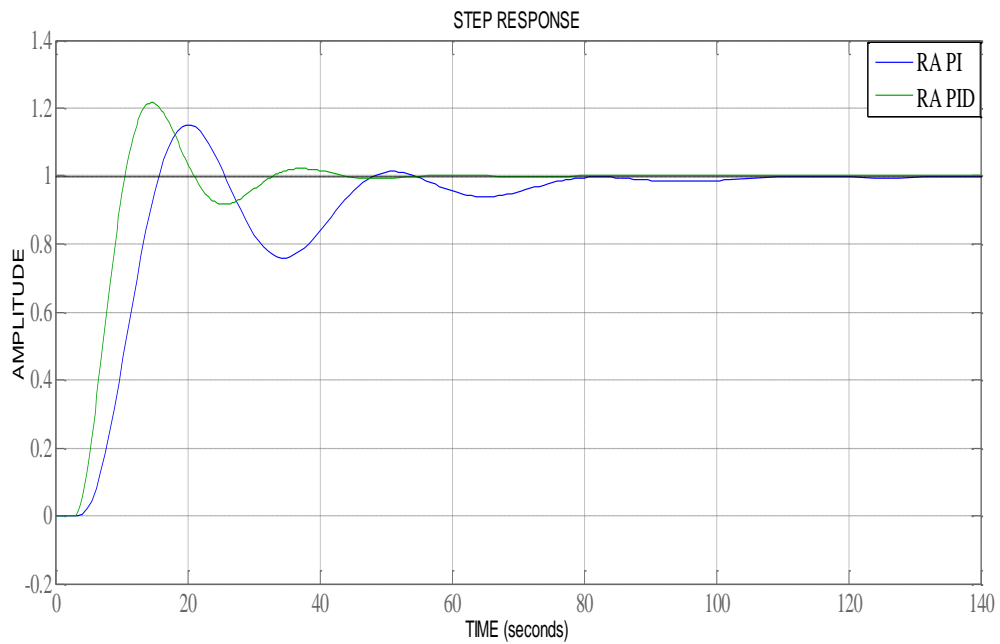
| TUNING METHOD              | Controller | Rise time(t <sub>r</sub> ) in sec | Settling time(t <sub>s</sub> ) in sec | Peak time(t <sub>p</sub> ) in sec | Overshoot (%) | Steady state error (e <sub>SS</sub> ) in sec |
|----------------------------|------------|-----------------------------------|---------------------------------------|-----------------------------------|---------------|--|
| Ziegler Nichols            | PI         | 7.7365                            | 74.4445                               | 19.8317                           | 16.4089       | 0  |
|                            | PID        | 4.9965                            | 38.0829                               | 13.9349                           | 21.8323       | 0  |
| Relay Auto tuning          | PI         | 7.9322                            | 75.0745                               | 19.8331                           | 15.0891       | 0  |
|                            | PID        | 5.2071                            | 38.5963                               | 14.4838                           | 21.5836       | 0  |
| Computational Optimization | PI         | NOT APPLICABLE                    |                                       |                                   |               |  |
|                            | PID        | 6.2210                            | 35.1791                               | 16.8937                           | 18.5908       | 0  |

Simulation results show that tuning a PID controller is better than PI controller for third order plant with delay. PID controller with Ziegler Nichols (ZN) tuning method results in

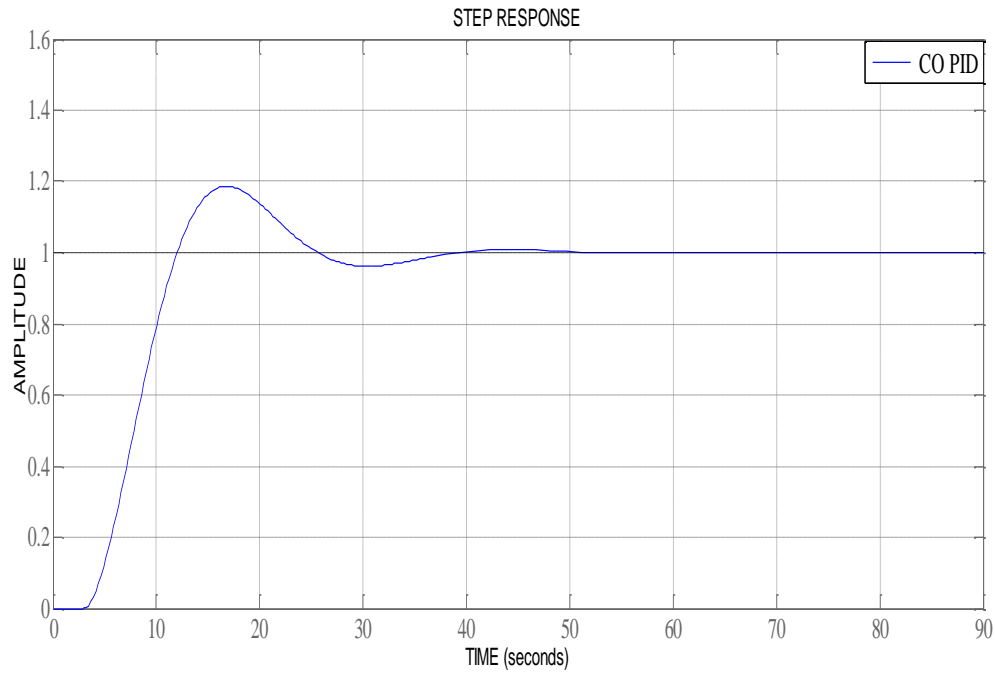
less rise time ( $t_r$ ) and peak time ( $t_p$ ). The Computational Optimization (CO) method is helpful when less settling time ( $t_s$ ) and overshoots are required.



**Figure 5.17:** Unit step response for the plant using Ziegler-Nichols tuning method



**Figure 5.18:** Unit step response for the plant using Relay Auto tuning method



**Figure 5.19:** Unit step response for the plant using Computational Optimization tuning method

Figures 5.17 to 5.19 shows the unit step response for the third order plant with delay using Ziegler-Nichols, Relay Auto tuning and Computational Optimization method respectively. The above response shows that PID controller is better than PI controller for third order plant with delay.

### 5.7 Third order plant by PSO

On simulation the following results have been obtained, first the system was tuned using Zeigler Nichols tuning method. The values of  $K_P$ ,  $K_I$  and  $K_D$  obtained were 1.002, 0.25, and 1.6733 respectively. Secondly the system has been tuned using PSO. The following PSO parameters were used to verify the performance of the PSO-PID controller parameters. In this paper, we define  $w=0.2$ ,  $c1=4$ ,  $c2=6$ .

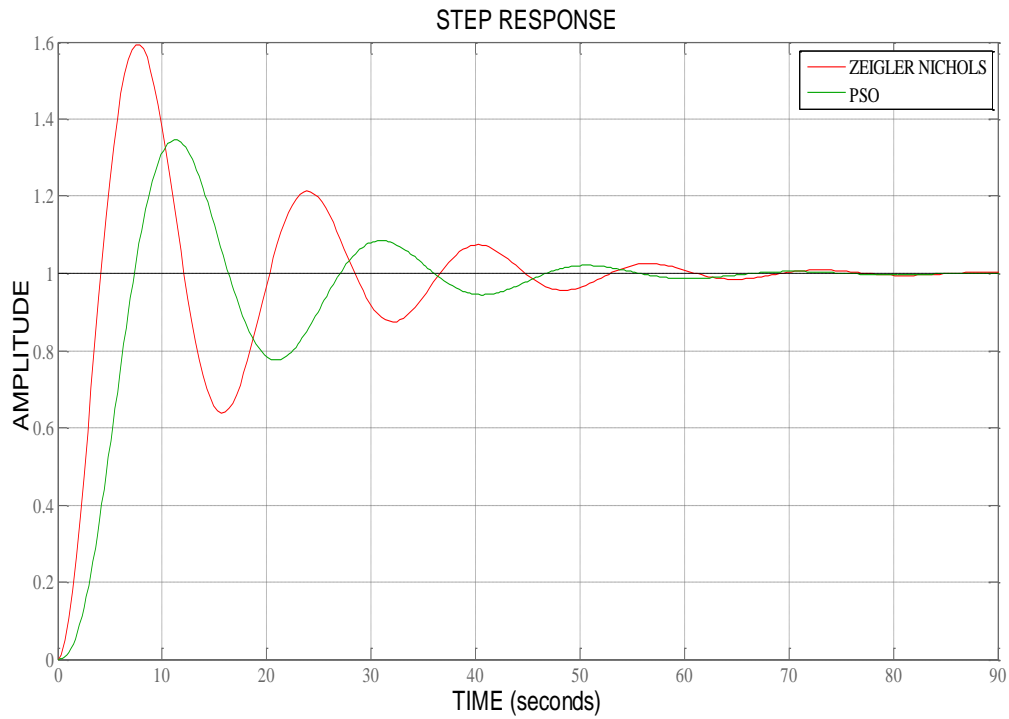
**Table 5.13:** The result of PID parameters for third order plant

| METHODS |                  |           |   | $K_p$  | $K_D$  | $K_I$  |
|---------|------------------|-----------|---|--------|--------|--------|
| ZN      |                  |           |   | 1.002  | 1.6733 | 0.25   |
| PSO     | Population<br>20 | Iteration | 2 | 0.5508 | 0.131  | 0.0423 |
|         |                  |           | 3 | 0.4516 | 0.0595 | 0.029  |
|         |                  |           | 4 | 0.422  | 0.3282 | 0.0393 |
|         | Population<br>25 | Iteration | 2 | 0.6618 | 0.9612 | 0.094  |
|         |                  |           | 3 | 0.4513 | 0.4759 | 0.0477 |
|         |                  |           | 4 | 0.3783 | 0.6519 | 0.0403 |

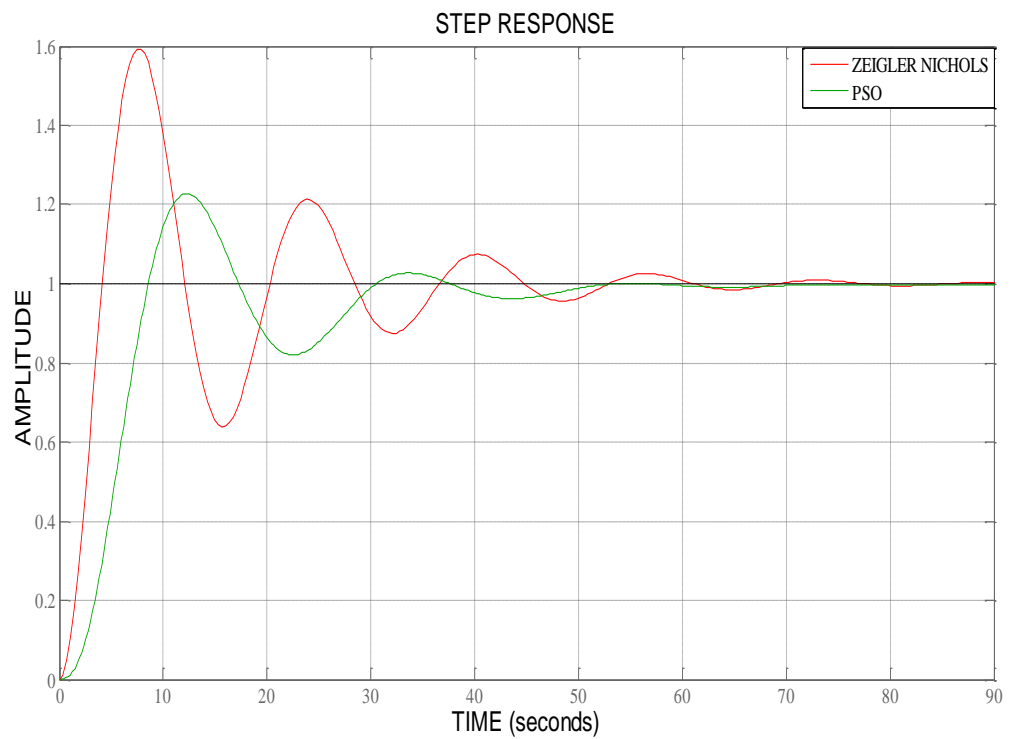
**Table 5.14:** The result of PID performance criteria

| Performance<br>Index            | ZN      | PSO           |         |         |               |         |         |
|---------------------------------|---------|---------------|---------|---------|---------------|---------|---------|
|                                 |         | Population=20 |         |         | Population=25 |         |         |
|                                 |         | iter=2        | iter=3  | iter=4  | iter=2        | iter=3  | iter=4  |
| Rise time ( $t_r$ ) in sec      | 2.7668  | 4.5411        | 5.2959  | 5.5760  | 3.9008        | 5.2757  | 6.2973  |
| Settling time ( $t_s$ ) in sec  | 58.3869 | 51.4911       | 48.3869 | 29.7107 | 32.664        | 28.6889 | 20.0487 |
| Peak time ( $t_p$ ) in sec      | 7.8374  | 11.1791       | 12.3390 | 12.6934 | 9.5900        | 11.9200 | 13.4794 |
| Overshoot (%)                   | 59.3524 | 34.5893       | 22.7791 | 20.8809 | 35.567        | 22.9596 | 10.8606 |
| Steady state error ( $e_{ss}$ ) | 0       | 0             | 0       | 0       | 0             | 0       | 0       |
| ISE                             | 4.1191  | 4.4886        | 4.5647  | 4.0303  | 3.2259        | 3.7497  | 3.5403  |
| IAE                             | 0.6667  | 3.9445        | 5.7401  | 4.2361  | 1.7721        | 3.4959  | 4.1354  |

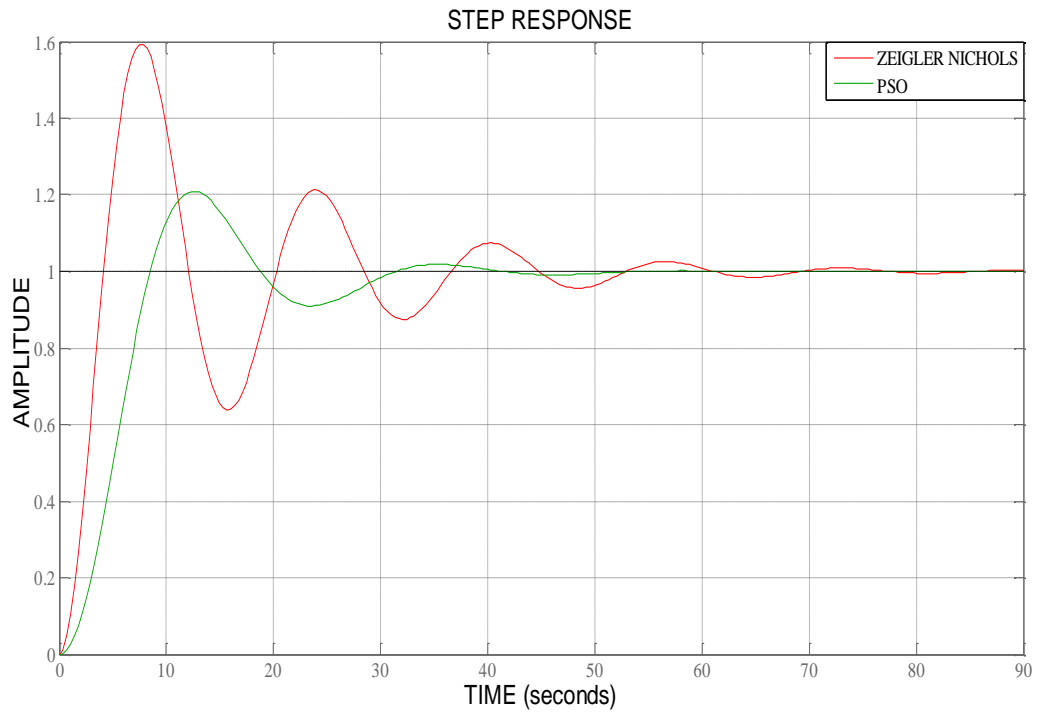
Step response with ZN and PSO when population size is 20 and 25 and iteration is 2, 3 and 4 is shown in Figure 5.20 to Figure 5.24. The settling time and overshoot of the step response by PSO method is less than the ZN method. The settling time and overshoot gradually decreasing when we increase the number of iterations. The step response is also improved when we increase the number of population size.



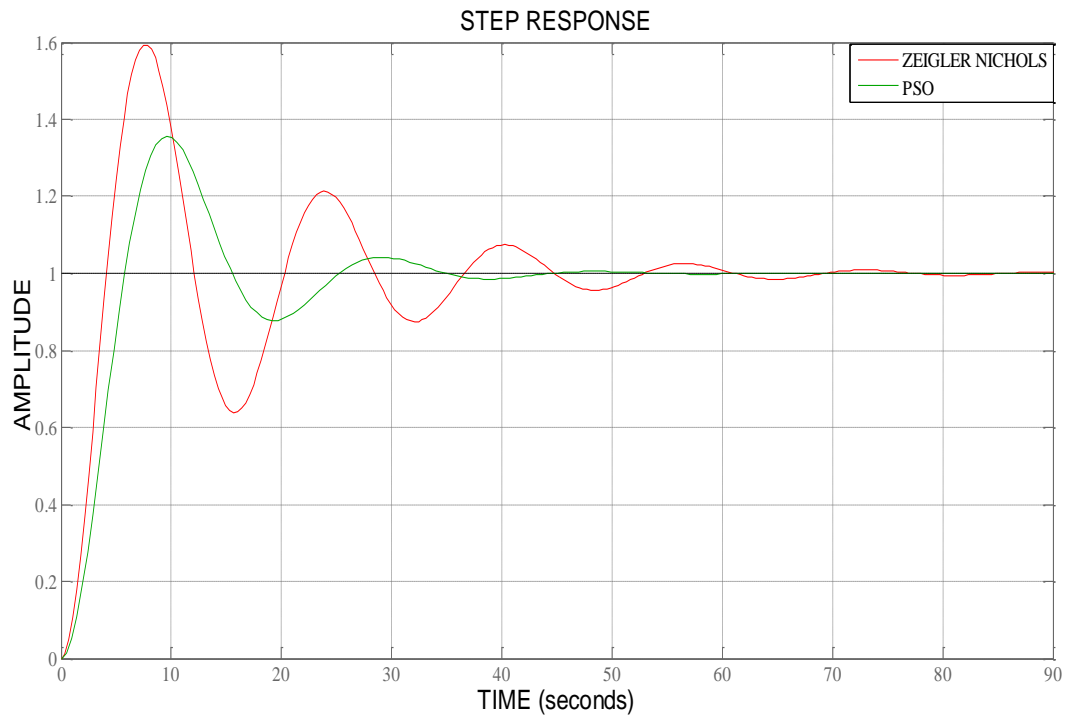
**Figure 5.20:** Step response with ZN and PSO when population size is 20 and iteration is 2



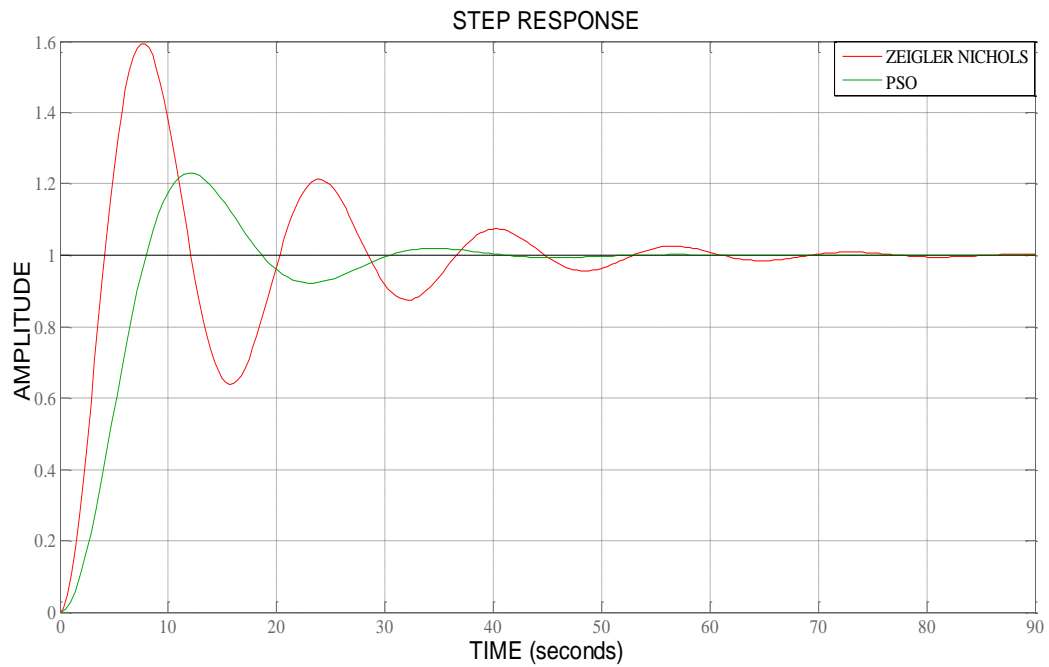
**Figure 5.21:** Step response with ZN and PSO when population size is 20 and iteration is 3



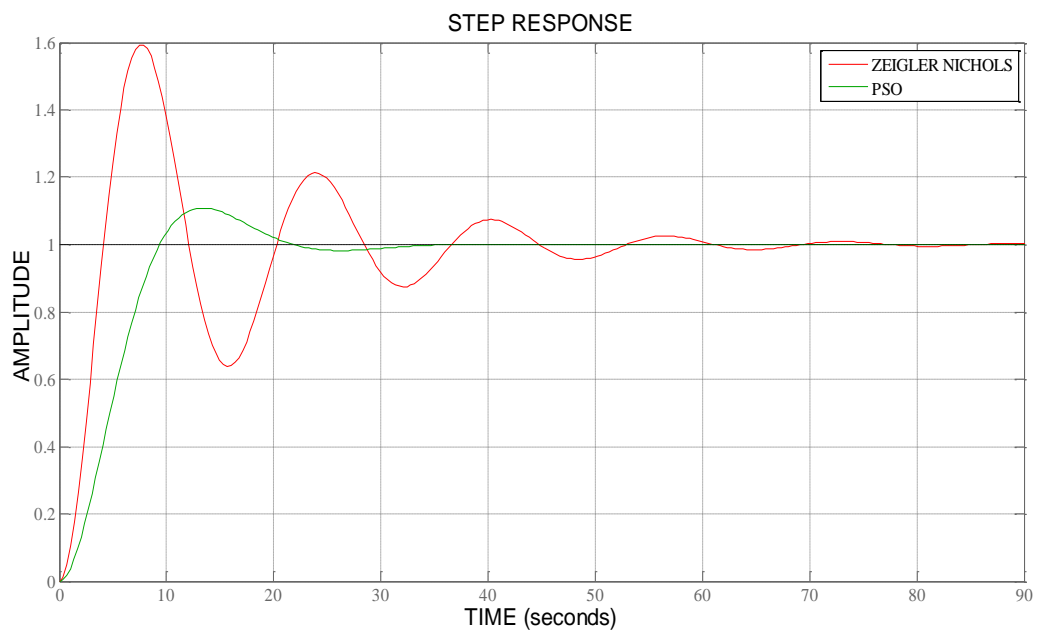
**Figure 5.22:** Step response with ZN and PSO when population size is 20 and iteration is 4



**Figure 5.23:** Step response with ZN and PSO when population size is 25 and iteration is 2



**Figure 5.24:** Step response with ZN and PSO when population size is 25 and iteration is 3



**Figure 5.25:** Step response with ZN and PSO when population size is 25 and iteration is 4

Unit step response with ZN and PSO with different population size and iteration is shown in Figures 5.20 to 5.25. The above response shows that PSO is better than ZN tuning method for third order plant.

## 5.8 Third order plant with different control schemes

We consider the following plant model for different control schemes:

$$G_{P1}(s) = \frac{6}{48s^3 + 44s^2 + 12s + 1}$$

$$G_{P2}(s) = \frac{1}{0.5s + 1}$$

$$G_d(s) = \frac{1}{5s + 1}$$

$$G_{mf} = G_{m1} = G_{m2} = 1$$

In this dissertation the plant  $G_{P2}(s)$  has been tuned by Direct Synthesis method and the plant  $G_{P1}(s)$  has been tuned by Zeigler Nichols and Relay Auto tuning method.

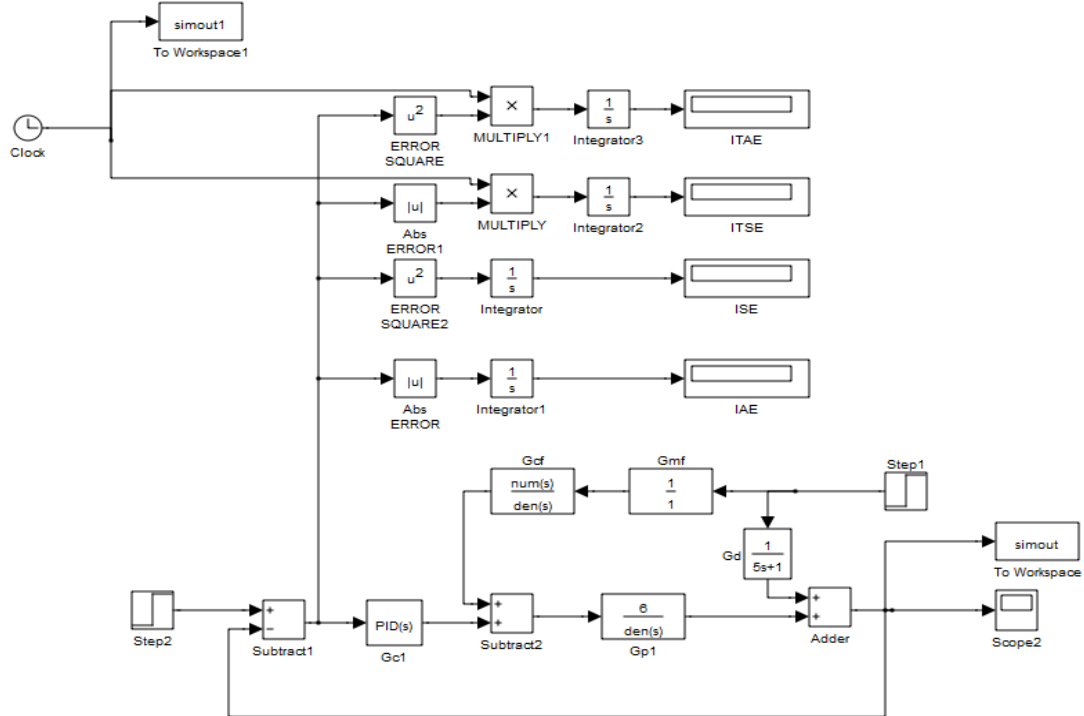


Figure 5.26: MATLAB Simulink diagram for feed forward plus feedback control by ZN

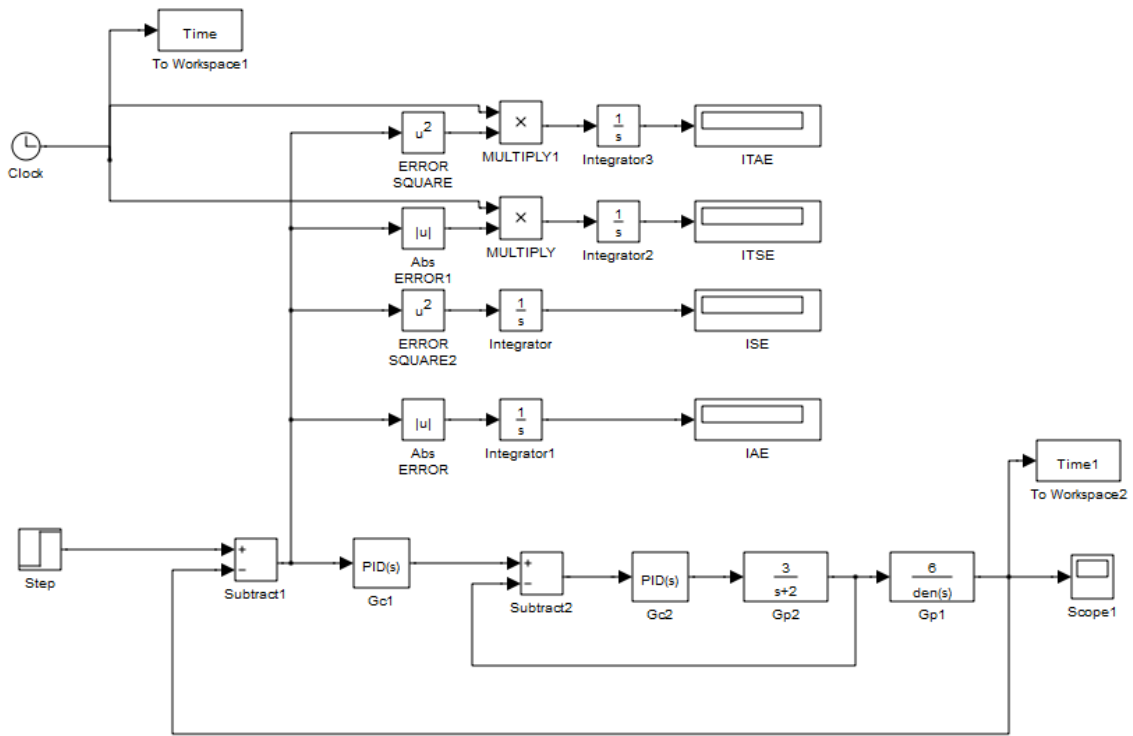


Figure 5.27: MATLAB Simulink diagram for cascade control by ZN method

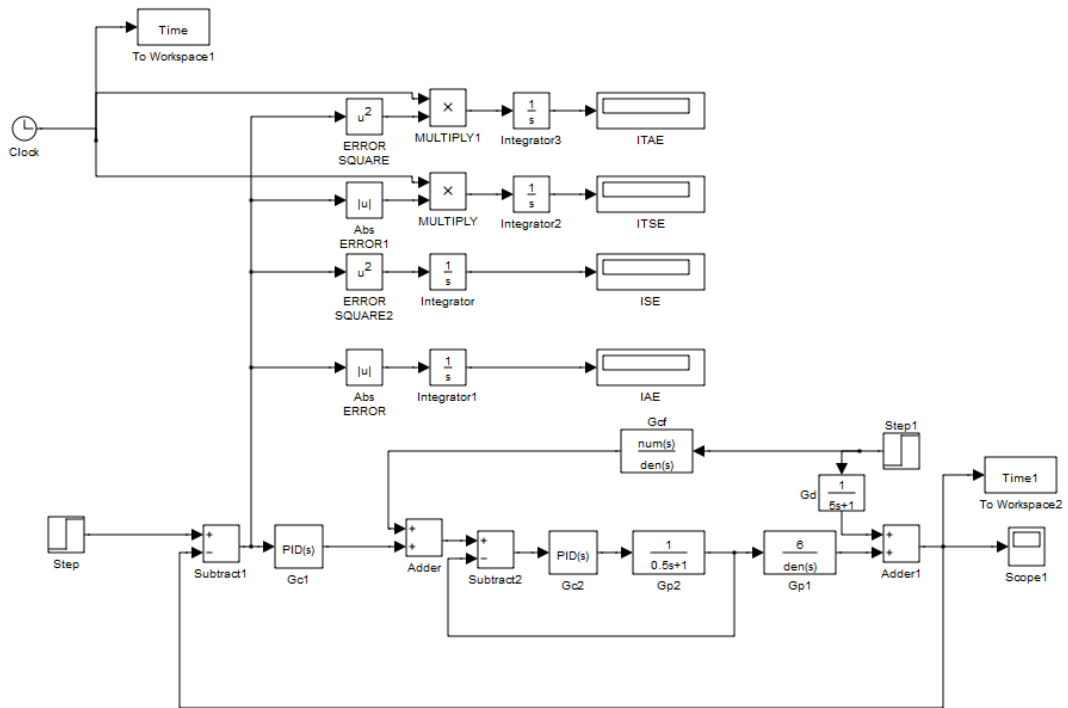


Figure 5.28: MATLAB Simulink diagram for cascade plus feed forward control by ZN method

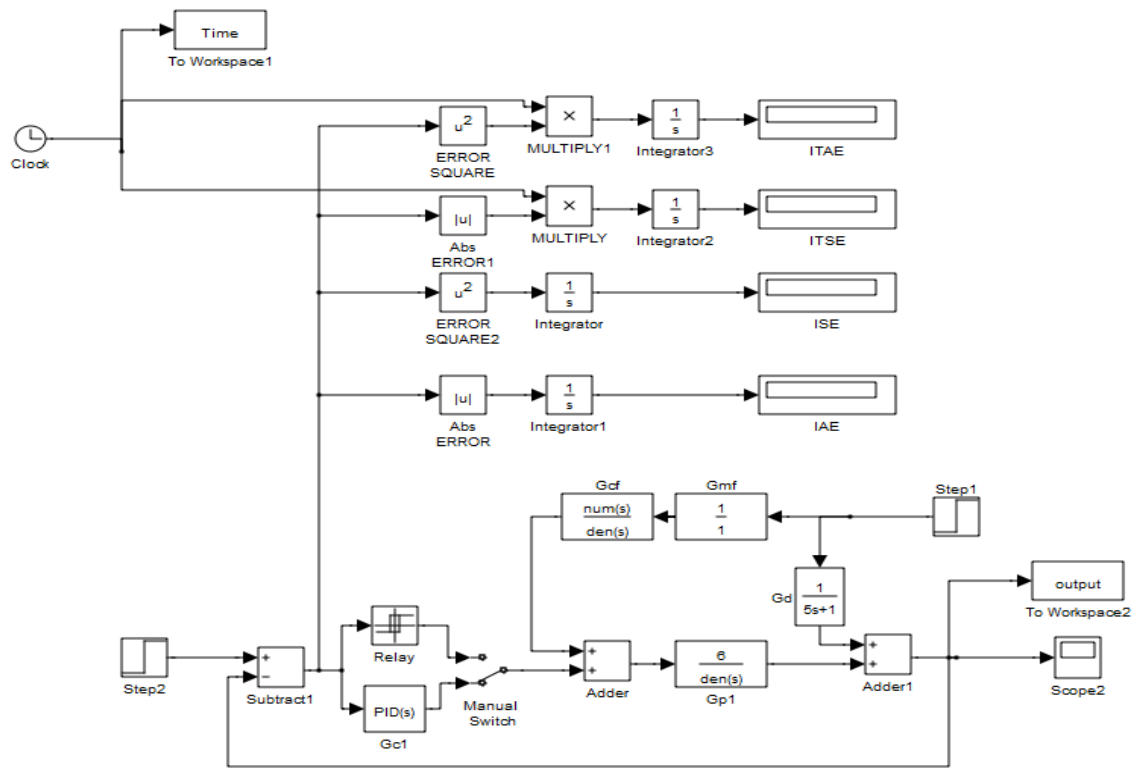


Figure 5.29: MATLAB Simulink diagram for feed forward plus feedback control by RA method

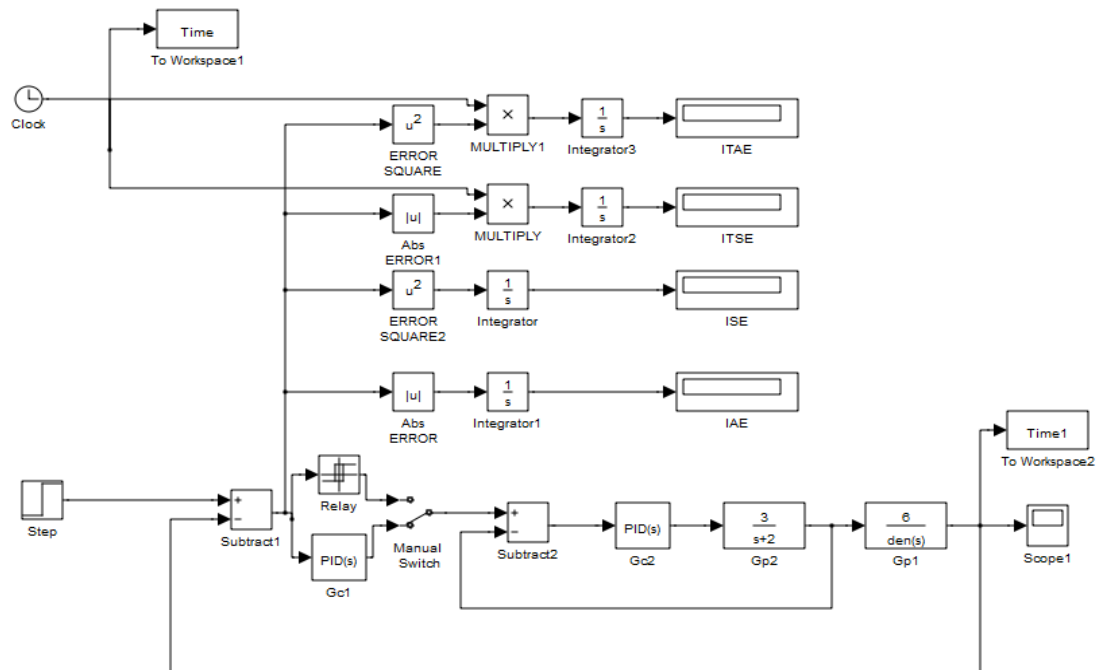
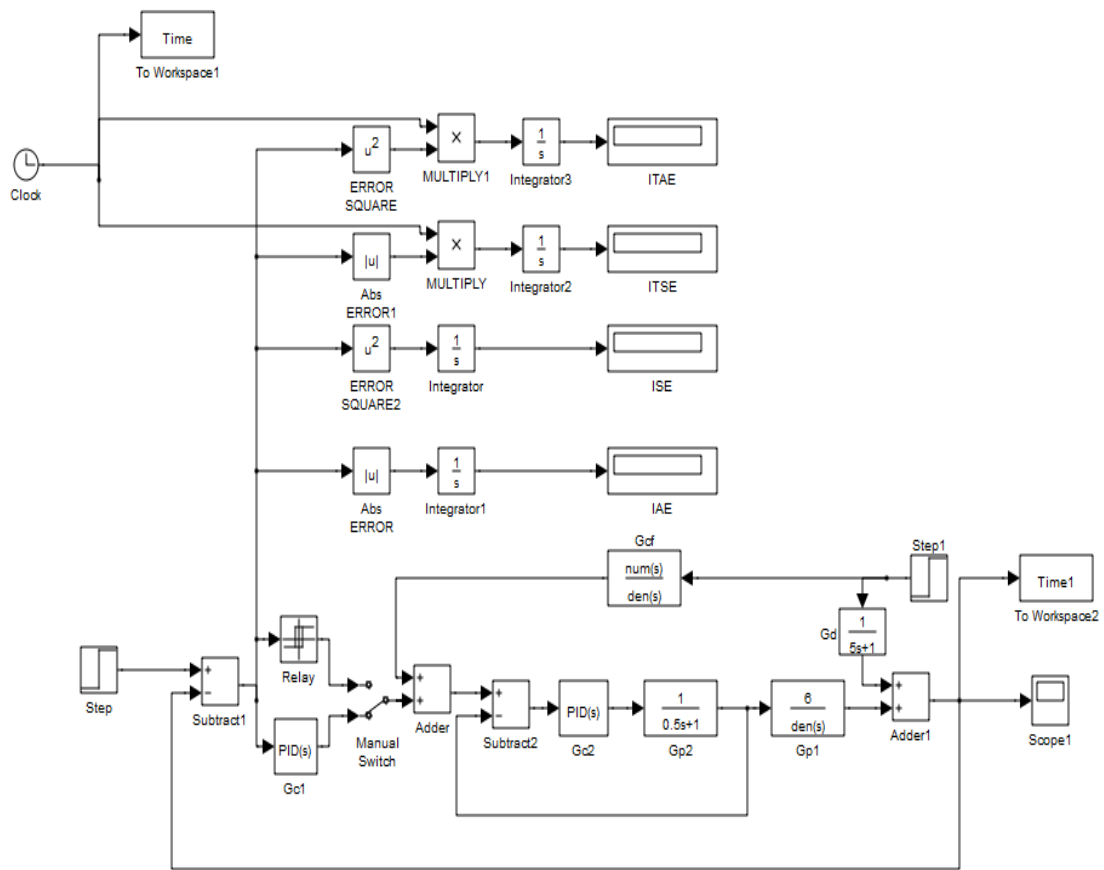


Figure 5.30: MATLAB Simulink diagram for cascade control by RA method



**Figure 5.31:** MATLAB Simulink diagram for cascade plus feed forward control by RA method

**Table 5.15:** The results of PID tuning parameters

| S.NO | CONTROL SCHEMES            | TUNING METHODS | $K_P$    | $K_I$    | $K_D$    |
|------|----------------------------|----------------|----------|----------|----------|
| 1    | FEED FORWARD PLUS FEEDBACK | ZN             | 1.00002  | 0.16798  | 1.488279 |
|      |                            | RA             | 0.98028  | 1.535    | 1.565    |
| 2    | CASCADE                    | ZN             | 0.5958   | 0.07341  | 1.2087   |
|      |                            | RA             | 0.581028 | 0.068858 | 1.22568  |
| 3    | CASCADE PLUS FEED FORWARD  | ZN             | 0.59544  | 0.07058  | 1.25578  |
|      |                            | RA             | 0.58101  | 0.069437 | 1.215385 |

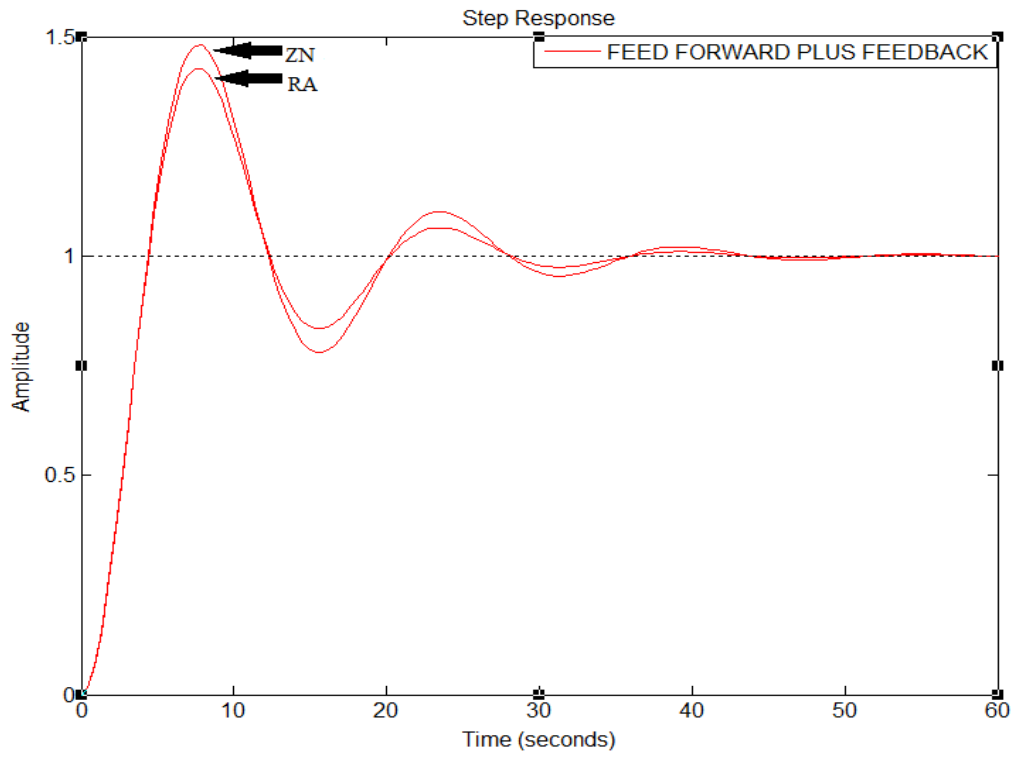
**Table 5.16:** The results of PID Performance Criteria

| S.NO | CONTROL SCHEMES            | TUNING METHODS | Rise time( $t_r$ ) in sec | Settling time( $t_s$ ) in sec | Peak time( $t_p$ ) in sec | Overshoot (%) | Steady state error ( $e_{ss}$ ) in sec |
|------|----------------------------|----------------|---------------------------|-------------------------------|---------------------------|---------------|--|
| 1    | FEED FORWARD PLUS FEEDBACK | ZN             | 2.94                      | 39.9                          | 7.89                      | 48.1          | 0                                      |
|      |                            | RA             | 2.98                      | 33                            | 7.45                      | 42.8          | 0                                      |
| 2    | CASCADE                    | ZN             | 3.99                      | 33.9                          | 10.5                      | 38            | 0                                      |
|      |                            | RA             | 4.06                      | 33.1                          | 11                        | 34.4          | 0                                      |
| 3    | CASCADE PLUS FEED FORWARD  | ZN             | 3.99                      | 32.9                          | 10.3                      | 35.3          | 0                                      |
|      |                            | RA             | 4.06                      | 33.4                          | 10.6                      | 35            | 0                                      |

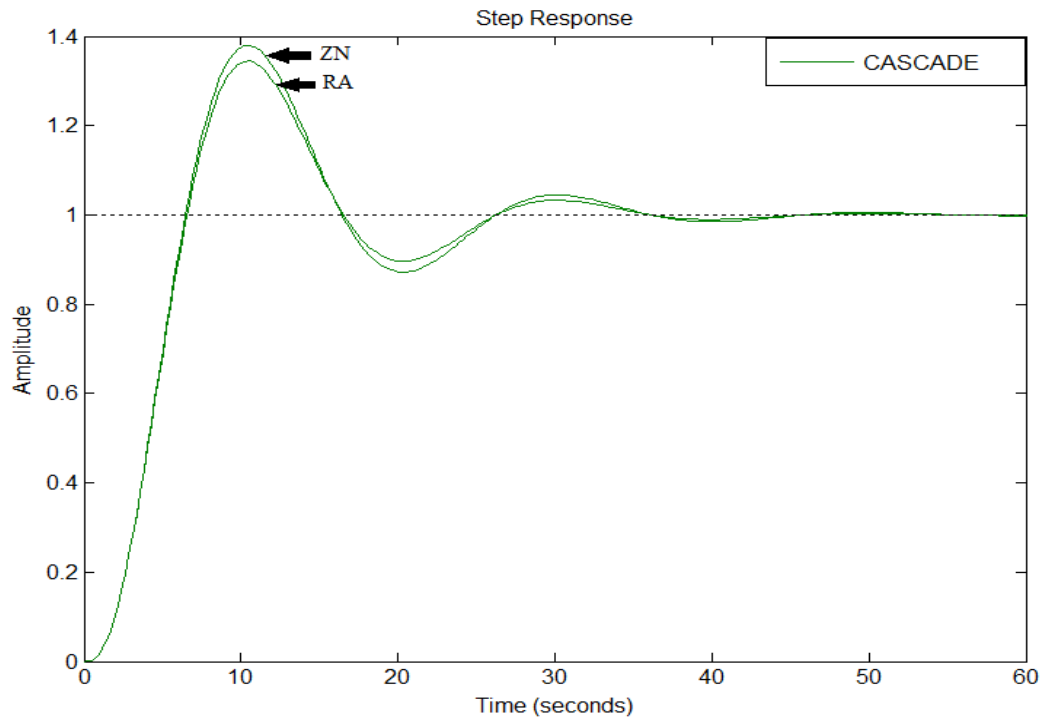
**Table 5.17:** Comparison of Performance Indices

| S.NO | CONTROL SCHEMES            | TUNING METHODS | IAE   | ISE   | ITSE  | ITAE  |
|------|----------------------------|----------------|-------|-------|-------|-------|
| 1    | FEED FORWARD PLUS FEEDBACK | ZN             | 4.103 | 1.377 | 39.62 | 5.365 |
|      |                            | RA             | 3.579 | 1.276 | 28.99 | 4.219 |
| 2    | CASCADE                    | ZN             | 7.569 | 3.978 | 73.43 | 19.31 |
|      |                            | RA             | 7.083 | 3.827 | 62.76 | 17.13 |
| 3    | CASCADE PLUS FEED FORWARD  | ZN             | 6.585 | 3.271 | 58.05 | 14.63 |
|      |                            | RA             | 6.658 | 3.313 | 59.15 | 14.93 |

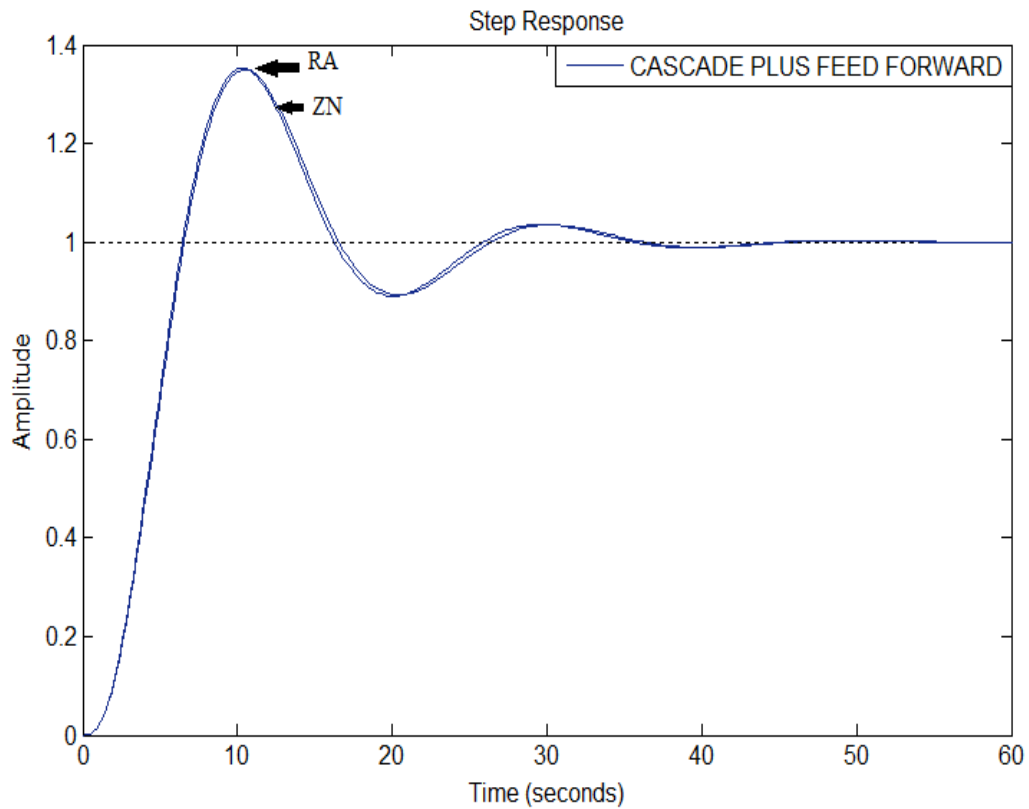
The performances of various tuning methods have been compared by applying a step input to the given process. Simulation results shows that tuning a PID controller with Relay auto (RA) tuning method results in less settling time ( $t_s$ ), overshoot (%), ISE, IAE, ITAE and ITSE. Simulation results also show that tuning a PID controller with Relay Auto (RA) tuning method is better than Ziegler Nichols (ZN) tuning method.



**Figure 5.32:** Unit step response for feed forward plus feedback control by using ZN and RA method



**Figure 5.33:** Unit step response for cascade control by using ZN and RA method



**Figure 5.34:** Unit step response for cascade plus feed forward control by using ZN and RA method

Unit step response for feed forward plus feedback, cascade and cascade plus feed forward control by using ZN and RA method is shown in Figures 5.32 to 5.34. From above response it is clear that RA method is better than ZN for feed forward plus feedback and cascade control. ZN method is better than RA tuning method for cascade plus feed forward control.

# CHAPTER 6

## **Conclusion and Future Scope**

---

In this dissertation, performance evaluation of different control schemes such as feedback, feedback plus feed forward, cascade and cascade plus feed forward using various conventional tuning methods like Ziegler-Nichols, Relay Auto tuning, Modulus Optimum and Computational Optimization has been compared. The performance comparison is in terms of rise time, settling time, peak time, overshoot and steady state error by applying a step response to a particular scheme. For feed forward plus feedback control scheme Relay Auto tuning method is better over Ziegler-Nichols method when less settling time, peak time, overshoot, IAE, ISE, ITSE, ITAE are required. While Ziegler-Nichols method is best when less rise time (2.94 sec in comparison to 2.98 sec of Relay Auto tuning) is required. For cascade control scheme Relay Auto tuning method gives better performance over Ziegler-Nichols method when less settling time, overshoot, IAE, ISE, ITSE, ITAE are required. While Ziegler-Nichols method is best when less rise time (3.99 sec in comparison to 4.06 sec of Relay Auto tuning method) and less peak time (10.5 sec in comparison 11 of Relay Auto tuning) are required. For cascade plus feed forward control scheme Ziegler-Nichols method provide better performance over Relay Auto tuning method when less rise time, settling time, peak time, IAE, ISE, ITSE, ITAE are required. While Relay Auto tuning method is best when less overshoot (35% in comparison 35.3% of Ziegler-Nichols method) is required.

### **Future Scope**

Any work however precise it may be always have the scope of future improvement. On the same lines, although a lot of work has been done in the dissertation performance evaluation of different control scheme using various tuning methods but still the tuning of PID controller using various intelligent methods such as Fuzzy logic, Genetic Algorithm,

Artificial Neural Network etc. may be done and their performance analysis may be carried out to find the best method for a particular application.

## References

1. N .Muruganath and B. Niagaraj: A Comparative Study of PID Controller Tuning Using GA, EP, PSO and ACO, 978-1-4244-7770-8/10/\$26.00 ©2010 IEEE, pp 305.
2. Mehmet Korkmaz, Ömer Aydoğdu and Hüseyin Doğan, “Design and Performance Comparison of Variable Parameter Nonlinear PID Controller and Genetic Algorithm Based PID Controller”, 978-1-4673-1448-0/12/\$31.00 ©2012 IEEE.
3. Kwang Myung, Yu and Jong An, Kim, “Model Reference PID Control and Tuning for Steam Temperature in Thermal Power Plant”, 2011 11th International Conference on Control, Automation and Systems Oct. 26-29, 2011 in KINTEX, Gyeonggi-do, Korea.
4. Salah kermiche and Hadj Ahmed Abbassi, “Fuzzy PID Controller Parameters Optimized by Genetic Algorithms”.
5. Wei-Min Qi1, Wei-You Cail, Qiao-ling Jil and Yuan-Chu Cheng, “A Design of Nonlinear Adaptive PID Controller Based ON Genetic Algorithm”, Proceedings of the 25th Chinese Control Conference 7-11 August, 2006, Harbin, Heilongjiang.
6. Guohan Lin and Guofan Liu, “Tuning PID Controller Using Adaptive Genetic Algorithms”, The 5th International Conference on Computer Science & Education Hefei, China. August 24–27, 2010.
7. Neil Kuyvenhoven, “PID Tuning Methods An Automatic PID Tuning Study with MathCad”, Neil Kuyvenhoven Calvin College ENGR. 315 December 19, 2002
8. Rao V.Dukkipati, “Analysis and design of control systems using MATLAB”, New age international publishers, pp 4-5.
9. Fayezyf . M. El-sousy, “design and implementation of 2dof i-pd controller for indirect field orientation induction machine system”, 0-7803-7090-2/01/\$10.000 2001 IEEE.
10. Jialong Chen, Guoqing Xu, Kun Xu, and Weimin Li, “Traction Control for Electric Vehicles: A Novel Control Scheme”, 978-1-4673-2237-9/12/\$31.00 ©2012 IEEE.

11. Mohsin A. Bandi and Mr. Naimesh B. Mehta, "Universal Controller Design Using Arm Controller", International Journal of Engineering Trends and Technology- Volume3Issue2- 2012.
12. Robert Rice and Douglas Cooper, "A Rule Based Design Methodology for the Control of Non Self-Regulating Processes", Copyright 2004 by ISA – The Instrumentation, Systems and Automation Society. Presented at the ISA 2004, 5-7 October 2004, Reliant Center Houston, Texas, [www.isa.org](http://www.isa.org).
13. B.Wayne Bequette, "process control modeling, design and simulation", PHI, pp 332
14. Sihai Song, Wenjian Cai and Ya-Gang Wang, "Auto-tuning of cascade control systems", 0019-0578/2003/\$ - see front matter © 2003 ISA The Instrumentation, Systems, and Automation Society.
15. R. E. Brown, G. N. Maliotis and J. A. Gibby, " PID Self-tuning Controller for Aluminum Rolling Mill", 90KH 2935-5/90/0@%1374%01.0001990 IEEE.
16. Roman W. Swiniarski, "Neural Network Based Self-Tuning PID Controller with Fourier Transformation of Temporal Patterns", 0879426004/90/1100-1227\$01.00 0 1990 IEEE.
17. Roman W. Swliniarski, "Novel Neural Network Based Self-Tuning PID Controller Which uses Pattern Recognition Technique".
18. David W. Augustine and K. S. P. Kumar, "A Method For Self-Tuning A Pid Controller For Control Of Small To Medium Sized Diesel Engines", CH3051-0/91/0000-0085 \$1.00 0 1991 IEEE.
19. M Zhuang D P Atherton, "Tuning Pid Controllers With Integral Performance Criteria".
20. P. Wang and D.P. Kwok, "Auto-Tuning Of Classical Pid Controllers Using An Advanced Genetic Algorithm", 0-7WM582-5 /92\$3.0001992 IEEE.
21. Hyeong-Pyo Hong, Suk-Joon Park, Sang-Joon Han Kyeong-Young Cho, Young-Chul Lim, Jong-Kun Park and Tae-Gon Kim, "A Design Of Auto-Tuning Pid Controller Using Fuzzy Logic", 0-7803-0582-5/ 92\$3.0001992 IEEE.

22. Cheng Ling and Thomas F. Edgar, "A New Fuzzy Gain Scheduling Algorithm for Process Control", 1992 ACC/TP3.
23. Vladimir Bobal, Marek Kubalcik, "Self-Tuning Controller For Temperature Control Of A Thermo-Analyzer", 0-7803-1872-2/94/\$4.CXIO 1994 IEEE.
24. Robert P. Copeland and Kuldip S . Rattan, "A Fuzzy Logic Supervisor for PID Control of Unknown Systems", 0-7803-1 990-7/94/\$4.00 91 994 IEEE.
25. G. CAECEV and R. GOREZ, "Iterative Technique for PID Controller Tuning", 0-7883-2685-7195\$ 4.00 0 1995 I 3209 IEEE.
26. Hyun-Joon Cho, Kwang-Bo Cho, and Bo-Hyeun Wang. "Automatic rule generation using genetic algorithms for Fuzzy-PID hybrid control", 0-7803-2978-3/96/\$5.00 0 1996 IEEE.
27. J.M. Soma, R. BabuSka and H.B. Verbruggen, "Internal Model Control with a Fuzzy Model:Application to an Air-conditioning System", 0-7803-3796 4/97/\$10.0001997 IEEE.
28. Yongling Zheng, Longhua Ma, Liyan Zhang and Jixin Qian, "Robust PID Controller DesignUsing Particle Swarm Optimizer", 0-7803-7891-1/03/\$17.00 0 2003 IEEE.
29. Francesco Cupertino, Vincenzo Giordano, David Naso, Luigi Salvatore, and Biagio Turchiano, "Optimization of fuzzy controllers for industrial manipulators via genetic algorithms", 0-7803-7906-3/03/\$17.00 02003 IEEE.
30. Damir VranB, Birgitta Kristiansson and Stanko Strmtnik, "Reduced MO Tuning Method for PID Controllers", 2004 5th Asian Controt Conference.
31. Huseyin Atakan Varol and Zafer Bingul, "A New PID Tuning Technique Using Ant Algorithm", 0-7803-8335-4/041\$17.00 02004 AACC.
32. Faisal A. Mohamed and Heikki N. Koivo, "Diesel Engine Systems with Genetic Algorithm Self Tuning PID Controller".
33. Asim Ali Khan and Nishkam Rapal, "Fuzzy PID Controller: Design, Tuning and Comparison with Conventional PID Controller", I1-4244-0457-6/06/\$20.OO ©2006 IEEE.

34. Junfeng Chen, Ziwu Ren and Xinnan Fan, “ Particle Swarm Optimization with Adaptive Mutation and Its Application Research in Tuning of PID Parameters”, Manuscript received November 25, (2005), pp 990-991.
35. M. Willjuice Iruthayarajan, and S. Baskar, “Optimization Of Pid Parameters Using Genetic Algorithm And Particle Swarm Optimization, “IET-UK International Conference on Information and Communication Technology in Electrical Sciences.
36. Xiao-Gang Duan, Han-Xiong Li and Hua Deng, “A Simple Tuning Method for Fuzzy PID Control”, 978-1-4244-1819-0/08/\$25.00 c\_2008 IEEE.
37. Dongyun Wang and Guan Wang, “Parameters Optimization of Fuzzy Controller Based on Improved Particle Swarm Optimization” , International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 978-0-7695-3278-3/08 \$25.00 ©( 2008) IEEE,pp 918.
38. Salah kermiche and Hadj Ahmed Abbassi, “Fuzzy PID Controller Parameters Optimized by Genetic Algorithms”.
39. YANG Bo , LI Wan-zhou and YANG Feng , A New PSO-PID Tuning Method for Time-delay Processes , 1-4244-2386-6/08/\$20.00 ©(2008) IEEE.
40. Liu Fan and Er Meng Joo: Design for Auto-tuning PID Controller Based on Genetic Algorithms , 978-1-4244-2800-7/09/\$25.00 ©2009 IEEE ,pp 1924.
41. Guohan Lin and Guofan Liu, “Tuning PID Controller Using Adaptive Genetic Algorithms”, 978-1-4244-6005-2/10/\$26.00 ©2010 IEEE.
42. H.M.Asifa and S.R.Vaishnav, “Particle Swarm Optimisation Algorithm based PID Controller”, 978-0-7695-4246-1/10 \$26.00 © 2010 IEEE DOI 10.1109/ ICETET. 2010.145.
43. GAN Shu-chuan and Guo Hui, “The Research of PID Self-tuning Based on Fuzzy Genetic Algorithm”, 978-1-61284-181-6/11/\$26.00 ©2011 IEEE.

44. Mazidah Tajjudin, Ramli Adnan, Norlela Ishak, Mohd Hezri Fazalul Rahiman ,and Hashimah Ismail “Model Reference Input for an Optimal PID Tuning Using PSO”, 2011 IEEE International Conference on Control System, Computing and Engineering pp 163.
45. Geetha. M , Balajee. K. A and Jovitha Jerome, “Optimal Tuning of Virtual Feedback PID Controller for a Continuous Stirred Tank Reactor (CSTR) using Particle Swarm Optimization (PSO) Algorithm”, ISBN: 978-81-909042-2-3 ©2012 IEEE.
46. Y.C. Kim, L.H. Keel and S.P. Bhattacharyya, “Transient Response Control via Characteristic Ratio Assignment”, O-7803-7298-0/02/\$17.00 © 2002 AACC.
47. Jeen-Shing Wang and C. S. George Lee, “Self-Adaptive Recurrent Neuro-Fuzzy Control of an Autonomous Underwater Vehicle”, Ieee Transactions On Robotics And Automation, Vol. 19, No. 2, April 2003.
48. Serdar Y˘uksel, Haitham Hindi and Lara Crawford, “Optimal Tracking with Feedback-Feedforward Control Separation over a Network”, 1-4244-0210-7/06/\$20.00 ©2006 IEEE.
49. A. E. Le´on, J. A. Solsona, C. Busada, H. Chiacchiarini and M. I. Valla, “A Novel Feedback/Feedforward Control Strategy for Three-Phase Voltage-Source Converters”, 1-4244-0755-9/07/\$20.00 ©2007 IEEE.
50. Wei Tang and Qian Feng Mengxiao Wang , Qian Hou and Leilei Wang, “Expert System Based Dissolved Oxygen Control in APMP Wastewater Aerobic Treatment Process”, 978-1-4244-2503-7/08/\$20.00 © 2008 IEEE.
51. Ieroham S. Baruch, and Rosalba Galvan-Guerra, “Decentralized Direct Adaptive Fuzzy-Neural Control of an Anaerobic Digestion Bioprocess Plant”, 978-1-4244-1739-1/08/\$25.00 ©2008 IEEE.
52. Wei Tang, Zhongfeng Wang, Qian Feng, Mengxiao Wang, “Application of Fuzzy Expert Control to APMP Pulping Wastewater Treatment Process of Aerobic”, 978-1-4244-5141-8/10/\$26.00 © 2010 IEEE.

53. Eugen Horatiu Gurban and Gheorghe-Daniel Andreescu, "Comparison Study of PID Controller Tuning for Greenhouse Climate with Feedback-Feedforward Linearization and Decoupling".
54. Douglas J.Cooper, "Practical Process Control using control station 3.7".
55. Hari Om Bansal, Rajamayyoor Sharma and P. R. Shreeraman, "PID Controller Tuning Techniques: A Review", JCET Vol. 2 Iss. 4 October 2012 PP. 168-176 www.ijcet.org ©C World Academic Publishing.
56. Robert P. Copeland and Kuldip S . Rattan, "A Fuzzy Logic Supervisor for PID Control of Unknown Systems", 0-7803-1 990-7/94/\$4.00 91 994 IEEE..
57. Nedjeljko Perić, Ivan Branica, Ivan Petrović, "Modification And Application Of Autotuning PID Controller".
58. Katsuhiko Ogata: Modern Control Engineering, Fifth Edition, Prentice Hall, 2010 ,pp 583.
59. R. Matousek, IAENG, P. Minar, S. Lang, and P. Pivonka, "HC12: Efficient Method in Optimal PID Tuning", Proceedings of the World Congress on Engineering and Computer Science 2011 Vol I WCECS 2011, October 19-21, 2011, San Francisco, USA.
60. Wen Tan, Jizhen Liu,Tongwen Chenb, and Horacio J. Marquez , " Comparison of some well-known PID tuning formulas", Science Direct Received 10 June2005; received in revised form 15 January 2006; accepted 3 April 2006 Available online 22 May 2006.
61. Mohammad Ali Nekoui, Mohammad Ali Khameneh and Mohammad Hosein Kazemi, "Optimal Design of PID Controller for a CSTR System Using Particle Swarm Optimization", 14th International Power Electronics and Motion Control Conference, EPE-PEMC (2010).
62. S. Pandaa, B.K.Sahu and P.K.Mohanty, "Design and performance analysis of PID controller for an automatic voltage regulator system using simplified particle swarm optimization", Journal of the Franklin Institute 349 (2012) 2609–2625.

63. Mojtaba Ahmadiéh Khanesar, Hassan Tavakoli, Mohammad Teshnehláb, “Novel Binary Particle Swarm Optimization”, [www.intechopen.com](http://www.intechopen.com).
64. Liu-Hsu Lin, Fu-Cheng Wang and Jia-Yush Yen, “Robust PID Controller Design Using Particle Swarm Optimization”, 978-89-956056-9-1/09/©2009 ACA .
65. Ming-Feng Yeh, Min-Shyang Leu and Kai-Min Chen, “Design of PID Controller Using Hybrid Particle Swarm Optimization”, Proceedings of 2012 International Conference on Fuzzy Theory and Its Applications National Chung Hsing University, Taichung, Taiwan, Nov.16-18, 2012.
66. S. Bouallègue, J. Haggège and M. Benrejeb. “A New Method for Tuning PID-Type Fuzzy Controllers Using Particle Swarm Optimization”, <http://dx.doi.org/10.5772/47139>.
67. Men-Shen Tsai and Wu-Chang Wu, “Novel Binary Coding Particle Swarm Optimization for Feeder Reconfiguration”, [www.intechopen.com](http://www.intechopen.com).
68. Po-Hung Chen, “Particle Swarm Optimization for Power Dispatch with Pumped Hydro”, [www.intechopen.com](http://www.intechopen.com).
69. Zwe-Lee Gaing, Member, IEEE, “A Particle Swarm Optimization Approach for Optimum Design of PID Controller in AVR System”, Ieee Transactions On Energy Conversion, Vol. 19, No. 2, June (2004), pp 386.
70. Majid M. Khodier, Member, IEEE and Christos G. Christodoulou, Fellow, IEEE, “Linear Array Geometry Synthesis With Minimum Sidelobe Level and Null Control Using Particle Swarm Optimization”, IEEE Transactions On Antennas And Propagation, Vol. 53, No. 8, August (2005), pp 2675.
71. Yongling Zheng, Longhua Ma, Liyan Zhang and Jixin Qian, “Robust PID Controller Design Using Particle Swarm Optimizer”, 0-7803-7891-1/03/\$17.00 © 2003 IEEE.
72. Mohammad DASHTI, Kambiz SHOJAEE, S.M.H. SEYEDKASHI and Mojtaba BEHNAM, “Tuning of Digital PID Controller Using Particle Swarm Optimization”, Proceedings of the 29th Chinese Control Conference July 29-31,( 2010), Beijing, China, pp 3384-3385.

73. M. B. Abdelhalim and S. E. –D. Habib, “Particle Swarm Optimization for HW/SW Partitioning”, [www.intechopen.com](http://www.intechopen.com).
74. Fundamentals of Particle Swarm Optimization Techniques.
75. Adel M. Sharaf and Adel A. A. El-Gammal, “An Integral Squared Error -ISE Optimal Parameters Tuning of Modified PID Controller for Industrial PMDC Motor Based on Particle Swarm Optimization-PSO”, 978-1-4244-3557-9/09/\$25.00@2009IEEE.

## **Publications**

- (1) Rajeev Kumar, Sunil K. Singla and Vikram, “A Comparative Analysis of different Methods for the tuning of PID Controller”, **International Journal of Electronics Communications and Electrical Engineering**, ISSN: 2277-7040 Volume 3 Issue 2 (February 2013).
- (2) Rajeev Kumar, Sunil K. Singla and Vikram, “PID CONTROLLER TUNING USING PARTICLE SWARM OPTIMIZATION”, **International Journal of Advanced Science and Technology**, ISSN: 2005-4238 (communicated).
- (3) Rajeev Kumar, Sunil K. Singla and Vikram, “Comparison among some well-known control schemes with different tuning methods”, **Journal of Applied Research and Technology**, ISSN: 1665-6423 (communicated).
- (4) Rajeev Kumar, Sunil K. Singla and Vikram, “Tuning of PID Controller: A Review”, **Acta Montanistica Slovaca**, ISSN: 1335-1788 (To communicate).