# CHAPTER ONE

# INTRODUCTION

## 1.1 General Review

The importance of PID controllers in process industry can not be over emphasized because more than half of the industrial controllers in use today utilize PID or modified PID control schemes. PID controller has a simple control structure that is easily understood by the operators which help them in tuning the PID satisfactorily. Tuning of PID is therefore an important aspect of its implementation. Fuzzy logic is used to represent qualitative knowledge, and provides interpretability to system models. By this it means that a system model is explicit and is understandable to a knowledge or systems engineer. This facilitates inspection of the model, and therefore simplifies and encourages its validation and maintenance. Zadeh has summarized fuzzy logic as a body of concepts and techniques for dealing with imprecision, information granulation, approximate reasoning and computing with words rather than numbers.

On the other hand, neural networks and fuzzy logic are two bio-mimetic techniques that are used to provide approximations to real-world problems. While for some people this biological plausibility provides some justification for their use, for others the important point is that, regardless of their origins, both approaches are known to be robust alternatives to conventional deterministic and programmed models. However, the two paradigms have distinct application domains. Neural networks are used to induce knowledge or functional relationships from instances of sampled data. This is useful when it is not possible to develop analytic models from first principles but the system is observable. However, in contrast to fuzzy systems, this knowledge is not readily understandable to the system designer because it is encapsulated in the so called black box [1].

## 1.2 Problem Statement

The flow control system has nonlinear and time varying behaviours thus it is difficult to derive and identify an appropriate dynamic model for traditional controllers. In addition, it is very difficult to get an accurate and linearized mathematical model for such system. With PID controller it is difficult to obtain the desired response due to fixed parameters of PID after been calculated when change occurs in environment or operation conditions.

## 1.3 Objectives

The main aims of this study are

- Design and simulation of flow control system using PID controller.
- Design and simulation of flow control system using fuzzy logic.
- Design and simulation of flow control system using neuro-fuzzy network.
- Comparison of all proposed controllers.

## 1.4 Methodology

In this study the type of fuzzy inference systems that can be implemented in the fuzzy logic toolbox: Mamdani-type. The mathematical model of the flow control will be developed and simulated using MATLAB toolbox programming. Back Propagation (BP) algorithm is used to learn the neural network. The BP algorithm learns the weights for a multilayer network, given a network with a fixed set of units and interconnections. Finally, evaluate the performance of the three controllers, based on simulation results.

## 1.5 Layout

This research consists of five chapters. Chapter one contains of background, research problems, objective, and methodology. While chapter two consists of an introduction flow system, The PID controller overview, general background of fuzzy logic and neural networks. Chapter three covers the controller's designs in MATLAB /SIMULINK. Chapter Four contains the simulation rustles and all proposed controllers. Chapter five consists of conclusion and recommendations.

# CHAPTER TWO

# LITERATURE REVIEW

## 2.1 Flow Control System

In many industrial processes control of liquid level is required. It was reported that about 25% of emergency shutdowns in the nuclear power plant are caused by poor control of the steam generator water level. Such shutdowns greatly decrease the plant availability and must be minimized water level control system is a very complex system because of the nonlinearities and uncertainties of a system need for performance improvement in existing water level regulators is therefore needed [2].

### 2.1.1 Fluids characteristic

Fluids are divided into liquids and gases. A liquid is hard to compress. It changes its shape according to the shape of its container with an upper free surface. Gas on the other hand is easy to compress and fully expands to fill its container. Consequently an important characteristic of a fluid from the viewpoint of fluid mechanics is its compressibility. Another characteristic is its viscosity.  Fluid increases its pressure against compression trying to retain its original volume. This characteristic is called compressibility. Furthermore, a fluid shows resistance whenever two layers slide over each other. This characteristic is called viscosity [3].

### 2.1.2 Mechanics of fluids

The mechanics of fluids is the field of study in which the fundamental principles of general mechanics are applied to liquids and gases. These principles are the conservation of matter, the conservation of energy Newton's laws of motion and laws of thermodynamic. By the use of these principles it is able to explain observed phenomena but also to predict the behaviour of fluids under specified conditions [4].

### 2.1.3  Classification and description of fluid flow

There are certain different types of fluid flow as flow :

### i-Internal and external flows

The distinction between internal and external flows often needs to be made when the motion of a fluid is between bounding surfaces the flow is described as internal flow. Airflow management systems are widely used to control the quality of air within buildings and vehicles the movement of air within the ducting which forms part of such a system is an example of an internal flow. Conversely when a body is surrounded by a fluid in motion the flow around the immersed body is described as external flow. Examples of external flows are the flows surrounding an aircraft wing.

### ii- Steady and unsteady flows

Steady flow is defined as the various parameters at any point do not change with time. When the various parameters change with time the flow is termed unsteady. In practice absolutely steady flow is the exception rather than the rule but many problems may be studied effectively by assuming that the flow is steady. A particular flow may appear steady to one observer but unsteady to another this is because all movement is relative any motion of one body can be described only by reference to another body [4].

## 2.2 Servo Control System

A servo control system is one of the most important and widely used forms of control system. Any machine or piece of equipment that has rotating parts will contain one or more servo control systems. The job of the control system may include:

• Maintaining the speed of a motor within certain limits even when the load might vary. This is called regulation.

• Varying the speed of a motor and load according to an externally set programme of values. This is called set point tracking [4].

### 2.2.1 Fundamentals of servo motion control

The basic reasons for using servo systems in contrast to open loop systems include the need to improve transient response times, reduce the steady state errors and reduce the sensitivity to load parameters. Improving the transient response time generally means increasing the system bandwidth. Faster response times mean quicker settling allowing for higher machine throughput. Reducing the steady state errors relates to servo system accuracy. Finally, reducing the sensitivity to load parameters means the servo system can tolerate fluctuations in both input and output parameters. An example of an input parameter fluctuation is the incoming power line voltage. Examples of output parameter fluctuations include a real time change in load inertia or mass and unexpected shaft torque disturbances. Servo controlling general can be broken in to two fundamental classes of problems. The first class deals with command tracking. It addresses the question of how well does the actual motion follow what is being commanded. The typical commands in rotary motion control are position, velocity, acceleration and torque. For linear motion force is used instead of torque. The part of servo control that directly deals with this is often referred to as "Feed forward" control. It can be thought of as what internal commands are needed such that the user's motion commands are followed without any error, assuming of course a sufficiently accurate model of both the motor and load is known. The second general class of servo control addresses the disturbance rejection characteristics of the system. Disturbances can be anything from torque disturbances on the motor shaft to incorrect motor parameter estimations used in the feed forward control [5].

### 2.2.2 Modelling a simple servo system

The basic form of a Directs Current (DC) servo system is made of an electric motor with an output shaft that has an inertial load (J) on it and friction (f) in the bearings of the motor and load. There will be an electric drive circuit where an input voltage u(t) is transformed by the motor into a torque t(t) in the motor output shaft. Using systems modelling ideas for mechanical systems a torque balance can be written between the input torque from the motor and the torque required to accelerate the load and overcome friction. This is shown in the Equation (2.1).

$$Y(s) = \frac{k}{s(Ts+1)}U(s) \tag{2.1}$$

Where Y(s) is the output shaft position and U(s) is the motor input. $k$ is the system gain and T the time constant**.** The transfer function model can be decomposed into the transfer function from the motor input to the motor speed V(s) and the transfer function from the motor speed to the output shaft position [4].

$$V(s) = \frac{k}{(Ts+1)}U(s) \tag{2.2}$$

## 2.3 The Proportional Integral Derivative Algorithm

The PID algorithm is the most popular feedback controller used within the process industries. It has been successfully used for over 50 years. It is a robust easily understood algorithm that can provide excellent control. The transfer function of the PID controller looks like the following:

$$k_p + \frac{K_i}{s} + k_d\,\mathrm{s} = \frac{k_p s + K_i + k_d s^2}{s} \tag{2.3}$$

Where :

•$k_p$ = Proportional gain

•$K_i$ = Integral gain

•$k_d$ = Derivative gain

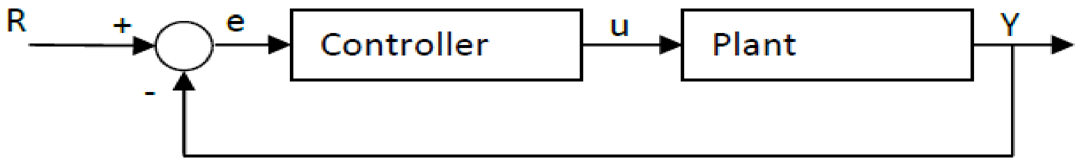


Figure 2.1: PID controller

The PID controller works in a closed-loop system as shown in Figure 2.1 the variable (e) represents the tracking error the difference between the desired input value (R) and the actual output (Y). This error signal will be sent to the PID controller and the controller computes both the derivative and the integral of this error signal. The PID controller time domain equation as:

$$\mathrm{U} = k_p \mathrm{e} + \mathrm{k_i} \int e dt + k_d \frac{de}{dt} \tag{2.4}$$

The signal (u) just past the controller is now equal to the proportional gain times the magnitude of the error plus the integral gain times the integral of the error plus the derivative gain times the derivative of the error. This signal (u) will beset to the plant and the new output will be obtained. This new output will be sent back to the sensor again to find the new error signal. The controller takes this new error signal and computes its derivative and its integral again this process goes on and on [6].

### 2.3.1 The 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 controller will have the effect of eliminating the steady-state error ($e_{ss}$) for a constant or step input, but it may make the transient response slower. A derivative controller will have the effect of increasing the stability of the system reducing the overshoot, and improving the transient response. The effects of each of controller parameters, $k_p$,$k_d$ and $K_i$ on a Closed Loop (CL) system are summarized in the Table 2.1.

Table2.1: The effect of P, I and D controller for time response

| CLRESPONSE | RISE TIME | OVERSHOOT | SETTLING TIME | S-S ERROR |
|---|---|---|---|---|
| $k_p$ | Decrease | Increase | Small Change | Decrease |
| $K_i$ | Decrease | Increase | Increase | Eliminate |
| $k_d$ | Small Change | Decrease | Decrease | No Change |

Note that these correlations may not be exactly accurate because $k_p$, $K_i$ and $k_d$ are dependent on 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 $k_p$ and $k_d$ [6].

### 2.3.2 Tuning ruled for PID controller

The mathematical model of the plant can be derived then it is possible to apply various design techniques for determining parameters of the controller that will meet the transient and steady-state specifications of the closed-loop system. However, if the plant is so complicated that its mathematical model cannot be easily obtained then an analytical approach to the design of a PID controller is not possible. Then we must resort to experimental approaches to the tuning of PID controllers. The process of selecting the controller parameters to meet given performance specifications is known as controller tuning. Ziegler and Nichols suggested rules for tuning PID controllers (meaning to set values $K_p$, $T_d$ and $T_i$) based on experimental step responses or based on the value of K that results in marginal stability when only proportional control action is used. Ziegler-Nichols rules which are useful when mathematical models of plants are not known. Such rules suggest a set of values of $K_p$, $T_i$ and $T_d$ that will give a stable operation of the system. However the resulting system may exhibit a large maximum overshoot in the step response which is unacceptable. In such a case series of fine tunings until an acceptable result is obtained. In fact the Ziegler-Nichols tuning rules give an educated guess for the parameter values and provide a starting point for fine tuning, rather than giving the final settings for $K_p$, $T_i$ and $T_d$ in a single shot [6].

### 2.3.3 Ziegler Nichols rules for tuning PID controllers

Ziegler and Nichols proposed rules for determining values of the proportional gain integral time and derivative time based on the transient response characteristics of a given plant. Such determination of the parameters of PID controllers or tuning of PID controllers can be made by engineers on-site by experiments on the plant. There are two methods called Ziegler-Nichols tuning rules

### I-First method

In the first method the response of the plant is obtain due  to a unit-step input as shown in Figure 2.2. If the plant involves neither integrator (s) nor dominant complex-conjugate poles then such a unit-step response curve may look S-shaped as shown in Figure 2.3. This method applies if the response to a step input exhibits an S-shaped curve. Such step response curves may be generated experimentally or from a dynamic simulation of the plant. The S-shaped curve may be characterized by two constants delay time L and time constant T. The delay time and time constant are determined by drawing a tangent line at the inflection point of the S-shaped curve and determining the intersections of the tan gent line with the time axis and line c(t) = K. Ziegler and Nichols suggested to set the values of $K_p$, $T_i$ and $T_d$ according to the formula shown in Table 2.2.

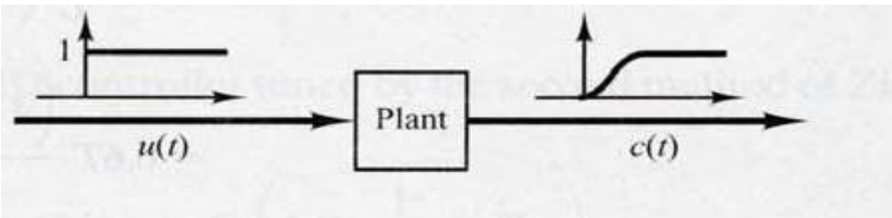


Figure 2.2: Unit-step response of a plant

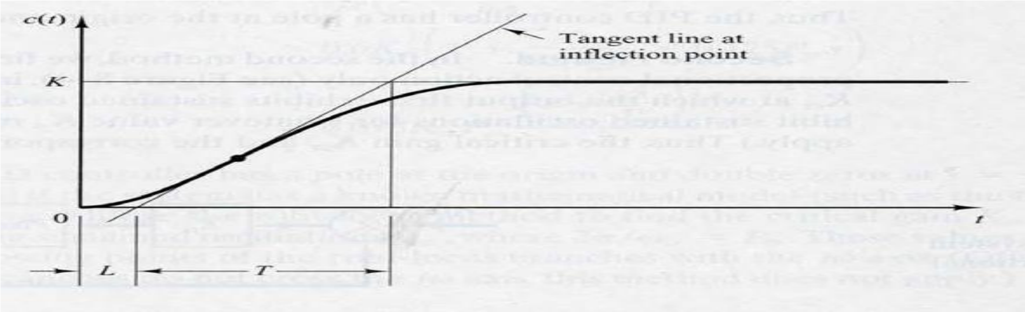


Figure 2.3: S-shaped response curve

Table 2.2: Ziegler-Nichols tuning rule based on step response of plant

| Type of controller | $K_P$ | $T_i$ | $T_d$ |
|---|---|---|---|
| P | $\frac{T}{L}$ | $\infty$ | 0 |
| PI | $0.9\frac{T}{L}$ | $\frac{L}{0.3}$ | 0 |
| PID | $1.2\frac{T}{L}$ | 2L | 0.5L |

Notice that the PID controller tuned by the first method of Ziegler-Nichols rules gives

$$G_C(S) = K_P\left(1+\frac{1}{T_{IS}}+T_D S\right) \quad (2.5)$$

$$= 1.2\frac{T}{L}\left(1+\frac{1}{2LS}+0.5LS\right) \quad (2.6)$$

$$=0.6T\frac{(S+\frac{1}{L})^2}{S} \quad (2.7)$$

Thus the PID controller has a pole at the origin and double zeros at s = -1/L

## ii-Second Method

In the second method first set $T_i = \infty$ and $T_d = 0$ using the proportional control action only increase $K_p$ from 0 to a critical value $K_{cr}$ at which the output first exhibits sustained oscillations. Thus the $K_{cr}$ and the corresponding period $P_{cr}$ are experimentally determined as shown in Figure 2.4. Ziegler and Nichols suggested that we set the values of the parameters $K_p$, $T_i$ and $T_d$ according to the formula shown in Table 2.3.

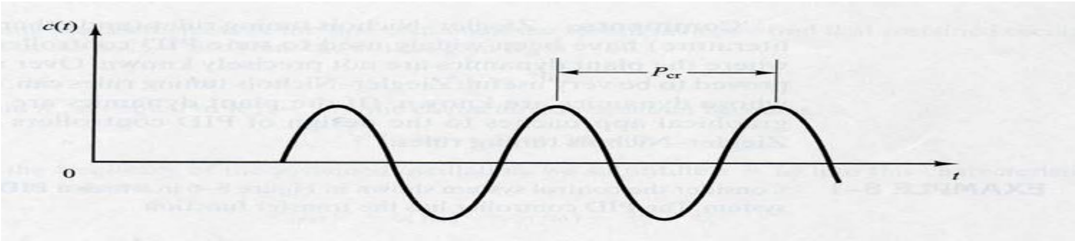


Figure 2.4: Sustained oscillation with period $P_{cr}$

Table 2.3: Ziegler-Nichols tuning rule based on $K_{cr}$, and $p_{cr}$

| Type of controller | $K_p$ | $T_i$ | $T_d$ |
|---|---|---|---|
| P | $0.5K_{cr}$ | $\infty$ | 0 |
| PI | $0.45\ K_{cr}$ | $\frac{1}{1.2}P_{cr}$ | 0 |
| PID | $0.6K_{cr}$ | $0.5P_{cr}$ | $0.125P_{cr}$ |

Notice that the PID controller tuned by the second method of Ziegler-Nichols rules gives:

$$G_C(S) = K_P\left(1+\frac{1}{T_{IS}}+T_dS\right) \tag{2.8}$$

$$G_C(S) = 0.6K_{CR}\left(1+\frac{1}{0.5P_{CR}S}+0.125P_{CR}S\right) \tag{2.9}$$

$$0.075K_{CR}P_{CR}\frac{\left(S+\frac{4}{P_{CR}}\right)^2}{S} \tag{2.10}$$

Thus the PID controller has a pole at the origin and double zeros at $s = -4/p_{cr}$ [7].

## 2.4 Fuzzy Logic Fundamentals

Over the past few years the use of fuzzy set theory or fuzzy logic in control systems has been gaining widespread popularity. Fuzzy logic-based control systems or simply Fuzzy Logic Controllers (FLCs) can be found in a growing number of products from washing machines to speedboats from air condition units to hand-held autofocus cameras. Fuzzy logic is exemplified in the speed governing system of a synchronous generator set. The success of fuzzy logic controllers is mainly due to their ability to cope with knowledge represented in a linguistic form instead of representation in the conventional mathematical framework. Control engineers have traditionally relied on mathematical models for their designs. However, the more complex a system, the less effective and the mathematical model this is the fundamental concept that provided the motivation for fuzzy logic and is formulated by Lofti Zadeh, the founder of fuzzy set theory as the principle of incompatibility [8].

### 2.4.1 Fuzzy sets and fuzzy logic

Classical set theory was founded by the German mathematician Georg Cantor. In the theory a universe of discourse $U$ is defined as a collection of objects all having the same characteristics. A classical set is then a collection of a number of those elements. The member elements of a classical set belong to the set 100 percent. Other elements in the universe of discourse which are non-member elements of the set are not related to the set at all. A definitive boundary can be drawn for the set as depicted in Figure 2.5.

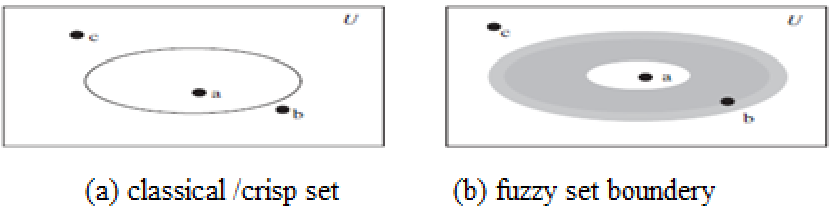


Figure (2.5): Classical/crisp set and fuzzy set boundary

A classical set can be denoted by $A = \{x \in U \mid P(x)\}$ where the elements of A have the property $P$ and U. The characteristic function $\mu_A(x)$: U $\longrightarrow$ {0, 1} is defined as '0' if $x$ is not an element of $A$ and '1' if $x$ is an element of $A$. Here U contains only two elements '1' and '0'. Therefore an element $x$ in the universe of discourse is either a member of set A or not a member of set $A$ there is ambiguity about membership. In fuzzy set theory the concept of characteristic function is extended into a more generalized form known as membership function $\mu_A(x)$: U $\rightarrow$ {0, 1} While a characteristic function exists in a two-element set of {0, 1} a membership function can take up any value between the unit interval [0, 1]. The set which is defined by this extended membership function is called a fuzzy set. In contrast a classical set which is defined by the two-element characteristic function. Fuzzy set theory essentially extends the concept of sets to encompass vagueness. Membership to a set is no longer a matter of 'true' or 'false', '1' or '0' but a matter of degree. The exact nature of the relation depends on the shape or the type of membership function used in the system [8].

### 2.4.2 Types of membership functions

Figure 2.6 shows various types of membership functions which are commonly used in fuzzy set theory. The choice of shape depends on the individual application.

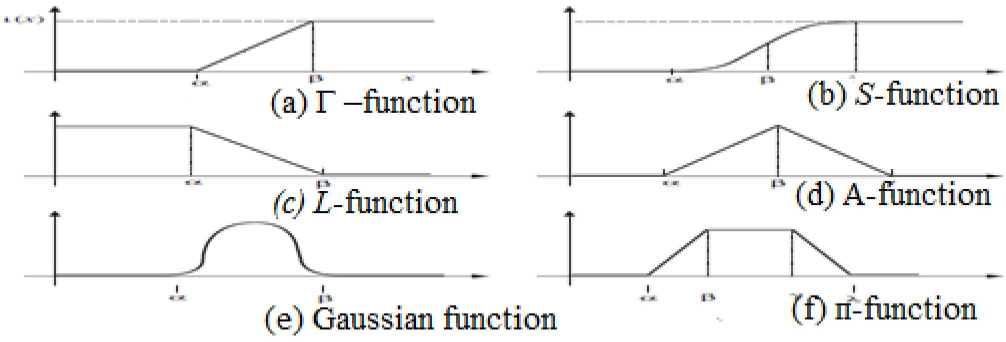


Figure 2.6: Types of membership functions

### 2.4.3 Linguistic variables

The concept of a linguistic variable a term which is later used to describe the inputs and outputs of the FLC is the foundation of (FLC) systems. A conventional variable is numerical and precise. It is not capable of supporting the vagueness in fuzzy set theory. By definition a linguistic variable is made up of words sentences or artificial language which is less precise than numbers. It provides the means of approximate characterization of complex or ill-defined phenomena. [8]

### 2.4.4 Fuzzy logic implementation

Fuzzy implication is an important connective in fuzzy control systems because the control strategies are embodied by sets of IF-THEN rules. There are various different techniques in which fuzzy implication may be defined. These relationships are mostly derived from multivalued logic theory. The following are some of the common techniques of fuzzy implication found in literature

### i-Mamdani system

This method is widely accepted for capturing expert knowledge. It allows us to describe the expertise in more intuitive, more humanlike manner. However, Mamdani type FIS entails a substantial computational burden.

### ii-Takagi-Sugeno

This method is computationally efficient and works well with optimization and adaptive techniques, which makes it very attractive in control problems particularly for dynamic nonlinear systems. These adaptive techniques can be used to customize the membership functions so that fuzzy system best models the data. The most fundamental difference between Mamdani type FIS and Sugeno type FIS is the way the crisp output is generated from the fuzzy inputs. While Mamdani type FIS uses the technique of defuzzification of a fuzzy output Sugeno type FIS uses weighted average to compute the crisp output. The expressive power and interpretability of Mamdani output is lost in the Sugeno FIS since the consequents of the rules are not fuzzy. But Sugeno has better processing time since the weighted average replace the time consuming defuzzification process. Due to the interpretable and intuitive nature of the rule base Mamdani type FIS is widely used in particular for decision support application. Other differences are that Mamdani FIS has output membership functions whereas Sugeno FIS has no output membership functions. Mamdani FIS is less flexible in system design in comparison to Sugeno FIS as latter can be integrated with ANFIS tool to optimize the outputs [8].

### 2.4.5 Fuzzy control systems

In typical fuzzy logic controller there are five principal elements as shown in Figure 2.7

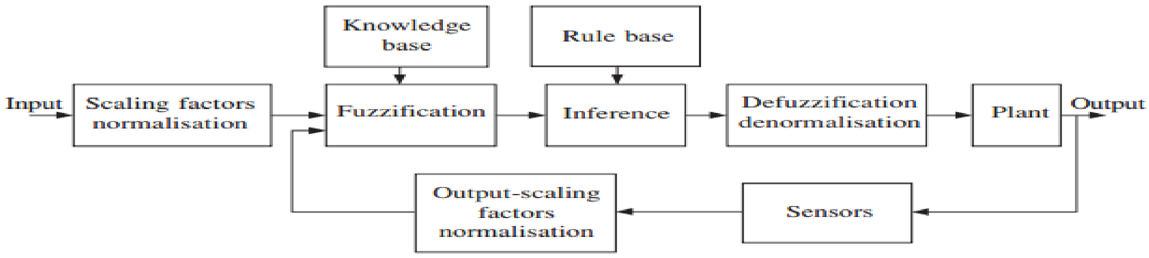


Figure 2.7: Block diagram of a typical fuzzy logic controller

- Fuzzification module (fuzzifier).
- Knowledge base.
- Rule base.
- Inference engine.
- Defuzzification module (defuzzifier).

Automatic changes in the design parameters of any of the five elements create an adaptive fuzzy controller. Fuzzy control systems with fixed parameters are non-adaptive. Other non-fuzzy elements which are also part of the control system include the sensors the analogue–digital converters the digital–analogue converters and the normalisation circuits. There are usually two types of normalisation circuits one maps the physical values of the control inputs onto a normalised universe of discourse and the other maps the normalised value of the control output variables back onto its physical domain.

- **Fuzzifier**

The fuzzification module converts the crisp values of the control inputs into fuzzy values so that they are compatible with the fuzzy set representation in the rule base. The choice of fuzzification strategy is dependent on the inference engine i.e. whether it is composition based or individual-rule-firing based.

- **Knowledge base**

The knowledge base consists of a database of the plant. It provides all the necessary definitions for the fuzzification process such as membership functions, fuzzy set representation of the input–output variables and the mapping functions between the physical and fuzzy domain.

- **Rule base**

The rule base is essentially the control strategy of the system. It is usually obtained from expert knowledge or heuristics and expressed as a set of IF-THEN rules. The rules are based on the fuzzy inference concept and the antecedents and consequents are associated with linguistic variables.

- **Defuzzifier**

The diagram in Figure 2.8 shows the membership functions related to a typical fuzzy controller's output variable defined over its universe of discourse. The FLC will process the input data and map the output to one or more of these linguistic values ($LU1$ to $LU5$). Depending on the conditions the membership functions of the linguistic values may be clipped. The union of the membership functions forms the fuzzy output value of the controller [9].

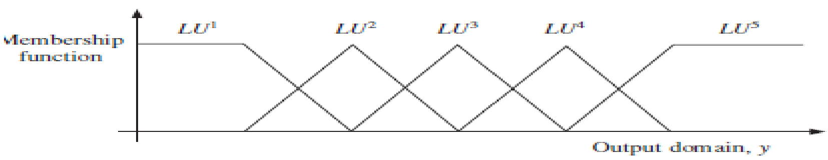


Figure 2.8: Membership functions of the output linguistic values

## 2.5 Artificial Neural Network

Neural control is a branch of the general field of intelligent control which is based on the concept of Artificial Intelligence (AI). AI can be defined as computer emulation of the human thinking process. The AI techniques are generally classified as Expert Systems (ES), Fuzzy Logic (FL) and Artificial Neural Networks (ANN). The classical ES are based on Boolean algebra and use precise calculations while FL systems involve calculations based on an approximate reasoning. The use of ANNs is the most powerful approach in AI. ANNs are information processing structures which emulate the architecture and operational mode of the biological nervous tissue. Any ANN is a system made up of several basic entities named neurons which are interconnected and operate in parallel transmitting signals to one another in order to achieve a certain processing task. One of the most outstanding features of ANNs is their capability to simulate the learning process. They are supplied with pairs of input and output signals from which general rules are automatically derived so that the ANN will be capable of generating the correct output for a signal that was not previously used. The neural approach can be combine with the fuzzy logic generating neuro – fuzzy system that combine the advantages of the two control paradigm [8].

### 2.5.1 Neural networks architectures

Artificial neural networks differ by the type of neurons they are made of and by the manner of their interconnection. There are two major classes of neural networks feed-forward ANNs and recurrent ANNs. Feed-Forward Artificial Neural Networks (FFANNs) are organized into cascaded layers of neurons. Each layer contains neurons receiving input signals from the neurons in the previous layer and transmitting outputs to the neurons in the subsequent layer. The neurons within a layer do not communicate to one another. The first network layer is named the input layer, while the last one is named the output layer. All the other neuron layers are known as the hidden layers of the neural network. FFANNs do not have any memory of the past inputs so that they are used for applications where the output is only a function of the present inputs. Therefore, each input vector is simply associated with an output vector. If step activation functions are used several analogue or discrete input vectors can be associated with a single discrete output vector. Such neural networks are used to solve classification problems. In a classification problem the set of all possible input vectors is divided into several arbitrary subsets. Each subset is a class. The problem consists of finding out to which class a given input vector belongs. The neural network associates each class with a binary vector and generates the corresponding code for any input vector. Recurrent artificial neural networks include architectures where neurons in the same layer communicate cellular neural networks or architectures where some of the outputs of a FFANN are used as inputs. These neural architectures can be described either by continuous time models or by discrete time models. Figure 2.9 shows the neural network architecture [8].

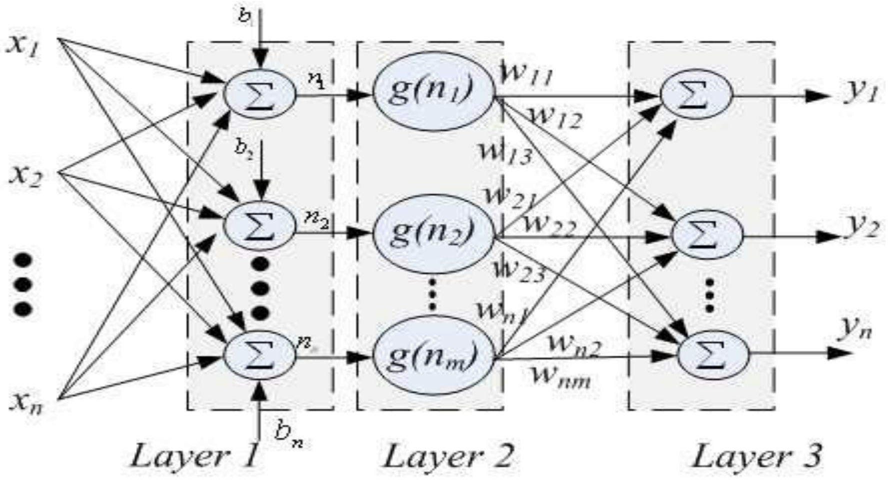


Figure 2.9: Neural network architectures

### 2.5.2 Training algorithms

One of the most important features of neural networks is their ability to learn and improve their operation using a set of examples named training data set the training process is controlled by mathematical algorithms that fall in two main classes constructive and non-constructive. The non-constructive training algorithms adapt only the connection weights and the threshold levels. The constructive algorithms modify all the network features including its architecture. All the algorithms modify the neuron weights and thresholds based on calculations that analyze the network response to particular inputs. The modifications are performed in a manner that brings the network outputs closer to the expected ones. Depending on the nature of the training data set there are two categories of algorithms supervised and unsupervised. The supervised algorithms use a training data set composed of input–output pairs. The unsupervised algorithms use only the input vectors. In the case of supervised algorithms, the training process is controlled by an external entity (the teacher) that is able to establish whether the network outputs are adequate to the inputs and what is the size of the error. Then the network parameters are modified according to the particular correction method defining each training algorithm. In case of unsupervised methods (the Hebbian rule, the 'winner takes all' algorithm, etc.), there are no means to know what the expected outputs are. The network evolves as a result of the experience gained from the previous input vectors. The weight values converge to a set of final values dictated by the input values used as training data set in conjunction with the particular training algorithm. The unsupervised family of training algorithms is mainly used for signal and image processing, where pattern classification, data clustering or compression algorithms are involved. The control engineering problems are better tackled by supervised training methods, as the relationship between inputs and desired outputs is better defined and easier to control [10].

### i-The error back-propagation algorithm

The most popular supervised training algorithm is the one named 'error back propagation' or simply 'back-propagation'. It involves training a FFANN structure made up of sigmoidal activation function neurones. The back propagation algorithm is a gradient method aiming to minimise the total operation error of the neural network. The total error is a function defined by equation (2.11) where $Oi^{ref}$ is the column vector of the reference outputs and $Oi$ is the column vector of the actual network outputs corresponding to the input pattern number '$i$'. The total error Err is the sum of the errors corresponding to all $n_p$ input patterns.

$$\text{Err} = \sum_{i=1}^{n_p} (oi^{erf} - oi)^T . (oi^{erf} - oi) = \sum_{i=1}^{n_p} \|(oi^{erf} - oi)^2\| \qquad (2.11)$$

For each training step the vector of all neurone weights and threshold weights ($W$) is updated in such a way that the total error Err is decreased. The vector $W$ can be associated to a point in a $NW$-dimensional space where $NW$ is the total number of weights and thresholds in the neural network. The most efficient way to perform the update is to shift the point $W$ along the curve indicated by the gradient of the total error. This principle is illustrated by Equation (2.12) where $W$ ($t$) is the parameter vector during the current training cycle $W(t + 1)$ is the parameter vector for the next training cycle and Ƞ is the learning-rate constant. The algorithm stops when the total error is zero. In practice it is stopped when the error is considered negligible

$$W(t+1) = W(t) - \eta \cdot \nabla Err = W(t) - \eta \cdot \left[ \frac{\partial Err}{\partial w_1} \frac{\partial Err}{\partial w_1} \frac{\partial Err}{\partial w_1} \ldots\ldots \frac{\partial Err}{\partial w_1} \right] \quad (2.12)$$

For the practical calculation of the error gradient ∇Err the components in the vector $W$ are usually rearranged as a three-dimensional matrix. The matrix has a number of rectangular layers equal to the number of neurone layers in the neural network. Each rectangular layer is a two-dimensional matrix containing one line for each neurone in the corresponding layer of the neural network. Each line includes the input weights and its threshold level of a neurone. Therefore, the element $w_{jkm}$ in the three-dimensional matrix is the weight '$m$' of the neurone '$k$' situated in the layer '$j$' inside the neural network. The threshold level corresponds to the last element in each line and is not treated any differently to the input weights because it can be considered as an extra weight supplied with a constant input signal –1. the back-propagation algorithm is not guaranteed to generate a satisfactory solution for all input output association problems and FFANN architectures. The training result depends on several factors as follow:

- Network architecture (number of layers, number of neurones in each layer).
- Initial parameter values $W(0)$.
- The details of the input–output mapping.
- Selected training data set (pairs of inputs and corresponding desired outputs).
- The learning-rate constant Ƞ.

Back-propagation is not a constructive algorithm the network architecture has to be chosen in advance. Unfortunately, there is no clearly defined set of rules to be followed in order to decide which the most appropriate architecture for a problem is choosing the architecture is a result of a trial and error process supported by previous experience. However, it has been mathematically demonstrated that any input–output mapping can be learned by a FFANN with only one hidden layer provided that the number of neurones in the hidden layer is large enough for the problem to be solved. This means that if a neural network proves incapable of learning how to perform a certain task the one possible solution is to increase the number of neurones in the hidden layer or layers. A different solution is to restart the algorithm with another set of initial parameters $W(0)$. This solution is based on the assumption that the previous failure was generated by stopping at a local minimum. The trajectory of vector $W$ in the parameter space is dependent on its starting point $W(0)$, therefore the situation may be avoided by changing the initial weights and thresholds. Another important aspect is choosing an adequate training data set, so that if the number of different input values is finite the training data set covers all the possibilities. Nevertheless, if this number is infinite or if the number is too large then only a selection of input combinations will be used to train the neural network. The quality of the training process is influenced by the way the training data set is generated [10].

### ii-Training algorithms for neurones with step activation functions

If the activation functions of the neurons in FFANN are not sigmoidal the back propagation algorithm cannot be used because the error function cannot be derived. However, other recursive methods presented in (2.13) and (2.14) are applicable to the FFANNs with only one layer. These are recursive methods like the back-propagation algorithm but in this case the training process always has a finite number of cycles provided that the desired input–output relation can be learned by a one layer network.

$$W_{JK}^{T+1} = W_{JK}^{T} + \eta . \sum_{I=1}^{N_P} X_{K-} \left(O_{ij}^{ref} - f_{ij}\right) \tag{2.13}$$

$$W_{JK}^{T+1} = W_{JK}^{T} + \eta . \sum_{I=1}^{N_P} X_{K-} \left(O_{ij}^{ref} - net_{ij}\right) \tag{2.14}$$

Finding the correct weights for a multilayer FFANN with step activation functions is a complicated problem. The two previous methods cannot be generalized for such networks and either constructive methods or genetic algorithms need to be used instead. There are many other training algorithms than the ones presented here. However, the algorithms presented are used in the vast majority of control applications [9].

## 2.6 Adaptive Neuro-Fuzzy Inference Systems

Adaptive Neuro-Fuzzy Inference Systems (ANFIS) is a hybrid network which consists of a combination of two controllers Fuzzy logic and neural network. These both controllers result in a single entity which enhances the features of controlling machine than using single controller alone. The main idea of fuzzy logic control is to build a model of a human control expert capable of controlling the plant without thinking in terms of a mathematical model. These control rules are translated into the framework of fuzzy set theory providing a calculus which can simulate the behaviour of the control expert. The specification of good linguistic rules depends on the knowledge of the control expert, but the translation of these rules into fuzzy set theory framework is not formalized and arbitrary choices concerning. The quality of fuzzy logic controller can be drastically affected by the choice of membership functions. Thus, methods for tuning fuzzy logic controllers are necessary.

Neural networks offer the possibility of solving the problem of tuning. Although a neural network is able to learn from the given data the trained neural network is generally understood as a black box. Neither it is possible to extract structural information from the trained neural network. On the other hand, a fuzzy logic controller is designed to work with the structured knowledge in the form of rules and nearly everything in the fuzzy system remains highly transparent and easily interpretable. However, there exists no formal framework for the choice of various design parameters and optimization of these parameters generally is done by trial and error. A combination of neural networks and fuzzy logic offers the possibility of solving tuning problems and design difficulties of fuzzy logic. The resulting network will be more transparent and can be easily recognized in the form of fuzzy logic control rules or semantics. This new approach combines the both methods and avoids the drawbacks of both [11].

# CAPTER THREE

# METHODOLOGY

## 3.1 System Description

The coupled tank system consists of two vertical tanks interconnected by a flow channel as shown in Figure 3.1 which causes the levels of the two tanks to interact. Each tank has an independent pump for inflow of liquid. The sectional area of the outlets present and the base of each tank and the channel connecting the two tanks can be varied with rotary valves. The amount of water which returns to the reservoir is approximately proportional to the head of water in the tank since the return tube is made of flexible tubing which aids in varying the hydraulic resistance.

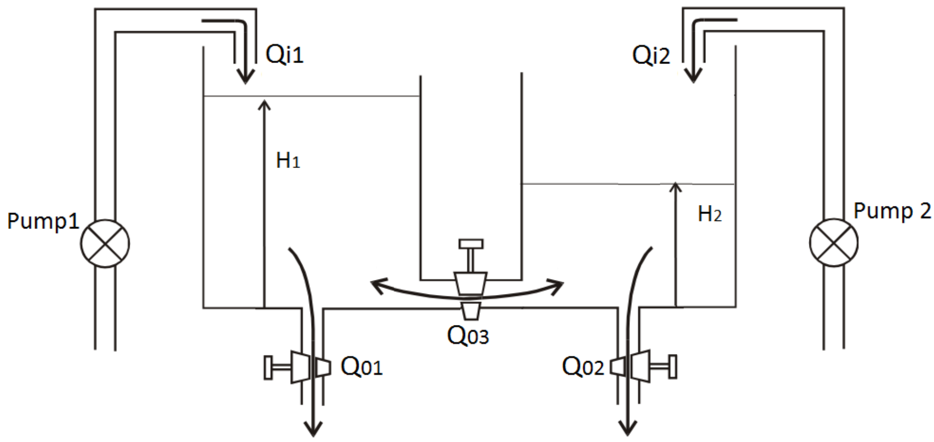


Figure 3.1: Layout of the interacting tanks system

The level of water in each tank is monitored by a capacitive – type probe signal conditioning circuits convert the measured capacitance to electrical signals in the range of 0 to +5 volt DC. The zero level has been calibrated to represent the rest point of the water level that is when tank is nearly empty (about 20 mm in the scale) while the full state (+5 volt) is calibrated at the level of the opening to the rear over flow stand pipes with scale showing 300mm approximately an internal baffle controls leakage between the two tanks to simulate interacting tank arrangement. The baffle is raised by a small amount by turning the wing nut on the top of the tank assembly in order to provide a useful range of inter tank resistance .A spring returns the baffle to the closed position when the wing nut is released [12].

### 3.1.1 Fundamental control principle of coupled tank system

Process variable or controlled variable for this system that is the variable which quantifies performance is actually the water level in the coupled tank system to maintain and control the water level at specified desired value. The inlet flow rate is adjusted the adjustment is made or actuated by pump voltage. The input flow rate is known as manipulated variable i.e. the variable that is used to maintain the process variable at its set point. The important characteristic of the system, the level will be maintained as long as the outflow rate, the pump flow rate and the outflow rate remained unchanged.

However, if any disturbance occurs which results in the change in either the inflow rate or the outflow rate or changes that maybe necessary for process then the liquid level in the tank would change and settle at different steady –state level. If the outflow rate is greater than the inflow rate, the liquid level will settle at the lower level than before. Assuming that a steady state condition had already been achieved before the tank is empty. Similarly if the inflow rate is higher than the liquid level will settle at a higher level, assuming that the steady state condition achieved before the tank is overflow. The control objective is that the input flow rate has to be adjusted in order to maintain the level at the previous condition. In the case where the out flow rate is greater than the inflow rate, the inflow rate has to be adjusted so that the liquid level in the tank increased and settled [12].

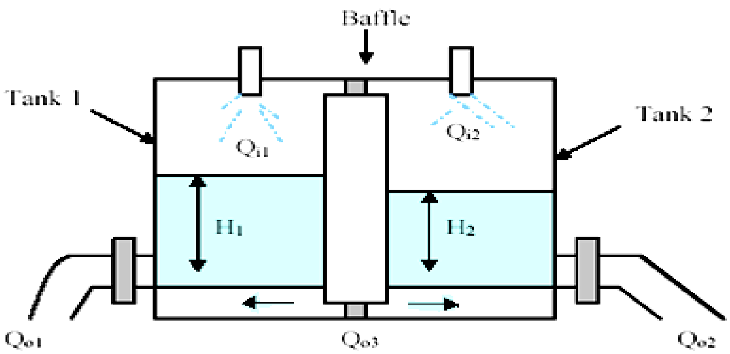


Figure 3.2: Schematic diagram of interacting tanks apparatus

Table 3.1: Parameters of the coupled –tank system

| Name | Expression | Value | | |
|---|---|---|---|---|
| Cross sectional area of the coupled tank reservoir | $A_1$ and $A_2$ | 1 $m^2$ | | |
| Proportionality α constant that depend on discharge coefficient orifice cross sectional area and gravitational constant | $\alpha_I$ subscript I denotes which tank it refers | $\alpha_1$ | $\alpha_2$ | $\alpha_3$ |
| | | 14.30 Cm | 14.30 cm | 20.00 cm |
| Elevation of coupled tank | $H_1$ | 60 cm | | |
| | $H_2$ | 50 cm | | |

### 3.1.2 Mathematical modelling of the coupled tank system

It is vital to understand the mathematical of how the coupled tank system behaves. System modelling involve devolving mathematical model by applying the fundamental physical laws of science and engineering in this system nonlinear dynamic model with time varying parameters are observed and steps are taken to drive each of the cross ponding linearized perturbation from the nonlinear model. A simple nonlinear mathematical model is derived with a help of Figure 3.2. Let $H_1$ and $H_2$ be the liquid level in each tank measured with respect to the corresponding outlet considering a simple mass balance the rate of change of liquid into the tank. Thus for each of tank 1 and tank 2 the dynamic equation is developed as follows:

$$A_1\frac{dH_1}{dt} = Q_{i1} - Q_{01} - Q_{03} \tag{3.1}$$

$$A_2\frac{dH_2}{dt} = Q_{i2} - Q_{02} - Q_{03} \tag{3.2}$$

Where are:

$H_1$, $H_2$: are heights of liquid in tank 1 and tank 2, respectively.

$A_1$, $A_2$: are cross-sectional areas of tank 1 and tank 2.

$Q_{03}$ : is the flow rate between tanks.

$Q_{i1}$, $Q_{i2}$: are pump flow rate into tank 1 and tank 2, respectively.

$Q_{01}$, $Q_{02}$: are the flow rate of liquid out of tank 1 and tank 2, respectively.

Each outlet drain can be modelled as a simple orifice Bernoulli's equation for steady a non-viscous incompressible fluid shows that the outlet flow in $tank_2$ is proportional to the square root of the head of the water in the tank similarly the flow between the two tanks is proportional to the square root of the head differential

$$Q_{01} = \alpha_1\sqrt{H_1} \tag{3.3}$$

$$Q_{02} = \alpha_2\sqrt{H_2} \tag{3.4}$$

$$Q_{03} = \alpha_3\sqrt{H_1 - H_2} \tag{3.5}$$

Where:

$\alpha_1$, $\alpha_2$ and $\alpha_3$ are proportionality constants which depend on the coefficients of discharge the cross sectional area of each area and gravitational constant. By using Equations from (3.3) to (3.5) in (3.1) and (3.2) the nonlinear equations that describe the dynamics of the multi-input and multi output system are derived as:

$$A_1\frac{dH1}{dt} = Q_{i1} - \alpha_1\sqrt{H_1} - \alpha_3\sqrt{H_1 - H_2} \tag{3.6}$$

$$A_2\frac{dH_2}{dt} = Q_{i2} - \alpha_2\sqrt{H_2} - \alpha_3\sqrt{H_1 - H_2} \tag{3.7}$$

Suppose that for set inflows $Q_{i1}$and $Q_{i2}$ the liquid level in the tanks is at some steady state levels $H_1$and $H_2$ consider small variations in each inflow, $q_1$ in $Q_{i1}$ and $q_2$ in $Q_{i2}$. Let the resulting perturbation in level be $h_1$ and $h_2$ respectively. From Equations (3.6) and (3.7) the following equations can be derived for tank1 and tank2 respectively.

$$A_1\frac{d(H_1+h)}{dt} = (Q_{i1}+q_1) - \alpha_1\sqrt{H_1 + h_1} - \alpha_3\sqrt{H_1 - H_2 + h_1 - h_2} \tag{3.8}$$

$$A_2\frac{d(H_2+h)}{dt} = (Q_{i2}+q_2) - \alpha_2\sqrt{H_2 + h_2} - \alpha_3\sqrt{H_1 - H_2 + h_1 - h_2} \tag{3.9}$$

In (3.8) and (3.9) note that the coefficients of the perturbations in the level are function of steady state operating points $H_1$ and $H_2$ .the two equations can also be written in the form (3.10) and (3.11) where $q_{01}$ and $q_{02}$ represent perturbation in the outflow at the drain pipes this would be appropriate in the case where outflow is controlled by attaching external clamp for instance based on the developed linearized model. Subtracting Equations (3.6) and (3.7) from Equations (3.8) and (3.9) the equations obtained are:

$$A_1\frac{dH_1}{dt} = q_1 - \alpha_1(\sqrt{H_1 + h_1} - \sqrt{h_1}) - \alpha_3(\sqrt{H_1 - H_2 + h_1 - h_2} - \sqrt{h_1 - h_2}) \tag{3.10}$$

$$A_2\frac{dH_2}{dt} = q_2 - \alpha_2(\sqrt{H_2 + h_2} - \sqrt{h_2}) - \alpha_3(\sqrt{H_1 - H_2 + h_1 - h_2} - \sqrt{h_1 - h_2}) \tag{3.11}$$

For small perturbations

$$\sqrt{H_1 + h_1} = \sqrt{H_1(1 + \frac{H_1}{2H_1})} \quad (3.12)$$

Therefore consequently

$$\sqrt{H_1 + h_1} - \sqrt{H_1} = \frac{h_1}{2\sqrt{H_1}} \quad (3.13)$$

Similarly

$$\sqrt{H_2 + h_2} - \sqrt{H_2} = \frac{h_2}{2\sqrt{H_2}} \quad (3.14)$$

And:

$$\sqrt{H_1 - H_2 + h_1 - h_2} - \sqrt{H_2 - H_1} = \frac{h_2 - h_1}{2\sqrt{H_2 - H_1}} \quad (3.15)$$

The configuration is achieved by having the baffle raised at as small height this a allows flow of water from $tank_1$ to $tank_2$ and with this second order configuration $h_2$ will be the process variable that is to be set whist $q_1$ is the manipulated variable that is to be controlled. The other variable like $q_2$ will be assumed zero as the model is derived under the circumstances of no disturbance abiding by this approximation (3.16) and (3.17) are established.

$$A_1\frac{dH_1}{dt} = q_1 - \alpha_1\frac{h_1}{2\sqrt{H_1}} - \alpha_3\frac{h_2 - h_1}{2\sqrt{H_2 - H_1}} \quad (3.16)$$

$$A_2\frac{dH_2}{dt} = q_2 - \alpha_1\frac{h_1}{2\sqrt{H_1}} - \alpha_3\frac{h_2 - h_1}{2\sqrt{H_2 - H_1}} \quad (3.17)$$

Performing Laplace transforms on (3.16) and (3.17) and assuming that initially all variables are at their steady state values.

$$A_1Sh_1(s) = q_1(s) - (\frac{\alpha 1}{2\sqrt{H_1}} + \frac{\alpha 3}{2\sqrt{H_2 - H_1}})\, h_1(s) + \frac{\alpha 3}{2\sqrt{H_2 - H_1}} h_2(s) \quad (3.18)$$

$$A_2Sh_2(s) = q_2(s) - (\frac{\alpha 2}{2\sqrt{H_2}} + \frac{\alpha 3}{2\sqrt{H_2 - H_1}}) h_2(s) + \frac{\alpha 3}{2\sqrt{H_2 - H_1}} h_1(s) \quad (3.19)$$

By rearranging and rewriting in abbreviated manners

$$(T_1S+1)\, h_1(s) = k_1q_1(s) + k_2q_2(s) \quad (3.20)$$

$$(T_2S+1)\, h_2(s) = k_2q_2(s) + k_{21}q_2(s) \quad (3.21)$$

**Step1:** substitute all values in table (3.1) into (3.20) and (3.21) can be obtained as

$$\tau_1 = \frac{A_1}{\frac{\alpha_1}{2\sqrt{H_1}}+\frac{\alpha_3}{2\sqrt{H_1-H_2}}} \qquad (3.22)$$

$$\tau_2 = \frac{A_2}{\frac{\alpha_2}{2\sqrt{H_2}}+\frac{\alpha_3}{2\sqrt{H_1-H_2}}} \qquad (3.23)$$

$$K_1 = \frac{1}{\frac{\alpha_1}{2\sqrt{H_1}}+\frac{\alpha_3}{2\sqrt{H_1-H2}}} \qquad (3.24)$$

$$K_2 = \frac{1}{\frac{\alpha_2}{2\sqrt{H_2}}+\frac{\alpha_3}{2\sqrt{H_1-H_2}}} \qquad (3.25)$$

$$K_{12} = \frac{\frac{\alpha_3}{2\sqrt{H1-H_2}}}{\frac{\alpha_1}{2\sqrt{H_1}}+\frac{\alpha_3}{2\sqrt{H_1-H_2}}} \qquad (3.26)$$

$$K_{21} = \frac{\frac{\alpha_3}{2\sqrt{H_1-H2}}}{\frac{\alpha_2}{2\sqrt{H_2}}+\frac{\alpha_3}{2\sqrt{H_1-H_2}}} \qquad (3.27)$$

**Step2:** Substitute all values from step 1 into Equations (3.20) and (3.21), relates between the manipulated variable $q_1$ and presses variable $h_2$ the final transfer function equation can be obtained as:

$$\frac{h2(s)}{q1(s)} = \frac{k1k2}{\tau1\tau2s^2+(\tau1+\tau2)s+(1-k12k21)} \qquad (3.28)$$

By substituting the parameters in the plant model the transfer function of the plant Shown in Equation (3.29) [12].

$\tau1=3.47$

$\tau2=3.38$

k1=0.11

k2=0.11

k12=0.77

k21=0.77

$$G(s) = \frac{0.0008}{s^2+0.85s+0.003} \qquad (3.29)$$

**Step3**: for the servo system considering the time constant T=0.2 and substituting in equation (2.2) the transfer function for the servo system shown in equation (3.30)

$$C(s) = \frac{50}{s+50} \qquad (3.30)$$

## 3.2 PID Controller Design

From the transfer function shown in Equation (3.29), PID controller has been designed using the second method of Ziegler-Nichols to get the exact values of proportional, integral and derivative gains the program written in m.file.

**Program 1**

```
Plant=tf(.0008,[1 .85 .003]);
actuator=tf(50,[1 50]);
sys=series(plant,actutor)
g=feedback(sys,1);
step(g)
figure
for k=100:5:100000
    sys1=k*sys ;
    sys2=feedback(sys1,1);
    [num den]=tfdata(sys2,'v');
    r=roots(den);
    m=max(real(r));
    if m>0
        break
    end
end
kcr=k
w=sqrt(den(4)/den(2));
pcr=((2*pi)/w)
kp=.6*kcr
kd=.125*pcr
ki=.5*pcr
controller=tf([kd kp ki],[1 0])
sys3=series(sys,controller)
g2=feedback(sys3,1)
step(g2)
```

The block diagram for tuned PID controller in MATLAB/SIMULINK is shown in Figure 3.3.

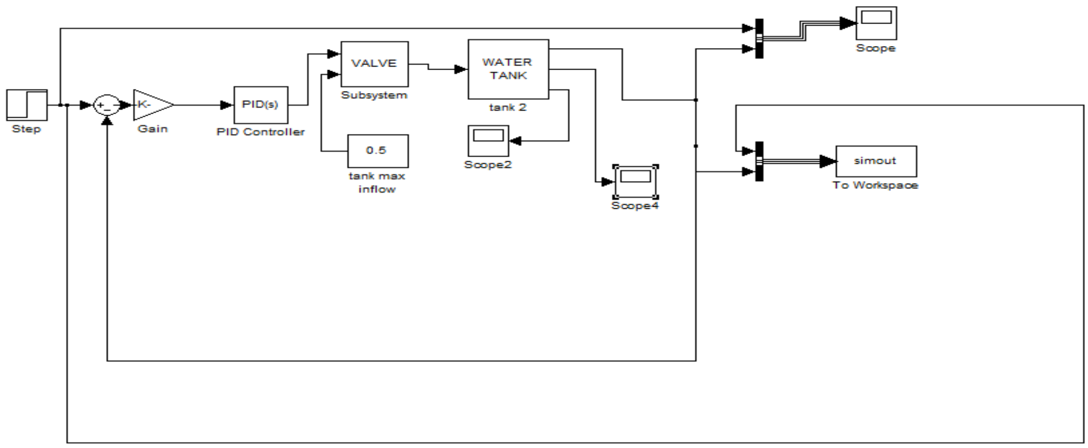


Figure 3.3: Model of PID controller

## 3.3 Fuzzy Controller Design

The  fuzzy controller in present work is a mamdani uses a rule base in linguistic terms There are two inputs the level and the change of liquid level and one output parameter the inlet valve control angle. The Gaussian membership functions are selected to fuzzify the inputs and the triangular is selected for output variables. There are set fuzzy sets taken (low, Ok and high) for level input and (negative, none and positive) for the rate input and five fuzzy sets for the output variable.

### 3.3.1 The FIS editor

The fuzzy logic controller makes use of two inputs and one output. The first input is the level of the tank denoted as level while the second is change of the level denoted as rate. According to the rules written in the rule editor the controller takes the action and governs the opening of the valve which is the output of the controller and is denoted by valve. In order to start the FIS editor, 'fuzzy' is typed in the MATLAB command window and the enter button is pressed. The FIS editor pops up with the mamdani style of fuzzyfication set as. The FIS editor opens with only one input and one output. Therefore in order to add a second input variable the 'add variable' option is selected from the edit option in the toolbar. The input and output blocks are selected and their names are written starting from the inputs as shown in Figure 3.4.

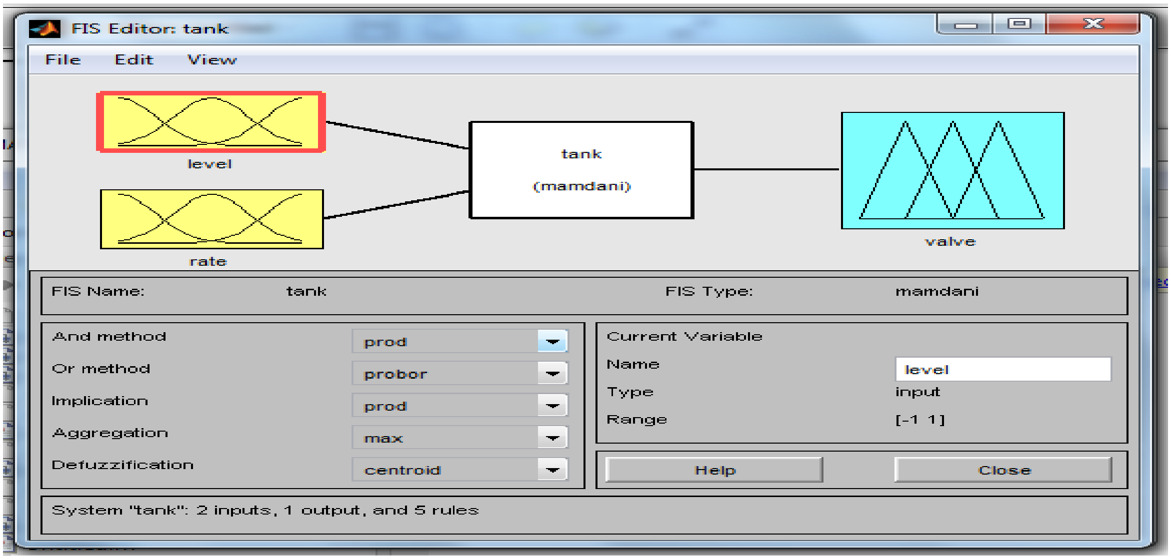


Figure 3.4: Fuzzy FIS editor

### 3.3.2 The membership function editor

The membership function editor is the tool that display and edit all of the membership function associated with all of the input and output variables for the entire fuzzy inference system. The second stage involves setting the membership functions of the two inputs and the output. Double clicking on level block found on the FIS editor opens the Membership function window. The name of the input is assigned by typing the name in the name box as seen in Figure 3.5.

The membership function window for the rate input parameter is gotten by double clicking on the rate block found in the FIS editor shown in Figure 3.6. Function types for the output the range is set to (-1, 1) to cover the output range. The close fast membership function will have the parameters (-1, -0.9, -0.8). The close low membership function will be (-0.6, -0.5, 0.4) the no change membership function will be (-0.1, 0, 0.1), the open slow membership function will be (0.2, 0.3, 0.4) and the open fast membership function will be (0.8, 0.9, 1) which are shown in Figure 3.7.

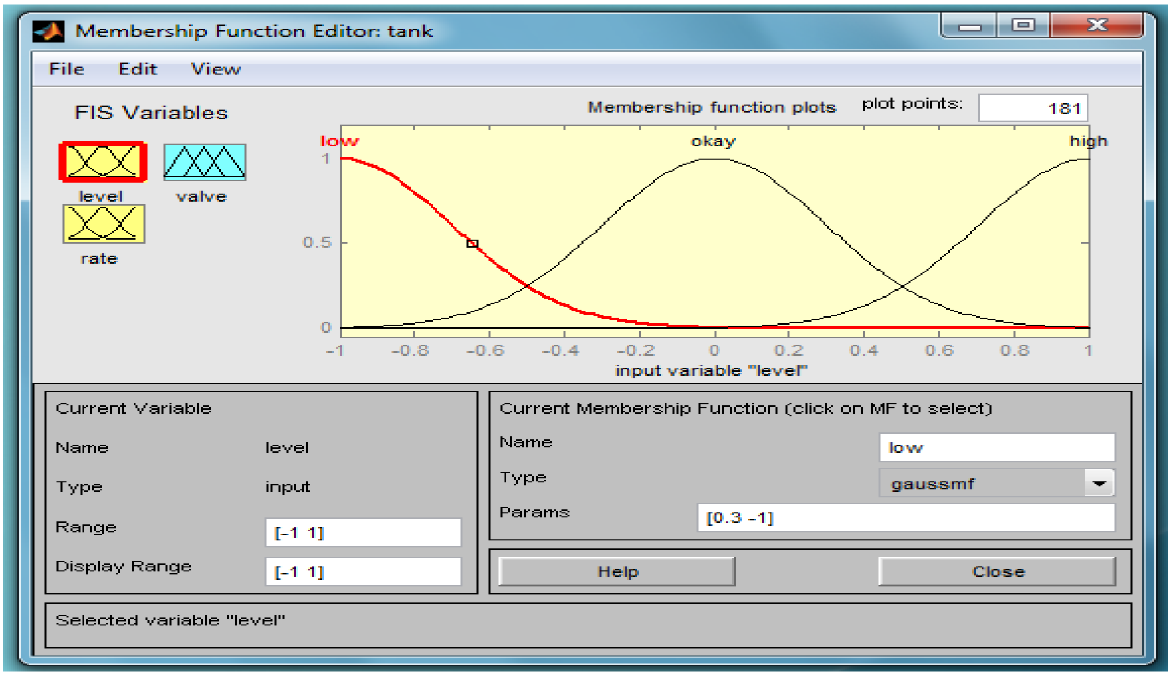


Figure 3.5: Membership function editor for level

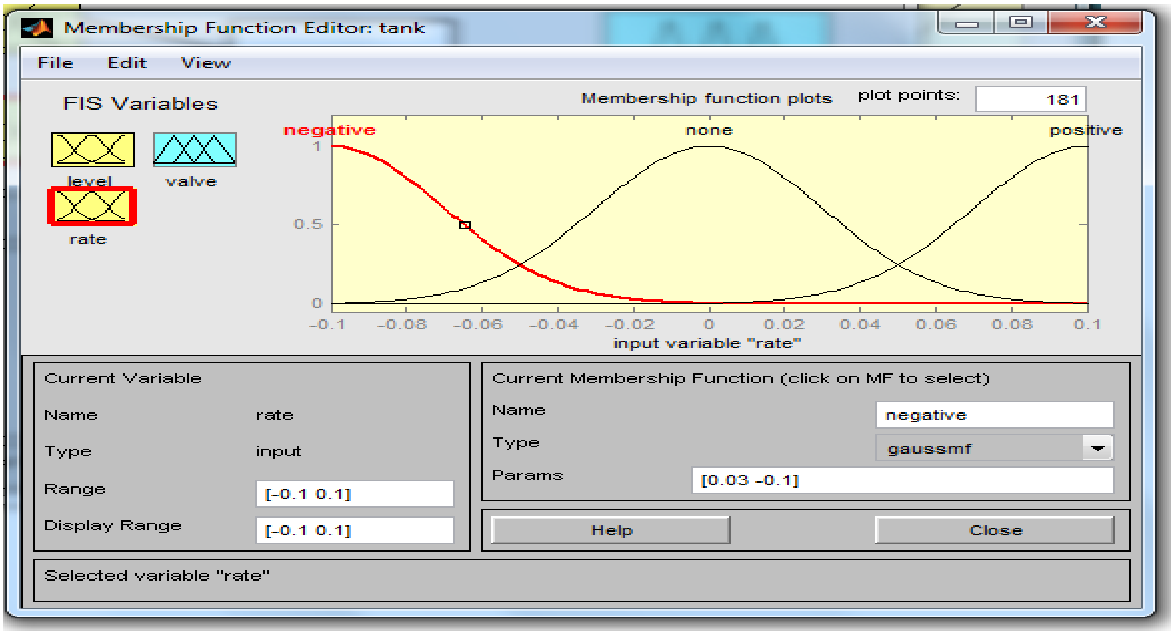


Figure 3.6: Membership function editor for the rate

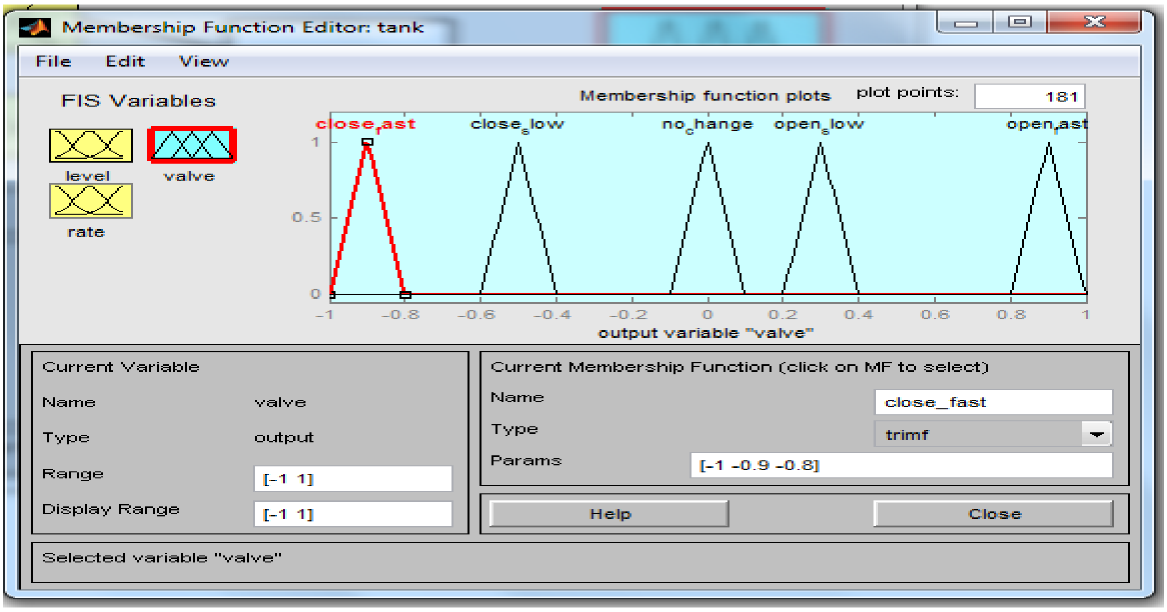


Figure 3.7: Membership function editor for the valve

### 3.3.3 The rule editor

Constructing rules using the graphical rule editor interface is fairly self-evident based on the descriptions of the input and output variables defined with the FIS. Editor the rule editor allows constructing the rule statements automatically by clicking and selecting one item in each input variable box one item in each output box and one connection item. Such as if (level is ok) then (valve is no change) the rest shown in Figure 3.8.

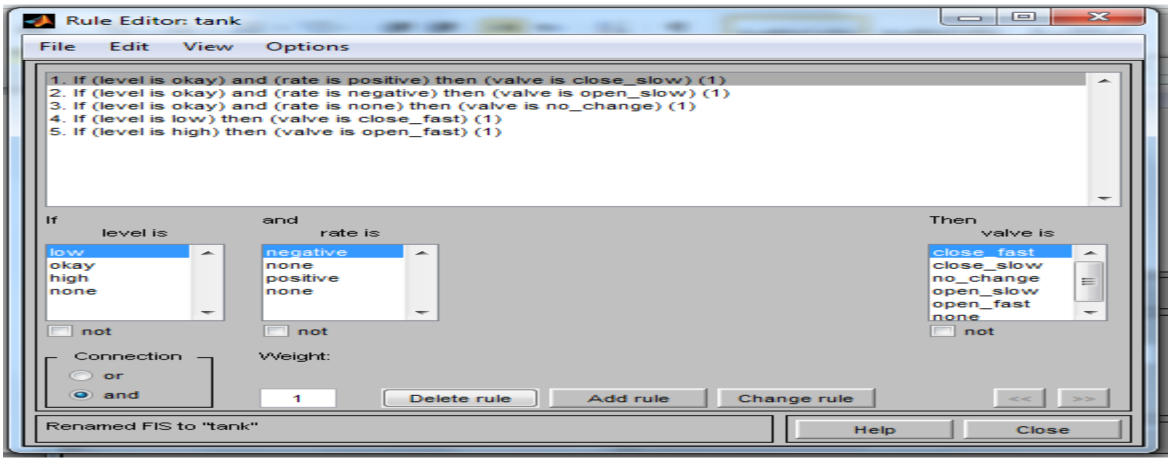


Figure 3.8: The fuzzy rule for fuzzy controller

The block diagram for fuzzy controller in MATLAB/SIMULINK and the rule viewer diagram are shown in Figure 3.9 and 3.10 respectively .

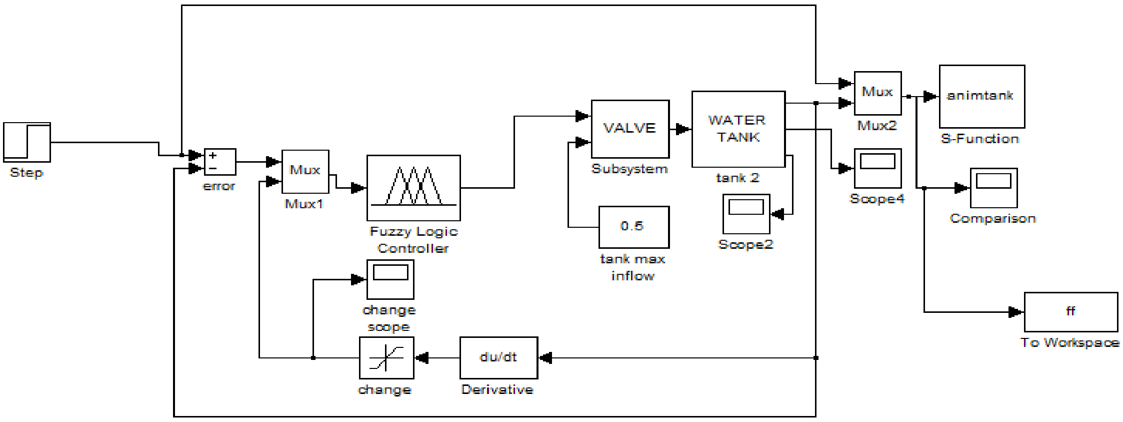


Figure 3.9: Model of fuzzy controller

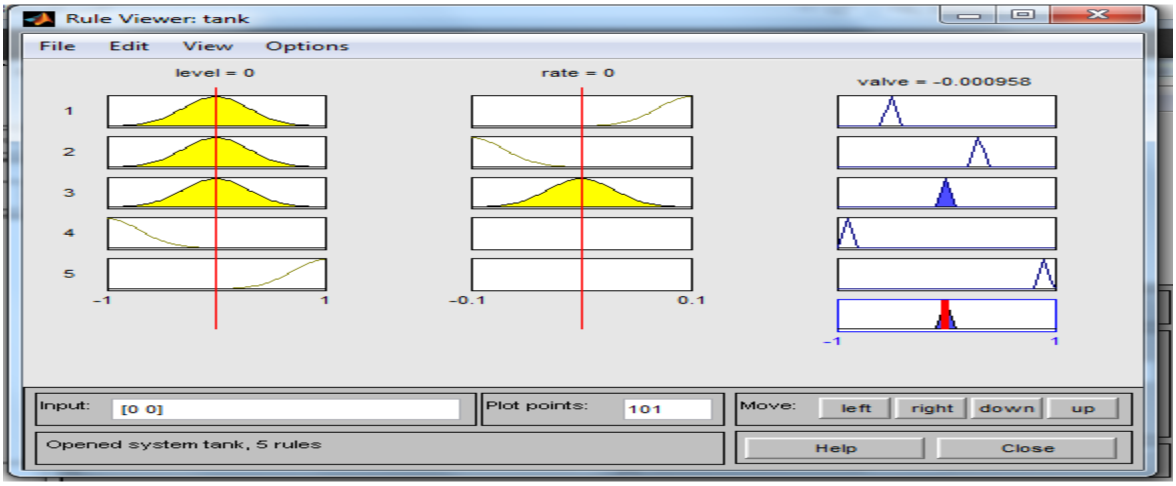


Figure 3.10: Rule viewer of fuzzy logic

Figure 3.11 shows the surface viewer indicating 3D graphical realization of the fuzzy rule base.

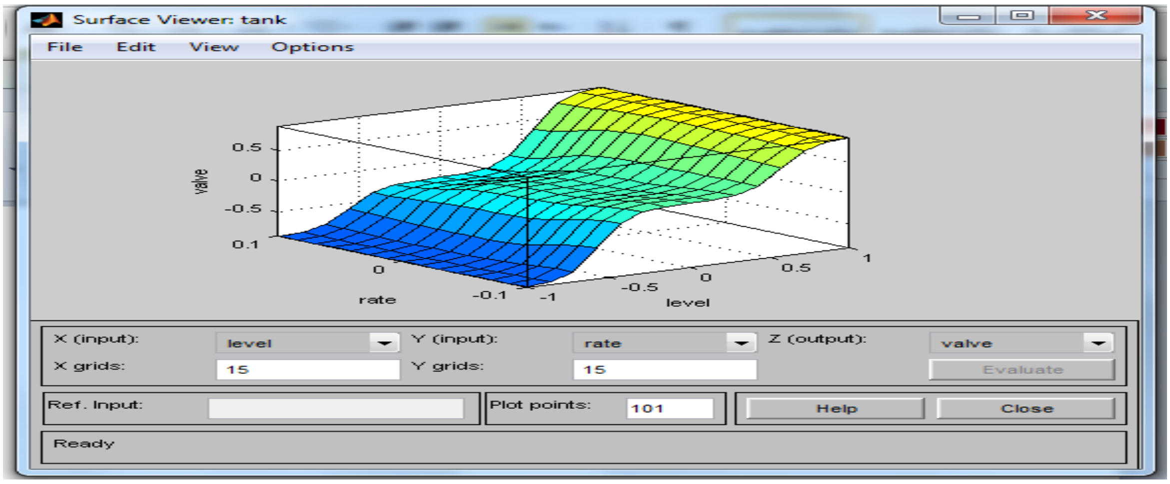


Figure 3.11: Surface viewer

## 3.4 Adaptive Neural Fuzzy Inference Systems Design

ANFIS editor window opens by typing "anfisedit" in MATLAB command window considering two inputs level and the rate and one output the valve. Among many FIS models the Sugeno fuzzy model is the most widely applied one for its high interpretability and computational efficiency and built in optimal and adaptive techniques. For a first order Sugeno fuzzy model a common rule set with two fuzzy if–then rules. Figure 3.13 shows the model of ANFIS controller.

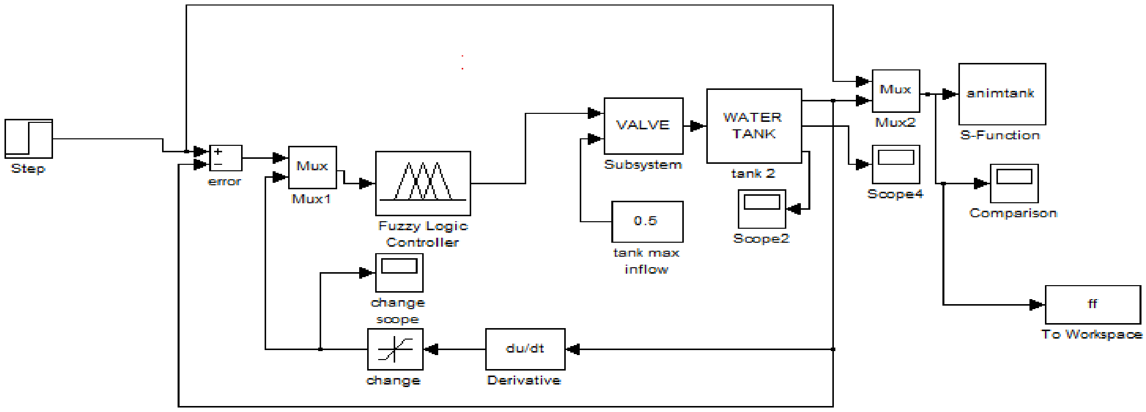


Figure 3.13: Model of ANFIS controller

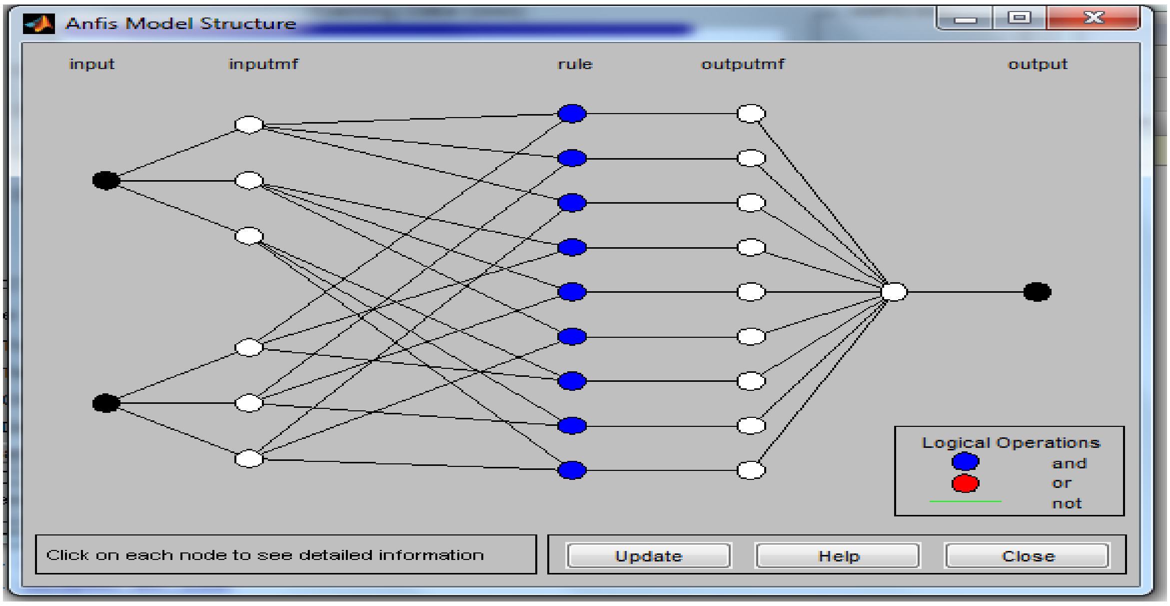


Figure 3.14: ANFIS model structure with two inputs and one output

The number of epochs was 300 for training. The number of MFs for the input variables level and rate is 3 and 3 respectively. The number of rules is 9. The Gaussian MF is used for two input variables. The data is loaded from fuzzy controller result.

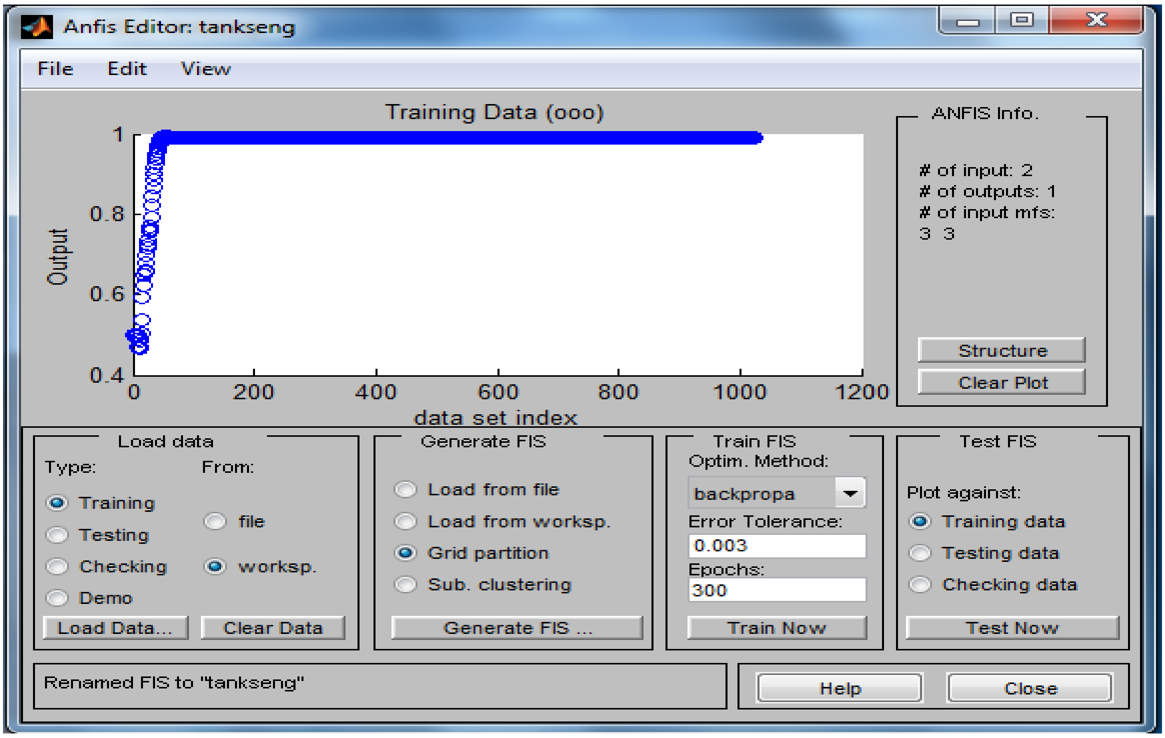


Figure 3.15: ANFIS editor training the rules using back-propagation algorithm

# CHAPTER FOUR

# RESULTS AND DISCUSSIONS

## 4.1 Simulation Result of Uncontrolled System

The SIMULINK model of the flow system without controller is shown in Figure 4.1. The time response with step input is shown in Figure 4.2.

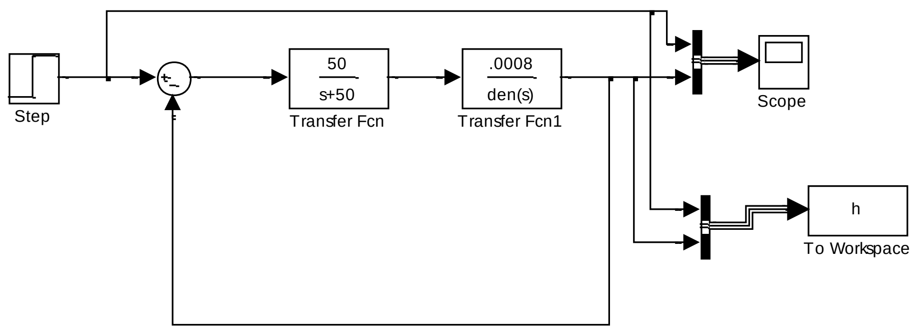


Figure 4.1: Module of flow system

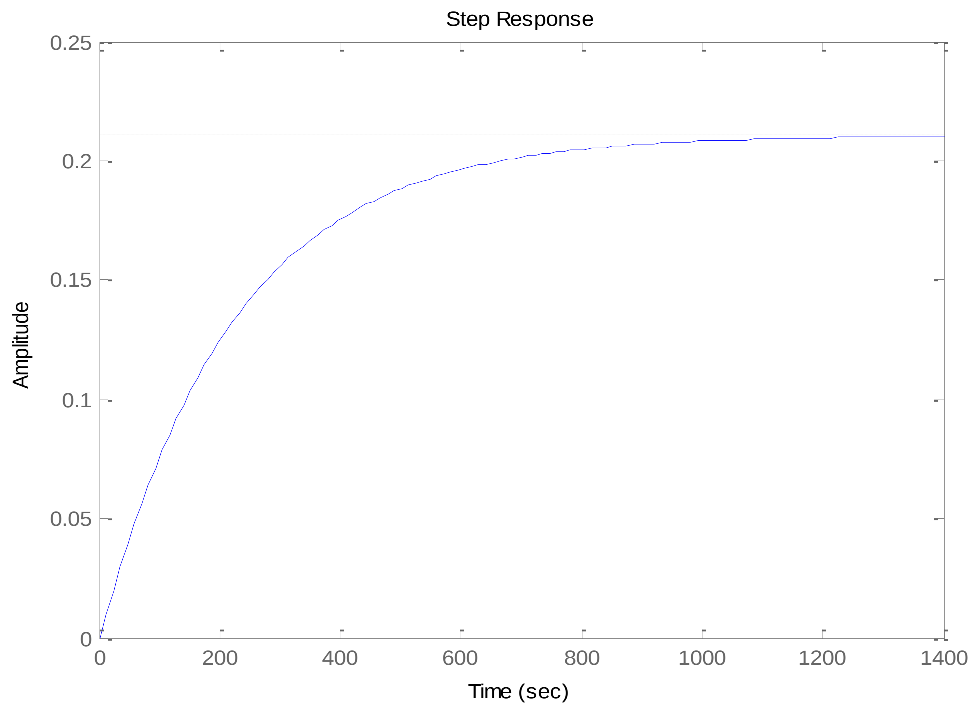


Figure 4.2: Result of uncontrolled system

## 4.2 Simulation Result of PID Controller

After running program 1 in command window, the transfer function is obtained as:

Transfer function:

$$\frac{4}{s^3 + 50.85\, s^2 + 42.5\, s + 0.15}$$

$k_{cr}$ = 54030
$p_{cr}$ = 0.9637
$k_p$ = 32418
$k_d$ =0.1205
$k_i$ =0.4819

By applying the above parameters values, the time response for the PID controller before tuning is shown in Figure 4.3.

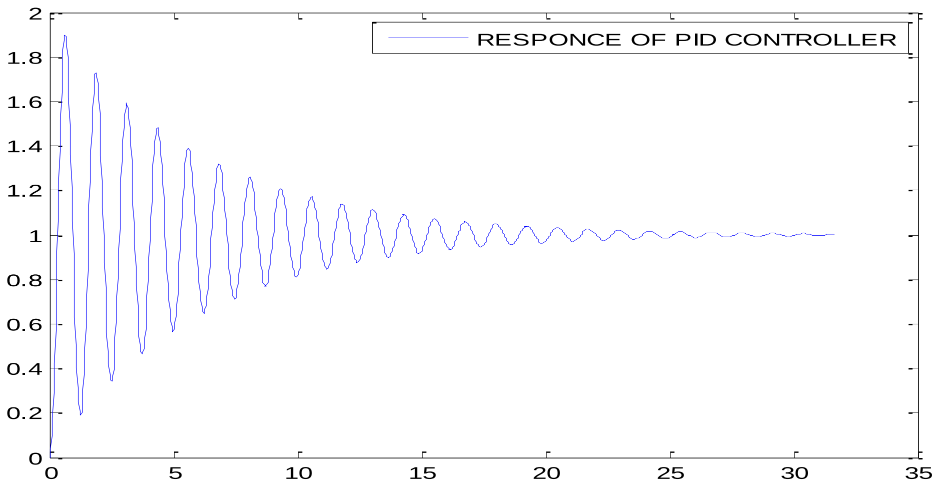


Figure 4.3: The response of PID controller before tuning

More tuning for the PID is applying using trial-and-error method the best values of $k_p$, $k_i$ and $k_d$ as shown in Figure 4.4.

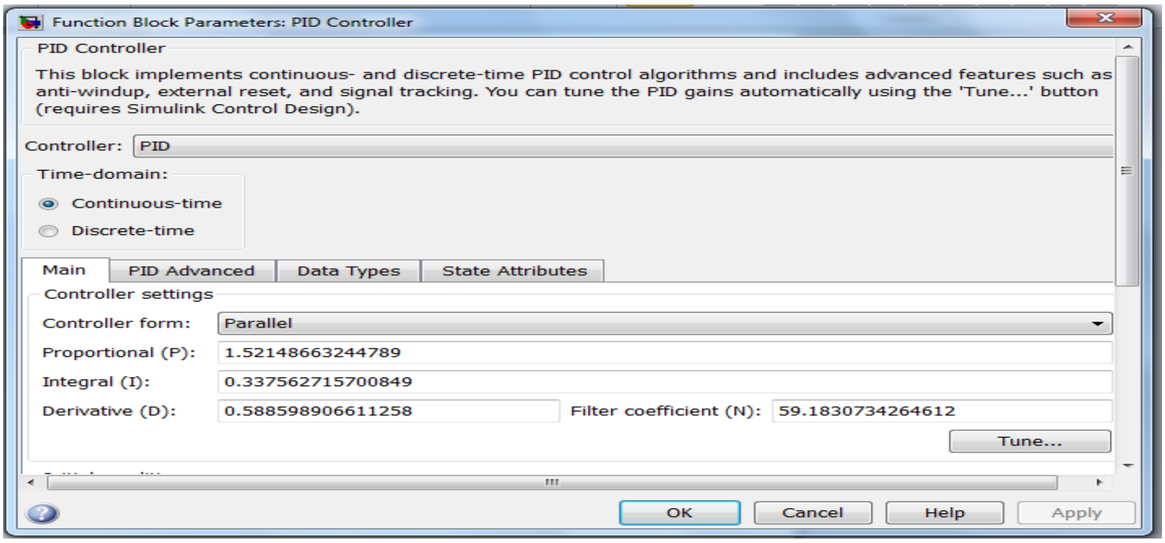


Figure 4.4: The value of $k_p$, $k_d$ and $k_i$ after tuning the PID

Figure 4.5 shows the response of the tuned PID controller when simulated with the given parameters. The graph shows that the controller has an overshoot and takes time to settle to the desired value of 1m. Figure 4.6 shows the comparison between the uncontrolled system and when using PID as a controller.

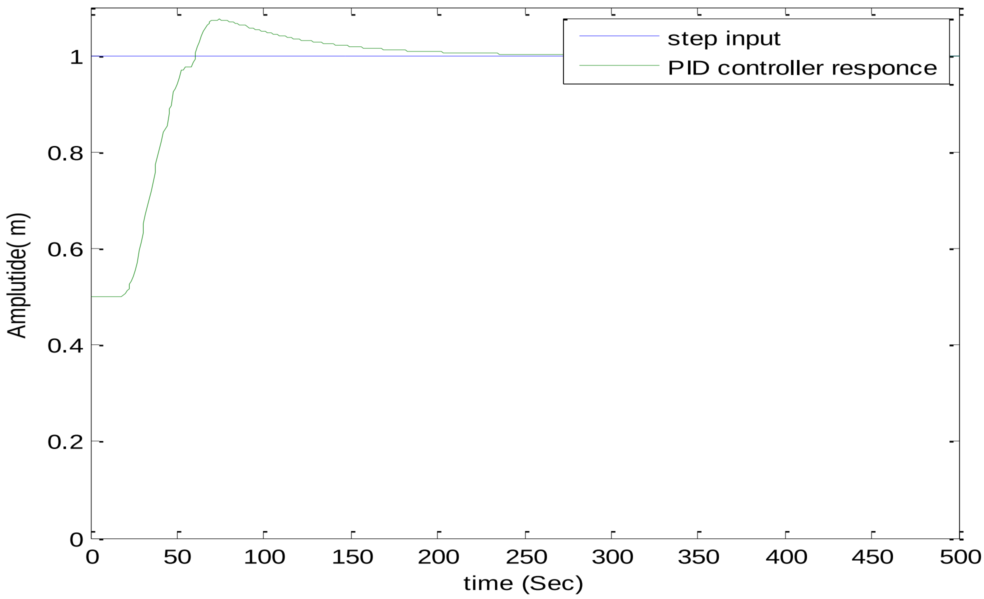


Figure 4.5: PID response

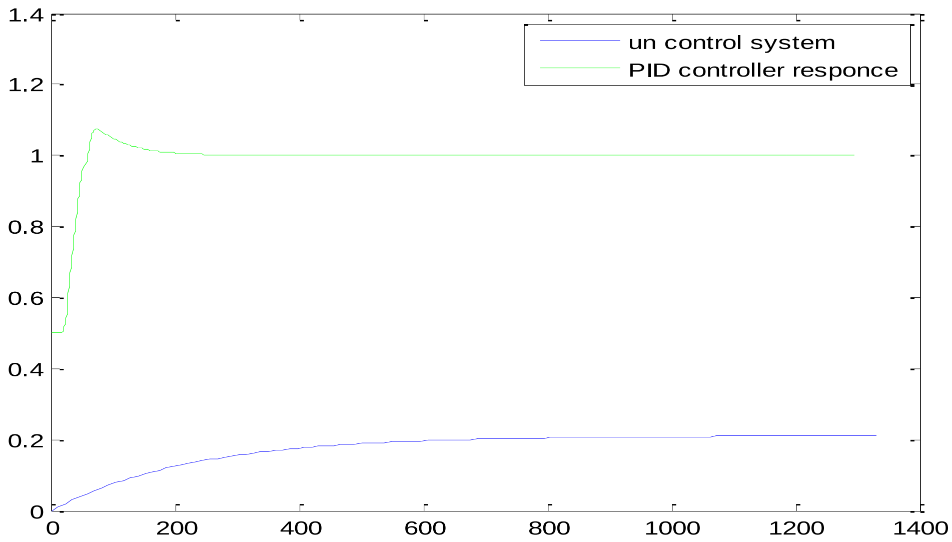


Figure 4.6: Comparison between PID response and uncontrolled system

Figure 4.7 shows the control signal of PID controller.

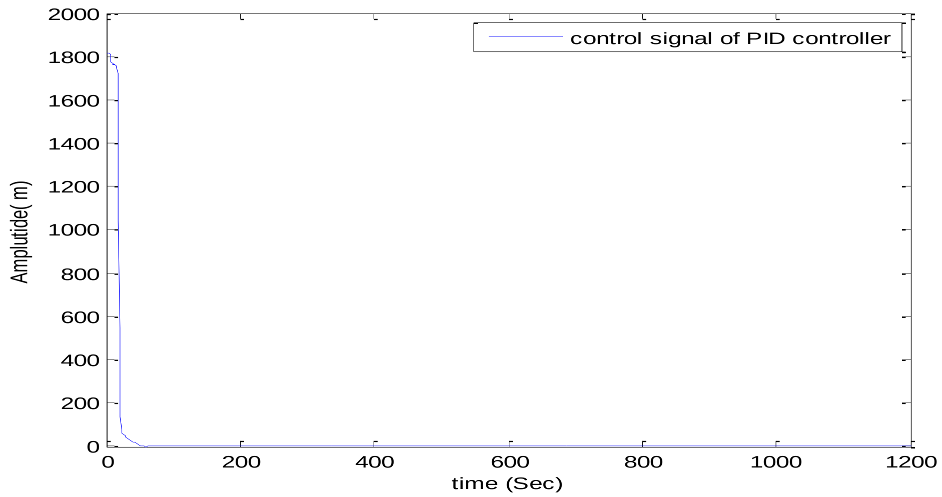


Figure 4.7: The control signal for PID controller

This system begins with high cost and decreased rapidly to zero which indicates the low cost of operation.

## 4.3 Simulation Result of Fuzzy Logic Controller

Two inputs and one output system is simulated with fuzzy logic toolbox in MATLAB. As explained in chapter three fuzzy levels are considered for each of the two inputs and five levels for the output parameter. Rule base consisting of five rules is activated to follow-up the desired liquid level. Figure 4.8 show the response of fuzzy controller. The controller settles at the desired water level very quickly (50 sec).

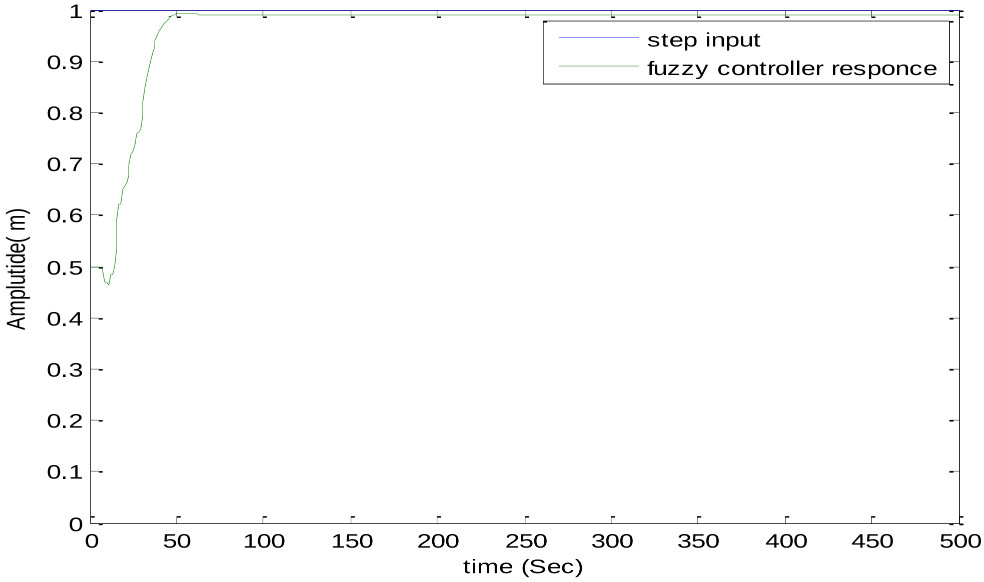


Figure 4.8: Fuzzy controller response

Figure 4.9 shows the control signal for fuzzy controller.

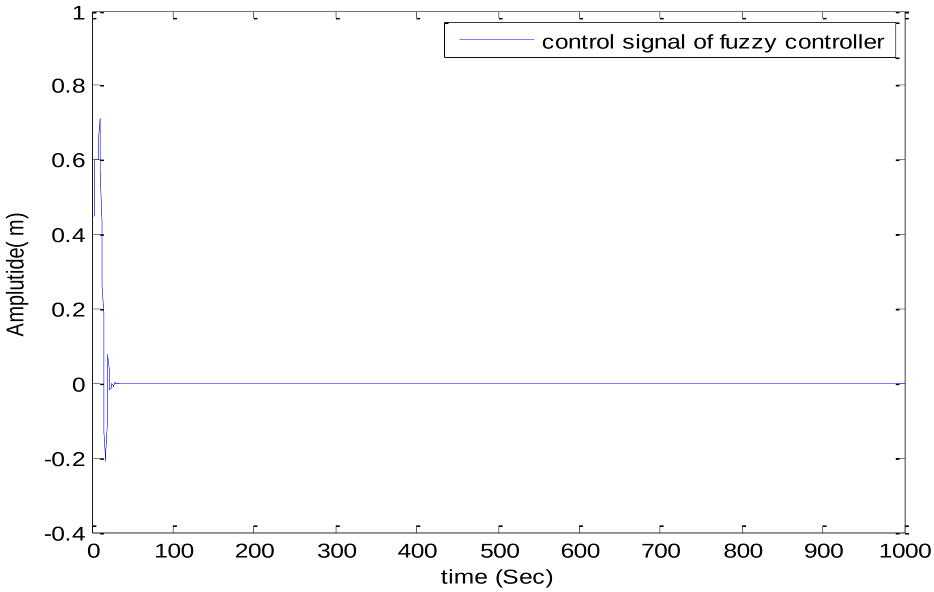


Figure 4.9: Control signal for fuzzy controller

Ti is clear from the graph that fuzzy controller has no offset

## 4.3 Simulation Result of ANFIS Controller

Two inputs and one output system is simulated with fuzzy logic toolbox in MATLAB. Figure 4.10 shows the result of ANFIS controller when using step input and it settle at desired value (1m) at time (20 sec).

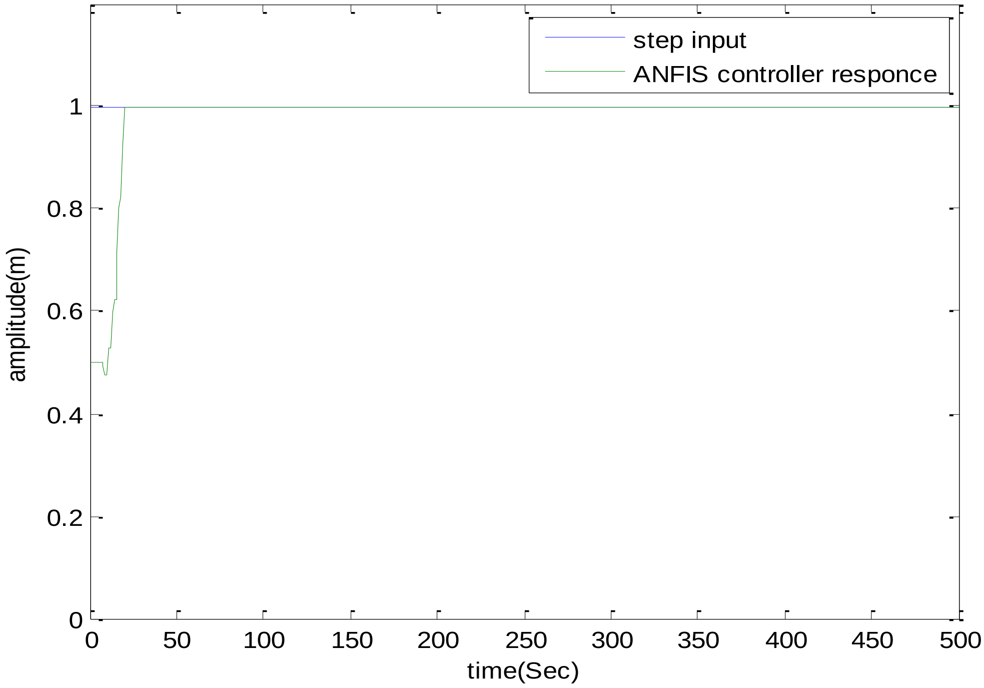


Figure 4.10: The result of ANFIS controller

Figure 4.11 shows the comparison of fuzzy and PID controller transient response for 1m desired level. It is clear from the graph that the PID controller has a large overshoot compared to the fuzzy controller and also takes a lot of time to settle at the desired level. Fuzzy logic on the other hand has little overshoot and steady state error and settles quickly providing accurate level control. The advantages and disadvantages of PID control and fizzy control just offset each other. The fuzzy controller can be used for rapid control and PID controller for accurate control The comparative shown that ANFIS controller is much better than fuzzy controller as it gives less rise time and settle faster than fuzzy controller.

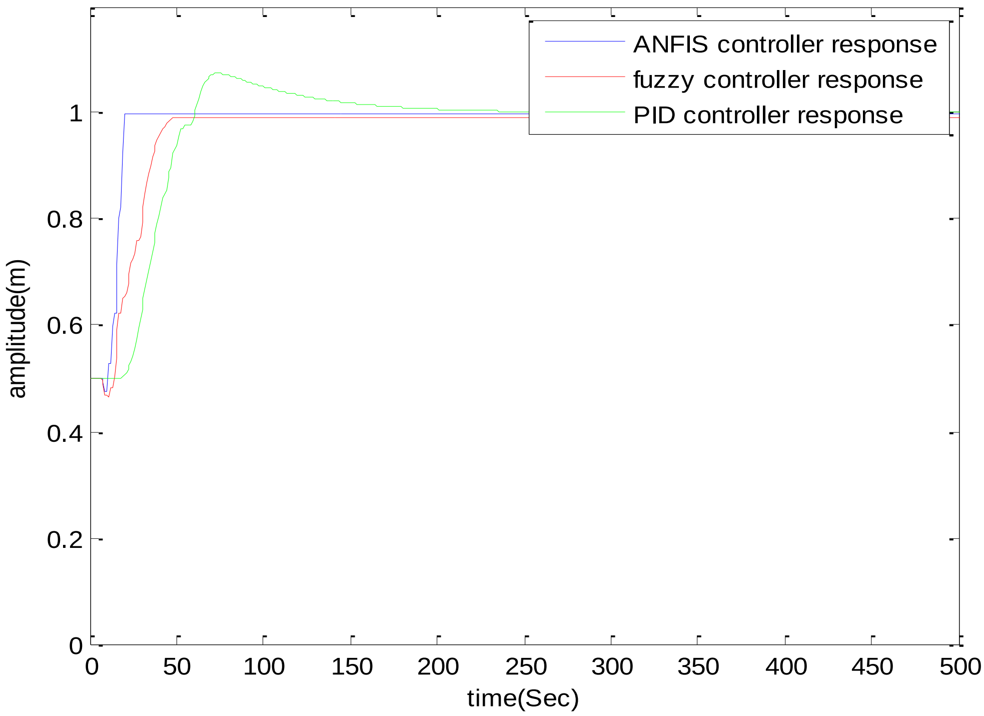


Figure 4.11: The response of PID, Fuzzy and ANFIS

Table 4.1: Performance of the controllers

| Type of Controller | Overshoot | Rise Time (Second) | Settling Time (Second) | Steady State Value |
|---|---|---|---|---|
| Un controlled | 0% | 489 | 800 | 0.211 |
| PID | 10 % | 40 | 120 | 1 |
| Fuzzy | 0% | 30 | 50 | 1 |
| ANFIS | 0% | 18 | 20 | 1 |

# CHAPTER FIVE

# CONCLUSION AND RECOMMENDATIONS

## 5.1 Conclusion

Fuzzy controllers have the advantage that can deal with nonlinear systems and implementation of human operator knowledge .fuzzy controller has to be compared it with classical PID controller. PID controller has three parameters require an adjustment. Controlled system shows good results in terms of response time and precision when these parameters are well adjusted. Fuzzy controller has a lot of parameters. The most important is to make a good choice of rule base and parameters of membership functions. Once a fuzzy controller is given the whole system can actually be considered as a deterministic system. When the parameters are well chosen, the response of the system has very good time domain characteristics. The fuzzy controlled system is very sensitive to the distribution of membership functions but not to the shape of membership functions.

PID controller can not be applied with the systems which have a fast change of parameters, because it would require the change of PID constants in the time. It is necessary to further study the possible combination of PID and fuzzy controller. It means that the system can be well controlled by PID which is supervised by a fuzzy system. According to the results of the MATLAB simulation the Adaptive Neuro-Fuzzy controller efficiently is better than the traditional fuzzy logic control (no overshoot, minimal rise time, Steady state error = 0). The performance of neuro–fuzzy is better because it determines the number of rules automatically reduces the computational time, learns faster and produces lower errors.

## 5.2 Recommendations

- As a future scope of this work the fuzzy logic controller can be implemented in a microcontroller with additional set of rules for more accurate control.
- The controllers can also be tested with periodically varying liquid level tracking applications.

# References

[1] Philip A. Adewuy, ”Paper on DC Motor Speed Control: A Case between PID Controller and Fuzzy Logic Controller”, Bells University of Technology, Ota, Nigeria, Vol. 4,No. 4, May 2013.

[2] Harshdeep Singh, ” Design of Water Level Controller Using Fuzzy Logic system”, National Statute of Technology Rourkela, 109ME0422.

[3] Y. Nakayama, ” Introduction to Fluid Mechanics”, Butterworth-Heinemann Linacre House, Jordan Hill, Oxford, 2000.

[4] Bernard Massey, ”Mechanics of Fluids”, Tavlor and Francis, London and New york, 2006.

[5] David Kaise, ” Fundamentals of Servo Motion Control”, Parker Compumotor, Vol. 64, 1942.

[6] Katsuhiko Ogata**,** “Modern Control Engineering**”,** University of Minnesota**,** Prentice-Hal, Fourth Edition, 2002.

[7] Ian C. Bruce, ” Control System Design”, Goodwin, Graebe, Salgado Prentice Hall 200, March 29, 2004.

[8] M.N. Cirstea, A. Dinu, J.G. Khor, M. McCormick, “Neural and Fuzzy Logic Control of Drives and Power Systems”, Newnes, Jordan Hill, Oxford, 2002.

[9] Kevin M. Passino, Stephen Yurkovich, ” Fuzzy Control”, Addison-Wesle,1998.

[10] Ra´ul Rojas, ” Neural Networks “, Springer-Verlag, Berlin, 1996.

[11] Gurpreet S. Sandhu and Kuldip S. Rattan, ”Paper On Design of a Neuro Fuzzy Controller”, Department of Electrical Engineering, Wright State University Dayton, Ohio 45435.

[12] Ama Albagul, ”Paper on Performance Comparison between PI and MRAC for Coupled Tank System”, Journal of Automation and Control Engineering, Vol. 2, No. 3, September 2014.