SWARM INTELLIGENCE BASED TUNING OF CONTROLLER PARAMETERS FOR CONCENTRATION CONTROL OF ISOTHERMAL CONTINUOUS STIRRED TANK REACTOR

A Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering

in

Electronic Instrumentation and Control

Submitted by

Vishal Vishnoi
Roll No: 801051027

Under the Guidance of

Dr. Gagandeep Kaur
Assistant Professor

Department of Electrical and Instrumentation Engineering

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

Patiala, 147004, Punjab, India

June 2012

II

DECLARATION

I hereby certify that the work is being presented in the thesis work entitled **"Swarm Intelligence Based Tuning of Controller Parameters for Concentration Control of Isothermal Stirred Tank Reactor"** in partial 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. Gagandeep Kaur,** Assistant Professor, Department of Electrical and Instrumentation Engineering, Thapar University, Patiala, Punjab.

Date: 5/6/12

Vishal Vishnoi

Roll No: 801051027

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

Date: 5/6/12

Dr. Gagandeep Kaur
Assistant Professor,
Department of Electrical
Instrumentation Engineering,
Thapar University, Patiala,
Punjab

Countersigned By

Dr. Smarajit Ghosh
Head of Department,
Department of Electrical and
Instrumentation Engineering,
Thapar University, Patiala,
Punjab

Dr. Saroj Kumar Mohapatra
Dean of Academic Affairs,
Thapar University, Patiala,
Punjab

ABSTRACT

Basically chemical reactor is a reactor which is used to perform chemical operations. Chemical reactors are of many types like isothermal chemical reactor and non-isothermal chemical reactor. In recent years control of process plant has gained a widespread research interest. Multi input multi output control, RGA based decoupling control are some of the advanced control techniques which are being used in industries.

The increasing complexity of modern control systems has emphasized the idea of applying new approaches in order to solve design problems for different control engineering applications. Proportional-Integral-Derivative (PID) control schemes have been widely used in most of process control systems represented by chemical processes for a long time. However, tuning of PID controller is a very complex task, involving a lot of design and process dynamic aspect. Swarm intelligence, which has caught the eyes of researchers due to its simplicity, low computational cost, and good performance, makes it a possible choice for tuning of PID controllers, to increase their performance.

This dissertation discusses, in detail, the Particle Swarm Optimization (PSO) algorithm, and its implementation in PID controller tuning to get the optimal tuning parameters. The PID controller is used to control the product concentration of isothermal CSTR. First of all mathematical modeling of the process is performed using experimental plant data. After the mathematical modeling, different controllers are designed to meet the control objective. PSO is used to optimally tune the PID controller.

ACKNOWLEDGEMENT

The real spirit of achieving a goal is through the way of excellence and austere discipline. I would have never succeeded in completing my task without the cooperation, encouragement and help provided to me by various personalities.

I am very thankful to **Dr. Abhijit Mukherjee**, Director of Thapar University, Patiala for providing the facilities for the completion of this thesis.

I express my deep sense of gratitude towards **Dr. Smarajit Ghosh**, Professor and Head of the Department of Electrical & Instrumentation Engineering, Thapar University, Patiala who has been a constant source of inspiration for me throughout this work.

With deep sense of gratitude I express my sincere thanks to my esteemed and worthy supervisor **Dr. Gagandeep Kaur**, Department of Electrical and Instrumentation Engineering, Thapar University, Patiala for her valuable guidance in carrying out this work under her effective supervision, encouragement, enlightenment and cooperation. Most of the novel ideas and solutions found in this thesis are the result of our numerous stimulating discussions. Her feedback and editorial comments were also invaluable for writing of this thesis.

We express our deep sense of gratitude to Mr. Subhransu Padhee for his valuable and encouraging guidance, which played a major role in the formation of this project work. Their co-oporation and timely suggestions have unparallel stimuli for us to travel eventually towards the completion of this project.

I am also thankful to all the staff members of the Department for their full cooperation and help. This acknowledgement would be incomplete if I do not mention the emotional support and blessings provided by my friends. I had a pleasant enjoyable and fruitful company with them.

Vishal Vishnoi

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

CSTR	Continuous stirred tank reactor
CC	Concentration controller
CM	Concentration measurement
ρ	Liquid phase density
V	Volume
F	Volumetric flow rate
C_{Afs}	Steady state feed concentration
r	Molar rate
x	State variable
y	Output variable
u	Input variable
PI	Proportional integral
PD	Proportional derivative
PID	Proportional integral derivative controller
b	Controller bias
K_c	Controller gain
K_p	Proportional gain
K_i	Integral gain
K_d	Derivative gain
T_i	Integral time
T_d	Derivative time
PB	Proportional band
ISE	Integral square error
IAE	Integral absolute error
ITSE	Integral time square error
ITAE	Integral time absolute error
AI	Artificial intelligence
HUMINT	Human intelligence
FIS	Fuzzy Inference System
MIMO	Multiple Input and Multiple Output
MISO	Multiple Input and Single Output

ANT	Ant colony optimization
PSO	Particle swarm optimization
GA	Genetic algorithm
T_p	Peak time
T_r	Rise time
T_s	Settling Time
M_p	Peak Overshoot
T_d	Delay time
β	Scaling factor
F	Fitness function

RELATED PUBLICATION

International Journal:

- Vishal Vishnoi, Gagandeep Kaur, Subhransu Padhee "Controller Performance Evaluation for Concentration Control of Isothermal Continuous Stirred Tank Reactor" International Journal of Scientific and Research Publications, Volume 2, Issue 6, June 2012 (ISSN 2250-3153).

Chapter 1

Introduction

1.1 Overview:

In recent years control of process plant has gained a widespread research interest. Multi input multi output control, RGA based decoupling control are some of the advanced control techniques used in industries. Basically chemical reactor is a reactor which is used to perform chemical operations. Chemical reactors are of many types like isothermal chemical reactor and non-isothermal chemical reactor.

Proportional-Integral-Derivatives (PID) controller or three term controller is the most widely used controller in industry process control for 50 years, even though great progress in industrial control theory has been made over the period. This is mainly due it simplicity which makes PID controller easier to be understood by the control engineer or operator and found to be adequate for most plant such as chemical plant etc. An investigation performed in 1989 in Japan indicated that more than 90% of the controllers used in process industries are PID controllers.

The implementation of PID controllers needs proper tuning of proportional gains, integral gains, and derivative gains of the controllers. A conventional PID controller may have poor control performance, when it is used for controlling non-linear and complex processes. In this dissertation work, PSO programs are used to determine the optimal values of the PID controller parameters to improve the transient response of the system.

1.2 Objective:

The main objective of this thesis is to control the concentration of isothermal continuous stirred tank reactor using conventional and intelligent controllers. This thesis considers van-de-vusee reaction scheme and implements control techniques to control the concentration of product fluid by manipulating the flow of inlet fluid. This thesis uses conventional PID controller, but the main disadvantage of PID controller is that tuning of PID controller is really difficult. So a swarm intelligence based technique; particle swarm optimization (PSO) is used to optimally tune the PID controllers. A comparative study of control performance of PID controller and intelligent controller is performed.

1.3 Organization of thesis:

The dissertation is organized as follows:

Chapter 1 gives introduction of thesis.

Chapter 2 discusses about different chemical reactors and mathematical modeling of isothermal CSTR.

Chapter 3 gives a relevant literature review regarding concentration control of CSTR.

Chapter 4 gives the basic idea of the control system and controller design.

Chapter 5 discusses about Intelligence and it's types, and soft computing techniques.

Chapter 6 is the problem formulation.

Chapter 7 shows the results and discussion.

Chapter 8 provides conclusion of entire thesis work and proposes the future scope.

Chapter 2

Chemical Reactor

2.1 Introduction:

A chemical reactor is a device which is used to contain controlled chemical reactions. These reactions take place inside the reactor, in conditions which can be monitored and controlled for safety and efficiency [1]. These types of reactors are used in the production of chemicals such as components of pharmaceutical compounds, and they can operate in several different ways. A number of scientific specialty companies produce chemical reactors and accessories such as replacement components for damaged devices.

Chemical reactors can be used as either tanks or pipes, depending on the needs, and they can vary in size considerably. Small bench top chemical reactor designs are intended for use in labs, for example, while large tanks can be used to make chemicals on an industrial scale [2]. The design also includes a variety of features which can be used to control conditions inside the reactor.

If we talk about the batch chemical reactor and continuous chemical reactor then we can say that with a batch chemical reactor, the components of the reaction are added to the reactor and a controlled reaction is allowed to take place. When the reaction is finished, the batch can be removed and the reactor can be prepared for another round. This type of reactor works best when people need chemicals on a small scale, as for example when research chemists are preparing compounds for pharmaceutical research. Continuous chemical reactors operate continuously, as long as the materials needed for the reaction are supplied. These are used to create a steady supply of a needed chemical. Continuous reactors are commonly used in the manufacture of industrial chemicals, when the need for a chemical is high and very consistent. These reactors are periodically shut down for maintenance or when they are not needed, in which case special steps may need to be taken when they are restarted so that their functionality will not be impaired [3].

These devices are designed by chemical engineers who are familiar with the needs of chemical reactors and the various ways in which they can be used.

For special applications, an engineer may design a custom reactor which is specifically built for the purpose, in which case the engineer is also involved in the design of the space where the reactor will be used, to ensure that it conforms with safety guidelines and to confirm that the space has been properly designed to accommodate the chemical reactor.

2.2 Chemical reactor:

Chemical reactors are the most important unit of a chemical plant used for unit operations. Basically a chemical reactor is a device in which chemical reaction takes place. Chemical reactors can be classified according to different properties.

1. Reaction phase
2. Operating modes

1. **Reaction Phase:** In industrial chemical processes, Phase reactors have a wide range of applications such as oxidation, hydrogenation, hydro-desulfurization. Multiphase reactors are defined as reactors with atleast two distinct phases in contact. Reactors, in general, may be classified based on the number of phase coexistence into the following category [4]:
   (a) Homogeneous reactor
   (b) Heterogeneous reactor

(a) Homogeneous reactor: One phase such as gas or liquid exists in the reactors. The hydrodynamic flow characteristics of the mixture which determines the reactor type such as plug, CSTR or batch. Examples of homogeneous reactions are gaseous fuel combustion (gas phase) and acid-base neutralization (liquid phase).

(b) Heterogeneous reactor: Two distinct phases of reactants (or catalyst) coexist. Examples of heterogeneous systems are carbon dioxide absorption into alkali (gas-liquid); coal combustion and automobile exhaust purification (gas-solid); water softening (liquid-solid); coal liquefaction and oil hydrogenation (gas-liquid-solid); and cake reduction of iron ore (solid-solid). This category may be classified into the following subcategories:

Catalyst reactors: Gas or liquid phase (or both) is in contact with a catalyst (mainly solid, but could be another liquid phase).

Example of this category include the (catalytic packed-bed catalytic gas reaction) and three phase trickle-bed (catalytic gas ± liquid reaction).

Non-catalytic reactors: Gas ± liquid or liquid ± liquid reactions are carried in a variety of contact vessels such as the gas ± liquid continuously stirred tank reactor.

2. **Operating modes:** Chemical reactors may be operated in batch, semibatch, or continuous modes.

(a) Continuous mode

(b) Batch mode

(c) Semi batch mode

When a reactor is operated in a **batch mode**, the reactants are charged, and the vessel is closed and brought to the desired temperature and pressure. These conditions are maintained for the time needed to achieve the desired conversion and selectivity, that is, the required quantity and quality of product. At the end of the reaction cycle, the entire mass is discharged and another cycle is begun. Batch operation is labor-intensive and therefore is commonly used only in industries involved in limited production of fine chemicals, such as pharmaceuticals. In a **semibatch reactor operation**, one or more reactants are in the batch mode, while the coreactant is fed and withdrawn continuously. In a chemical reactor designed for **continuous operation**, there is continuous addition to, and withdrawal of reactants and products from, the reactor system [1] [5].

2.3 Designing of a chemical reactor:

The design of a chemical reactor deals with multiple aspects of chemical engineering. Chemical engineers design reactors to maximize net present value for the given reaction. Designers ensure that the reaction proceeds with the highest efficiency towards the desired output product, producing the highest yield of product while requiring the least amount of money to purchase and operate.

While designing a chemical reactor following factor has to be considered

1. Overall size of reactor
2. Products emerging from reactor
3. Temperature inside the reactor

4. Pressure inside the reactor
5. Rate of reaction
6. Activity and mode of catalyst
7. Stability and Controllability of reactor

The design and operation of chemical reactors are dictated by the state of aggregation of the interacting substances and by the conditions (temperature, pressure, reactant concentrations) required to ensure the desired reaction rate and direction. According to the first criterion, a distinction is made between chemical reactors designed for reactions in homogeneous systems (single-phase gaseous or liquid) and reactors used for heterogeneous systems (multiphase, for example, gaseous-liquid-solid). The second criterion is used to classify reactors as low-, medium-, or high-pressure; low- or high-temperature; and batchwise, semicontinuous, or continuous-operation. The sizes of the reactors may be varied in order to minimize the *total* capital investment required to implement the process [1],[5].

2.4 Continuous stirred tank reactor:

Continuous Stirred Tank Reactor System (CSTR) is a typical chemical reactor system with complex nonlinear dynamic characteristics. There has been considerable interest in its state estimation and real time control based on mathematical modeling. However, The lack of understanding of the dynamics of the process, the highly sensitive and nonlinear behaviour of the reactor, has made difficult to develop the precise mathematical modeling of the system. The CSTR is an easily constructed, versatile and cheap reactor, which allows simple catalyst charging and replacement. Its well -mixed nature permits straightforward control over the temperature and pH of the reaction and the supply or removal of gases. There are two types of continuous stirred tank reacor:

1. Exothermal CSTR
2. Isothermal CSTR

1. Exothermal CSTR:

The exothermic continuous stirred tank reactor (CSTR) is a classical study case of nonlinear systems. Indeed the dynamical behaviour exhibits complex features, such as multiple equilibrium points. Up to now no precise physical interpretation of the complex behaviour of the exothermic reactor has been found [6].

Power-shaping control [7] has been developed in the past years as an extension of energy-balancing passivity based control [8][9]. In energy-balancing passivity based control, the controller reshapes the energy function of the system so that it has a minimum at the desired equilibrium point. The controller provides the system a finite amount of energy so as to drive the system to the desired state. This concept has been applied to electro-mechanical systems [10][11][12] and also to thermodynamic systems where the storage function is the entropy instead of the energy [13].

2. **Isothermal CSTR:**

Isothermal CSTR is a type of CSTR which is operating at a constant temperature. The volume is also assumed to be constant. In this thesis work, we are taking the model of Isothermal CSTR.

2.4.1 Mathematical model of isothermal CSTR:

The reaction scheme consists of the following irreversible reactions. The Continuous Stirred Tank Reactor with single input and single output is shown in Fig. 2.1. Here isothermal series-parallel reaction (Van de-Vusse reaction) is considered to study the steady state and dynamic behaviour of CSTR.

The two reactions are $A \xrightarrow{k_1} B \xrightarrow{k_2} C$ and $2A \xrightarrow{k_3} D$

A – Cyclopentaddiene, B – Cyclopentenol, C – Cyclopentanediol, D – Dicyclopendtadiene

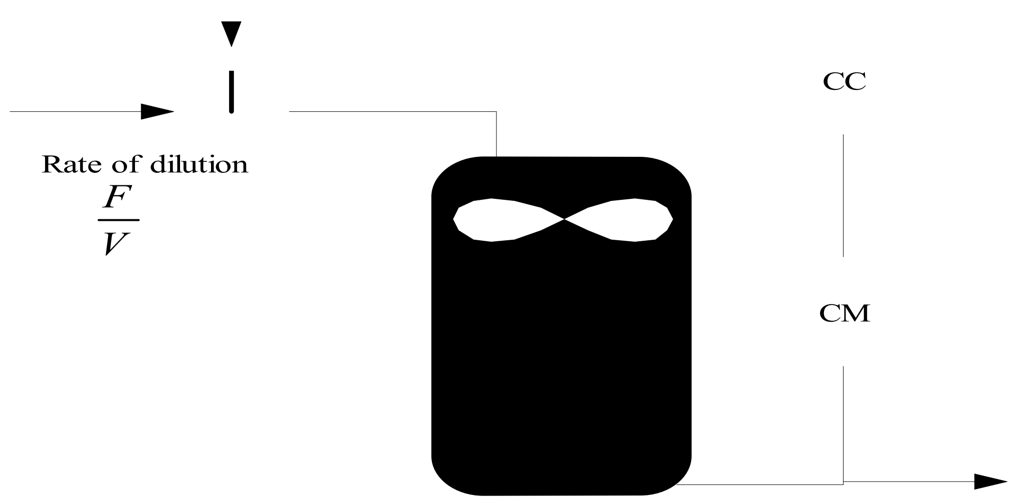


Figure 2.1: Isothermal CSTR.

For the above reaction the values of rate constant are

$k_1 = 50h^{-1} = 0.83\min^{-1}$

$k_2 = 100h^{-1} = 1.66\min^{-1}$

$k_3 = 10moll^{-1}h^{-1} = 0.166moll^{-1}\min^{-1}$

Steady state feed concentration is $C_{Afs} = 10gmoll^{-1}$

Overall material balance is given as $\frac{d(V\rho)}{dt} = F_i\rho - F\rho$ (1)

So, $F = F_i$ (2)

Where ρ = liquid-phase density,

V = volume,

F = Volumetric flow rate.

Component material balance can be shown as

$$\frac{d(VC_A)}{dt} = F(C_{Af} - C_A) - Vk_1C_A - Vk_3C_A^2 \quad (3)$$

Simplifying eq(3) we obtain eq(4)

$$\frac{dC_A}{dt} = \frac{F}{V}(C_{Af} - C_A) - k_1C_A - k_3C_A^2 \quad (4)$$

$$\frac{dC_B}{dt} = -\frac{F}{V}C_B + k_1C_A - k_2C_B \quad (5)$$

Where C_A ,C_B are the concentration of A,B respectively and k_1,k_2,k_3 are the reaction rate constant.

$$\frac{dC_C}{dt} = -\frac{F}{V}C_C + k_2C_B \quad (6)$$

$$\frac{dC_D}{dt} = -\frac{F}{V}C_D + \frac{1}{2}k_3C_A^2 \quad (7)$$

These modeling equations assume a constant volume. The equations for C_C and C_D are neglected because C_B is not dependent on them.

The molar rate of formation for each component (per unit volume) is

$$r_A = -k_1 C_A - k_3 C_A^2 \qquad (8)$$

$$r_B = k_1 C_A - k_2 C_B \qquad (9)$$

$$r_C = k_2 C_B \qquad (10)$$

$$r_D = \frac{1}{2} k_3 C_A^2 \qquad (11)$$

Solving eq(4) and eq(5)

$$-k_3 C_{As}^2 + \left(-k_1 - \frac{F_s}{V}\right) C_{As} + \frac{F_s}{V} C_{Afs} = 0 \qquad (12)$$

Steady state concentration of A and B is defined as

$$C_{As} = \frac{-\left(k_1 + \frac{F_s}{V}\right)}{2k_3} + \frac{\sqrt{\left(k_1 + \frac{F_s}{V}\right)^2 + 4k_3 \frac{F_s}{V} C_{Afs}}}{2k_3} \qquad (13)$$

$$C_{Bs} = \frac{k_1 C_{As}}{\frac{F_s}{V} + k_2} \qquad (14)$$

The linear state space model is represented as

$$x = Ax + Bu$$

$$y = Cx + Du$$

The state variable is represented as $x = \begin{bmatrix} C_A & -C_{As} \\ C_B & -C_{Bs} \end{bmatrix}$

The output variable is represented as $y = \begin{bmatrix} C_A & -C_{As} \\ C_B & -C_{Bs} \end{bmatrix}$

The input variable is represented as $u = \begin{bmatrix} \frac{F}{V} & \frac{-F_s}{V} \end{bmatrix}$

Two dynamic functional equation is represented as

$$\frac{dC_A}{dt} = f_1\left(C_A, C_B, \frac{F}{V}\right) = \frac{F}{V}\left(C_{Af} - C_A\right) - k_1 C_A - k_3 C_A^2$$

$$\frac{dC_B}{dt} = f_2\left(C_A, C_B, \frac{F}{V}\right) = -\frac{F}{V}C_B + k_1 C_A - k_2 C_B$$

The elements of state space A matrix is found by $A_{ij} = \left.\frac{\partial f_i}{\partial x_j}\right|_{x_s, u_s}$

The elements of state space B matrix is found by $B_{ij} = \left.\frac{\partial f_i}{\partial u_j}\right|_{x_s, u_s}$

The state space model is represented as

$$A = \begin{bmatrix} -\frac{F_s}{V} - k_1 - 2k_3 C_{As} & 0 \\ k_1 & \frac{F_s}{V} - k_2 \end{bmatrix}$$

$$B = \begin{bmatrix} C_{Afs} - C_{As} & \frac{F_s}{V} \\ -C_{Bs} & 0 \end{bmatrix}$$

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

$$D = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

Based on steady state operating point $C_{As} = 3 gmoll^{-1}$, $C_{Bs} = 1.117 gmoll^{-1}$,
$\frac{F_s}{V} = 0.5714 \min^{-1}$

$$A = \begin{bmatrix} -2.4 & 0 \\ 0.83 & -2.23 \end{bmatrix}$$

$$B = \begin{bmatrix} 7 & 0.57 \\ -1.117 & 0 \end{bmatrix}$$

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

$$D = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

Converting the state space model to transfer function

$$G(s) = C\left(sI - A\right)^{-1} B$$

The manipulated input-output process transfer function for the reactor is

$$g_p(s) = \frac{-1.117s + 3.1472}{s^2 + 4.6429s + 5.3821} \qquad (15)$$

With delay, transfer function is

$$g_p(s) = \frac{-1.117s + 3.1472e^{-0.5s}}{s^2 + 4.6429s + 5.3821} \qquad (16)$$

and the disturbance input-output transfer function is

$$g_d(s) = \frac{0.4762}{s^2 + 4.6429s + 5.3821} \qquad (17)$$

The transfer function poles (-2.23 and -2.4) are equal to the eigen values of the A matrix. Also, the positive zero(1/0.3549) in process transfer function yields the inverse response shown in figure.

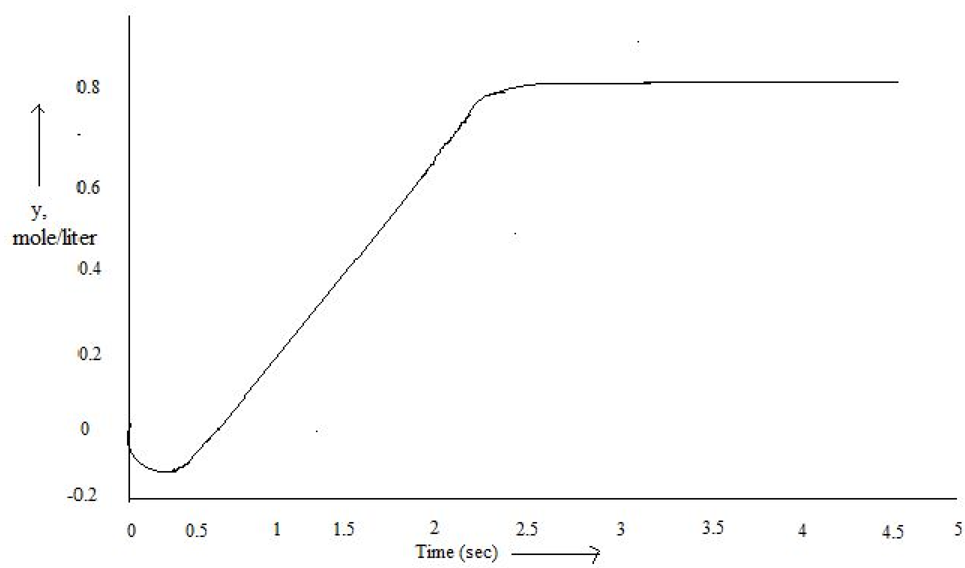


Figure 2.2: Response of the concentration as a function of time

We see that it is straightforward to convert state space models to transfer function models [14].

2.4.2 Block diagram of CSTR:

Controller: The controller is the unit designed to create a stable closed-loop system and also achieve some pre-specified dynamic and static process performance requirements. The input to the controller unit is usually an error signal based on the difference between a desired set point and the actual measured output.

Actuation: The actuator is a unit that supplies power input to the process. The actuator can be considered to act through amplification.

Process: This is the actual system for which some specific physical variables are to be regulated. If we talk about the examples of process industry, these are kilns, furnaces and distillation towers.

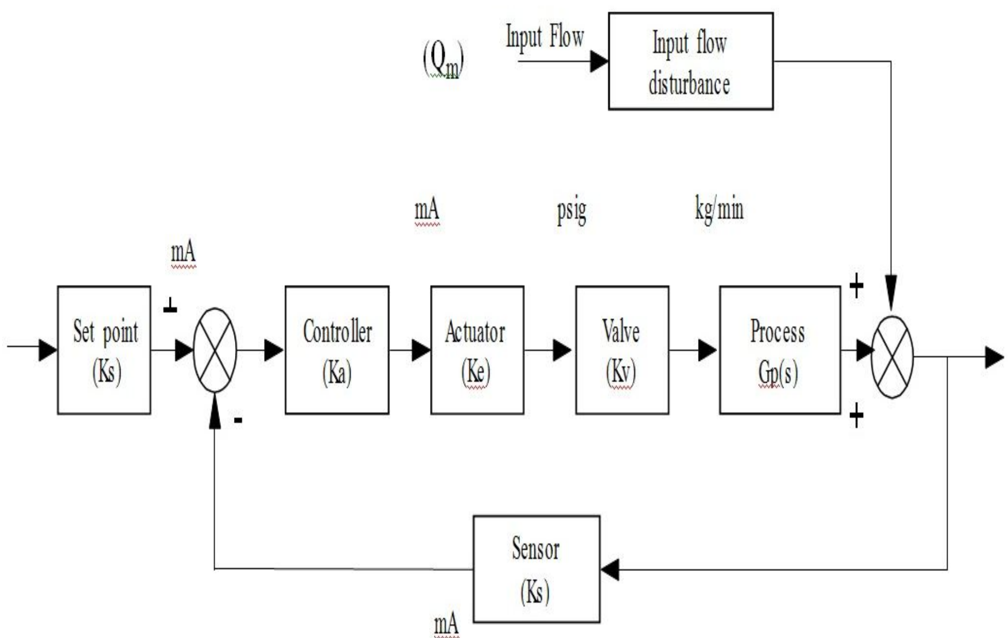


Figure 2.3: Block diagram based feedback control approach for concentration control of isothermal CSTR

Measurement: The common adage is that *without measurement there will be no control*. Typically, the measurement process incorporates a transducer and associated signal processing components. The transducer will comprise a sensor to detect a specific physical property (such as temperature) and will output a representation of the property in a different physical form (such as voltage). It is quite possible that the measured output will be a noisy signal and that some of that noise will still manage to pass through the signal-conditioning component of the measurement device into the control loop [15].

Flow disturbances: Disturbances are variables that fluctuate and cause the process output to move from the desired operating value (set point). A disturbance could be a change in flow, temperature of the surroundings, pressure etc. Disturbance variables can normally be further classified in terms of measured or unmeasured signals.

2.4.3 Application of CSTR:

- A continuously stirred tank bioreactor (CSTR) was used to optimize feasible and reliable bioprocess system in order to treat hydrocarbon-rich industrial wastewaters.
- The continuous-flow, well-stirred tank reactor or *CSTR* finds wide application in the chemical industry from pilot plant to full-scale production operation.

Chapter 3

Literature Review

Regalado-Méndez Alejandro, Cid-Rodríguez Ma. Del Rosario P, And Báez-González Juan G. (2010) presented Problem Based Learning (PBL) in their research paper and described that how to analysis of continuous stirred tank chemical reactors with a process control. [3]

Sanju Nanda (2008) designed fundamentals of Reactors for Chemical Reactions and told that Reactors, in general, may be classified based on the number of phase coexistence into such categories Homogeneous reactor and Heterogeneous reactor. One phase such as gas or liquid exists in the reactors i.e. Homogeneous reactor and Two distinct phases of reactants (or catalyst) coexist i.e. Heterogeneous reactor. [4]

Schmidt, Lanny D. (1998) has given a complete overview of The Engineering of Chemical Reactions, how to design of a chemical reactor deals with multiple aspects of chemical engineering and told that chemical engineers design reactors to maximize net present value for the given reaction. [5]

B. Wayne Bequette (2010) introduced Process Control Modeling, Design, and Simulation. He developed the mathematical modeling of continuous stirred tank reactor and described, how to design the reactor and how to simulink the model of process control. He also described the transient response of PID controllers and how to tune the PID controller by Ziegler-Nichols method. [14]

Michael A. Johnson (1948) proposed New Identification and Design Methods of PID Controller. In his research paper, He explained about some units such as controller, actuator, process, measurement units. [15]

Dale e. Seborg, Thomas F. Edgar, Duncan A. Mellichamp (2004) presented the research paper on Process Dynamics and Control. In their research paper, they explained about process control system and it's designing. Control system design begins with a formulation of control objective which is based on efficient performance, management and financial aspects of the plant, process knowledge and operational requirement of the plant and also explained some consideration such as Stable plant operation, Economic plant operation, Safety considerations, Environmental regulations, Product specification and production rate while design controller for any application. [16]

B. C. Kuo (1995) introduced Automatic Control System. He explained all the types of controller and proportional, integral, derivative control action. He also explained the transient response of controller. [17]

Amir Hossein Fathi, Hamid Khaloozadeh, Mohammad Ali Nekoui, Reza Shisheie (2012) presented research paper on the optimizing the parameters of PID controller using PSO and GA. In their research paper, they described about the PID controller and how to tune the PID controller using PSO. They also discussed some of various error performance indices. [18]

Kiam heong Ang et.al (2005) has given a complete overview of modern tuning methods of PID controller, different patents in PID controllers, commercial hardware modules and software packages of PID controller available in market. This paper also reviews the contemporary intelligent PID controllers and reviews the future PID controller like plug and play PID controller [21].

Nina F. Thornhill et.al (2008) presents the simulation of CSTR. In this article, volumetric and heat balance equations are presented along with algebraic equations derived from experimental data for calibration of sensors and actuators and unknown quantities such heat transfer through the heating coils. Many of these relationships have nonlinearities, and hard constraints such as the tank being full are also captured. A valuable feature is that the model uses measured, not simulated, noise and disturbances and therefore provides a realistic platform for data-driven identification and fault detection [22].

R Suja Mani Malar et.al (2009) has proposed the use of Artificial intelligence technique to model and control the CSTR [23].

Sufian Ashraf Mazhari et.al,(2008) has proposed a fuzzy PD+I controller for PUMA 560 robot and used different swarm intelligence and evolutionary techniques to tune the fuzzy PD+I controller. This paper also gives a comparative study of different swarm intelligence and evolutionary algorithm based tuning methods. [24]

L. A. Zadeh (1984) presented theory of commonsense knowledge. He told that knowledge representation is one of the most basic and actively researched areas of artificial intelligence. The conventional approaches to knowledge representation lack the means for representing the meaning of fuzzy concepts. [28]

R. C. Moore (1982) has proposed the role of logic in knowledge representation and commonsense reasoning. He gave some of the basic ideas underlying fuzzy logic and described their application to the problem of knowledge representation in an environment of uncertainty and imprecision. [25]

Mamdani, E.H. and Gaines, B.R. (1981), in their research paper, presented Fuzzy Reasoning and its Applications. They developed rule base of the fuzzy logic controller and told that, in fuzzy inference system, there are two inputs as error and change in error and one output. [37]

L. S. Gottfredson (1997), in his research, He told about intelligence that it is a very general mental capability that, among other things, involves the ability to reason, plan, solve problems, think abstractly, comprehend complex ideas, learn quickly and learn from experience, in his research.[41]

Dennis J. Reimer (2006) described about human intelligence that it is the mental quality that consists of the abilities to learn from experience, adapt to new situations and also described many types of approach methods for human intelligence such as direct approach, incentive approach, emotional approaches etc. [45]

Omkar Pathak (2011), in his research, he explained about artificial intelligence that artificial intelligence is a human endeavour to create a non-organic machine-based entity, that has all the above abilities of natural organic intelligence and also explained about applications of artificial intelligence such as computer vision, game playing, weather forecasting, swarm intelligence, in expert system and in neural network etc. [46].

C.C. Lee (1990), presented research paper on fuzzy logic in control systems. He described that fuzzy logic is a powerful problem-solving methodology with a myriad of applications in embedded control and information processing. The fuzzy logic controller provides an algorithm, which converts the expert knowledge into an automatic control strategy. [61]

Kennedy, J., Eberhart, R. C., and Shi, Y. (2001) described about Swarm Intelligence that it is an artificial intelligence technique based around on the study of collective behavior in decentralized, self-organized systems. Swarm intelligence systems are typically made up of a population of simple agents interacting locally with one another and also with their environment. [62]

J.Kennedy and R.Eberhart (1995), introduced Particle Swarm Optimization. PSO is a computational algorithm technique based on swarm intelligence. It has simplicity, easy implementation, it has been found to continuous in solving of continuous nonlinear optimization problem. This method is motivated by the observation of social interaction and animal behaviors such as fish schooling and bird flocking [68].

T. Bartz–Beielstein K.E. Parsopoulos and M.N. Vrahatis (2004), in their research paper, presented, Analysis of Particle Swarm Optimization Using Computational Statistics and described that in a PSO system, particles fly around in a multi-dimensional search space adjusting its position according to its own experience and the experience of its neighbouring particle. [71]

Eberhart, R. C., and Hu, X. (1999) did human tremor analysis using particle swarm optimization. He told that particle swarm optimization has been applied to the analysis of human tremor. The diagnosis of human tremor, including Parkinson's disease and essential tremor, is a very challenging area [76].

Yoshida, H., Kawata, K., Fukuyama, Y., and Nakanishi, Y. (1999) described an application of particle swarm optimization for reactive power and voltage control considering voltage stability. Particle swarm optimization was used to determine a control strategy with continuous and discrete control variables, resulting in a sort of hybrid binary and real-valued version of the algorithm. Voltage stability in the system was achieved using a continuation power flow technique. [78]

Russell C. Eberhart (2001) described that particle swarm optimization is attractive is that there are very few parameters to adjust. One version, with very slight variations (or none at all) works well in a wide variety of applications and also described about development and resources of particle swarm optimization. [80]

Karl O. Jones (2005) told about algorithm of particle swarm intelligence that in PSO algorithm, the system is initialized with a population of random solutions, which are called particles, and each potential solution is also assigned a randomized velocity. [88]

Chapter- 4

Control System & Controller Design

4.1 Introduction:

Control system design begins with a formulation of control objective which is based on efficient performance, management and financial aspects of the plant, process knowledge and operational requirement of the plant. But apart from that there are number of general considerations while designing the controller for any application. They are listed as below [16]:

(1) Stable plant operation
(2) Economic plant operation
(3) Safety considerations
(4) Environmental regulations
(5) Product specification and production rate.

(1) Stable Plant Operation: The control system should facilitate smooth, stable plant operation without excessive oscillation in key process variables. Thus, it is desirable to have smooth, rapid set-point changes and rapid recovery from plant disturbances such as changes in feed composition.

(2) Economic Plant Operation: It is an economic reality that the plant operation over long periods of time must be profitable. Thus, the control objectives must be consistent with the economic objectives.

(3) Safety considerations: It is imperative that industrial plants operate safely so as to promote the well-being of people and equipment within the plant and in the nearby communities. Thus, plant safety is always the most important control objective.

(4) Environmental Regulations: Industrial plants must comply with environmental regulations concerning the discharge of gases, liquids, and solids beyond the plant boundaries.

(5) Product Specifications and Production Rate: In order to be profitable, a plant must make products that meet specifications concerning product quality and production rate.

4.2 General approaches of control system design:

There are two general approaches to control system design.

1. **Traditional Approach:** Control hardware is selected by the knowledge of process and experience and then tuning of the controller done.
2. **Model Based Approach:** A dynamic model is prepared and computer simulation is done before going for control hardware.

4.2.1 Steps in control system design:

The basic steps in control system design are as follows [16]:

1. Select the controlled, manipulated and measured variable.
2. Choose the control strategy and control structure.
3. Specify the controller settings.

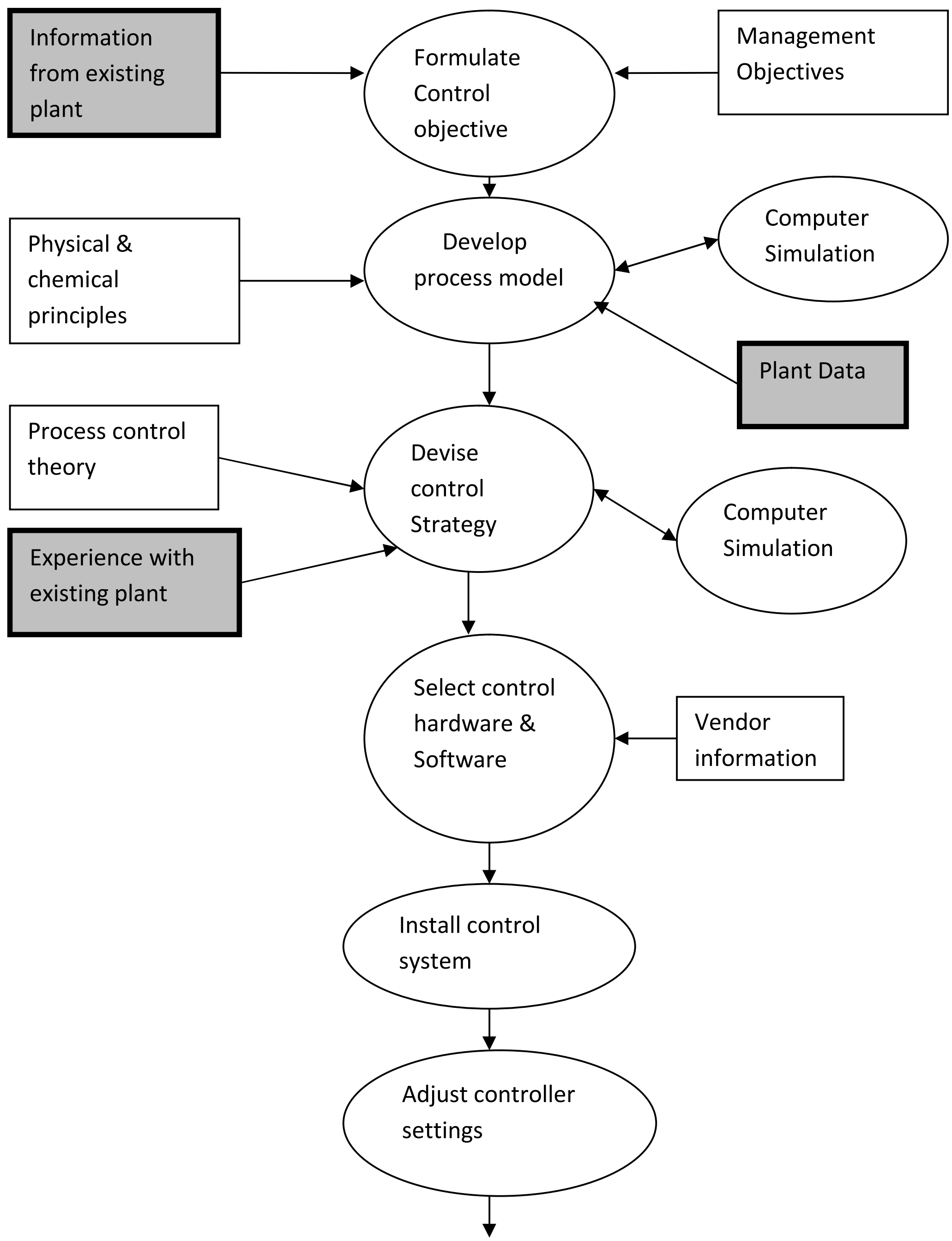
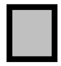


Figure 4.1: Major steps in control system development

Engineering Activity → □

Information base → ○

Data (if available) → 

As can be seen from the typical industrial control loop structure given, even simple process loops comprise more than four engineering components. The main components can be grouped according to the following loop operations [15]:

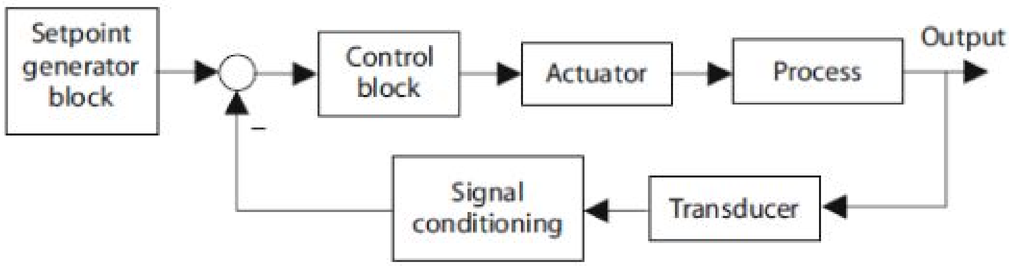


Figure 4.2: Components of industrial control loop

Process: This is the actual system for which some specific physical variables are to be controlled or regulated. Typical process industry examples are boilers, kilns, furnaces and distillation towers.

Actuation: The actuator is a process unit that supplies material or power input to the process. The actuator can be considered to act through amplification. For example, the control signal could be a small movement on a valve stems controlling a large flow of natural gas into a gas-fired industrial boiler.

Measurement: The common adage is that *without measurement there will be no control*. Typically, the measurement process incorporates a transducer and associated signal processing components. The transducer will comprise a sensor to detect a specific physical property (such as temperature) and will output a representation of the property in a different physical form (such as voltage). It is quite possible that the measured output will be a noisy signal and that some of that noise will still manage to pass through the signal-conditioning component of the measurement device into the control loop.

Controller: The controller is the unit designed to create a stable closed-loop system and also achieve some pre-specified dynamic and static process performance requirements. The input to the controller unit is usually an error signal based on the difference between a desired setpoint or reference signal and the actual measured output.

Communications: The above units and components in the control loop are all linked together. In small local loops, the control system is usually hardwired, but in spatially distributed processes with distant operational control rooms, computer communication components (networks, transmitters and receivers) will possibly be needed. This aspect of control engineering is not often discussed; however, the presence of communication delays in the loop may be an important obstacle to good control system performance. To specify the controller, the performance objectives of the loop must be considered carefully, but in many situations it is after the loop has been commissioned and in use for a longer period of production activity that new and unforeseen process problems are identified. Consequently industrial control engineering often has two stages of activity: (i) control design and commissioning and (ii) post-commissioning control redesign.

4.3 Types of controller:

There are different types of controller in an industry. Some of them are listed below [17]:

1. Proportional control
2. Integral control
3. Derivative control
4. Proportional-integral controller (PI controller)
5. Proportional-derivative controller (PD controller)
6. Proportional-integral-derivative Controller (PID controller)

4.3.1 Proportional control:

Proportional control is defined as the control action that occurs in direct proportion with the system error. The output of a proportional controller varies proportionally to the system error:

$$u(t) = K_p e(t) + b$$

where u(t) is the controller output, e(t) is the error, K_p is proportional gain, b is the controller bias. Proportional control action responds to only the present error. For a small value of proportional gain, a large error yields a small corrective control action. Conversely, a large proportional gain will result in a small error and hence a large control signal. The controller bias is necessary in order to ensure that a minimum control action is always present in the control loop.

Proportional band:

The gain of a proportional controller is usually described in terms of its proportional band (*PB*). It is the percentage of full controller range by which the measured value must change in order to cause the correcting device to change by 100%.

It is defined as, $PB = \frac{1}{K_c}$

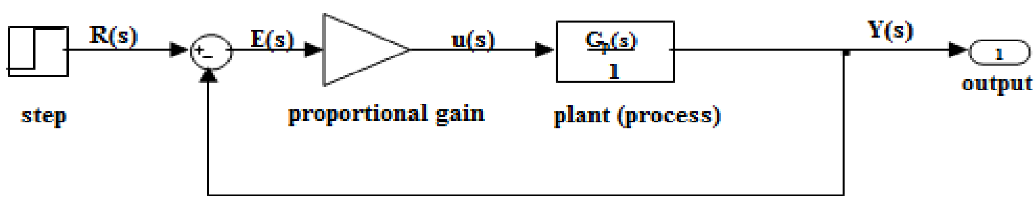


Figure4.3: Proportional controller block diagram

4.3.2 Integral control:

Integral control eliminates the steady state error but introduces phase lag and has a very sluggish response and seldom used alone. Its primary effect on a process control system is to permanently attempt to gradually eliminate the error. The action of the integral controller is based on the principle that the control action should exist as long as the error is different from zero, and it has the tendency to gradually reduce the error to zero. The integrator control signal u(t) is proportional to the duration of the error and is given by:

$$u(t) = K_i\left(\int_0^t e(t)dt\right)$$

Where $k_i = k_c/Ti$

Ti= Reset time

1/Ti = Reset rate

The smaller the integral time constant, the more often the proportional control action is repeated, therefore resulting in greater integral contribution toward the control signal. For a large integral time constant, the integral action is reduced. Integral control can be seen as continuously looking at the total past history of the error by continuously integrating the area under the error curve and reducing any offset. There will be the greater the error signal and the larger the correcting action from the integral controller.

4.3.2.1 Undesirable effects of integral control:

Although integral control is very useful for removing steady-state errors it is also responsible for sometimes introducing undesirable effects into the control loop in the form of increase settling time, reduced stability and integral windup.

Increased settling time: An increase of the closed-loop system settling time is usually caused by the increased oscillations as a consequence of the present integral action.

Reduced stability: The presence of the integral action may lead to increased oscillations within the control loop. These oscillations generally have a tendency to move the system towards the boundary of instability. In some cases these oscillations will result in the loop becoming unstable.

4.3.3 Derivative control:

It has a fast response and it anticipates the future behaviour of the error. It is not used for turbulent process variable like flow. It is used in process involving temperature. With derivative action, the controller output is proportional to the rate of change of the measurement or error. Some manufacturers use the term rate or pre-act instead of derivative. Derivative, rate and pre-act are the same thing. The controller output is calculated by the rate of change of the error with time.

$$u(t) = K_d \left(\frac{de(t)}{dt} \right)$$

Derivative control action tends to improve the dynamic response of the controlled variable by decreasing the process settling time, the time it takes the process to reach the steady state. But if the process measurement is noisy that is if I contains high frequency, random fluctuations, then the derivative of the measured variable will change widely and derivative action will amplify noise unless the measurement is filtered. So the derivative action is seldom used for flow control loops because the flow control responds quickly and the flow measurement is noisy.

4.3.4 PI controller:

It eliminates the steady state error and the speed of response of PI controller is less than P controller but more than I controller.

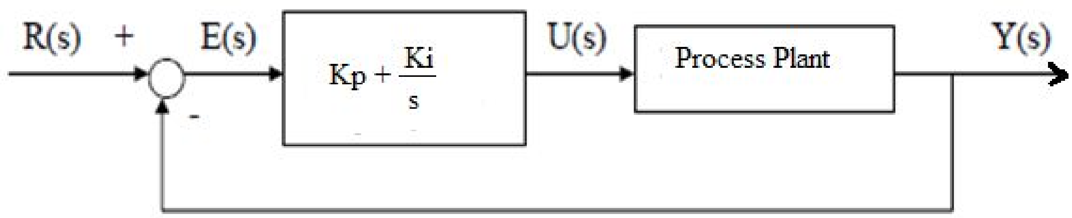


Figure 4.4: PI controller block diagram

In this context, the controller outputs in time domain is given by:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt$$

No offset, better dynamic response than I alone, Possibilities exists for instability due to lag introduced, steady state error reduces tremendously for the same type of inputs.

4.3.5 PD controller:

The phase lead of PD controller is less than D controller and decreases the speed of response of D controller. It is only used in batch process control. The ideal PD control algorithm is physically unrealizable because it can't be implemented exactly using either analog or digital components. It is stable, gives less offset than P alone, more rapid response, reduce lag, reduces the settling time and peak overshoot.

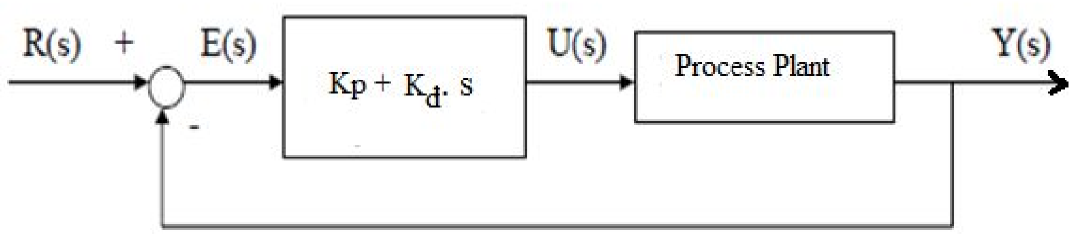


Figure 4.5: PD controller block diagram

In this context, the controller outputs in time domain is given by: $u(t) = K_p e(t) + K_d \frac{de(t)}{dt}$

4.3.6 PID controller:

The Proportional-Integral-Derivative (PID) controllers have been the most commonly used controller in process industries for over 50 years even though significant development have been made in advanced control theory. According to a survey conducted by Japan Electric Measuring Instrument Manufacturers Association in 1989, 90 % of the control loops in industries are of the PID type. The proportional action adjusts controller output according to the size of the error, the integral action eliminates the steady state offset and the future is anticipated via derivative action. These useful functions are sufficient for a large number of process applications and the transparency of the features lead to wide acceptance by the users. Strength of the PID controller is that it also deals with important practical issues such as actuator saturation and integrator windup. PID controllers perform well for a wide class of processes and they give robust performance for a wide range of operating conditions and are easy to implement using analog or digital hardware. Moreover, due to process uncertainties, a more sophisticated control scheme is not necessarily more efficient than a well tuned PID controller. A large industrial process may have hundreds of PID controllers. Proper tuning of the controllers is crucial for achieving the desired response characteristics. They have to be tuned individually to match the process dynamics in order to provide good and robust control performance. The tuning procedure, if done manually, is very tedious and time consuming; the resultant system performance mainly depends on the experience and the process knowledge of the engineers. It is recognized that in practice, many industrial control loops are poorly tuned. However with the advent of the auto-tuning of PID controller concept, this problem has been solved to a considerable extent. Automatic tuning techniques thus draw more and more attention of the researchers and practicing engineers. By automatic tuning, we mean a method which enables the controller to be tuned automatically on demand from an operator or an external signal. Typically, the user will either push a button or send a command to the controller.

PID controllers are simple and easy to implement and they are widely used in industry to solve different control problem. The standard PID controller structure is as shown in figure 4.6 and transfer function is described by following equation in the continuous s-domain (Laplace operator).

$$G_{PID}(s) = P + I + D = \frac{U(s)}{E(s)} = K_P + \frac{K_i}{s} + K_d\, s$$

Or

$$G_{PID}(s) = K_P + \frac{K_P}{T_i\, s} + K_P T_d\, s$$

Where $U(s)$ is control signal and $E(s)$ is control error in s-domain, respectively; K_P is the proportional gain, $K_i = \frac{K_P}{T_i}$ is integration gain, and K_d=$K_P T_d$ is the derivative gain. T_i is the integration time coefficient and T_d is referred to as the derivation time coefficient. In this context, the controller outputs in time domain is given by:

$$u(t) = K_p e(t) + K_d \frac{de(t)}{dt} + K_i \int_0^t e(t)dt$$

Where u(t) and e(t) are the control output and error signals in time domain, respectively.

The action of the proportional gain, K_P reduces the rise time of the system response and steady state error but it never eliminates error. On the other hand the integral gain, K_i reduces the steady sate error and derivate gain, K_d improves stability of system and reduces the overshoot of system as well as improving transient response [18].

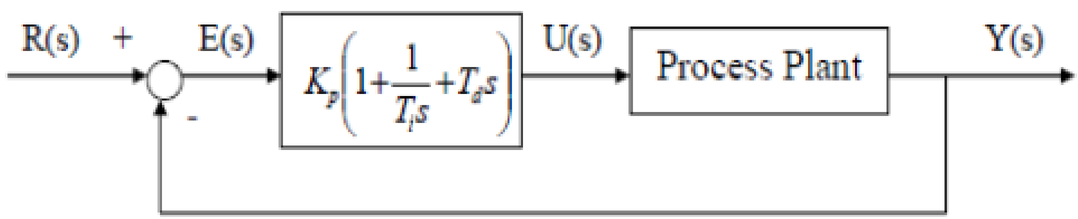


Figure 4.6: PID controller block diagram

4.4 Characteristics of P, I, and D controllers:

A proportional controller (K_P) will have the effect of reducing the rise time and will reduce, but never eliminate, the steady-state error. An integral control (K_i) will have the effect of eliminating the steady-state error, but it may make the transient response worse. A derivative control (K_d) will have the effect of increasing the stability of the system, reducing the overshoot, and improving the transient response [17]. Effects of each of controllers K_P, K_d, and K_i on a closed-loop system are shown in the table 4.1.

Table 4.1: Controller responses

Controller Response	Rise Time	Overshoot	Settling Time	Steady State Error
P	Decrease	Increase	Small Change	Decrease
I	Decrease	Increase	Increase	Eliminate
D	Small Change	Decrease	Decrease	Small Change

Note that these correlations may not be exactly accurate, because K_P, K_i, and K_d are dependent of each other. In fact, changing one of these variables can change the effect of the other two. For this reason, the table should only be used as a reference when you are determining the values for K_P, K_i and K_d.

4.5 Limitations of PID control:

While PID controllers are applicable to many control problems, and often perform satisfactorily without any improvements or even tuning, they can perform poorly in some applications, and do not in general provide optimal control. The fundamental difficulty with PID control is that it is a feedback system, with constant parameters, and no direct knowledge of the process and thus overall performance is reactive and a compromise – while PID control is the best controller with no model of the process [19], better performance can be obtained by incorporating a model of the process. The most significant improvement is to incorporate feed-forward control with knowledge about the system, and using the PID only to control error. Alternatively, PIDs can be modified in more minor ways, such as by changing the parameters (either gain scheduling in different use cases or adaptively modifying them based on performance), improving measurement (higher sampling rate, precision, and accuracy, and low-pass filtering if necessary), or cascading multiple PID controllers. PID controllers, when used alone, can give poor performance when the PID loop gains must be reduced so that the control system does not overshoot, oscillate or hunt about the control setpoint value. They also have difficulties in the presence of non-linearities, may trade-off regulation versus response time, do not react to changing process behavior (say, the process changes after it has warmed up), and have lag in responding to large disturbances.

4.6 Tuning of PID controller:

Tuning of PID controller is to choose the numerical values for the PID coefficients. Many industrial process companies have in-house manuals that provide guidelines for the tuning of PID controllers for particular process plant units. Thus for simple processes it is often possible to provide rules and empirical formulae for the PID controller tuning procedure. Some of these manuals base their procedures on the *pro forma* routines of the famous Ziegler–Nichols methods and their numerous extensions of the associated rules (Ziegler and Nichols, 1942). Ziegler–Nichols method uses an on-line process experiment followed by the use of rules to calculate the numerical values of the PID coefficients. In the 1980s, when analogue control was being replaced by digital processing hardware, industrial control companies took the opportunity to develop new PID controller methods for use with the new ranges of controller technology appearing. Consequently, the Ziegler–Nichols methods became the focus of research and have since, become better understood. New versions of the Ziegler–Nichols procedures were introduced, notably the Åström and Hägglund relay experiment (Åström and Hägglund, 1985).In many applications, the implicit underdamped closed-loop performance inherent in the original Ziegler–Nichols design rules was found to be unacceptable. The result was an extensive development of the rule-base for PID controller tuning. O'Dwyer (1998a,b) has published summaries of a large class of the available results. For the tuning of PID controller, here we are using Ziegler-Nichols Tuning method [20][21].

4.6.1 Ziegler-Nichols tuning:

In 1942, Ziegler and Nichols, both of them are the employees of Taylor Instruments. They described simple mathematical procedures in the form of the first and second methods, for tuning PID controllers. These procedures are now accepted as standard in control systems practice. Both techniques make a priori assumptions on the system model, but do not require that these models be specifically known. Ziegler-Nichols formulae for specifying the controllers are based on plant step responses.

This technique is designed to result in a closed loop system with 25% overshoot. This is rarely achieved as Ziegler and Nichols determined the adjustments based on a specific plant model.

The steps for tuning a PID controller via the Ziegler and Nichols method are as follows:

Using only proportional feedback control:

- Reduce the integrator and derivative gains to 0.
- Increase K_p from 0 to some critical value $K_p=K_{cr}$ at which sustained oscillations occur. If it does not occur then another method has to be applied.
- Note the value K_{cr} and the corresponding period of sustained oscillation, P_{cr}.

The controller gains are now specified as follows [20]:

Table 4.2: Ziegler Nichols method

PID Type	K_p	T_i	T_d
P	$0.5K_{cr}$	∞	0
PI	$0.45K_{cr}$	$\frac{P_{cr}}{1.2}$	0
PID	$0.6K_{cr}$	$\frac{P_{cr}}{2}$	$\frac{P_{cr}}{8}$

In the above table 4.2, proportional gain, integral time and derivative time for different controller types are derived using ultimate gain and ultimate period using Ziegler-Nichols method.

4.7 Transient response of PID controller:

Following figures shows the transient response of PID controllers [14]. The below figure 4.7 shows the unit step response of second order system by varying the values of K_p and keeping the values of K_i and K_d constant.

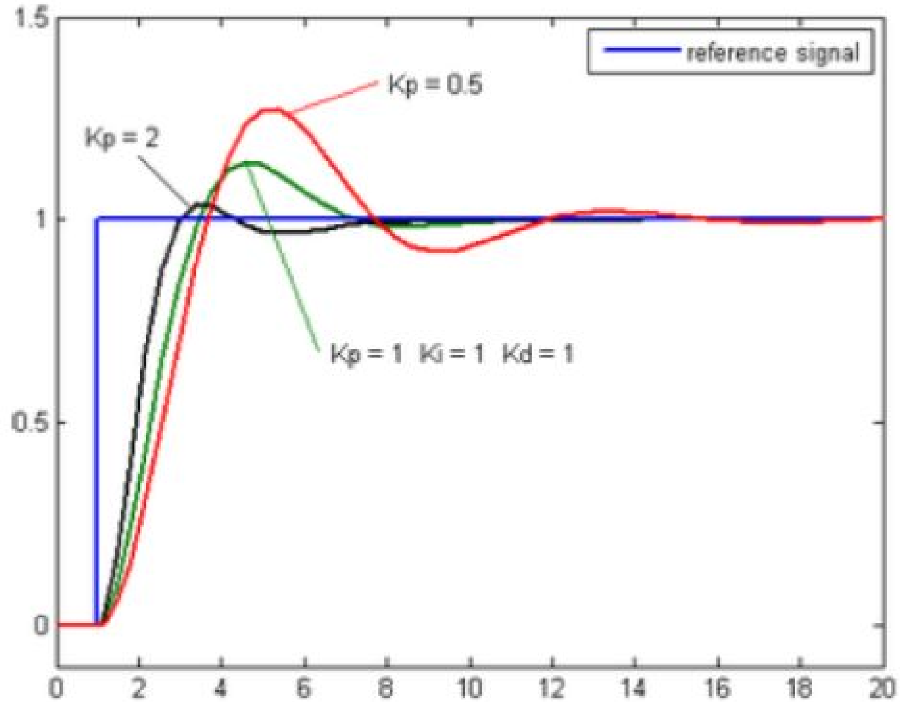


Figure 4.7: Plot of PV vs time, for three values of Kp (Ki and Kd held constant)

figure 4.8 shows the unit step response of second order system by varying the values of K_i and keeping the values of K_p and K_d constant.

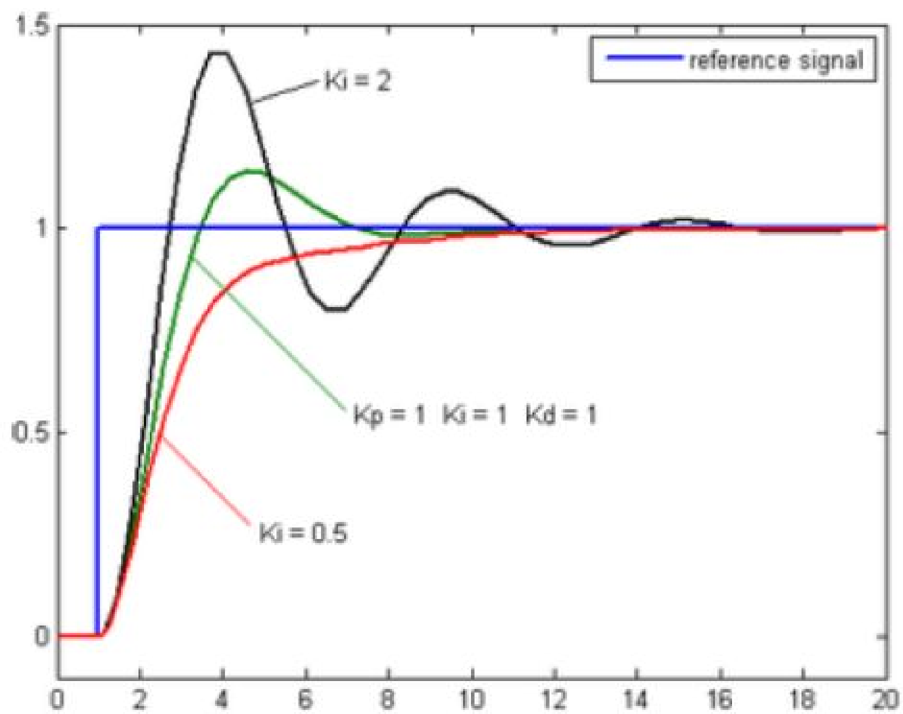


Figure 4.8: Plot of PV vs time, for three values of Ki (Kp and Kd held constant)

figure 4.9 shows the unit step response of second order system by varying the values of K_d and keeping the values of K_p and K_i constant.

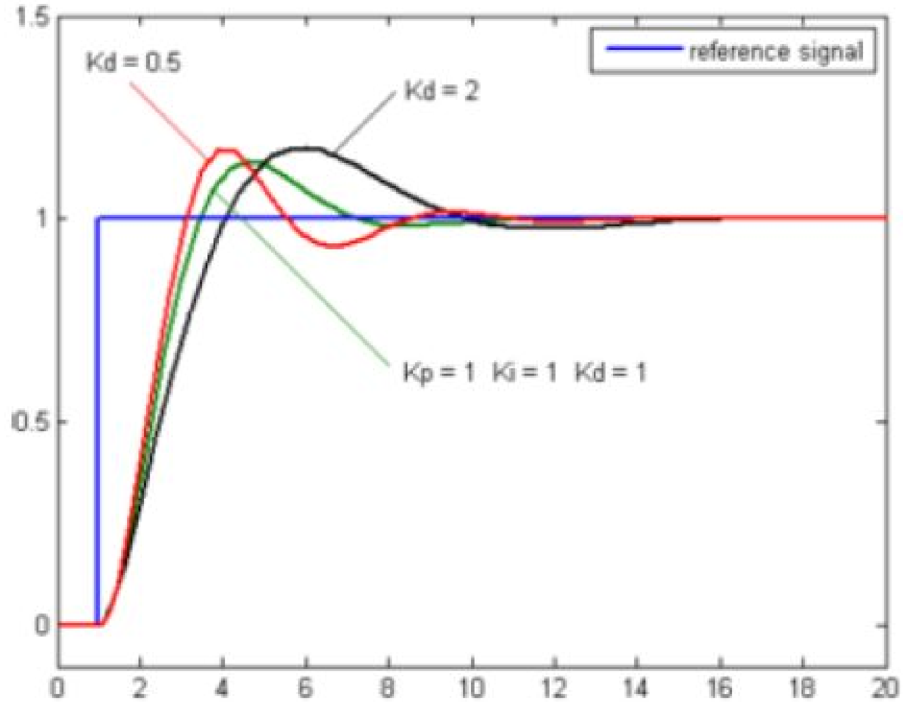


Figure 4.9: Plot of PV vs time, for three values of Kd (Kp and Ki held constant)

4.8 Performance indices:

The optimal control can not be define precisely in general. A solution, which is optimum for one particular application for given set of conditions, may not be optimal for another problem. In such case, defining performance criterion and minimizing the value of such performance index, optimal constants can be obtained as required for economic and practical problem.

One of such index is cost function, which is used by most of the control engineers. Such performance indices help us in assessing the quality of a control system.

Performance index (P.I.):

A performance index is a quantitative measure of the performance of the system. A system is considered an optimal control system when the system parameters are adjusted so that the index reaches an extreme value, commonly a minimum value. Now we will discuss some of various error performance indices which are commonly used in this thesis work [18].

- **ISE (Integral of square error):** A suitable fitness function (Performance index) is the integral of square error.

$$ISE = \int_0^\infty e^2(t)dt$$

- **IAE (Integral of absolute error):** Another fitness function is the integral of the absolute error as follow:

$$IAE = \int_0^\infty |e(t)|\,dt$$

This criterion penalized large errors less heavily and small errors more heavily compared to ISE.

- **ITSE (Integral of time multiplied square error criterion):** It is generally desirable to step response patterns when used as a basis for optimum design. This has a characteristic that in unit step response of the system a large initial error is weighted lightly, while errors occurring late in the transient response are penalized heavily.

$$ITSE = \int_0^\infty te^2(t)dt$$

- **ITAE (Integral of time multiplied by absolute value of error):** It is achieved by adding time weighting to IAE and defined as:

$$ITAE = \int_0^\infty t|e(t)|\,dt$$

This generally leads to systems with reasonable transient characteristics. It is comparable in many respects to ITSE but is not mathematically analytic. This also has a property that initial large errors gets weighted lightly and late occurring error gets penalized heavily.

Chapter 5

Artificial Intelligence & Soft Computing Techniques

5.1 Common sense conventional knowledge:

Knowledge representation is one of the most basic and actively researched areas of **AI** [25], [26], [27]. And yet, there are many important issues underlying knowledge representation which have not been adequately addressed. One such issue is that of the representation of knowledge which is lexically imprecise and/or uncertain.

As a case in point, the conventional knowledge representation techniques do not provide effective tools for representing the meaning of or inferring from the kind of everyday type facts exemplified by the following.

1) Usually it takes about an hour to drive from Berkeley to Stanford in light traffic.

2) Unemployment is not likely to undergo a sharp decline during the next few months.

3) Most experts believe that the likelihood of a severe earthquake in the near future is very low.

The italicized words in these assertions are the labels of fuzzy predicates, fuzzy quantifiers, and fuzzy probabilities. The conventional approaches to knowledge representation lack the means for representing the meaning of fuzzy concepts. As a consequence, the approaches based on first-order logic and classical probability theory do not provide an appropriate conceptual framework for dealing with the representation of common sense knowledge, since such knowledge is by its nature both lexically imprecise and non categorical [26], [28].

The development of fuzzy logic was motivated in large measure by the need for a conceptual framework which can address the issues of uncertainty and lexical imprecision. The principal objective of this chapter is to present a summary of some of the basic ideas underlying fuzzy logic and to describe their application to the problem of knowledge representation in an environment of uncertainty and imprecision. A more detailed discussion of these ideas may be found in Zadeh 1591, [29] and other references. In this chapter, we will discuss about Intelligence and soft computing techniques but here only told about essential of fuzzy logic.

5.1.1 Essentials of fuzzy logic:

The importance of fuzzy logic derives from the fact that most modes of human reasoning-and especially common sense reasoning-are approximate in nature. It is of interest to note that, despite its pervasiveness, approximate reasoning falls outside the purview of classical logic largely because it is a deeply entrenched tradition in logic to be concerned with those and only those modes of reasoning which lend themselves to precise formulation and analysis.

Some of the essential characteristics of fuzzy logic relate to the following.

- *In fuzzy logic, exact reasoning is viewed as a limiting case of approximate reasoning.*
- *In fuzzy logic, everything is a matter of degree. Any logical system can be fuzzified.*
- *In fuzzy logic, knowledge* ***is*** *interpreted as a collection of elastic or, equivalently, fuzzy constraint on a collection of variables.*
- *Inference is viewed as a process of propagation of elastic constraints.*

Fuzzy logic differs from traditional logical systems both in spirit and in detail. Some of the principal differences are summarized in the following **[30].**

Predicates: In bivalent systems, the predicates are crisp, e.g., mortal, even, larger than. In fuzzy logic, the predicates are fuzzy, e.g., tall, ill, soon, swifi, much larger than. It should be noted that most of the predicates in a natural language are fuzzy rather than crisp.

Predicate Modijers: In classical systems, the only widely used predicate modifier is the negation, not. In fuzzy logic, there is a variety of predicate modifiers which act as hedges, e.g., very, more or less, quite, rather, extremely. Such predicate modifiers play an essential role in the generation of the values of a linguistic variable, e.g., very young, not very young, more or less young, etc. **[31].**

Quantifiers: In classical logical systems there are just two quantifiers: universal and existential. Fuzzy logic admits, in addition, a wide variety of fuzzy quantifiers exemplified by few, several, usually, most, almost always, frequently etc. In fuzzy logic, a fuzzy quantifier is interpreted as a fuzzy number or a fuzzy proportion **[32].**

Truth: In bivalent logical systems, truth can have only two values: true or false. In multivalued systems, the truth value of a proposition may be an element of: a) a finite set; b) an interval such as [0, 11; or c) a boolean algebra. In fuzzy logic, the truth value of a proposition may be a fuzzy subset of any partially ordered set, but usually it is assumed to be a fuzzy subset of the interval [0, 1**]** or, more simply, a point in this interval. The so-called linguistic truth values expressed as true, very true, not quite true, etc., are interpreted as labels of fuzzy subsets of the unit interval.

Probabilities: In classical logical systems, probability is numerical or interval-valued. In fuzzy logic, one has the additional option of employing linguistic or, more generally, fuzzy probabilities exemplified by likely, unlikely, very likely, around 0.8, high, etc. **[29].** Such probabilities may be interpreted as fuzzy numbers which may be manipulated through the use of fuzzy arithmetic **[33]**

In addition to fuzzy probabilities, fuzzy logic makes it possible to deal with fuzzy events. An example of a fuzzy event is: tomorrow will be a warm day, where warm is a fuzzy predicate. The probability of a fuzzy event may be a crisp or fuzzy number **[34].**

A concept which plays a central role in fuzzy logic is that of a possibility distribution [**35].** Briefly, if X is a variable taking values in a universe of discourse U, then the possibility distribution of X, π_x**,** is the fuzzy set of all possible values of X. More specifically, let $\boldsymbol{\pi_x}$(u) denote the possibility that X can take the value u, u∈U. Then the membership function of X is numerically equal to the possibility distribution function $\boldsymbol{\pi_x}$(u): U → [0,1**],** which associates with each element u, u∈U the possibility that X may take u as its value. More about possibilities and possibility distributions will be said at a later point in this chapter.

5.1.2 Meaning and knowledge representation:

The conventional knowledge representation techniques based on the use of predicate calculus and related methods are not well-suited for the representation of commonsense knowledge because the predicates in propositions which represent commonsense knowledge do not, in general, have crisp denotations. For example, the proposition Most Frenchmen are not very tall cannot be represented as a well-formed formula in predicate calculus because the sets which constitute the denotations of the predicate tall and the quantifier most in their respective universes of discourse are fuzzy rather than crisp.

More generally, the inapplicability of predicate calculus and related logical systems to the representation of commonsense knowledge reflects the fact that such systems make no provision for dealing with uncertainty. Thus, in predicate logic, for example, a proposition is either true or false and no gradations of truth or membership are allowed. By contrast, in the case of commonsense knowledge, a typical proposition contains a multiplicity of sources of uncertainty. For example, in the case of the proposition If a car which is offered for sale is cheap and old then it probably is not in good shape, there are five sources of uncertainty: (i) the temporal uncertainty associated with the fuzzy predicate old; (ii) the uncertainty associated with the fuzzy predicate cheap; (iii) the uncertainty associated with the fuzzy with the event The car is not in good shape; and (v) the uncertainty associated with the fuzzy characterization of the probability of the event in question as probable.

The approach to the representation of commonsense knowledge which is described in this paper is based on the idea that propositions which characterize commonsense knowledge are, for the most part, dispositions, that is, propositions with implied fuzzy quantifiers. In this sense, the proposition Tall men are not very agile is a disposition which upon restoration is converted into the proposition Most tall men are not very agile. In this proposition, most is an explicit fuzzy quantifier which provides an approximate characterization of the proportion of men who are not very agile among men who are tall [36].

To deal with dispositions in a systematic fashion, we shall employ fuzzy logic -- which is the logic underlying approximate or fuzzy reasoning [37] [38]. Basically, fuzzy logic has two principal components. The first component is, in effect, a translation system for representing the meaning of propositions and other types of semantic entities. We shall employ the suggestive term test-score semantics to refer to this translation system because it involves an aggregation of the test scores of elastic constraints which are induced by the semantic entity whose meaning is represented.

The second component is an inferential system for arriving at an answer to a question which relates to the information which is resident in a knowledge base. In the present paper, the focus of our attention will be the problem of meaning representation in the context of commonsense knowledge, and our discussion of the inferential component will be limited in scope.

5.2 Intelligence:

Intelligence has been defined in many different ways, including the abilities, but not limited to, abstract thought, understanding, self-awareness, communication, reasoning, learning, having emotional knowledge, retaining, planning, and problem solving.

"Intelligence is part of the internal environment that shows through at the interface between person and external environment as a function of cognitive task demands."

A person possesses intelligence insofar as he has learned, or can learn, to adjust himself to his environment." S. S. Colvin quoted in [39]

"We shall use the term 'intelligence' to mean the ability of an organism to solve new problems" W. V. Bingham [40]

Intelligence is a very general mental capability that, among other things, involves the ability to reason, plan, solve problems, think abstractly, comprehend complex ideas, learn quickly and learn from experience." [41]

There are probably as many definitions of intelligence as there are experts who study it. Simply put, however, intelligence is the ability to learn about, learn from, understand, and interact with one's environment. This general ability consists of a number of specific abilities, which include these specific abilities:

- Adaptability to a new environment or to changes in the current environment
- Capacity for knowledge and the ability to acquire it
- Capacity for reason and abstract thought
- Ability to comprehend relationships
- Ability to evaluate and judge
- Capacity for original and productive thought

Additional specific abilities might be added to the list, but they would all be abilities allowing a person to learn about, learn from, understand, and interact with the environment. Environment in this definition doesn't mean the environment of the earth, such as the desert, the mountains, etc., although it can mean that kind of environment. It has a wider meaning that includes a person's immediate surroundings, including the people around him or her. Environment in this case can also be something as small as a family, the workplace, or a classroom.

5.3 Type of intelligence: There are three types of Intelligence.

1. Human intelligence
2. Artificial intelligence
3. Machine intelligence

5.3.1 Human intelligence:

Human Intelligence, refers to intelligence gathering by means of interpersonal contact, as opposed to the more technical intelligence gathering disciplines such as SIGINT, IMINT and MASINT. NATO defines HUMINT as "a category of intelligence derived from information collected and provided by human sources [42].

Human intelligence, mental quality that consists of the abilities to learn from experience, adapt to new situations, understand and handle abstract concepts, and use knowledge to manipulate one's environment. Much of the excitement among investigators in the field of intelligence derives from their attempts to determine exactly what intelligence is. Different investigators have emphasized different aspects of intelligence in their definitions. For example, in a 1921 symposium the American psychologists Lewis M. Terman and Edward L. Thorndike differed over the definition of intelligence, Terman stressing the ability to think abstractly and Thorndike emphasizing learning and the ability to give good responses to questions.

More recently, however, psychologists have generally agreed that adaptation to the environment is the key to understanding both what intelligence is and what it does. Such adaptation may occur in a variety of settings: a student in school learns the material he needs to know in order to do well in a course; a physician treating a patient with unfamiliar symptoms learns about the underlying disease; or an artist reworks a painting to convey a more coherent impression [43].

Effective adaptation draws upon a number of cognitive processes, such as perception, learning, memory, reasoning, and problem solving. The main emphasis in a definition of intelligence, then, is that it is not a cognitive or mental process per se but rather a selective combination of these processes that is purposively directed toward effective adaptation. Thus, the physician who learns about a new disease adapts by perceiving material on the disease in medical literature, learning what the material contains, remembering the crucial aspects that are needed to treat the patient, and then utilizing reason to solve the problem of applying the information to the needs of the patient [43].

Here are 10 weaknesses of human intelligence [44]:

1. **Poor networking capabilities** – Transferring data between two human minds is slow, tedious, and error-prone, and the protocols are beyond confusing. It's a safe bet that Microsoft is involved.
2. **High maintenance costs** – Who'd want to use hardware that takes 8 hours to reboot, suffers frequent data loss, and is routinely riddled with viruses? OK, aside from the Dept of Homeland Security.
3. **Legacy code** – System instability often results from running outdated limbic legacy code. An impressive display of human intelligence is to queue up your fight-or-flight response when asking someone out on a date. It's nice to be prepared just in case she tries to eat you after she rejects you. DOS was great while it lasted, but it won't help us save Antarctica.
4. **Lack of error correction** – Once errors get into the system, they tend to stick around for a while. One bad decision left uncorrected will soon see more errors piled on top of it. This is how people sink into debt, put on weight, and get jobs, only to regret it later.
5. **Slow CPU** - Let's just say you're not the sharpest tool in the shed.
6. **Limited sensory input -** Our input channels are restricted to five senses (six for some of us), which all have a limited range of capabilities. We can't see what's behind a wall, we can't touch people at a distance, and we can't hear what people in the next building are saying… unless of course we work for the CIA.
7. **Unreliable hardware** – It's only a matter of time before a critical component suffers an irreparable crash, and then the whole system gets dumped in a human landfill.
8. **Faulty hard drive -** Witness recall is notoriously inaccurate. When asked to describe something they just saw, people frequently overlook critically important details, get existing details wrong, and add details that weren't present at all. It's a bad idea to put too much faith in your memories, since they're likely riddled with errors. You probably don't even remember what you're wearing.
9. **Low RAM** – You can only load and process so much complexity in your mind at once. There's so much you don't understand because your mind lacks the

capacity to store all the subtleties needed for true comprehension. Consequently, you’re probably making a total mess of things.

10. **Infinite loops** - Once a pattern of thought and behavior has become conditioned, it can be very difficult to reprogram. This causes bad habits, addictions, and Republicans.

5.3.1.1 Methods of human intelligence:

The approaches listed are not guaranteed solutions for every situation. Some individual approaches that may be suitable for one operating environment, such as when conducting HUMINT (Human Intelligence) contact operations, may be ineffective in another, such as interrogation. Some will be successful with one source and ineffective with another [45].

There are many types of approach Methods for Human Intelligence that can be employed on any detainee regardless of status or characterization, including EPWs (Enemy prisoners of war).

1. DIRECT APPROACH:

Almost all HUMINT collection begins with the direct approach. The exception to this is during elicitation operations that by their very nature are indirect. In using the direct approach, the HUMINT collector asks direct questions. The initial questions may be administrative or nonpertinent but the HUMINT collector quickly begins asking pertinent questions. The HUMINT collector will continue to use direct questions as long as the source is answering the questions in a truthful manner. When the source refuses to answer, avoids answering, or falsely answers a pertinent question, the HUMINT collector will begin an alternate approach strategy.

2. INCENTIVE APPROACH

The incentive approach is trading something that the source wants for information. The thing that you give up may be a material reward, an emotional reward, or the removal of a real or perceived negative stimulus. Even when the direct approach is successful, the HUMINT collector may use incentives to enhance rapport and to reward the source for cooperation and truthfulness. HUMINT collectors must be cautious in the use of incentives for the following reasons:

- The incentive must be believable and attainable. The incentive must be within the capability of the HUMINT collector's assumed persona to achieve. For **example**, if the detainee was captured after killing a US soldier, an incentive of release would not be realistic or believable.
- The HUMINT collector must provide any promised incentive. A simple promise of an incentive may be sufficient to obtain immediate cooperation.

3. ESTABLISH YOUR IDENTITY:

In using this approach, the HUMINT collector insists the detained source has been correctly identified as an infamous individual wanted by higher authorities on serious charges, and he is not the person he purports to be. In an effort to clear himself of this allegation, the source makes a genuine and detailed effort to establish or substantiate his true identity.

4. EMOTIONAL APPROACHES:

- Emotional approaches are centered on how the source views himself and his interrelationships with others. Through source observation and initial questioning, the HUMINT collector can often identify dominant emotions that motivate the EPW/detainee. The motivating emotion may be greed, love, hate, revenge, or others. The emotion may be directed inward (feelings of pride or helplessness) or outward (love of family).
- Although the emotion is the key factor, an emotional approach is normally worthless without an attached incentive. The incentive must meet the criteria listed above for the incentive approach to ensure that the incentive is believable and attainable. For **example**, this technique can be used on the enemy prisoners of war (EPW/detainee) who has a great love for his unit and fellow soldiers. Simply having the source express this emotion is not enough. After the source expresses this emotion.
- One common danger to the use of emotional approaches is the development of an emotional attachment on the part of the HUMINTcollector. It is natural that a source will develop an emotional attachment to the HUMINT collector. The HUMINT collector will often foster this attachment. The following are types of emotional approaches.

Emotional Love Approach:

- Love in its many forms (friendship, comradeship, patriotism, love of family) is a dominant emotion for most people. The HUMINT collector focuses on the anxiety felt by the source about the circumstances in which he finds himself, his isolation from those he loves, and his feelings of helplessness. The HUMINT collector directs the love the source feels toward the appropriate object: family, homeland, or comrades.
- The key to the successful use of this approach is to identify an action that can realistically evoke this emotion (an incentive) that can be tied to a detained source's cooperation. For example, if the source cooperates, he can see his family sooner, end the war, protect his comrades, help his country, help his ethnic group. A good HUMINT collector will usually orchestrate some futility with an emotional love approach to hasten the source's reaching the breaking point. In other words if the source does not cooperate, these things may never happen or be delayed in happening.

Emotional Hate Approach:

- The emotional hate approach focuses on any genuine hate, or possibly a desire for revenge, the source may feel. The HUMINT collector must clearly identify the object of the source's hate and, if necessary, build on those feelings so the emotion overrides the source's rational side. The source may have negative feelings about his country's regime, immediate superiors, officers in general, or fellow soldiers.
- The emotional hate approach may be effective on members of racial or religious minorities who have or feel that they have faced discrimination in military and civilian life. The "hate" may be very specific. For example, a source may have great love for his country, but may hate the regime in control. The HUMINT collector must be sure to correctly identify the specific object of the hate.

Emotional Fear-Up Approach:

- Fear is another dominant emotion that can be exploited by the HUMINT collector. In the fear-up approach, the HUMINT collector identifies a preexisting fear or creates a fear within the source. He then links the elimination or reduction of the fear to cooperation on the part of the source. If there is a justifiable fear, the HUMINT collector should present it and present a plan to mitigate it if the source cooperates (combination of emotional and incentive approaches). For example, an EPW source says that he will not cooperate because if he does his fellow prisoners will kill him or, if a contact source says that if people find out he is cooperating, his family will suffer.

- If there is no justified fear, the HUMINT collector can make use of nonspecific fears. "You know what can happen to you here?" A fear-up approach is normally presented in a level, unemotional tone of voice. For example, "We have heard many allegations of atrocities committed in your area and anyone that was involved will be severely punished" (non-specific fear).

Emotional Fear-Down Approach:

- The emotion of fear may dominate the source to the point where he is unable to respond rationally to questioning, especially in interrogation sources. However, the fear-down approach may be used in any MSO (*Military Source Operations)* where the source's state of mind indicates that it would be an appropriate approach to use [45].

5.3.1.2 Application of human intelligence:

The HUMINT (Human Intelligence) collector must be knowledgeable in a variety of areas in order to question sources effectively. The collector must prepare himself for operations in a particular area of intelligence responsibility (AOIR) by conducting research. Some of these areas of required knowledge are:

1. **The area of operations (AO)** including the social, political, and economic institutions; geography; history; language; and culture of the target area. Collectors must be aware of all ethnic, social, religious, political, criminal, tribal, and economic groups and the interrelationships between these groups.
2. **All current and potential threat forces** within the AOIR and their organization, equipment, motivation, capabilities, limitations, and normal operational methodology.
3. **The collection requirements,** including all specific information requirements and indicators that will lead to the answering of the intelligence requirements.

4. **Cultural awareness** in the various areas of operations will have different social and regional considerations that affect communications and can affect the conduct of operations. These may include social taboos, desired behaviors, customs, and courtesies. The staff must include this information in pre-deployment training at all levels to ensure that personnel are properly equipped to interact with the local populace.
5. **Proficiency in the target language**. The HUMINT collector can normally use an interpreter and machine translation as they are developed to conduct questioning. Language proficiency is a benefit to the HUMINT collector in a number of ways: He can save time in questioning, be more aware of nuances in the language that might verify or deny truthfulness, and better control and evaluate interpreters.
6. **Understanding basic human behavior**. A HUMINT collector can best adapt himself to the source's personality and control of the source's reactions when he understands basic behavioral factors, traits, attitudes, drives, motivations, and inhibitions. He must not only understand basic behavioral principles but also know how these principles are manifested in the area and culture in which he is operating.
7. **Neurolinguistics**. Neurolinguistics is a behavioral communication model and a set of procedures that improve communication skills. The HUMINT collector should read and react to nonverbal communications. He must be aware of the specific neurolinguistic clues of the cultural framework in which he is operating [45].

5.3.2 Artificial intelligence:

Intelligence is the ability to think, to imagine, to create, memorize, understand, recognized patterns, make choices, adapt to change and learn from experience. Artificial intelligence is a human endeavor to create a non-organic machine-based entity that has all the above abilities of natural organic intelligence. Hence it is known as 'Artificial Intelligence' (AI) [46].

The ability of a computer or other machine to perform actions thought to require intelligence. Among these actions are logical deduction and inference, creativity, the ability to make decisions based on past experience or insufficient or conflicting information, and the ability to understand spoken language.

The goal of research on artificial intelligence is to understand the nature of thought and intelligent behavior and to design intelligent systems. A computer is not really intelligent; it just follows directions very quickly. At the same time, it is the speed and memory of modern computers that allows researchers to manage the huge quantities of data necessary to model human thought and behavior. An intelligent machine would be more flexible than a computer and would engage in the kind of "thinking" that people actually do. An example is vision. In theory, a network of sensors combined with systems for interpreting the data could produce the kind of pattern recognition that we take for granted as seeing and understanding what we see. In fact, developing software that can recognize subtle differences in objects (such as those we use to recognize human faces) is very difficult. The recognition of differences that we can perceive without deliberate effort would require massive amounts of data and elaborate guidelines to be recognized by an artificial intelligence system. According to the famous Turing Test, proposed in 1950 by British mathematician and logician Alan Turing, a machine would be considered intelligent if it could convince human observers that another human, rather than a machine, was answering their questions in conversation [47].

There are four approaches to AI have been followed, each by different people by the different methods:

1. **Acting humanly: The Turing Test approach:**

The Turing Test, proposed by Alan Turing (1950), was designed to provide a satisfactory operational definition of intelligence. A computer passes the test if a human interrogator, after posing some written questions, cannot tell whether the written responses come from a person or from a computer. The computer would need to possess the following capabilities:

- **Natural language processing** to enable it to communicate successfully in English.
- **Knowledge representation** to store what it knows or hears.
- **Automated reasoning** to use the stored information to answer questions and to draw new conclusions.
- **Machine learning** to adapt to new circumstances and to detect and extrapolate patterns.
- **Computer vision** to perceive objects, and ROBOTICS.

- **Robotics** to manipulate objects and move about.

2. **Thinking humanly: The cognitive modeling approach:**

If we are going to say that a given program thinks like a human, we must have some way of determining how humans think. We need to get inside the actual workings of human minds.

There are three ways to do this: through introspection—trying to catch our own thoughts as they go by; through psychological experiments—observing a person in action; and through brain imaging—observing the brain in action. Once we have a sufficiently precise theory of the mind, it becomes possible to express the theory as a computer program. If the program's input—output behavior matches corresponding human behavior, that is evidence that some of the program's mechanisms could also be operating in humans.

3. **Thinking rationally: The "laws of thought" approach:**

The Greek philosopher Aristotle was one of the first to attempt to codify "right thinking," that SYLLOGISM is, irrefutable reasoning processes. His syllogisms provided patterns for argument structures that always yielded correct conclusions when given correct premises—for example, "Socrates is a man; all men are mortal; therefore, Socrates is mortal." These laws of thought were LDG IC supposed to govern the operation of the mind; their study initiated the field called logic.

There are two main obstacles to this approach. First, it is not easy to take informal knowledge and state it in the formal terms required by logical notation, particularly when the knowledge is less than 100% certain. Second, there is a big difference between solving a problem "in principle" and solving it in practice. Even problems with just a few hundred facts can exhaust the computational resources of any computer unless it has some guidance as to which reasoning steps to try first. Although both of these obstacles apply to any attempt to build computational reasoning systems, they appeared first in the logicist tradition.

4. **Acting rationally: The rational agent approach:**

An agent is just something that acts (agent comes from the Latin agere, to do). Of course, all computer programs do something, but computer agents are expected to do more: operate autonomously, perceive their environment, persist over a prolonged time period, adapt to RATIONAL AGENT change, and create and pursue goals. A rational agent is one that acts so as to achieve the best outcome or, when there is uncertainty, the best expected outcome [48].

5.3.2.1 Methods of artificial intelligence:

Initially, researchers thought that creating an AI would be simply writing programs for each and every function of intelligence, performs! As they went on with this task, they realized that this approach was too shallow. Even simple functions like face recognition, special sense, pattern recognition and language comprehension were beyond their programming skills!

They understood that to create an AI, they must delve deeper into natural intelligence first. They tried to understand how cognition, comprehension, decision-making happen in the human mind. They had to understand what understanding really means! Some went into the study of the brain and tried to understand how the network of neurons creates the mind.

Thus, researchers branched into different approaches, but they had the same goal of creating intelligent machines. Let us introduce ourselves to some of the main approaches to artificial intelligence. They are divided into two main lines of thought, the bottom up and the top down approach [46]:

- **Neural Networks**:

This is the bottom up approach. It basically aims at mimicking the structure and functioning of the human brain, to create intelligent behavior. Researchers are attempting to build a silicon-based electronic network that is modelled on the working and form of the human brain! Our brain is a network of billions of neurons, each connected with the other.

At an individual level, a neuron has very little intelligence, in the sense that it operates by a simple set of rules, conducting electric signals through its network. However, the combined network of all these neurons creates intelligent behavior that is unrivalled and unsurpassed. So these researchers created network of electronic analogues of a neuron, based on Boolean logic. Memory was recognized to be an electronic signal pattern in a closed neural network.

How the human brain works is, it learns to realize patterns and remembers them. Similarly, the neural networks developed have the ability to learn patterns and remember. This approach has its limitations due to the scale and complexity of developing an exact replica of a human brain, as the neurons number in billions! Currently, through simulation techniques, people create virtual neural networks. This approach has not been able to achieve the ultimate goal but there is a very positive progress in the field. The progress in the development of parallel computing will aid it in the future.

- **Expert Systems**:

This is the top down approach. Instead of starting at the base level of neurons, by taking advantage of the phenomenal computational power of the modern computers, followers of the expert systems approach are designing intelligent machines that solve problems by deductive logic. It is like the dialectic approach in philosophy. This is an intensive approach as opposed to the extensive approach in neural networks. As the name expert systems suggest, these are machines devoted to solving problems in very specific niche areas. They have total expertise in a specific domain of human thought. Their tools are like those of a detective or sleuth. They are programmed to use statistical analysis and data mining to solve problems. They arrive at a decision through a logical flow developed by answering yes-no questions.

Chess computers like Fritz and its successors that beat chess grandmaster Kasparov are examples of expert systems. Chess is known as the drosophila or experimental specimen of artificial intelligence.

5.3.2.2 Application of artificial intelligence:

There are the many examples where these type of technologies have been used, given below:

1. Computer vision:

The world is composed of three-dimensional objects, but the inputs to the human eye and computers' TV cameras are two dimensional. Some useful programs can work solely in two dimensions, but full computer vision requires partial three-dimensional information that is not just a set of two-dimensional views. At present there are only limited ways of representing three-dimensional information directly, and they are not as good as what humans evidently use.

2. Understanding natural language:

Just getting a sequence of words into a computer is not enough. Parsing sentences is not enough either. The computer has to be provided with an understanding of the domain the text is about, and this is presently possible only for very limited domains.

3. **Game playing:**

You can buy machines that can play master level chess for a few hundred dollars. There is some AI in them, but they play well against people mainly through brute force computation--looking at hundreds of thousands of positions. To beat a world champion by brute force and known reliable heuristics requires being able to look at 200 million positions per second [49].

4. **Weather Forecast:**

Neural networks are used for predicting weather conditions. Previous data is fed to a neural network which learns the pattern and uses that knowledge to predict weather patterns [46].

5. **Swarm Intelligence:**

This is an approach to, as well as application of artificial intelligence similar to a neural network. Here, programmers study how intelligence emerges in natural systems like swarms of bees even though on an individual level, a bee just follows simple rules. They study relationships in nature like the prey-predator relationships that give an insight into how intelligence emerges in a swarm or collection from simple rules at an individual level. They develop intelligent systems by creating agent programs that mimic the behavior of these natural systems! [46].

6. **Speech recognition:**

In the 1990s, computer speech recognition reached a practical level for limited purposes. Thus United Airlines has replaced its keyboard tree for flight information by a system using speech recognition of flight numbers and city names. It is quite convenient. On the the other hand, while it is possible to instruct some computers using speech, most users have gone back to the keyboard and the mouse as still more convenient [49].

In addition **examples** where the technologies have been effectively utilized will be presented.

7. **Neural Network:**

- This technology is based loosely upon the cellular structure of the human brain.

Cells, or storage locations, and connections between the locations are established in the computer. As in the human brain, connections among the cells are strengthened or weakened based upon their ability to yield "productive" results.

- Neural Networks are used today for helping to identify the presence or absence of certain types of explosives in checked airline baggage. As the baggage passes through a special scanner, the scanner “sniffs” out the explosives by bombarding the luggage with harmless atomic level particles. The system then passes the pattern produced through the neural network that has been trained to see the peculiar patterns of the explosives [50].

8. **Expert systems:**

- A “knowledge engineer” interviews experts in a certain domain and tries to embody their knowledge in a computer program for carrying out some task. How well this works depends on whether the intellectual mechanisms required for the task are within the present state of AI. When this turned out not to be so, there were many disappointing results. One of the first expert systems was MYCIN in 1974, which diagnosed bacterial infections of the blood and suggested treatments. It did better than medical students or practicing doctors, provided its limitations were observed. Namely, its ontology included bacteria, symptoms, and treatments and did not include patients, doctors, hospitals, death, recovery, and events occurring in time. Its interactions depended on a single patient being considered. Since the experts consulted by the knowledge engineers knew about patients, doctors, death, recovery, etc., it is clear that the knowledge engineers forced what the experts told them into a predetermined framework. In the present state of AI, this has to be true. The usefulness of current expert systems depends on their users having common sense [49].
- These systems are usually built using large sets of “rules.”An expert, who has developed them mentally after perhaps a decade or more of practice in a specialty area, establishes these rules. A specialist, known as a knowledge engineer, extracts the rules from the expert and programs them into a computer. An example of a small rule set follows [51].

(a) *‘IF the relief pressure valve is less than .25 open and the pressure setting is greater than 160 kg,*

THEN pressure-category is high.

IF then temperature is less than 250 centigrade,

THEN category is normal,

ELSE temperature-category is hot.

IF pressure-category is high and temperature-category is hot,

THEN send operator alert'.

Expert Systems are established for processes where there is a need;

1) for a narrow area of expertise to be more widely known, or
2) to allow sophisticated processes to be run without human intervention.

(b) A **classic example** of the former is a need for understanding and interpreting the rules of code and regulations set forth by the U.S. Internal Revenue Service. To provide benefit to the average citizen preparing their taxes, these rules are programmed into popular software packages such as Tax Cut and TurboTax. A classic case of the second need is a back end system programmed by the Credit Card Division of the American Express Company. This system uses sophisticated rules to determine whether a credit transaction should be approved, denied, or be interrupted by human intervention [51].

5.3.3 Machine intelligence:

In this section, we will start with an influential attempt to define `intelligence', and then we will move to a consideration of how human intelligence is to be investigated on the machine model.

The Turing Test:

One approach to the mind has been to avoid its mysteries by simply *defining* the mental in terms of the behavioral. This approach has been popular among thinkers who fear that acknowledging mental states that do not reduce to behavior would make psychology unscientific, because unreduced mental states are not intersubjectively accessible in the manner of the entities of the hard sciences. "Behaviorism", as the attempt to reduce the mental to the behavioral is called, has often been regarded as refuted, but it periodically reappears in new forms.

'Behaviorists' don't define the mental in terms of just plain *behavior*, since after all something can be intelligent even if it has never had the chance to exhibit its intelligence. Behaviorists define the mental not in terms of behavior, but rather behavioral *dispositions*, the tendency to emit certain behaviors given certain stimuli. It is important that the stimuli and the behavior be specified non-mentalistically. Thus, intelligence could not be defined in terms of the disposition to give sensible responses to questions, since that would be to define a mental notion in terms of another mental notion (indeed, a closely related one).

An especially influential behaviorist definition of intelligence was put forward by A. M. Turing (1950). Turing, one of the mathematicians who cracked the German code during World War II, formulated the idea of the universal Turing machine, which contains, in mathematical form, the essence of the programmable digital computer. Turing wanted to define intelligence in a way that applied to both men and machines, and indeed, to anything that is intelligent [52]. His version of behaviorism formulates the issue of whether machines could think or be intelligent in terms of whether they could pass the following test: a judge in one room communicates by teletype (This was 1950!) with a computer in a second room and a person in a third room for some specified period. (Let's say an hour).

Turing suggested that we replace the concept of intelligence with the concept of passing the Turing test. But what is the replacement *for*? If the purpose of the replacement is practical, the Turing test is not enormously useful. If one wants to know if a machine does well at playing chess or diagnosing pneumonia or planning football strategy, it is better to see how the machine performs in action than to make it take a Turing test. For one thing, what we care about is that it do well at detecting pneumonia, not that it do it in a way indistinguishable from the way a person would do it. So if it does the job, who cares if it doesn't pass the Turing test?

A second purpose might be utility for theoretical purposes. But machines that can pass the Turing test such as Weizenbaum's ELIZA (see below) have been dead ends in artificial intelligence research, not exciting beginnings [53]. A third purpose is the purpose of conceptual clarification that comes closest to Turing's intentions. Turing was famous for having formulated a precise mathematical concept that he offered as a replacement for the vague idea of mechanical computability. The precise concept (computability by a Turing machine) did everything one would want a precise concept of mechanical computability to do. No doubt, Turing hoped that the Turing test conception of intelligence would yield everything one would want from a definition of intelligence without the vagueness of the ordinary concept.

5.3.3.1 Methods of machine intelligence:

We can understand the methods of machine intelligence by two kinds of definitions of intelligence.

Two kinds of definitions of intelligence:

We have been talking about an attempt to define intelligence using the resources of the Turing Test. However, there is a very different approach to defining intelligence.

To explain this approach, it will be useful to contrast two kinds of definitions of water. One might be better regarded as a definition of the word *`water'*. The word might be defined as the colorless, odorless, tasteless liquid that is found in lakes and oceans. In this sense of ‘definition', the definition of `water' is available to anyone who speaks the language, even someone who knows no science. But one might also define water by saying what water really is, that is, by saying what physico-chemical structure in fact makes something pure water. The answer to this question would involve its chemical constitution: H2O. Defining a *word* is something we can do in our armchair, by consulting our linguistic intuitions about hypothetical cases, or, bypassing this process, by simply stipulating a meaning for a word. Defining (or explicating) the *thing* is an activity that involves empirical investigation into the nature of something in the world.

What we have been discussing so far is the first kind of definition of intelligence, the definition of the word, not the thing. Turing's definition is not the result of an empirical investigation into the components of intelligence of the sort that led to the definition of water as H2O. Rather, he hoped to avoid muddy thinking about machine intelligence by stipulating that the word `intelligent' should be used a certain way, at least with regard to machines. Quite a different way of proceeding is to investigate intelligence *itself* as physical chemists investigate water. We will consider how this might be done in the next section, but first we should note a complication.

There are two kinds (at least) of kinds: *structural* kinds such as *water* or *tiger*, and *functional* kinds such as *mouse-trap* or *gene*. A structural kind has a "hidden compositional essence"; in the case of water, the compositional essence is a matter of its molecules consisting of two hydrogen molecules and one oxygen molecule. Functional kinds, by contrast, have no essence that is a matter of composition. A certain sort of function, a causal role, is the key to being a mousetrap or a carburetor. (The full story is quite complex: something can be a mousetrap because it is made to be one even if it doesn't fulfill that function very well.) What makes a bit of DNA a gene is its function with respect to mechanisms that can read the information that it encodes and use this information to make a biological product.

Now the property of being intelligent is no doubt a functional kind, but it still makes sense to investigate it experimentally, just as it makes sense to investigate genes experimentally. One topic of investigation is the role of intelligence in problem solving, planning, decision making, etc. Just what functions are involved in a functional kind is often a difficult and important empirical question. The project of Mendelian genetics has been to investigate the function of genes at a level of description that does not involve their molecular realizations. A second topic of investigation is the nature of the realizations that have the function in us, in humans: DNA in the case of genes. Of course, if there are Martians, their genes may not be composed of DNA. Similarly, we can investigate the functional details and physical basis of human intelligence without attention to the fact that our results will not apply to other mechanisms of other hypothetical intelligences.

Functional analysis:

Both types of projects just mentioned can be pursued via a common methodology, a methodology sometimes known as *functional analysis*. Think of the human mind as represented by an intelligent being in the head, a "homunculus". Think of this homunculus as being composed of smaller and stupider homunculi, and each of these being composed of still smaller and still stupider homunculi until you reach a level of completely mechanical homunculi [54].

5.3.3.2 Application of machine intelligence:

The acceptance of AI technology in industry will clearly depend on the capability of moving developments from the laboratory into the field. There are much of the development of AI in universities and research laboratories necessarily on prototype scale projects. Very few of these projects progress into fully fledged production systems. This application claims that the challenges highlighted in developing and maintaining production systems are different to those encountered in developing prototype systems.

This claim is based on BHP Central Research Laboratories' experiences in developing intelligent operator guidance systems (OGS's), in particular areal time system to advise process operators in an iron ore sinter plant that has been on-line in test mode since late 1986 [55].

Prototype system:

Iron *ore* sintering is the initial process that iron ore undergoes on its path to conversion into steel. The process is required to prepare the feed for the blast furnace, which produces pig iron. The important feats of the process are:

1. The process is complicated by a complex inter-relationship between multiple variables and long time constants.
2. The fundamental knowledge about the process is poor.
3. Consistent operating practices are essential yet difficult to achieve **as** operator performance will vary from shift to shift [55].

There are other applications are given below [56]:

- **Capacity planning user manual:**

The Capacity and Calibration Model is part of Scianta Intelligence's infrastructure performance management suite. The capacity model correlates incoming data streams from client applications and servers as well as network traffic to discover the periodicity of normal behavior, learn the combined behavior of the data streams over time, and predict threshold violations, anomalous behaviors, and large scale behavior changes in the network ecosystem.

- **Crew scheduler concepts presentation:**

Designed as a high level introduction to the over-all crew scheduler and its features, this slide presentation (in PDF format) is aimed at middle level managers, engineers, system architects, and analysts who are looking to understand the basic workings of our crews scheduling optimization system.

- **Crew scheduler principles and concepts:**

Our integrated family of Java J2EE classes support a comprehensive but flexible crew scheduling and optimization capability. The Principles and Concepts manual provides a detailed introduction to and discussion of the underlying scheduling and resource allocation concepts, the idea behind resource constrained scheduling, and the powerful genetic scheduler used to quickly find and rank optimal and near-optimal solutions. This user manual is intended for managers, analysts, and IT operations staff as a reference in understanding the use and capabilities of the model [55].

5.4 Application of human intelligence in AI (Observation):

The intelligence of a machine that can be successfully perform any intellectual task that a human being can. It is a primary goal of artificial intelligence research and an important topic for science fiction writers and futurists. If we talk about Strong AI, we can say that it is also referred to as "artificial general intelligence" or as the ability to perform "general intelligent action". Science fiction associates strong AI with such human traits as consciousness, sentience, sapience and self-awareness.

I am not sure that all human intelligence can be described with algorithms, for example emotional intelligence (as Lee mentions), empathy, the kind of creative genius that creates mind blowing art. And if the computers are created by humans, maybe they will be kind of human as well? Or the distinction might become more unclear, what is man and what is machine [57].

How can consciousness arise from algorithms seems to be the main hurdle and emotions, and creativity, and sexuality.

There are thoughts in different directions trying to see how this might (in principle) be done, and what some problems are.

I would conclude that AI can supersede HI, but it seems we don't seem to know how. How long it will take is difficult to predict, but somewhere between 2 and 100 years would be broad boundaries [57].

Everything in the body, including the brain, is very dependent on every other part of it. That is how life works. The models that describe how the brain or the nervous system works are just models, not the complex reality. It is popular to talk about the different parts of the brain and a lot of research has been done, but in reality the parts of the brain mostly works together and functions are more often than not shared by nearby areas [58].

Our bodies are biochemical machines, not electronic mechanic devices. I have still not met anyone who can comprehend ALL the different new data that keep developing in the fields of medicine, biology, life sciences and similar subjects. And I think that comprehension is lacking in this conversation, because WE are humans and the people who want to create AI are humans so they will create something they can comprehend [58].

Although our mind and body are intertwined, or, as I would put it: our mind is a part of our body (I don't like the dualism, as it sometimes seem to suggest that there is something as an independent mind without a body. There are independent bodies without a mind. The ones we call dead for example).

This does not imply one needs a human body.

Any kind of body that can do the necessary computations would do. Saying only a human body could do this, is making a very anthropocentric fallacy!

Of course you need to have a lot of sensors on your bot, as that are ways to obtain (new) data which is essential for the learning process of a bot. I would add a lot more sensors than we humans have (infrared, ultrasound, UV, more chemical sensitivity, finer thermal, gravity, accelerometers, pressure, torque, luminance, magnetism, electricity etc.) [57].

5.5 Artificial intelligence in 2012:

While this victory of machine over man was considered by many a triumph for artificial intelligence (AI), John McCarthy (Sept. 4, 1927–Oct. 24, 2011), who not only was one of the founding pioneers of AI but also coined the very name of the field, was rather dismissive of this accomplishment. "The fixation of most computer chess work on success in tournament play has come at scientific cost," he argued. McCarthy was disappointed by the fact that the key to Deep Blue's success was its sheer compute power rather than a deep understanding, exhibited by expert chess players, of the game itself.

AI's next major milestone occurred last February with IBM's Watson program winning a "Jeopardy!" match against Brad Rutter, the biggest all-time money winner, and Ken Jennings, the record holder for the longest championship streak. This achievement was also dismissed by some. "Watson doesn't know it won on "Jeopardy!"," argued the philosopher John Searle, asserting that "IBM invented an ingenious program, not a computer that can think."

In fact, AI has been controversial from its early days. Many of its early pioneers overpromised. "Machines will be capable, within 20 years, of doing any work a man can do," wrote Herbert Simon in 1965. At the same time, AI's accomplishments tended to be underappreciated. "As soon as it works, no one calls it AI anymore," complained McCarthy. Yet it is recent worries about AI that indicate, I believe, how far AI as come.

In April 2000, Bill Joy, the technologists' technologist, wrote a "heretic" article entitled "Why the Future Doesn't Need Us" for *Wired* magazine, "Our most powerful 21st-century technologies—robotics, genetic engineering, and nanotech—are threatening to make humans an endangered species," he wrote. Joy's article was mostly ignored, but in August 2011 Jaron Lanier, another widely respected technologist, wrote about the impact of AI on the job market. In the not-too-far future, he predicted, it would just be inconceivable to put a person behind the wheel of a truck or a cab. "What do all those people do?" he asked.

Slate magazine ran a series of articles in September 2011 titled "Will Robots Steal Your Job?" According to writer Farhad Manjoo, who detailed the many jobs we can expect to see taken over by computers and robots in the coming years, "You're highly educated. You make a lot of money. You should still be afraid."

In fact, worries about the impact of technology on the job market are not only about the far, but also the not too far future [59].

5.6 Soft computing:

Soft Computing has been given a broader definition in the literature to include Fuzzy Sets, Rough Sets, Neural Networks, Evolutionary Computing, Probabilistic and Evidential Reasoning, Multivalued Logic, and related fields. Other scientists proposed the notion of Extended Soft Computing (ESC) as a new discipline developed by adding Chaos Computing and Immune Network Theory to the classical Soft Computing, as defined and proposed by Lotfi Zadeh. ESC was proposed for explaining complex systems and cognitive and reactive AIs. Moreover, Fuzzy Logic, which is the basis on which Soft Computing is built, has been expanded into what is known today as Type-2 Fuzzy Logic.

5.6.1 Fuzzy logic:

There has been a lot of research and development in the field of fuzzy sets and fuzzy logic after pioneering work by L Zadeh. In fact, it made a revolution in the field of uncertainty, which challenged the probability theory. Instead of two valued logic as used in probability thoery, fuzzy set advocated whole range of values from 0 to 1.

In 1965, Professor L.A. Zadeh of the University of California, Berkely presented his paper outlining fuzzy theory in which he introduced the concept of fuzzy set theory and operation, fuzzy logic based controller etc. In about 1970, fuzzy logic theory began to produce result in Japan, Europe and China [83]. In the year 1987, 16 station subway railway systems were built which worked with a fuzzy logic-based automatic train operation control system in sendai, japan. The ride is so smooth that the riders do not need to hold straps, and the controller makes seventy percent fewer judgmental errors in acceleration and braking than human operators do. Fuzzy logic is a powerful problem-solving methodology with a myriad of applications in embedded control and information processing. Fuzzy provides a remarkably simple way to draw definite conclusions from vague, ambiguous or imprecise information. In a sense, fuzzy logic resembles human decision making with its ability to work from approximate data and find precise solutions. Unlike classical logic, which requires a deep understanding of a system, exact equations, and precise numeric values, fuzzy logic incorporates an alternative way of thinking, which allows modeling complex systems using a higher level of abstraction originating from our knowledge and experience. Fuzzy logic allows expressing this knowledge with subjective concepts such as very hot, bright red, and a long time, which are mapped into exact numeric ranges.

5.6.1.1 Fuzzy set:

Fuzzy set is defined as $A = \{(x, \mu_A(x)) : x \in X\}$, where, $\mu_A(x)$ is called the membership function (MF) for the fuzzy set *A*. The MF maps each element of *X* to a membership degree between 0 and 1 (included). Obviously, the definition of a fuzzy set is a simple extension of the definition of a classical (crisp) set in which the characteristic function is permitted to have any value between 0 and 1. If the value of the membership function is restricted to either 0 or 1, then *A* is reduced to a classical set. Usually, *X* is referred to as the universe of discourse, or simply the universe, and it may consist of discrete (ordered or non-ordered) objects or it can be a continuous space.

Support: The support of a fuzzy set A is the set of all points with nonzero membership degree in A: $\sup(A) = \{x \in X, \mu_A(x) > 0\}$

Core: The core of a fuzzy set A is the set of all points with unit membership degree in A:

$$core(A) = \{x \in X, \mu_A(x) = 1\}$$

Normality: A fuzzy set A is normal if its core is nonempty. In other words, we can always find at least a point $x \in X$ such that $\mu_A(x) = 1$

Crossover points: A crossover point of a fuzzy set A is a point $x \in X$ at which $\mu_A(x) = 0.5$

Fuzzy numbers: A fuzzy number A is a fuzzy set in the real line that satisfies the conditions for both normality and convexity. Most fuzzy sets used in the literature satisfy the conditions for normality and convexity, so fuzzy numbers are the most basic type of fuzzy sets. Union, intersection, and complement are the most basic operations on classical sets. On the basis of these three operations, a number of identities can be established. This definition of containment is, of course, a natural extension of the one for ordinary sets [60].

5.6.1.2 Fuzzy logic controller:

The fuzzy logic controller provides an algorithm, which converts the expert knowledge into an automatic control strategy. Fuzzy logic is capable of handling approximate information in a systematic way and therefore it is suited for controlling non linear systems and is used for modeling complex systems, where an inexact model exists or systems where ambiguity or vagueness is common. The fuzzy control systems are rule-based systems in which a set of fuzzy rules represent a control decision mechanism for adjusting the effects of certain system stimuli. With an effective rule base, the fuzzy control systems can replace a skilled human operator. The rule base reflects the human expert knowledge, expressed as linguistic variables, while the membership functions represent expert interpretation of those variables.

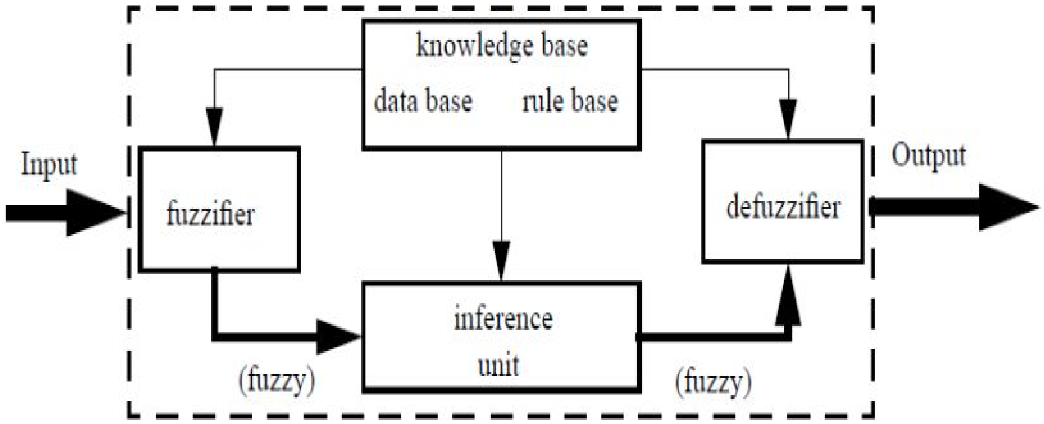


Figure 5.1: Block diagram of fuzzy logic system

The above figure 5.1 shows the block diagram of fuzzy logic system. It has different blocks: fuzzifier, knowledge base, inference unit and defuzzifier. The crisp inputs are supplied to the input side Fuzzification unit. The Fuzzification unit converts the crisp input in to fuzzy variable. The fuzzy variables are then passed through the fuzzy rule base. The fuzzy rule base computes the input according to the rules and gives the output. The output is then passed through de- fuzzification unit where the fuzzy output is converted to crisp output (The role of the fuzzifier in a Fuzzy Logic System is to convert a *crisp* input variable into a fuzzy set that is ready to be processed by the inference engine. The inference engine using the fuzzified inputs and the rules stored in the rule base processes the incoming data and produces an (fuzzy) output. This output needs to be used in the outside world and thus needs to be converted from fuzzy to crisp; the defuzzifier performs this operation) [61].

5.6.1.3 Applications of fuzzy logic:

In the decade after Dr. Zadeh's seminal paper on fuzzy sets many theoretical developments in fuzzy logic took place in the United States, Europe, and Japan. From the mid-Seventies to the present, however, Japanese researchers have been a primary force in advancing the practical implementation of the theory; they have done an excellent job of commercializing this technology. Fuzzy logic affects many disciplines. In videography, for instance, Fisher, Sanyo, and others make fuzzy logic camcorders, which offer fuzzy focusing and image stabilization. Mitsubishi manufactures a fuzzy air conditioner that controls temperature changes according to human comfort indexes. Matsushita builds a fuzzy washing machine that combines smart sensors with fuzzy logic. The sensors detect the color and kind of clothes present and the quantity of grit, and a fuzzy microprocessor selects the most appropriate combination from 600 available combinations for water temperature, detergent amount and washes and spins cycle times. The Japanese City of Sendai has a 16-station subway system that is controlled by a fuzzy computer. The ride is so smooth that the riders do not need to hold straps, and the controller makes 70 percent fewer judgmental errors in acceleration and braking than human operators do. Nissan introduced a fuzzy automatic transmission and a fuzzy anti-skid braking system in one of their recent luxury cars. Tokyo's stock market has stock-trading portfolios based on fuzzy logic that outperformed the Nikkei Exchange average. In Japan, there are fuzzy golf diagnostic systems, fuzzy toasters, and many other industrial fuzzy control processes. .

With increasing complexities in system engineering, the focus of fuzzy control is moving from elementary control problems to higher levels in the system hierarchy such as supervisory control, monitoring and diagnosis, and logistic support. It is to be noted that telecommunications, which is one of the major future industries, has started investigating fuzzy control for communication systems and that several pilot projects have been initiated for tackling routing and overload handling problems. So far, the majority of existing applications are purely software-based. However, general purpose fuzzy logic processors or coprocessors will be found to be useful in extremely time critical applications like pattern recognition task in a complex plant automation and in mass produced automotive electronics. From the above discussions it is apparent that fuzzy control has tremendous scope in the knowledge based systems approach to closed loop control system, which may be defined as:

A knowledge based system for closed loop control is a system which enhances the performance, reliability and robustness of control by incorporating knowledge which can not be captured in the analytical model used for controller design and that is taken care of by manual modes of operation or by other safety and ancillary logic mechanism.

In this thesis work, we will optimize the parameter of PID controller using Particle Swarm Optimization. So we will first study of the swarm intelligence.

5.6.2 Swarm intelligence:

It is an artificial intelligence technique based around on the study of collective behavior in decentralized, self-organized systems [62]. The expression "swarm intelligence" was introduced by Beni & Wang in 1989, in the context of cellular robotic systems. Swarm intelligence systems are typically made up of a population of simple agents interacting locally with one another and also with their environment. Usually there is no centralized control structure dictating how the individual agents should behave, but local interactions between such agents often lead to the emergence of a global behavior. Examples of systems like this can be found in nature, including ant colonies, bird flocking, bee swarming, animal herding, bacteria molding and fish schooling.

5.6.2.1 Swarm intelligence as an optimization method:

Two of the most successful swarm intelligence techniques [62] currently in existence are **Ant Colony Optimization** (ACO) and **Particle Swarm Optimization** (PSO). ACO is a meta-heuristic optimization algorithm that can be used to find approximate solutions to difficult combinatorial optimization problems. In ACO artificial ants build solutions by moving on the problem graph and they, mimicking real ants, deposit artificial **pheromone** on the graph in such a way that future artificial ants can build better solutions. ACO has been applied successfully to an impressive number of optimization problems. PSO is a global minimization technique for dealing with problems in which a best solution can be represented as a point or surface in an n-dimensional space.

5.7 ANT colony optimization:

In computer science and operations research, the ant colony optimization algorithm **(ACO)** is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs.

This algorithm is a member of the ant colony algorithms family, in swarm intelligence methods, and it constitutes some metaheuristic optimizations. Initially proposed by Marco Dorigo in 1992 in his PhD thesis [63], the first algorithm was aiming to search for an optimal path in a graph, based on the behavior of ants seeking a path between their colony and a source of food. The original idea has since diversified to solve a wider class of numerical problems, and as a result, several problems have emerged, drawing on various aspects of the behavior of ants. So we can say, Ant Colony Optimization (ACO) is a paradigm for designing metaheuristic algorithms for combinatorial optimization problems.

The first algorithm which can be classified within this framework was presented in 1991 [64] and, since then, many diverse variants of the basic principle have been reported in the literature. The essential trait of ACO algorithms is the combination of a priori information about the structure of a promising solution with a posteriori information about the structure of previously obtained good solutions.

The **characteristic** of ACO algorithms is their explicit use of elements of previous solutions. In fact, they drive a constructive low-level solution, as GRASP [65] does, but including it in a population framework and randomizing the construction in a Monte Carlo way. A Monte Carlo combination of different solution elements is suggested also by Genetic Algorithms [66], but in the case of ACO the probability distribution is explicitly defined by previously obtained solution comp onents. an ACO algorithm includes two more mechanisms *: trail evaporation* and, optionally, *daemon actions*. Trail evaporation decreases all trail values over time, in order to avoid unlimited accumulation of trails over some comp onent. Daemon actions can be used to implement centralized actions which cannot be performed by single ants, such as the invocation of a local optimization procedure, or the update of global information to be used to decide whether to bias the search process from a non-local perspective [67].

5.7.1 Applications of ACO:

Water resources systems analysis is the science of developing and applying mathematical operations research methodologies to water resources systems problems comprised of reservoirs, rivers, watersheds, groundwater, distribution systems, and others, as standalone or integrated systems, for single or multiobjective problems, deterministic or stochastic. The scientific and practical challenge in dealing quantitatively with water resources systems analysis problems is in taking into consideration from a systems perspective, social, economic, environmental, and technical dimensions, and integrating them into a single framework for trading-off in time and in space competitive objectives. Inherently, such problems involve modeling of water quantity and quality for surface water, groundwater, water distribution systems, reservoirs, rivers, lakes, and other systems as stand alone or combined systems.

5.8 Particle swarm optimization:

Particle Swarm Optimization (PSO) originally was introduced by Kennedy and Eberhart in 1995 [68] and it has been applied to a wide variety of applications and because of it has simplicity, easy implementation, it has been found to continuous in solving of continuous nonlinear optimization problem. PSO is a computational algorithm technique based on swarm intelligence. This method is motivated by the observation of social interaction and animal behaviors such as fish schooling and bird flocking.

It mimics the way they find food by the cooperation and competition among the entire population [69]. A swarm consists of individuals, called particles, each of which represents a different possible set of the unknown parameters to be optimized. The swarm is initialized with a population of random solutions [70]. In a PSO system, particles fly around in a multi-dimensional search space adjusting its position according to its own experience and the experience of its neighbouring particle. The goal is to efficiently search the solution space by swarming the particles towards the best fitting solution encountered in previous iterations with the intention of encountering better solutions through the course of the process and eventually converging on a single minimum or maximum solution [71]. The performance of each particle is measured according to a pre-defined fitness function, which is related to the problem being solved. PSO has been regarded as a promising optimization algorithm due to its simplicity, low computational cost and good performance [72].

Some of the active features of the PSO include ease of implementation and the fact that no gradient information is required. It can be used to solve a wide array of different optimization problems. It has been proven that PSO has simplicity, robustness against nonlinearity, non differentiability and high dimensionality. The **advantages** of PSO compared to other optimization techniques are as follow: easy implementation, few parameters to be adjusted and the fact that no gradient information is required [73]. In analysis, PSO has advantages and disadvantages.

5.8.1 Advantages of PSO:

Advantages of the basic particle swarm optimization algorithm: PSO is based on the intelligence. It can be applied into both scientific research and engineering use. Then PSO have no overlapping and mutation calculation. The search can be carried out by the speed of the particle. During the development of several generations, only the most optimist particle can transmit information onto the other particles, and the speed of the researching is very fast. After that the calculation in PSO is very simple. Compared with the other developing calculations, it occupies the bigger optimization ability and it can be completed easily. The last one is PSO adopts the real number code, and it is decided directly by the solution. The number of the dimension is equal to the constant of the solution [74].

5.8.2 Disadvantages of PSO:

Disadvantages of the basic particle swarm optimization algorithm are the method easily suffers from the partial optimism, which causes the less exact at the regulation of its speed and the direction. Then the method cannot work out the problems of scattering and optimization and the method cannot work out the problems of non-coordinate system, such as the solution to the energy field and the moving rules of the particles in the energy field [74].

5.8.3 Applications of PSO:

- One of the reasons that particle swarm optimization is attractive is that there are very few parameters to adjust. One version, with very slight variations (or none at all) works well in a wide variety of applications.
- Particle swarm optimization has been used for approaches that can be used across a wide range of applications, as well as for specific applications focused on a specific requirement. In this brief section, we cannot describe all of particle swarm's applications, or describe any single application in detail. Rather, we summarize a small sample.
- The first application represents an approach, or method, that can be used for many applications: evolving artificial neural networks. Particle swarm optimization is being used to evolve not only the network weights, but also the network structure [75] [62]. The method is so simple and efficient that we have almost completely ceased using traditional neural network training paradigms such as back

propagation. Instead, we evolve our networks using particle swarms. The approach is effective for any network architecture.

As an **example** of evolving neural networks, particle swarm optimization has been applied to the analysis of human tremor. The diagnosis of human tremor, including Parkinson's disease and essential tremor, is a very challenging area [76]. PSO has been used to evolve a neural network that distinguishes between normal subjects and those with tremor. Inputs to the network are normalized movement amplitudes obtained from an actigraph system. The method is fast and accurate.

As another example, end milling is a fundamental and commonly encountered metal removal operation in manufacturing environments. While development of computer numerically controlled machine tools has significantly improved productivity, the operation is far from optimized. None of the methods previously developed is sufficiently general to be applied in numerous situations with high accuracy. A new and successful approach involves using artificial neural networks for process simulation and PSO for multi-dimensional Optimization. The application was implemented using computer-aided design and computer-aided manufacturing (CAD/CAM) and other standard engineering development tools as the platform [77].

- Another application is the use of particle swarm optimization for reactive power and voltage control by a Japanese electric utility [78]. Here, particle swarm optimization was used to determine a control strategy with continuous and discrete control variables, resulting in a sort of hybrid binary and real-valued version of the algorithm. Voltage stability in the system was achieved using a continuation power flow technique.
- Particle swarm optimization has also been used in conjunction with a back propagation algorithm to train a neural network as a state-of-charge estimator for a battery pack for electric vehicle use. Determination of the battery pack state of charge is an important issue in the development of electric and hybrid/electric vehicle technology. The state of charge is basically the fuel gauge of an electric vehicle.
- A strategy was developed to train the neural network based on a combination of particle swarm optimization and the back propagation algorithm. One innovation was to use this combination to optimize the training data set. We can't say much

more about this, since the application is proprietary, but the results are significantly more accurate than those provided by any other method [79].

- Finally, one of the most exciting applications of PSO **is** that by a major American corporation to ingredient mix optimization. In this work, “ingredient mix” refers to the mixture of ingredients that are used to grow production strains of microorganisms that naturally secrete or manufacture something of interest. Here, PSO was used in parallel with traditional industrial optimization methods.
- PSO provided an optimized ingredient mix that provided over twice the fitness as the mix found using traditional methods, at a very different location in ingredient space. PSO was shown to be robust: the occurrence of an ingredient becoming contaminated hampered the search for a few iterations but in the end did not result in poor final results. PSO, by its nature, searched a much larger portion of the problem space than the traditional method.
- Generally speaking, particle swarm optimization, like the other evolutionary computation algorithms, can be applied to solve most optimization problems and problems that can be converted to optimization problems. Among the application areas with the most potential are system design, multi-objective optimization, classification, pattern recognition, biological system modeling, scheduling (planning), signal processing, games, robotic applications, decision making, simulation and identification. Examples include fuzzy controller design, job shop scheduling, real time robot path planning, image segmentation, EEG signal simulation, speaker verification, time-frequency analysis, modeling of the spread of antibiotic resistance, burn diagnosing, gesture recognition and automatic target detection, to name a few [80].

5.9 PSO algorithm:

Optimization algorithms are another area that has been receiving increased attention in the past few years by the research community as well as the industry [81] [82]. An optimization algorithm is a numerical method or algorithm for finding the maxima or the minima of a function operating with certain constraints [83]. As described by Eberhart and Kennedy, the PSO algorithm is an adaptive algorithm based on a social-psychological metaphor; a population of individuals (referred to as particles) adapts by returning stochastically toward previously successful regions [84]. It is one of the evolutionary computational optimization which is based on natural system developed in 1995 [68], through simulation of bird flocking in two-dimension space [85] [86].

A basic variant of the PSO algorithm works by having a population (called a swarm) of candidate solution (called particles). These particles are moved around in the search-space according to a few simple formulae. The movements of the particles are guided by their own best known position in the search-space as well as the entire swarm's best known position. When improved positions are being discovered these will then come to guide the movements of the swarm. The process is repeated and by doing so it is hoped, but not guaranteed, that a satisfactory solution will eventually be discovered. Here in this technique a set of particles are put in d-dimensional search space with randomly choosing velocity and position. The initial position of the particle is taken as the best position for the start and then the velocity of the particle is updated based on the experience of other particles of the swarming population.

Particle Swarm has two primary operators: Velocity update and Position update. During each generation each particle is accelerated toward the particles previous best position and the global best position. At each iteration a new velocity value for each particle is calculated based on its current velocity, the distance from its previous best position, and the distance from the global best position. The new velocity value is then used to calculate the next position of the particle in the search space. This process is then iterated a set number of times, or until a minimum error is achieved [87]. The detailed operation of particle swarm optimisation is given below [88]:

Step 1: Initialisation: The velocity and position of all particles are randomly set to within pre-defined ranges.

Step 2: Velocity Updating: At each iteration, the velocities of all particles are updated according to:

$$\vec{v}_i = w\vec{v}_i + c_1 R_1 \left(\vec{p}_{i,best} - \vec{p}_i\right) + c_2 R_2 \left(\vec{g}_{i,best} - \vec{p}_i\right) \quad .....(1)$$

Where $\vec{v}_i$ and $\vec{p}_i$ are the velocity and the position of particle i, respectively; $\vec{p}_{i,best}$ and $\vec{g}_{i,best}$ is the position with the best objective value found so far by particle *i* and the entire population respectively; w is a population controlling the flying dynamics; R1 and R2 are random variables in the range [0, 1]; c_1 and c_2 are factors controlling the related weighting of corresponding terms. The inclusion of random variables endows the PSO with the ability of stochastic searching. The weighting factors, c_1 and c_2, compromise the inevitable tradeoff between exploration and exploitation. After updating, $\vec{v}_i$ should be checked and secured within a pre-specified range to avoid violent random walking.

Step 3: Position Updating: Assuming a unit time interval between successive iterations, the positions of all particles are updated according to:

$$\vec{\boldsymbol{p}}_i = \vec{\boldsymbol{p}}_i + \vec{\boldsymbol{v}}_i \quad .....(2)$$

After updating, $\vec{p}_i$ should be checked and limited to the allowed range.

Step 4: Memory updating. Update $\vec{p}_{i,best}$ and $\vec{g}_{i,best}$ when condition is met.

$$\vec{\boldsymbol{p}}_{i,best} = \vec{\boldsymbol{p}}_i \quad \textbf{if} \quad f(\vec{\boldsymbol{p}}_i) > f(\vec{\boldsymbol{p}}_{i,best})$$

$$\vec{\boldsymbol{g}}_{i,best} = \vec{\boldsymbol{g}}_i \quad \textbf{if} \quad f(\vec{\boldsymbol{g}}_i) > f(\vec{\boldsymbol{g}}_{i,best})$$

where $f(\vec{x})$ is the objective function subject to maximization.

Step 5: Termination Checking. The algorithm repeats Steps 2 to 4 until certain termination conditions are met, such as a pre-defined number of iterations or a failure to make progress for a certain number of iterations. Once terminated, the algorithm reports the values of $\vec{p}_{i,best}$ and $\vec{g}_{i,best}$ as its solution.

Particles' velocities on each dimension are clamped to a maximum velocity *Vmax.* If the sum of accelerations would cause the velocity on that dimension to exceed *Vmax,* which is a parameter specified by the user, then the velocity on that dimension is limited to *Vmax.*

Vmax is therefore an important parameter. It determines the resolution, or fineness, with which regions between the present position and the target (best so far) position are searched. If *Vmax* is too high, particles might fly past good solutions. If *Vmax* is too small, on the other hand, particles may not explore sufficiently beyond locally good regions. In fact, they could become trapped in local optima, unable to move far enough to reach a better position in the problem space [80].

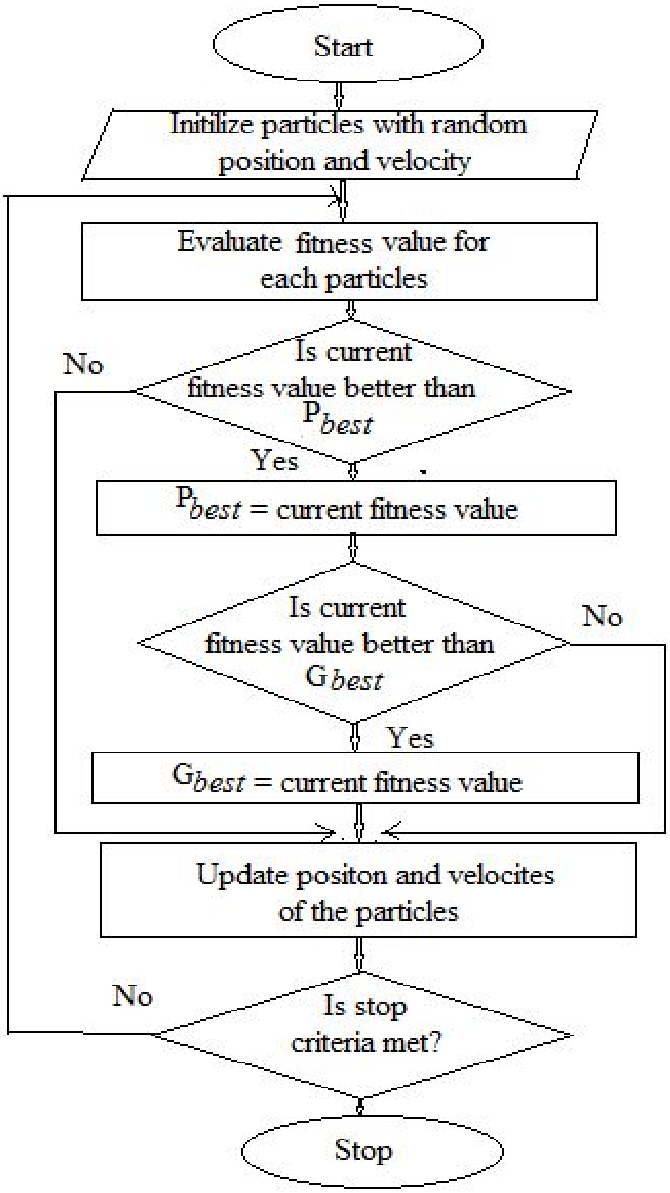


Figure 5.2: Flow chart of PSO algorithm

Figure 5.2 shows the flow chart of PSO algorithm. In PSO algorithm, the system is initialized with a population of random solutions, which are called particles, and each potential solution is also assigned a randomized velocity. PSO relies on the exchange of information between particles of the population called swarm. Each particle adjusts its trajectory towards its best solution (fitness) that is achieved so far. This value is called ***pbest***. Each particle also modifies its trajectory towards the best previous position attained by any member of its neighborhood. This value is called ***gbest***. Each particle moves in the search space with an adaptive velocity.

The fitness function evaluates the performance of particles to determine whether the best fitting solution is achieved. During the run, the fitness of the best individual improves over time and typically tends to stagnate towards the end of the run. Ideally, the stagnation of the process coincides with the successful discovery of the global optimum [89].

The acceleration constants c_1 and c_2 in equation **(1)** represent the weighting of the stochastic acceleration terms that pull each particle toward *pbest* and *gbest* positions. Thus, adjustment of these constants changes the amount of "tension" in the system. Low values allow particles to roam far from target regions before being tugged back, while high values result in abrupt movement toward, or past, target regions [80].

5.10 Factors affecting PSO performance:

The swarm's size and velocity, plus the behavior of the swarm influence the performance of the PSO process.

i) Swarm size and Velocity: The number of particles in the swarm significantly affects the run-time of the algorithm, thus a balance between variety (more particles) and speed (less particles) must be sought. Another important factor in the convergence speed of the algorithm is the maximum velocity parameter *(Vmax)*. This parameter limits the maximum jump that a particle can make in one step, thus a very large value for this parameter will result in oscillations [62]. On the other hand, a very small value could cause the particle to become trapped within local minima.

ii) Swarm Behaviour: The behavior of the swarm is dictated by the summation of the behaviors of individual particles. Each particle 'flies' in the direction of a better solution, weighted by some random factor, maybe overshooting, or potentially finding an individual or global better position. The interaction between the particles in the swarm helps to prevent straying off, whilst keeping close to the optimal solution [90]. Some particles will explore far beyond the current minimum, while the population still remembers the global best.

5.11 Attractive features of PSO based PID tuning:

In this study the PSO algorithm was used as an alternative to finding suitable tuning parameters for a variety of processes.

i) *Fast convergence*: The PSO is influenced by the simulation of social behaviour rather than the survival of the fittest as in the GA. Each individual benefits from its history and its interactions with other agents within the population. This sharing of knowledge helps facilitates faster convergence to an optimal solution.

ii) S*imple operating algorithm*: The use of simple mathematical operators facilitates a faster computational time and makes the algorithm suitable for determining tuning parameters under high-speed dynamical conditions for processes that lend themselves to tuning of this nature, such as flow and pressure control.

iii) *Efficient operating algorithm*: PSO parameters provide the yielded the best control performance – this is evident from the low ITAE that was observed during the tests.

iv) *Repeatability:* If we compare the PSO to the GA evolutionary algorithm. Tuning parameters obtained with the PSO are consistent over a number of tuning sessions. This does not apply to the GA based tuning method.

Chapter 6

Problem Formulation

6.1 Process description:

In this thesis work, the primary control objective is to control the product concentration of isothermal CSTR by varying the rate of dilution of the feed flow. The schematic diagram of the feedback control loop of isothermal CSTR is shown in figure 6.1.

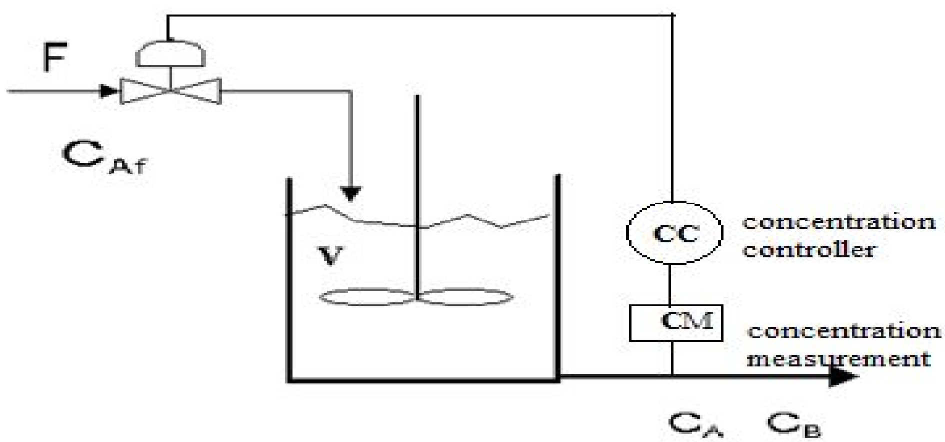


Figure 6.1: Feedback control mechanism for concentration control of isothermal CSTR

Here CM represents the measurement of concentration and CC represents the concentration controller.

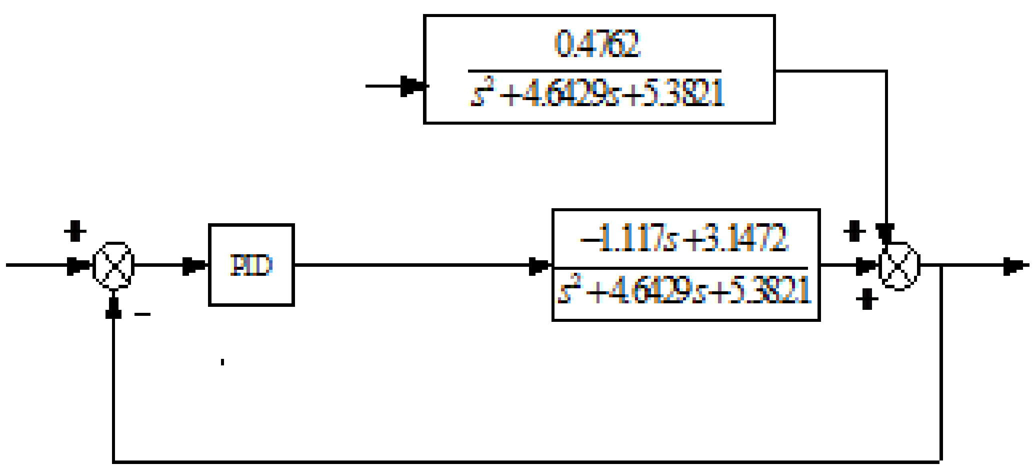


Figure 6.2: Transfer function based feedback control approach for concentration control of isothermal CSTR

Figure 6.2 shows the transfer function model of the feedback control scheme for concentration control of isothermal CSTR. The transfer function for process and the disturbance is derived in chapter 2.

The manipulated input-output process transfer function for the reactor is

$$g_p(s) = \frac{-1.117s + 3.1472}{s^2 + 4.6429s + 5.3821} \qquad \text{..... (15)}$$

With delay, transfer function is

$$g_p(s) = \frac{-1.117s + 3.1472e^{-0.5s}}{s^2 + 4.6429s + 5.3821} \qquad \text{..... (16)}$$

And the disturbance input-output transfer function is

$$g_d(s) = \frac{0.4762}{s^2 + 4.6429s + 5.3821} \qquad \text{..... (17)}$$

The transfer function poles (-2.23 and -2.4) are equal to the eigen values of the A matrix which is shown in Mathematical model of Isothermal CSTR in chapter 2.

The PID control block diagram is shown in figure 4.6 in chapter 3. The main objective of the controller is to control the concentration of the product liquid. So, PID controller is used. Ideal PID controller in continuous time is given as:

$$u(t) = K_c \left(e(t) + \frac{1}{\tau_i} \int_0^t e(t)dt + \tau_d \frac{de(t)}{dt} \right)$$

The PID controller is tuned using Zeigler-Nichols criteria of tuning. Simulink model [22] of PID controller is shown in the figure 6.3.

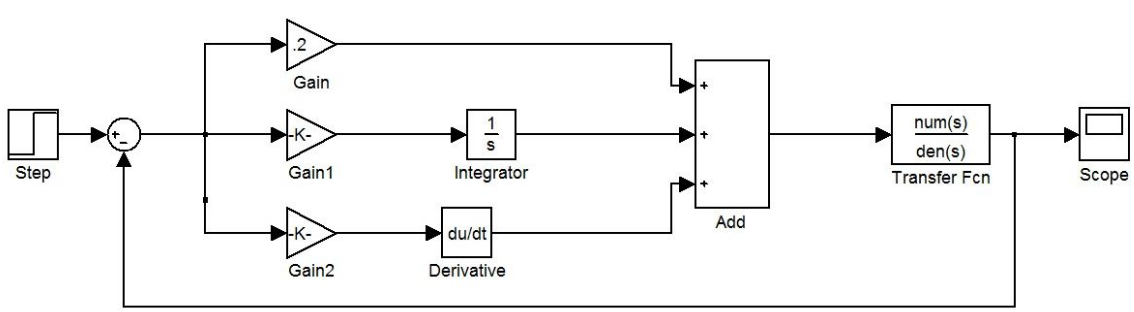


Figure 6.3: Simulation of PID controller

In this model, the values of proportional gain, integral gain and derivative gain of PID controller are 0.2, 0.95 and 0.23 respectively which are shown in table 6.3. The unit step response of feedback control is shown in figure 6.4.

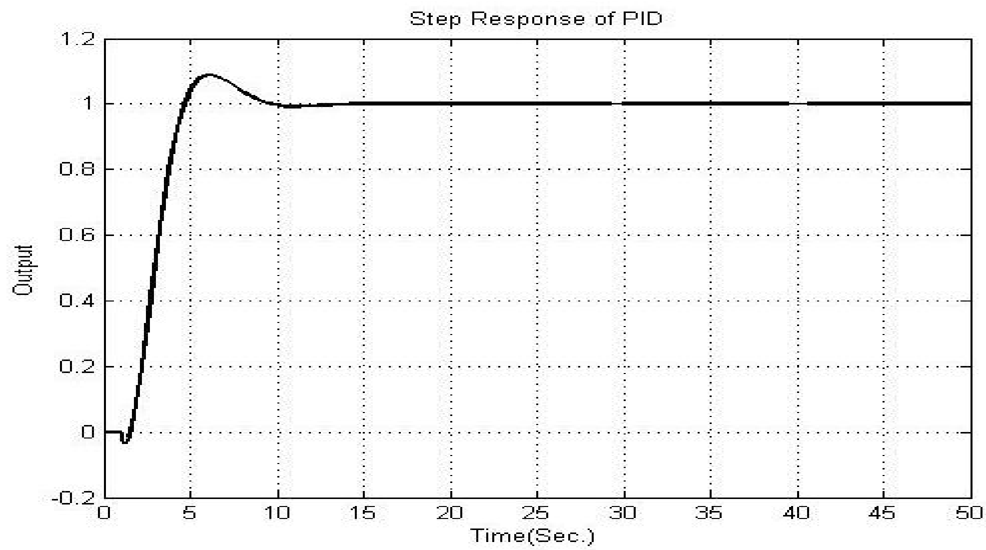


Figure 6.4: Unit step response of PID controller for concentration control

Here the overshoot is 9.12 %, settling time is 7.40 sec. peak time is 6.1 sec. etc. Which are shown in chapter 7, result and discussion.

We have seen that figure 6.4 represents the unit step response of PID controller. For good result, we compare the responses (such as overshoot, settling time, peak time etc.) of controllers. So we first simulate the model of PID controller with disturbance and PID controller with (distubance and delay). Simulink model of PID controller with disturbance is shown in figure 6.5.

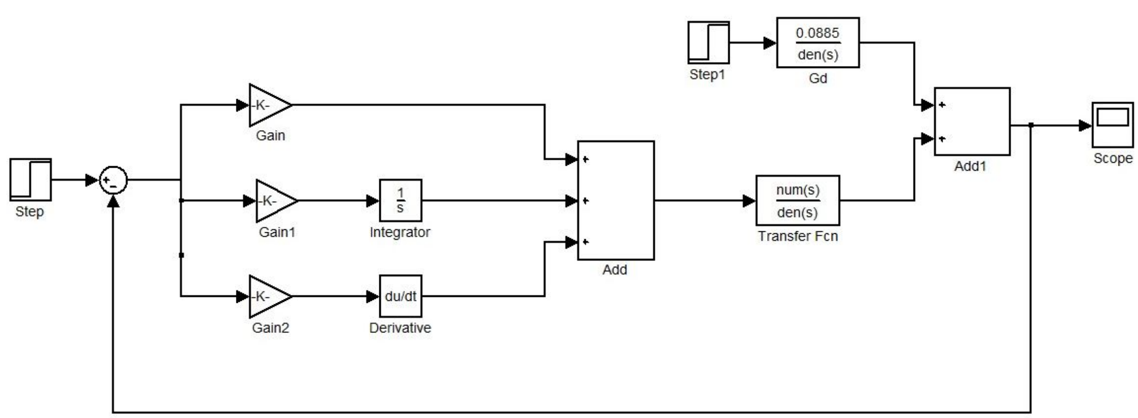


Figure 6.5: Simulation of PID controller with disturbance

Figure 6.6 shows the unit step response of feedback control scheme with a disturbance. The values of proportional gain, integral gain and derivative gain of PID controller are 0.001, 1.09 and 0.23 respectively which are shown in table 6.3.

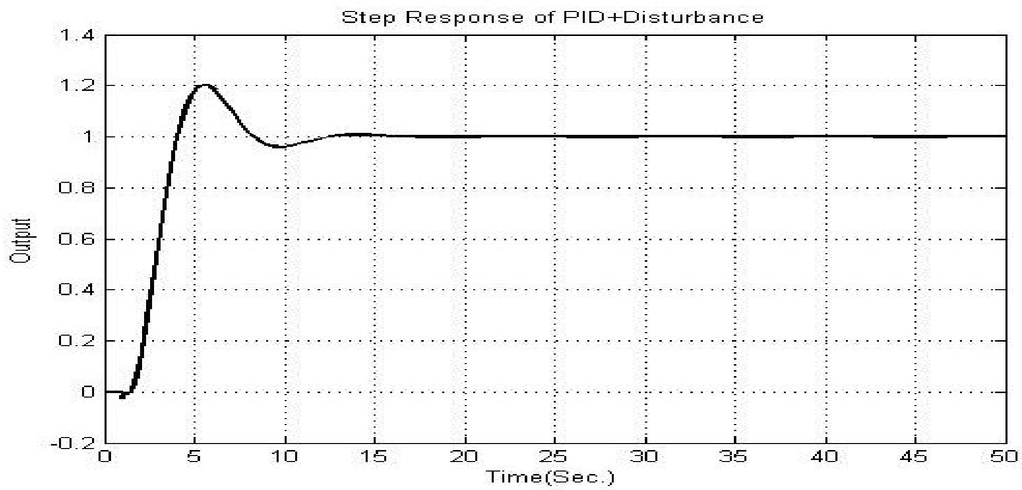


Figure 6.6: Unit step response of PID controller with disturbances

Due to this disturbance, the peak overshoot increases and settling time, peak time are decreasing here which is shown in chapter 7, result and discussion.

The Simulation of PID controller with (disturbance and delay) is shown in figure 6.7. The values of proportional gain, integral gain and derivative gain of PID controller are 0.001, 0.95 and 0.23 respectively which are shown in table 6.3.

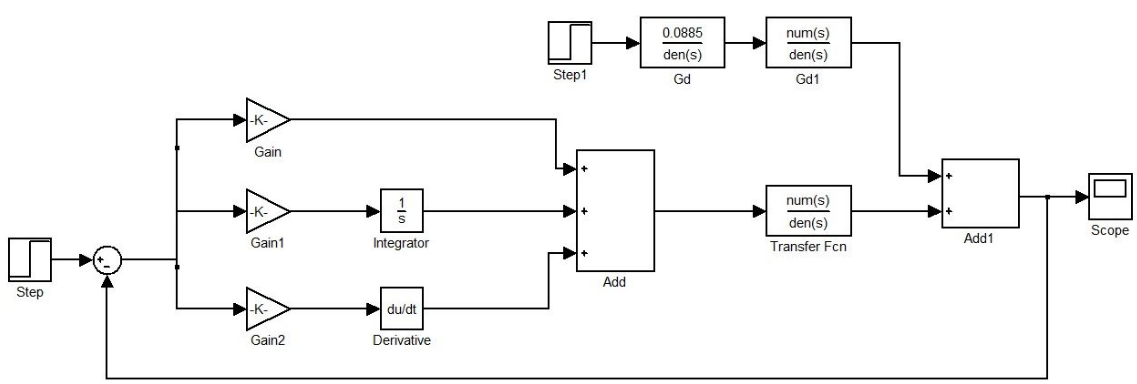


Figure 6.7: Simulation of PID controller with (disturbance and delay)

Figure 6.8 shows the unit step response of feedback control scheme with a (disturbance and delay).

Due to disturbance, peak overshoot increases but when we have delay with the disturbance, it decreases from the PID with disturbance.

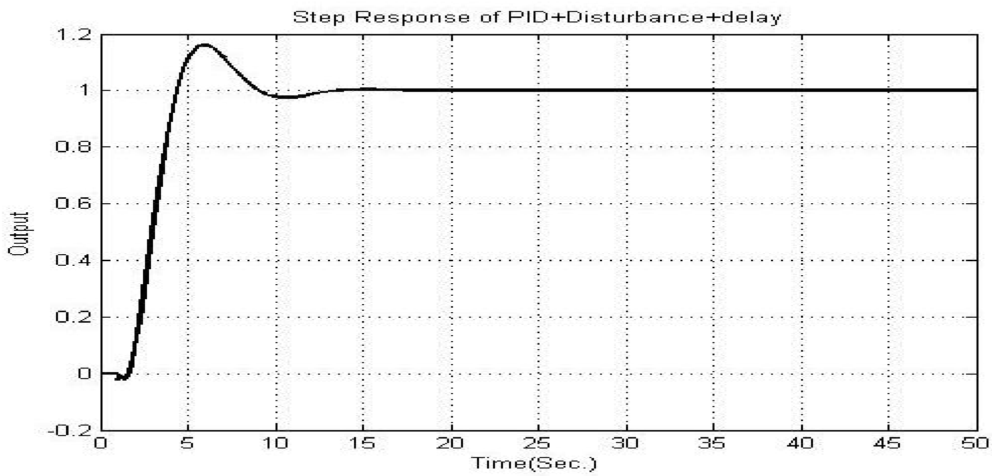


Figure 6.8: Unit step response of PID controller with delay and disturbance

In figure 6.6, we see that due to disturbance, peak overshoot increases but when we have delay with the disturbance, it decreases from the PID with disturbance, is shown in the figure 6.8. And if we talk about other responses, settling time and peak time increases from PID with disturbance.

For better result, we will compare all controllers with Fuzzy controllers. Now we take Fuzzy controller.

6.2 Fuzzy logic controller: Design:

PID controller is a standard control structure for classical control theory. But the performance is greatly distorted and the efficiency is reduced due to nonlinearity in the process plant. The fuzzy PID controllers are the natural extension of their conventional version, which preserve their linear structure of PID controller. The fuzzy PID controllers are designed using fuzzy logic control principle in order to obtain a new controller that possesses analytical formulas very similar to digital PID controllers. Fuzzy PID controllers have variable control gains in their linear structure. These variable gains are nonlinear function of the errors and changing rates of error signals. The main contribution of these variable gains in improving the control performance is that they are self- tuned gains and can adapt to rapid changes of the errors and rate of change of error caused by time delay effects, nonlinearities and uncertainties of the underlying process.

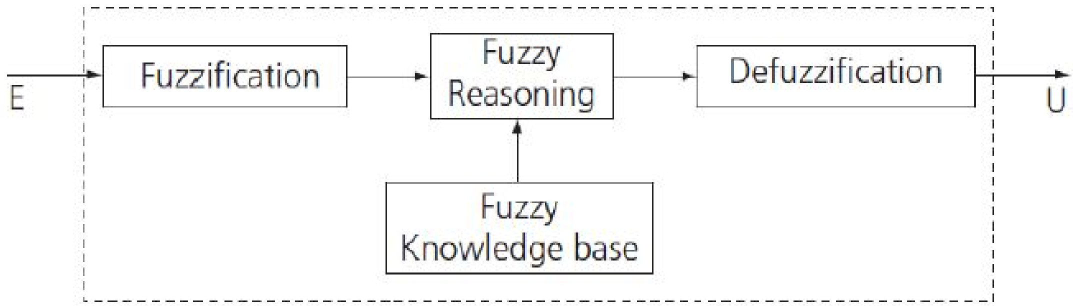


Figure 6.9: Architecture of fuzzy control

Where E is the input and U is the output. E input is first fuzzified using fuzzifier and then using knowledge base and rule base, output is derived and then defuzzification is done using defuzzifier.

6.2.1 Hybrid PD-fuzzy controller:

Although it is possible to design a fuzzy logic type of PID controller [24] by a simple modification of the conventional ones, via inserting some meaningful fuzzy logic IF- THEN rules into the control system, these approaches in general complicate the overall design and do not come up with new fuzzy PID controllers that capture the essential characteristics and nature of the conventional PID controllers. Besides, they generally do not have analytic formulas to use for control specification and stability analysis. The fuzzy PD, PI, and PI+D controllers to be introduced below are natural extensions of their conventional versions, which preserve the linear structures of the PID controllers, with simple and conventional analytical formulas as the final results of the design. Thus, they can directly replace the conventional PID controllers in any operating control systems (plants, processes).

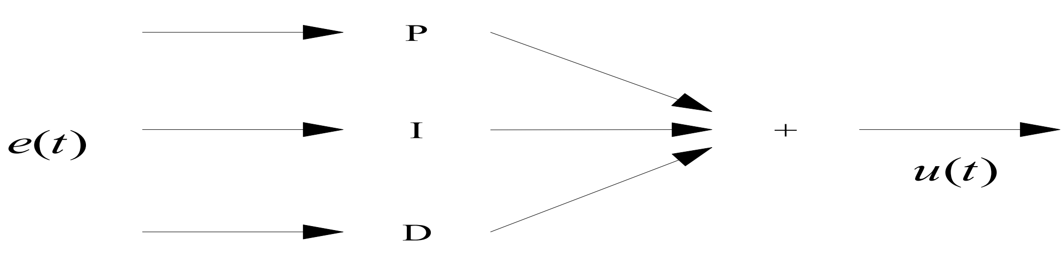


Figure 6.10: Parallel form of PID controller

The conventional design of PID controller was some what modified and a new hybrid fuzzy PID controller was designed. Instead of summation effect a mamdani based fuzzy inference system is implemented. The inputs to the mamdani based fuzzy inference system are error and change in error.

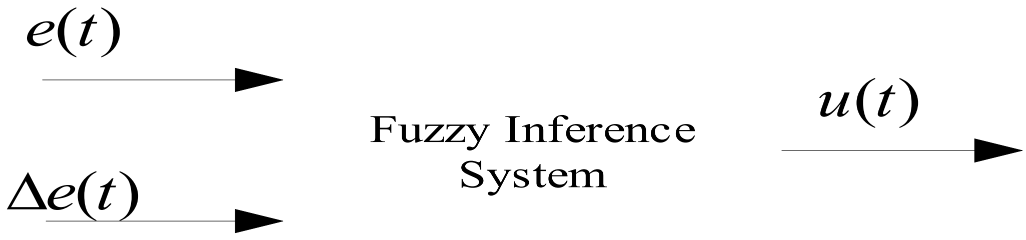


Figure 6.11: Fuzzy inference system

Figure 6.11 shows the fuzzy inference system developed for hybrid fuzzy controller. Figure 6.12 shows the structure of hybrid fuzzy logic controller, which keeps the general architecture of PID controller as shown with some slight modifications. A mamdani based fuzzy inference system is implemented in between proportional and derivative term. The integral term is then added to the output of fuzzy inference system.

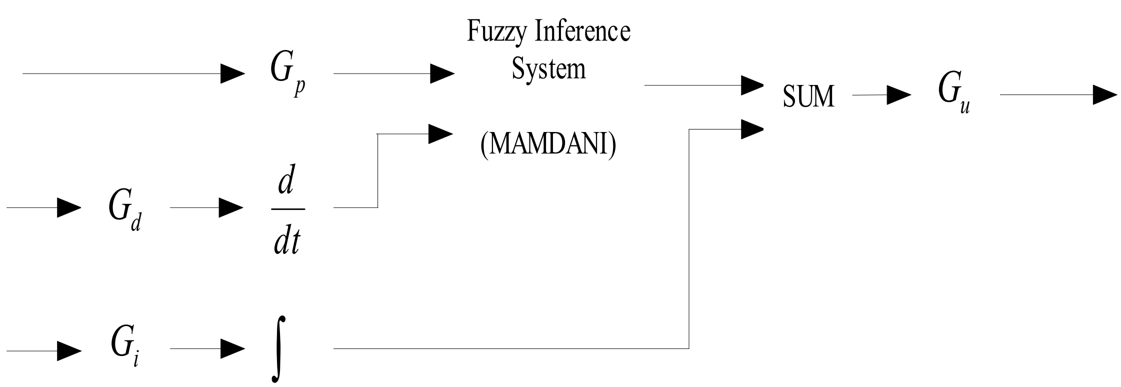


Figure 6.12: Architecture of proposed hybrid fuzzy controller

G_p, G_d and G_i are scaling factors for the input where as G_u is the scaling factor for the output. In this design the input and output scaling factors are determined by trial and error methods and are taken very small. Now we will prepare rule base for the fuzzy controller.

Table 6.1: Linguistic variable of fuzzy logic control

Error e(t)		Change in error Δe(t)		Controller output u(t)	
NB	Negative Big	NB	Negative Big	NB	Negative Big
NM	Negative Medium	NM	Negative Medium	NM	Negative Medium
NS	Negative Small	NS	Negative Small	NS	Negative Small
ZO	Zero	ZO	Zero	ZO	Zero
PS	Positive Small	PS	Positive Small	PS	Positive Small
PM	Positive Medium	PM	Positive Medium	PM	Positive Medium
PB	Positive Big	PB	Positive Big	PB	Positive Big

The above table 6.1 shows the linguistic variables for error, change in error and controller output. There are seven linguistic variables for error, change in error and controller output.

Table 6.2: IF-THEN rule base for fuzzy logic control

u(t)	**e(t)**							
		NB	**NM**	**NS**	**ZO**	**PS**	**PM**	**PB**
Δe(t)	**NB**	NB	NB	NB	NB	NM	NS	ZO
	NM	NB	NB	NB	NM	NS	ZO	PS
	NS	NB	NB	NM	NS	NS	PS	PS
	ZO	NB	NM	NS	ZO	ZO	PM	PM
	PS	NM	NS	ZO	PS	PS	PB	PB
	PM	NS	ZO	PS	PM	PM	PB	PB
	PB	ZO	PS	PM	PB	PB	PB	PB

The above table 6.2 shows the rule base of the fuzzy logic controller. Mamdani inference system [37] is used for developing rule base of the fuzzy logic controller. There are total fourty nine rules in this rule base. NB means negative big, NM means negative medium, NS means negative small,ZO means zero, PS means positive small, PM means positive medium and PB means positive big. The rules are shown in the above table as IF e(t) is NB and Δe(t) is PS THEN u(t) is NM.

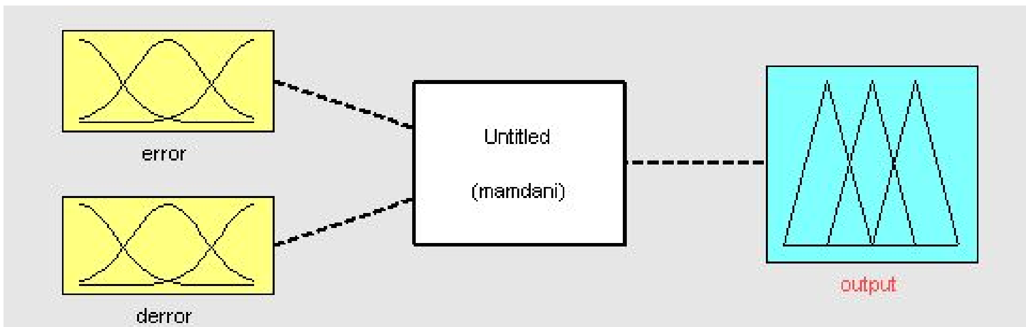


Figure 6.13: Mamdani fuzzy inference system developed for fuzzy controller

The above figure 6.13 shows the fuzzy inference system, in which there are two inputs as error and change in error and one output.

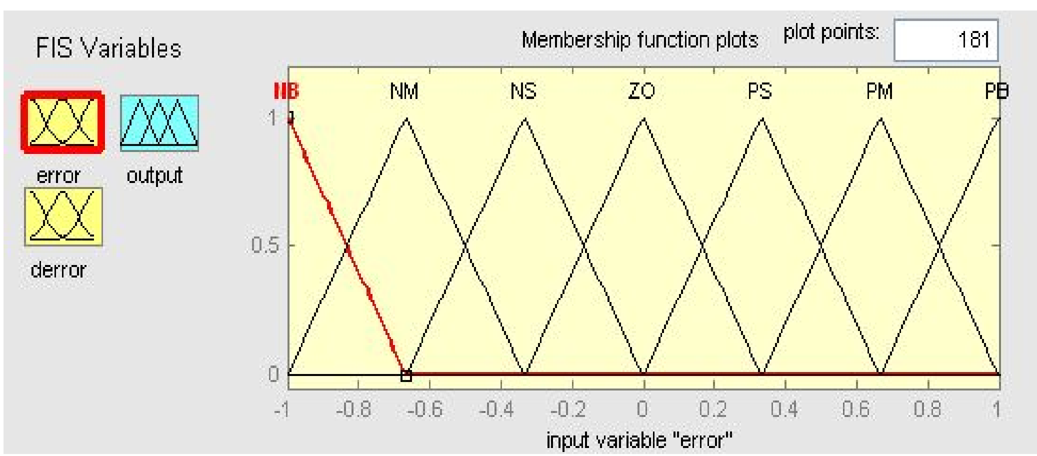


Figure 6.14: Membership function for input 1

The above figure 6.14 shows the membership function plots for the first input variable. Triangular membership function is taken for the input 1.

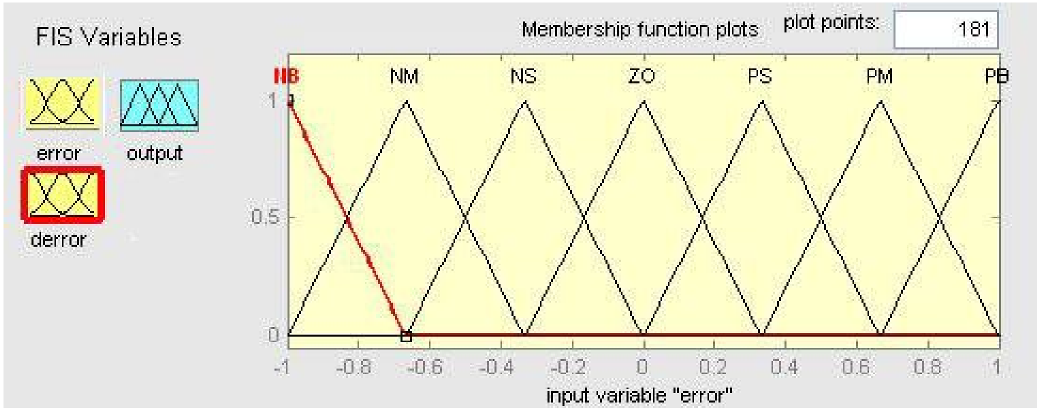


Figure 6.15: Membership function for input 2

The above figure 6.15 shows the membership function plots for the second input variable. Triangular membership function is taken for the input 2.

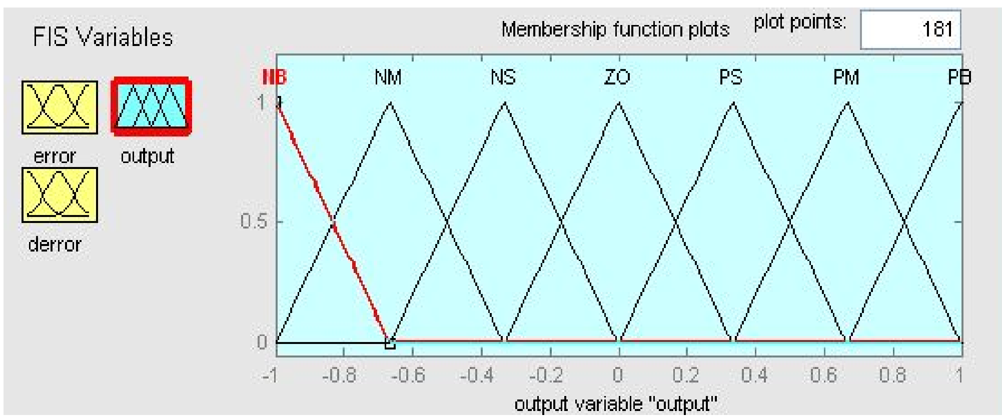


Figure 6.16: Membership function for output

The above figure 6.16 shows the membership function plots for the output variable. Triangular membership function is taken for the output.

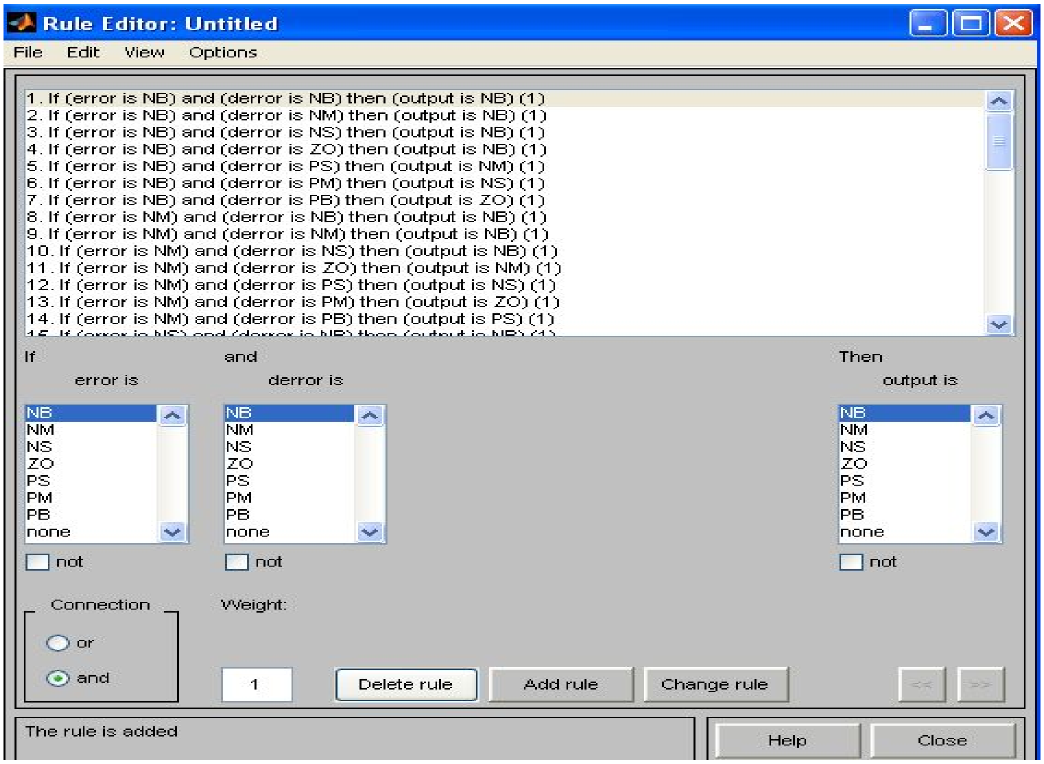


Figure 6.17: Rule base

The above figure 6.17 shows the rule base of the fuzzy logic controller. It consists of 49 rule base using If-and-then rule condition.

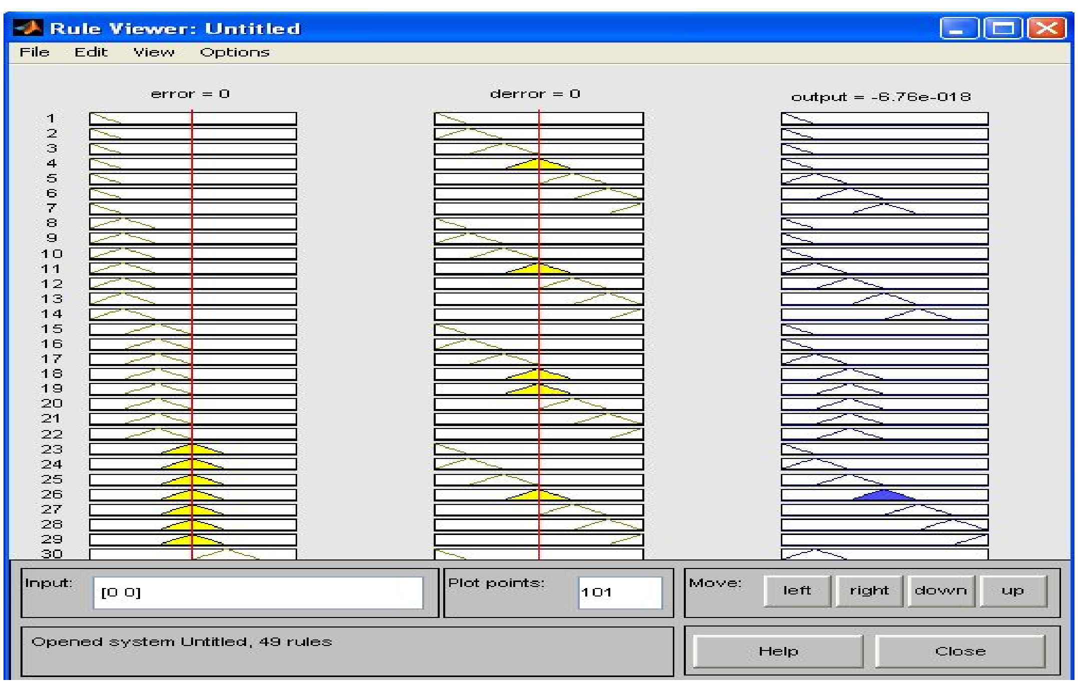


Figure 6.18: Rule viewer for fuzzy inference system

The above figure 6.18 shows the rule viewer for the fuzzy inference system. The rule viewer shows one calculation at a time and in great detail. If the entire output surface of system is to be viewed, that is, the entire span of the output set based on the entire span of the input set, the surface viewer is required. Now, we simulate the model [23] of Fuzzy controller.

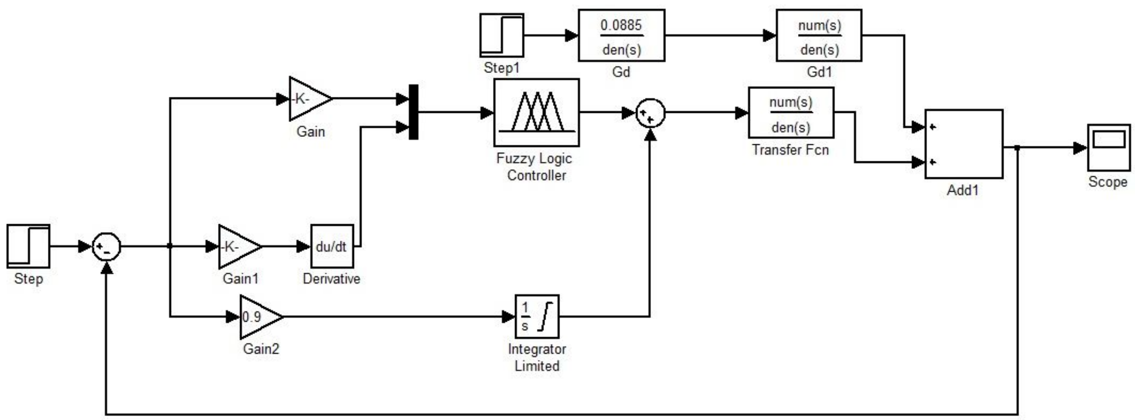


Figure 6.19: Simulink model of fuzzy controller

This model is shown in the figure 6.19. In this model, the values of proportional gain, integral gain and derivative gain of PID controller are 0.001, 0.9 and 0.001 respectively which are shown in table 6.3.

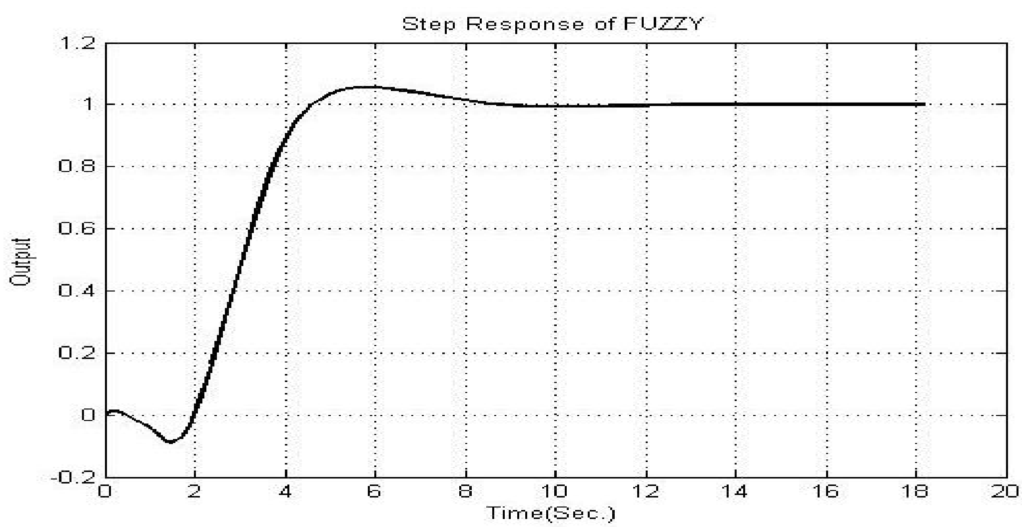


Figure 6.20: Unit step response of hybrid fuzzy controller for concentration control of isothermal CSTR

Figure 6.20 shows the unit step response of hybrid fuzzy controller for concentration control of isothermal CSTR. It is evident from the response that the overshoot is less as compared to the PID controllers, which shows the efficiency of fuzzy based controllers. Comparision of the controller's responses are shown in chapter 7.

Table 6.3: Gain values for the different controllers

Parameters/ Type	K_p	K_i	K_d
PID	.2	.95896	.239625
PID+Dis	.001	1.09586	.239625
PID+Dis+Delay	.001	.9586	.239625
Fuzzy	.001	.9	.001

For the best optimal solution, we used PSO (Partical swarm optimization). Now we will tune the PID controller and will optimize the parameter of PID controller using PSO. We will prepare the simulink model of PID controller for the different iteration M= 10, 15, 20, 25, 30, 40, 45 and will start simulatin then we will get unit step response for the different iterations M which shows peak overshoot, peak time, settling time etc. Then we will compare those responses. After comparision, we will choose best output results. Now we formulate this problem.

6.3 Algorithmic approach for the specified design:

In our case, we cast the PID controller design problem in PSO framework as given in figure 6.22. We consider the three dimensional search space. K_p , K_i and K_d are them three dimensions. We consider the fitness function based on time domain characteristics for adaptation. We set the number of adaptation iterations based on expected parameters and time of computation.

6.4 Concept of fitness function for the design:

For our case of design, we had to tune all the three parameters of PID such that it gives the best output results or in other words we have to optimize all the parameters of the PID for best results. Here we define a three dimensional search space in which all the three dimensions represent three different parameters of the PID. Each particular point in the search space represent a particular combination of $[K_p \ \ K_i \ \ K_d]$ for which a particular response is obtained.

The performance of the point or the combination of PID parameters is determined by a fitness function or the cost function. For the case of our design, we have taken four component functions to define fitness function. The fitness function is a function of steady state error, peak overshoot, rise time and settling time. However the contribution of these component functions towards the original fitness function is determined by a scale factor that depends upon the choice of the designer. For this design the best point is the point where the fitness function has the minimal value.

The chosen fitness function is:-

F = (1-exp(-β)) (Mp +ESS) + (exp(-β))(Ts - Tr)

Where F: Fitness function

Mp: Peak Overshoot

Ts: Settling Time

Tr: Rise Time

β: Scaling Factor(Depends upon the choice of designer) For our case of design we have taken the scaling factor β = 1.

In the matlab library we have defined a fitness function. It has the format:-

Function [F] = fitness ($\boldsymbol{K_d} \ \boldsymbol{K_P} \ \ \boldsymbol{K_i}$)

which has PID parameters as input values and it returns the fitness value of the PID based controlled model as its output.

6.5 PID controller tuning using PSO:

Figure 6.21 shows the structure of PID controller optimization process.

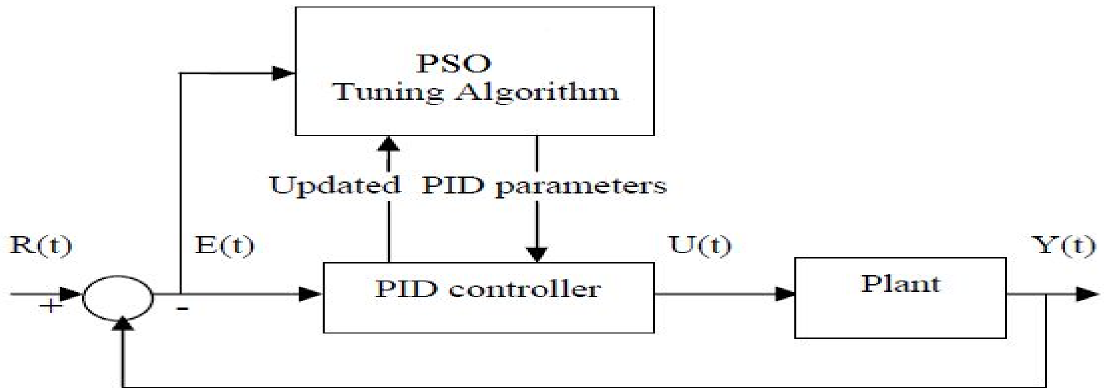


Figure 6.21: Block diagram of PID controller tuning

We propose PSO to tune value of three parameters repeatedly until they achieve an acceptance level of performance [18].

6.6 PSO based simulation and results:

In our simulations using PSO algorithm, we have varied the number of iterations. We present a comparative study of the performance of the initial global best position out of randomly initialized swarm particles to the performance of the final global best position which comes after the application of "particle swarm optimization" algorithm.

6.6.1 Simulation results for different number of iterations:

Simulink model of PID controller for iteration M=25 of PSO is shown in figure 6.22. In this model, the values of proportional gain, integral gain and derivative gain of PID controller are 0.30, 0.85 and 0.29 respectively. And the value of f_{best} and elapsed time are 0.1621 and 157.97 respectively for this iteration.

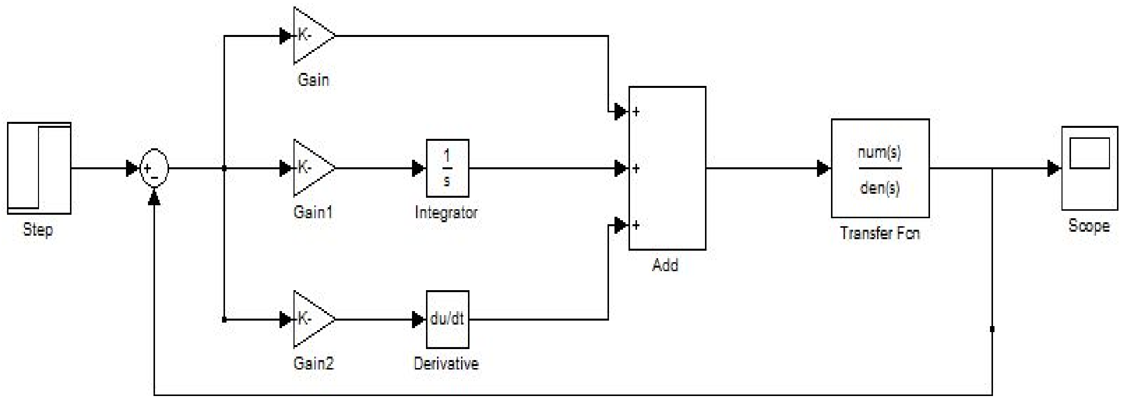
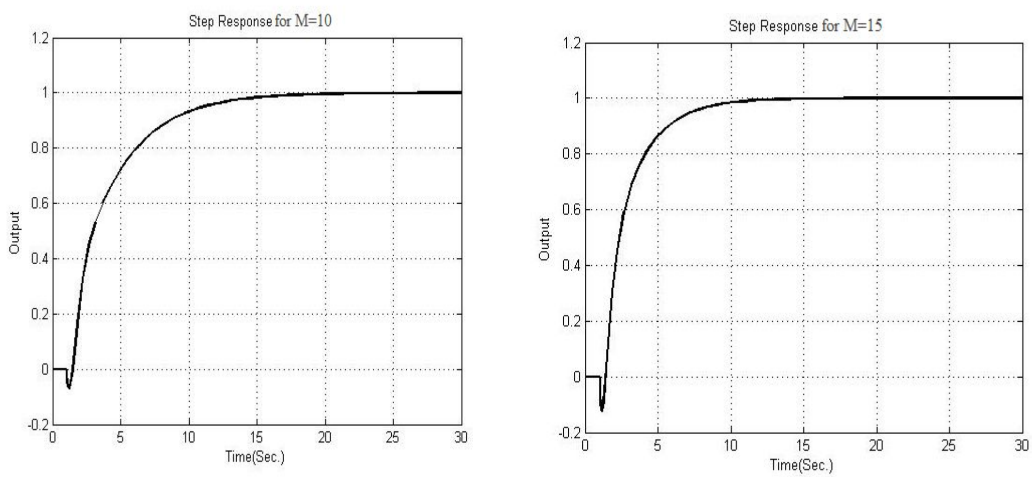


Figure 6.22: Simulink model of PID controller for iteration M=25 of PSO

Thus we make the simulink model of the PID controller for the different iterations such as M= 15, 20, 25, 30, 40, 45 and for tuning, we vary the values of the proportional gain, integral gain and derivative gain. Unit step response of feedback control scheme for these all iterations are shown in the figure 6.23.



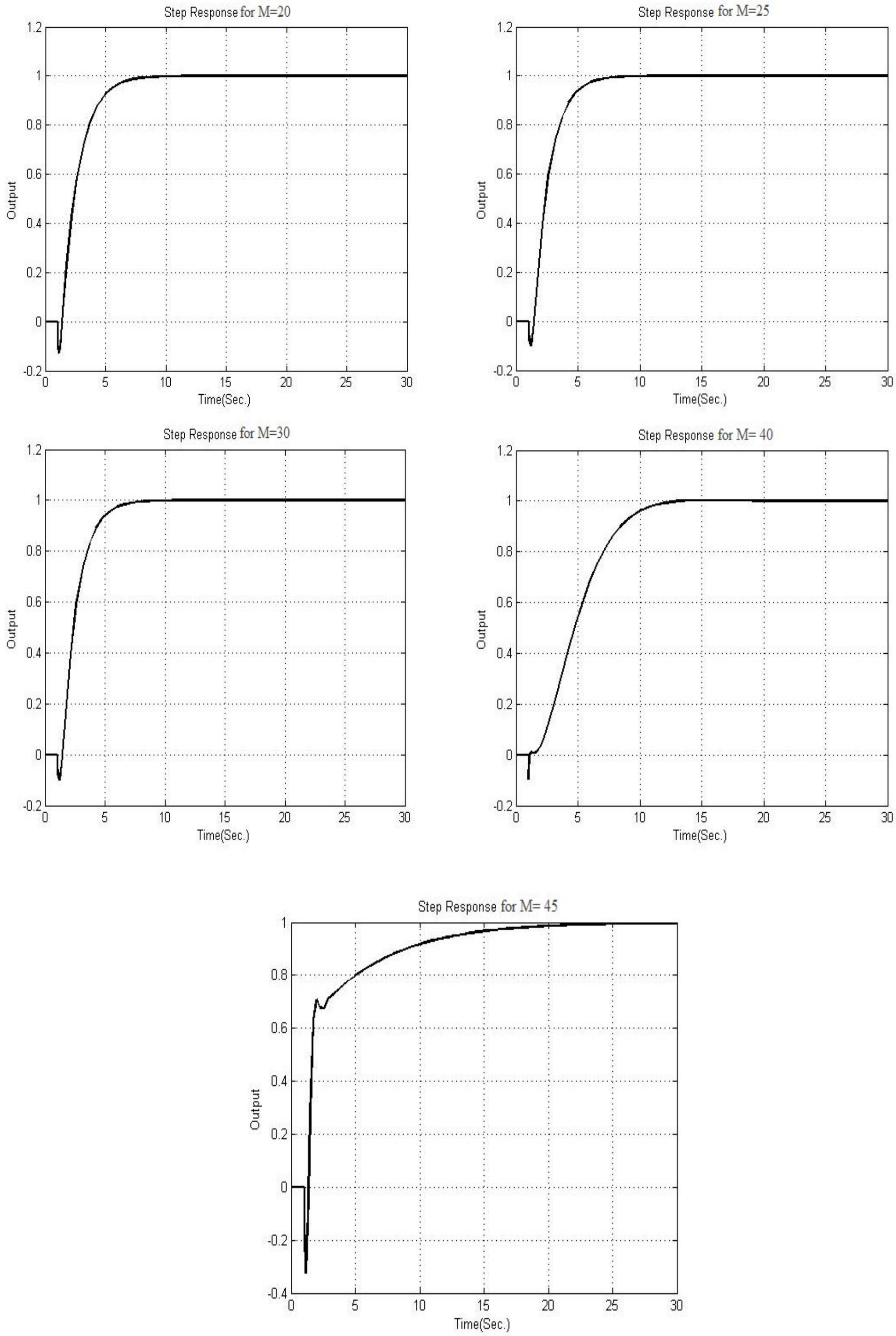


Figure 6.23: Step responses of PID controller for different no. of the iterations such as M=10, 15, 20, 25, 30, 40, 45 of PSO.

In figure 6.23, Each unit response of different iteration shows peak overshoot, settling time, peak time etc. If we compare these responses, we get the best output result. Comparison of these unit step response are shown in chapter 7.

Chapter 7

Results & Discussion

7.1 Step response analysis:

Comparison of the step response of different controllers, are shown in the figure 7.1.

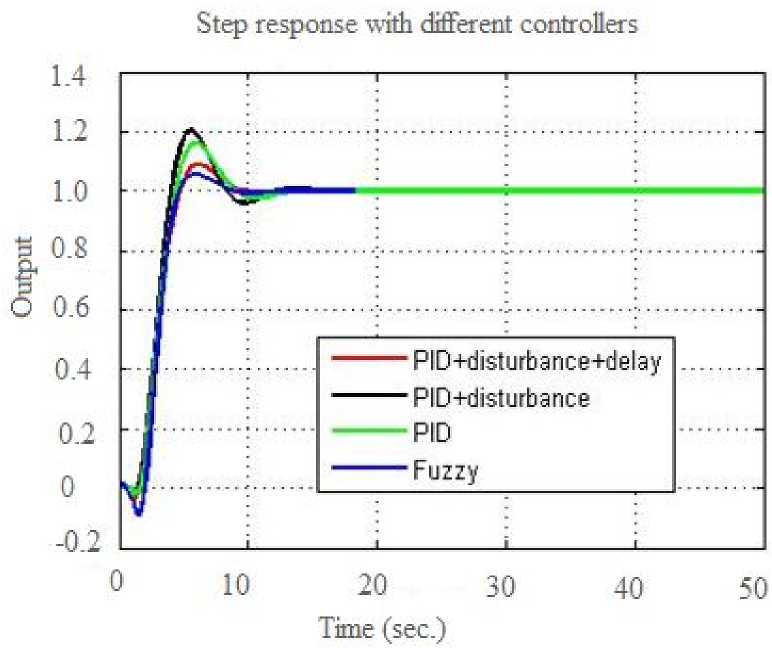


Figure 7.1: Comparison of the unit step response of different controllers

Transient response such as peak overshoot, rise time, delay time, settling time and peak time, of the controllers are shown in the table 7.1.

Table 7.1: Transient response of controllers

Parameters/ Type	Peak overshoot	Rise time	Delay time	Settling time	Peak time
PID	9.1251	4.679	3.0459	9.4062	6.1049
PID+ Disturbance	20.3425	4.365	2.956	13.098	5.6760
PID+ Dist.+ Delay	16.0946	4.379	2.982	12.519	6.0825

Fuzzy	5.5092	4.580	3.115	8.5306	6.0402

In these controllers response, we see that best output result come for Fuzzy controller. In this controller, peak overshoot is less than the other controllers. Table 7.2 shows the performance criteria of the different controllers.

Table 7.2: Performance criteria of controllers

Parameters/ Type	**Integral Absolute Error (IAE)**	**Integral Square Error (ISE)**	**Integral Time Absolute Error (ITAE)**
PID	2.279	1.579	6.226
PID+ Disturbance	2.49	1.552	8.2
PID + Dist. + Delay	2.528	1.661	7.974
Fuzzy	2.242	1.472	5.702

7.2 PID Controller Tuning Using PSO:

Figure 7.2 shows comparison of the step response of PID controller with different no. of the iterations M for PSO.

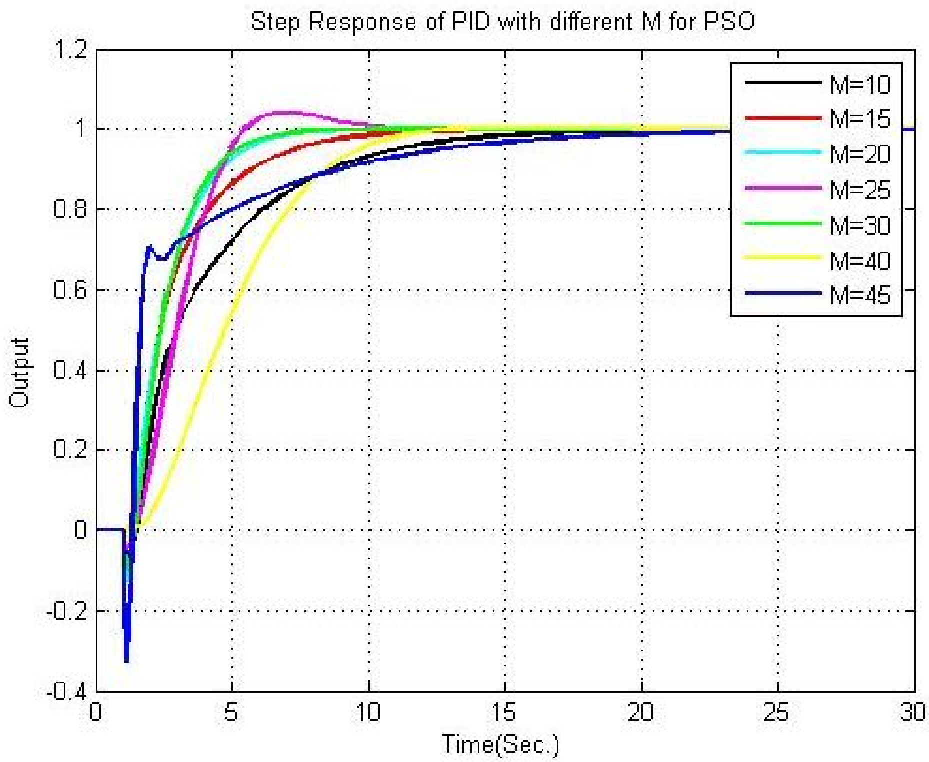


Figure 7.2: Comparison of step response of PID controller with different no. of the iterations M of PSO

Gain values and Fitness function for the different iteration of PSO are shown in table 7.3.

Table 7.3: Gain values and Fitness function for the different iteration of PSO

M (No. Of Iteration)	K_p	K_i	K_d	**Optimal best fitness function** f_{best}	**Elapsed Time 't' (sec.)**
10	.6451	.5179	.0368	.1967	53.614
15	.9964	.7949	.2895	.1902	103.454
20	.9402	.9356	.3960	.1836	269.127
25	.3065	.8568	.2970	.1621	157.971
30	.8033	.9580	.2073	.1589	175.742

40	.0035	.4130	.4628	.1324	249.522
45	2.2519	.6329	.3927	.0984	1106.322

Table 7.4 shows the performance criteria of the different iteration of PSO.

Table 7.4: Performance criteria for different iteration of PSO

M (No. Of Iteration)	Integral Absolute Error (IAE)	Integral Square Error (ISE)	Integral Time Absolute Error (ITAE)
10	3.301	1.739	14.2
15	2.151	1.234	6.502
20	1.828	1.178	4.368
25	1.547	1.168	3.849
30	1.785	1.201	4.086
40	4.171	2.866	15.75
45	2.689	1.085	15.26

Table 7.5 shows the Comparative transient response such as peak overshoot, rise time, delay time, settling time and peak time, of PID controller for different no. of the iterations.

Table 7.5: Transient response for different iteration of PSO

M (No. Of Iteration)	Peak overshoot	Rise time	Delay time	Settling time	Peak time
10	-6.7717	15.023	2.936	10	10
15	-1.6014	10.319	2.682	7.275	10

20	-0.1857	7.393	2.581	5.466	10
25	4.1394	7.259	2.359	4.807	6.857
30	-0.1117	7.212	2.312	5.203	10
40	-4.0377	12.193	4.810	9.624	10
45	-3.2679	21.901	1.892	10	10

When we see the table 7.5, we can say that step response for iteration M=25, PSO have the best output results.

Chapter 8

Conclusion and future scope

Process control is one of the research area, which demands design of an optimal controller. There are many controller architecture available in control literature, but PID controller is one of the most versatile and widely used controller. More than 90% of the process plants use this as controlling element.

This dissertation considers an isothermal CSTR and models it using state space technique. The primary goal of the controller is to control the product concentration of isothermal CSTR while manipulating the input flow rate. This dissertation designs a PID controller, a fuzzy controller and tunes the PID controller using particle swarm optimization technique. The gain values and transient responses obtained through PID controller using particle swarm optimization, provides best output results than other controller such as PID controller, PID controller with disturbance, PID controller with (disturbance and delay) and fuzzy based controller.

In the future, the following work may be carried out. Hybrid Neuro Fuzzy (HNF) approach can be used to improve the performance of PSO based controller. Tabu Search (TS) algorithm can be used to optimize PSO based controller. In the present work, I have used simulation to show superiority of the algorithm. Further works may be done to apply it to real system.

References

[1] Levenspiel O., “The Chemical Reactor Omni book, Oregon St Univ Bookstores”, 1993.

[2] Fogler H., S., “Elementos de Ingeniería de las Reacciones Químicas”, Prentice-Hall, 2001.

[3] Regalado-Méndez Alejandro, Cid-Rodríguez Ma. Del Rosario P, And Báez-González Juan G., “Problem Based Learning (PBL): Analysis Of Continuous Stirred Tank Chemical Reactors With A Process Control Approach”, Industries Institute, Universidad Del Mar, Puerto Angel, Oaxaca, Mexico, October 2010.

[4] Dr. Sanju Nanda, “Reactors and Fundamentals of Reactors Design for Chemical Reaction”, Dept. of Pharmaceutical Sciences, M.D. University, Rohtak, Haryana (24-01-2008).

[5] Schmidt, Lanny D., “The Engineering of Chemical Reactions”, New York: Oxford University Press, 1998.

[6] A. Favache and D. Dochain, “Thermodynamics and chemical systems stability: The CSTR case study revisited,” Journal of Process Control, vol. 19, pp. 371–379, 2009.

[7] R. Ortega, D. Jeltsema, and J. Scherpen, “Power shaping: A new paradigm for stabilization of nonlinear RLC-circuits,” IEEE Transactions on Automatic Control, vol. 48, no. 10, pp. 1762–1767, october 2003.

[8] R. Ortega, A. Van Der Schaft, I. Mareels, and B. Maschke, “Putting energy back in control,” IEEE Control Systems Magazine, vol. 21, no. 2, pp. 18–33, April 2001.

[9] D. Jeltsema, R. Ortega, and J. M.A. Scherpen, “An energy-balancing perspective of interconnection and damping assignment control of nonlinear systems,” Automatica, vol. 40, no. 9, pp. 1643–1646, Sept., 2004.

[10] R. Ortega, A. van der Schaft, B. Maschke, and G. Escobar, “Energyshaping of port-controlled Hamiltonian systems by interconnection,” in Proc. 38th IEEE Conference on Decision and Control, Phoenix (USA), December 1999, pp. 1646–1651.

[11] B. Maschke, R. Ortega, and A. van der Schaft, “Energy-based Lyapunov functions for forced Hamiltonian systems with dissipation,” IEEE Transactions on Automatic

Control, vol. 45, no. 8, pp. 1498– 1502, 2000.

[12] R. Ortega, A. Van Der Schaft, B. Maschke, And G. Escobar, “Interconnection And Damping Assignment: Passivity-Based Control Of Port controlled Hamiltonian Systems,” Automatica, Vol. 38, pp. 585–596, 2002.

[13] A. Alonso, B. Ydstie, and J. Banga, “From irreversible thermodynamics to a robust control theory for distributed process systems,” Journal of Process Control, vol. 12, pp. 507–517, 2002.

[14] B. Wayne Bequette, “Process Control Modeling, Design, and Simulation” Rensselaer Polytechnic Institute, Publication- PHI Learning Private Limited, New Delhi , 2010.

[15] Michael A. Johnson, “PID Control: New Identification and Design Methods” University of Strathclyde, UK, 1948.

[16] DALE E. SEBORG, THOMAS F. EDGAR, DUNCAN A. MELLICHAMP, “Process Dynamics and Control” Second Edition, WILEY INDIA EDITION, 2004.

[17] B. C. Kuo, Automatic Control System. Englewood Cliffs, N.J.: Prentice Hall, 1995.

[18] Amir Hossein Fathi, Hamid Khaloozadeh, Mohammad Ali Nekoui, Reza Shisheie, “Using PSO and GA for Optimization of PID Parameters”, International Journal of Intelligent Information Processing (IJIIP) Volume3. Number1. March 2012.

[19] Bennett, Stuart, “A history of control engineering”, 1930-1955. IET. p. p. 48. ISBN 978-0-86341-299-8, 1993.

[20] Dr M.J. Willis, “Proportional-Integral-Derivative Control”, Dept. of Chemical and Process Engineering University of Newcastle, 17th November, 1998.

[21] Kiam Heong Ang, Gregory Chong and Yun Li, “PID control system analysis, design and technology,” IEEE Transactions on Control System Technology, vol. 13, no. 4, pp. 559-576, July 2005,

[22] Nina F Thornhill, Sachin C Patwardhan, Sirish L Shah, “A continuous stirred tank heater simulation model with applications,” Journal of Process Control, 18, pp. 347 – 360, 2008.

[23] R M Suja Malar, T Thyagarajan, “Modeling of continuous stirred tank reactor using artificial intelligence techniques,” International Journal of Modeling, 8, 3, pp. 145 – 155, 2009.

[24] Sufian Ashraf Mazhari and Surendra Kumar, “Heuristic search algorithm for tuning PUMA 560 fuzzy PID controller,” International Journal of Computer Science, vol. 3, no. 4, pp. 277-286, 2008.

[25] R. C. Moore, “The role of logic in knowledge representation and commonsense reasoning,” in Proc. Nar. Con$ Artificial Intell., pp. 428-433, 1982.

[26] H. J. Levesque and R. Brachman, “Expressiveness and tractability in knowledge representation and reasoning,” Comput. Intell., vol. 3, pp. 78-93, 1987.

[27] C. V. Negoita, Expert Systems and Fuzzy Systems. Menlo Park, CA: Benjamin/Cummings, 1985.

[28] L. A. Zadeh, “A theory of commonsense knowledge,” in Aspects of Yagueness, H. J. Skala, S . Termini, and E. Trillas, Eds. Dordrecht, The Netherlands: Reidel, 1984.

[29] L. A. Zadeh, “Test-score semantics as a basis for a computational approach to the representation of meaning,” Literary Ling. Comput., vol. 1, pp. 24-35, 1986.

[30] L. A. Zadeh, “The role of fuzzy logic in the management of uncertainty in expert systems,” Fuzzy Sets Syst., vol. 11, pp. 199-227, 1983.

[31] L. A. Zadeh, “Outline of a new approach to the analysis of complex systems and decision processes,” IEEE Trans. Syst., Man, Cybern., vol. SMC-3, pp. 28-44, 1973.

[32] L. A. Zadeh, “A fuzzy-set-theoretic approach to fuzzy quantifiers in natural languages,” Comput. Math., vol. 9, pp. 149-184, 1983.

[33] A. Kaufmann and M. M. Gupta, Introduction to Fuzzy Arithmetic. New York: Van Nostrand, 1985.

[34] L. A. Zadeh, “Probability measures of fuzzy events,” J. Math. Anal. Appl., vol. 23, pp. 421-427, 1968.

[35] D. Dubois and H. Prade, Fuzzy Sets and Systems: Theory and Applications. New

York: Academic, 1980.

[36] Zadeh, L.A., A computational approach to fuzzy quantifiers in natural languages, Computers and Mathematics 9, pp. 149-184, 1983.

[37] Mamdani, E.H. and Gaines, B.R., Fuzzy Reasoning and its Applications. London: Academic Press, 1981.

[38] Zadeh, L.A., Fuzzy logic and approximate reasoning (in memory of Grigore Moisil), Synthese 30, pp 407-428, 1975.

[39] R. J. Sternberg, editor. “Handbook of Intelligence. Cambridge University Press”, 2000.

[40] W. V. Bingham. Aptitudes and aptitude testing. Harper & Brothers, New York, 1937.

[41] L. S. Gottfredson. Mainstream science on intelligence: An editorial with 52 signatories, history, and bibliography. Intelligence, 24(1):13–23, 1997.

[42] AAP-6, NATO Glossary of terms and definitions, 2004.

[43] Lewis M. Terman and Edward L. Thorndike, “Human intelligence” Ohio: university of Akron, 1921.

[44] Steve Pavlina, “10 weakness of Human Intelligence” June 25th, 2007.

[45] Dennis J. Reimer, “Human Intelligence Collector Operations” Headquarters, Department Of The Army, Washington, DC, 6 September 2006.

[46] Omkar Pathak, “Applications of Artificial Intelligence” 20 September, 2011.

[47] Houghton Mifflin Company, American Heritage® Science Dictionary, 2005.

[48] Stuart Russell and Peter Norvig, Artificial Intelligence A modern Approach, by Pearson Education, Inc., Upper Saddle River, New Jersey 07458, 2010.

[49] John Mccarthy, “What Is Artificial Intelligence?”,Computer Science Department, Stanford University, Revised November 12, 2007

[50] John H. Saunders, “A Primer on Artificial Intelligence Technologies”, Compiled by the Intelligent Decision Systems Course Team June 27, 2000.

[51] John H. Saunders, “A Primer on Artificial Intelligence Technologies”, AI experts: ceased publishing in 21 July 1999.

[52] Turing, A. M., "Computing Machinery and Intelligence", Mind 59/433-460, 1950.

[53] Shieber, S., "The First Turing Test". Proceedings of the ACM, 1994.

[54] Cummins, Robert, "Functional Analysis", Journal of Philosophy 72/741-765, 1975. Partially reprinted in Block (1980).

[55] Alan R MCNAMARA, Laurie G LOCK LEE, Kee Chye TEH and Hok Min LIE, “ intelligent Operator Guidance Systems Beyond the Prototype”, BHP Central Research Laboratories PO Box 188, Wallsend, 2287 NSW, Australia.

[56] Scianta Intelligence, “Application of Machine Intelligence”, 2005.

[57] Christophe Cop, “Artificial Intelligence will supersede Human intelligence”, Antwerpen Belgium, Apr 4, 2011.

[58] Anna Hoffmann , “Artificial Intelligence will supersede Human intelligence”, 8 april, 2011.

[59] Moshe Y. Vardi , “Artificial Intelligence: Past and Future”, Communications of the ACM, Vol. 55 No. 1, Page 5, 10.1145/2063176.2063177.

[60] J.C. Bezdek, “Editorial: Fuzzy Models - What Are They, and Why?”, IEEE Transactions on Fuzzy Systems, Vol. 1, No. 1, February 1993.

[61] C.C. Lee, “Fuzzy Logic in Control Systems: Fuzzy Logic Controllers,” (parts I and II), IEEE Transactions on Systems, Man, and Cybernetics, Vol. 20, 1990.

[62] Kennedy, J., Eberhart, R. C., and Shi, Y., Swarm Intelligence, San Francisco: Morgan Kaufmann Publishers, 2001.

[63] A. Colorni, M. Dorigo et V. Maniezzo, Distributed Optimization by Ant Colonies, actes de la première conférence européenne sur la vie artificielle, Paris, France, Elsevier Publishing, 134-142, 1991.

[64] A. Colorni, M. Dorigo, and V. Maniezzo, Distributed optimization by ant colonies, Proceedings of ECAL'91, European Conference on Artificial Life, Elsevier Publishing,

Amsterdam, 1991.

[65] T.A. Feo and M.G.C. Resende, Greedy randomized adaptive search procedures, Journal of Global Optimization 6, pp. 109-133, 1995.

[66] J.H. Holland, Adaptation in natural and artificial systems, University of Michigan Press, 1975.

[67] J.Y. Potvin and S. Bengio, The vehicle routing problem with time windows - part {II: genetic search, Informs Journal of Computing 8, pp. 165-172, 1996.

[68] J.Kennedy and R.Eberhart, "Particle Swarm Optimization", In Proceeding IEEE Int. Conf. Neural Networks, vol. 4, pp. 1942-1947, 1995.

[69] Thomas Beielstein , K.E. Parsopoulos and Michael N. Vrahatis , "Tuning PSO Parameters Through Sensitivity Analysis, Technical Report of the Collaborative Research Center 531 Computational Intelligence CI--124/02, University of Dortmund, January 2002.

[70] Y Zheng, Liyan Zhang, JixinQianLonghua Ma "Robust PID Controller Design using PSO" International Symposium on Intelligent Control IEEE Oct (2003).

[71] T.Bartz–Beielstein K.E. Parsopoulos and M.N. Vrahatis, "Analysis of Particle Swarm Optimization Using Computational Statistics", International conference on numerical analysis and applied mathematics ICNAAM- (2004).

[72] Haibing Hu, Qingbo Hu, Zhengyu Lu, DehongXu "Optimal PID controller Design in PMSM Servo System via Particle Swarm Optimization" 31st Annual Conference of IEEE, Industrial Electronics Society.

[73] Y.Chen et al, "A local linear wavelet neural network", In 5th world Congress on Intelligent Control and Automation, China, pp. 15-19, 2004.

[74] Q. Bai, "Analysis of Particle Swarm Optimization Algorithm," Computer and Information Science, vol. volume 3 No 1, Pebruari, 2010.

[75] Eberhart, R. C., and Shi, Y., (a). Evolving artificial neural networks. Proc. 1998 Int'l. Conf. on Neural Networks and Brain, Beijing, P.R.C., PL5-PLl3, 1998.

[76] Eberhart, R. C., and Hu, X., Human tremor analysis using particle swarm optimization. Proc. Congress on Evolutionary Computation 1999, Washington, DC, pp 1927-1930. Piscataway, NJ: IEEE Service Center, 1999.

[77] Tandon, V., Closing the gap between CAD/CAM and optimized CNC end milling. Master's thesis, Purdue School of Engineering and Technology, Indiana University Purdue University Indianapolis, 2000.

[78] Yoshida, H., Kawata, K., Fukuyama, Y., and Nakanishi, Y. A particle swarm optimization for reactive power and voltage control considering voltage stability. In G. L. Torres and A. P. Alves da Silva, Eds., Proc. Intl. Conf. on Intelligent System Application to Power Systems, Rio de Janeiro, Brazil, pp. 117-121, 1999.

[79] Eberhart, R. C., Simpson, P. K., and Dobbins, R. W. Compufational Intelligence PC Tools. Boston, MA: Academic Press Professional, 1996.

[80] Russell C. Eberhart, "Particle Swarm Optimization: Developments, Applications and Resources" Purdue School of Engineering and Technology 799 West Michigan Street Indianapolis, IN 46202 USA, IEEE, 2001.

[81] K.E.Parsopoulos and M.N.Vrahatis, "Particle swarm optimizer in noisy and continuously changing environment", Indianapolis.IN (2001).

[82] Hirotaka Yoshida, Kenichi Kawata, yoshikazuFukuyana, yosuke Nakanishi, "A particle swarm optimization for reactive power and voltage control considering voltage stability", IEEE international conference on intelligent system applications to power systems(ISAP"99), Rio de Janeiro, April 4-8 (1999).

[83] Zwe-Lee Gaing, " A particle swarm optimization approach for optimum design of PID controller in AVR system", IEEE transactions on energy conversion, November 6, (2002).

[84] J. Kennedy and R. Eberhart. Swarm Intelligence. Morgan Kaufmann Publishers, Inc., San Francisco, CA, 2001.

[85] J. Kennedy et al., "Particle Swarm Optimization", Proc. of IEEE 'Interanational conference on Neural Networks, Vol. IV, 1942-1948, 1995.

[86] H. Yoshida et al., “A Particle Swarm Optimization for Reactive Power and Voltage Control Considering Voltage Security Assessment”, IEEE Trans. On Power Systems, Vo1.15, No.4, November 2000.

[87] Matthew Settles, “An Introduction To Particle Swarm Optimization”,Department Of Computer Science, University Of Idaho, Moscow, Idaho U.S.A 83844, November 7, 2005.

[88] Dr. Karl O. Jones, “COMPARISON OF GENETIC ALGORITHM AND PARTICLE SWARM OPTIMISATION”, International Conference on Computer Systems and Technologies – CompSysTech’2005.

[89] S. M. Giriraj Kumar, “PSO based Tuning of a PID Controller for a High Performance Drilling Machine”, International Journal of Computer Applications (0975 - 8887) Volume 1 – No. 19, 2010.

[90] Matt Settles, “Neural Network Learning Using Particle Swarm Optimizers”, School of Engineering School of Engineering, University of Portland University of Portland, Portland, OR 97203 Portland, OR 97203 U.S.A, 2001.