بسم الله الرحمن الرحيم

Sudan University of Science and Technology

College of Graduated Studies

A Comparison between Proportional-Integral-Derivative and Fuzzy Controllers for Water Tank

مقارنة بين المتحكم التناسبي-التكاملي-التفاضلي والمتحكم الغامض لخزان مياه

A Thesis Submitted in Partial Fulfillment for the Requirements of the Degree of M.Sc. in Electrical Engineering (Microprocessor and Control)

By:

Ahmed Elfadil Elnaeem Ahmed

Supervised by

Dr. Awadalla Taifour Ali

April. 2017

الآية

بسم الله الرحمن الرحيم

(إن في خلق السماوات والأرض واختلاف الليل والنهار لآيات لأولي الألباب)

سورة آل عمران الآية (191)

صدق الله العظيم

Dedication

Specially dedicated to

My beloved family and those people who have guided and inspired me

Throughout my journey of education

Acknowledgment

Firstly, I praise God who aids me to complete this research in this way and I would like to express my sincere thanks to the many people who have contributed to the success of this research, in particular my thesis supervisor Dr. Awadalla Taifour Ali, for his support, encouragement and professional assistance throughout the work of this research. Special thanks to all other Sudan University staff members that I may have called upon for assistance, I can’t end without thanking my best friend Abd Almajed Abd Allah and my family, on whose constant encouragement and love I have relied throughout my time during this work. Finally, I would like to thank the Sudan University of Science and Technology for accepting me in its graduate program and motivated me to do this work.

مستخلص

المنطق الغامض هو نموذج لطرق التصميم البديلة والتي يمكن تطبيقها في تطوير الأنظمة الخطية والأنظمة الغير خطية على حد سواء. أحد التطبيقات الناجحة التي يمكن إستخدام المنطق الغامض فيها هي مراقبة منسوب السوائل, حيث أصبح نظام التحكم في منسوب السوئل في تطور مستمر في السنوات الأخيرة نتيجة للتقدم الصناعي والعمراني والسعي للحصول على بديل يحل محل الطرق الميكانيكية والتقليدية السابقة من أجل تطويرموثوقية الأداء وخفض إستهلاك الطاقة. لذلك وللحصول على أفضل نظام يضمن الإستقرار لمستوى السائل اُستخدم في هذا المشروع نظام محاكاة لتصميم متحكم غامض لخزان مياه عن طريق التحكم الغامض بواسطة برنامج (الماتلاب). ومن أجل المحافظة على مستوى المياه في الخزان تم مراعاة بعض العوامل حيث تم إستخدام ثلاثة قواعد للمتحكم الغامض ثم زيدت الى خمس قواعد لضمان الإستقرارية. هذا المشروع يركزعلى الجانب البرمجي لتصميم المتحكم الغامض وسوف يكون عامل مساعد لتطوير الأنظمة المختلفة المتعلقة بالتحكم في مستوى السوائل التي تستخدم على نطاق واسع في الصناعة في هذه الأيام. في الفترات السابقة كان التحكم في منسوب السوائل عن طريق المتحكمات التقليدية التي تشكو من صعوبة تقدير الثوابت التي تضمن الأداء الأمثل للنظام خاصة في الأنظمة (الغير خطية) مما يؤدي للخلل في الأداء إذا اُختيرت بصورة خاطئة, بينما المتحكم الغامض لايتأثر بذلك لأن تصميم المتحكم الغامض لا يعتمد على النموذج الرياضي للنظام وثوابته بل على فهم طريقة أداء النظام (خواصه الديناميكية و الفيزيائية) ثم إستنتاج القواعد بناءً على ذلك بواسطة الخبراء لضمان إستقرارية النظام. ولذلك عند إستخدام المتحكم الغامض تم تجاوز هذه الأخطار حيث قدم تأثير سيطرة كبير مقارنة بالمتحكمات التقليدية و بالإضافة أيضا لسرعة أدائه وخفض تكاليفه. تم عمل تحليل ومقارنة في هذا المشروع بين المتحكم الغامض والمتحكم التقليدي نوع المتحكم (التناسبي-التكاملي-التفاضلي) حيث تفوق المتحكم الغامض عليه في كثير من الجوانب وأتضح أن المتحكم الغامض قام بخفض تجاوز الهدف الاقصى وقلل من الخطأ عند الإستقرار ومن زمن السكون وقدم تأثير سيطرة أبسط وأسهل مقارنة بالمتحكم (التناسبي-التكاملي-التفاضلي).

Abstract

Fuzzy logic is a paradigm for an alternative design methodology, which can be applied in developing both linear and non-linear systems for embedded control. One of successful application that can use fuzzy control is liquid level control. This control system keeps developing from time to time to replace the ordinary system which applies mechanical functions in its control in order to improve the system reliability. In order to find the best design to stabilize the liquid level, the purpose of this thesis is to design a simulation system of fuzzy logic controller for water tank level control by using simulation GUI package which in MATLAB software. The water level was controlled by using three rules of membership function which then extended to five rules for verification purpose. This thesis is focus to the software part only, by doing some modification the design will be very useful for the system relates to liquid level control that widely used in industry nowadays. For a long time, the choice and definition of the parameters of PID controller are very difficult especially in nonlinear systems. There must be a bad effect if parameters wasn't chosen nicely but the fuzzy controller can solve this problems because it's based on the operator's understanding of the behavior of the process (physical and dynamical response) instead of its detailed mathematical model to conclude the rules by the experts that match desired behavior. In this thesis, the water tank system was tested by both fuzzy and PID controller to analyze the control effect and comparing them with each other. As a result of comparing, fuzzy control is superior to PID control, which indicated that the fuzzy logic controller reduced the maximum overshoot and significantly reduced the steady state error and settling time as compared with PID controller.

List of Figures

List of Tables

Abbreviations

I/O: Input/output

FLT: Fuzzy Logic Toolbox

FLC: Fuzzy Logic Controller

FIS: Fuzzy Inference System

MFs: membership functions

PID: Proportional, Integral and Derivative

CoG: Center of Gravity

GUI: Graphical User Interface

LUTs: lookup tables

LI: Linguistic Input

LO: Linguistic Output

MAX: Maximum

MIN: Minimum

SCADA: Supervisory Control And Data Acquisition

ProbOR: Probabilistic OR

Table of Contents

Content Page No.

Chapter One: Introduction

Chapter Two: Theoretical Background and Literature Review

CHAPTER ONE

INTRODUCTION

1.1 General

Liquid level control is one of the most important aspects widely used in industrial process control and modern intelligent buildings. It got a wide range of applications like in nuclear power plants, chemical industries, water treatment system, boilers, etc. As the parameters of the plant changes frequently, maintaining the water level at its set point value is difficult. Hence, it is required to design a controller which controls the level of the liquid at its set point value and also it must accept the variable disturbances on the plant. PID controller is the earliest used controller in the industries. But it is difficult to get the efficient control because the PID gains setting is a challenging task. It is hard to control nonlinear processes with linear classic controllers. If the nonlinear process working point changes all the conventional controller parameters must be modified in order to keep the control performance. An interesting approach like fuzzy logic controller can be used to solve this problem.

1.2 Problem Statement

Water tank systems are used in many resident and industrial areas. Most of the time they are functioning well, but there is some condition that the system faces the problem of overflows because of its non-linear characteristics. The system cannot detect whether the water level have properly reach the desired level or the flow-in rate of the water are not proportional with the flow-out rate. It can cause the water tank system empties faster than it fills up. Consequently, the pressure of the water cannot support the distribution of the water. Construction of new tank, valve or system could be the solution to completely eliminate one these problems. However, such schemes are expensive and can also be extremely disruptive because the system networks may extend across wide geographical areas.

1.3 Proposed Solution

The implementation of the fuzzy logic controller in water tank level control system can be used to overcome this problem. Although there are a lot of method to control the water level but using fuzzy controller is more accurate and cheaper. One interesting features of applying fuzzy controller in non-linear system like this water tank system is the tank empties much slower than it fill up because of the specific value of the outflow diameter pipe as compare to speed of filling it up with the pump which can cause difficult in control process. This can be solved in easy way by setting of the valve membership function. And this is one advantage of fuzzy logic over another conventional method like PID controllers.

Although this project only focusing on software simulation not really on the hardware, but this is the best step that should be considered before any implementation of the system be constructed. By testing the system in this simulation area, the expected output from the input can be set earlier based on the rules set.

1.4 Objectives

The aim of this thesis is to perform a design simulation of fuzzy logic controller for stabilizing the water tank level control and examining whether it is better to handle this system which is done by using MATLAB/Simulink and the graphical user interface (GUI) tools provided by the fuzzy logic toolbox. This is a simple and easy approach to know more about water level system, including its level movements, valve setting, and data consistency and also about the rules of the variables.

The purpose of this study is to provide a feasibility study on the water level control simulation based on fuzzy logic system and compare it with conventional method like PID controller to see which is better for non-linear system like this. In order to achieve the study main objectives, there are some tasks that have to be done. The overall structure of the project will basically base on following:

- Understand the foundation of fuzzy logic which covers the introduction to the general concepts.
- Study and be familiar with MATLAB fuzzy logic toolbox.
- Building system with the fuzzy logic toolbox which goes into detail about the step taken to build and edit the fuzzy controller.
- Evaluate the result obtained from the simulation and compare it with the result obtained by using the same system driven by PID controller.

1.5 Methodology

The purpose methodology of this thesis is to design a simulation system of fuzzy logic controller for water tank level control by using simulation GUI package which is (Fuzzy Logic Toolbox and Simulink) in MATLAB software. This thesis is designed to make use of the great advantages of the fuzzy logic toolbox and integrate it with simulink. The fuzzy logic toolbox has the ability to take fuzzy systems directly into simulink and test them out in a simulation environment.

The simulation also will display the animation of the water tank level that controlled based on the rules of fuzzy sets. This project covers the processes of developing the application of fuzzy expert system in water tank level control. It starts from the theory until it implemented into the simulation environment.

In addition, this project also makes the analysis of the variety results that obtained from system. Different numbers of rules that used in the system will give the different result, so the analysis for results will be conducted.

Besides that, the same system will be also test by using PID controller the purpose is to find the best way to get the result as close as the requirement for stability of the level control for the water tank system and to compare between these two controllers methods.

1.6 Thesis Layout

This thesis consists of five chapters including chapter one. The content of each chapters are outlined as follows:

Chapter Two introduces the theoretical background knowledge and literature review of fuzzy logic Controller and gives a brief review of earlier related works done in liquid level control strategies.

Chapter Three discusses the fuzzy logic toolbox .This chapter describes all the terms that been used in the toolbox to created or edited the fuzzy Inference system (FIS).

Chapter Four deals with system modeling and design. This chapter describes all the terms and steps that been used for modeling and designing of the water tank system which controlled by fuzzy logic controller and PID controller for comparing reason and also shows the result of the system performance.

Chapter Five handles the conclusion and the recommendations of this thesis.

CHAPTER TWO

THEORETICAL BACKGROUND AND LITERTURE REVIEW

2.1 Introduction

Many engineering applications are concerned with level control. It may be a single loop level control or multi-loop level controls. The process industries such as refineries petrol, petro-chemical industries, paper making and water treatment industries require liquids to be pumped, stored in tanks, and then pumped to another tank. There are various approaches to the design of the liquid level controllers. Among them, the proportional-integral-derivative (PID) controllers have turned out to be famous for liquid level control [1].

PID controllers can be designed to sustain the level of liquid flow, but the limitation is its feedback type, the controller will take control action only after the output is affected by error. Also, it doesn't recognize the unanticipated alteration in the set point especially in time-varying and non-linear systems. Conventional control approaches are not convenient to solve the complex issues in this highly non-linear system. To overcome the difficulties innate in controlling liquid level, a controller based on fuzzy logic can solve these issues. Fuzzy logic control has emerged over the years and became one of the most active areas of research. There are many works in literature addressed the liquid level control issues using fuzzy logic, due to its simplicity [1].

Fuzzy controllers were developed to imitate the performance of human expert operators by encoding their knowledge in the form of linguistic rules [2].

It is a means of computing with expressions rather than equations. It enables computerized devices to reason more like humans, and imitates the capability to reason and use estimated data to find answers. It offers a wholly special approach to solve control problem which find in time-varying and non-linear systems, it is leading to new, more human, intelligent systems. [3]

2.2 PID Controller

The PID controller has been widely used in process industries, energy production, and transportation as well as in manufacturing. It is the most fundamental control strategy in the control area. PID controller is generally preferred for control actions because of its simple algorithm, ability to adapt to wide range of applications where it can ensure excellent control performances. PID controllers have survived from many changes in technology from mechanics and pneumatics to microprocessors. Especially, improvement of microprocessors has given a highlighted importance for the evaluation of the PID controllers. These improvements on the microprocessors have provided additional features on PID controllers such as automatic tuning, gain scheduling and continuous adaptation [4].

PID controllers can also be used in control systems where the precise mathematical model of the systems is not available and hence analytical design methods or conventional design methods cannot be used. Recent research has indicated that even though PID controllers may not provide optimal control, it provides satisfactory control [5].

Figure 2.1 shows the basic block diagram of PID controller where the error e(t) is the controller input and the actuator input is the controller output u(t) [5].

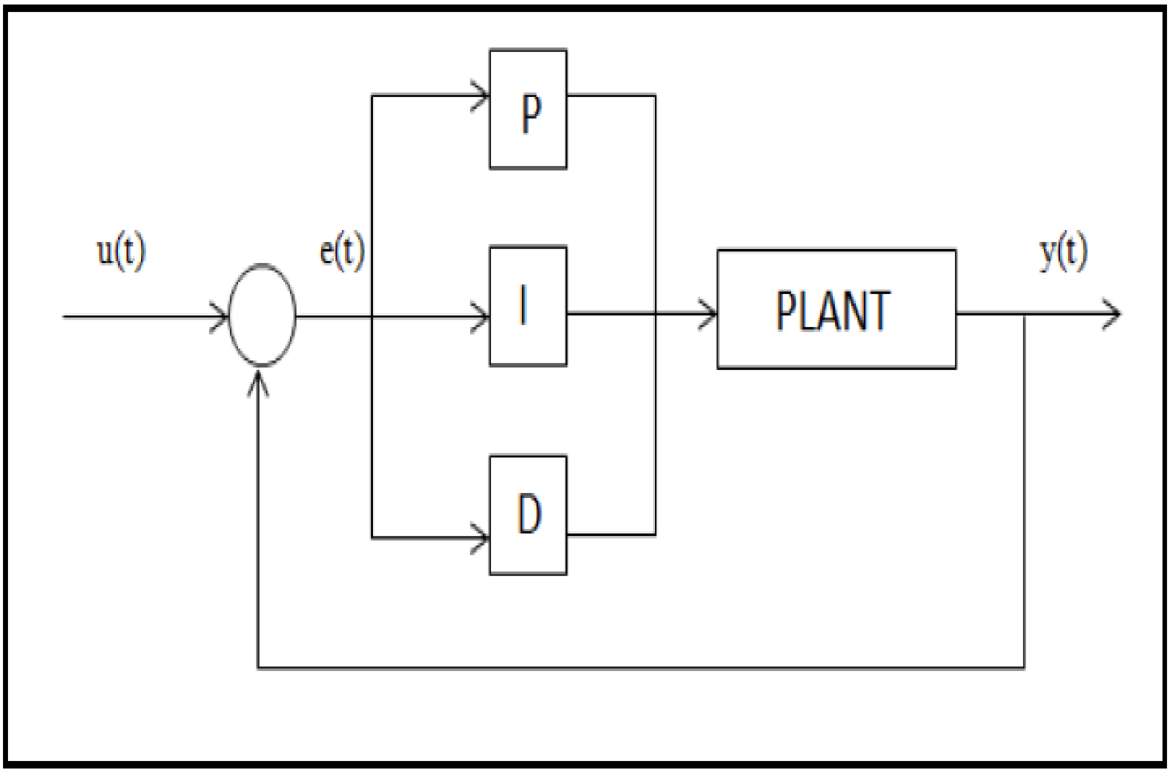

Figure 2.1: Basic block diagram of PID controller

The PID control law is stated in equation as shown:

$$u(t) = Kp\, e(t) + Kd\, \frac{de}{dt} + Ki \int e(\tau)d\tau \tag{2.1}$$

Where u(t) is control input, e(t) is error which is difference between system input and output.

The “P” term is proportional to the error, the “I” term which is proportional to the integral of the error, and “D” term refer to the derivative of the error. The controller parameters are called proportional gain, integral time and derivative time respectively. The proportional element gives a controller output depending on the magnitude of this error; integral action serves to eliminate steady-state error, while derivative action works to inhibit rapid changes in the process variable. These three control signals are combined to give a unified control effort [5].

2.3 Tuning Rules for PID Controllers

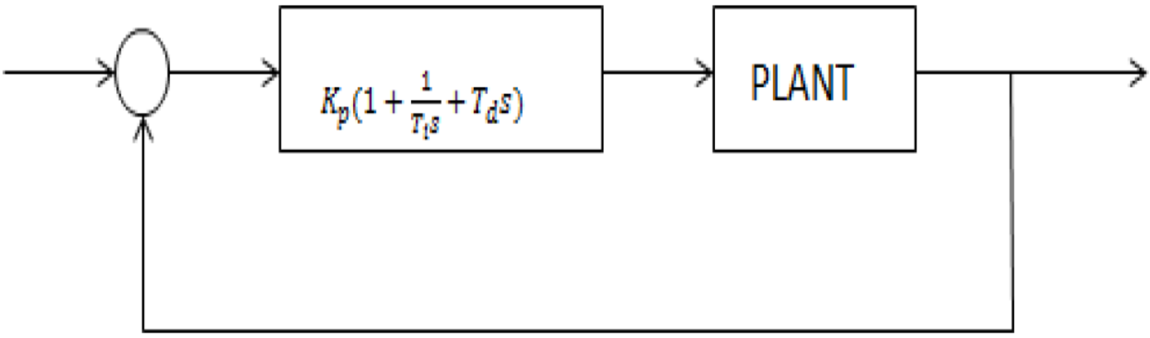

Figure 2.2: PID control of a plant

As shown in Figure 2.2, for a PID controller, the tuning of the parameters indicated in the controller block can be very challenging. If the mathematical model of a plant can be derived, then we can conclude that it is most likely to implement various design strategies called as a (fixed parameter tuning) methods, to find out the parameters of the controller that will meet steady state and transient specifications [6].

Nevertheless, if the mathematical model is not known or hard to derive then fixed parameter tuning methods can be applied just for only starting point and necessity of trial and error approach will be required without ensuring good performance. Therefore, we should go through heuristics approaches for tuning the PID parameters; some of the fixed parameter tuning methods was briefly reviewed in this thesis such as the Ziegler – Nichols method and Cohen – Coon method.

2.3.1 Ziegler – Nichols method

In 1942 G. Ziegler and N.B. Nichols published two tuning methods for PID controllers, the Ultimate Cycling method, and The Process Reaction-Curve method, often called the Ziegler-Nichols [7]

The Process Reaction-Curve method, often called the Ziegler-Nichols Open-Loop tuning method. Controller tuning means selecting the controller parameters that will meet given performance specifications. Ziegler – Nichols (ZN) is a tuning rule that proposes tuning strategy in terms of finding a set value for K_p, T_i and T_d based on experimental responses or based on the value of K_p. This method is implemented when mathematical model of a system cannot be derived. It needs to be noticed that Ziegler Nichols (ZN) method cannot guarantee minimum overshoot in the step response. This method provides only the starting point for obtaining the optimal PID parameters. We need a sequence of fine tunings until an acceptable result is obtained [7].

Ziegler – Nichols method offers two ways of implementing the tuning rules. In the first method, the step response of the plant is obtained experimentally. If the plant does not involve either integrators or dominant complex conjugates poles, the response will be seen as "S" shaped curve given in the Figure 2.3 [8].

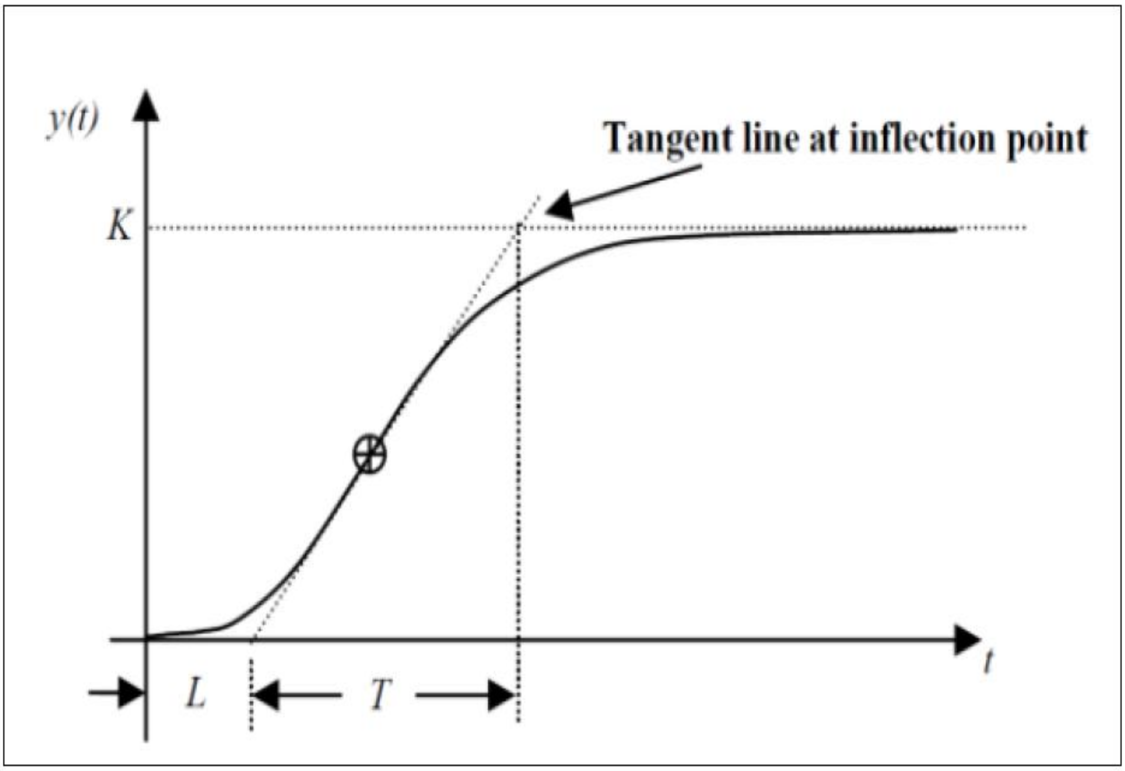

Figure 2.3: S-shaped response curve

The "S" shape curve is defined by two constants; delay time L and time constant T, which are derived by drawing a tangent line at inflection point of the curve. The intersection of tangent line and coordinate axes give the parameters T, L. ZN method gives PID parameters directly as functions of T and L stated in Table 2.1 [8].

Table 2.1: PID controller parameter obtained from ZN first method

Controller	K	Ti	Td
P	T/L	∞	0
PI	$0.9\,T/L$	$L/0.3$	0
PID	$1.2\,T/L$	2L	0.5L

In the second version of the Ziegler Nichols method, the plant controller parameters T_i and T_d are set to ∞ and zero respectively, which make the controller as proportional controller. When settings are done, K_p needs to be increased from zero to K_{cr} (ultimate gain) at which the output first start to oscillate. It needs to be noticed that if the output is not oscillatory for any gain value then this method cannot be implemented to the system. After finding the ultimate gain and ultimate period, the controller parameters can be calculated from Table 2.2 [8].

Table 2.2: PID controller parameter obtained from ZN second method

Controller	K	Ti	Td
P	0.5Kcr	∞	0
PI	0.45Kcr	$1/1.2\,Pcr$	0
PID	0.6Kcr	0.5Pcr	0.125Pcr

2.3.2 Cohen-Coon method

Based on the number of Google searches in 2010, the Cohen-Coon tuning rules are second in popularity only to the Ziegler-Nichols tuning rules [9]. Cohen and Coon published their tuning method in 1953, eleven years after

Ziegler and Nichols published theirs. The Cohen-Coon tuning rules are suited to a wider variety of processes than the Ziegler-Nichols tuning rules [10].

The Ziegler-Nichols rules work well only on processes where the dead time is less than half the length of the time constant [10].

The Cohen-Coon tuning method is based on the first order plus dead time delay process model with main design specification as quarter amplitude decay ratio in response to load disturbance. The main design objectives are to maximize the gain and minimize the steady-state error for P and PD controller. For PI and PID control, the integral gain is maximized. This corresponds to minimization of integrated error, the integral error due to a unit step load disturbance. For PID controllers three closed loop poles are assigned; two poles are complex and the third pole is located at the same distance from the origin as the other poles [11].

If the system can be defined by K_p, L and T, then it is possible to give tuning formulas with the help of Table 2.3 [11].

Table 2.3: : Controller parameters for Cohen-Coon method

Controller	K	Ti	Td
P	$\frac{1}{k}(\frac{\tau}{\theta}+0.35)$		
PI	$\frac{0.9}{k}(\frac{\tau}{\theta}+0.92)$	$\frac{3.3\tau+0.3\theta}{\tau+2.2\theta}\theta$	
PD	$\frac{124}{k}(\frac{\tau}{\theta}+0.13)$		$\frac{0.27\tau-0.09\theta}{\tau+0.13\theta}\theta$
PID	$\frac{1.35}{k}(\frac{\tau}{\theta}+0.18)$	$\frac{2.5\tau+0.5\theta}{\tau+0.61\theta}\theta$	$\frac{0.37\tau}{\tau+0.19\theta}\theta$

Where:

$$\theta = {}^{K_p L}/_{T} \tag{2.2}$$

$$\tau = {}^{L}/_{L + T} \tag{2.3}$$

It may be difficult to choose desired closed-loop poles for higher order systems if τ is small, controller parameters are close to others that are obtained by Ziegler Nichols tuning rules [11].

2.4 Fuzzy Logic

Fuzzy logic is a part of artificial intelligence or machine learning which interprets a human’s actions. Computers can interpret only true (0) or false (1) values but a human being can reason the degree of truth or degree of falseness. Fuzzy models interpret the human actions [12].

Fuzzy logic was first proposed by an American professor, Lotfi A. Zadeh, in 1965 when he presented his seminal paper on "Fuzzy Sets". Zadeh showed that fuzzy logic unlike classical logic can realize values between false (0) and true (1). Basically, he transformed the crisp set into the linguistic set. Zadeh introduced a new concept for applying natural language terms, and he became the master of fuzzy logic. Fuzzy sets thus have movable boundaries. The elements of such sets not only represent true or false values but also represent the degree of truth or degree of falseness for each input. Although an American professor introduced the fuzzy logic theory where it was first implemented in the control of cement kilns in Denmark in 1977 but the Japanese were successful in applying fuzzy logic in railway systems and commercial goods in 1988 like, automobiles and washing machines and their current fuzzy logic market is estimated around $2 billion dollar [13].

2.5 Fuzzy Set

In mathematics a set, by definition, is a collection of things that belong to some definition. Any item either belongs to that set or does not belong to that set. One of the most commonly used examples to compare real set with a fuzzy set is the set of tall people example as shown in Figure 2.4 [14].

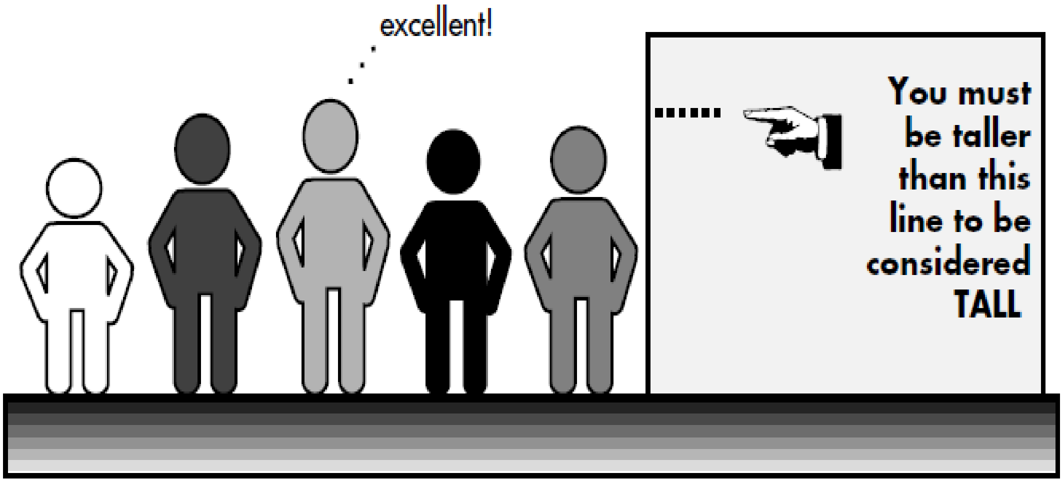

Figure 2.4: Set of tall people

In this case the universe of discourse is all potential heights, let said from 3 feet to 9 feet, and the word “tall” would correspond to a curve that defines the degree to which any person is tall. If the set of tall people is given the well-defined (crisp) boundary of a classical set, we might say all people taller than six feet are officially considered tall. But such a distinction is clearly absurd. It may make sense to consider the set of all real numbers greater than six feet is tall, but when we want to talk about real people, it is unreasonable to call one person short and another one tall when they differ in height by the width of a hair. But if the kind of distinction shown above is unworkable, then what is the right way to define the set of tall people? Here’s where fussy sets comes. From this another example of the set of tall people using normal set, we shall said that people taller than or equal to 6 feet are tall. This set can be represented graphically as shown in Figure 2.5 [14].

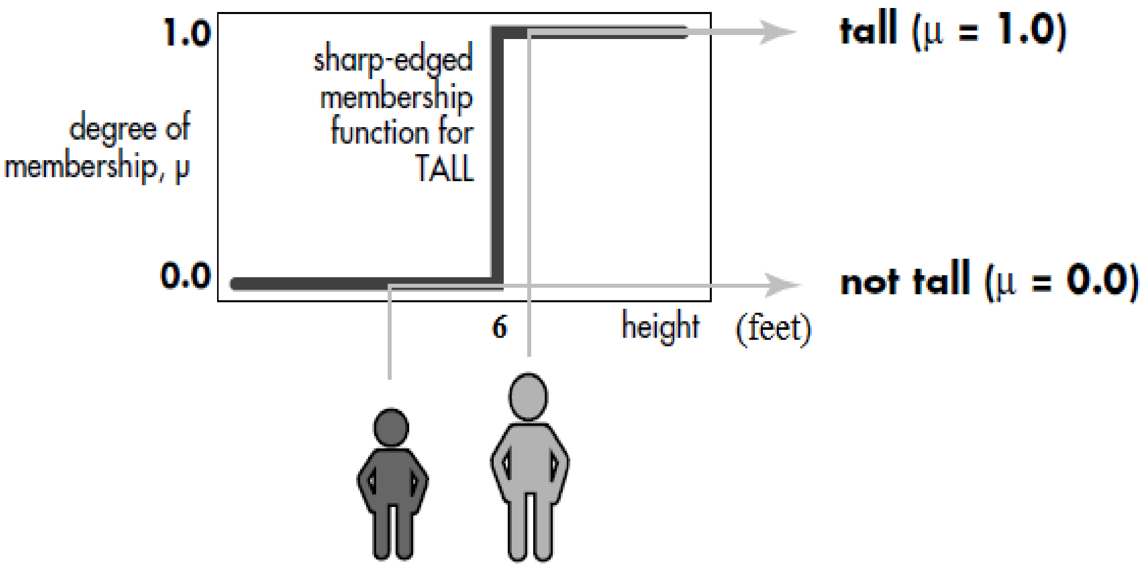

Figure 2.5: Normal set

The function shown above describes the membership of the 'tall' set, you are either in it or you are not in it. This sharp edged membership functions works nicely for binary operations and mathematics, but it does not work as nicely in describing the real world.

The membership function makes no distinction between somebody who is 6 feet and someone who is 10 feet, they are both simply tall. Clearly there is a significant difference between the two heights. The other side of this lack of distinction is the difference between a 5.9 feet and 6 feet man. This is only a difference of one inch, however this membership function just says one is tall and the other is not tall.

The fuzzy set approach to the set of tall men provides a much better representation of the tallness of a person as shown in Figure 2.6 [14]. Which showed a smoothly varying curve that passes from not-tall to tall? Both people are tall to some degree, but one is significantly less tall than the other. The output-axis is a number known as the membership value between 0 and 1.

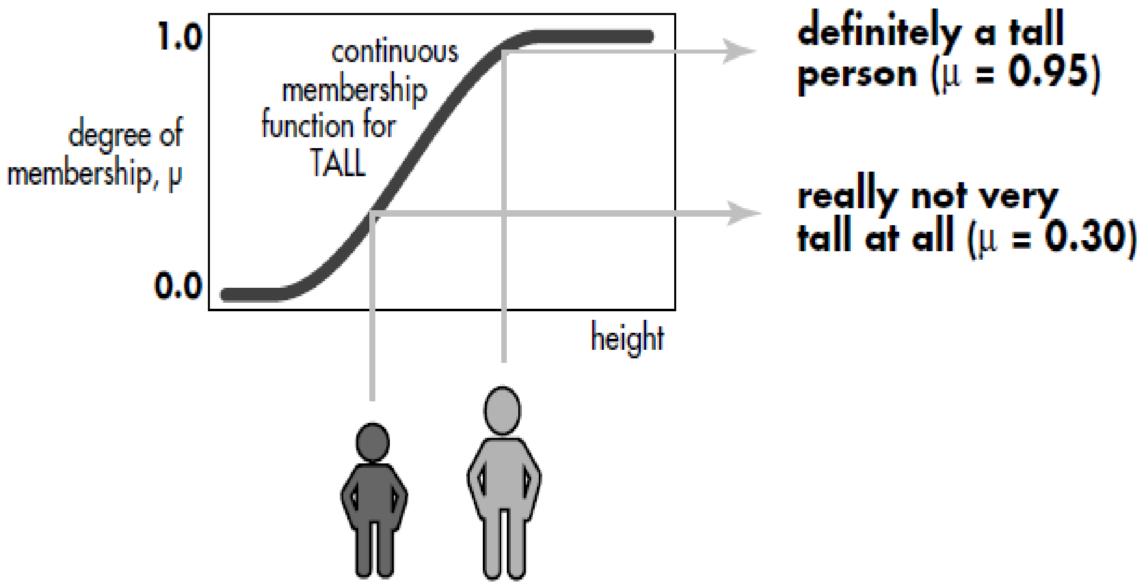

Figure 2.6: Fuzzy set

The curve is known as a membership function and is often given the designation of "μ". This curve defines the transition from not tall to tall. Both people are tall to some degree, but one is significantly less tall than the other [14].

2.6 Fuzzy Logic Controller

The fuzzy logic controller is mainly composed of four main components. First part is the "Fuzzification" which converts controller inputs into information that inference mechanism can easily perceive to activate and implement rules. Second part is the "Rule-base" where the knowledge is kept in the form of fuzzy logic sets of rules. Third part is the "Inference engine" which evaluates the expert's decision making in interpreting and deciding what the control input to the plant should be given. Last part of the fuzzy controller is the "Defuzzification" which converts fuzzy outputs decided by the inference engine into the crisp input to the plant. Figure 2.7 shows the internal structure of fuzzy logic controller [15].

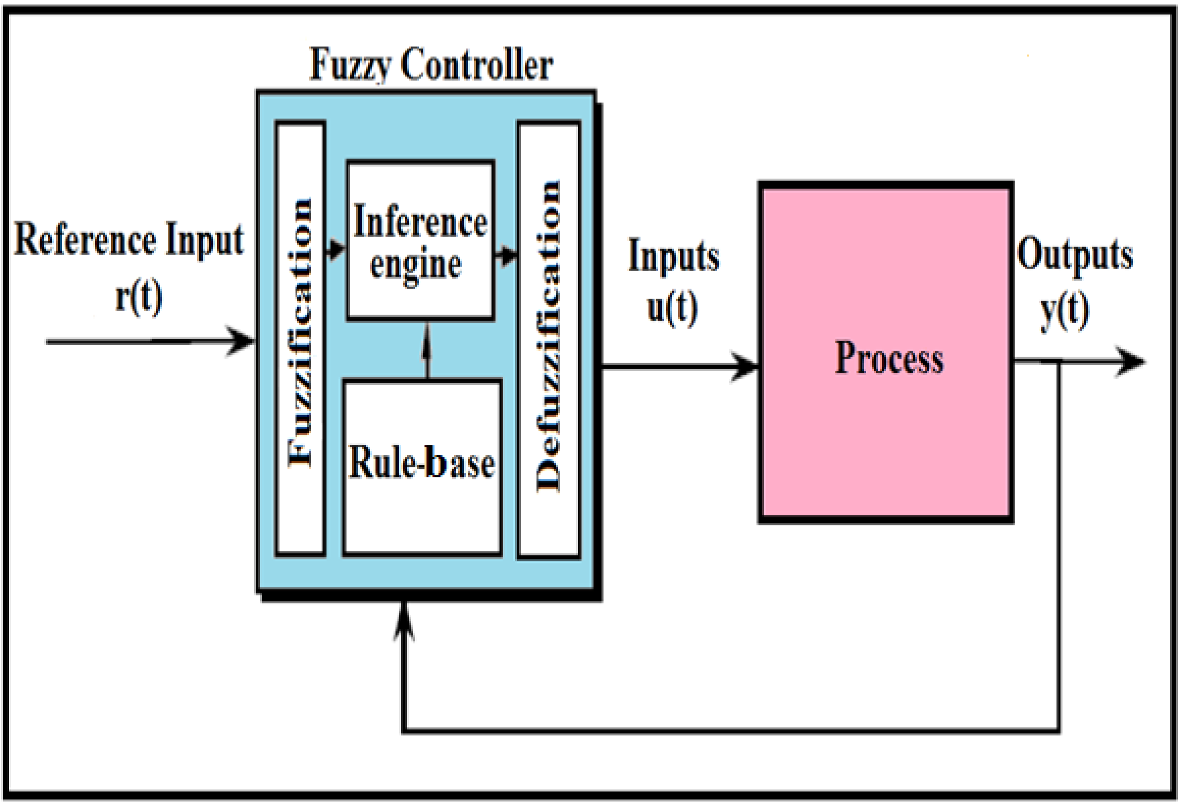

Figure 2.7: Internal structure of fuzzy logic controller.

2.6.1 Fuzzification

The process of transforming a crisp input variable into a fuzzy variable is known as "fuzzification". In another mean converting the crisp values into the grades of membership for linguistic terms of fuzzy sets [15].

The fuzzification interface involves the following 2 steps:

- **Fuzzify Inputs**

Measures the value of input variables and performs the function of fuzzification by converting the crisp values into the grades of membership for linguistic terms of fuzzy sets**.**

- **Apply Fuzzy Operator**

After the inputs are fuzzified and the degree is known for each part of the antecedent is satisfied for each rule. If the antecedent of a given rule has more than one part, the fuzzy operator is applied to obtain one number that represents the result of the antecedent for that rule. This numbers is then applied by AND (min) and OR (max) operation.

2.6.2 Rule-base

The rule base contains linguistic rules that are provided by experts. Fuzzy rules are linguistic IF-THEN constructions which have the general form of “if A then B” where A is called “antecedent” and B is called “consequent” which are condition and conclusion respectively [15].

2.6.3 Inference engine

The Inference engine has two main tasks:

- Implication
- Aggregation

The implication method is defined as the shaping of the output membership functions on the basis of the firing strength of the rule. The input for the implication process is a single number given by the antecedent, and the output is a fuzzy set [15].

Two commonly used methods of implication are:

- The minimum.
- The product.

The Aggregation is to combine the outputs of fired rules to obtain a fuzzy set as the overall output of the inference mechanism [16]. This output will be the input of the defuzification stage where it is converted to a crisp value.

Two commonly used methods of aggregation are:

- MIN method.
- MAX method.

2.6.4 Defuzzification

The output of the inference engine (the aggregated output membership functions of the fuzzy set) is the input of the defuzzification stage. The fuzzy set which is converted to a crisp value is calling defuzzification process. In order to obtain a crisp output, we need a defuzzification technique [16].

There are many different methods of defuzzification available like (maximum-decomposition method, height defuzzification method), but the most commonly used is the center of gravity method "CoG" in which the center of the mass of the result (the aggregated output fuzzy set) gives the crisp value [16]. This can be calculated as follows:

$$\text{CoG} = \frac{\sum_i Ci\ Ai}{Ai} \tag{2.1}$$

Where Ci the centre of the MFs is, Ai denotes the area of the MF, and i is the number of the MFs [16].

2.7 Fuzzy Inference Techniques

Fuzzy inference techniques that used for (firing the rules, Implication, aggregation or defuzzification process) can be done by one of these methods [17]:

a) Min method, min (A,B)
b) Max method, max (A,B)
c) Product Method, prod(A,B) = AB
d) Probabilistic OR**,** probor(A,B) = A + B - AB
e) Custom, In addition you can create your own methods by writing any function and setting that to be your method of choice.

2.8 Benefits of Fuzzy Logic

Fuzzy systems have been used in a wide range of applications in science, medicine, engineering, business etc. Fuzzy control has been successfully used in aircrafts, automobiles, manufacturing systems, process control and robotics [18].

Advantages of the fuzzy control can be summarized based on the discussions in literature and on newsgroups or other researches and papers where following reasons can be extracted:

i. Fuzzy logic is based on natural language. The basis for fuzzy logic is the basis for human communication, which make it easy to understand [19].

ii. Fuzzy logic can be blended with conventional control techniques. Fuzzy systems don't necessarily replace conventional control methods. In many cases fuzzy systems augment them and simplify their implementation [20].

iii. The development of fuzzy controllers requires less skilled personnel than the development of conventional controllers. Because it's based on the operator understands of the behavior of the process instead of its detailed complex mathematical equations [20].

iv. Fuzzy controllers are more appropriate to control nonlinear and time-varying systems than conventional controllers like PID controllers [21].

v. Fuzzy control is cheaper to design than conventional controllers [22].

In conventional and microprocessor control systems, inputs are mapped to outputs by one of two primary methods. First method, either the controller has a precompiled lookup tables "LUTs" which are used to perform all Boolean algebra operation that used by controller as shown in Figure 2.8 [22].

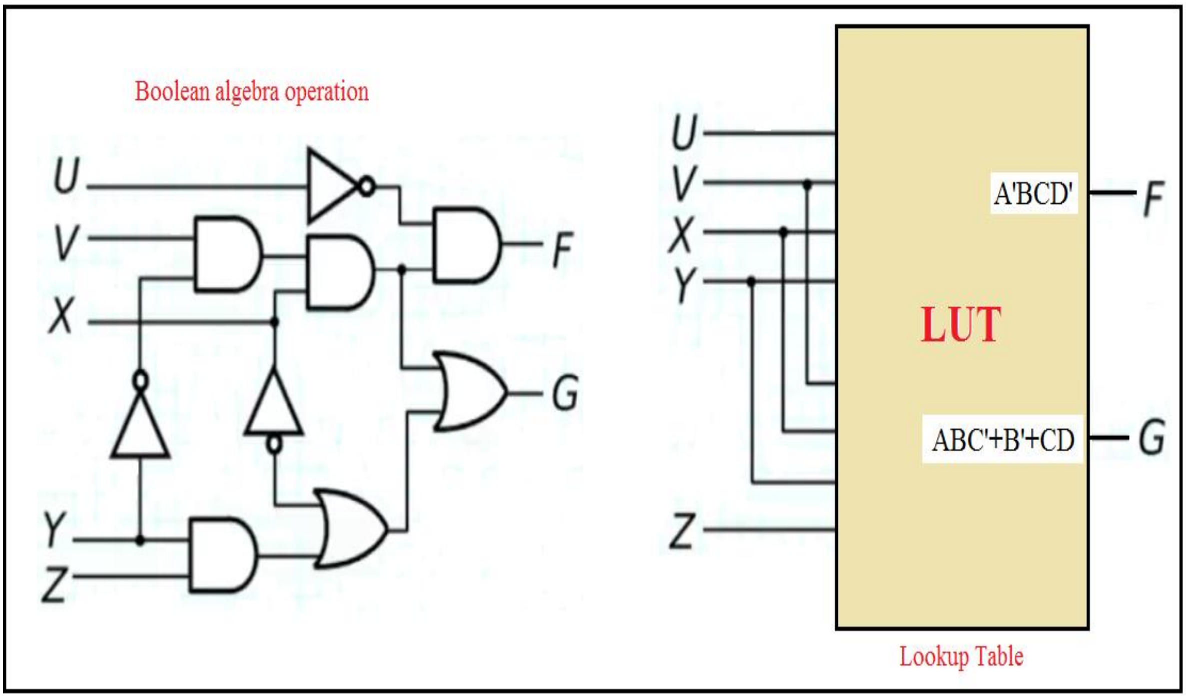

Figure 2.8: Lookup Table

Where the LUTs stored in RAM or ROM memory to relates an input state to the appropriate output state.

The second method, the controller performs some mathematical transform function via microprocessor which relates the input condition to the correct output condition [22].

In the LUTs implementation of a control system mapping, the computing of the functions only takes a single memory look up regardless of the complexity of the functions. When the controller senses a change in its input, it looks up the correct output condition for that input state, so it's responds with the pre-programmed response very fast. Since most microprocessors take only a few clock cycles to retrieve a value from a lookup table, this implementation allows the controller to quickly respond to changes at its inputs. Thus, the controller only has to wait a few clock cycles before knowing how to respond to an input condition. This can be a great advantage in real-time control systems where the controller must almost instantly respond to changes in the system [22].

However, these types of systems require a large amount of memory to implement the LUTs that contains every single input/output operations condition. So, systems incorporating lookup tables must have access to large amounts of RAM or ROM memory. This also requires a great deal of preplanning to insure that all possible input/output conditions are implemented into the LUTs. So, lookup tables for large complex systems can have the advantage of increased system response time, yet, they require large amounts of computer memory and can be difficult to implement, test and maintain [22].

In another hand some control systems use mathematical mapping functions to calculate the appropriate output for a particular input state. These systems require that a microprocessor recalculate an output state for every change in the input state. This type of system implementation does not need the large amount of RAM or ROM memory required to store large lookup tables. They only need enough memory to store the mapping formula and any required temporary variables [23].

While this type of system implementation does not require a large amount of RAM or ROM memory, it does require a significant amount of (processing power) to calculate the output result. This means a processor that can handle large complex mathematical operations must be selected. Thus, the microprocessor must execute many instructions to solve the input/output mapping function. This can result in delays between input state changes and output responses. A control system must either compensate for these processing delays or operate at higher clock frequencies. Also, a design engineer must be able to mathematically model a control system in order to develop an input/output mapping function. This can require a great deal of effort since not all systems can be modeled by simple mapping functions [23].

So, in the two traditional control system implementations, an engineer must either allow for large amounts of memory to incorporate lookup tables LUTs or choose a processor that can handle the computation needs of the mapping function. In both cases, the control system cost is increased. Fortunately, fuzzy logic provides for a workable compromise between large memory and high processing power requirements. Fuzzy logic control systems utilize both lookup tables, as well as mathematical functions to implement input/output mappings. Yet, both are used on a significantly smaller scale than in either of the traditional implementations [23].

In fuzzy logic systems, small lookup tables are used to hold the fuzzy input and output functions and relational rules. Then these simple rules are executed in order to perform input/output conversions. These conversions do not require a lot of heavy duty processing. This means that a fuzzy system must have enough memory to hold these tables and enough processing power to do these simple conversion calculations [23].

So fuzzy logic provides a control system platform that allows for a workable compromise between large precompiled lookup tables (LUTs) and complex mathematical mapping functions, thus allowing a designer to design hardware systems that require small amounts of memory and can still operate in microprocessors which amounts low processing power which make fuzzy controller design is low cost. Figure 2.9 shows comparisons between controller implementation methods [24].

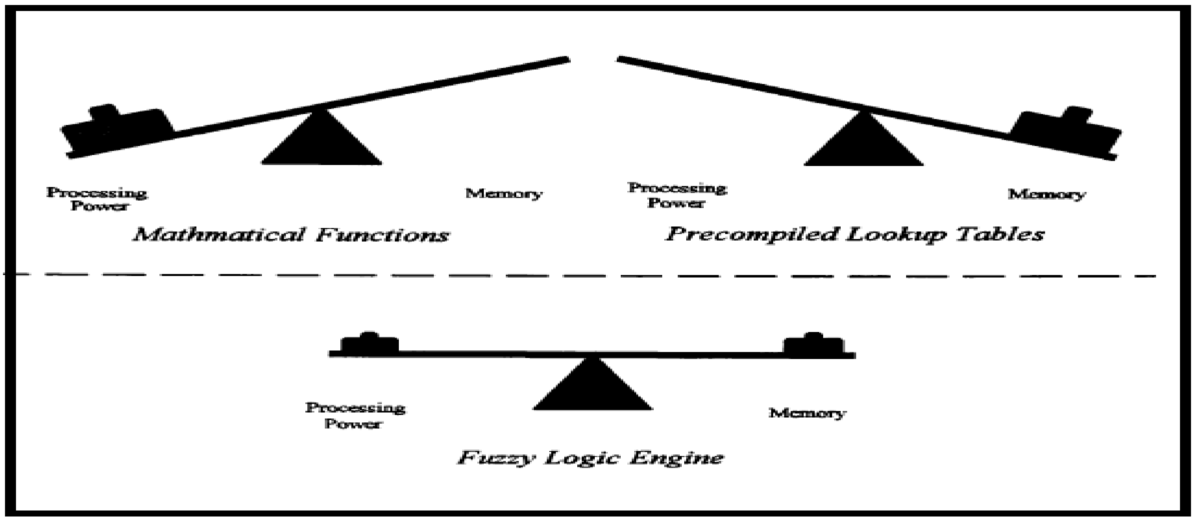

Figure 2.9: Comparisons between controller implementation methods

2.9 Related Works

Many earlier works dealt with various techniques of monitoring and controlling of liquid levels in industrial and domestic applications. Broadly this automatic control problem can be achieved under two means: mechanical methods and electrical methods [25].

Mechanical controls are used to control the simple level and flow system, for the examples: limit switches, mechanical valve, and electro-pneumatic valve. However, mechanical system could not give an accurate and precise output in controlling. Furthermore, the mechanical control performances are affected by the tear and wear process. Float ball type liquid level control is a popular method of mechanical control still used in practice for normal applications such as overhead tank overflow, restrictors, etc. The electrical methods of control include microcontroller-based circuits which automatically predict the liquid levels and accordingly active the circuit to operate motors [25].

In the previous years, there are many control strategy approaches for controller's design of liquid level control system such as:

- In 1986 Masato Kahara et al [26], presented a robust PID controller for multi-tank system.
- In 1993 Qiang Bi et al [27], presented a Multiple Model Predictive Control for level control.

Also there was some control strategies system that used fuzzy logic for controlling of liquid level control in the previous years such as:

- In 1998 Ming He et al 1988 [28], presented Fuzzy-PD control for multi-tank system.
- In 2002 Galzina et al [29], presented applied fuzzy logic for water level control in boiler drum and combustion quality control, fuzzy control rules

were extracted from operator knowledge based on relative ruling criteria for existing boiler room.

- Recently, In 2007 Satean Tunyasrirut et al [30], proposed a fuzzy logic controller and cascade controller which can control the liquid level in horizontal tank.

From all of previous presented paper and researches, various method and controller were used to control the water level especially that used fuzzy and PID controller; the following result can be extracted from previous paper and researches:

- Fuzzy controllers are more robust than other types of controllers because they can cover a much wider range of operating conditions than PID controllers and can operate with noise and disturbances, which can be done by editing the membership functions of fuzzy controller.
- Fuzzy controllers are more appropriate to control nonlinear and time-varying systems than conventional controllers like PID. As it seen from controlling the water tank system which have a nonlinear characteristics.
- Fuzzy controller design is cheaper than conventional controllers. As discussed before.

In this thesis, these results from past researches were noticed well and new result was added from this study that fuzzy logic controller is also more flexible than conventional controller like PID controller. With any given system, it's easy to massage it or layer more functionality on top of it without starting again from scratch because it's not requires focus on the model of the system but on the behavior of the model (dynamical and physical response of the system) to conclude the rules and this what fuzzy controller only need.

This is the reasons which make it more superior as compare to conventional controllers. So in this thesis the plant was modeled in MATLAB by simulink tools (graphically) using first principle (dynamical and physical response of the system) to prove that the fuzzy controller will control this system whatever the transfer function is exist or not, and this is the benefit of using fuzzy controller in nonlinear and complex mathematical model systems over conventional PID controllers.

CHAPTER THREE

FUZZY LOGIC TOOLBOX

3.1 Introduction

Fuzzy logic toolbox allows several things to be done, but the most important things are to be a place where a fuzzy inference system can be created or edited [31].

Although it is possible to use the fuzzy logic toolbox by working strictly from the command line, in general it is much easier to build a system graphically. So let’s discuss MATLAB fuzzy inference system that given by fuzzy logic toolbox.

3.2 Fuzzy Inference System

There are five primary GUI tools for building, editing, and observing fuzzy inference systems in the fuzzy logic toolbox as shown in Figure 3.1 [32].

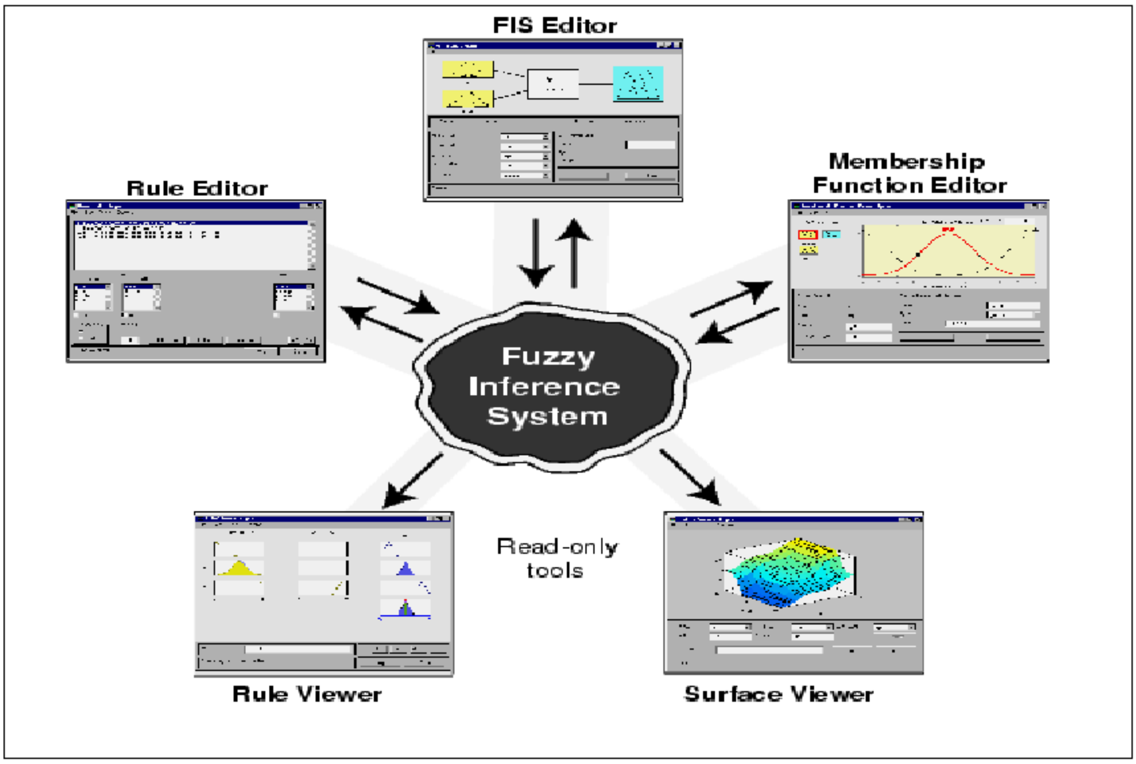

Figure 3.1: Fuzzy inference systems

These GUIs are dynamically linked, if any changes made to the FIS that using one of them can affect to the other opened GUIs. You can have any or all of them open for any given system. These five primary GUIs can all interact and exchange information. Any one of them can read and write both to the workspace. For any fuzzy inference system, if more than one of these GUIs editors was opened for a single system, the various GUI windows were aware of the existence of the others, and will, if necessary, update related windows. Thus if the names of the membership functions are changed using the membership function editor, those changes are reflected in the rules shown in the rule editor [32].

The FIS editor, the membership function editor, and the rule editor can all read and modify the FIS data, but the rule viewer and the surface viewer do not modify the FIS data in any way [33].

3.2.1 FIS editor

The FIS editor handles the high-level issues for the system. “How many inputs and output variables?”, “What are their names?” .The fuzzy logic toolbox doesn’t limit the number of inputs. However, the number of inputs may be limited by the available memory of your machine. If the number of inputs is too large, or the number of membership functions is too big, then it may also be difficult to analyze the FIS using the other GUI tools. Figure 3.2 shows FIS editor [34].

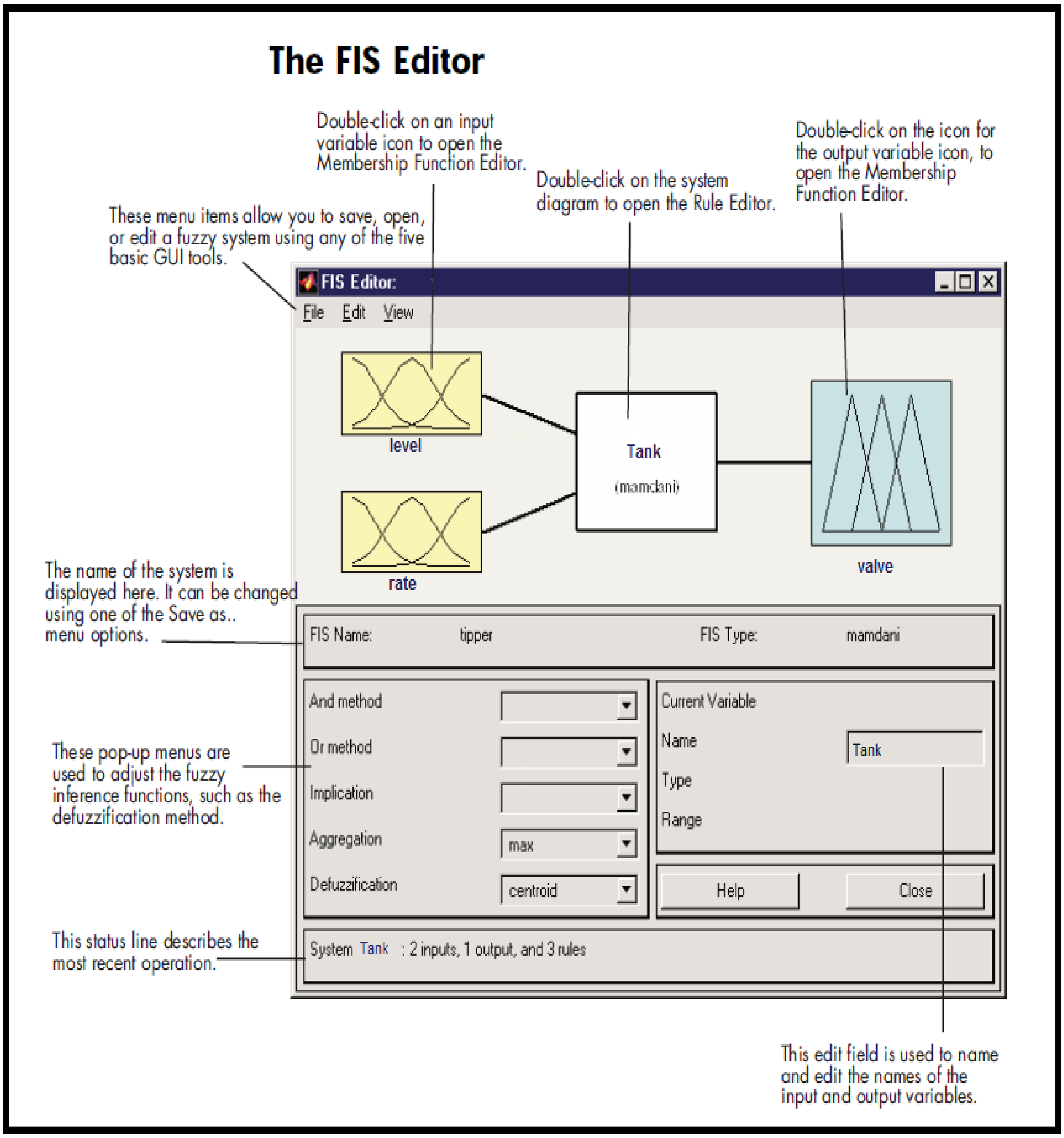

Figure 3.2: FIS editor

3.2.2 Membership function editor

The membership function editor is used to define the shapes of all the membership functions associated with each variable as shown in Figure 3.3 [34].

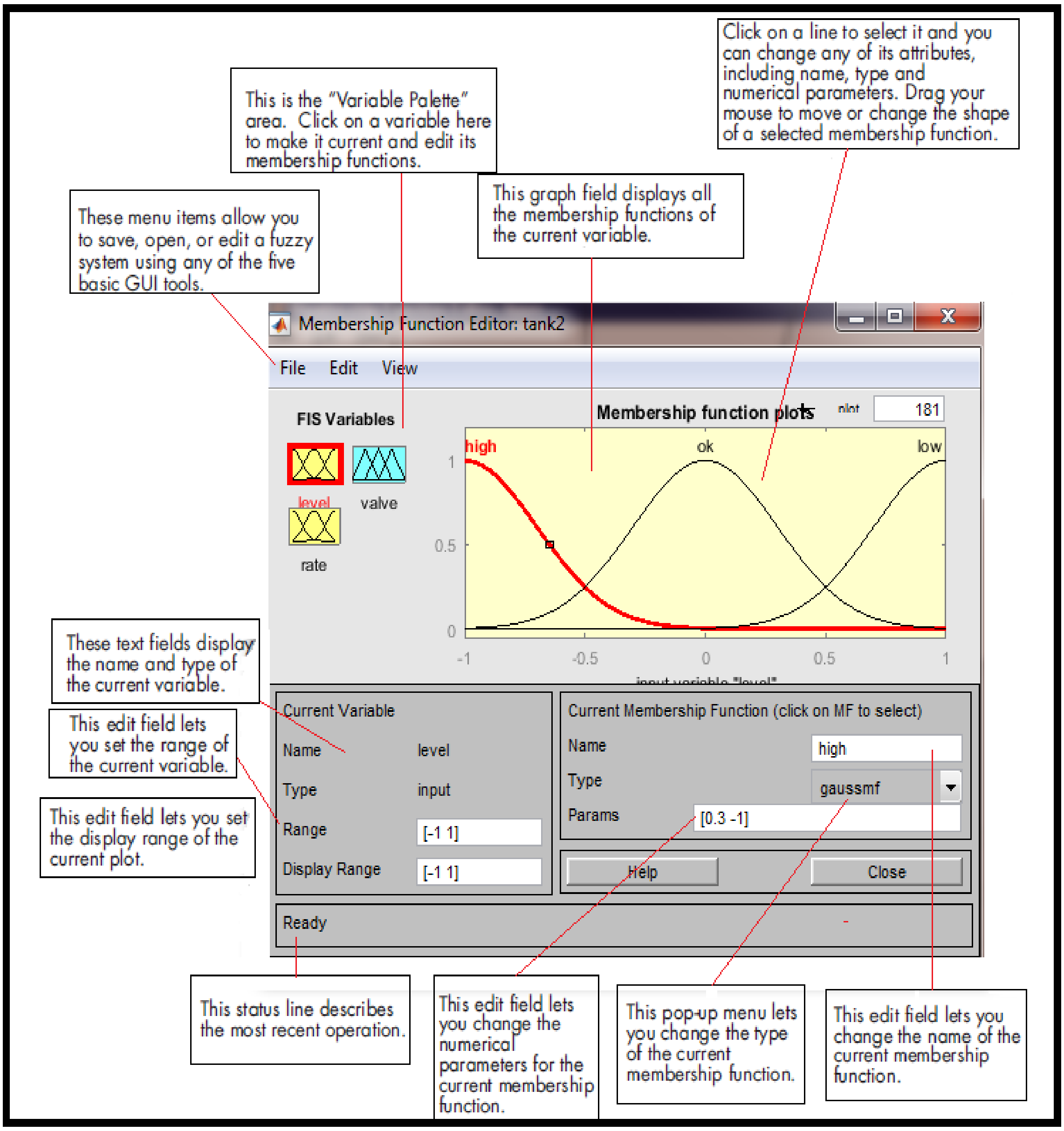

Figure 3.3: Membership function editor

The membership function shares some features with the FIS editor. In fact, all of the five basic GUI tools have similar menu options, status lines, and help and close buttons. The membership function editor is the tool that lets you display and edits all of the membership functions associated with all of the input and output variables for the entire fuzzy inference system. When the membership function editor are opened to work on a fuzzy inference system that does not already exist in the workspace, there is not yet any membership functions associated with the variables that have been defined with the FIS editor [34].

On the upper left side of the graph area in the membership function editor is a ‘’variable palette” that set the membership functions for a given variable. To set up membership functions associated with an input or an output variable for the FIS, an FIS variable in this region by clicking on it. Next by selecting the edit pull-down menu, and choosing add MFS, a new window will appears, which allows selecting both the membership function type and the number of membership functions associated with the selected variable [34].

In the lower right corner of the window is the control that permits change of the name, type, and parameters (shape), of the membership function, once it has been selected. The membership functions from the current variable are displayed in the main graph. These membership functions can be manipulated in two ways: Using mouse to select a particular membership function associated with a given variable quality, (such as high, for the variable, level), and then dragging the membership function from side to side. This will affect the mathematical description of the quality associated with that membership function for a given variable [34].

The selected membership function can also be tagged for dilation or contraction by clicking on the small square drag points on the membership function, and then dragging the function with the mouse toward the outside, for dilation, or toward the inside, for contraction. This will change the parameters associated with that membership function. Below the variable palette is some information about the type and name of the current variable. There is a text field in this region that lets you change the limits of the current variable’s range (universe of discourse) and another that lets you set the limits of the current plot (which has no real effect on the system) [34].

3.2.3 Rule editor

The rule editor is used for editing the list of rules that defines the behavior of the system [35]. 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 for constructing the rule statements automatically, by clicking on and selecting one item in each input variable box, one item in each output box, and one connection item. Choosing none as one of the variable qualities will exclude that variable from a given rule. Choosing not under any variable name will negate the associated quality. Rules may be changed, deleted, or added, by clicking on the appropriate button. The rule editor also has some familiar landmarks, similar to those in the FIS editor and the membership function editor, including the menu bar and the status line [35].

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 to constructing the rule statements automatically, by clicking on and selecting one item in each input variable box, one item in each output box, and one connection item. Choosing none as one of the variable qualities will exclude that variable from a given rule. Choosing not under any variable name will negate the associated quality. Rules may be changed, deleted, or added, by clicking on the appropriate button. Figure 3.4 shows the rule editor [35].

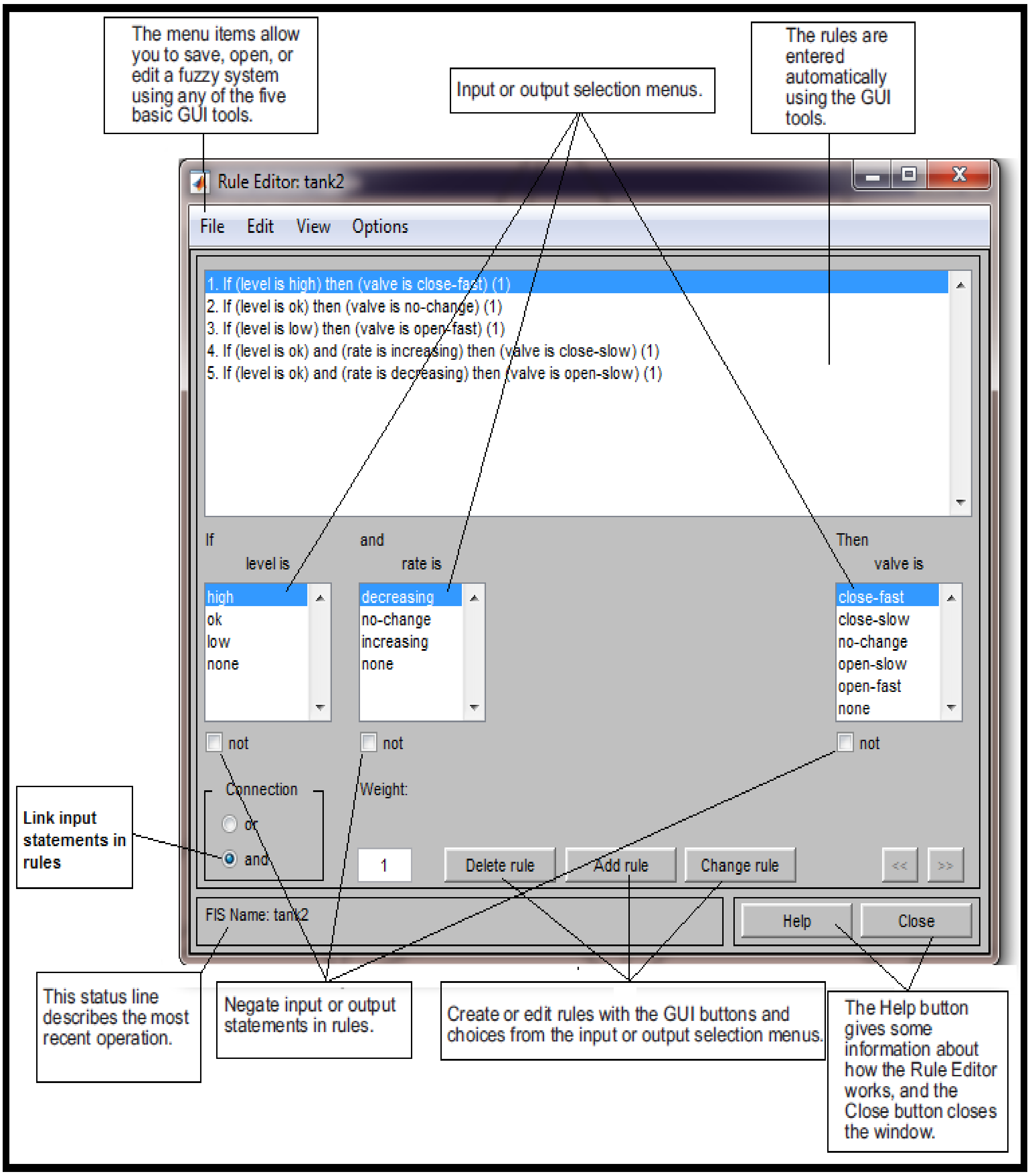

Figure 3.4: Rule editor

3.2.4 Rule viewer

Rule viewer is used for looking at, as opposed to editing the FIS. They are strictly read-only tools. The rule viewer is a MATLAB based display of the fuzzy inference diagram used as a diagnostic; it can show (for example) which rules are active, or how individual membership function shapes are influencing the results. It displays a roadmap of the whole fuzzy inference process. Figure 3.5 shows the rule viewer [36].

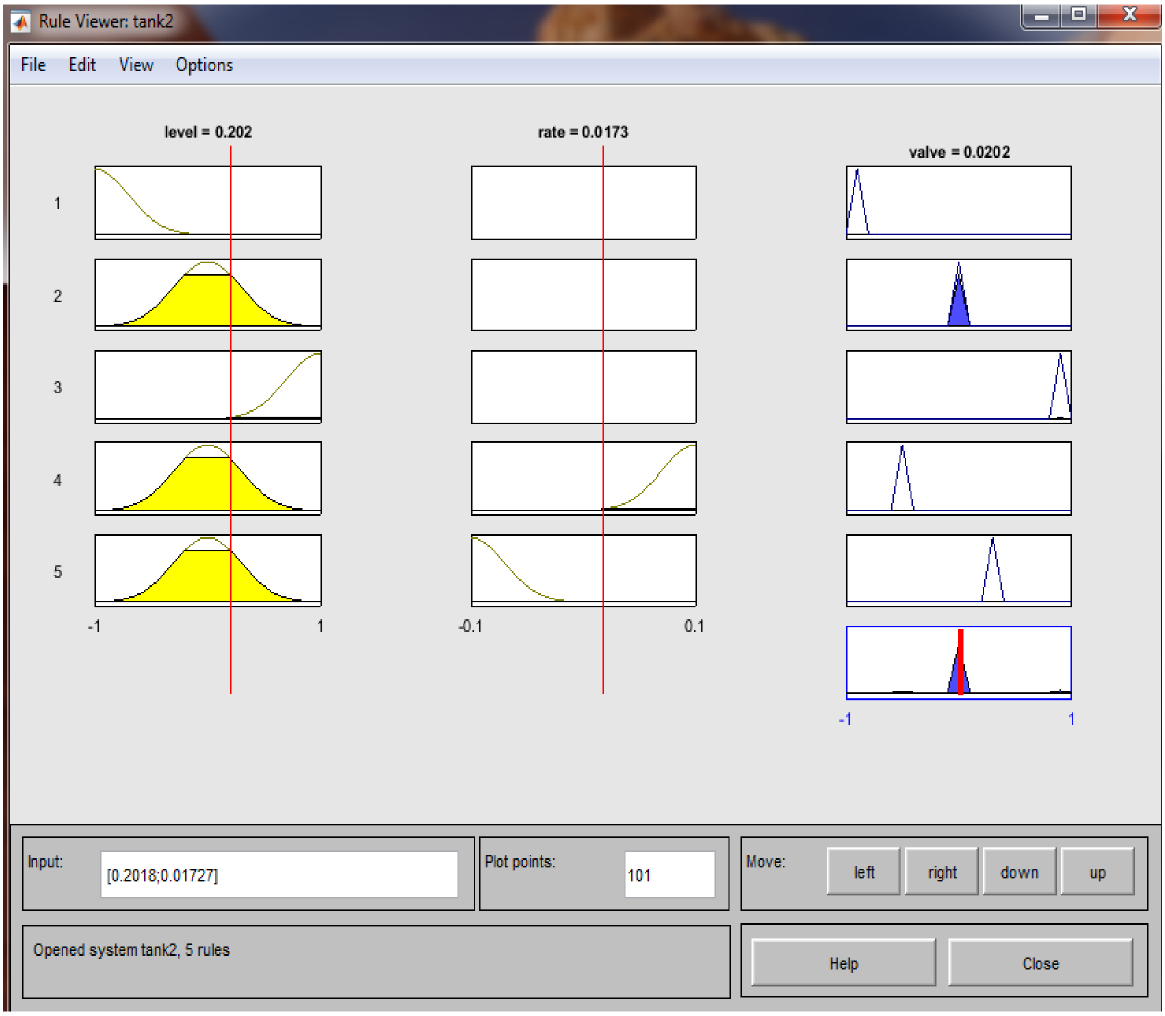

Figure 3.5: Rule viewer

The three small plots across the top of the Figure 3.5 [36] represent the antecedent and consequent of the first rule. Each rule is a row of plots, and each column is a variable. The first two columns of plots (the six yellow plots) show the membership functions referenced by the antecedent, or the if-part of each rule. The fourth plot in the third column of plots represents the aggregate weighted decision for the given inference system. The rule viewer allows you to interpret the entire fuzzy inference process at once.

The rule viewer also shows how the shape of certain membership functions influences the overall result. Since it plots every part of every rule, it can become unwieldy for particularly large systems, but, for a relatively small number of inputs and outputs, it performs well (depending on how much screen space you devote to it) with up to 30 rules and as many as 6 or 7 variables [36].

The rule viewer shows one calculation at a time and in great detail. In this sense, it presents a sort of micro view of the fuzzy inference system, to see the entire output surface of system surface viewer must open [37].

3.2.5 Surface viewer

Surface viewer is used to display the dependency of one of the outputs on any one or two of the inputs that is, it generates and plots an output surface map for the system. Figure 3.6 shows surface viewer [37].

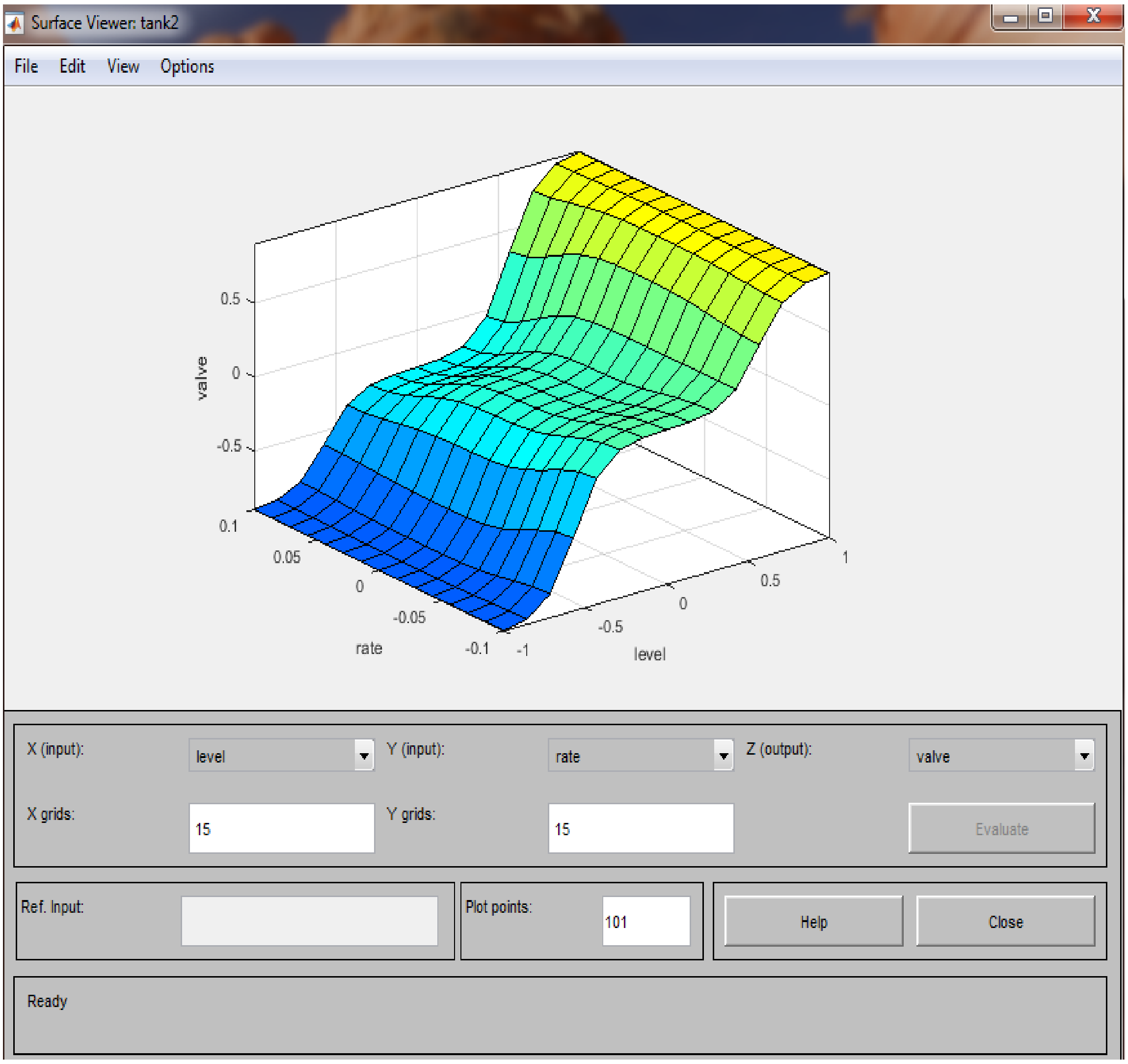

Figure 3.6: Surface viewer

This is the last of our five basic GUI tools in the fuzzy logic toolbox and it opened by selecting view surface from the view menu.

CHAPTER FOUR

SYSTEM MODELLING AND DESIGN

4.1 System Modeling

As mention before in chapter two as a result from this thesis, fuzzy logic controller don't focus on the model of the system but on the behavior of the model (dynamical and physical response of the system) because the design of fuzzy controllers is based on the operator's understanding of the behavior of the process instead of its detailed mathematical model to conclude the rules.

So in this chapter the plant was modeled in MATLAB by simulink tools (graphically) using first principle (dynamical and physical response of the system) to prove that fuzzy controller will control this system whatever the transfer function are instead of classical modeling which would be not accurate for nonlinear and complex mathematical model systems and this is the best benefit of fuzzy logic controller over conventional control method like PID controller.

4.1.1 Water tank physical and dynamical response

The tank system model can be driven graphically using simulink as mentioned in chapter two by understand the model behavior (dynamical and physical response of the system).

The water tank model can be written in mathematical form by relating the inlet Q_{in} to the outlet flow Q_{out} through drainage pipe as shown in Figure 4.1.

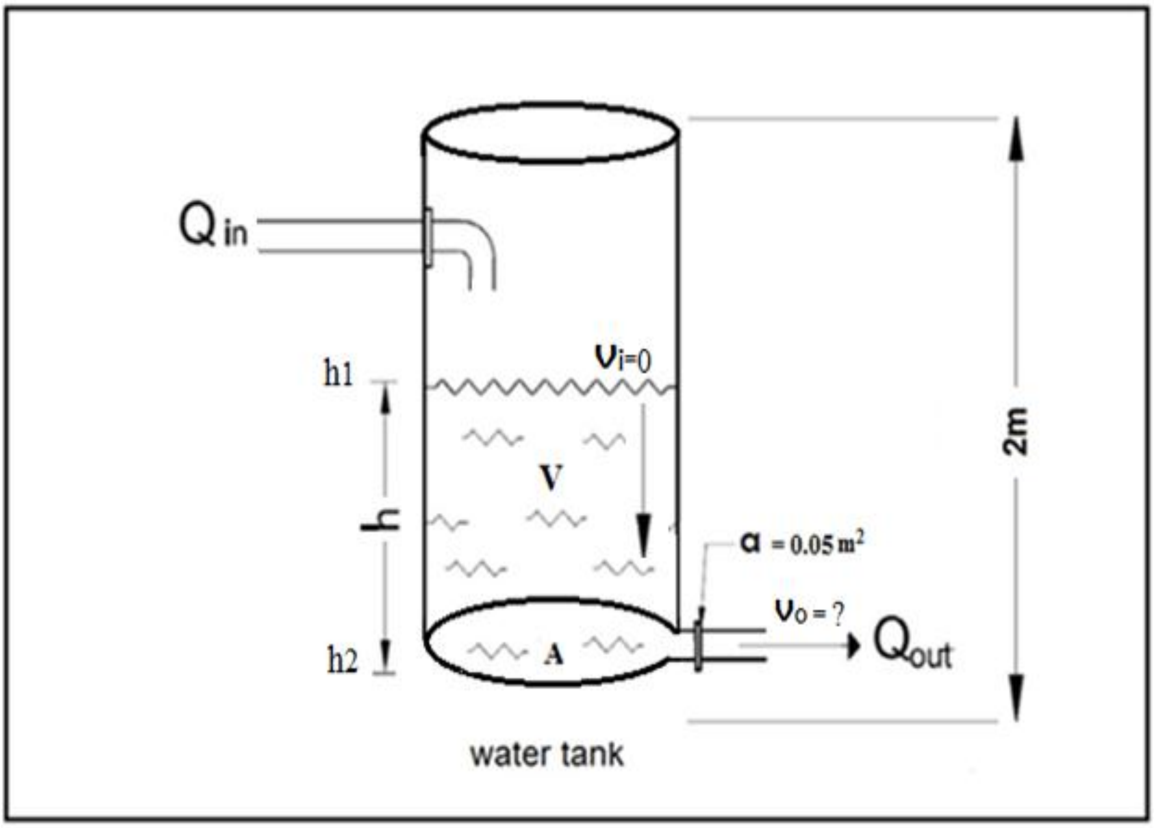

Figure 4.1: Schematic diagram for water tank

Where:

Q_{in} : inlet flow.

Q_{out} : outlet flow.

V: water volume.

A: bottom area of the tank.

α: tank outlet pipe cross-sectional area.

h: water level in the Tank.

$\nu_o(t)$: is the velocity of outlet flow water at tank outlet pipe cross-sectional area.

h1: the water level at the top of tank.

h2: the water level at the outlet.

υ_i : the velocity of water at the top level (at the beginning acceleration of the water without pump) .

Since the tank shape is cylindrical then the volume of water V(t) can be driven from cylinder volume equation:

$$V(t) = A \times h(t) \qquad (4.1)$$

Where A is the bottom area of the tank. Then from equation (1) the level of water h(t) can be written as:

$$h(t) = \frac{1}{A} \times V \qquad (4.2)$$

Beside if there is (no pump) sucking out the water, the outlet flow of water Q_{out} can be driven from equation:

$$Q_{out} = a \times \upsilon_o(t) \qquad (4.3)$$

Where $\upsilon_o(t)$ is the velocity of outlet flow water at tank outlet pipe cross-sectional "a".

To calculate $\upsilon_o(t)$ at the outlet pipe the (Bernoulli's principle equation) was used which written as:

$$P1+(p1 \times g \times h1)+(\frac{1}{2} \times p1 \times \upsilon_1^2)=P2+(p2 \times g \times h2)+(\frac{1}{2} \times p2 \times \upsilon_2^2) \qquad (4.4)$$

By applying Equation (4.4) to the schematic diagram for water tank at Figure 4.2 the equation can be written as:

$$P1+ (p \times g \times h1) + (\frac{1}{2} \times p \times \upsilon_i^2) = P2+ (p \times g \times h2) + (\frac{1}{2} \times p \times \upsilon_o^2) \qquad (4.5)$$

Where:

P1: the pressure at the inlet.

P2: the pressure at the outlet.

P: the density of water.

g: = gravity force.

By assuming there are no pump was connected yet, from equation (4.5) and Figure 4.1 the velocity of water at the top surface "vi" is equal to 0 ($v_i = 0$) and also since the pressure at the two points (top and ground level of tank) is the same "atmosphere pressure then (P1=P2), the potential energy was measured from outlet at ground level (h2=0) ,so the potential energy term goes away on the left side, and the kinetic energy term is zero on the right hand side , this reduces the equation (4.5) to:

$$(\rho \times g \times h1) = (\frac{1}{2} \times \rho \times v_o^2) \tag{4.6}$$

By cancelling ρ from each side of equation (5) then it became as:

$$v_o = \sqrt{2gh} \tag{4.7}$$

Now equation (4.3) can become as:

$$Q_{out} = a \times \sqrt{2gh} \tag{4.8}$$

4.1.2 Water tank simulink modeling

The water tank can be represented in simulink by using the following block model as shown in Figure 4.2.

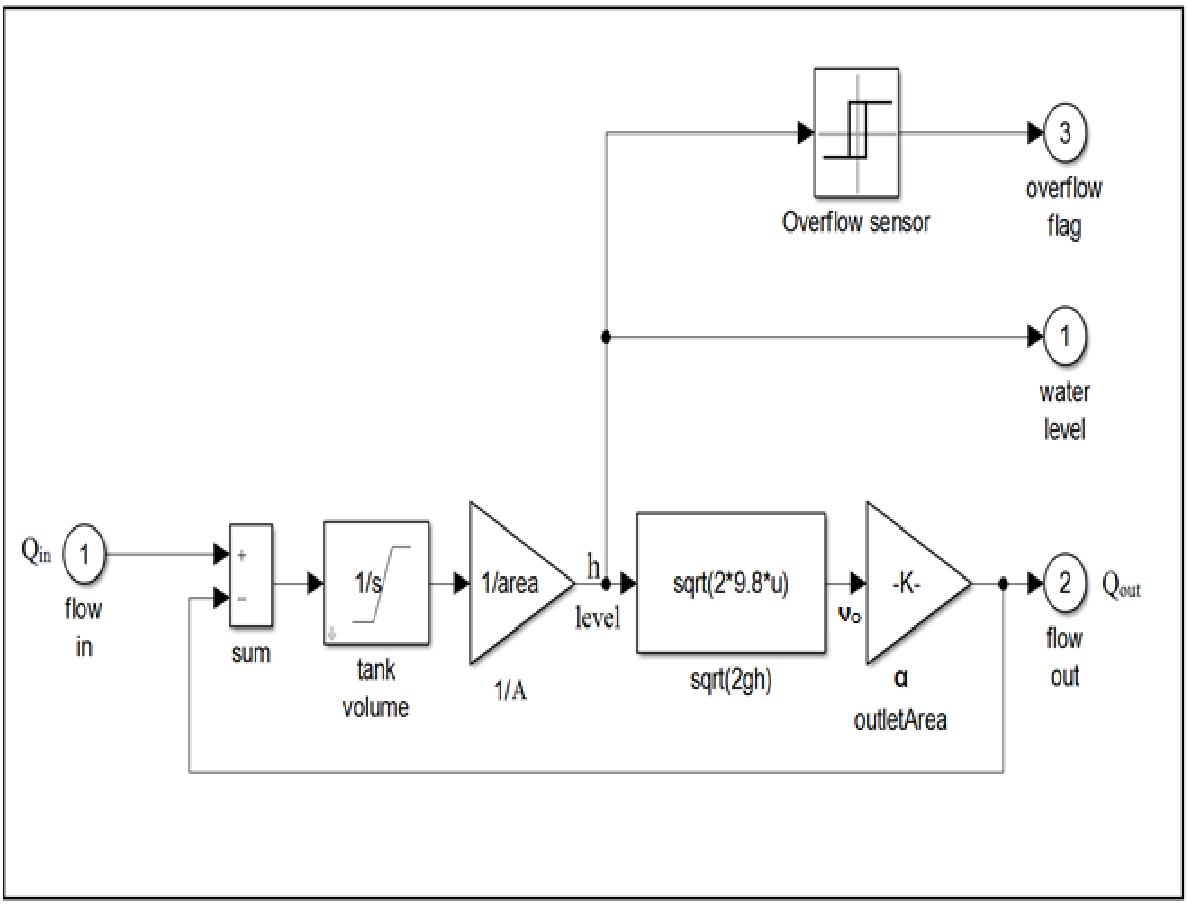

Figure 4.2 Water tank simulink block model

As seen in Figure 4.2, by considering the water tank volume as integrator with limit (saturation function).The level of water "h" can be constructed from the equation (4.2), then we cascaded it with (sqrt2gu) block function in Simulink to calculate the speed of water "υ_o" as in equation (4.7) through the outlet area pipe "α" to calculate "Q_{out}" as in equation (4.3).

Note that from the level "h" line, an indicator sensor and overflow sensor could be used for alarming and monitoring but it didn't used in this thesis it's not important for conclude the rules or designing of fuzzy controller. Finally feedback summing was made between "Q_{in}" and "Q_{out}" to relating the inlet flow of the tank "Q_{in}" to the outlet flow "Q_{out}" through drainage pipe "α".

One of the important physical characteristics of this system is that the tank empties much slower than it fill up because of the specific value of the outflow diameter pipe as compare to speed of filling it up with the pump. This solved by setting of the valve membership function and this will discussed in the steps design of fuzzy controller in this chapter. Figure 4.3 shows the parameters that used for water tank in simulink.

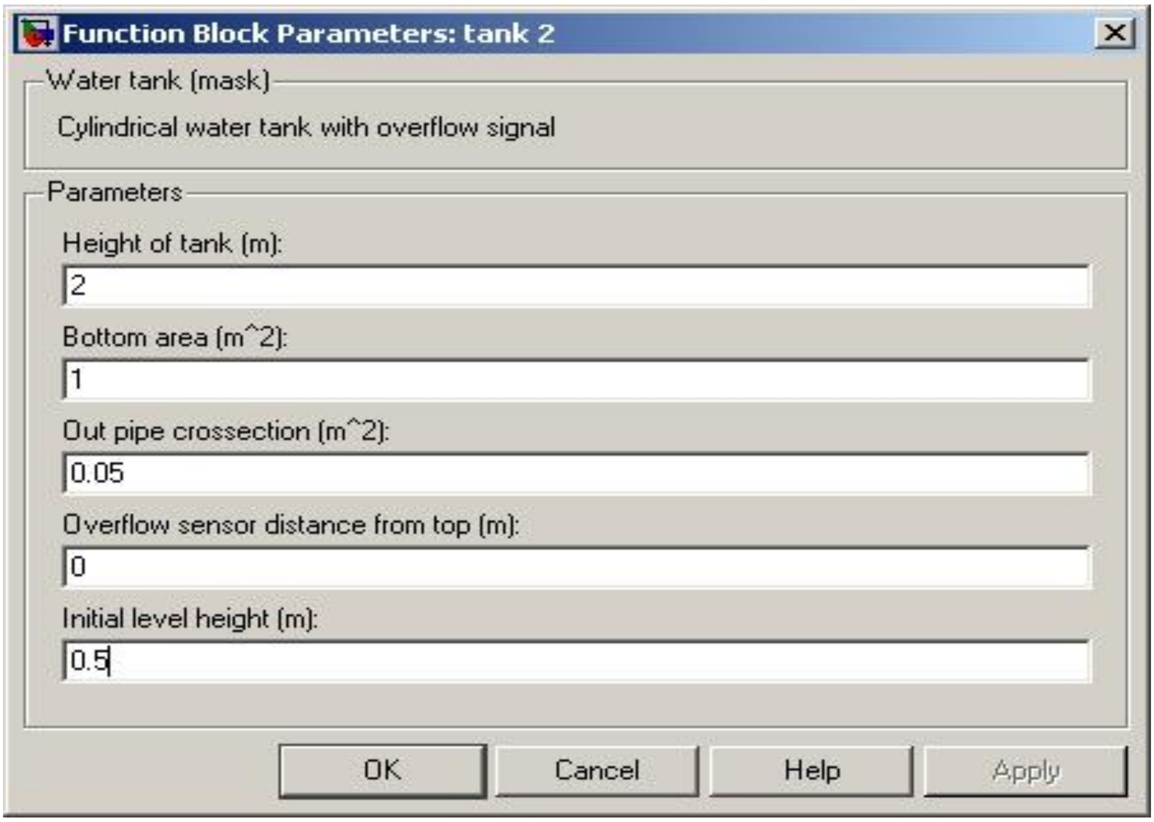

Figure 4.3: Water tank simulink parameters

4.1.3 Pump and valve simulink modeling

By considering the pump flow as a constant Q_v, the inlet flow of water Q_{in} to the tank can be driven as:

$$\mathbf{Q}_{in} = Q_{\mathbf{v}} \times \text{valve}(t) \tag{4.9}$$

Where:

Q_{in}: inlet flow

Q_v: the pump flow (constant)

valve(t): the valve position

Figure 4.4 shows pump and valve simulink block model.

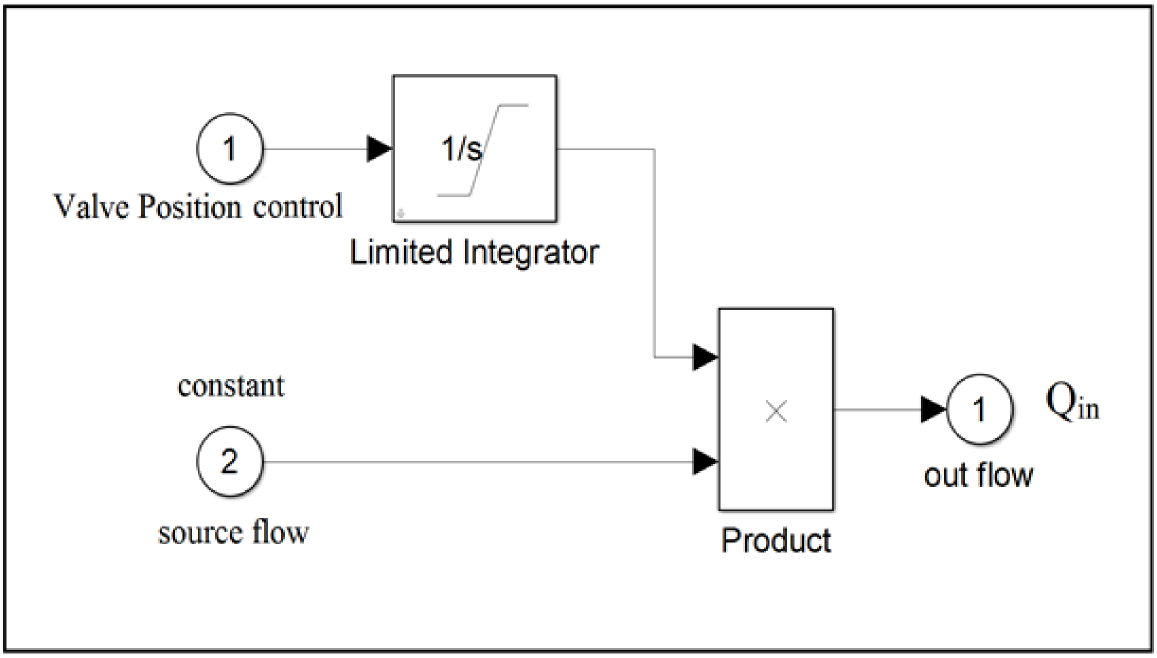

Figure 4.4: Pump and valve simulink block model

As seen from Figure 4.4, the "source flow" represent the pump flow Q_v which considered as a (constant) were multiplied with valve position "valve(t)" that controlled by fuzzy controller to find the inlet flow Q_{in} as showed in equation (4.9).

The valve control position was considered as limited integral (with hard limit) to represent the valve four position as shown in Figure 4.5.

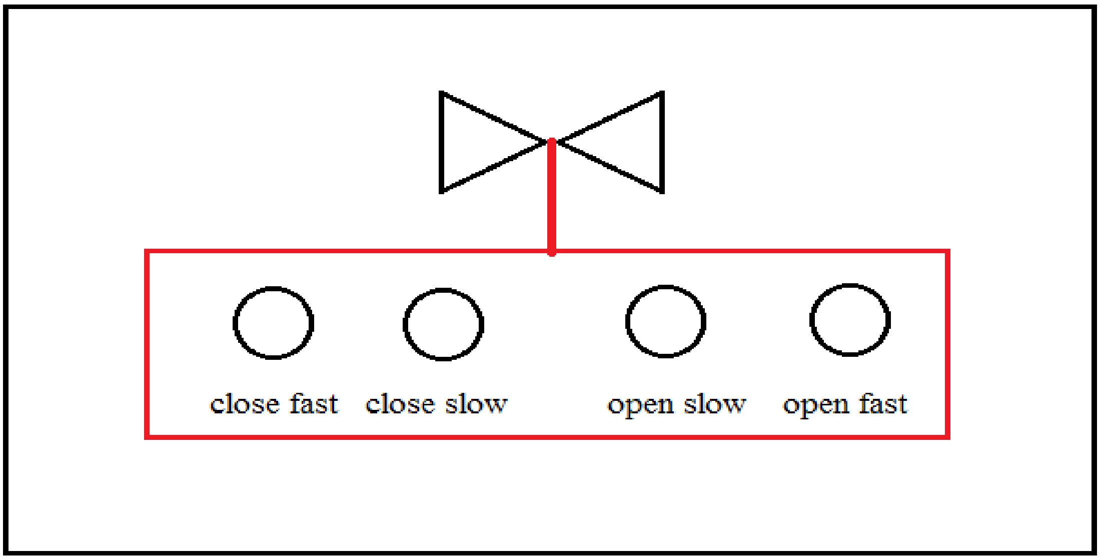

Figure 4.5 Valve control position

4.2 System Design

Figure 4.6 shows the simulink diagram of the entire water tank system that have been controlled by fuzzy logic controller or PID controller for comparing reason as discussed before.

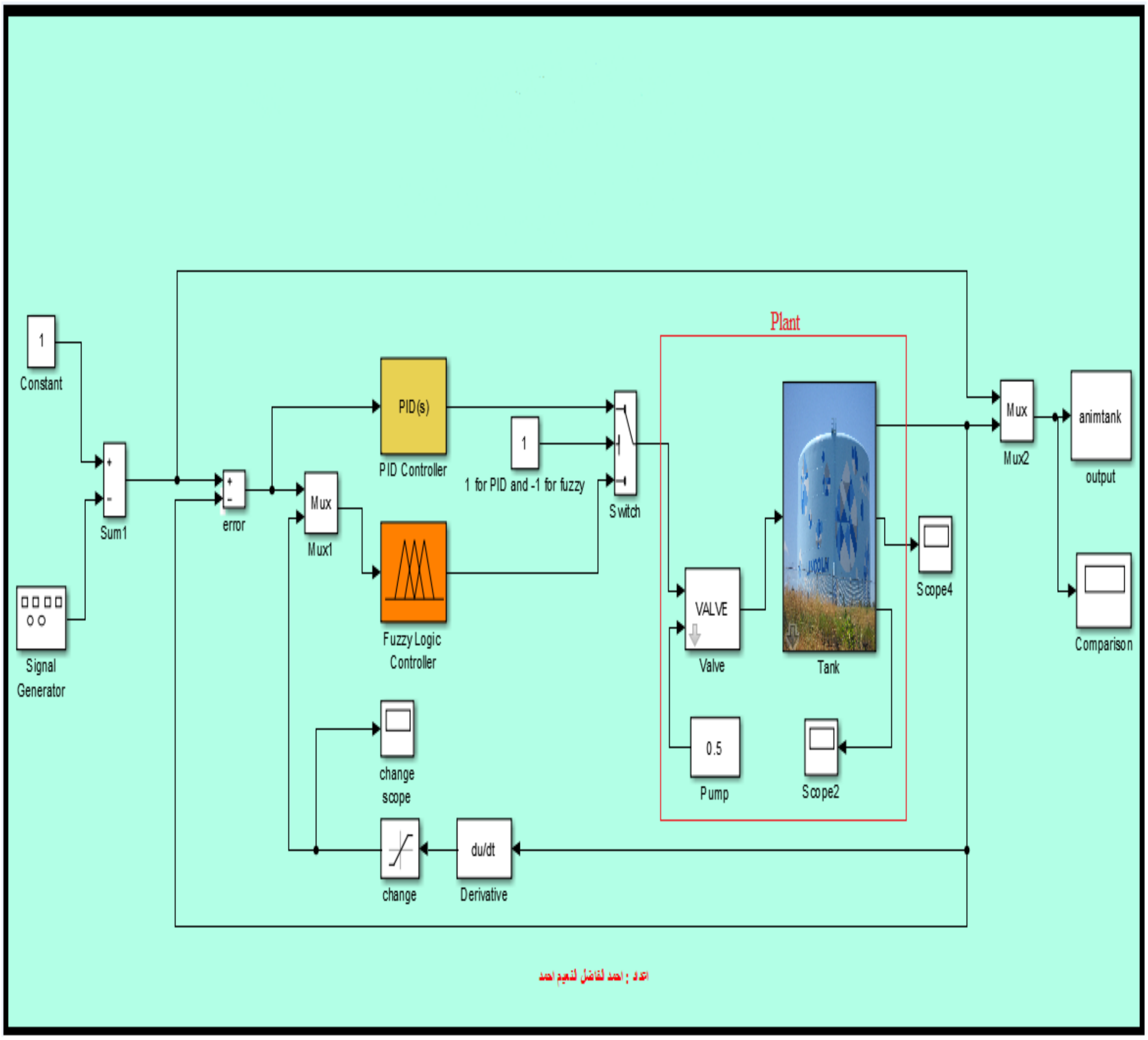

Figure 4.6: Water tank system

4.2.1 PID Controller Design

The PID gain can be designed just based on the system tracking of error and treats the system to be "black box" if the system parameters are unknown.

And this is reason for designing PID controller to control this nonlinear system without mathematical model (transfer function).

The PID controlling jobs is to adjusting the output at a level, where become there is no “error” (the difference between the sensing value (the reference desired input) and the set point (the output level of water)).

Figure 4.7 shows the Simulink block model that was used for the PID controller.

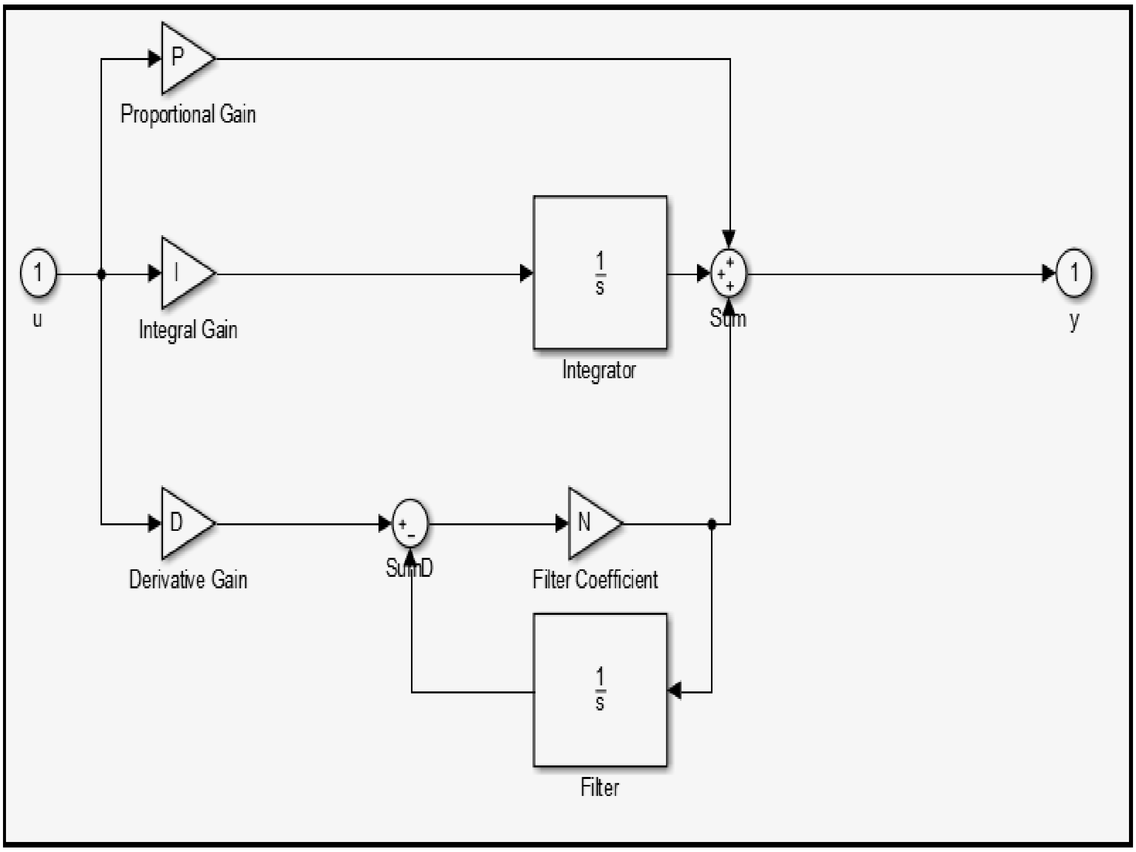

Figure 4.7: PID controller block model

Since this water tank system has nonlinear characteristics, so it would be not accurate to use fixed parameter tuning methods like (Ziegler – Nichols) method or (Cohen – Coon) method for tuning The PID parameters in this thesis because it tunes the loop for (quarter-amplitude-damping response) which also called quarter amplitude decay or QAD, which overshoots and oscillates quite a bit in nonlinear system as shown in Figure 4.8.

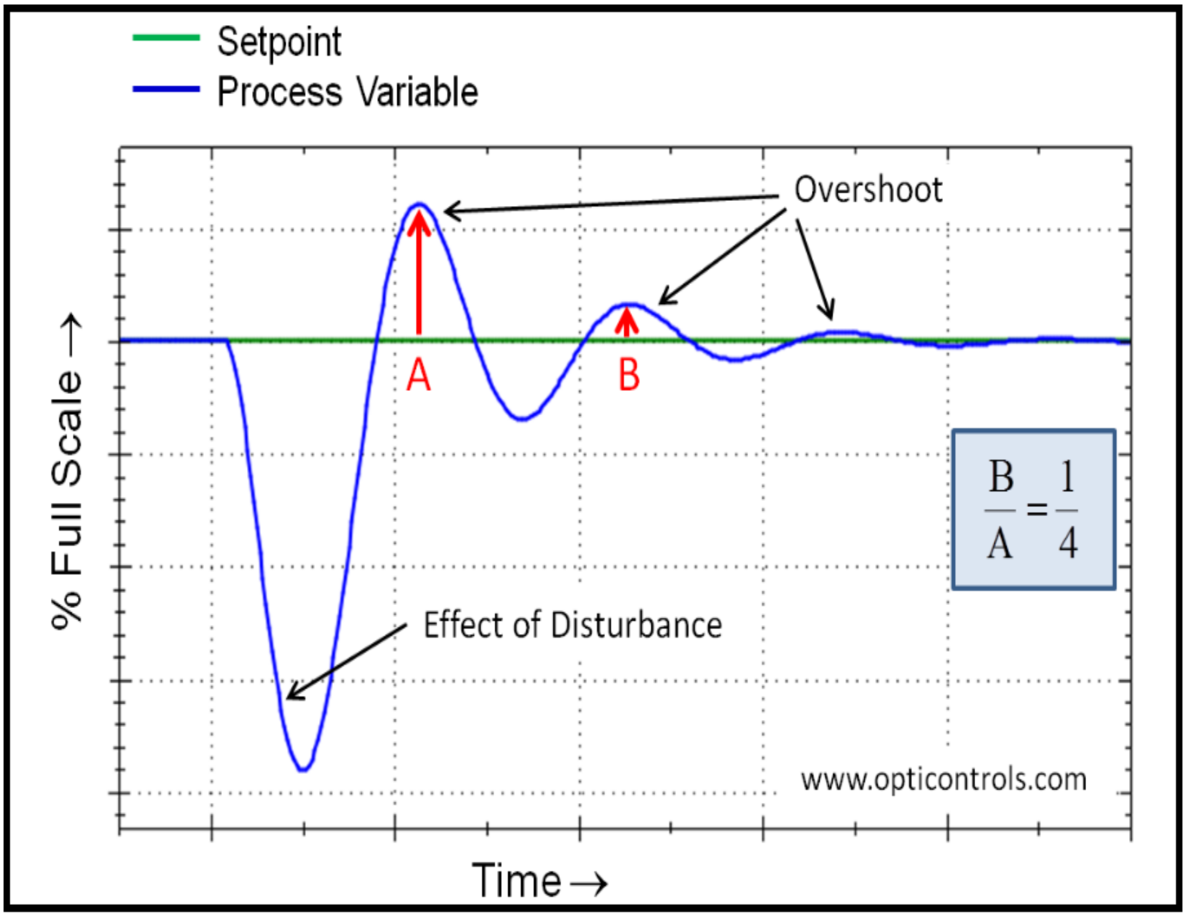

Figure 4.8: Quarter amplitude decay effect for nonlinear system

From Figure 4.8 It seen that it eliminate the disturbance but it causes the process to overshoot its set point and to oscillate around it a few times before eventually settling down.

So this is the reason why in this thesis for starting point the PID controller was tuned with (trial and error approach) without ensuring good performance.

The PID controller was tuned with these parameters as shown in Figure 4.9.

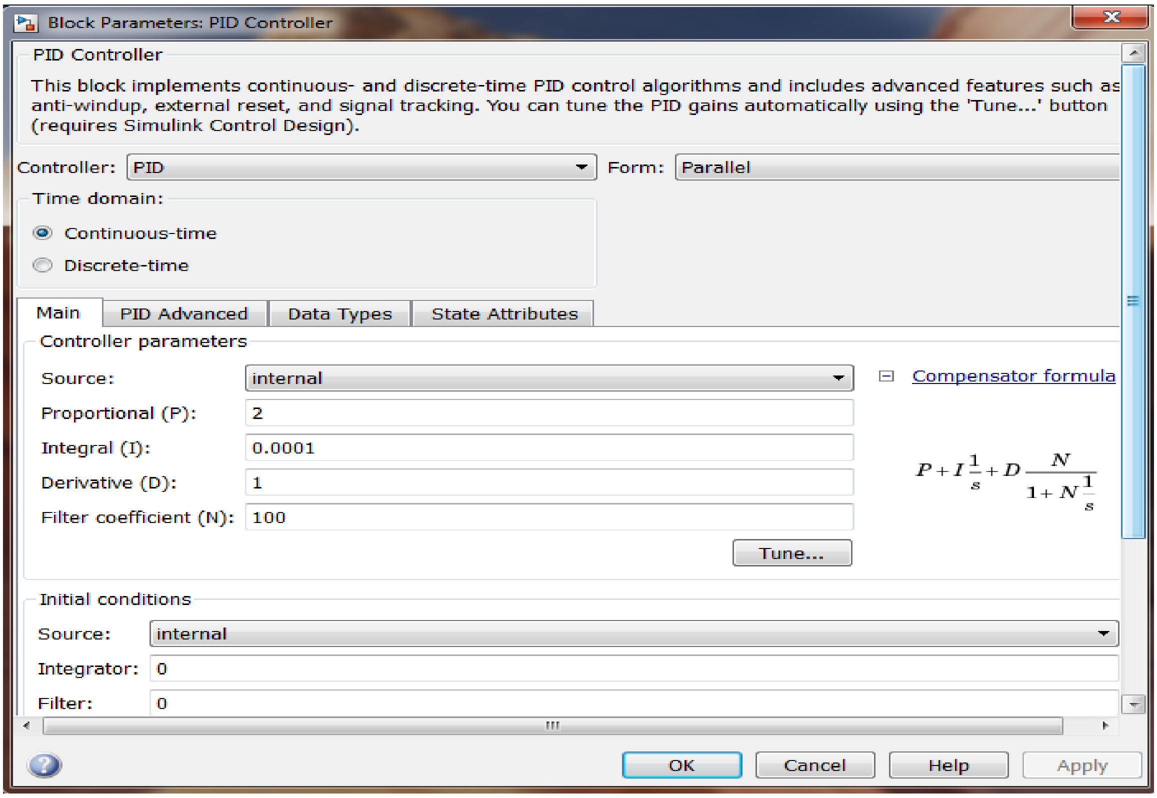

Figure 4.9: PID controller block model

4.2.2 Operation of fuzzy logic controller

Before dealing with the steps of designing the fuzzy logic controller by fuzzy logic toolbox, let's discuss the operation of a fuzzy logic controller. Figure 4.10 shows the operation of a fuzzy logic controller for water tank system.

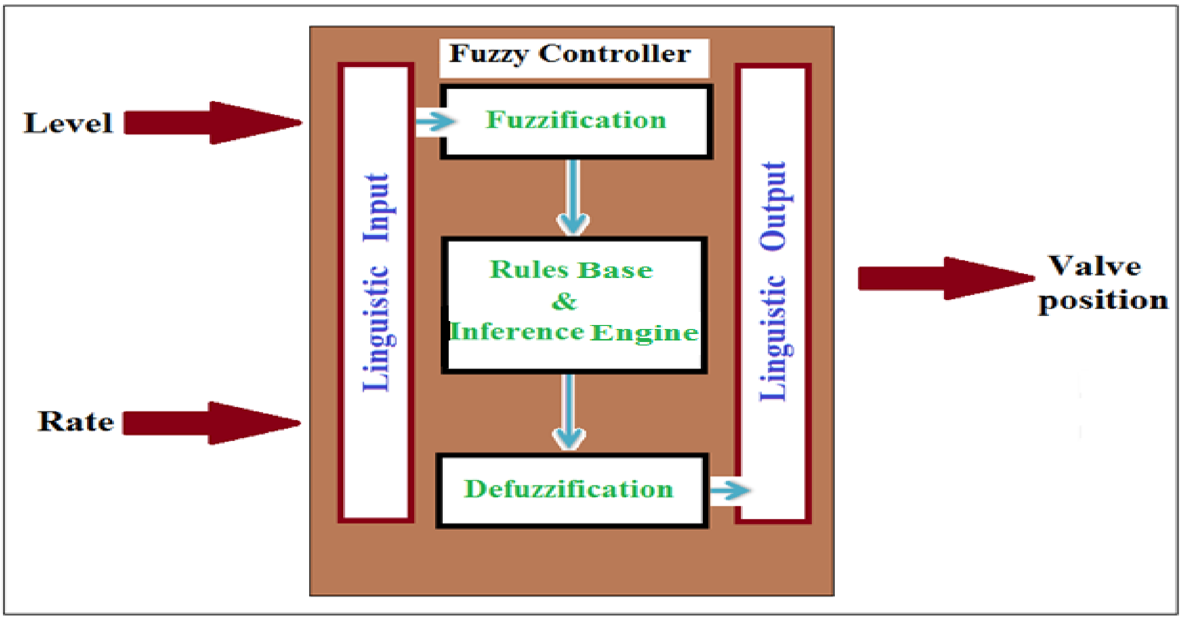

Figure 4.10: Operation of a fuzzy logic controller

i. Linguistic inputs

The linguistic inputs that determine the valve position along with their adjectives are shown in Table 4.1.

Table 4.1: Linguistic inputs

Linguistic inputs	First adjective	Second adjective	Third adjective
level	high	normal	low
rate	decreasing	no-change	increasing

As shown in table 4.1, there are three different parameters or variables (known as linguistic inputs) that are taken into consideration to determine the final output (which is valve position). The first Linguistic Inputs is the “level” of the water (which is either “high”,” normal” or “low”). Similarly, the second linguistic Inputs is the “rate” of the water (which is either “decreasing”, “no-change” or “increasing”). The fuzzy controller takes these three inputs, processes the information and outputs the valve position.

ii. Linguistic outputs

The linguistic output contains the adjectives for the final output which is the position of the valve that determine by the fuzzy controller to control the out flow water.

The linguistic outputs along with its adjectives are shown in Table 4.2.

Table 4.2: Linguistic output

Linguistic output	First adjective	Second adjective	Third adjective	Fourth adjective	Fifth adjective
valve position	close-fast	close-slow	no-change	open-slow	open-fast

As shown in Table 4.2, there is one output parameter or variable (known as linguistic output) that represents the final output “valve position”. This linguistic output variable is the “valve position” (which is either “close-fast”, “close-slow”, “no-change”, “open-slow” or “open-fast”).

iii. Fuzzification method

Linguistic inputs variables are fuzzified according to the membership function graphs as shown in Figure 4.11.

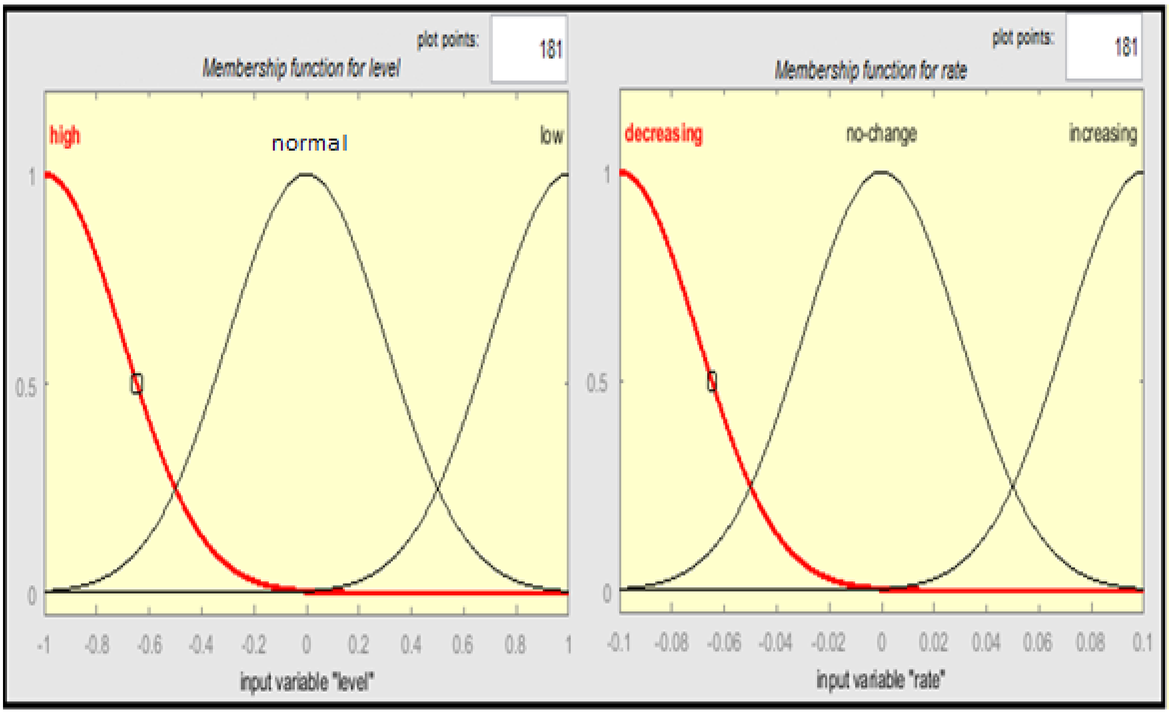

Figure 4.11: Membership function graphs for linguistic inputs

The data which used to obtain the membership function graphs are purely from sensible assumptions.

X axis of each of the graphs in Figure 4.11 corresponds to the respective linguistic inputs values that are obtained from the sensors.

The working of the sensors and how those crisp input values (from -1 to 1 or -0.1 to 0.1) as shown in x axis are obtained is beyond the scope of this thesis. It is assumed that the sensors provide the crisp values in these ranges.

Y axis of each graph in Figure 4.11 corresponds to the grade of membership ranging from 0 to 1. Similarly, linguistic output variable (valve position) is fuzzified according to the membership function graph as shown in Figure 4.12.

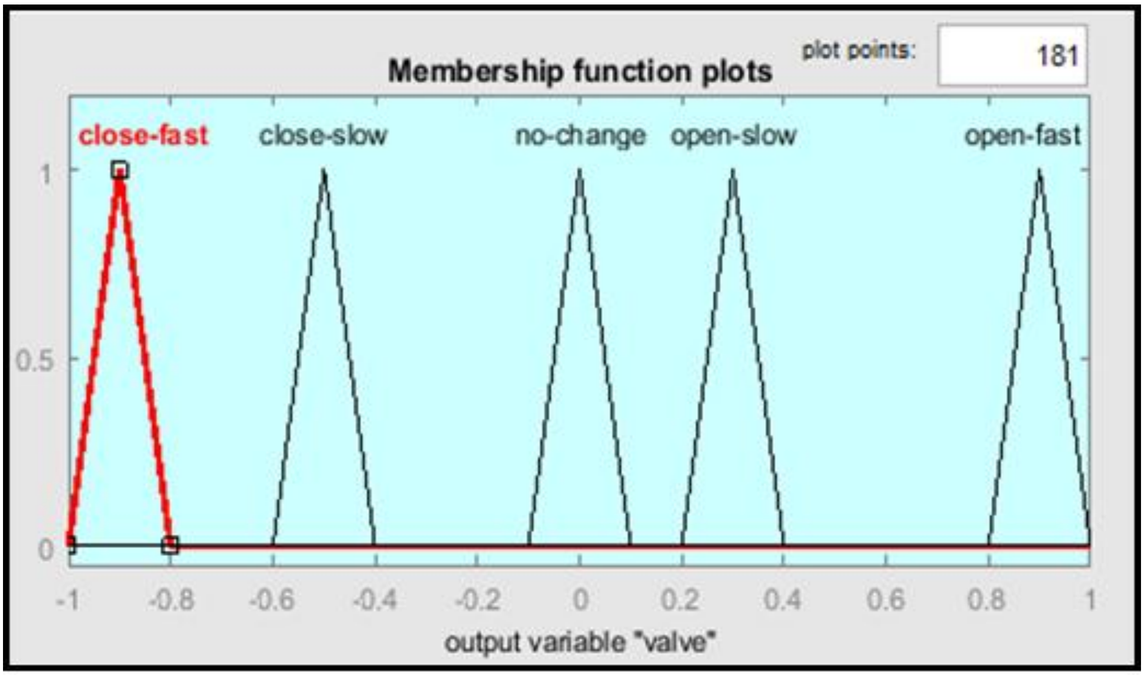

Figure 4.12: Membership function graph for linguistic output

The data used to obtain the membership function graphs is also purely from a sensible assumption.

x axis of each graph in Figure 4.12 corresponds to crisp value for linguistic output “valve position”.

Y axis of each graph in Figure 4.12 corresponds to the grade of membership ranging from -1 to 1.

This whole process of converting the crisp values to fuzzified values (membership grade values) is the fuzzification process.

iv. Rule base and inference

The decision which the fuzzy controller made is derived from the rules known as “fuzzy rules”.

As mentioned in Chapter 2, the fuzzy rules are the sets of “IF” and “THEN” statements. Fuzzy rules are intuitive and easy to understand since they are common English statements.

The set of rules that used to control water tank valve position are shown in Table 4.3.

Table 4.3: Rules for valve position

RULE NUMBER	LINGUISTIC INPUTS		LINGUISTIC OUTPUT
	Level	Rate	Valve Position
1.	high		close-fast
2.	normal		no-change
3.	low		open-fast
4.	normal	increasing	close-slow
5.	normal	decreasing	open-slow

The rules outlined in Table 4.3 can be read in terms of “if” and “then” statements as shown below:

•If (level is high) then (valve is close-fast)

•If (level is normal) then (valve is no-change)

•If (level is low) then (valve is open-fast)

•If (level is normal) and (rate is increasing) then (valve is close-slow)

•If (level is normal) and (rate is increasing) then (valve is close-slow)

All the above fuzzy inference operation were combined together with fuzzy inference techniques such as (AND-OR-Product-MIN-MAX-Probor) which discussed in Chapter 2 as shown on Figure 4.13.

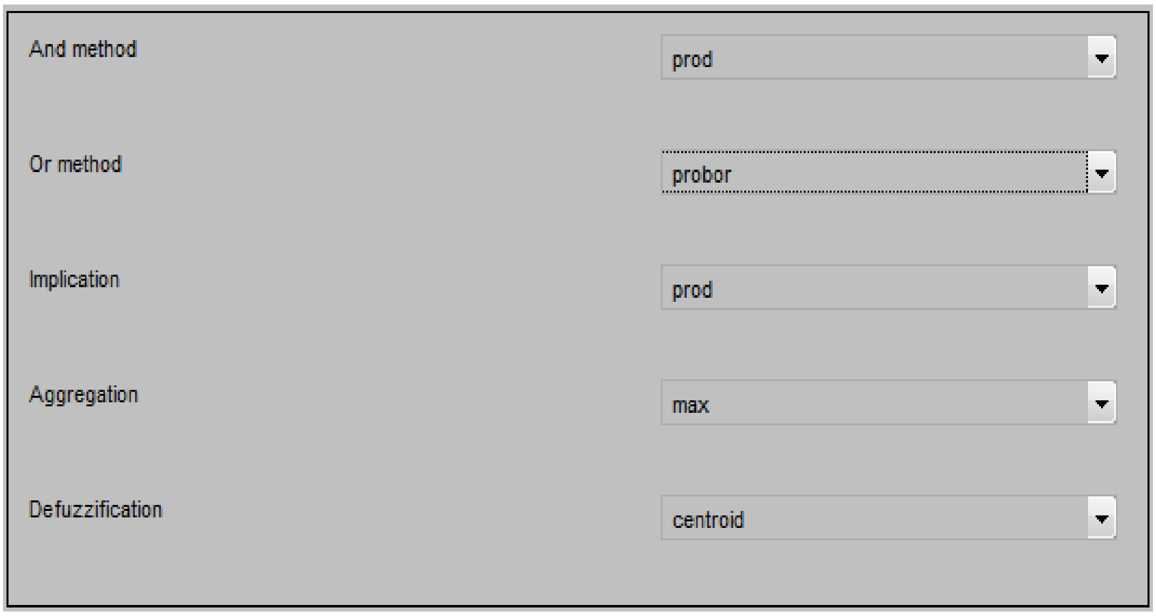

Figure 4.13: Fuzzy inference techniques

v. Defuzzification

Once the result from the fuzzy inference engine is obtained it is then necessary to produce a quantifiable result which in this case is the valve position.
The defuzzification process is used to interpret the membership degrees of the fuzzy sets into a specific real value.
The CoG method "centroid" was selected as the defuzzification technique for controlling the water tank system as shown in Figure 4.13 above.

4.2.3 Fuzzy controller design

The Steps process for designing fuzzy logic controller by using the fuzzy logic toolbox as following:

- First the FIS editor was opened by typing "fuzzy" on command window as shown in Figure 4.14.

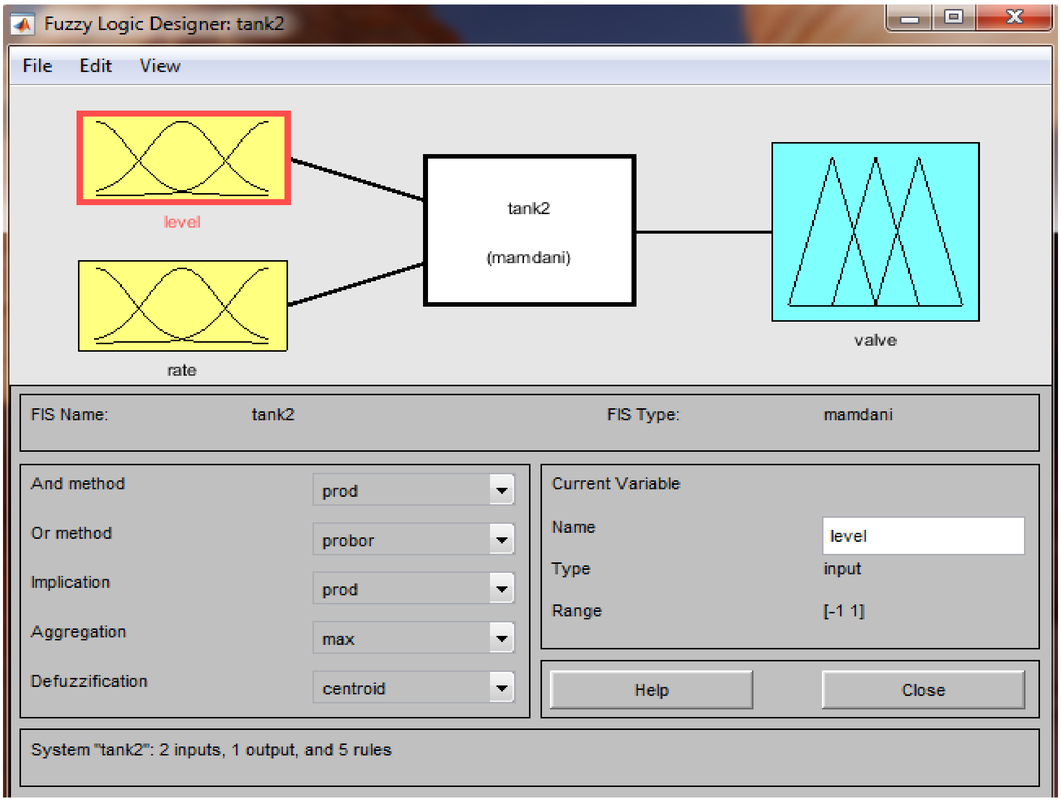

Figure: 4.14 FIS editor

- Then, the two input variables were selected, by double-clicking on the variable mark in edit tab on FIS window.
- The first variable was named to “level” and the second as “rate” and the output variable was named to “valve”, fuzzy inference techniques (And method, Or method, Implication, aggregation and Defuzzification) was adjusted as it shown on Figure 4.14.
- Now by double-click on the input variable mark, the membership function editor will open as shown on Figure 4.15.

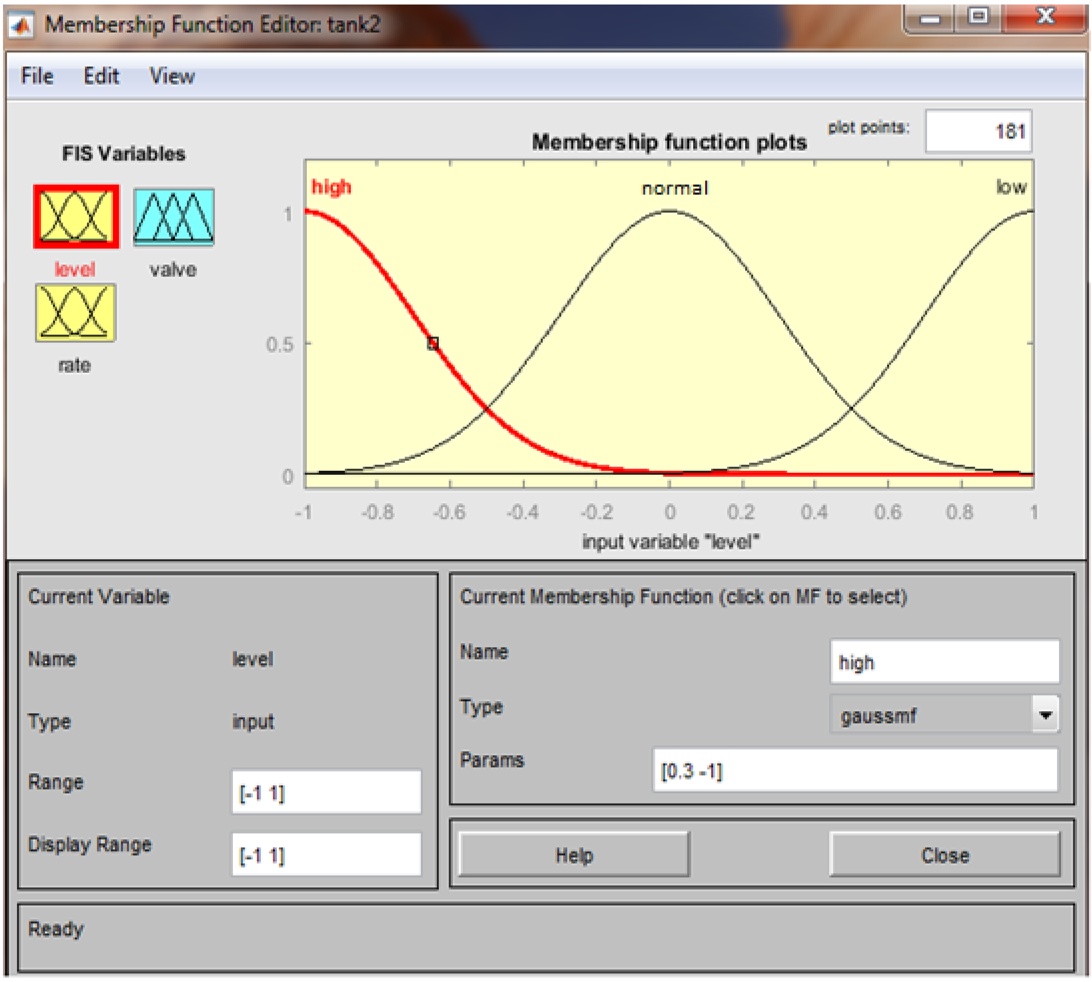

Figure 4.15: Membership function editor for input variables

- The "type" tab was used to choose "gaussmf" for MF Type. This added three "Gaussian" curves to the input variable service.
- The "range" and the "display range" tabs were adjusted to the vector [-1 1].
- By clicking once on the curve with the leftmost hump. The first curve was named to "high". The default parameter listed for this curve is [0 3 -1].
- The curve with the middle hump was named "normal", and the curve with the rightmost hump was named "low".
- The second input variable was named "rate".
- The range and the display range was adjusted to the vector [-0.1 0.1].
- Three "gaussmf" curves were added to the input variable "rate".
- By clicking once directly on the curve with the leftmost trapezoid. The curve was named "decreasing". To adjust the shape of the membership

function, the mouse was used. The default parameter that listed for this curve is [0.03 -1].

- The curve with the rightmost trapezoid was named “no-change”, the default parameter listed for this curve is [0.03 0].
- The next curve was named “increasing”, the default parameter listed for this curve is [0.03 0.1].
- Next, the output variable membership functions “valve” were created as shown in Figure 4.16.

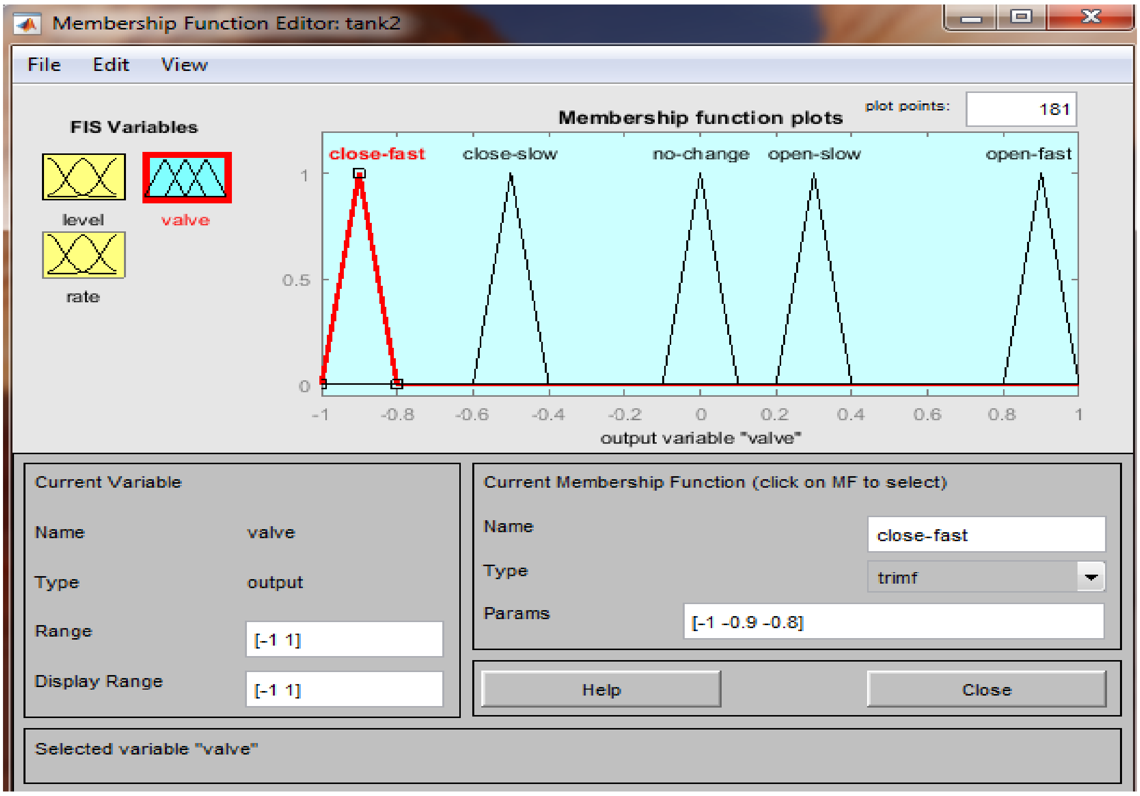

Figure 4.16: Membership functions for the output variable

- Five curves were added to the membership functions for the output variable and they were renamed respectively as (close-fast, close-slow, no-change, open-slow, open-fast).
- To adjust the shape of their membership functions as shown in Figure 4.16, theses desired parameters were selected as shown on table 4.4.

Table 4.4: Crisp range for valve position

Fuzzy Variable	MF Used	Crisp output Range
close-fast	Triangular MF	[-1 -0.9 -0.8]
close-slow	Triangular MF	[-0.6 -0.5 -0.4]
no-change	Triangular MF	[-0.1 0 0.1]
open-slow	Triangular MF	[0.2 0.3 0.4]
open-fast	Triangular MF	[0.8 0.9 1]

The process steps for specifying the rules which selected as following:

- "high" under the variable level
- "none" under the variable "rate"
- "close-fast" under the output variable "valve".

The resulting rule is:

If (level is high) then (valve is close-fast) (1)

The numbers in the parentheses represent weights that can be applied to each rule; it used to specify the strength apriority of the rules. If there is no number was selected, the weights are assumed automatically to be unity (1).

The all rules that used for driving the fuzzy controller were selected to be as:

a. If (level is high) then (valve is close-fast) (1)
b. If (level is normal) then (valve is no-change) (1)
c. If (level is low) then (valve is open-fast) (1)
d. if (level is normal) and (rate is increasing) then (valve is close-slow) (1)
e. If (level is normal) and (rate is decreasing) then (valve is open-slow) (1)

The system was tested with "three" rules first then it was extend to "five" rules to compare the performance of the system at two status and to see the difference between fuzzy controller method and PID controller method and which is the best for controlling the water tank systems.

4.3 System Simulation

Before discussing the simulation result let's discuss the other component of the system which shown in Figure 4.17.

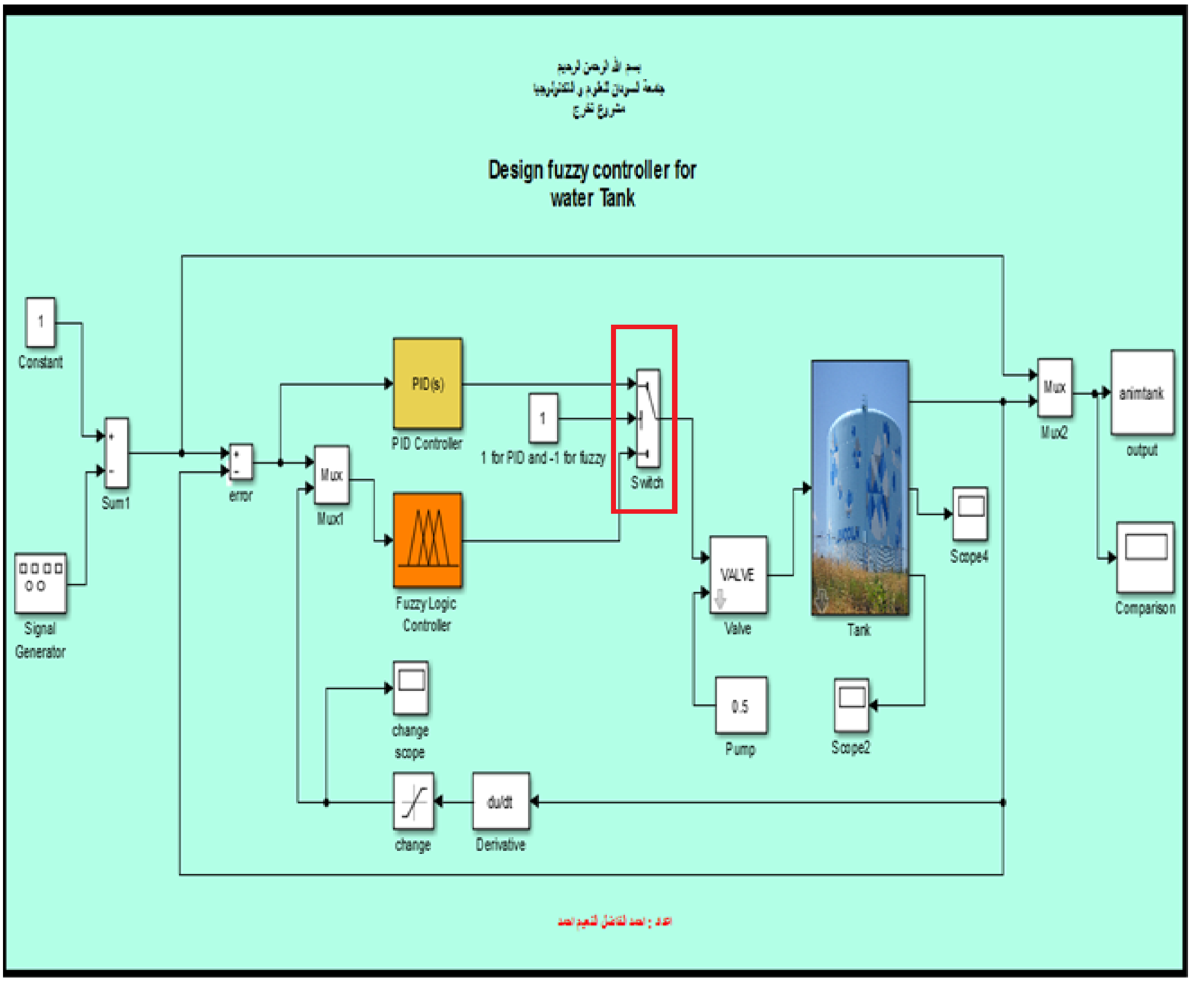

Figure 4.17: Water tank system

The switch as shown in Figure 4.17 is used to combine the two controller circuitry to one circuitry for simplification. It determines which controllers were used to drive the water tank system, "1" for PID controller and "-1" for.

Notice that there is one input goes to the PID controller which is the "error" but the fuzzy controller has two inputs (error "level"& the divertive of the output "rate").

For both systems, a step sequences “desired input” was generated as a reference to study the system performance which created via signal generator in simulink as shown in Figure 4.18.

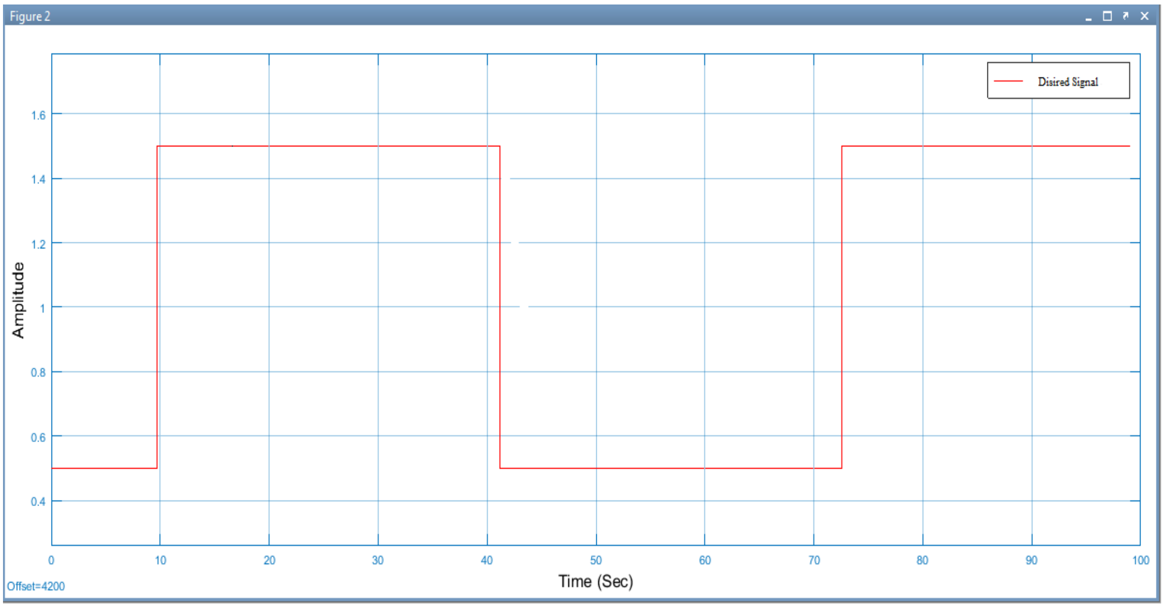

Figure 4.18: Desired input

The parameters that used to create this sequence with signal generator as shown in Figure 4.19.

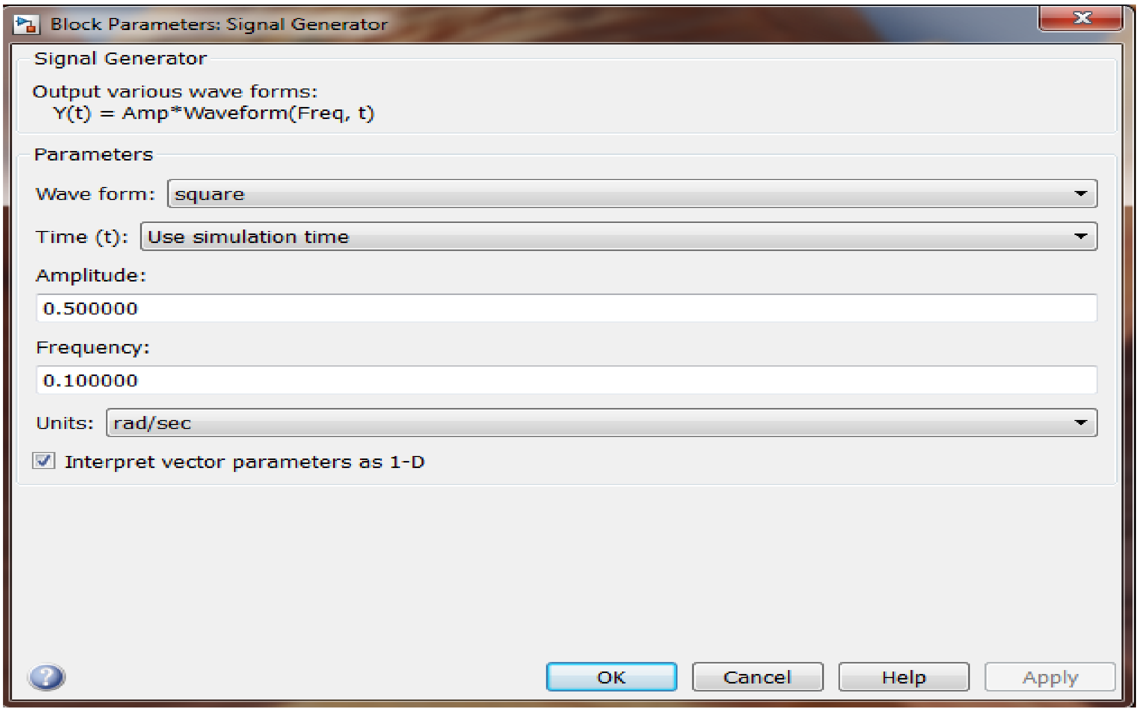

Figure 4.19: Parameters for desired input

4.3.1 PID controller result

The PID controller result as shown in Figure 4.20.

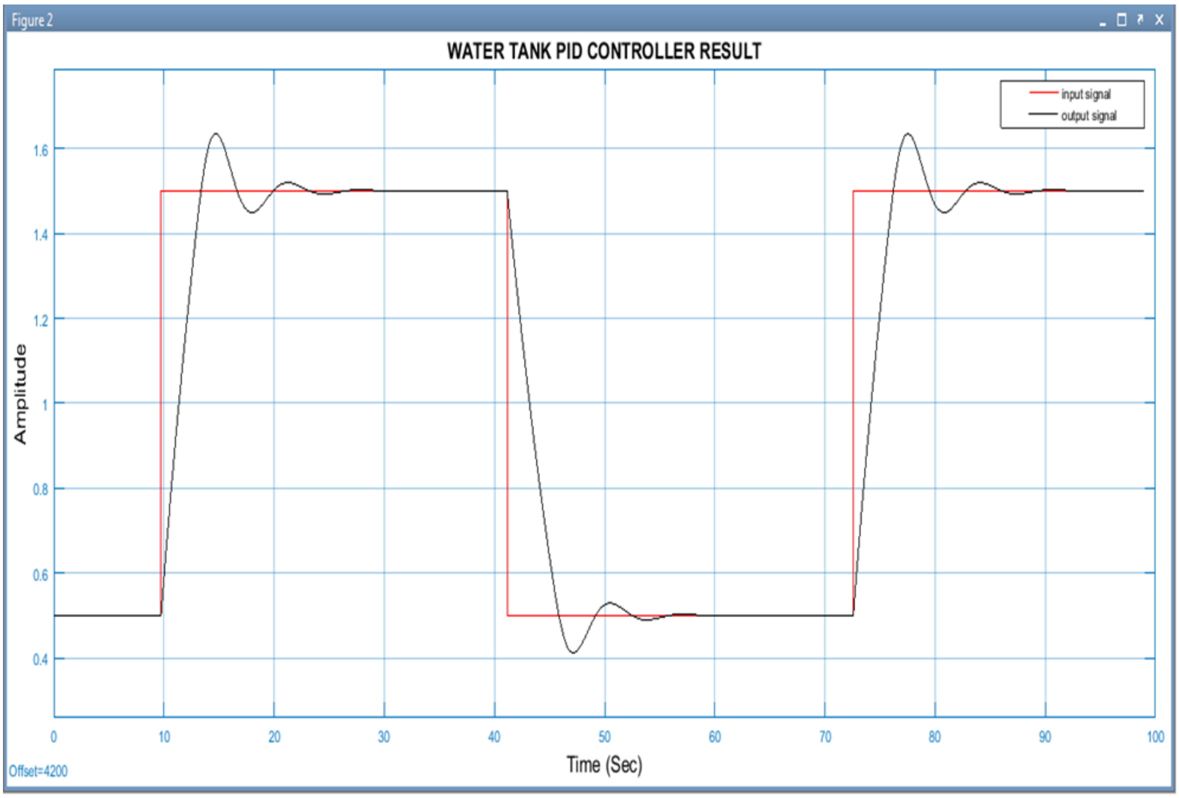

Figure 4.20: PID controller result

From Figure 4.20, it is seen that PID controller derived the system unstable due to mismatch error generated by the inaccurate time delay parameter used in the plant model.

Transient and overshoot were presented when PID controller was used to control the water tank. Also it seen that there was no steady state error because of the nature of the system because there was integral behavior on the water tank characteristics that reduced the steady state error.

So to reduce the transient and the overshoot, the PID parameters must be tuned with (auto tuning") method which would be the best way for tuning this nonlinear system without even needing the transfer function, but unfortunately, this method requires using the benefit of (microprocessors) for tuning the parameters of PID. It can solve the problem and make the system stable by finding the ideal parameters as shown in Figure 4.21.

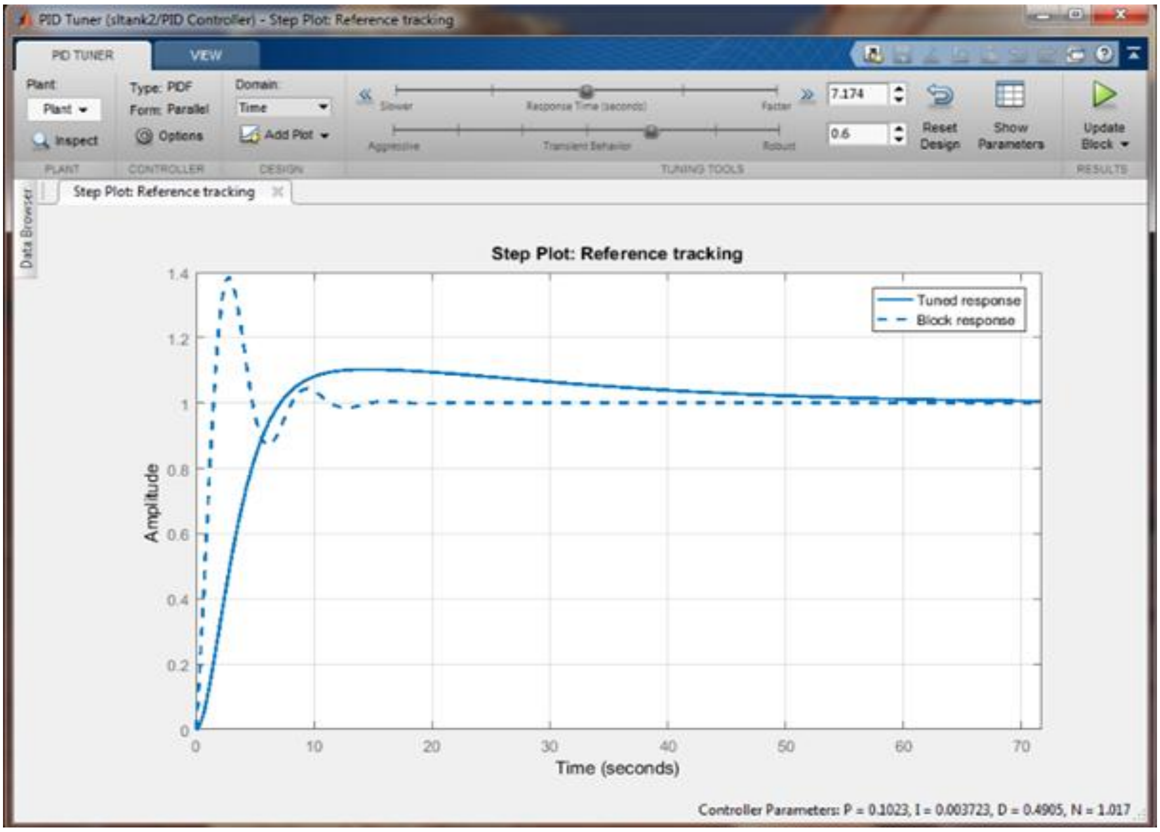

Figure 4.21: PID auto tuning method

But, using microprocessors will raise the cost of controller, because it requires large amounts of memory (RAM or ROM) to incorporate lookup tables and it does require a more (processing power) to handle the computation needs of the mapping function as been discussed in chapter two.

4.3.2 Fuzzy controller result

First thing to show the benefit of using fuzzy controller over the PID controller and how it is easy to tune the system without need to modify the parameters like PID controller, the system was tested with three rules then with 5 rules. The result of fuzzy controller is shown in Figure 4.22 and Figure 4.23.

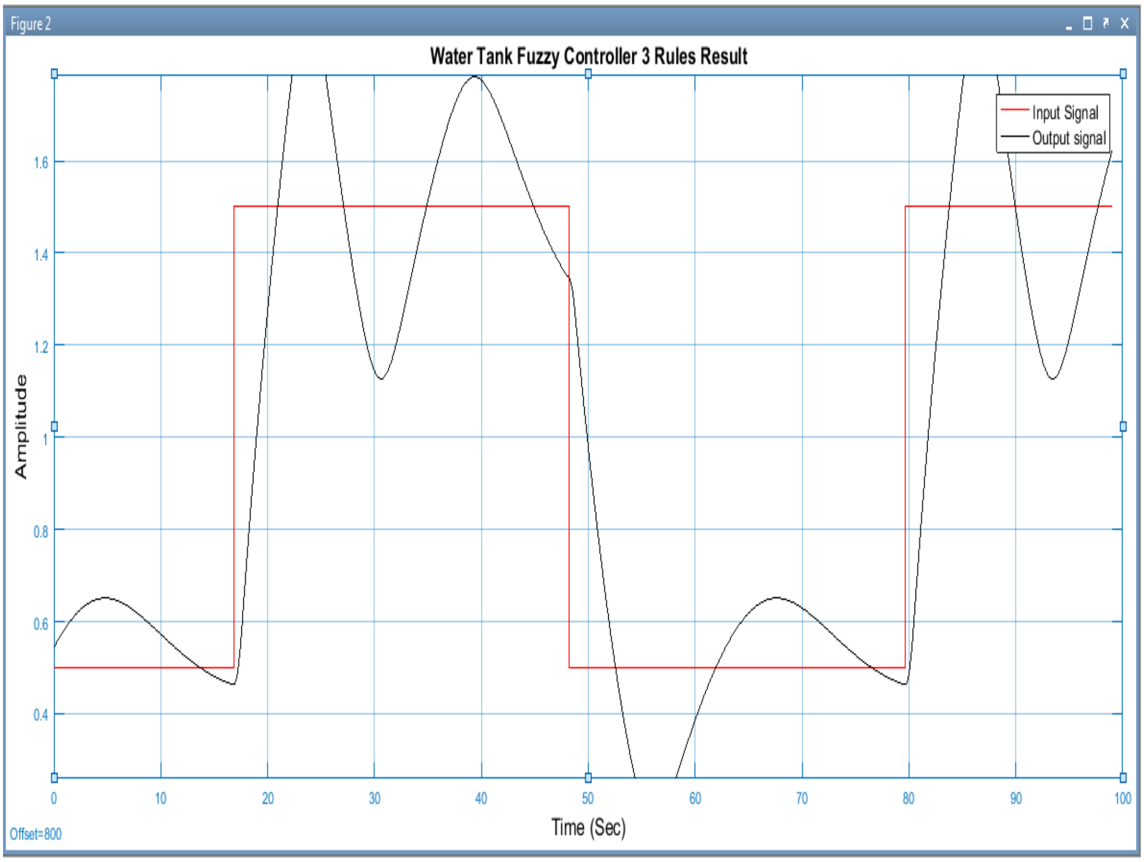

Figure 4.22: Fuzzy controller with 3 rules result

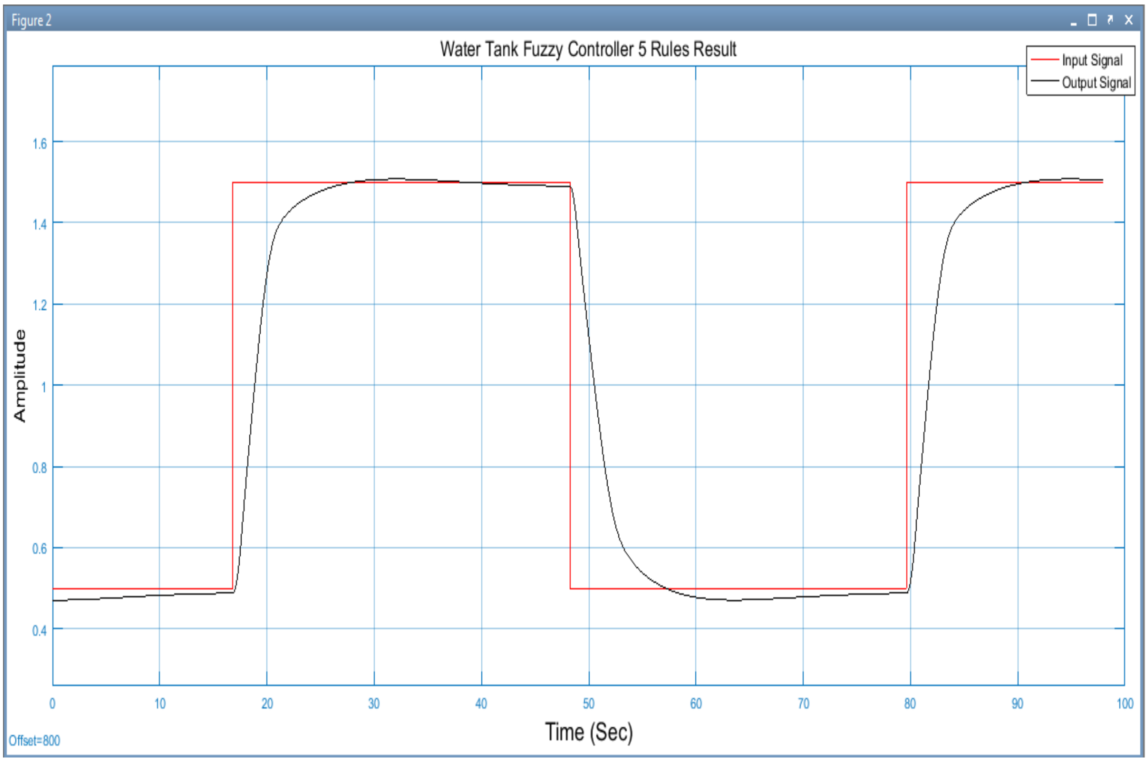

Figure 4.23: Fuzzy controller with 5 rules result

It seen that the system was became more stable just in a simple step by adding two more rules which observed that the fuzzy controller was reduced overshoot and was taking less time to reach the steady state and better rise time. Table 4.5 shows comparison between PID and fuzzy logic controller result.

Table 4.5: PID and fuzzy logic controller result

Parameters	PID Controller	Fuzzy Logic Controller
overshoot	present	not present
settling time	more	less
rise time	less	more
steady state error	not present	not present

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

First things and important things that proved through this study, the fuzzy controller design is not based on the mathematical model of the process it's based on the operator's understanding of the behavior of the process (physical and dynamical response) instead of its detailed mathematical model (transfer function) to conclude the rules that match desired behavior of the system.

PID controllers can be designed to sustain the level of liquid flow, but the limitation is its feedback type, the controller will take control action only after the output is affected by error. Also, it doesn't recognize the unanticipated alteration in the set point especially in time-varying and non-linear systems. Conventional control approaches are not convenient to solve the complex issues in highly non-linear system. To overcome the difficulties innate in controlling liquid level, a controller based on fuzzy logic can solve these issues.

Fuzzy controller is easier than PID controller for controlling nonlinear process. Because it eliminated the overshoot and significantly reduced the steady state error in just simple step by adding two more rules instead of changing the parameters like PID controller.

Finally, Fuzzy controller is less expensive to design than conventional and microprocessor control systems. Because it's a means of controlling with sentences (rules) rather than complex mathematical equations (typically deferential equations) which requires large amounts of memory (RAM or ROM) to incorporate lookup tables and it does require a more (processing power) to handle the computation needs of the mapping function .

5.2 Recommendations

From this study, the result showed that, though the fuzzy controller is cheaper and easier to implement than PID controller but the PID controller have the advantage on speed (rise time) because the fuzzy controller takes time for defuzzification process. So it's recommended to avoid (CoG) defuzzification method because it takes time for subtracting computing. Or use the benefit of two controllers and use it as one unit (fuzzy-PID controller).

The design of fuzzy logic controllers is based on the "operator's understanding" of the behavior of the process instead of its detailed mathematical model to conclude the rules and this can be a disadvantage of fuzzy logic controller if the rules were chosen wrong and this happened when the first three rules was fired, but the problem was solved by just adding two more rules. So it's recommended to choose the rules carefully.

Using fuzzy controller in SCADA system in water distribution system especially here in Sudan would be the best application for fuzzy control, because it will reduce the generating power that requires for water distribution, because the motors will operate only when the set point change, so it will save energy cost.

REFERENCES

[1] Mc. Neill, F. Martin and T. Allen, “PID Design”, Academic Press. International Journal of Man Machine Studies, vol.8, pp.669-678, 1997.

[2] A. Shome and D. Ashok, “Fuzzy Logic”, International Journal of Scientific and Engineering Research, vol.7, pp.1-6, 2012.

[3] Elmer P. Dadios, “Fuzzy Logic Controls”, Concepts, Theories and Applications", First Edition, Janeza Trdine 9, 51000 Rijeka, Croatia,vol.2 428pp, 2012.

[4] Mc. Neill, F. Martin and T. Allen, “PID Design”, Academic Press. International Journal of Man Machine Studies, vol.8, pp.669-678, 1997.

[5] Xiao,“Fuzzy Adaptive PID Control Tank Level”, IEEE, International Conference on Multimedia Communications,vol.7, p149-152, 2013.

[6] V.Novák, I.Perfilieva and J.Močkoř, “Mathematical principles of PID Controller”, Kluwer Academic, vol.2, pp.1-6, 2012.

[7] Mc. Neill, F. Martin and T. Allen, “PID Design”, Academic Press. International Journal of Man Machine Studies, vol.8, pp.220, 1997.

[8] Driankov D, Hellendoorn H, Reinfrank , “An introduction to PID Control”, Springer, vol.3, pp.145, Mar. 2002.

[9] T. Saric, R. Lujic , “PID Controllers Applications”, Technical Gazette, International Journal of Emerging Science and Engineering, vol. 30, pp.115-118, 2006.

[10] Chang and Shih, “PID Controller Design of Nonlinear Systems”, vol.1, pp.15-21, 2009.

[11] Yanwei, Hui and Huidang, Z. ‘PID Controller Parameters Tuning in Servo System’, Vol.2, pp.776-778, Mar. 2010.

[12] Hiyama T, Kugimiya M, Satoh H, "Advanced Fuzzy Controller for liquid Level Control", pp.3134-39, 2005.

[13] P. J. King and E. H. Mamdani, "The application of fuzzy logic control systems to industrial processes", Automated, vol. 11, pp. 235-212, 1997.

[14] Fang, Kwok, "Automatic Fuzzy Function", vol.1, pp.15-21, 2009.

[15] V.Novák, I.Perfilieva and J.Močkoř, "Mathematical principles of fuzzy logic", Vol.1, pp.776-778, Mar. 2010

[16] D. Wu, F. Karray, I. Song, "Water level control by Fuzzy Logic and Neural Networks", IEEE Conference on Control Applications, pp.3134-39, 2005.

[17] Q. Li, Y. Fang, J. Song, J. Wang, "The application of fuzzy control in liquid level system", "IEEE/ICMTMA", vol.4, pp.126-128, Mar. 2012.

[18] L.X.Wang, "Stable adaptive fuzzy control of nonlinear systems", in transactions on fuzzy system, vol.1, pp.146-155, 1993.

[19] P.H.Seong, 'Application of a fuzzy logic controller to level control', vol.166, pp. 426-428, 2012.

[20] Mitsumoto M and Zimmermann, "Comparison of fuzzy reasoning

methods", vol.4, pp.629–634, 2000.

[21] Sehab, Remy and Renotte, "An Approach to Desing Fuzzy PI Supervisor for Nonlinear System", vol. 11, pp. 220-224, 2000.

[22] Benjamin and Menlo Park, "self-organizing fuzzy logic controller", vol.167, pp.345-356, 1997.

[23] V. Galzina, T. Saric and R. Lujic , "Expert systems and fuzzy systems", Technical Gazette, vol. 1, pp.15-21, 2003.

[24] Dawson JG, Gao Z, "Fuzzy logic control of linear systems with variable

time delay", vol.1, pp.135-138, 1993.

[25] Cox E, "The fuzzy systems hand book" ,a practitioners guide to building using and maintaining fuzzy system, pp.53-56, 2005.

[26] Q. Li, Y. Fang, J. Song, J. Wang, "The application of PID control in liquid level system, vol.7, pp.54-58, Mar. 2010.

[27] A. Shome and D. Ashok, "Fuzzy Logic", International Journal of Scientific and Engineering Research, vol.7, pp.1-6, 2012.

[28] V.Novák, I.Perfilieva and J.Močkoř, "Mathematical principles of fuzzy logic", Vol.1, pp.776-778, Mar. 2010.

[29] Buckley J, Siler W, Tucker FLOPS, "a fuzzy expert system", applications and perspectives, vol.167, pp.310-316, 2007.

[30] E. H. Mamdani, "Automated fuzzy knowledge base generation and tuning," Automated, vol. 11, pp. 235-212, 1997.

[31] D. Wu, F. Karray, I. Song, "Water level control by Fuzzy Logic and Neural Networks", IEEE Conference on Control Applications, pp.3134-39, 2005.

[32] Castillo O, Melin, "A new fuzzy inference system for reasoning with multiple differential equations",vol.20, pp.145-151, 1999.

[33] Cavallo A, Setola R, Vasca F, "Using Matlab, Simulink and control system toolbox",vol.12, pp.65-68, 2012.

[34] Driankov D, Hellendoorn H, Reinfrank M, "An introduction to fuzzy control, vol.4, pp.429–434, 1996.

[35] Mendel JM, "Fuzzy logic systems for engineering: a tutorial", IEEE Conference on Control Applications, pp.120-121, 1993.

[36] Cavallo A, Setola R, Vasca F, "Using Matlab, Simulink and control system toolbox",vol.16, pp.35-40, 2014.

[37] Kugimiya M, Satoh H, "Fuzzy Inference System", vol.34, pp34-39, 2006.

APPENDICES

Appendix A

```
%ANIMTANK Animation of water tank system.

function [sys,x0]=animtank(t,x,u,flag,ts) %#ok<INUSL>

global tankdemo

if flag==2,
    if any(get(0,'Children')==tankdemo),
        if strcmp(get(tankdemo,'Name'),'Tank Demo'),

            % Update tank one level
            tankHndlList=get(tankdemo,'UserData');
            yData=get(tankHndlList(1),'YData');
            yOffset=yData(1);
            yData(3:4)=[1 1]*u(2)+yOffset;
            set(tankHndlList(1),'YData',yData);

            yData=get(tankHndlList(2),'YData');
            yData([3 4])=[1 1]*u(2)+yOffset;
            set(tankHndlList(2),'YData',yData);

            yData=[1 1]*u(1)+1;
            set(tankHndlList(3),'YData',yData);

            drawnow;
        end
    end
    sys=[];
    x0=[];

elseif flag == 4 % Return next sample hit

    % ns stores the number of samples
    ns = t/ts;

    % This is the time of the next sample hit.
    sys = (1 + floor(ns + 1e-13*(1+ns)))*ts;
    x0=[];

elseif flag==0,

    % Initialize the figure for use with this simulation
    fuzzy_animinit('Tank Demo');
    tankdemo=findobj(0,'Name','Tank Demo');

    tank1Wid=1;
    tank1Ht=2;
    tank1Init=0;
```

```
    setPt=0.5;

    tankX=[0 0 1 1]-0.5;
    tankY=[1 0 0 1];
    % Draw the tank

line(1.1*tankX*tank1Wid+1,tankY*tank1Ht+0.95,'LineWidth',2,'C
olor','black');
    tankX=[0 1 1 0 0]-0.5;
    tankY=[0 0 1 1 0];
    % Draw the water
    waterX=tankX*tank1Wid+1;
    waterY=tankY*tank1Init+1;
    tank1Hndl=patch(waterX,waterY,'blue','EdgeColor','none');
    % Draw the gray wall
    waterY([1 2 5])=tank1Ht*[1 1 1]+1;
    waterY([3 4])=tank1Init*[1 1]+1;
    tank2Hndl=patch(waterX,waterY,[.9 .9
.9],'EdgeColor','none');
    % Draw the set point
    lineHndl=line([0 0.4],setPt*[1
1]+1,'Color','red','LineWidth',4);

    set(gcf, ...
        'Color',[.9 .9 .9], ...
        'UserData',[tank1Hndl tank2Hndl lineHndl]);
    set(gca, ...
        'XLim',[0 2],'YLim',[0 3.5], ...
        'XColor','black','YColor','black', ...
        'Box','on');
    axis equal
    xlabel('Water Level
Control','Color','black','FontSize',10);
    set(get(gca,'XLabel'),'Visible','on')

    sys=[0 0 0 2 0 0];
    x0=[];

end;
```